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. © 1998 Elsevier Science Ltd. All rights reserved

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. 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.

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: 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: 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.
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\(^2\): 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\(^3\): These methods 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 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 \( C(t_p) \) and its target position \( P_j \), where \( t_p \) represents the parameter value of point \( C(t_p) \). Express these two points by the following NURBS form\(^4\):

\[
\sum_{i=1}^{n+2} N_i, (t_p) W_i B_i = P_j - C(t_p)
\]

where \( W_i \) and \( W'_i \) express the weights for control points \( B_i \) before and after curve modification, \( N_i, \) 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:

\[
\sum N_i, (t_p) W_i \sum N_i, (t_p) (W_i + \Delta W_i) \Delta B_i
\]

Multiply the above equation by \( \sum N_i, (t_p) W_i \) to give

\[
\sum N_i, (t_p) (W_i + \Delta W_i) \Delta B_i + \sum N_i, (t_p) \Delta W_i B_i
\]
Modification of a NURBS curve: Wei-Cheng Lu and Jiung-Ming Huang

\[ -\sum N_{i,k}(t_p) \Delta W_i \sum N_{i,k}(t_p) W_i B_i = \sum N_{i,k}(t_p) W_i (P_j - C(t_p)) + \sum N_{i,k}(t_p) \Delta W_i (P_j - C(t_p)) \]

Since

\[ C(t_p) = \frac{\sum N_{i,k}(t_p) W_i B_i}{\sum N_{i,k}(t_p) W_i} \]

the above equation becomes

\[ \sum N_{i,k}(t_p) (W_i + \Delta W_i) \Delta B_i + \sum N_{i,k}(t_p) \Delta W_i (B_i - P_j) = \sum N_{i,k}(t_p) W_i (P_j - C(t_p)) \]  

Equation (2) expresses the weight changes \( \Delta W_i \) and control point movements \( \Delta B_i \) to move a curve point from curve point \( C(t_p) \) toward point \( P_j \).

Nose points

In this paper, it is assumed that there is only a single curve point \( \Delta C(t_p) \) 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-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 identity and modify these nose points, the nose points \( \Delta C(t_p) \) 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

\[ \sum N_{i,k}(t_p) W_i \Delta B_i = \sum N_{i,k}(t_p) W_i (P_j - C(t_p)) \]  

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

\[ N_{i+1,k}(t_p) (W_i + \Delta W_i) \Delta B_i + \sum N_{i,k}(t_p) \Delta W_i (B_i - P_j) = N_{i+1,k}(t_p) W_i (P_j - C(t_p)) \]  

Rearrange the above equation and give

\[ \Delta B_{i+1} = \frac{\sum N_{i+1,k}(t_p) W_i (P_j - C(t_p))}{N_{i+1,k}(t_p) W_i} \]

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 \( B_2 B_1 \) when a control point \( B_2 \) is moved to \( B_1' \).

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:
Modification of a NURBS curve: Wei-Cheng Lu and Jiung-Ming Huang

\[ b_1 \Delta B_1 + b_2 \Delta B_2 + b_3 \Delta B_3 = a_1 \Delta C(t_{p_1}) \]
\[ b_2 \Delta B_2 + b_3 \Delta B_3 + b_4 \Delta B_4 = a_2 \Delta C(t_{p_2}) \]
\[ \vdots \]
\[ b_n \Delta B_n + b_{n+1} \Delta B_{n+1} + b_{n+2} \Delta B_{n+2} = a_n \Delta C(t_{p_n}) \]

Equation (6)

where \( a_i = \sum n_i(t_{p_i}) W_i b_{n,i} = N_{n,i}(t_{p_i}) W_i \).

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

\[
\begin{bmatrix}
  b_{1,2} & b_{1,3} & 0 & \cdots & 0 \\
  b_{2,2} & b_{2,3} & b_{2,4} & 0 & \cdots & 0 \\
  0 & 0 & \cdots & \cdots & \cdots & \cdots \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  0 & b_{n-1,2} & b_{n-1,3} & b_{n-1,4} & b_{n-1,5} & 0 \\
  0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
  0 & b_{n,2} & b_{n,3} & b_{n,4} & b_{n,5} & \cdots & 0 \\
  0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  0 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}
\begin{bmatrix}
  \Delta B_1 \\
  \Delta B_2 \\
  \vdots \\
  \Delta B_n \\
  \Delta B_{n+1} \\
\end{bmatrix}
= \begin{bmatrix}
  a_1 \Delta C(t_{p_1}) \\
  a_2 \Delta C(t_{p_2}) \\
  \vdots \\
  a_n \Delta C(t_{p_n})
\end{bmatrix}
\]

Equation (7)

The control point movements \( \Delta B_i \) 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 \( P_t \)s 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 \( t : [0 0 0 1.855 2.843 3.287 3.287 3.287 3.287 3.287] \), 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

\[
\sum_{i=1}^{n+2} n_i(t_{p_i}) W_i \Delta B_i = \sum_{i=1}^{n+2} n_i(t_{p_i}) W_i (P_j - C(t_{p_i}))
\]

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 \( P_j \):

\[
\Delta W_i j_{i,j} + \Delta W_{i+1,j} j_{i+1,j} + \Delta W_{i+2,j} j_{i+2,j} + \sum_{i,j=1}^{n+2} n_i(t_{p_i}) W_i (P_j - C(t_{p_i}))
\]

where

\[
I_{ij} = n_i(t_{p_i})(B_i - P_j)
\]

Express the above equation by the following form:

\[
F_1 + F_2 + F_3 = \sum_{i,j=1}^{n+2} n_i(t_{p_i}) W_i \Delta C(t_{p_i})
\]

where

\[
F_{i-1,i+1} = n_i(t_{p_i}) \Delta W_i (B_i - P_j)
\]

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 \( P_i \) is pulled by three forces, yielding displacement \( \Delta C(t_{p_j}) \) as illustrated in Figure 4.

Let \( u_i \) be the unit tangent vector of \( B_{i+1} - P_i \), \( v_i \) be the unit tangent normal to \( u_i \), and each nose point \( C(t_{p_j}) \) be selected on line \( P_iB_{i+1} \), a scalar expression in \( u_i \)-direction for equation (9) can be written below:

\[
\Delta W_i = \sum_{j=i}^{j+2} N_j(t_{p_j}) W_i \Delta C(t_{p_j})
\]

where \( \Delta C(t_{p_j}) = u_j \Delta C(t_{p_j}) \)

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

\[
(N_i(t_{p_j}) \Delta W_i)(B_{i+1} - P_i) \times u_i = (N_j(t_{p_j}) \Delta W_j)(B_{i+1} - P_i) \times u_i
\]

Rewrite the above equation into the following form

\[
\frac{\Delta W_{i+2}}{\Delta W_i} = \frac{(B_{i+1} - P_i) \times u_i}{(B_{i+1} - P_i) \times u_i} N_j(t_{p_j})
\]

To get true solutions of weight changes \( \Delta W_i \), 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 \( \Delta W_i \) in \( u_i \)-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 \( \Delta C(t_{p_j}) \) 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. \( \Delta W_i = \Delta W_{i+2} = 0 \). From the above two constraints, \( n \) equations with \( n \) unknown weight changes \( \Delta W_i \) expressed by equation (10) can be written as the following matrix form:

\[
\begin{bmatrix}
J_{1,1} & J_{1,2} & 0 & \cdots & 0 \\
J_{2,1} & J_{2,2} & b_{2,4} & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & J_{n-1,n-1} & J_{n-3,n-3} & \cdots & 0 \\
0 & 0 & 0 & \cdots & J_{n,n}
\end{bmatrix}
\begin{bmatrix}
\Delta W_1 \\
\Delta W_2 \\
\Delta W_{n+1}
\end{bmatrix}
= \begin{bmatrix}
\sum_{i=1}^{n+2} N_i(t_{p_j}) W_i \Delta C(t_{p_j}) \\
\sum_{i=1}^{n+2} N_i(t_{p_j}) W_i \Delta C(t_{p_j}) \vdots \\
\sum_{i=1}^{n+2} N_i(t_{p_j}) W_i \Delta C(t_{p_j}) \\
\sum_{i=1}^{n+2} N_i(t_{p_j}) W_i \Delta C(t_{p_j})
\end{bmatrix}
\]

The new weights should always remain positive for modifying a NURBS curve. However, weight changes \( \Delta W_i \) 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

\[
\Delta W = [4.680 2.967 -0.400 -2.418 -0.757 0.847 \\
-0.693 -0.937 -0.923 -0.302 -0.898 \\
-0.470 -0.793 -0.522 -0.467 -0.236]
\]

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
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 \( u \)-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 \( \Delta W'_i \) be the difference of \( \Delta W_i \); the following equation can be derived from equation (10) to express the relationship between the differences of weight changes.

\[
\Delta W'_i \cdot J_{i,1} + \Delta W'_i \cdot J_{i,2} + \Delta W'_i \cdot J_{i,3} = 0 \quad (14)
\]

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](image)

**Figure 7** Curve modification in need of allocation of weight changes.
Modification of a NURBS curve: Wei-Cheng Lu and Jie-Ming Huang 259

[Process 1]:

For(i=2;i<=n+1;i++)
{
\{ \\
\quad \text{if the weight for control point of segment } i-1 \text{ will become negative according to equation (13)} \\
\quad \text{then:} \\
\quad \quad \Delta W_{i-1} = -AW_{i-1} \\
\quad \begin{align*}
\text{Allocate difference of } \Delta W \text{ for segment } i-1 \\
\quad \quad (\Delta W_{i-1}, \Delta W_{i-2}, \Delta W_{i-3}) \text{ according to equation (14), where } d_B = 1 \\
\quad \end{align*}
\quad \text{make sure the weight changes for segment } i-1 \text{ are positive} \\
\quad \text{For(j=i-1;j>=1;j--) make } \\
\quad \quad \Delta W_{i-1} = \Delta W_{i-1} + \Delta W_{i-2} + \Delta W_{i-3}; \\
\text{calculate weight changes} \\
\text{For(j=i+1;j+1<=n+1;j++) make } \\
\quad \quad \Delta W_{i+1} = - \Delta W_{i-1} + \Delta W_{i-2} + \Delta W_{i-3}; \\
\text{calculate weight changes} \\
\text{For(j=1;j<=n+1;j++) compute weight changes} \\
\}
\}

algorithm of Process 1, equation (10) will still keep true, but the curve point will deflect slightly in the \( v \)-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

\[
\Delta W = [19.417 5.333 -0.400 0.000 -0.176 3.264 \\
-0.097 -0.152 1.033 -0.276 -0.852 0.724 \\
1.411 -0.923 -0.072 0.157]
\]

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

Solutions of \( \Delta W \) for both \( u \)- and \( v \)-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 \( u \)-direction. Instead, they may also move in the \( v \)-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:

\[
W_{i-1} + v_j + W_{i+1} + 2v_j - v_j = 0
\]

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 \( v \)-direction. If the nose point movements are evident in the \( v \)-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 \( \Delta W_{i+1} \) for the nose control point to a constant, and allocate the two weight changes \( \Delta W \) and \( \Delta W_{i+2} \) by combining both equations (10) and (12). These two equations are combined into the following form:

\[
\Delta W_{i+1} + \Delta W_{i+2} = \Delta W_{i+1} + \Delta W_{i+2}
\]

where \( \Delta W^* \) is the new value of \( \Delta W \), and

\[
d_j = \frac{|V_{i+j} \times u_j|}{|V_{i+j} \times u_j|} \frac{N_{i+j}(t_{p_j})}{N_{i+j+2}(t_{p_j})}
\]

Finally, the governing weight changes for the curve segment become

| Table 1 Results of curve modification for five times of \( \Delta W \) |
|-----------------|-----------------|-----------------|-----------------|
| \( W \): weight vector | \( \Delta W \): weight change vector | \( Err \): error vector of nose points |
| Initial model | \( W_{[1 1 1 1 1]} \) | \( \Delta W_{[0.519 2.717 2.717 1.788]} \) | \( 8.960, 5.530, 8.275, 0.430 \) | \( 0.763, 1.513, 6.535, 2.591 \) |
| 1 Weight change | \( W_{[1 1 1 1 1]} \) | \( \Delta W_{[0.570 2.718 2.718 1.788]} \) | \( 1.860, 1.360, 4.780, 1.138 \) | \( 1.280, 1.329, 4.851, 0.197 \) |
| 2 Weight changes | \( W_{[1 1 1 1 1]} \) | \( \Delta W_{[0.356 3.501 2.159 1.329]} \) | \( 1.287, 1.315, 4.862, 0.184 \) | \( 1.287, 1.315, 4.862, 0.184 \) |
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).

\[ \Delta W^*_{j+1} = \Delta W_{j+1} \]

\[ \Delta W^*_{j+2} = d \Delta W^*_j \]  

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

\[ [X] = [0 \ 0 \ 0 \ \Delta_{i_1} \ \Delta_{i_2} \ \Delta_{i_3} \ldots \ \Delta_{i_{14}} \ldots \] 

where \( \Delta_{i_1} = \Delta_1 + \Delta_2 \ldots + \Delta_i, \ldots \) 

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.

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.
Modification of a NURBS curve: Wei-Cheng Lu and Jiung-Ming Huang

![Diagram](a) Initial curve

(b) Modified curve without constraining knot vector $\Delta W$

(c) Modified curve with joining points constrained $\Delta W / X$

**Figure 8** The joining points $F_i$ and knot spans $\Delta_i$ relationship.

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

<table>
<thead>
<tr>
<th>$W$: weight vector</th>
<th>$\Delta W$: weight change vector</th>
<th>$X$: knot vector</th>
<th>$Err$: error vector of nose points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial fitted curve</td>
<td>$W$: [1 1 1 1 1 1]</td>
<td>$X$: [0 0 0 1 1.855 2.952 3.044 3.136 3.387 3.287]</td>
<td>$Err$: [8.960, 5.550, 8.275, 0.396]</td>
</tr>
<tr>
<td>1 Weight change</td>
<td>$\Delta W$: [0.5 0.57 0.59 0.717 0.778]</td>
<td>$X$: [0 0 1 1.860 2.340 2.518 2.518]</td>
<td>$Err$: [7.584, 4.038, 7.932, 4.507]</td>
</tr>
<tr>
<td>2 Weight changes</td>
<td>$\Delta W$: [-0.321 0.354 -0.321 0.354 -0.321]</td>
<td>$X$: [0 0 0 1 1.882 2.340 2.518 2.518]</td>
<td>$Err$: [7.788, 3.319 5.195 0.734]</td>
</tr>
<tr>
<td>3 Weight changes</td>
<td>$\Delta W$: [0.067 -0.139 -0.156 0.073]</td>
<td>$X$: [0 0 0 1 1.868 2.340 2.397 2.397]</td>
<td>$Err$: [1.384, 2.069, 2.692, 0.690]</td>
</tr>
<tr>
<td>4 Weight changes</td>
<td>$\Delta W$: [0.067 -0.139 -0.156 0.073]</td>
<td>$X$: [0 0 0 1 1.845 2.163 2.312 2.312]</td>
<td>$Err$: [1.391, 2.172, 2.305, 0.903]</td>
</tr>
<tr>
<td>5 Weight changes</td>
<td>$\Delta W$: [0.067 -0.139 -0.156 0.073]</td>
<td>$X$: [0 0 0 1 1.823 2.113 2.260 2.260]</td>
<td>$Err$: [1.402, 2.402, 2.682, 0.903]</td>
</tr>
</tbody>
</table>
From the above observations, a number of joining points constrained to weight changes $\Delta W/X$ followed by control point movements $\Delta B$ 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 $\Delta Err: [(e_{i,1}, \Delta e_{i,2}, \ldots, \Delta e_{i,n})$; the improvement efficiency is defined as:

$$\rho = \frac{\Delta e_{i,1} + \Delta e_{i,2} + \ldots + \Delta e_{i,n}}{e_{i-1,1} + e_{i-1,2} + \ldots + e_{i-1,n}}$$  \hspace{1cm} (22)

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

When the improvement efficiency $\rho$ has been less than a certain value or even become negative, stop.

---

**Figure 9** Flow chart for NURBS curve modification.
the ΔWX 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.](image)

**Table 3 Improvement efficiencies of (ΔWX) × 4 (mm)**

<table>
<thead>
<tr>
<th>No. of ΔWX modification</th>
<th>E_b</th>
<th>E_C</th>
<th>ρ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modification for 1st time of ΔWX</td>
<td>0.456</td>
<td>2.765</td>
<td>0.973</td>
</tr>
<tr>
<td>Modification for 2nd time of ΔWX</td>
<td>0.218</td>
<td>0.542</td>
<td>0.731</td>
</tr>
<tr>
<td>Modification for 3rd time of ΔWX</td>
<td>0.183</td>
<td>0.364</td>
<td>0.280</td>
</tr>
<tr>
<td>Modification for 4th time of ΔWX</td>
<td>0.147</td>
<td>0.269</td>
<td>0.218</td>
</tr>
</tbody>
</table>
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 \( W = 1 \). 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 \( \rho_{\text{MIN}} \) is set to 0.25. The improvement efficiency values \( p \) for moving nose point B toward target position in the first, second, third and fourth times \( \Delta WX \) are illustrated in Table 3. \( p \) is smaller than \( \rho_{\text{MIN}} \), and the displacement of neighboring nose point C is smaller than the allowable value in the fourth time of \( \Delta WX \). Thus four times of the changes of weights and knot vector resulted from moving the control point using equation (7). Figure 10(b) and (d) show the nose point modification which resulted from changing the weights, while Figure 10(c) and (e) depict the nose point modification which resulted from moving the control point using equation (7).

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 \( \rho_{\text{MIN}} \) is set to 0.25. The improvement efficiency values \( p \) for moving nose point B toward target position in the first, second, third and fourth times \( \Delta WX \) are illustrated in Table 3. \( p \) is smaller than \( \rho_{\text{MIN}} \), and the displacement of neighboring nose point C is smaller than the allowable value in the fourth time of \( \Delta WX \). Thus four times of the changes of weights and knot vector resulted from moving the control point using equation (7). Figure 10(b) and (d) show the nose point modification which resulted from changing the weights, while Figure 10(c) and (e) depict the nose point modification which resulted from moving the control point using equation (7).

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


Appendix

For a third order NURBS curve, a curve segment can be expressed as:

\[
C'(t) = \sum_{m=i+2}^{n+3} \frac{N_{m,i}(t)W_m}{\sum_{m=1}^{n+3} N_{m,i}(t)W_m} B_m
\]

(A1)

the basis functions are formulated as:

\[
N_{m,i}(t) = \left( \frac{(t-x_m)N_{m,k-l}(t)}{x_{m+k-l} - x_m} + \frac{(x_m-t)N_{m,i+1-l}(t)}{x_{m+i-1} - x_m} \right)
\]

\[
= \frac{(t-x_m)N_{m,k-l}(t)}{\Delta_{m+k-l} + \Delta_{m+k-l-1} + \ldots + \Delta_{m-1}} + \frac{(x_m-t)N_{m,i+1-l}(t)}{\Delta_{m+i} + \Delta_{m+i-1} + \ldots + \Delta_{m-1}}
\]

(A2)

\[
N_{m,i}(t) = \begin{cases} 
1 & \text{for } m = i + 2 \\
0 & \text{for other } m
\end{cases}
\]

(A3)

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

\[
[X] = \begin{bmatrix} x_1, x_2, x_3, \ldots, x_{n+3}, x_{n+4}, R_{n+5} \end{bmatrix}
\]

\[
= [0, 0, x_1, x_2, x_3, x_4, x_5, \ldots, x_{n+3}, x_{n+4}, x_{n+5}]\]

(A4)
From equations (A2) and (A3), the basis functions on knot \( x_{i+1} \), for parameter range \( x_{i+2} \leq t < x_{i+3} \):

\[
N_{i+1}(x_{i+1}) = \frac{(x_{i+2} - x_{i+1})N_{i+1,0}(x_{i+1})}{\Delta_i + \Delta_{i+1}}
\]

\[
N_{i+2,1}(x_{i+1}) = \frac{(x_{i+3} - x_{i+2})N_{i+2,1}(x_{i+2})}{\Delta_i + \Delta_{i+1}}
\]

\[
N_{i+2,2}(x_{i+1}) = \frac{(x_{i+3} - x_{i+2})N_{i+2,2}(x_{i+3})}{\Delta_i + \Delta_{i+1}}
\]

where span \( \Delta_i = x_{i+2} - x_{i+1} \). Substituting equations (A7) and (A8) for equations (A5) and (A6)

\[
N_{i+1}(x_{i+1}) = \frac{\Delta_{i+1}}{\Delta_i + \Delta_{i+1}}
\]

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

\[
F_{i+1} = C(x_{i+1}) = \frac{N_{i+1,0}(x_{i+1})W_{i+1}}{\sum \limits_{m=1}^{\nu} N_{m,0}(x_{i+1})W_m} B_{i+1}
\]

+ \[
\sum \limits_{m=1}^{\nu} \frac{N_{m,0}(x_{i+1})W_m}{\sum \limits_{m=1}^{\nu} N_{m,0}(x_{i+1})W_m} W_{i+1} B_{i+1}
\]

\[
\Delta_{i+1} = \frac{\Delta_{i+1}}{\Delta_i + \Delta_{i+1}} W_{i+1} B_{i+1}
\]

Rearrange equation (A11) into

\[
\Delta_{i+1} = \frac{W_{i+2}(B_{i+2} - F_{i+1})}{W_{i+2}(F_{i+2} - B_{i+1}) - \Delta_{i+1}} W_{i+2} B_{i+2}
\]

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

For a non-rational NURBS curve \((W=1)\), rather, the knot point (feature point) is formulated as:

\[
F_{i+1} = C(x_{i+1}) = \frac{\Delta_{i+1}}{\Delta_{i+1} + \Delta_i} B_{i+1}
\]

Once \( \Delta_i \) is given, \( \Delta_{i+1} \) will be substantiated following

\[
\Delta_{i+1} = \frac{B_{i+2} - F_{i+1}}{F_{i+2} - B_{i+1}} \Delta_i
\]

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

\[
\Delta_{i+1} = \frac{W_{i+2}(B_{i+2} - F_{i+1})}{W_{i+2}(F_{i+2} - B_{i+1}) - \Delta_{i+1}} W_{i+2} B_{i+2}
\]

A general solution for \( \Delta_i \) can be written as

\[
\Delta_i = \frac{W_{i+1}(B_{i+1} - F_i)}{W_i F_i - B_i} \left( \frac{B_i - F_i}{F_i - B_i} \right)^\Delta_i
\]

for \( i \geq 2 \). The knot vector is designated:

\[
[X] = [0 \ 0 \ 0 \ \Delta_{i+1} \ \Delta_{i+2} \ \Delta_{i+3} \ \ldots \ \Delta_i \ \ldots \ \Delta_{i+2} \ \Delta_{i+3} \ \ldots \ \Delta_{i+1} \ \Delta_{i+2} \ \Delta_{i+3} \ \ldots \ \Delta_{i+1} \ \ldots \ \Delta_{i+1}]
\]

where \( \Delta_j = \Delta_j + \ldots + \Delta_i + \Delta_j \).