Weight Control and Knot Placement for Rational B-spline Curve Interpolation

Tae-Wan Kim*, Kunwoo Lee
Computer Aided Design and Analysis Laboratory, School of Mechanical and Aerospace Engineering, Seoul National University

We consider an interpolation problem with nonuniform rational B-spline curves given ordered data points. The existing approaches assume that weight for each point is available. But, it is not the case in practical applications. Schneider suggested a method which interpolates data points by automatically determining the weight of each control point. However, a drawback of Schneider's approach is that there is no guarantee of avoiding undesired poles; avoiding negative weights. Based on a quadratic programming technique, we use the weights of the control points for interpolating additional data. The weights are restricted to appropriate intervals; this guarantees the regularity of the interpolating curve. In addition, a knot placement is proposed for pleasing interpolation. In comparison with integral B-spline interpolation, the proposed scheme leads to B-spline curves with fewer control points.

Key Words : Computer Graphics, Computer Aided Design, Rational B-splines, Curves, Interpolation

1. Introduction

NURBS stands for Nonuniform Rational B-Splines. They first appeared in the context of geometric modeling in Computer Aided Design/ Computer Aided Manufacturing (CAD/CAM) (Cho and Yang, 1995), Computer Graphics, and Computer Aided Geometric Design (CAGD) in Vesprille's Ph. D. thesis in 1975. As NURBS curves and surfaces have become an Initial Graphics Exchange Specification (IGES) standard in CAD/CAM industries, they are now utilized extensively, see (Lee, 1999).

A typical interpolation problem refers to the determination of control points from given data in two or higher dimensional spaces. In this paper, we study interpolation algorithms for defining B-spline curves from a sequence of discrete points. Possible applications of such algorithms include the generation of planar or spatial curves.

We give a brief survey over existing approaches to the NURBS interpolation problem. The following notation is used to represent the homogeneous coordinates $\underline{\mathbf{d}}_i=[\underline{d}_{i,1}, \underline{d}_{i,2}, \underline{d}_{i,3}, \underline{d}_{i,4}]^T\in R^4$ of a point $\mathbf{d}_i=[d_{i,1}, d_{i,2}, d_{i,3}]^T\in R^3$ with the associated weight $w_i\in R\backslash\{0\}$:

$$\underline{\mathbf{d}}_i=[w_i d_{i,1}, w_i d_{i,2}, w_i d_{i,3}, w_i]^T=[w_i\mathbf{d}_i, w_i]^T \quad (1)$$

A NURBS curve is given by the parametric representation

$$\underline{\mathbf{y}}(t)=\sum_{i=0}^{M}N_i^d(t)\underline{\mathbf{d}}_i \quad (2)$$

with the weighted control points $\underline{\mathbf{d}}_i$ and with the B-spline basis functions $N_i^d(t)$ of degree d which are defined over certain knots. See (Farin, 1996; Farin, 1999; Hoschek and Lasser, 1993) for more details.

For interpolation by NURBS, some authors

* Corresponding Author,
E-mail : tkim@cad.snu.ac.kr
TEL : +82-2-880-7447 ; **FAX** : +82-2-887-2283
Computer Aided Design and Analysis Laboratory, School of Mechanical and Aerospace Engineering, Seoul National University (Manuscript **Received** June 1, 2000; **Revised** November 16, 2000)

suggest that the weighted data points $\underline{\mathbf{x}}_j = [h\mathbf{x}_j h_j]^T$, $j=0, 1, \cdots, P$, where $h \neq 0$ is the weight of the point $\mathbf{x}_j \in R^3$ in 3D space, are interpolated by a curve in 4-D homogeneous space R^4, see (Farin, 1996; Piegl, 1990).

● *Standard Scheme.* Given a set of data points $\mathbf{x}_0, \cdots, \mathbf{x}_N \in R^3$ with the corresponding parameter values t_j and weights h_j find a C^2 rational cubic B-spline curve with control points $\mathbf{d}_i$ and weights w_i that interpolates to the given data points $\mathbf{x}_j$ and the weights of the data points h_j. For finding the NURBS curve, Farin (1996) solves a 4D interpolation problem for the data $\underline{\mathbf{x}}_j \in R^4$ with parameters t_j. The only difference compared to integral B-spline curves is the dimension of input and output, i. e., 4D instead of 3D. However, the estimation of appropriate weights h_i for the data is rather difficult. The choice $h_i=1$ will produce the result of the standard integral interpolation scheme, whereas other choices may produce poles (points at infinity) of the interpolating curve.

● *Piegl's Scheme.* Piegl (1987) suggests the use of tangent information at the sample points. Given a set of data points $\mathbf{x}_0, \cdots, \mathbf{x}_N \in R^3$, the corresponding derivatives $D_t\mathbf{x}_j$, parameter values t_j, and weights h_j, find an interpolating rational B-spline curve with control points $\mathbf{d}_i$, and weights w_i. The solution of this problem, as proposed by Piegl, is to interpolate the rational curve in the 4D homogeneous space from the 4D weighted points $[h_j\mathbf{x}_j h_j]^T$ and their derivatives $D_t[h_j\mathbf{x}_j h_j]^T$. Since $D_t h_j x_j = (D_t h_j)x_j + h_j(d_t x_j)$, only the $D_t h_j$ need to be computed for further interpolation. Piegl suggests to compute a 1D spline $w(t)$ through the data weights h_j from which the $D_t h_j$ can be computed.

● *Schneider(1992)'s Scheme.* Given the data points $\mathbf{x}_0, \cdots, \mathbf{x}_N \in R^3$ with the parameters t_i, the interpolation conditions for the curve $\underline{\mathbf{y}}(t) = [\underline{y}_1(t)\ \underline{y}_2(t)\ \underline{y}_3(t)\ \underline{y}_4(t)]^T$ are derived from

$$\underline{y}_4(t_i)\begin{bmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \end{bmatrix} - \begin{bmatrix} \underline{y}_1(t_i) \\ \underline{y}_2(t_i) \\ \underline{y}_3(t_i) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

Inserting the parametric representation (2) into these conditions leads to a homogeneous system of $3(N+1)$ linear equations for the $4(M+1)$ unknowns $\underline{d}_{i,1}, \cdots, \underline{d}_{i,4}$. If the number M of B-spline control points $\mathbf{d}_i$, fulfills $3N=4M$, then we get at least one non-trivial solution. In comparison with the above two methods, Schneider's scheme may interpolate the data using a curve with fewer control points. Moreover, it does not need the weights h_i of the data points; these weights are eliminated by interpolating more points. Schneider's method also applies to the interpolation of given tangent directions, see (Schneider, 1992).

Both standard and Piegl's approaches suppose that the weights for the data points are available. This is however not the case for practical applications. A major drawback of Schneider's approach is the difficulty of avoiding curve points at infinity (poles), i. e., zeros of the weight function $\underline{y}_4(t)$. We overcome these difficulties by restricting the weights of the control points to appropriate intervals. The interpolating spline curve is now computed by solving a simple quadratic programming problem which is one of the standard problems in mathematical optimization. The solution is invariant with respect to affine transformations of the data. Moreover, it is shown that an appropriate knot distribution guarantees the uniqueness of the solution. The interpolating NURBS curves generally require fewer control points than the corresponding integral spline curve. As observed in our numerical experiments, NURBS interpolation often produces less oscillating curves, resulting from the smaller number of control points. As the main new feature of our interpolation scheme, we show how to make real use of the additional degrees of freedoms provided by using NURBS instead of integral spline curves. This results in a more complex mathematical problem: whereas interpolation by integral spline curves leads to banded linear systems, our scheme requires the solution of a relatively simple optimization problem. So, the interpolation by NURBS is much more complicated than standard integral techniques.

2. The Interpolation Problem

We consider the following problem. A sequence of $N+1$ data points $\mathbf{x}_0, \cdots, \mathbf{x}_N \in R^3$ in 3D space is given. We want to find an interpolating NURBS curve $\underline{\mathbf{y}}(t)$ in homogeneous coordinates, see (Hoschek and Pottmann, 1995), which interpolates the data. We denote by d and S the degree and the number of polynomial segments of the NURBS curve, respectively. Whereas the degree d is to be chosen by the user (e. g., from continuity requirements: the spline curve will d-1 times continuously differentiable), we want to choose the minimal number S of spline segments.

To be more explicit, one needs to compute the following unknowns merely from the set of $N+1$ points for solving the above problem:

- t_j: the parameters for the given points ($j=0, \cdots, N$),
- S and u_i: the number of spline segments and the knot sequence ($i=0, \cdots, S$), and
- $\underline{\mathbf{d}}_i$: the weighted control points ($i=0, \cdots, M$) with $M=S+d-1$.

In order to avoid non-linear equations, the unknowns t_j, S and u_i are estimated from the data at first, as described below. The $4(M+1)$ components of the weighted control points $\underline{\mathbf{d}}_i$ remain unknown.

3. Parameterization Methods

For practical applications, there are four commonly used parameterization methods to assign the parameters t_j : i. e., uniform, cumulative chord length, centripetal model and Foley's parameterization, see (Hoschek and Lasser, 1993). The uniform parameterization yields overall poor results if two data points are near each other and the next point is significantly farther away because it does not consider the geometry of the data points. Chord length parameterization (which approximates the arc length of a parametric curve) usually produces better results than uniform parameterization, although not in all cases. The centripetal parameterization method as proposed by Lee (1989) was motivated by the paradigm of a car traveling through the data points $\mathbf{x}_j$. One would like to slow down for corners and travel at a constant speed on linear stretches. Foley and Nielson (1989) developed an affinely invariant parameterization scheme which produces good results in many cases.

4. Knot Placement

The B-spline basis functions $N_i^d(t)(i=0, \cdots, M)$ are defined over the knot sequence

$$(\underbrace{u_0, \cdots, u_0}_{(d+1)\text{times}}, u_1, u_2, \cdots, u_{S-1}, \underbrace{u_S, \cdots, u_S}_{(d+1)\text{times}}) \quad (4)$$

By choosing $d+1$ fold boundary knots, we obtain a NURBS curve which interpolates its first and last control point. For more information on B-splines, see (de Boor, 1978). We propose a knot placement scheme in which the knots are distributed as equally as possible. We associate with the i-th B-spline basis function $N_i^d(t)$ the number r_i,

$$r_i=\min(d+1, i+1, M-i+1) \quad (5)$$

of spline segments where the function is nonzero. For each spline segment $[u_i, u_{i+1}]$, the sum

$$s_i=\sum_{j=1}^{i+d}\frac{1}{r_j} \text{ for } i=0, \cdots, S-1 \quad (6)$$

(where j runs through the indices of all basis functions which are nonzero in that interval) represents the average number of degrees of freedom for this spline segment. For segments which are sufficiently far away from the boundaries, we get $s_i=1$, but s_i will be greater for segments close to the boundaries. This is due to the fact that a spline curve has more degrees of freedom at the boundary than in the inner segments.

As an example, we consider a knot sequence with $d=3$ and $S=5$. We get $r_0=r_7=1$, $r_2=r_6=2$, $r_3=r_5=3$, and $r_3=r_4=4$. Hence, $s_0=s_4=\frac{25}{12}\approx 2.08$, $s_1=s_3=\frac{4}{3}\approx 1.33$, and $s_2=\frac{7}{6}\approx 1.17$.

For choosing the knots $u_0, \cdots, u_S$, we consider the piecewise linear function $f(a)$: $[0, 1] \rightarrow [t_0, t_N]$ with

$$f(a)=(Na-(j-1))t_{j-1}+Na-j)t_j$$

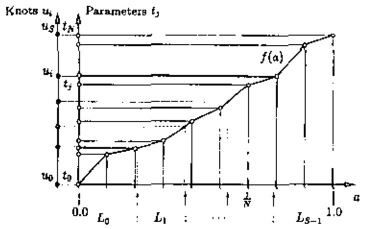

Fig. 1 Knot placement by piecewise linear interpolation

for $a \in \left[\frac{j-1}{N}, \frac{j}{N}\right]$ $(j=1, \cdots, N)$ (7)

Figure 1 shows the knot placement.

This function satisfies $f\left(\frac{j}{n}\right)=t_j$ and it is monotonically increasing as the parameters obtained from the above methods are monotonic. We subdivide the unit interval into S segments with the lengths $L_0, \cdots, L_{S-1}$ satisfying

$$L_0: L_1: L_2: \cdots: L_{S-2}: L_{S-1}=(s_0-\sigma): s_1: s_2: \cdots: s_{S-2}: (s_{S-1}-\sigma) \text{ and } L_0+L_1+L_2+\cdots+L_{S-2}+L_{S-1}=1 \quad (8)$$

where σ is a certain constant $0<\sigma<1$, e.g.,$\sigma=0.5$. This correction is introduced as the first and the last de Boor points $\underline{\mathbf{d}}_0$, $\underline{\mathbf{d}}_M$ are almost fixed by the first and the last data point, only the weights are still available. The knots u_i are now chosen according to

$$u_i=f\left(\sum_{j=0}^{i-1} L_j\right), \ (i=0, 1, \cdots, S), \quad (9)$$

in particular $u_0=t_0$ and $u_S=t_N$. In this way, we obtain a reasonable knot sequence, where the distribution of the parameters t_j reflects the number of degrees of freedom of the spline segments.

5. Interpolation Conditions and Weight Restrictions

The NURBS curve $\underline{\mathbf{y}}(t)$, see (Hoschek and Pottmann 1995), is to interpolate the given data points $\mathbf{x}_j$. Provided that the weight function $\underline{y}_4(t)$ does not vanish at $t=t_j$, this is equivalent to Schneider's interpolation conditions (3). After substituting (2) into (3), we get the $3(N+1)$ linear conditions:

$$[A_{0,j}|A_{1,j}|\cdots|A_{M,j}]\begin{bmatrix}\underline{\mathbf{d}}_0\\ \underline{\mathbf{d}}_1\\ \vdots\\ \underline{\mathbf{d}}_M\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix} \ (j=0, \cdots, N) \quad (10)$$

where

$$A_{i,j}=\begin{bmatrix}-N_i^d(t_j) & 0 & 0 & N_i^d(t_j)x_{i,1}\\ 0 & -N_i^d(t_j) & 0 & N_i^d(t_j)x_{j,2}\\ 0 & 0 & -N_i^d(t_j) & N_i^d(t_j)x_{j,3}\end{bmatrix}$$
$$(i=0, \cdots, M) \quad (11)$$

for the weighted control points $\underline{\mathbf{d}}_i$.

Schneider (1992) used these linear conditions in order to build a system of linear equations for the control points, but this approach cannot avoid negative weights of the resulting control points. In our approach, we use the interpolation conditions as linear equality constraints in a simple optimization problem. In addition, we consider the linear inequality constraints for the weights

$$\varepsilon \leq \underline{d}_{i,4}=\omega_i \leq \frac{1}{\varepsilon} \text{ for } i=0, \cdots, N \quad (12)$$

where the constant $0<\varepsilon \ll 1$ has to be specified by the user e. g., $\varepsilon=0.01$. Thus, we restrict the weights of the control points to a "reasonable" interval.

Note that (12) is a sufficient condition which guarantees $\varepsilon \leq \underline{y}_{i,4}(t) \leq \frac{1}{\varepsilon}$. One may derive "tighter" (less sufficient) constraints by considering the weights of the NURBS representation of the curve with respect to a refined knot vector. This, however, increases the number of required linear inequalities.

6. Quadratic Programming

After computing the parameters t_i of the given data points, we choose the number of segments according to

$$S_0=\left[\frac{3}{4}N-d+1\right] \quad (13)$$

For $S=S_0$, the number of unknown components of the control points equals the number $3(N+1)$ of interpolation conditions (10). After generating the knots $u_0, \cdots, u_S$, we consider the equality constraints (10) and the weight inequalities (12). Using the simplex algorithm (or a similar tool from Linear Optimization, e. g., the LOQO package which is available from http://www.princeton.edu/~rvdb/loqoexecs.html) we check whether solutions to the linear equality and inequality constraints exist. If this is not the case, then we increase the number S of segments by 1 and try again.

According to the variation diminishing property of NURBS curves with positive weights, an arbitrary plane P intersects the curve (Jung, 1994) not more often than the control polygon (Farin, 1996; Hoscheck and Lasser, 1993). Thus, a NURBS curve (2) possesses M such intersections at most. On the other hand, consider the polygon formed by the data points $\mathbf{x}_0, \cdots, \mathbf{x}_N$. An arbitrary plane P intersects the curve at least as many times as the data polygon. Let K be the maximal number of intersections of a plane and the data polygon. As a necessary condition for the existence of solutions to the interpolation problem (11), we find $M \geq K$, hence $S \geq K-d+1$. Thus, the initial value of the above-described iteration may be chosen as $\max(S_0, K-d+1)$.

For $S=N-d+1$ at the latest we will find solutions (provided the knots u_i fulfill the Schoenberg-Whitney conditions, see (de Boor, 1978)), as this choice corresponds to the usual polynomial interpolation scheme. However, in most cases one may expect to obtain interpolating spline curves with fewer segments.

After identifying a suitable number S of segments for the interpolating NURBS curve, we compute the solution by minimizing the quadratic objective function

$$\sum_{i=0}^{M}(\omega_i-\alpha_i)^2 \rightarrow Min \tag{14}$$

subject to the linear constraints of (10) and (12). The α_i in (14) are user-defined positive constants. If we set $\alpha_0=\cdots=\alpha_M=1$, then we try to choose the solution by minimizing the objective function weight as close to 1 as possible, i. e., as close to an integral curve as possible.

This optimization problem (14) can be solved exactly in finite time, e. g., with simplex-type algorithms as proposed in the textbook by Fletcher (1990). Alternatively, one may use an interior point method like that of the LOQO package.

The solution obtained from our scheme has the following properties:

- In the general case, our interpolating NURBS curve requires fewer segments than the corresponding integral interpolating spline curve.
- It is unique, provided that the Schoenberg-Whitney conditions, see (de Boor, 1978), are fulfilled for an appropriate subset of the data points. One needs to consider the subset, as we generally have more data points than B-spline basis functions, $N \geq M$.
- The solution is invariant with respect to affine mappings of the data, provided that we keep the parameters t_i and the knots u_j. This results from the fact that the weights of the control points are affinely invariant.

We illustrate the interpolation scheme by two examples. Both examples are 2D data sets which are interpolated by planar NURBS curves, simply by choosing $\underline{d}_{i+3}=0$ for all control points. In the planar case, the interpolation conditions (10) yield only $2(N+1)$ equations (as the third equations are trivially fulfilled) for the $3(M+1)$ remaining components $\underline{d}_{i,j}$, $j\neq 3$, of the control points. Thus, for choosing

$$S_0=\left[\frac{2}{3}N-d+1\right] \tag{15}$$

the number of unknown components equals the number of interpolation conditions. The data for both examples is taken from (Schneider, 1992), and $\varepsilon=0.01$ has been chosen. We used LOQO for solving the quadratic programming problems.

Figure 2 shows the interpolation of Ω-shaped data with a centripetal parameterization and $\alpha_0=\cdots=\alpha_M=1$. Whereas the integral spline curve (a) needs 10 segments, the NURBS curve has only 7 segments. The weights of the NURBS curve vary within the interval [0.01, 2.38], the diameters of the control points (marked by black dots) indicate the weights on a logarithmic scale. The grey dots are the data points, whereas the circles on the spline curve represent the knots (i. e., the segment

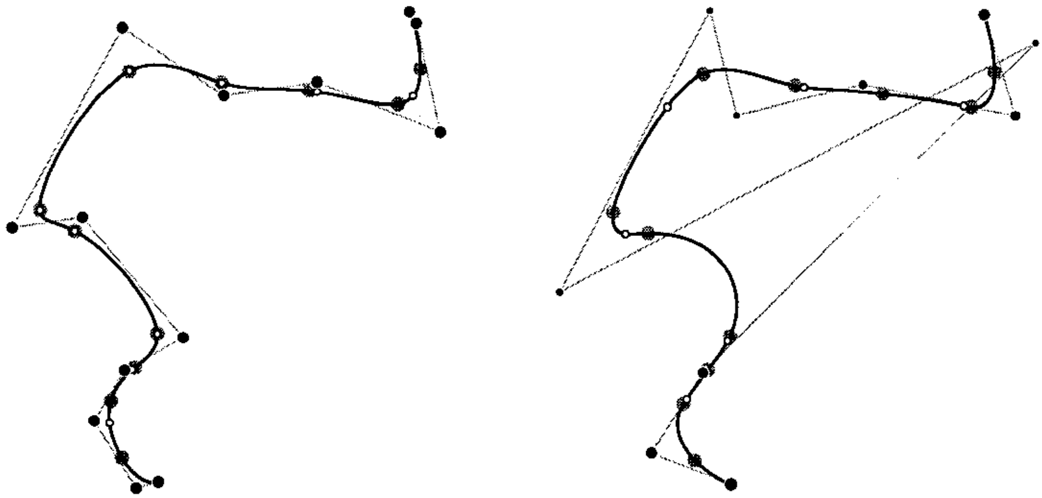

Fig. 2 Interpolation of Ω-shaped data with centripetal parameterization, $N=12$. Integral cubic spline curve (a) with 10 segments, $M=12$. Cubic NURBS spline curve (b) with 7 segments, $M=9$

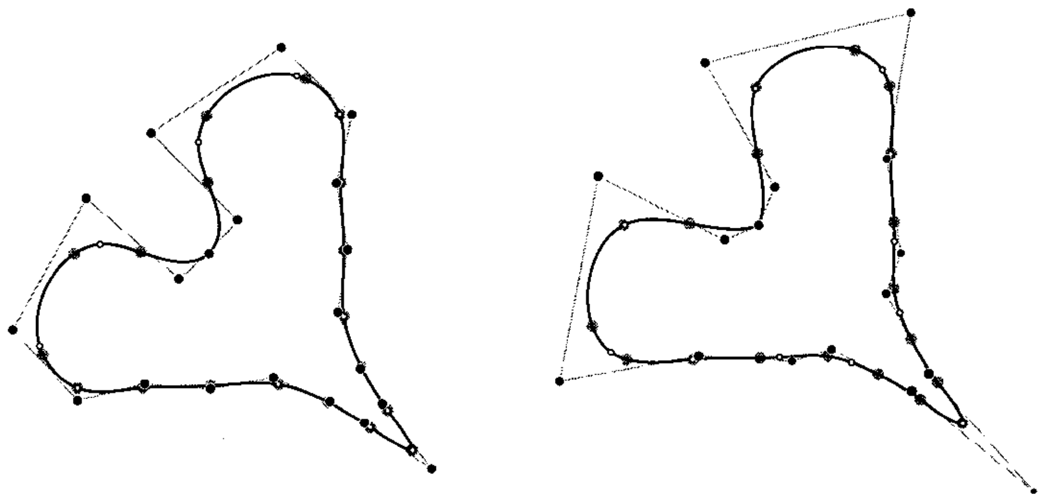

Fig. 3 Interpolation of heart-shaped data with chordal parameterization, $N=20$. Integral cubic spline curve (a) with 18 segments, $M=20$. Cubic NURBS spline curve (b) with 14 segments, $M=16$

boundaries) of the spline curve.

Figure 3 shows the interpolation of heart-shaped data with a chordal parameterization and $\alpha_0=\cdots=\alpha_M=1$. Whereas the integral spline curve (a) has 18 segments, the NURBS curve (b) needs only 14 segments. The weights of the NURBS curve vary within the interval [0.37, 1.29]. Thus, no constraints of the quadratic programming problem are active for the solution. Again, the diameters of the control points (marked by black dots) indicate the weights, but the logarithmic scale is different from that of the previous example.

7. Conclusion

The presented research was motivated by the question for a NURBS interpolation scheme which makes reasonable use of the additional degrees of freedom provided by rational representations, i. e., of the weights. We avoided undesired shape of Schneider's method, i. e., poles of a resulting rational B-spline curve by utilizing quadratic programming. The result of interpolation is depending on knot placement from location parameters of given data points. To determine location parameters, both common chord and centripetal parameterization methods worked well without big difference. Knot placement is more important than determination of location parameters of data points. The proposed scheme for knot placement produces a "pleasing" curve. In comparison with standard integral spline interpolation, our scheme leads to spline curves with fewer segments and control points. Fewer control points lead to a curve which is smoother than one with more control points.

A challenging problem is to assign target weights α_i; $i=0, 1, \cdots, M$ in objective function of quadratic programming. Extension of the quadratic programming to rational surface interpolation is a future work. For the surface interpolation, parameterization is generally more difficult than one for the curve case.

Acknowledgement

The authors thank Bert Juettler for his help. The authors acknowledge the financial support from the Korean Ministry of Science and Technology through National Research Laboratory grant.

References

Cho, H. D. and Yang, M. Y., 1995, "A Study on the Five-Axis End Milling for Sculptured Surfaces," *KSME International Journal*, Vol. 9, No. 4, pp. 428~437.

de Boor, C., 1978, *A Practical Guide to Splines*, Springer, New York.

Farin, G. E., 1996, *Curves and Surfaces for CAGD: A Practical Guide*, Academic Press, Boston, USA, (4th ed.).

Farin, G. E., 1999, *NURBS from Projective Geometry to Practical Use*. A K Peters, Natick, 2nd edition.

Fletcher, R., 1990, *Practical Methods of Optimization*, Wiley, Chichester, UK, (2nd ed.).

Foley, T. and Nielson, G., 1989, "Knot Selection for Parametric Spline Interpolation," in Lyche, T. and Schumaker, L. L. (eds.), *Mathematical Methods for Curves and Surfaces*, Academic Press, Boston, USA, pp. 261~271.

Hoschek, J. and Lasser, D., 1993, *Fundamentals of Computer Aided Geometric Design*, AK Peters, Wellesley, USA.

Hoschek, J. and Pottmann, H., 1995, "Interpolation and Approximation with Developable B-Spline Surfaces," in Daehien, M., Lyche, T. and Schumaker, L. L. (eds.), *Mathematical Methods for Curves and Surfaces*, Vanderbilt University Press, Nashville, USA, pp. 255~264.

Jung, H. B., 1994, "A Method for Surface/Surface Intersection," *KSME International Journal*, Vol. 9, No. 1, pp. 51~67.

Lee, E., 1989, "Choosing Nodes in Parametric Curve Interpolation," *Computer Aided Design*, Vol. 21, No. 6, pp. 363~370.

Lee, K., 1999, *Principles of CAD/CAM/CAE Systems*, Addison-Wesley.

Piegl, L., 1987, "Interactive Data Interpolation by Rational Bezier Curves," *IEEE Comp. Graphics and Appl.*, Vol. 7, No. 4, pp. 45~58.

Piegl, L., 1990, "On NURBS: A Survey," *IEEE Comp. Graphics and Appl.*, Vol. 11, No. 1, pp. 55~71.

Schneider, F. -J., 1992, Interpolation, Approximation und Konvertierung mit Rationalen B-Spline Kurven und Flächen, Ph. D. thesis, TH Darmstadt, Germany.

Vesprille, K., 1975, Computer Aided Design Applications of the Rational B-Spline Approximation Form, Ph. D. thesis, Syracuse Univ.