1998

A Universal Parametrization in B-Spline Curve and Surface Interpolation and Its Performance Evaluation.

Choong-gyoo Lim
Louisiana State University and Agricultural & Mechanical College

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

A UNIVERSAL PARAMETRIZATION IN B-SPLINE CURVE AND SURFACE INTERPOLATION AND ITS PERFORMANCE EVALUATION

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Computer Science

by
Choong-Gyoo Lim
B.S., Seoul National University, 1988
M.Ed., Seoul National University, 1990
May 1998

To My Parents

Acknowledgments

First of all, I'd like to thank Dr. Sukhamay Kundu for his constant effort to lift the quality of this dissertation in every aspect. I'd also like to thank Dr. Warren J. Waggenspack, Jr. for his excellent teaching in Computer Graphics and Computer Aided Geometric Modeling. I appreciate his suggestions throughout this work.

Finally, I'd like to thank my parents for my education which they made possible with thier priceless effort.

Contents

List of Tables

List of Figures

Abstract

The choice of a proper parametrization method is critical in curve and surface fitting using parametric B-splines. Conventional parametrization methods do not work well partly because they are based only on the geometric properties of given data points such as the distances between consecutive data points and the angles between consecutive line segments. The resulting interpolation curves don't look natural and they are often not affine invariant. The conventional parametrization methods don't work well for odd orders k. If a data point is altered, the effect is not limited locally at all with these methods. The localness property with respect to data points is critical in interactive modeling. We present a new parametrization based on the nature of the basis functions called B-splines. It assigns to each data point the parameter value at which the corresponding B-spline $N_{ik}(t)$ is maximum. The new method overcomes all four problems mentioned above;

(1) It works well for all orders k, (2) it generates affine invariant curves, (3) the resulting curves look more natural, in general, and (4) it has the semi-localness property with respect to data points, The new method is also computationally more efficient and the resulting curve has more regular behavior of the curvature.

Fairness evaluation and knot removal are performed on curves obtained from various parametrizations. The results also show that the new parametrization is superior. Fairness is evaluated in terms of total curvature, total length, and curvature plot. The curvature plots are looking natural for the curves obtained from the new parametrization. For the curves obtained from the new method, knot removal is able to provide with the curves which are very close to the original curves. A more efficient and effective method is also presented

for knot removal in B-spline curve. A global norm is utilized for approximation unlike other methods which are using some local norms. A geometrical view makes the computation more efficient.

Chapter 1
Introduction

1.1 Curve and Surface Interpolation

The following *interpolation* problem occurs often in Computer Aided Geometric Modeling[12, 18, 28].

Definition 1.1 Interpolation Problem

Given a list of data points $P_i \in R^m$, $0 \leq i \leq n$, one needs to find a *smooth* curve or surface which goes through the points P_i. A typical smoothness condition is that the curve has continuous derivatives of degree up to $k-1$ for the interpolating curve of degree k, $k \geq 0$

Figure 1.1 shows some examples of interpolating curve and surface. There are several curve (surface) modeling methods available in the literature, which include polynomial, Lagrangean, Hermitian, Bézier, and B-spline. In this dissertation, B-spline is studied to get better curves and surfaces.

B-spline is a piecewise polynomial approach to curve fitting. A B-spline curve is a linear combination of the basis functions called B-splines which are themselves piecewise polynomials with a certain smoothness condition. B-spline has been frequently used due to its maximum smoothness a piecewise approach can have[18, 28, 30]. It has continuous derivatives of degree up to $k-1$ when the B-spline curve is of degree $k(\geq 0)$. Another reason for its popular use is the localness property[18, 28, 30]. Localness means that the curve is defined locally. In other words, a local segment of curve is constructed by a unique set of factors which are different in other segments. Practically, it means that if a control point d_j is replaced by a new control point d'_j, then the new B-spline curve differs from the original curve only around a few neighboring points. This is critical in interactive curve

(a) A B-spline curve

(b) A B-spline surface

Figure 1.1: A B-spline curve and a surface via interpolation
$*$: interpolating points

modeling. Note that it does not mean that if a point P_i is moved to a new position P_i', then the curve remains unchanged except its neighborhood.

1.2 Applications

We often face the problem of curve and surface fitting in many scientific areas. A good method for curve and surface fitting helps us doing many things effectively such as interpolation, data compression, and data enlargement. Its purpose is to get a best possible curve and surface in terms of closeness to the original curve and surface with possibly less memory space to represent it. Curve and surface fitting is also found in modeling the shapes of automobiles and airplanes[12, 18, 28, 30]. It has been successfully utilized in signal and image processing as well[29, 54, 55, 57]. It can be used to define a vector font if the resulting curve is affine invariant.

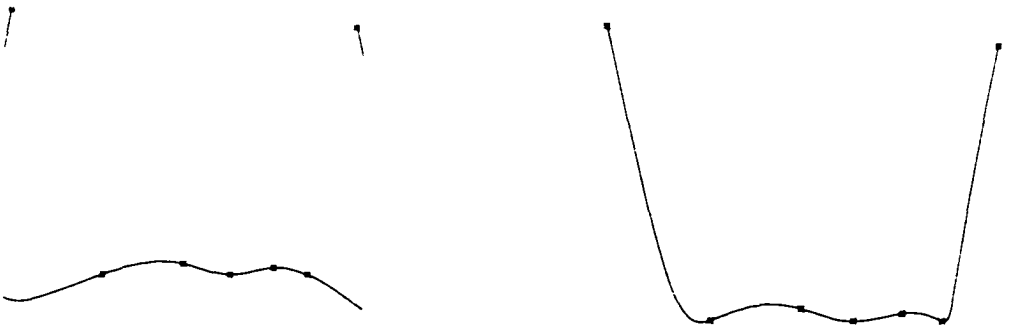
1.3 Previous Studies and Related Problems

There are two different curve and surface representations, namely, implicit and parametric representations. Parametric representation has some computational advantages over implicit representation. Thus, parametric B-spline curve and surface is the mathematical

expression for curve and surface which is being used in many applications. Once a list of data points is given, we need to assign a parameter value to each data point, which is called *parametrization*. And, the resulting curve $X(t)$ from B-spline interpolation is affected heavily by parametrization. Many works have been done on parametrization. Some of well-known parametrizations in the literature are equidistant, chordal, centripetal, and Foley parametrizations[12, 18, 28, 33].

There are four major problems with the existing parametrization methods in B-spline interpolation :

1. The curves and surfaces obtained often do not look natural. It is illustrated in Figure 1.2 and 1.3. A undesired longer curve is obtained in Figure 1.2(a). In Figure 1.2(b), sharp corners are not so expected. The surface has too many peaks in Figure 1.3(c). It is obvious in its contour image as shown in Figure 1.3(d). Equidistant parametrization is used with the tensor-product method for surface interpolation. In some cases, there are too many wiggles and bumps.
2. The resulting curves are not affine invariant[20, 28, 43] or transformation invariant in a more general sense. The curves are not preserved under affine transformations as shown in Figure 1.4. The affine transformation here is scaling with a factor of n along the y-axis. The resulting curves are not affine invariant through chordal parametrization or centripetal parametrization. The curves are changed along the x-axis when the data points are scaled only along the y-axis. The interpolation curves are not so natural to us as human beings. Some authors consider lack of affine invariancy to be a deficiency[44].
3. The conventional methods do not work well for odd orders k. The quality of curve is so poor in the resulting curves of orders 3 and 5 as shown in Figure 1.5. The curves



(a) chordal parametrization (b) Foley parametrization

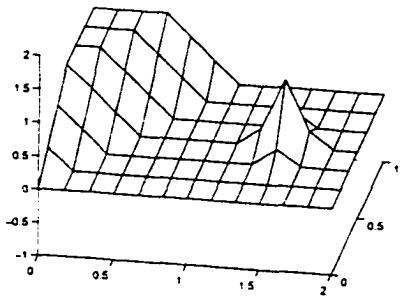
Figure 1.2: Curves don't look natural with conventional methods

were obtained from Foley parametrization. It seems that they do not work well for odd orders k.

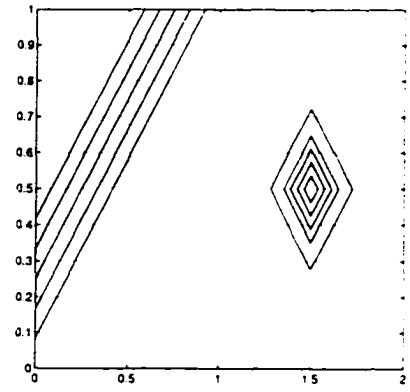
4. With conventional methods, the effect of a changed data point on the shape of curve is not locally limited at all. In Figure 1.6, P_4 is changed to P_4'. The effect is rather global.

Most of the conventional parametrizations make use of some geometric properties of given data points. Distances $\|P_i - P_{i-1}\|$ and angles between two line segments $\overline{P_iP_{i-1}}$ and $\overline{P_{i+1}P_i}$ are utilized. However, these methods don't work well in some cases partly because they don't take into account the nature of the basis B-spline functions $N_{i,k}(t)$ from which one builds the interpolation curve $X(t)$. We here attempt to rectify this problem by choosing parameter values in a special way. It assigns to each data point the parameter value where the corresponding $N_{i,k}(t)$ is maximum.

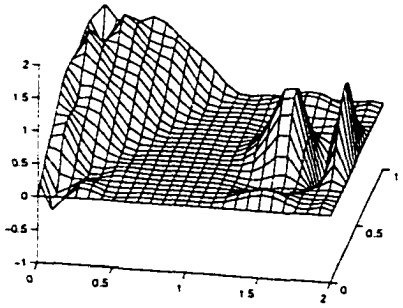
We here present a new parametrization which works well in all orders k. The new method provides us with transformation invariant curves while the resulting curves look natural. Also, the resulting curves have the semi-localness property with respect to data points. In Chapter 2, we present some important definitions in B-spline interpolation, mentioning



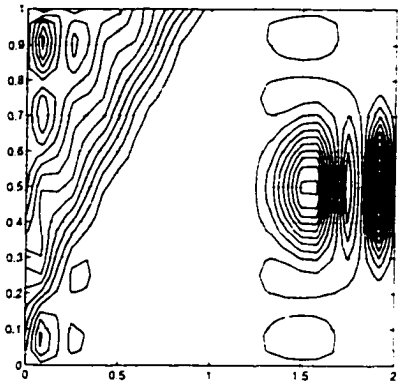
(a) Data points on surface



(b) Its contour

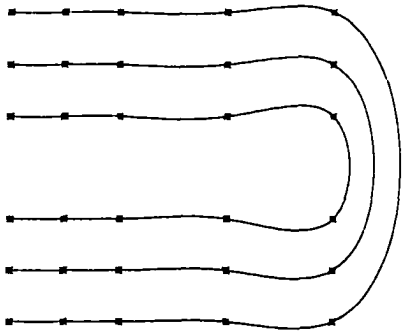


(c) Equidistant parametrization

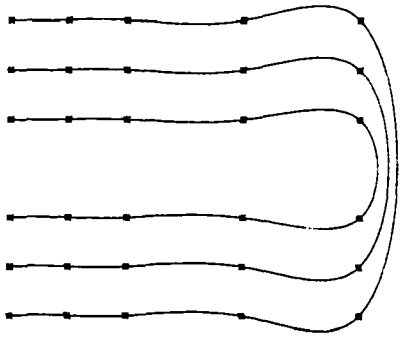


(d) Its contour

Figure 1.3: Surfaces don't look natural with conventional methods

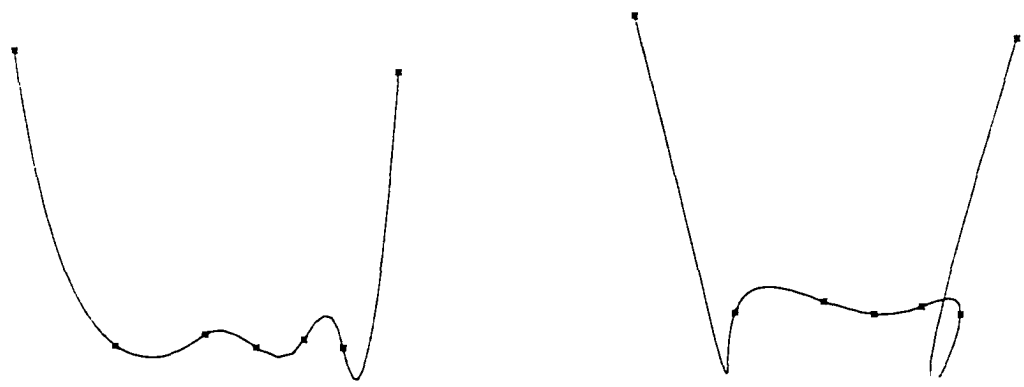


(a) chordal parametrization



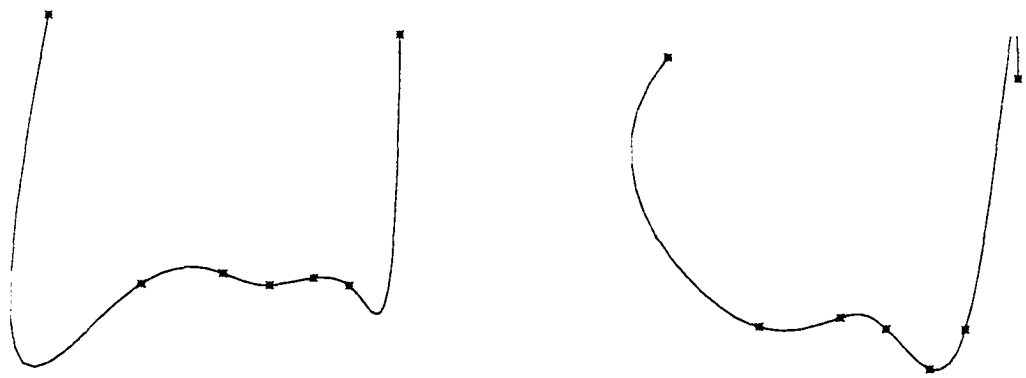
(b) centripetal transformation

Figure 1.4: Curves are not preserved under an affine transformation
Data points are scaled up with a factor of n along the y-axis.



(a) Order 3 (b) Order 5

Figure 1.5: The conventional methods don't work well for odd orders k



(a) Order 3 (b) Order 5

Figure 1.6: The effect of an altered data point is not locally limited at all

about the role of the knot vector. The importance of knot vector selection is illustrated with some example curves.

A variety of parametrization methods are briefly explained in Chapter 2. Foley's affine invariant parametrization is discussed in detail. A new parametrization is presented in Chapter 3. The new method is based on a very simple selection of parameter and knot values. It takes advantage of the nature of the basis functions B-splines, not the geometric properties of given data points. In Chapter 4, fairness[18] is evaluated on the curves obtained through various parametrizations. Total curvature and total length are computed for evaluation. Curvature plots are also drawn for accurate evaluation. Knot removal is applied to B-spline curves to get compact representations in Chapter 5. Two efficient and effective methods are presented to compute new de Boor points. We conclude in Chapter 6. A few directions for possible future research will be given.

1.4 Summary

B-spline is one of the most frequently used models for curve and surface in CAGM. It is because of its smoothness and localness properties. Parametric representation is preferred in the literature due to its easy manipulation. Parametrization is quite important in B-spline curve and surface interpolation because the quality of the curve is heavily affected by parametrization. The curves obtained from conventional parametrizations are not so natural in many cases. They don't work well in some orders k and the curves are not affine invariant. The conventional methods don't have the localness property with data points. A new parametrization will be presented to overcome these problems. The new method is based on the properties of the basis functions called B-splines, not based on the geometric properties of given data points. It appears that there are many important intrinsic properties of B-splines, which have been overlooked by the previous works.

Chapter 2
Conventional Parametrizations

There are two representation schemes for curve and surface in Computer Aided Geometric Modeling. One is implicit and the other is parametric representation[18, 26, 28, 30, 42]. Because parametric curves are easy to manipulate, in general, parametric representation is preferred in the literature[26, 42]. The parametric B-spline will be discussed in this dissertation for the same reason. With implicit representation, modeling operations such as union, intersection, and difference are easy.

Example 2.1 Examples of implicit representation:

$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ and $a'x + b'y + c' = 0$.

$$\text{intersection}: \quad \left(\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1\right)^2 + (a'x + b'y + c')^2 = 0.$$

$$\text{union}: \quad \left(\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1\right)(a'x + b'y + c') = 0.$$

However, curves and surfaces are not easy to construct or to transform with implicit representation because a complicated computation is often involved. On the other hand, curves and surfaces are easy to construct or to transform with parametric representation.

Example 2.2 Examples of parametric representation:

$X(t) = (a\cos(t), b\sin(t))$ and $X(s,t) = \sum a_{ij}N_{ik}(s)N_{jl}(t)$.

Some advantages and disadvantages are listed in Table 2.1. In practical applications, both representations are used simultaneously to make use of the advantages that they have on their own.

Once data points have been selected for interpolation, parametrization heavily affects the shape of curve. Some of well-known parametrizations are equidistant, chordal, centripetal,

Table 2.1: Comparison between parametric and implicit representations

	Parametric	Implicit
Computational Efficiency	O	X
Axis Independency	O	X
Shape Definition	O	△
Modeling Operations	X	O

and Foley[18, 28, 33]. Before going into the details of those conventional parametrizations, let's define B-splines, B-spline curve, and B-spline interpolation in a formal fashion.

2.1 B-Spline and Its Interpolation Problem

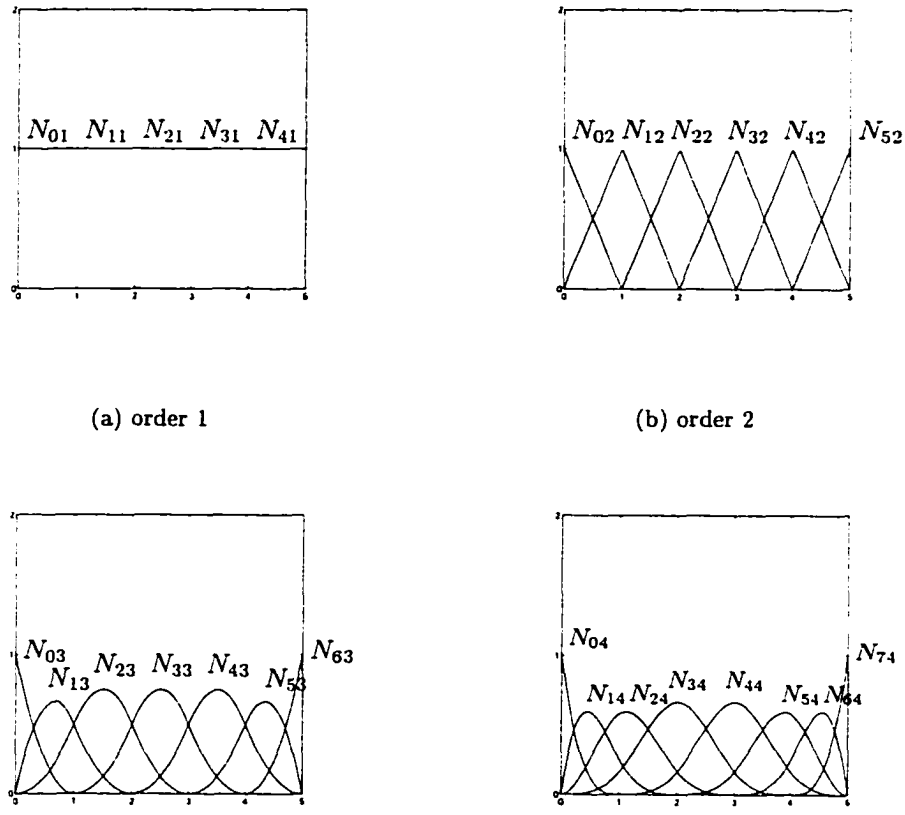
Let's define B-splines and B-spline curve first. Because of the parametric nature of B-spline, many works have been done on parametrization for B-spline interpolation[18, 20, 28, 33, 44]. However, our experimentation shows that the selection of knot vector is no less important than parametrization. Notice that the knot vector plays an important role in the following definitions. There are several different methods to define B-splines. B-splines have been defined and studied mainly through divided differences, convolution, recursion and dual functional[12, 28]. In this dissertation, recursion is used to define B-splines.

Definition 2.3 B-splines Given a knot vector $T = (T_0, T_1, \cdots, T_{n-1}, T_n, T_{n+1}, \cdots, T_{n+k})$, $k \geq 1$ and $n \geq 0$, the associated normalized B-splines N_{ik} of order k (of degree $k-1$) are defined by

$$N_{i1}(t) = \begin{cases} 1, & T_i \leq t < T_{i+1}, \\ 0, & \text{otherwise} \end{cases}$$

for $k = 1$ and

$$N_{ik}(t) = \frac{(t - T_i)}{(T_{i+k-1} - T_i)} N_{i,k-1}(t) + \frac{(T_{i+k} - t)}{(T_{i+k} - T_{i+1})} N_{i+1,k-1}(t)$$



(a) order 1

(b) order 2

(c) order 3

(d) order 4

Figure 2.1: Normalized B-splines

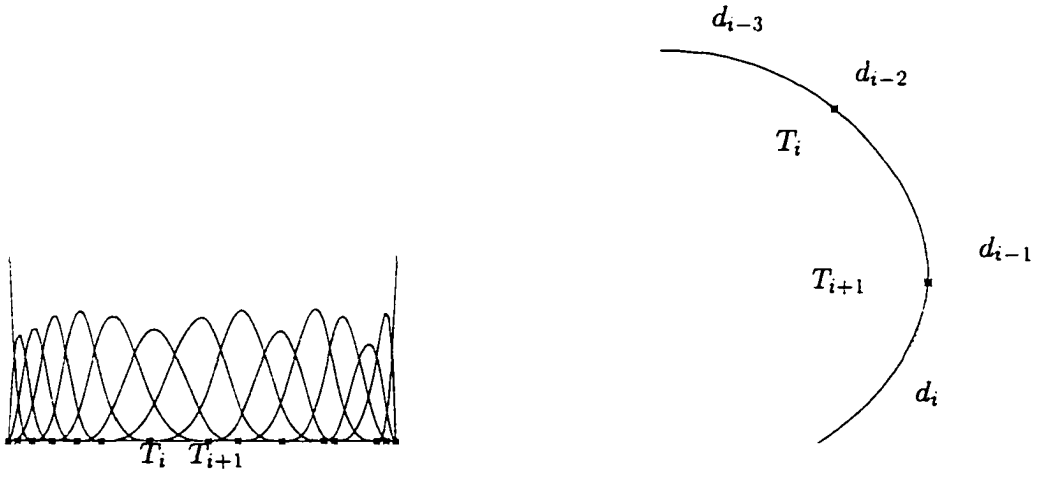
for $k > 1$ and $i = 0,\ 1,\ \cdots,\ n$.

Some example B-splines of various orders k are drawn in Figure 2.1.

Based on these B-splines, one could define a B-spline curve.

Definition 2.4 B-spline curve Given an ordered list of deBoor points (also called control points) $d_i \in R^m (m \geq 1),\ 0 \leq i \leq n$, and a knot vector T, then one defines a B-spline curve of order k by

$$X(t) = \sum_{i=0}^{n} d_i N_{ik}(t) \quad \text{for} \quad T_0 \leq t < T_{n+k}.$$



(a) B-splines

(b) B-spline curve

Figure 2.2: B-splines and B-spline curve

An example of 4th order B-splines is shown in Figure 2.2(a). A B-spline curve based on these B-splines is partially shown in Figure 2.2(b). Now, one could define the problem of B-spline interpolation as follows.

Definition 2.5 B-spline interpolation Given a list of data points $P_i \in R^m$, $0 \leq i \leq n$, the B-spline interpolation problem of order k is to find: (1) the knot vector $T = (T_0, T_1, \cdots, T_{n+k-1}, T_{n+k})$, (2) the parameter value t_i for each P_i, $0 \leq i \leq n$, and (3) the deBoor points d_i such that the resulting B-spline curve $X(t) = \sum_{i=0}^{n} d_i N_{ik}(t)$ has the property $X(t_i) = P_i$, $0 \leq i \leq n$.

The computation of B-spline interpolation is straightforward. First, we choose the knot vector $T = (T_0, T_1, \cdots, T_{n+k-1}, T_{n+k})$ so that B-splines $N_{ik}(t)$ can be defined. In open curve cases, the first k T_i's are equal and the last k T_i's are equal, too, i.e., $T_0 = T_1 = \cdots = T_{k-1}$ and $T_{n+1} = T_{n+2} = \cdots = T_{n+k}$ because the interpolating curve $X(t)$ often needs to have the property $X(t_{start}) = d_0$ and $X(t_{end}) = d_n$[28]. For the other knot values, equally-spaced values can be assigned, i.e., $T_i - T_{i-1} = constant$, for example.

Other selections of nondecreasing knot values are also possible. The parametrization method of choice assigns an appropriate parameter value t_i to each data point P_i. Once we have done parametrization, the following equation holds.

$$d_0 N_{0k}(t_i) + d_1 N_{1k}(t_i) + \cdots + d_n N_{nk}(t_i) = P_i, \ 0 \leq i \leq n.$$

In the matrix form, the equation becomes

$$Ad = P,$$

where

$$A = \begin{pmatrix} N_{0k}(t_0) & N_{1k}(t_0) & \cdots & N_{nk}(t_0) \\ N_{0k}(t_1) & N_{1k}(t_1) & \cdots & N_{nk}(t_1) \\ \vdots & \vdots & \vdots & \vdots \\ N_{0k}(t_n) & N_{1k}(t_n) & \cdots & N_{nk}(t_n) \end{pmatrix}, \tag{2.1}$$

$d = (d_0, d_1, \cdots, d_n)^T$ and $P = (P_0, P_1, \cdots, P_n)^T$. The matrix A is nonsingular when t_i's are distinct. d, the list of deBoor points, is obtained solving the above linear equation. i.e.. $d = A^{-1}P$.

2.2 B-Spline Computation

One can directly evaluate a B-spline curve, $X(t)$, computing $N_{ik}(t)$ first. Also, a more systematic method called deBoor algorithm is available.

2.2.1 Direct Computation

For direct computation, $N_{ik}(t)$ is first evaluated with a given knot vector T. The curve $X(t) = \sum_{i=0}^{n} d_i N_{ik}(t)$ is then computed as a linear combination of d_i's.

Example 2.6 For order $k = 4$, let $T = (0, 0, 0, 0, 1, 2, 3, 3, 3, 3)$. Then for $t = 1.5$,

$$N_{04}(1.5) = 0.0 \quad N_{14}(1.5) = 0.0312 \quad N_{24}(1.5) = 0.4688$$

$$N_{34}(1.5) = 0.4688 \quad N_{44}(1.5) = 0.0312 \quad N_{54}(1.5) = 0.0$$

At $t = 1.5$, the curve value is $X(1.5) = d_1N_{14}(1.5)+d_2N_{24}(1.5)+d_3N_{34}(1.5)+d_4N_{44}(1.5)$. Note that there are only four non-zero $N_{ik}(t)$'s in order 4 because of the localness property of B-spline.

2.2.2 The deBoor Algorithm

By the recurrence relation in Definition 2.3, a B-spline curve of order k, $X(t) = \sum_{i=0}^{n} d_i N_{i,k}(t)$, can be represented as a linear combination of B-splines of order $k-1$ as follows.

$$\begin{aligned} X(t) &= \sum_{i=0}^{n} d_i N_{i,k}(t) \\ &= \sum_{i=0}^{n} d_i \frac{t-T_i}{T_{i+k-1}-T_i} N_{i,k-1}(t) + \sum_{i=0}^{n} d_i \frac{T_{i+k}-t}{T_{i+k}-T_{i+1}} N_{i+1,k-1}(t) \\ &= \sum_{i=0}^{n} d_i \frac{t-T_i}{T_{i+k-1}-T_i} N_{i,k-1}(t) + \sum_{i=1}^{n+1} d_{i-1} \frac{T_{i+k-1}-t}{T_{i+k-1}-T_i} N_{i,k-1}(t) \\ &= \sum_{i=0}^{n+1} (\alpha_i d_i + (1-\alpha_i) d_{i-1}) N_{i,k-1}(t), \end{aligned}$$

where $d_{-1} = 0$, $d_{n+1} = 0$, and

$$\alpha_i = \frac{t-T_i}{T_{i+k-1}-T_i}.$$

By repeating the above process, the curve becomes

$$X(t) = \sum_{i=0}^{n+j} d_i^j N_{i,k-j}(t), \ j = 0, \ 1, \ \cdots, \ k-1,$$

where $d_i^j = (1-\alpha_i^j) d_{i-1}^{j-1} + \alpha_i^j d_i^{j-1}$, $j > 0$, with $\alpha_i^j = \frac{t-T_i}{T_{i+k-j}-T_i}$ and $d_i^0 = d_i$. At the final step, i.e., $j = k-1$,

$$X(t) = \sum_{i=0}^{n+k} d_i^{k-1} N_{i,1}(t).$$

For $t \in [T_r, T_{r+1})$, it becomes $X(t) = d_r^{k-1}$.

The above process can be represented systematically as in Table 2.2. Due to the localness of B-spline, we restrict ourselves to the deBoor points $d_{i-k+1}, \cdots, d_i$. The deBoor algorithm can be represented step by step as follows[28].

Table 2.2: Computation of the deBoor algorithm

$$
\begin{array}{lllll}
d_{r-k+1} = d^0_{r-k+1} & & & & \\
d_{r-k+2} = d^0_{r-k+2} & d^1_{r-k+2} & & & \\
d_{r-k+3} = d^0_{r-k+3} & d^1_{r-k+3} & d^2_{r-k+3} & & \\
\vdots & \vdots & \vdots & \cdots & \\
d_{r-1} = d^0_{r-1} & d^1_{r-1} & d^2_{r-1} & \cdots & d^{k-2}_{r-1} \\
d_r = d^0_r & d^1_r & d^2_r & \cdots & d^{k-2}_r \quad d^{k-1}_r
\end{array}
$$

Step 1. Find r with $T_r \leq t^* < T_{r+1}$.

Step 2. Let $\alpha_i^1 = \frac{t^* - T_i}{T_{i+k-1} - T_i}$, $i = r-k+2,\ r-k+3,\ \cdots,\ r$. and compute $d_i^1 = (1-\alpha_i^1)d_{i-1} + \alpha_i^1 d_i$.

Step 3. For $l = 2,\ 3,\ \cdots,\ k-1$ and for $i = r-k+l+1,\ r+k+l+2,\ \cdots,\ r$, let $\alpha_i^l = \frac{t^* - T_i}{T_{i+k-l} - T_i}$. Set $d_i^l = (1-\alpha_i^l)d_{i-1}^{l-1} + \alpha_i^l d_i^{l-1}$.

Step 4. The function value $X(t)$ at $t = t^*$ is $X(t^*) = d_r^{k-1}$.

2.3 Parametrizations with B-spline Curve

The *quality* of an interpolating curve $X(t)$ heavily depends on the selection of parametric values t_i for given data points P_i[18, 20, 28, 30]. There have been two types of parametrization techniques in the literature. One is the *data-oriented* parametrizations which try to take into account the geometric properties of given data points such as distance and angle. They are chordal, centripetal, and Foley parametrizations. The other is the *optimizing* parametrizations which attempt to optimize parameter values with certain objective functions[8, 24]. The curve energy or strain energy is often used as an objective function. However, the energy function is too complicated to be used in the general optimization technique. Total curvature is a major part of the energy function. A simplified

version of the energy function is frequently used, which is not a good approximation to the original energy function. This often results in a poor interpolating curve. The simplified energy function will be discussed in detail in Chapter 4.

2.3.1 Equidistant Parametrization

First of all, we could have equidistant or equally-spaced parameter values, i.e., $\Delta t_i = t_i - t_{i-1} = constant$. The *equidistant* parametrization is the most primitive one and performs poorly[18, 28, 33]. Knot values are also equally spaced in general. This method often comes up with an interpolating curve with loops which are highly undesirable[33].

2.3.2 Chordal Parametrization

The *chordal* parametrization selects the parametric values according to the distances $\|P_i - P_{i-1}\|$. The idea here is that if we think of parameter t as the time of travel on the curve, then Δt_i has to be proportional to the arc distances between the points P_{i-1} and P_i. The chordal distances approximate the arc distances.

Note that if $\|P_i - P_{i-1}\|$ is the same for all i, then it is, in fact, the equidistant parametrization. Chordal parametrization has been a dominant choice because of the common belief that arc length has to do something with parameter values and chordal length is close to arc length in many cases. To compute an exact arc length, some iterative methods have been attempted[12, 33]. No convergency is guaranteed in general.

However, this method often ends up with not so pleasing curves, especially when distances vary wildly[18, 28, 33].

2.3.3 Centripetal Parametrization

Chordal parametrization is an analogy to driving a car at a constant speed. However, one has to slow down at sharp corners. The car will skid off the road otherwise. To prevent the car from skidding off, more time is needed to go through. This is the physical analogy

that Lee's centripetal parametrization is based on[33]. It uses $\Delta t_i = \|P_i - P_{i-1}\|^\alpha$. ($\alpha = \frac{1}{2}$ is recommended in the literature[28].) The distances become larger if $\|P_i - P_{i-1}\| < 1$ and smaller otherwise. This means that the distances between data points at corners becomes larger because data points are usually dense at corners. However, the distances on flat areas also become larger if they happen to be small.

Usually, the interpolating curves stay close to the data point polygon, which the designer might have in mind. However, niether chordal nor centripetal parametrization is invariant under affine transformations. They are not invariant under scale transformations as shown in Figure 1.4.

2.3.4 Foley Parametrization

To make an affine invariant parametrization, Nielson metric is exploited by Foley[20, 43, 44]. He uses the metric to measure the distance. He suggests another parametrization which not only takes account of distances but also angles between consecutive points. For the case of $m = 2$ (two dimensional points), we define a matrix $Q = \{q_{ij}\}$, $i, j = 1, 2$, for given data points $P_i = (x_i, y_i)$, $0 \leq i \leq n$, by

$$q_{11} = \frac{V_y}{g}, \quad q_{22} = \frac{V_x}{g}, \quad \text{and} \quad q_{12} = q_{21} = -\frac{V_{xy}}{g},$$

where $g = V_x V_y - (V_{xy})^2$,

$$V_x = \frac{\sum_{i=0}^{n}(x_i - \bar{x})^2}{n+1}, \quad V_y = \frac{\sum_{i=0}^{n}(y_i - \bar{y})^2}{n+1},$$
$$V_{xy} = \frac{\sum_{i=0}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n+1},$$

and $\bar{x} = \frac{\sum_{i=0}^{n} x_i}{n+1}$, $\bar{y} = \frac{\sum_{i=0}^{n} y_i}{n+1}$.

The distance between two points U and V is then defined by

$$M[P](U, V) = \sqrt{(U - V)Q(U - V)^T}. \tag{2.2}$$

Table 2.3: Data points in Figure 2.3

x	105	152	205	219	242	290	317	359
y	185	208	260	265	225	248	229	208

The metric defined here is affine invariant[43]. The proof for $m \geq 2$ is given later in Theorem 2.10 in Section 2.5. A definition of affine invariant is also given in Definition 2.8.

The Foley parametrization defines Δt_i's as follows.

$$\Delta t_1 = d_1 \left[1 + \frac{3\hat{\theta}_1 d_2}{2(d_1 + d_2)}\right],$$

$$\Delta t_i = d_i \left[1 + \frac{3\hat{\theta}_{i-1} d_{i-1}}{2(d_{i-1} + d_i)} + \frac{3\hat{\theta}_i d_{i+1}}{2(d_i + d_{i+1})}\right], \quad \text{for} \quad i = 2,\ 3,\ \cdots,\ n-1, \quad \text{and}$$

$$\Delta t_n = d_n,$$

where $d_i = M[P](P_i,\ P_{i-1})$, $\hat{\theta}_i = min(\theta_i,\ \frac{\pi}{2})$, and

$$\theta_i = \pi - arccos\left[\frac{d_{i-1}^2 + d_i^2 - M^2[P](P_{i-1},\ P_{i+1})}{2 d_i d_{i-1}}\right].$$

The angle θ_i is also affine invariant because the distances are measured by Nielson metric. Thus, $\hat{\theta}_i$ is affine invariant if $\theta_i \leq \frac{\pi}{2}$.

For the purpose of comparison, we applied the above parametrizations to the same data points as shown in Figure 2.3(a), (b), (c), and (d). The data points are listed in Table 2.3. The uniform knot vector is used, i.e., $T_i - T_{i-1} = T_{i+1} - T_i$. The result from Foley parametrization is the best among these curves in terms of closeness to the data point polygon.

2.4 The Effect of Knot Vector Selection

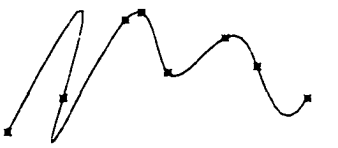
Though Foley parametrization is known to perform well in general[28], our experimentations show that the knot vector selection plays a more important role than the selection of the parameter values. Intuitively, the parametrizations control the speed of the curve

$t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7$

$T_0 \quad T_4 \quad T_5 \quad T_6 \quad T_7 \quad T_8 =$

$= \cdots = T_3 \qquad\qquad \cdots = T_{11}$

(a) equidistant parametrization

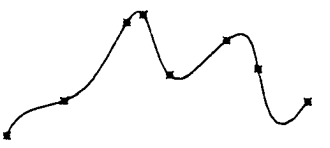


$t_0 \quad t_1 \quad t_2 t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7$

$T_0 \quad T_4 \quad T_5 \quad T_6 \quad T_7 \quad T_8 =$

$= \cdots = T_3 \qquad\qquad \cdots = T_{11}$

(b) chordal parametrization



$t_0 \quad t_1 \quad t_2 \, t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7$

$T_0 \quad T_4 \quad T_5 \quad T_6 \quad T_7 \quad T_8 =$

$= \cdots = T_3 \qquad\qquad \cdots = T_{11}$

(c) centripetal parametrization

$t_0 \, t_1 \quad t_2 t_3 \quad t_4 \quad t_5 \quad t_6 \, t_7$

$T_0 \quad T_4 \quad T_5 \quad T_6 \quad T_7 \quad T_8 =$

$= \cdots = T_3 \qquad\qquad \cdots = T_{11}$

(d) Foley parametrization

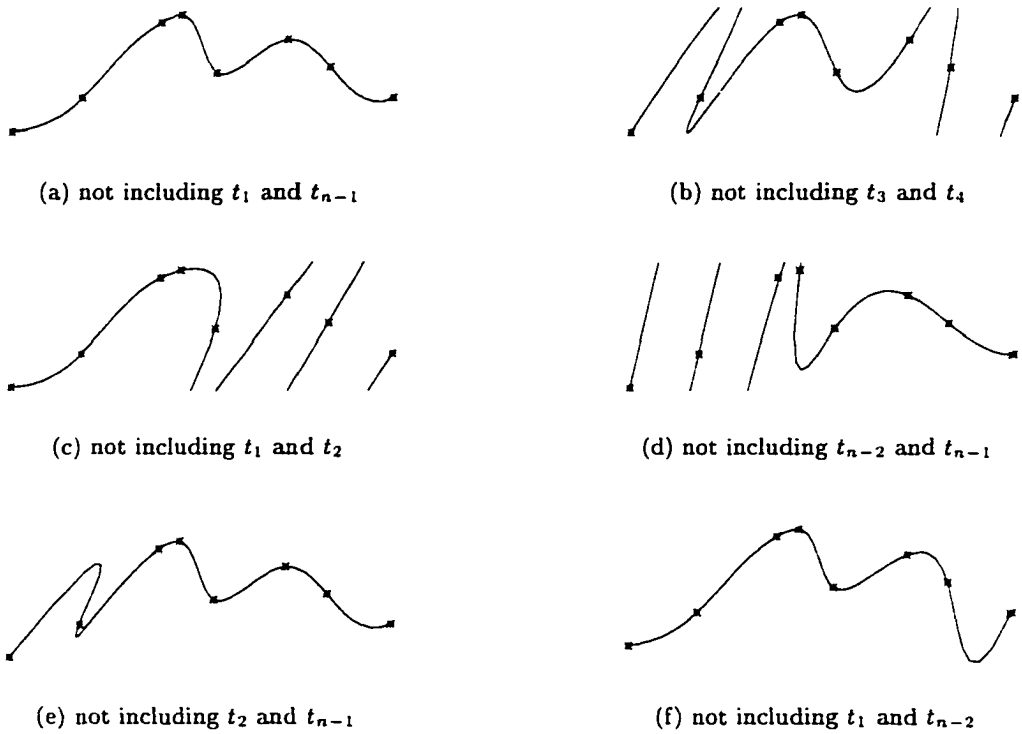
Figure 2.3: Various parametrizations with a uniform knot vector
× : parameter values, + : knot values

according to the distances between data points[20, 33] (however, it seems that nobody has successfully shown any specific relationship among parametrization, speed of curve, and distances between data points).

Let's take a close look at the B-spline curve of order 4 in Figure 2.2. The curve segment between $t = T_i$ and $t = T_{i+1}$ is affected only by the points d_{i-3}, d_{i-2}, d_{i-1}, and d_i and the differences among the values at T_{i-2}, T_{i-1}, $\cdots$, T_{i+3}, and t, $(T_i \leq t \leq T_{i+1})$. More generally,

Theorem 2.7 Suppose $X(t) = \sum_{i=0}^{n} d_i N_{ik}(t)$ is a B-spline curve associated with the knot vector $T = (T_0, T_1, \cdots, T_n, T_{n+1}, \cdots, T_{n+k})$. Then the curve $X(t)$ between T_i and T_{i+1} is completely determined by the deBoor points d_{i-k+1}, $\cdots$, d_i, the knot values T_{i-k+2}, $\cdots$, T_{i+k-1}, and t.

The fact suggests that if there is something we have to control differently on some parts of the interpolating curve $X(t)$, as the speed of curve is controlled by parameter values in chordal and centripetal parametrizations, then a segment between T_i and T_{i+1} should be treated differently from other segments. In other words, the segment between T_i and T_{i+1} should be constructed by the set of various factors which are different in other segments. When we are given a set of data points, on the other hand, the resulting curve $X(t)$ is often expected to behave differently between data points P_i. From this observation, we choose the knot values T_i such that the parameter values are the same as the knot values. (Note that the parametrization has been done by one of the methods discussed previously.) In this way, each segment between T_i and T_{i+1} is governed by a distinct set of factors mentioned above. Here, we have $n + 1$ t_i's already assigned to P_i and $n - 1$ T_i's to be set in case of order 4. Two parameter values are redundant. The selection can be arbitrary. The knot



(a) not including t_1 and t_{n-1}

(b) not including t_3 and t_4

(c) not including t_1 and t_2

(d) not including t_{n-2} and t_{n-1}

(e) not including t_2 and t_{n-1}

(f) not including t_1 and t_{n-2}

Figure 2.4: Several knot selections with chordal parametrization

vector can be selected as follows.

$$\begin{aligned} T_0 = T_1 = T_2 = T_3 &= t_0, \\ T_4 &= t_2, \\ T_5 &= t_3, \\ &\cdots \\ T_n &= t_{n-2}, \text{ and} \\ T_{n+1} = T_{n+2} = T_{n+3} = T_{n+4} &= t_n, \end{aligned}$$

where t_1 and t_{n-1} are not used in setting T_j's. This turns out to be the best selection among the several choices as shown in Figure 2.4. Chordal parametrization is used here. The effects seem to be the same with other parametrizations.

The improved curves are shown together with thier B-splines in Figure 2.5. In this example, every parametrization is improved by using similar knot values to parameter values

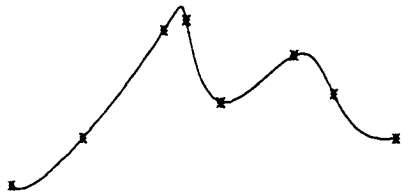
as shown in Figure 2.3(a) and Figure 2.5(a) through Figure 2.3(d) and Figure 2.5(g). More importantly, the improvements of the new knot vector selection due to different parametrizations in Figure 2.5(a), (c), (e) and (g) are not as much as those for cases of the uniform knot vector in Figure 2.3(a), (b), (c) and (d). It appears to indicate that knot vector selection is more important than parametrization.

The new knot selection not only gives nice-looking curves but also has the advantage that it doesn't need to adjust the parameter values t_i's. In the conventional methods, an extra care should be taken for parameter values t_i's to spread out somewhat evenly[28]. In other words, the number of t_i's between T_j and T_{j+1} should be as uniform as possible for each j, $k-1 \leq j \leq n$.

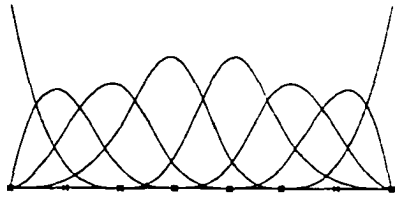
The new knot selection also makes the computation of A in (2.1) faster because there are $k-1$ non-zero terms $N_{ik}(t_j)$ for case of order k in most of its rows instead of k non-zero terms in the conventional knot selections. It happens when the parameter value t_i is equal to one of the knot values T_j. (Note that we know which terms will be non-zero from the knowledge of T_i's.)

2.5 Affine Invariant B-spline Interpolation

A B-spline interpolation curve often has to be affine invariant because it does not look natural otherwise. Intuitively, affine invariancy means that when a list of data points is transformed under an affine transformation, the interpolating curve is transformed in a similar fashion. All the parametrization methods discussed in Section 2.3 give us translation and rotation invariant curves. It's because the relative differences between t_i's remain the same under translation and rotation. However, none of them are scaling invariant except Foley parametrization because of the altered relative differences. Note that scaling is one

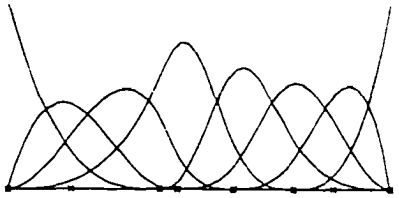


(a) equidistant parametrization

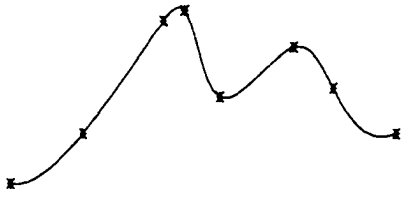


(b) B-splines of
equidistant parametrization

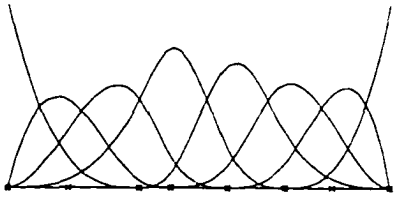
(c) chordal parametrization



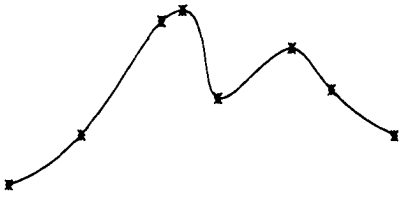
(d) B-splines of
chordal parametrization



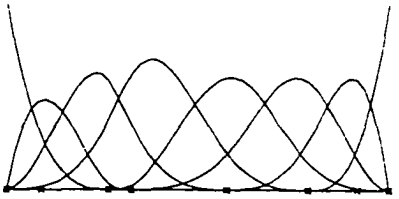
(e) centripetal parametrization



(f) B-splines of
centripetal parametrization



(g) Foley parametrization



(h) B-splines of
Foley parametrization

Figure 2.5: The effect of choosing knot values similar to parameter values
× : parameter values, + : knot values

of the basic affine transformations and affine invariancy is important in the true type or vector font applications[20].

Definition 2.8 Affine (transformation)invariant A metric M is affine invariant if, given any affine (transformation)transformation matrix A, the distances are the same after the affine transformation for two points U and V, i.e., $M(U,V) = M(UA, VA)$.

Definition 2.9 An interpolation curve $X(t)$ is (affine) transformation invariant with respect to a parametrization if (1) $X^*(t)$ is obtained via interpolation with transformed data points using the same parametrization and (2) $X^*(t) = X(t)M$ where M is a (affine) transformation.

Note that these transformation matrices here are linear.

In order to get affine invariant curves, Foley uses the metric introduced by Nielson[43]. The Nielson metric itself remains the same under affine transformations. However, the Nielson parametrization doesn't give us natural-looking curves[20]. To overcome this problem, Foley takes account of not only distances but also angles between consecutive data points.

With the modified parametrization, we can get natural looking curves as shown in Figure 2.3(d) and Figure 2.5(g).

Foley parametrization can be easily generalized for $m(> 2)$ dimension. In the equation (2.2), the matrix Q is the inverse of $\frac{1}{n}\vec{V}^T\vec{V}$, where

$$\vec{V} = \begin{pmatrix} x_0 - \bar{x} & y_0 - \bar{y} \\ x_1 - \bar{x} & y_1 - \bar{y} \\ \cdots & \cdots \\ x_n - \bar{x} & y_n - \bar{y} \end{pmatrix}$$

For m dimensional case, i.e., each point $P_i \in R^m$, the matrix $\vec{V}$ will have m columns.

The new metric given by Nielson is affine invariant. The following theorem is provided for completeness.

Theorem 2.10 The metric $M[P](U,V) = \sqrt{(U-V)Q(U-V)^T}$ is affine invariant.

Proof: Let A be an affine transformation. Then,

$$\begin{aligned}
M[PA](UA,VA) &= \sqrt{(UA-VA)Q(UA-VA)^T} \\
&= \sqrt{(U-V)A[\frac{1}{n}\overrightarrow{VA}^T\overrightarrow{VA}]^{-1}A^T(U-V)^T} \\
&= \sqrt{(U-V)A\frac{1}{n}(A^T\overrightarrow{V}^T\overrightarrow{V}A)^{-1}A^T(U-V)^T} \\
&= \sqrt{(U-V)A\frac{1}{n}A^{-1}[\overrightarrow{V}^T\overrightarrow{V}]^{-1}A^{T^{-1}}A^T(U-V)^T} \\
&= \sqrt{(U-V)[\frac{1}{n}\overrightarrow{V}^T\overrightarrow{V}]^{-1}(U-V)^T} \\
&= \sqrt{(U-V)Q(U-V)^T} \\
&= M[P](U,V)
\end{aligned}$$

Thus, the metric $M[P](U,V)$ is affine invariant. Q.E.D.

It is not so clear what the Nielson metric does intuitively. However, we could get some idea when some unit disks with the metric are examined.

Let Q be a 2×2 matrix which is

$$Q = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix}.$$

Then the equation of the unit disk is

$$q_{11}x^2 + (q_{12}+q_{21})xy + q_{22}y^2 = 1$$

It effectively represents a rotated ellipse with an angle θ from the original ellipse of

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1,$$

(a) Unit disk with Nielson metric

(b) With a list of data points

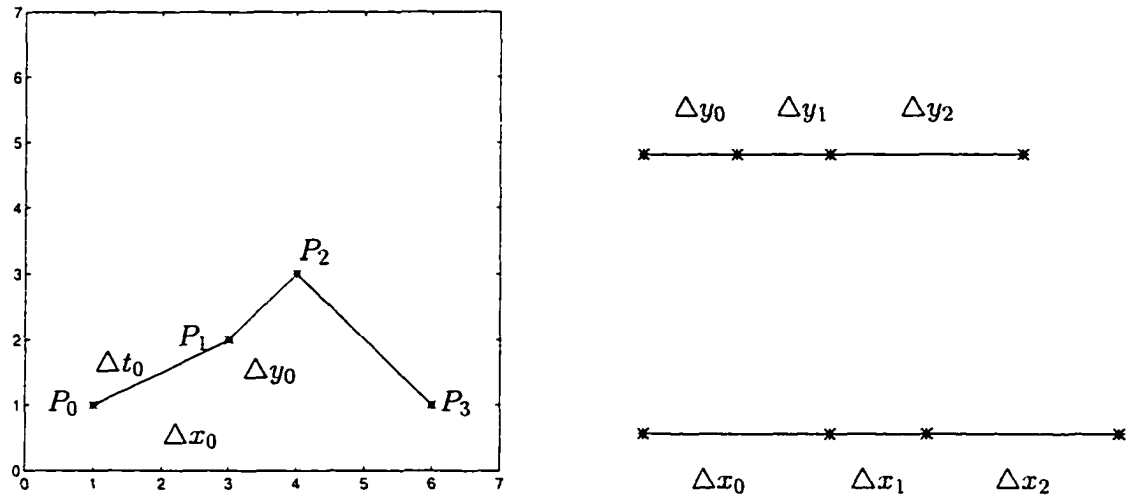
Figure 2.6: Unit disks with Nielson metric

where

$$\begin{aligned} \tan\theta &= \frac{q_{22} - q_{11} \pm \sqrt{(q_{22} - q_{11})^2 + (q_{12} + q_{21})^2}}{q_{12} + q_{21}}, \\ a^2 &= \frac{1}{q_{11}\cos^2\theta + (q_{12} + q_{21})\cos\theta\sin\theta + q_{22}\sin^2\theta}, \text{ and} \\ b^2 &= \frac{1}{q_{11}\sin^2\theta - (q_{12} + q_{21})\cos\theta\sin\theta + q_{22}\cos^2\theta} \end{aligned}$$

Note that there are two distinct θ's because there are two distinct ellipses with a and b exchanged. It is illustrated as in Figure 2.6(a). In Figure 2.6(b), the unit disk is drawn with the metric established from the data points used in Figure 2.3.

A final remark on data-oriented parametrizations needs to be made here. They are based mostly on distances $\|P_i - P_{i-1}\|$ as discussed in Section 2.3. The distances are used for both x and y in 2 dimensional case. However, the computation for x is independent of that of y in (2.1) and the distances are generally different from $\triangle x_i = x_i - x_{i-1}$ or $\triangle y_i = y_i - y_{i-1}$ as shown in Figure 2.7. It makes the data-oriented approach questionable. It seems that the nature of B-spline and its curve have to be exploited for better parametrization.



(a) Data points

(b) Δx_i and Δy_i

Figure 2.7: Curves via equidistant, chordal, and centripetal param etrizations

2.6 Summary

In this chapter, B-splines and B-spline curve are formally defined. Also, a formal definition of B-spline interpolation is given. Several conventional parametrizations are discussed here. Even though Foley parametrization works well in general in terms of closeness to data point polygons, a significant improvement can be accomplished if a knot vector is selected similar to the parameter values regardless of parametrization. In some cases, the knot vector selection is more important than parametrization. With the new knot vector selection, the computation for interpolation becomes efficient. Importantly, the study so far shows that some intrinsic properties of B-splines and B-spline curves have to be fully investigated for better parametrization.

Chapter 3
A New Universal Parametrization

As pointed out in Chapter 1, there are several related problems with conventional parametrizations. They don't work well in some orders k and the resulting curves are not affine invariant with most methods. The curves don't appear so natural to us as human beings. It's probably because they don't take into account the nature of the B-spline basis functions. In this chapter, a new parametrization is presented, based on the nature of B-splines instead of the geometric properties of given data points.

3.1 Affine or Transformation Invariant B-spline Curve

As it have been implied before, to have affine invariant interpolation curves is not so difficult. We can simply use the same parameter and knot values for the transformed curve as for the original curve. In fact, the B-spline curve is invariant under linear transformation if the same knot values are used. It can be stated formally as follows.

Lemma 3.1 Suppose that d_i and d_i^*, $0 \leq i \leq n$, are two lists of deBoor points such that $d_i M = d_i^*$ with a transformation matrix M. Let the B-spline curves $X(t) = \sum_{i=0}^{n} d_i N_{i,k}(t)$ and $X^*(t) = \sum_{i=0}^{n} d_i^* N_{i,k}^*(t)$. If the same knot vector $T = (T_0, T_1, \cdots, T_{n+k})$ is used both for $N_{i,k}(t)$ and $N_{i,k}^*(t)$, then $X(t)M = X^*(t)$ for all $t \in [T_{k-1}, T_{n+1}]$

Proof: Because of the same knot vector T, $N_{i,k}^*(t) = N_{i,k}(t)$. And due to $d_i^* = d_i M$,

$$\begin{aligned} X^*(t) &= \sum_{i=0}^{n} d_i^* N_{i,k}^*(t) = \sum_{i=0}^{n} d_i M N_{i,k}(t) \\ &= \sum_{i=0}^{n} d_i N_{i,k}(t) M = X(t)M \end{aligned}$$

That is, $X(t)M = X^*(t)$. Q.E.D.

Moreover, the B-spline interpolation curve is transformation invariant if the same parameter and knot values are used.

Theorem 3.2 Given a transformation matrix M and the points P_i, let $P_i^* = P_iM$, $i = 0, 1, \cdots, n$. If the same knot values T_i and the same parameter values t_i are used for B-spline interpolation, then $X^*(t) = X(t)M$ for $t \in [T_{k-1}, T_{n+1}]$. In other words, the interpolating B-spline curve is transformation invariant.

Proof: Let $N_{i,k}(t)$ be the B-splines for the knot vector $T = (T_0, T_1, \cdots, T_{n+k})$. Then, we have $X(t) = \sum_{i=0}^{n} d_i N_{i,k}(t)$ and $X^*(t) = \sum_{i=0}^{n} d_i^* N_{i,k}(t)$, $t \in [T_{k-1}, T_{n+1}]$.

Also, $Ad = P$ and $Ad^* = P^*$,

where

$$\begin{aligned} A_{ij} &= N_{ik}(t_j), \\ d &= (d_0, d_1, \cdots, d_n)^T,\ d^* = (d_0^*, d_1^*, \cdots, d_n^*)^T, \text{ and} \\ P &= (P_0, P_1, \cdots, P_n)^T,\ P^* = (P_0^*, P_1^*, \cdots, P_n^*)^T. \end{aligned}$$

Therefore,

$$\begin{aligned} d^* &= A^{-1}P^* \\ &= A^{-1}PM \\ &= dM. \end{aligned}$$

By Lemma 3.1, $X(t)M = X^*(t)$. Q.E.D.

Equidistant parametrization is transformation invariant by Theorem 3.2. The performance of the primitive one is very poor as illustrated before.

3.2 Universal Parametrization

There is an important observation as far as the performance of the interpolating curve is concerned. Our experimentation indicates that the matrix A in (2.1) is diagonally dominant when the interpolation curve looks natural. For example, the corresponding matrices are

Table 3.1: Data points in Figure 3.1(a)

x	106	184	250	289	325	354	394
y	379	150	159	149	156	149	363

shown below for the curves in Figure 2.3(b) and (d). The largest element in each row is in bold.

$$\begin{pmatrix}
\mathbf{1.0000} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0063 & 0.4028 & \mathbf{0.5005} & 0.0904 & 0 & 0 & 0 & 0 \\
0 & 0.0000 & 0.1806 & \mathbf{0.6659} & 0.1535 & 0 & 0 & 0 \\
0 & 0 & 0.0839 & \mathbf{0.6291} & 0.2856 & 0.0014 & 0 & 0 \\
0 & 0 & 0.0001 & 0.2074 & \mathbf{0.6611} & 0.1314 & 0 & 0 \\
0 & 0 & 0 & 0.0025 & 0.3128 & \mathbf{0.5777} & 0.1069 & 0 \\
0 & 0 & 0 & 0 & 0.0653 & 0.4441 & \mathbf{0.4712} & 0.0193 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1.0000}
\end{pmatrix}$$

$$\begin{pmatrix}
\mathbf{1.0000} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.1629 & \mathbf{0.5982} & 0.2233 & 0.0156 & 0 & 0 & 0 & 0 \\
0 & 0.0758 & \mathbf{0.5513} & 0.3670 & 0.0059 & 0 & 0 & 0 \\
0 & 0.0146 & 0.4019 & \mathbf{0.5453} & 0.0382 & 0 & 0 & 0 \\
0 & 0 & 0.0004 & 0.2431 & \mathbf{0.6492} & 0.1072 & 0 & 0 \\
0 & 0 & 0 & 0.0000 & 0.1837 & \mathbf{0.5902} & 0.2261 & 0 \\
0 & 0 & 0 & 0 & 0.0106 & 0.1804 & \mathbf{0.5917} & 0.2174 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1.0000}
\end{pmatrix}$$

It can be noticed easily by taking a close look at Figures 2.5(b), (d), (f), and (h). Now, we can assign appropriate parameter values t_i such that the diagonal element of the matrix is the largest in the corresponding row by choosing the parameter value t_i where N_{ik} is maximum. The uniform knot vector is used here. The selection of parameter and knot values in case of order 4 are illustrated in Figure 3.1(b). The data points are provided in Table 3.1. Detailed computations of order 3 and 4 are given here.

Example 3.3 For order $k = 3$, let $T = (0,\ 0,\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 5,\ 5)$. Then, $N_{i3}(t)$ is maximum at

i	0	1	2	3	4	5	6
t_i	0	2/3	3/2	5/2	7/2	13/3	5

(3.1)

Example 3.4 For order $k = 4$, let $T = (0,\ 0,\ 0,\ 0,\ 1,\ 2,\ 3,\ 4,\ 4,\ 4,\ 4)$. Then, $N_{i4}(t)$ is maximum at

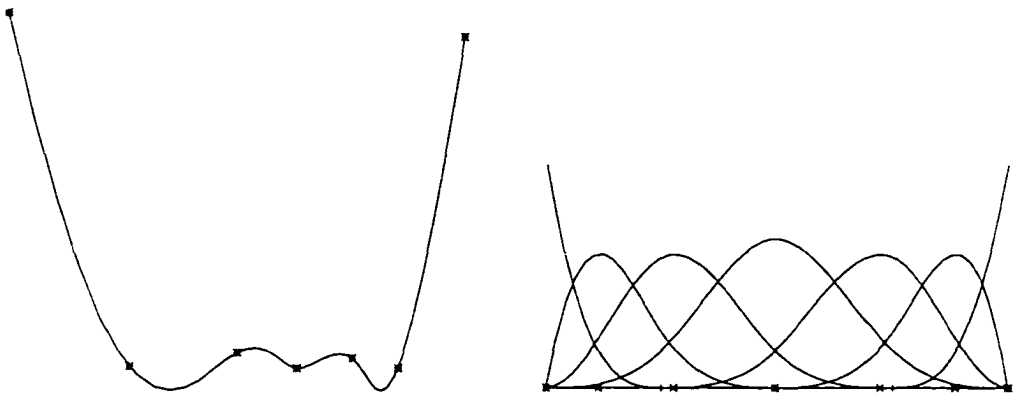
i	0	1	2	3	4	5	6
t_i	0	$\frac{6-2\sqrt{2}}{7}$	$\frac{12-3\sqrt{2}}{7}$	2	$4-\frac{12-3\sqrt{2}}{7}$	$4-\frac{6-2\sqrt{2}}{7}$	4

(3.2)

The following is the matrix A in (2.1) for this case.

$$\begin{pmatrix} \mathbf{1.0000} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1636 & \mathbf{0.5982} & 0.2227 & 0.01550 & 0 & 0 & 0 \\ 0 & 0.1773 & \mathbf{0.5965} & 0.2260 & 0.0002 & 0 & 0 \\ 0 & 0 & 0.1667 & \mathbf{0.6667} & 0.1667 & 0 & 0 \\ 0 & 0 & 0.0002 & 0.2260 & \mathbf{0.5965} & 0.1773 & 0 \\ 0 & 0 & 0 & 0.0155 & 0.2227 & \mathbf{0.5982} & 0.1636 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1.0000} \end{pmatrix} \tag{3.3}$$

Note that this matrix is dependent only of order k and the number of data points.

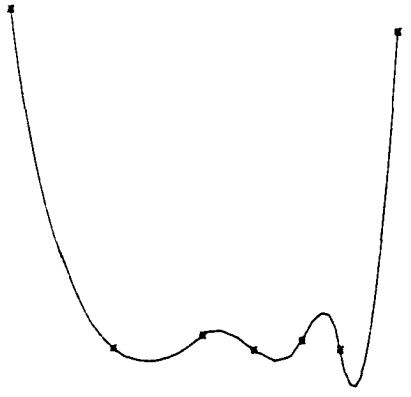


(a) universal parametrization

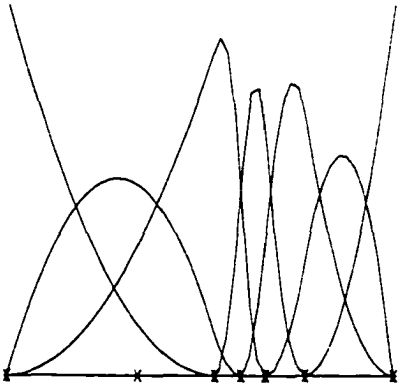
(b) B-splines of universal parametrization

Figure 3.1: The universal selection of parameter and knot values
$\times$: parameter values, $+$: knot values

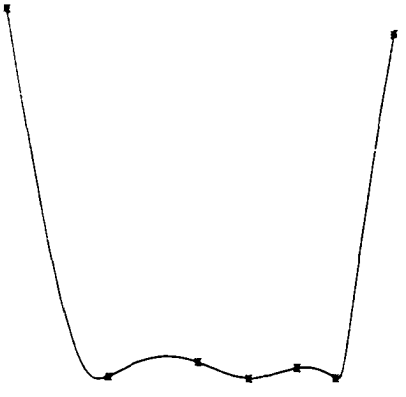
The performance of the new parametrization is not as good as Foley's in terms of the largest bulge between two consecutive data points. However, it seems more natural than those methods in some sense. We may expect a long-stretching leg to continue its way a little more. Sharp corners are not so expected. Moreover, this scheme works well in all orders k. As shown in Figures 3.2 and 3.3, Foley parametrization doesn't work well in cases of orders 3 and 5 while the new method does in any case. This also implies that the new scheme is natural. It appears that the conventional methods do not work well in odd orders. Lee suggests that $t'_i = \frac{t_i + t_{i+1}}{2}$ can be used in odd orders without any justification[33]. Because the selection of parameter and knot values is independent of given data points, the computation of interpolation is simpler and faster than any other methods. In addition, the interpolating curves remain the same under any linear transformation. A scale transformation is applied to several interpolating curves in Figure 3.5. The data points are scaled along the y-axis with a factor of n. The curves via Foley and universal transformations are scaled only along the y-axis while the curves via chordal and centripetal parametrizations are scaled both along the x-axis and the y-axis.



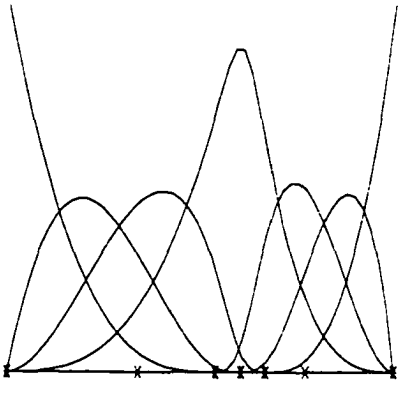
(a) order 3



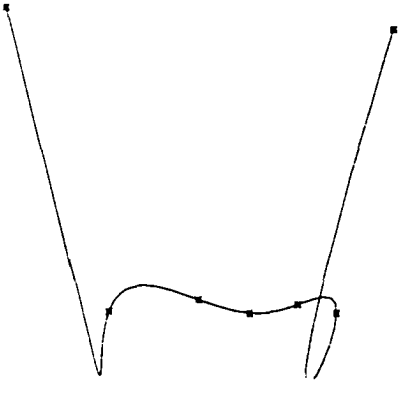
(b) B-splines of order 3



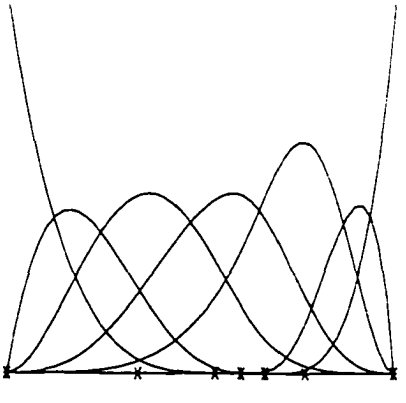
(c) order 4



(d) B-splines of order 4

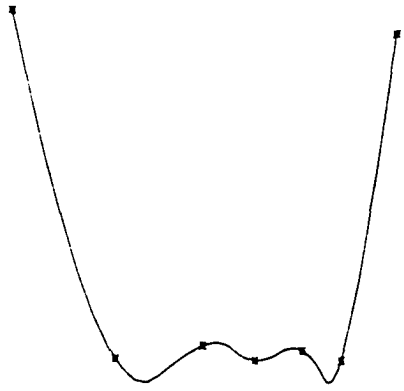


(e) order 5

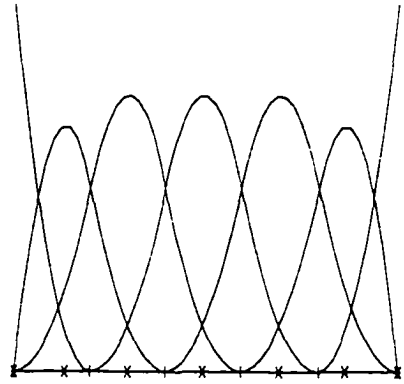


(f) B-splines of order 5

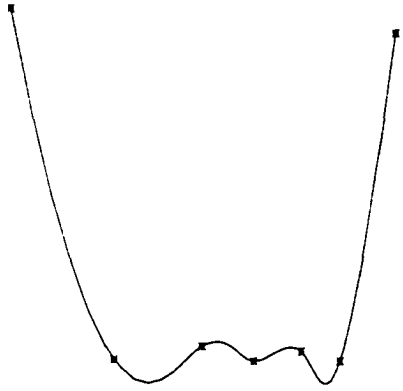
Figure 3.2: Results of Foley parametrization with different orders
× : parameter values, + : knot values



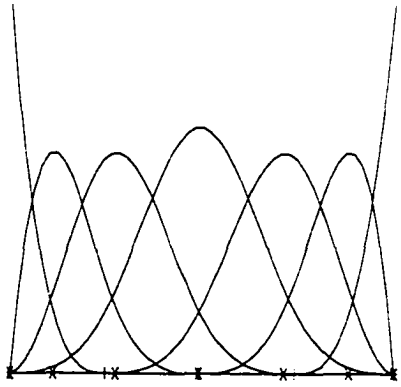
(a) order 3



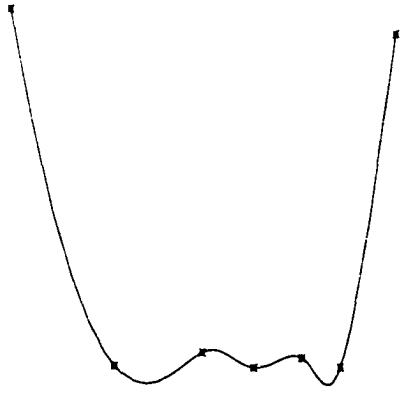
(b) B-splines of order 3



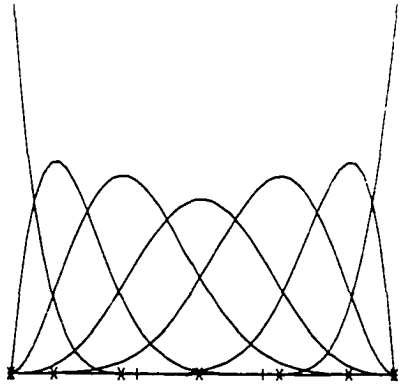
(c) order 4



(d) B-splines of order 4

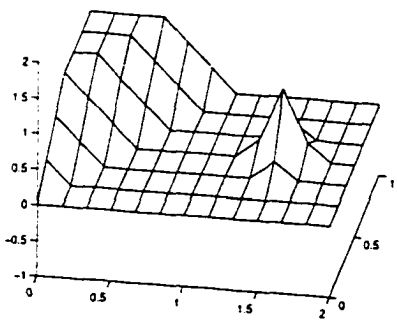


(e) order 5

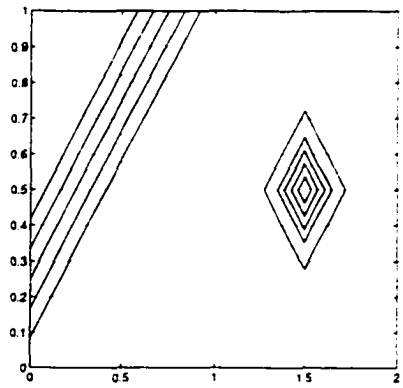


(f) B-splines of order 5

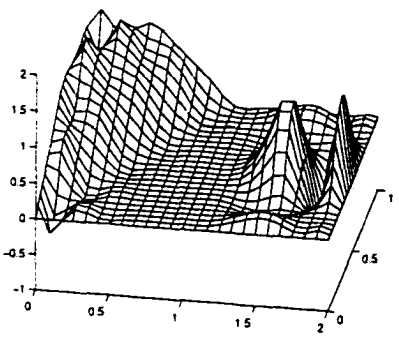
Figure 3.3: Results of universal parametrization with different orders
× : parameter values, + : knot values



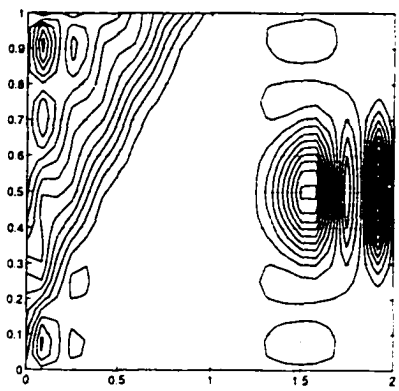
(a) Data points on surface



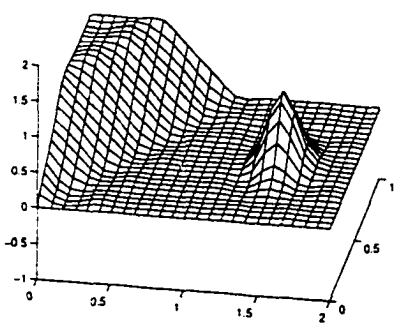
(b) Contour of its mesh



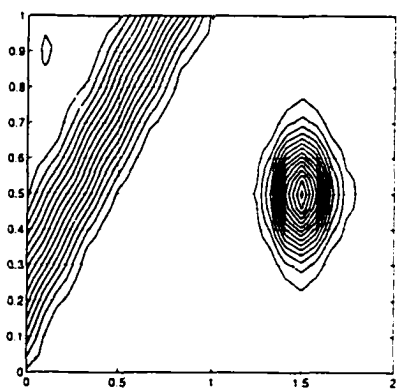
(c) Equidistant parametrization



(d) Its contour



(e) Universal parametrization



(f) Its contour

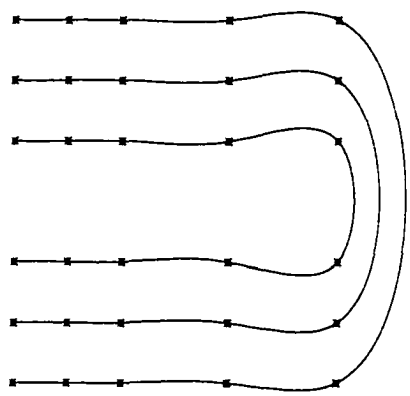
Figure 3.4: Surface interpolation with two different parametrizations

We apply universal parametrization to a set of data points on a rectangular grid for surface interpolation using the well-known tensor product form $X(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} d_{ij} N_{ik}(u)N_{jl}(v)$[28]. A more natural looking surface is obtained as shown in Figure 3.4. The data points used here are from the following surface.

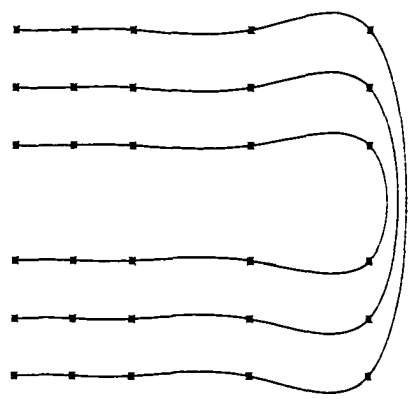
$$z = \begin{cases} 1, & \text{if } y - x \geq \frac{1}{2}, \\ 2(y-x), & \text{if } 0 \leq y - x \leq \frac{1}{2}, \\ \frac{1}{2}\left\{\cos\left(4\pi[(x-\frac{3}{2})^2 + (y-\frac{1}{2})^2]^{\frac{1}{2}}\right) + 1\right\}, & \text{if } \left[(x-\frac{3}{2})^2 + (y-\frac{1}{2})^2\right] \leq \frac{1}{16}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

As illustrated in Figure 3.4(d) and (f), the surface from universal parametrization has fewer fluctuations and more pleasing contour curves. Notice that the tensor product can be applied only with equidistant or universal parametrization.

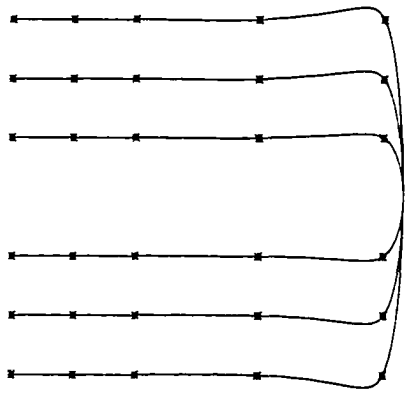
There is another interesting property of the new parametrization. As we move a single data point P_i to P_i', the effect on the shape of the interpolating curve is mostly on its neighborhood as shown in Figure 3.6(c) and (d). It is because most diagonally dominant matrices have the inverse matrices which are also diagonally dominant. On the other hand, the effect is not locally limited at all with chordal or centripetal parametrization. Note that the effect is not totally limited to a local neighborhood with universal parametrization. Thus, it may be called the semi-localness property with respect to data points. This local effect is quite critical in interactive modeling. For example, the inverse matrix for A in (3.3)



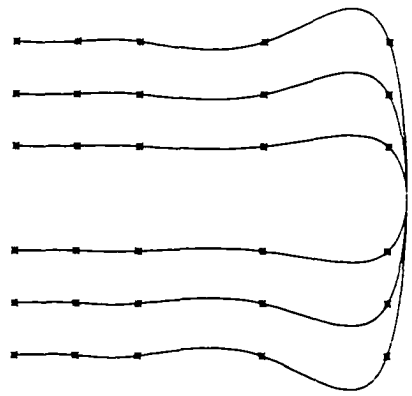
(a) Chordal parametrization



(b) Centripetal parametrization

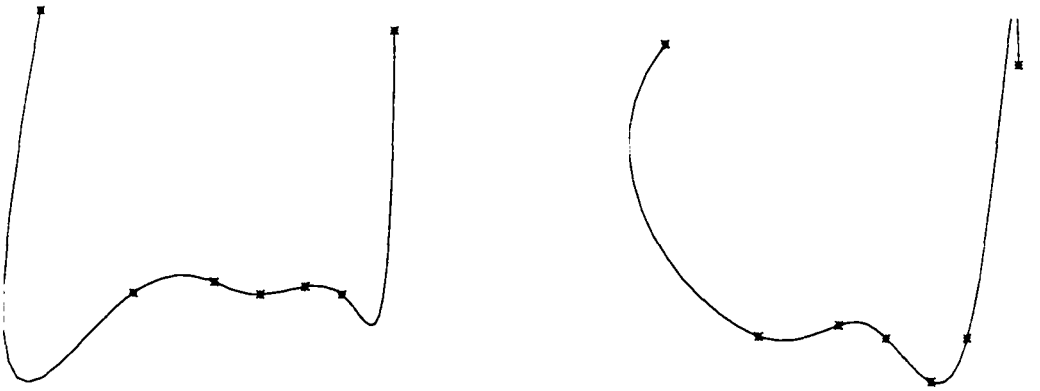


(c) Foley parametrization



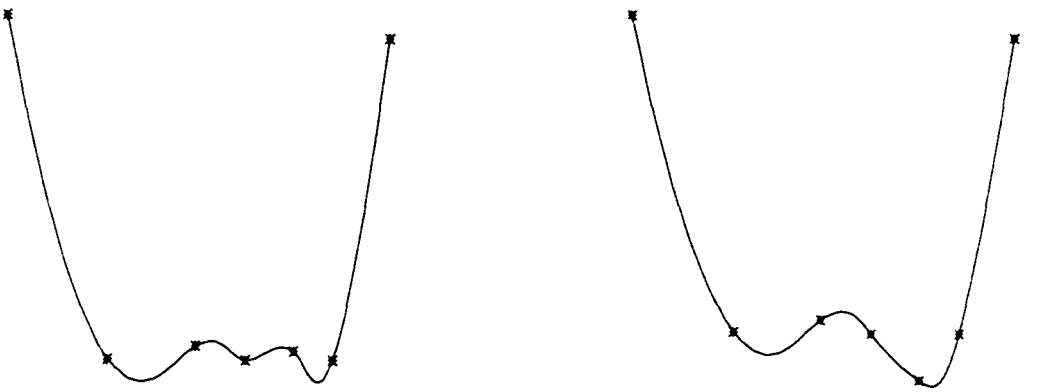
(d) Universal parametrization

Figure 3.5: Resulting curves under an affine transformation
Data points are scaled up with a factor of n along the y-axis.



(a) Curve obtained centripetal parametrization

(b) After changing P_4 to P_4'



(c) Curve via universal: same as in Figure 3.1(a)

(d) After changing P_4 to P_4'

Figure 3.6: The semi-localness with respect to data points

is given here.

$$\begin{pmatrix} \mathbf{1.0000} & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.3112 & \mathbf{1.9024} & -0.7786 & 0.2452 & -0.0767 & 0.0227 & -0.0037 \\ 0.1034 & -0.6323 & \mathbf{2.1334} & -0.7907 & 0.2476 & -0.0734 & 0.0120 \\ -0.0289 & 0.1764 & -0.5952 & \mathbf{1.8953} & -0.5952 & 0.1764 & -0.0289 \\ 0.0120 & -0.0734 & 0.2476 & -0.7907 & \mathbf{2.1334} & -0.6323 & 0.1034 \\ -0.0037 & 0.0227 & -0.0767 & 0.2452 & -0.7786 & \mathbf{1.9024} & -0.3112 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1.0000} \end{pmatrix}$$

The fact that the inverse matrix is diagonally dominant can be proved in some special cases as follows.

Theorem 3.5 If a square matrix A has the property that $|a_{ii}| \gg |a_{jk}|$ for $i,\ j,\ k = 1,\ 2,\ \cdots,\ n$ and $j \neq k$, then the inverse matrix $A^{-1} = B$ is diagonally dominant in the sense that $|b_{ii}| \geq |b_{ij}|$ and $|b_{ji}|,\ i \neq j$.

Proof: It is enough to show that $|A_{ii}| \geq |A_{ij}|$ and $|A_{ji}|$ for $i,\ j = 1,\ 2,\ \cdots,\ n$ and $i \neq j$. Let A^{ij} be the matrix of size $(n-1) \times (n-1)$ with ith row and jth column removed from A.

Suppose that $j > i + 1$. Then,

$$\begin{aligned} A_{ii} &= (-1)^{j} a_{1,j} A^{ii}_{1,j-1} + (-1)^{j+1} a_{2,j} A^{ii}_{2,j-1} + \cdots \\ &\quad + (-1)^{i+j-2} a_{i-1,j} A^{ii}_{i-1,j} + (-1)^{i+j-1} a_{i+1,j} A^{ii}_{i+1,j} + \cdots \\ &\quad + (-1)^{2j-2} a_{j,j} A^{ii}_{j,j} + \cdots \\ &\quad + (-1)^{n+j-2} a_{n,j} A^{i,i}_{n-1,j} \end{aligned}$$

and

$$A_{ij} = (-1)^{i-1} a_{1,i} A^{ij}_{1,i} + (-1)^{i} a_{2,i} A^{ij}_{2,j} + \cdots$$

$$+ (-1)^1 a_{i-1,i} A^{ij}_{i-1,i} + (-1)^2 a_{i+1,i} A^{ij}_{i+1,i} + \cdots$$
$$+ (-1)^{i-n+2} a_{n,i} A^{ij}_{n-1,i}.$$

Here,

$$\begin{aligned} A^{ii}_{1,j-1} &= A^{ij}_{1,i} \\ A^{ii}_{2,j-1} &= A^{ij}_{2,i} \\ &\cdots \\ A^{ii}_{n-1,j-1} &= A^{ij}_{n-1,i}. \end{aligned}$$

And only A_{ii} has a diagonal element, i.e., a_{jj}. Thus $|A_{ii}| \geq |A_{ij}|$ due to the assumption that $|a_{ii}| \gg |a_{jk}|$. Similarly, $|A_{ii}| \geq |A_{ji}|$.

Therefore, the inverse matrix A^{-1} is also diagonally dominant. Q.E.D.

3.3 Summary

In this chapter, the new universal parametrization is presented, based on the nature of B-splines. Universal parametrization has the following advantages. (1) It is able to make very natural looking curves and surfaces. (2) It works well in all orders k. (3) The resulting curves are transformation invariant. And, (4) it has the semi-localness property with respect to data points, which is very critical in interactive modeling.

Chapter 4
Fairness Evaluation

The quality of curve is subject to one's personal perspective. In the literature, the quality of modeling curve or surface is often expressed in terms of *fairness*[18, 52]. *Curve energy* is used to represent fairness quantitatively. One major component of the curve energy is *bending energy* which is much related to the total curvature of curve. *Stretching energy* is the other major component, which is proportional to the total length of curve. In short, both total curvature and total length are being used simultaneously to represent fairness quantitatively. The shape of curvature plot is also taken into account to evaluate curve fairness.

In Section 4.1, some properties of conventional curve measures are discussed. Total curvature, total length and curvature plot will be discussed. The parametrization methods are compared quantitatively with some example curves in Section 4.2. Curvature plots are also drawn to measure the fairness of curves. The evaluation is summarized in Section 4.3.

4.1 Measures for Curve Quality

The quality of curve is often expressed in terms of its energy. One of the major components of curve energy is the bending energy[8, 58]. The curvature is a convenient intuitive measure of the bending energy because of its geometric nature. Some authors use the second parametric derivative as an approximation to curvature when $\|X'(t)\| = 1$[17, 41, 58]. However, it is not a good approximation in general as shown in Figure 4.1.

The *total arc-length curvature* $\int \kappa(s)ds$ is often used. Interestingly, it is invariant under linear reparametrization as follows.

First, arc-length curvatures are invariant under linear reparametrization. Let t and u be two different parametrizations for the same 2D curve where $(x_1(t), y_1(t))$ and $(x_2(u), y_2(u))$

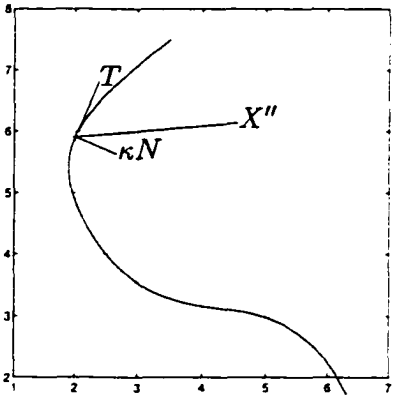


Figure 4.1: Curvature and the second derivative
T : unit tangent vector, N : unit normal vector,
κ : curvature, and X'' : parametric second derivative

the curves for t and u, respectively with $u = at + b$. Then, $x_2(u) = x_2(at + b) = x_1(t)$ and $y_2(u) = y_2(at + b) = y_1(t)$.

Also,

$$\begin{aligned} x_2'(u) &= x_1'(t)\frac{dt}{du} = x_1'(t)\frac{1}{a}, \\ y_2'(u) &= y_1'(t)\frac{dt}{du} = y_1'(t)\frac{1}{a}, \\ x_2''(u) &= \frac{d(x_2'(u))}{dt}\frac{dt}{du} = x_1''(t)\frac{1}{a^2} \text{ and} \\ y_2''(u) &= \frac{d(y_2'(u))}{dt}\frac{dt}{du} = y_1''(t)\frac{1}{a^2}. \end{aligned}$$

Therefore,

$$\begin{aligned} \kappa_2(u) &= \frac{x_2'(u)y_2''(u) - y_2'(u)x_2''(u)}{(\sqrt{x_2'(u)^2 + y_2'(u)^2})^3} \\ &= \frac{\frac{1}{a^3}(x_1'(t)y_1''(t) - y_1'(t)x_1''(t))}{\frac{1}{a^3}(\sqrt{x_1'(t)^2 + y_1'(t)^2})^3} \\ &= \kappa_1(t). \end{aligned}$$

Lemma 4.1 The arc-length curvature is invariant under linear reparametrization. That is, $\kappa_1(t) = \kappa_2(u)$ with $u = at + b$, where $\kappa_1(t)$ and $\kappa_2(u)$ are the arc-length curvature functions with respect to parameters t and u, respectively.

Theorem 4.2 The total curvature with respect to arc length is invariant under linear reparametrization.

Proof: Let s be the reparametrization of u with respect to arc length. And let $u_0 = at_0 + b$ and $u_1 = at_1 + b$. Then, the total curvature with respect to arc length is

$$\begin{aligned}
\int \kappa(s)^2 ds &= \int_{u_0}^{u_1} \kappa(s(u))^2 \frac{ds}{du} du \\
&= \int_{u_0}^{u_1} \kappa_2(u)^2 \sqrt{x_2'(u)^2 + y_2'(u)^2} du \\
&= \int_{t_0}^{t_1} \kappa_1(t)^2 \sqrt{x_1'(t)^2 + y_1'(t)^2}\ \frac{1}{a}\ a\ dt \\
&= \int_{t_0}^{t_1} \kappa_1(t)^2 \sqrt{x_1'(t)^2 + y_1'(t)^2} dt.
\end{aligned}$$

Therefore, the total arc-length curvature is invariant under linear reparametrization.

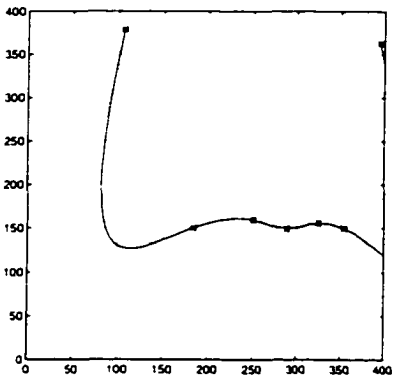
However, the total arc-length curvature is not enough to represent the curve energy. As shown in Figure 4.2(a),(b),(e) and (f), it is possible for a modeling curve to have a very pleasing curvature curve while it goes around with a large length. Therefore, the length of curve must be taken into account. Similar statements can be found in Lee[34].

Some authors use a weighted sum of stretching energy and bending energy as the curve energy[8, 58]. The stretching energy is proportional to the total length of curve.

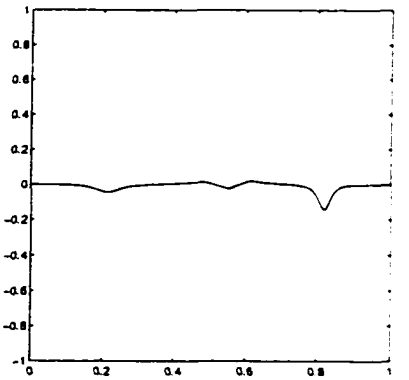
$$\text{curve energy} = \int \alpha X'(t)^2 + \beta X''(t)^2 dt,$$

where α and β are freely selected coefficients. The choice of α and β is dependent of applications at hand. Without a good justification, one combination of two energy components is not so convincing. There has not been much justification. It seems that niether of these terms can be solely used to measure the quality of curve.

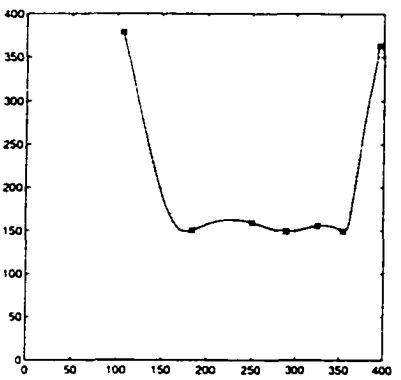
In general cases, each of these two needs to be considered by itself. To compensate the lack of decisiveness of these two quantitative measures, the curvature plots will be drawn for consideration.



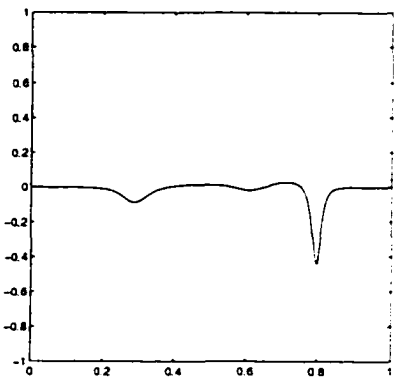
(a) Chordal



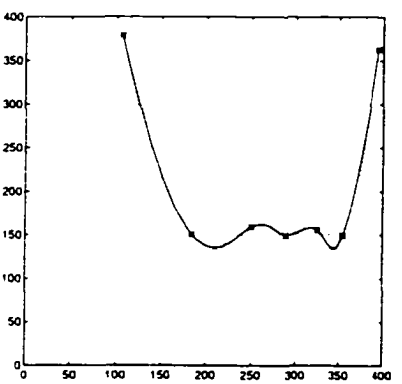
(b) Its curvature plot



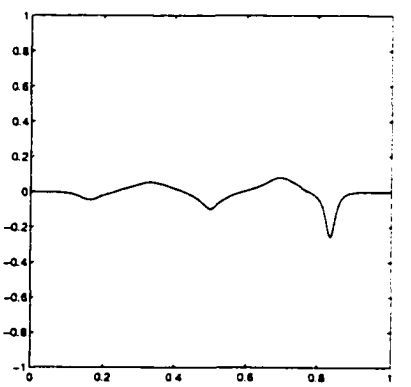
(c) Foley



(d) Its curvature plot



(e) Universal



(f) Its curvature plot

Figure 4.2: Interpolation curves via three different parametrizations with the first set of data points

Table 4.1: Data points in Figure 4.3

x	105	152	205	242	290	317	359
y	185	208	260	225	248	229	208

Table 4.2: Data points in Figure 4.4

x	1.0	1.8	2.9	4.1	5.4	6.4	7.5	8.0	8.0	8.0	7.5	6.5	5.5	5.5	6.5	7.5
y	2.0	3.3	4.1	4.1	3.2	2.3	1.3	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5

4.2 Comparison

Chordal, Foley and universal parametrizations are compared in this section. First, the curvatures $\int \kappa(s)ds$ are computed for three sets of data points as in Table 4.3. The resulting curves with three different sets of data points are drawn in Figure 4.2, 4.3, and 4.4. The data points used in Figure 4.3 and 4.4 are given in Table 4.1 and 4.2, respectively. The total lengths are also computed in Table 4.4.

With the first set of data points, it is hard to tell which performs best because chordal parametrization generates a curve with the smallest total curvature 0.2601 while Foley parametrization gives us the shortest curve with total length 649.1800 with much larger total curvatures of 0.5709.

Table 4.3: Total curvatures with respect to arc length

	chordal	Foley	universal
curve 1	0.2601	0.5709	0.5832
curve 2	0.3388	1.0667	0.3256
curve 3	7.3083	8.6298	9.5213

Table 4.4: Total lengths

	chordal	Foley	universal
curve 1	869.4386	649.1800	668.3566
curve 2	327.4424	316.3698	312.3161
curve 3	19.0727	19.0731	19.1133

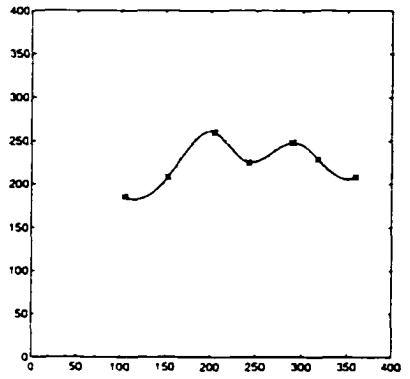
On the other hand, universal parametrization performs best with the second set of data points. Its interpolating curve has the smallest total curvature 0.3256 with the shortest total length 312.3161.

With the third set of data points, chordal parametrization performs best with the smallest total curvature 7.3083 and the shortest curve length 19.0727. However, the curvature plot of the curve has too many wiggles and is not so pleasing as shown in Figure 4.4(b).

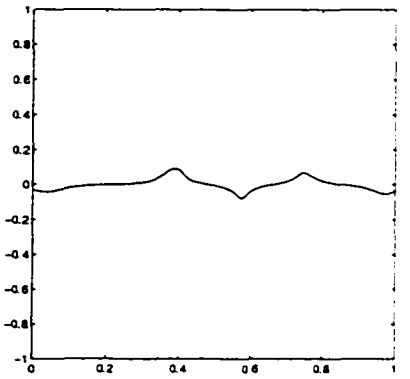
Now, it is obvious that niether total curvature nor total length can be solely used to measure the quality of curve. Throughout this and further experimentations. universal parametrization generates fair curves with pleasing curvature plots. Importantly, natural looking curvature plots could be obtained without much increased total curvature or total length.

4.3 Summary

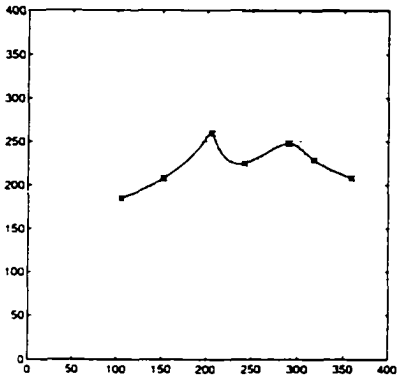
The comparison shows that chordal parametrization gives us quite fair curves in terms of total curvature. However, it often makes longer curves than other parametrizations do. In some cases, it has too many wiggles in its curvature plot and the curvature plot is not so pleasing. On the other hand, universal parametrization generates fair curves with very pleasing curvature plots. The pleasing curvature plots could be obtained without much increased total curvature or total length.



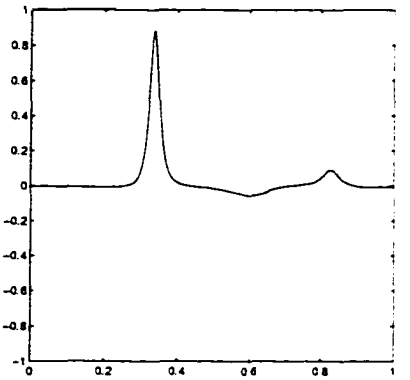
(a) Chordal



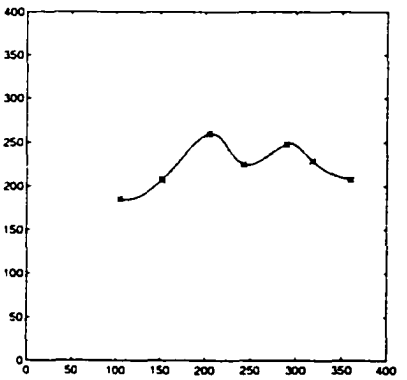
(b) Its curvature plot



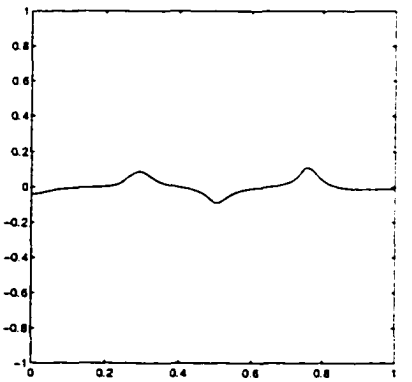
(c) Foley



(d) Its curvature plot

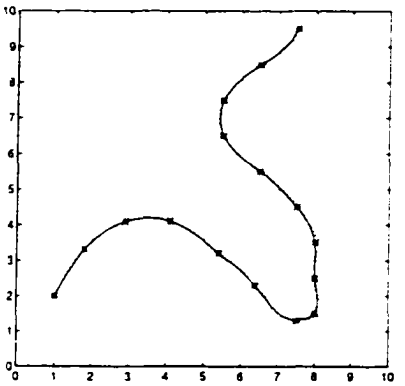


(e) Universal

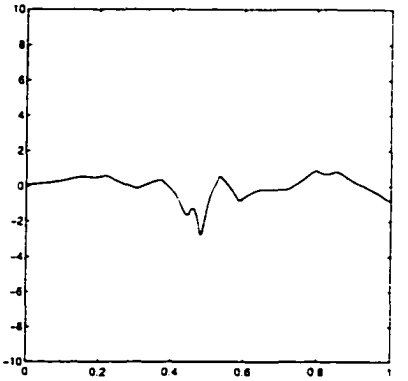


(f) Its curvature plot

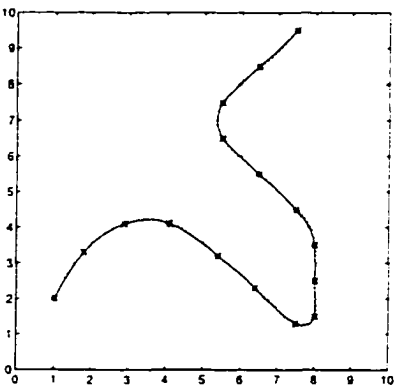
Figure 4.3: Interpolation curves via three different parametrizations with the second set of data points



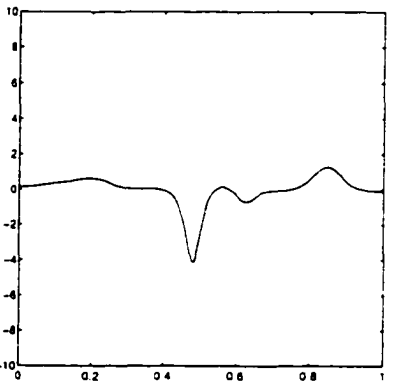
(a) Chordal



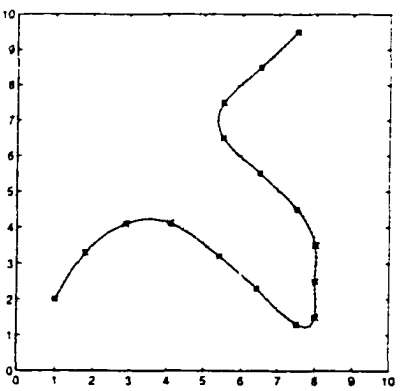
(b) Its curvature plot



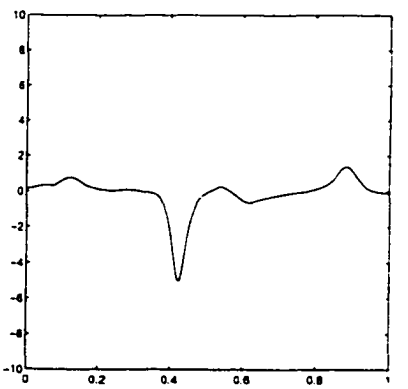
(c) Foley



(d) Its curvature plot



(e) Universal



(f) Its curvature plot

Figure 4.4: Interpolation curves via three different parametrizations with the third set of data points

In the chapter, three different parametrizations are compared in order 4 (degree 3). It's not easy to decide which performs best in general. However, it becomes obvious where these parametrizations are compared in order 5 (degree 4). The universal parametrization performs best. It's clear when looking at the example curves in Chapter 3.

Chapter 5
Knot Removal

In 1980, knot removal was introduced by Boehm as the inversion of knot insertion[6]. Knot removal has been successfully used to reduce a number of knot values and hence the same number of deBoor points[16, 39, 40, 53]. There are two major reasons why knot removal is important in curve and surface modeling.

1. More compact representations for curves and surfaces can be obtained. It means that less memory space is needed to represent curves and surfaces.
2. It is often believed that if we are given more data points, the interpolation curve is closer to the original curve, regardless of parametrization. Knot removal gives us a freedom to choose as many data points as needed. Because it removes unnecessary knot values and deBoor points after the interpolating curves have been obtained. Note that it is not easy, in general, to select appropriate data points in terms of the number of data points and the quality of the resulting curve.

In this chapter, an effective and efficient method for knot removal is presented. The knot removal method is applied to the curves obtained through several different parametrizations. The resulting curves are compared in terms of differences from the original curves. Very close curves to the original curves can be obtained when the original curves are generated through universal parametrization. It seems to once again indicate that universal parametrization is a reasonable choice in parametric B-spline curve and surface.

5.1 Knot Insertion

We are given a knot vector $T = (T_0, T_1, \cdots, T_{n+k})$ with a non-negative integer n and a positive integer k. Knot insertion is the process to insert a new knot T^*, $T_l < T^* \leq T_{l+1}$, into the knot vector T. The new knot vector $\hat{T}$ becomes $\hat{T} = (\hat{T}_0, \hat{T}_1, \cdots, \hat{T}_l, \hat{T}_{l+1}, \hat{T}_{l+2}, \cdots, \hat{T}_{n+k+1})$

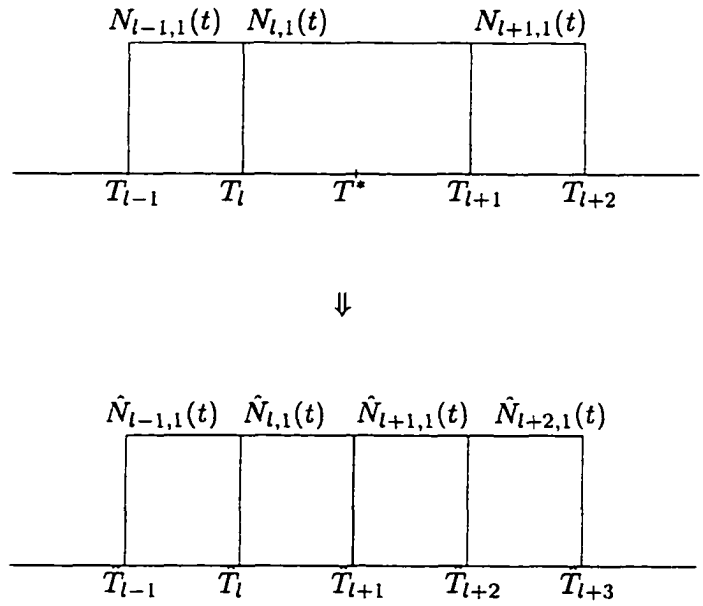


Figure 5.1: Before and after inserting a new knot value

where $\hat{T}_i = T_i$ for $0 \leq i \leq l$, $\hat{T}_{l+1} = T^*$, and $\hat{T}_i = T_{i-1}$ for $l+2 \leq i \leq n+k+1$. Based these two knot vectors, the corresponding normalized B-splines $N_{i,k}(t)$, $0 \leq i \leq n$, and $\hat{N}_{i,k}(t)$, $0 \leq i \leq n+1$, are defined as illustrated in Chapter 2.

The insertion process is trivial in case of order $k = 1$ as shown in Figure 5.1. The relationship between the two B-splines $N_{i,k}(t)$ and $\hat{N}_{i,k}(t)$ is

$$N_{i,1}(t) = \begin{cases} \hat{N}_{i,1}(t), & 0 \leq i \leq l-1, \\ \hat{N}_{i,1}(t) + \hat{N}_{i+1,1}(t), & i = l, \\ \hat{N}_{i+1,1}(t), & l+1 \leq i \leq n. \end{cases} \tag{5.1}$$

The relationships in higher orders can be established using the recurrence in B-splines. For example, in case of order 2,

$$N_{i,2}(t) = \frac{t - T_i}{T_{i+1} - T_i} N_{i,1}(t) + \frac{T_{i+2} - t}{T_{i+2} - T_{i+1}} N_{i+1,1}(t)$$

and

$$\hat{N}_{i,2}(t) = \frac{t - \hat{T}_i}{\hat{T}_{i+1} - \hat{T}_i} \hat{N}_{i,1}(t) + \frac{\hat{T}_{i+2} - t}{\hat{T}_{i+2} - \hat{T}_{i+1}} \hat{N}_{i+1,1}(t).$$

Then the following relationships are also trivial.

- $0 \leq i \leq l-2$,

$$\begin{aligned} N_{i,2}(t) &= \frac{t-T_i}{T_{i+1}-T_i}N_{i,1}(t) + \frac{T_{i+2}-t}{T_{i+2}-T_{i+1}}N_{i+1,1}(t) \\ &= \frac{t-\hat{T}_i}{\hat{T}_{i+1}-\hat{T}_i}\hat{N}_{i,1}(t) + \frac{\hat{T}_{i+2}-t}{\hat{T}_{i+2}-\hat{T}_{i+1}}\hat{N}_{i+1,1}(t) \\ &= \hat{N}_{i,2}(t). \end{aligned}$$

- $i = l-1$,

$$\begin{aligned} N_{l-1,2}(t) &= \frac{t-T_{l-1}}{T_l-T_{l-1}}N_{l-1,1}(t) + \frac{T_{l+1}-t}{T_{l+1}-T_l}N_{l,1}(t) \\ &= \frac{t-\hat{T}_{l-1}}{\hat{T}_l-\hat{T}_{l-1}}\hat{N}_{l-1,1}(t) + \frac{\hat{T}_{l+2}-t}{\hat{T}_{l+2}-\hat{T}_l}(\hat{N}_{l,1}(t)+\hat{N}_{l+1,1}(t)) \\ &= \frac{t-\hat{T}_{l-1}}{\hat{T}_l-\hat{T}_{l-1}}\hat{N}_{l-1,1}(t) + \frac{\hat{T}_{l+1}-t}{\hat{T}_{l+1}-\hat{T}_l}\hat{N}_{l,1}(t) + \\ &\quad \frac{\hat{T}_{l+2}-\hat{T}_{l+1}}{\hat{T}_{l+2}-\hat{T}_l}\left(\frac{t-\hat{T}_l}{\hat{T}_{l+1}-\hat{T}_l}\hat{N}_{l,1}(t) + \frac{\hat{T}_{l+2}-t}{\hat{T}_{l+2}-\hat{T}_{l+1}}\hat{N}_{l+1,1}(t)\right) \\ &= \hat{N}_{l-1,2}(t) + \frac{\hat{T}_{l+2}-\hat{T}_{l+1}}{\hat{T}_{l+2}-\hat{T}_l}\hat{N}_{l,2}(t). \end{aligned}$$

- $i = l$,

$$\begin{aligned} N_{l,2}(t) &= \frac{t-T_l}{T_{l+1}-T_l}N_{l,1}(t) + \frac{T_{l+2}-t}{T_{l+2}-T_{l+1}}N_{l+1,1}(t) \\ &= \frac{t-\hat{T}_l}{\hat{T}_{l+2}-\hat{T}_l}(\hat{N}_{l,1}(t)+\hat{N}_{l+1,1}(t)) + \frac{\hat{T}_{l+3}-t}{\hat{T}_{l+3}-\hat{T}_{l+2}}\hat{N}_{l+2,1}(t) \\ &= \frac{\hat{T}_{l+1}-\hat{T}_l}{\hat{T}_{l+2}-\hat{T}_l}\left(\frac{t-\hat{T}_l}{\hat{T}_{l+1}-\hat{T}_l}\hat{N}_{l,1}(t) + \frac{\hat{T}_{l+2}-t}{\hat{T}_{l+2}-\hat{T}_{l+1}}\hat{N}_{l+1,1}(t)\right) + \\ &\quad \frac{t-\hat{T}_{l+1}}{\hat{T}_{l+2}-\hat{T}_{l+1}}\hat{N}_{l+1,1}(t) + \frac{\hat{T}_{l+3}-t}{\hat{T}_{l+3}-\hat{T}_{l+2}}\hat{N}_{l+2,1}(t) \\ &= \frac{\hat{T}_{l+1}-\hat{T}_l}{\hat{T}_{l+2}-\hat{T}_l}\hat{N}_{l,2}(t) + \hat{N}_{l+1,2}(t). \end{aligned}$$

And

- $l+1 \leq i \leq n$,

$$N_{i,2}(t) = \frac{t-T_i}{T_{i+1}-T_i}N_{i,1}(t) + \frac{T_{i+2}-t}{T_{i+2}-T_{i+1}}N_{i+1,1}(t)$$

$$= \frac{t-\hat{T}_{i+1}}{\hat{T}_{i+2}-\hat{T}_{i+1}}\hat{N}_{i+1,1}(t) + \frac{\hat{T}_{i+3}-t}{\hat{T}_{i+3}-\hat{T}_{i+2}}\hat{N}_{i+2,1}(t)$$
$$= \hat{N}_{i+1,2}(t).$$

In short,

$$N_{i,2}(t) = \begin{cases} \hat{N}_{i,2}(t), & 0 \le i \le l-2, \\ \hat{N}_{i,2}(t) + \frac{\hat{T}_{i+3}-\hat{T}_{l+1}}{\hat{T}_{i+3}-\hat{T}_{i+1}}\hat{N}_{i+1,2}(t), & i = l-1, \\ \frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+2}-\hat{T}_i}\hat{N}_{i,2}(t) + \hat{N}_{i+1,2}(t), & i = l, \\ \hat{N}_{i+1,2}(t), & l+1 \le i \le n. \end{cases} \tag{5.2}$$

In other words,

$$N_{i,2}(t) = \begin{cases} \hat{N}_{i,2}(t), & 0 \le i \le l-2, \\ \frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+2}-\hat{T}_i}\hat{N}_{i,2}(t) + \frac{\hat{T}_{i+3}-\hat{T}_{l+1}}{\hat{T}_{i+3}-\hat{T}_{i+1}}\hat{N}_{i+1,2}(t), & l-1 \le i \le l, \\ \hat{N}_{i+1,2}(t), & l+1 \le i \le n. \end{cases} \tag{5.3}$$

The general relationship in order k can be represented as follows.

Theorem 5.1 Given a knot vector $T = (T_0, T_1, \cdots, T_{n+k})$ with a non-negative integer n and a positive integer k, the associated B-splines $N_{i,k}(t)$ are defined for $i = 0, 1, \cdots, n$.

Suppose that a new knot value $T^* \in (T_l, T_{l+1}]$, $k-1 \le l \le n$, is inserted, i.e., $T' = (T_0, T_1, \cdots, T_l, T^*, T_{l+1}, \cdots, T_{n+k})$ and the new knot vector is re-indexed as follows.

$$\hat{T} = (\hat{T}_0, \hat{T}_1, \cdots, \hat{T}_l, \hat{T}_{l+1}, \hat{T}_{l+2}, \cdots, \hat{T}_{n+k}, \hat{T}_{n+k+1}),$$

where

$$\hat{T}_i = \begin{cases} T_i, & 0 \le i \le l, \\ T^*, & i = l+1, \\ T_{i-1}, & l+2 \le i \le n. \end{cases}$$

Then, the B-splines $\hat{N}_{i,k}(t)$ based on the new knot vector $\hat{T}$ has the following relationship with the original B-splines $N_{i,k}(t)$.

If $k = 1$,

$$N_{i,1}(t) = \begin{cases} \hat{N}_{i,1}(t), & 0 \le i \le l-1, \\ \hat{N}_{i,1}(t) + \hat{N}_{i+1,1}(t), & i = l, \\ \hat{N}_{i+1,1}(t), & l+1 \le i \le n. \end{cases}$$

If $k \ge 2$,

$$N_{i,k}(t) = \begin{cases} \hat{N}_{i,k}(t), & 0 \le i \le l-k, \\ \hat{N}_{i,k}(t) + \frac{\hat{T}_{i+k+1}-\hat{T}_{l+1}}{\hat{T}_{i+k+1}-\hat{T}_{i+1}}\hat{N}_{i+1,k}(t), & i = l-k+1, \\ \frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+k}-\hat{T}_i}\hat{N}_{i,k}(t) + \frac{\hat{T}_{i-k+1}-\hat{T}_{l+1}}{\hat{T}_{i-k+1}-\hat{T}_{i+1}}\hat{N}_{i+1,k}(t), & l-k+2 \le i \le l-1, \\ \frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+k}-\hat{T}_i}\hat{N}_{i,k}(t) + \hat{N}_{i+1,k}(t), & i = l, \\ \hat{N}_{i+1,k}(t), & l+1 \le i \le n. \end{cases} \tag{5.4}$$

In other words,

$$N_{i,k}(t) = \begin{cases} \hat{N}_{i,k}(t), & 0 \le i \le l-k, \\ \frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+k}-\hat{T}_i}\hat{N}_{i,k}(t) + \frac{\hat{T}_{i+k+1}-\hat{T}_{l+1}}{\hat{T}_{i+k+1}-\hat{T}_{i+1}}\hat{N}_{i+1,k}(t), & l-k+1 \le i \le l, \\ \hat{N}_{i+1,k}(t), & l+1 \le i \le n. \end{cases} \tag{5.5}$$

Proof : The relationships in orders 1 and 2 are already shown as in (5.1) and (5.2). Let's assume that they holds in order $k = m$, i.e.,

$$N_{i,m}(t) = \begin{cases} \hat{N}_{i,m}(t), & 0 \le i \le l-m, \\ \hat{N}_{i,m}(t) + \frac{\hat{T}_{i+m+1}-\hat{T}_{l+1}}{\hat{T}_{i+m+1}-\hat{T}_{i+1}}\hat{N}_{i+1,m}(t), & i = l-m+1, \\ \frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+m}-\hat{T}_i}\hat{N}_{i,m}(t) + \frac{\hat{T}_{i+m+1}-\hat{T}_{l+1}}{\hat{T}_{i+m+1}-\hat{T}_{i+1}}\hat{N}_{i+1,m}(t), & l-m+2 \le i \le l-1, \\ \frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+m}-\hat{T}_i}\hat{N}_{i,m}(t) + \hat{N}_{i+1,m}(t), & i = l, \\ \hat{N}_{i+1,m}(t), & l+1 \le i \le n. \end{cases}$$

In case of order $k = m+1$,

1. $0 \le i \le l-(m+1)$,

$$N_{i,m+1}(t) = \frac{t-T_i}{T_{i+m}-T_i}N_{i,m}(t) + \frac{T_{i+m+1}-t}{T_{i+m+1}-T_{i+1}}N_{i+1,m}(t)$$

$$= \frac{t-\hat{T}_i}{\hat{T}_{i+m}-\hat{T}_i}\hat{N}_{i,m}(t) + \frac{\hat{T}_{i+m+1}-t}{\hat{T}_{i+m+1}-\hat{T}_{i+1}}\hat{N}_{i+1,m}(t)$$
$$= \hat{N}_{i,m+1}(t).$$

2. $i = l-(m+1)+1$,

$$\begin{aligned}
N_{l-m,m+1}(t) &= \frac{t-T_{l-m}}{T_l-T_{l-m}}N_{l-m,m}(t) + \frac{T_{l+1}-t}{T_{l+1}-T_{l-m+1}}N_{l-m+1,m}(t) \\
&= \frac{t-\hat{T}_{l-m}}{\hat{T}_l-\hat{T}_{l-m}}\hat{N}_{l-m,m}(t) + \\
&\quad \frac{\hat{T}_{l+2}-t}{\hat{T}_{l+2}-\hat{T}_{l-m+1}}\left(\hat{N}_{l-m+1,m}(t) + \frac{\hat{T}_{l+2}-\hat{T}_{l+1}}{\hat{T}_{l+2}-\hat{T}_{l-m+2}}\hat{N}_{l-m+2,m}(t)\right) \\
&= \frac{t-\hat{T}_{l-m}}{\hat{T}_l-\hat{T}_{l-m}}\hat{N}_{l-m,m}(t) + \frac{\hat{T}_{l+1}-t}{\hat{T}_{l+1}-\hat{T}_{l-m+1}}\hat{N}_{l-m+1,m}(t) + \\
&\quad \frac{\hat{T}_{l+2}-\hat{T}_{l+1}}{\hat{T}_{l+2}-\hat{T}_{l-m+1}}\left(\frac{t-\hat{T}_{l-m+1}}{\hat{T}_{l+1}-\hat{T}_{l-m+1}}\hat{N}_{l-m+1,m}(t)+\right. \\
&\quad \left.\frac{\hat{T}_{l+2}-t}{\hat{T}_{l+2}-\hat{T}_{l-m+2}}\hat{N}_{l-m+2,m}(t)\right) \\
&= \hat{N}_{l-m,m+1}(t) + \frac{\hat{T}_{l+2}-\hat{T}_{l+1}}{\hat{T}_{l+2}-\hat{T}_{l-m+1}}\hat{N}_{l-m+1,m+1}(t) \\
&= \hat{N}_{l-(m+1)+1,m+1}(t) + \frac{\hat{T}_{l+2}-\hat{T}_{l+1}}{\hat{T}_{l+2}-\hat{T}_{l-(m+1)+2}}\hat{N}_{l-(m+1)+2,m+1}(t).
\end{aligned}$$

3. $l-(m+1)+2 \le i \le l-1$,

- $i = l-(m+1)+2$,

$$\begin{aligned}
N_{l-(m+1)+2,m+1}(t) &= \frac{t-T_{l-m+1}}{T_{l+1}-T_{l-m+1}}N_{l-m+1,m}(t) + \frac{T_{l+2}-t}{T_{l+2}-T_{l-m+2}}N_{l-m+2,m}(t) \\
&= \frac{t-\hat{T}_{l-m+1}}{\hat{T}_{l+2}-\hat{T}_{l-m+1}}N_{l-m+1,m}(t) + \frac{\hat{T}_{l+3}-t}{\hat{T}_{l+3}-\hat{T}_{l-m+2}}N_{l-m+2,m}(t) \\
&= \frac{t-\hat{T}_{l-m+1}}{\hat{T}_{l+2}-\hat{T}_{l-m+1}}\left(\hat{N}_{l-m+1,m}(t) + \frac{\hat{T}_{l+2}-\hat{T}_{l+1}}{\hat{T}_{l+2}-\hat{T}_{l-m+2}}\hat{N}_{l-m+2,m}(t)\right) \\
&\quad + \frac{\hat{T}_{l+3}-t}{\hat{T}_{l+3}-\hat{T}_{l-m+2}}\left(\frac{\hat{T}_{l+1}-\hat{T}_{l-m+2}}{\hat{T}_{l+2}-\hat{T}_{l-m+2}}\hat{N}_{l-m+2,m}(t)+\right. \\
&\quad \left.\frac{\hat{T}_{l+3}-\hat{T}_{l+1}}{\hat{T}_{l+3}-\hat{T}_{l+m+3}}\hat{N}_{l-m+3,m}(t)\right) \\
&= \frac{\hat{T}_{l+1}-\hat{T}_{l-m+1}}{\hat{T}_{l+2}-\hat{T}_{l-m+1}}\left(\frac{t-\hat{T}_{l-m+1}}{\hat{T}_{l+1}-\hat{T}_{l-m+1}}\hat{N}_{l-m+1,m}(t)+\right.
\end{aligned}$$

$$\frac{\hat{T}_{l+2}-t}{\hat{T}_{l+2}-\hat{T}_{l-m+2}}\hat{N}_{l-m+2,m}(t)\Big) +$$
$$\frac{\hat{T}_{l+3}-\hat{T}_{l+1}}{\hat{T}_{l+3}-\hat{T}_{l-m+2}}\left(\frac{t-\hat{T}_{l-m+2}}{\hat{T}_{l+2}-\hat{T}_{l-m+2}}\hat{N}_{l-m+2,m}(t)+\right.$$
$$\left.\frac{\hat{T}_{l+3}-t}{\hat{T}_{l+3}-\hat{T}_{l-m+3}}\hat{N}_{l-m+3,m}(t)\right)$$
$$= \frac{\hat{T}_{l+1}-\hat{T}_{l-m+1}}{\hat{T}_{l+2}-\hat{T}_{l-(m+1)+2}}\hat{N}_{l-(m+1)+2,m+1}(t) +$$
$$\frac{\hat{T}_{l+3}-\hat{T}_{l+1}}{\hat{T}_{l+3}-\hat{T}_{l-(m+1)+3}}\hat{N}_{l-(m+1)+3,m+1}(t).$$

- $l-(m+1)+3 \le i \le l-2$,

$$\begin{aligned}
N_{i,m+1}(t) &= \frac{t-T_i}{T_{i+m}-T_i}N_{i,m}(t)+\frac{T_{i+m+1}-t}{T_{i+m+1}-T_{i+1}}N_{i+1,m}(t)\\
&= \frac{t-\hat{T}_i}{\hat{T}_{i+m+1}-\hat{T}_i}\left(\frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+m}-\hat{T}_i}\hat{N}_{i,m}(t)+\frac{\hat{T}_{i+m+1}-\hat{T}_{l+1}}{\hat{T}_{i+m+1}-\hat{T}_{i+1}}\hat{N}_{i+1,m}(t)\right)+\\
&\quad \frac{\hat{T}_{i+m+2}-t}{\hat{T}_{i+m+2}-\hat{T}_{i+1}}\left(\frac{\hat{T}_{l+1}-\hat{T}_{i+1}}{\hat{T}_{i+m+1}-\hat{T}_{i+1}}\hat{N}_{i+1,m}(t)+\frac{\hat{T}_{i+m+2}-\hat{T}_{l+1}}{\hat{T}_{i+m+2}-\hat{T}_{i+2}}\hat{N}_{i+2,m}(t)\right)\\
&= \frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+m+1}-\hat{T}_i}\left(\frac{t-\hat{T}_i}{\hat{T}_{i+m}-\hat{T}_i}\hat{N}_{i,m}(t)+\frac{\hat{T}_{i+m+1}-t}{\hat{T}_{i+m+1}-\hat{T}_{i+1}}\hat{N}_{i+1,m}(t)\right)+\\
&\quad \frac{\hat{T}_{i+m+2}-\hat{T}_{l+1}}{\hat{T}_{i+m+2}-\hat{T}_{i+1}}\left(\frac{t-\hat{T}_{i+1}}{\hat{T}_{i+m+1}-\hat{T}_{i+1}}\hat{N}_{i+1,m}(t)+\frac{\hat{T}_{i+m+2}-t}{\hat{T}_{i+m+2}-\hat{T}_{i+2}}\hat{N}_{i+2,m}(t)\right)\\
&= \frac{\hat{T}_{l+1}-\hat{T}_i}{\hat{T}_{i+(m+1)}-\hat{T}_i}\hat{N}_{i,m+1}(t)+\frac{\hat{T}_{i+(m+1)+1}-\hat{T}_{l+1}}{\hat{T}_{i+(m+1)+1}-\hat{T}_{i+1}}\hat{N}_{i+1,m+1}(t).
\end{aligned}$$

- $i = l-1$,

$$\begin{aligned}
N_{l-1,m+1}(t) &= \frac{t-T_{l-1}}{T_{l+m-1}-T_{l-1}}N_{l-1,m}(t)+\frac{T_{l+m}-t}{T_{l+m}-T_l}N_{l,m}(t)\\
&= \frac{t-\hat{T}_{l-1}}{\hat{T}_{l+m}-\hat{T}_{l-1}}\left(\frac{\hat{T}_{l+1}-\hat{T}_{l-1}}{\hat{T}_{l+m-1}-\hat{T}_{l-1}}\hat{N}_{l-1,m}(t)+\frac{\hat{T}_{l+m}-\hat{T}_{l+1}}{\hat{T}_{l+m}-\hat{T}_l}\hat{N}_{l,m}(t)\right)\\
&\quad +\frac{\hat{T}_{l+m+1}-t}{\hat{T}_{l+m+1}-\hat{T}_l}\left(\frac{\hat{T}_{l+1}-\hat{T}_l}{\hat{T}_{l+m}-\hat{T}_l}\hat{N}_{l,m}(t)+\hat{N}_{l+1,m}(t)\right)\\
&= \frac{\hat{T}_{l+1}-\hat{T}_{l-1}}{\hat{T}_{l+m}-\hat{T}_{l-1}}\left(\frac{t-\hat{T}_{l-1}}{\hat{T}_{l+m-1}-\hat{T}_{l-1}}\hat{N}_{l-1,m}(t)+\frac{\hat{T}_{l+m}-t}{\hat{T}_{l+m}-\hat{T}_l}\hat{N}_{l,m}(t)\right)+\\
&\quad \frac{\hat{T}_{l+m+1}-\hat{T}_{l+1}}{\hat{T}_{l+m+1}-\hat{T}_l}\left(\frac{t-\hat{T}_l}{\hat{T}_{l+m}-\hat{T}_l}\hat{N}_{l,m}(t)+\frac{\hat{T}_{l+m+1}-t}{\hat{T}_{l+m+1}-\hat{T}_{l+1}}\hat{N}_{l+1,m}(t)\right)\\
&= \frac{\hat{T}_{l+1}-\hat{T}_{l-1}}{\hat{T}_{l+(m+1)-1}-\hat{T}_{l-1}}\hat{N}_{l-1,m+1}(t)+\frac{\hat{T}_{l+(m+1)}-\hat{T}_{l+1}}{\hat{T}_{l+(m+1)}-\hat{T}_l}\hat{N}_{l,m+1}(t).
\end{aligned}$$

4. $i = l$,

$$\begin{aligned}
N_{l,m+1}(t) &= \frac{t-T_l}{T_{l+m}-T_l}N_{l,m}(t) + \frac{T_{l+m+1}-t}{T_{l+m+1}-T_{l+1}}N_{l+1,m}(t) \\
&= \frac{t-\hat{T}_l}{\hat{T}_{l+m+1}-\hat{T}_l}\left(\frac{\hat{T}_{l+1}-\hat{T}_l}{\hat{T}_{l+m}-\hat{T}_l}\hat{N}_{l,m}(t) + \hat{N}_{l+1,m}(t)\right) + \frac{\hat{T}_{l+m+2}-t}{\hat{T}_{l+m+2}-\hat{T}_{l+2}}\hat{N}_{l+2,m}(t) \\
&= \frac{\hat{T}_{l+1}-\hat{T}_l}{\hat{T}_{l+m+1}-\hat{T}_l}\left(\frac{t-\hat{T}_l}{\hat{T}_{l+m}-\hat{T}_l}\hat{N}_{l,m}(t) + \frac{\hat{T}_{l+m+1}-t}{\hat{T}_{l+m+1}-\hat{T}_{l+1}}\hat{N}_{l+1,m}(t)\right) + \\
&\quad \frac{t-\hat{T}_{l+1}}{\hat{T}_{l+m+1}-\hat{T}_{l+1}}\hat{N}_{l+1,m}(t) + \frac{\hat{T}_{l+m+2}-t}{\hat{T}_{l+m+2}-\hat{T}_{l+2}}\hat{N}_{l+2,m}(t) \\
&= \frac{\hat{T}_{l+1}-\hat{T}_l}{\hat{T}_{l+(m+1)}-\hat{T}_l}\hat{N}_{l,m+1}(t) + \hat{N}_{l+1,m+1}(t).
\end{aligned}$$

5. $i \geq l+1$,

$$\begin{aligned}
N_{i,m+1}(t) &= \frac{t-T_i}{T_{i+m}-T_i}N_{i,m}(t) + \frac{T_{i+m+1}-t}{T_{i+m+1}-T_{i+1}}N_{i+1,m}(t) \\
&= \frac{t-\hat{T}_{i+1}}{\hat{T}_{i+m+1}-\hat{T}_{i+1}}\hat{N}_{i+1,m}(t) + \frac{\hat{T}_{i+m+2}-t}{\hat{T}_{i+m+2}-\hat{T}_{i+2}}\hat{N}_{i+2,m}(t) \\
&= \hat{N}_{i+1,m+1}(t).
\end{aligned}$$

Therefore, (5.4) holds by 1 through 5 the mathematical induction. Q.E.D.

The above theorem has been proved by Boehm using k-th divided difference[6]. However, the proof here is based on the recurrence property of B-spline. It is provided for completeness because the B-splines $N_{i,k}(t)$ are defined by the recurrence throughout this dissertation.

With the relationship between B-splines $N_{i,k}(t)$ and $\hat{N}_{i,k}(t)$, the relationship between the corresponding deBoor points d_i and $\hat{d}_i$ can be easily obtained as follows[6].

$$\begin{aligned}
\sum_{i=l-k+1}^{l} d_i N_{i,k}(t) &= \sum_{i=l-k+1}^{l} d_i \left(\frac{\hat{T}_{i+1}-\hat{T}_i}{\hat{T}_{i+k}-\hat{T}_i}\hat{N}_{i,k}(t) + \frac{\hat{T}_{i+k+1}-\hat{T}_{i+1}}{\hat{T}_{i-k+1}-\hat{T}_{i+1}}\hat{N}_{i+1,k}(t)\right) \\
&= \sum_{i=l-k+1}^{l+1} \left(\frac{\hat{T}_{i+1}-\hat{T}_i}{\hat{T}_{i+k}-\hat{T}_i}d_i + \frac{\hat{T}_{i+k}-\hat{T}_{i+1}}{\hat{T}_{i+k}-\hat{T}_i}d_{i-1}\right)\hat{N}_{i,k}(t).
\end{aligned} \tag{5.6}$$

Thus, $\hat{d}_i = \frac{\hat{T}_{i+1}-\hat{T}_i}{\hat{T}_{i+k}-\hat{T}_i}d_i + \frac{\hat{T}_{i+k}-\hat{T}_{i+1}}{\hat{T}_{i+k}-\hat{T}_i}d_{i-1}$, $i = l-k+1, \cdots, l+1$. Geometrically, $\hat{d}_i$ divides the line segment $\overline{d_{i-1}d_i}$ with ratio of $\frac{\hat{T}_{i+1}-\hat{T}_i}{\hat{T}_{i+1}-\hat{T}_i} : 1-\frac{\hat{T}_{i+1}-\hat{T}_i}{\hat{T}_{i+1}-\hat{T}_i}$ as pointed out by Boehm. We are interested in inserting a single knot because inserting mutiple knots can be done by doing a single knot insertion repeatedly.

5.2 Knot Removal

Knot removal is the inverse process of knot insertion. If a knot value is inserted artificially, then knot removal is nothing but removing that artificailly inserted knot. This means that the knot value and its corresponding deBoor point can be removed without changing the shape of the curve. Such a knot value is called *removable.*

From (5.6), $\hat{T}_{l+1}$ is removable (and hence $\hat{d}_{l+1}$) if there are the deBoor points d_{l-k+1}, d_{l-k+2}, $\cdots$, d_l so that

$$\begin{aligned} \hat{d}_{l-k+1} &= d_{l-k+1}, \\ \hat{d}_i &= \alpha_i d_i + (1-\alpha_i) d_{i-1},\ i = l-k+2,\ \cdots,\ l \\ &\quad \text{where } \alpha_i = \frac{\hat{T}_{l+1} - \hat{T}_i}{\hat{T}_{i+k} - \hat{T}_i}, \\ \hat{d}_{l+1} &= d_l. \end{aligned} \tag{5.7}$$

The resulting curve is $X(t) = \sum_{i=0}^{n} d_i N_{i,k}(t)$ and the shape of curve is the same as that of the original curve.

However, there are not many removable knots in real applications. If some error can be tolerated to some extent, knot removal could be used to get approximate curves and surfaces with a compact representation. Generally, knot removal is used for data reduction and/or approximation. It is also used to combine piecewise polynomial curves into a single B-spline curve[53]. The various uses of knot removal are discussed by Lyche and Morken[39]. When some error is acceptable, the problem becomes to find the closest points d_{l-k+1}, d_{l-k+2}, $\cdots$, d_l so that

$$\begin{aligned} \hat{d}_{l-k+1} &\cong d_{l-k+1}, \\ \hat{d}_i &\cong \alpha_i d_i + (1-\alpha_i) d_{i-1},\ i = l-k+2,\ \cdots,\ l \\ &\quad \text{where } \alpha_i = \frac{\hat{T}_{l+1} - \hat{T}_i}{\hat{T}_{i+k} - \hat{T}_i}, \end{aligned}$$

$$\hat{d}_{l+1} \cong d_l.$$

There are some methods to find d_i's, available in the literature. One simple method was used by Tiller[53]. First, d_{l-k+1} and d_l are set to be $\hat{d}_{l-k+1}$ and $\hat{d}_{l+1}$, respectively. Then, d_{l-k+2} and d_{l-1} and so on. For each step i, $i = 1, \cdots, \lceil \frac{k}{2} \rceil$, d_{l-k+i} and d_{l-i+1} are computed. After the final iteration, two computed $d_{l-k+\lceil \frac{k}{2} \rceil}$ are compared if k is odd. Otherwise, $d_{l-k+\lceil \frac{k}{2} \rceil}$ and $d_{l-\lceil \frac{k}{2} \rceil+1}$ are inserted into the following equation.

$$\hat{d}_{l-\lceil \frac{k}{2} \rceil+1} = \alpha_{l-\lceil \frac{k}{2} \rceil+1} d_{l-\lceil \frac{k}{2} \rceil+1} + (1 - \alpha_{l-\lceil \frac{k}{2} \rceil+1}) d_{l-k+\lceil \frac{k}{2} \rceil}.$$

If they are close enough, then $\hat{T}_{l+1}$ and $\hat{d}_{l+1}$ could be removed. The deBoor points d_i's along the process are used to compute the new B-spline curve $X(t) = \sum_{i=0}^{n} d_i N_{i,k}(t)$.

Example 5.2 In case of order $k = 4$, the new deBoor points are computed as follows.

- Iteration 1

$$d_{l-3} = \hat{d}_{l-3} \text{ and } d_l = \hat{d}_{l+1}.$$

- Iteration 2 $= \lceil \frac{k}{2} \rceil$

$$d_{l-2} = \frac{1}{\alpha_{l-2}}(\hat{d}_{l-2} - (1 - \alpha_{l-2}) d_{l-3}) \text{ and } d_{l-1} = \frac{1}{1 - \alpha_l}(\hat{d}_l - \alpha_l d_l).$$

d_{l-2} and d_{l-1} is substituted into the following.

$$\hat{d}_{l-1} = \alpha_{l-1} d_{l-1} + (1 - \alpha_{l-1}) d_{l-2}.$$

If they are close enough, the deBoor points d_i's are used as the new deBoor points.

Example 5.3 In case of order $k = 5$, the detailed computation for the new deBoor points is as follows.

- Iteration 1

$$d_{l-4} = \hat{d}_{l-4} \text{ and } d_l = \hat{d}_{l+1}.$$

- Iteration 2

$$d_{l-3} = \frac{1}{\alpha_{l-3}}(\hat{d}_{l-3} - (1-\alpha_{l-3})d_{l-4}) \text{ and } d_{l-1} = \frac{1}{1-\alpha_l}(\hat{d}_l - \alpha_l d_l).$$

- Iteration 3 $= \lceil \frac{k}{2} \rceil$

$$d_{l-2} = \frac{1}{\alpha_{l-2}}(\hat{d}_{l-2} - (1-\alpha_{l-2})d_{l-3}) \text{ and } d_{l-2} = \frac{1}{1-\alpha_l}(\hat{d}_l - \alpha_l d_l).$$

Again, if they are close enough, the deBoor points d_i's are used as the new deBoor points.

Eck and Hadenfeld use a little different approach[16]. Two sets of the new deBoor points are computed. The deBoor points in each set are computed successively as follows.

$$d_i^I = \begin{cases} \hat{d}_i, & 0 \le i \le l-k+1, \\ \frac{1}{\alpha_i}(\hat{d}_i - (1-\alpha_i)d_{i-1}), & l-k+2 \le i \le l, \\ \hat{d}_{i+1}, & l+1 \le i \le n. \end{cases}$$

$$d_i^{II} = \begin{cases} \hat{d}_{i+1}, & n \ge i \ge l, \\ \frac{1}{1-\alpha_{i+1}}(\hat{d}_{i+1} - \alpha_{i+1}d_{i+1}), & l-1 \ge i \ge l-k-1, \\ \hat{d}_i, & l-k \ge i \ge 0. \end{cases}$$

Then, real numbers μ_i, $l-k \le i \le l-1$, are introduced to get the new deBoor points as follows.

$$d_i = \begin{cases} \hat{d}_i, & 0 \le i \le l-k-1, \\ (1-\mu_i)d_i^I + \mu_i d_i^{II}, & l-k \le i \le l-1, \\ \hat{d}_{i+1}, & l \le i \le n. \end{cases}$$

These μ_i's are optimized according to L_2- and $L_\infty-$ error minimization techniques. A similar method is found in Brou's work[7]. In these methods, the pointwise norms below have been used[7, 16, 53].

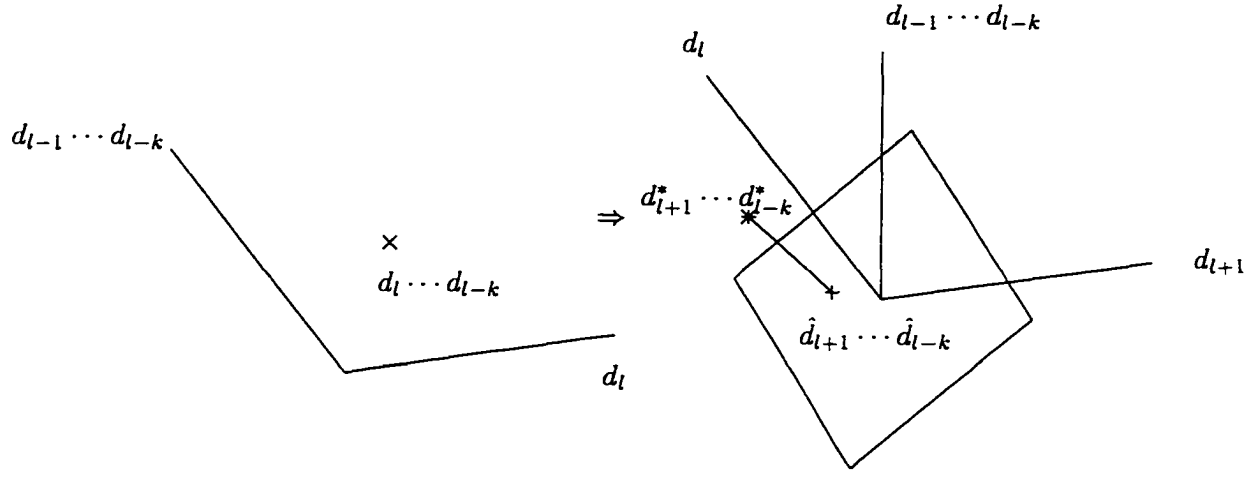


Figure 5.2: A geometrical view of inserting a single knot

$$d(X(t), \hat{X}(t)) = \mathrm{Max}_{t \in (T_k,\ T_{n-1}]} \|X(t) - \hat{X}(t)\|_2 \text{ or}$$

$$d(X(t), \hat{X}(t)) = \mathrm{Max}_{t \in (T_k,\ T_{n-1}]} \|X(t) - \hat{X}(t)\|_\infty.$$

However, a general error optimization technique can be applied to find the closest deBoor points. The problem is now to find the closest deBoor points $d_{l-k+1}, \cdots, d_l$ such that the error function $E = \sum_{i=l-k+1}^{l} (d_i - \dot{d}_i)^2$ is minimum, where

$$\begin{aligned}
\hat{d}_{l-k+1} &= \dot{d}_{l-k+1}, \\
\hat{d}_i &= \alpha_i \dot{d}_i + (1 - \alpha_i)\dot{d}_{i-1},\ i = l-k+2, \cdots, l \\
&\quad \text{where } \alpha_i = \frac{\hat{T}_{l+1} - \hat{T}_i}{\hat{T}_{i+k} - \hat{T}_i}, \\
\hat{d}_{l+1} &= \dot{d}_l.
\end{aligned}$$

The general error optimization is applied here with a global norm, i.e.,

$$d(X(t),\ \hat{X}(t)) = \sum_{i=l-k+1}^{l} (d_i - \dot{d}_i)^2.$$

This global norm seems to be very close to $\int \|X(t) - \hat{X}(t)\|^2 dt$. However, $\dot{d}_i$'s have yet to be determined to compute d_i's. Let's have a close look at (5.7). Geometrically, it is a

Table 5.1: Original deBoor points

i	0	1	2	3	4	5	6
x	1.0	1.7412	5.9417	4.6295	4.3402	5.4347	7.4
y	4.7	1.1161	3.1431	4.0986	5.0626	7.6322	7.6

linear transformation from a k dimensional space to a $k+1$ dimensional space as shown in Figure 5.2. Now, in the $k+1$-dimensional space, the above error function becomes

$$E = (d_{l-k+1} - \hat{d}_{l-k+1})^2 + (\alpha_{l-k+2}d_{l-k+2} + (1-\alpha_{l-k+2})d_{l-k+1} - \hat{d}_{l-k+2})^2 + \cdots + (d_l - \hat{d}_{l+1})^2.$$

Then, a linear system of k equations with k unknown d_i's is obtained from the k partial derivatives $\frac{\partial E}{\partial d_i}$ set to be 0. The linear system is

$$Ad = B,$$

where $A =$

$$\begin{pmatrix}
1+(1-\alpha_{l-k+2})^2 & \alpha_{l-k+2}(1-\alpha_{l-k+2}) & 0 & \cdots & 0 & 0 \\
(1-\alpha_{l-k+2})\alpha_{l-k+2} & \alpha_{l-k+2}^2+(1-\alpha_{l-k+3})^2 & 0 & \cdots & 0 & 0 \\
0 & (1-\alpha_{l-k+3})\alpha_{l-k+3} & \alpha_{l-k+3}^2+ & \cdots & 0 & 0 \\
 & & (1-\alpha_{l-k+4})^2 & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \alpha_{l-1}^2+ & \alpha_l(1-\alpha_l) \\
 & & & & (1-\alpha_l)^2 & \\
0 & 0 & 0 & \cdots & (1-\alpha_l)\alpha_l & \alpha_l^2+1
\end{pmatrix},$$

$B_i = \hat{d}_{l-k+i} + (1-\alpha_{l-k+1+i})\hat{d}_{l-k+1+i}$, $i = 1, \cdots, k$, and $d = (d_{l-k+1}, \cdots, d_l)^T$. Note that the solution here is the optimum because of linearity.

Example 5.4 Knot removal is applied to a B-spline curve in Figure 5.3 with data points in Table 5.1. For each i, $i = 4, 5, 6$, the optimal d_i's and the sums of squared differences,

Table 5.2: New deBoor points

i	0	1	2	3	4	5
x	1.0	1.8843	7.3264	3.0775	5.5778	7.4
y	4.7	1.1015	4.2295	3.8508	7.6176	7.6

i.e., E are computed as follows.

l	E		d_{l-4}	d_{l-3}	d_{l-2}	d_{l-1}
4	5.7712	x	0.4733	5.1158	5.2334	4.2524
		y	4.0145	0.9599	4.4246	4.9484
5	0.7449	x	1.8843	7.3264	3.0775	5.5778
		y	1.1015	4.2295	3.8508	7.6176
6	0.7882	x	5.9213	4.3077	4.0812	7.2776
		y	3.1923	4.1383	6.1888	7.8951

$\hat{T}_5$ and $\hat{d}_5$ are removed from the original B-spline curve because of its smallest difference from the original curve. The original knot vector T and the new one $\hat{T}$ are as follows.

T	0	0	0	0	1/4	1/2	3/4	1	1	1	1
$\hat{T}$	0	0	0	0	1/4		3/4	1	1	1	1

We could compute the deBoor points d_i's more efficiently by recognizing that (5.7) transforms a k dimensional space to a hyperplane in a $k+1$ dimensional space, which is linear. Suppose the hyperplane is $\sum_{i=l-k+1}^{l+1} a_i d_i = 0$. Let d^* be the point on the plane which is the closet to $\hat{d}$. Then, d^* is computed as follows.

$$d_i^* = \hat{d}_i + t a_i,$$

where (a_i) is the norm vector of the hyperplane. Substituting these d_i^* into the hyperplane,

$$\sum a_i(\hat{d}_i + t a_i) = 0.$$

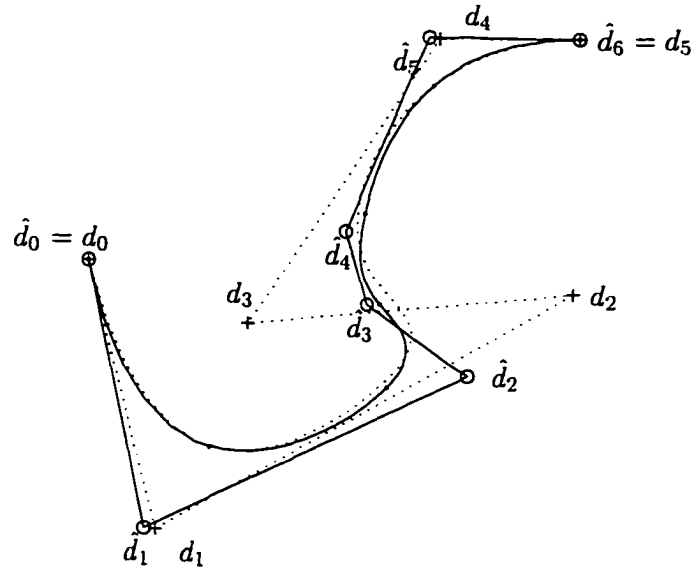


Figure 5.3: Removing a single knot
— : original curve and its control polygon
$\cdots$: approximate curve and its control polygon

Then,

$$t = \frac{\sum a_i \hat{d}_i}{\sum a_i^2}.$$

Thus, the closest point on the hyperplane to $\hat{d}$ is

$$d_i^* = \hat{d}_i - \frac{\sum a_i \hat{d}_i}{\sum a_i^2} a_i.$$

The problem becomes to find the deBoor points d_{l-k+1}, d_{l-k+2}, $\cdots$, d_l so that

$$\begin{aligned}
\hat{d}_{l-k+1} + t a_{l-k+1} &= d_{l-k+1}, \\
\hat{d}_i + t a_i &= \alpha_i d_i + (1-\alpha_i) d_{i-1}, \; i = l-k+2, \; \cdots, \; l \\
&\quad \text{where } \alpha_i = \frac{\hat{T}_{l+1} - \hat{T}_i}{\hat{T}_{i+k} - \hat{T}_i}, \\
\hat{d}_{l+1} + t a_{l+1} &= d_l,
\end{aligned}$$

where a_i's are the elements of the norm vector on the hyperplane and $t = -\frac{\sum a_i \hat{d}_i}{\sum a_i^2}$.

Example 5.5 This new method for knot removal is applied to the same curve in the previous example. For $l = 4, \; 5, \; and 6$, t and the distance between d^* and $\hat{d}$ are computed.

Table 5.3: Errors between the original curves and the resulting curves

E	1 knot removed	2 knots removed
chordal	0.5986	4.1005
Foley	0.0078	0.1832
universal	0.0048	0.0108

l	$\|tA\|^2$	t
4	5.7712	1.0283
5	0.74449	0.0584
6	0.7882	−0.4427

Note that $\|tA\|^2$ are the same as E in the previous method.

5.3 Knot removal applied to different parametrizations

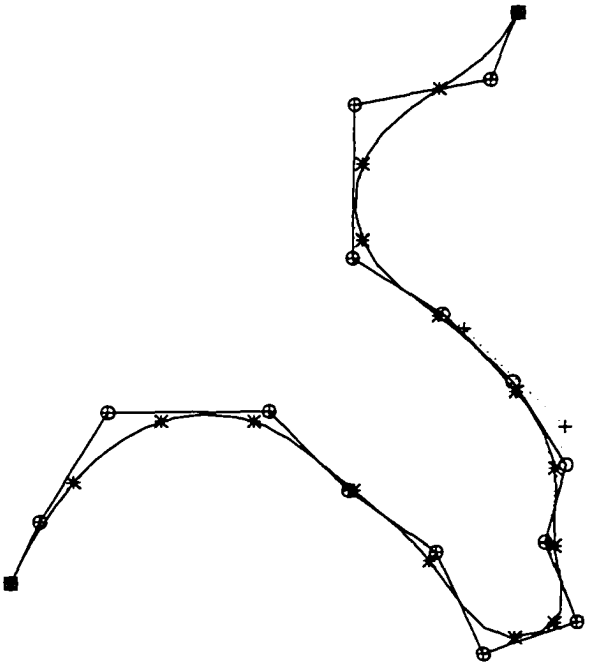
Occasionally, we are given some freedom to choose data points. However, it is not easy to select the best set of data points at the beginning in terms of the number of data points and the quality of the resulting curve. It is often believed that if more points are used, then better curves are obtained. If that is the case, knot removal can be applied to the resulting curves which might have removable knot values and deBoor points because the curves might have obtained with more than necessary number of data points. More compact representation is obtained removing such removable knot values and corresponding deBoor points.

In Figure 5.4, 5.5, and 5.6, knot removal is applied to several different curves obtained through three different parametrizations.

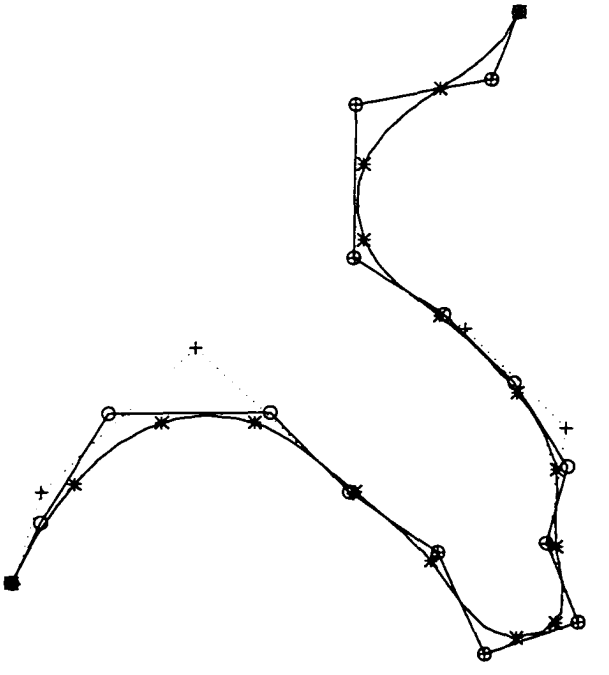
Very close curves are obtained when one or two knots are removed. The sums of squared differences between the original curves and the resulting curves are computed in Table 5.3. Though the differences between parametrizations is not so significant in some cases, the error with universal parametrization is less than with other parametrizations.

5.4 Summary

In this chapter, an optimal and also efficient method was presented to compute the new deBoor points in knot removal with a global norm. The idea of the new method is based on the geometrical view of knot insertion and knot removal. The new method was applied to the curves obtained by various parametrizations. Closer curves to original curves could be generated with the same number of deBoor points removed when the original curves were obtained from universal parametrization. This suggests once again that universal parametrization is a reasonable choice for parametrization in B-spline curve and surface.

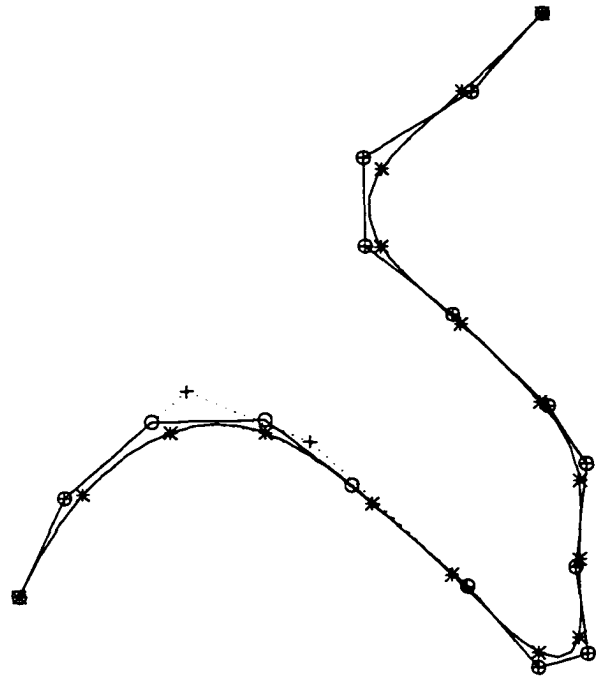


(a) Removing 1 knot

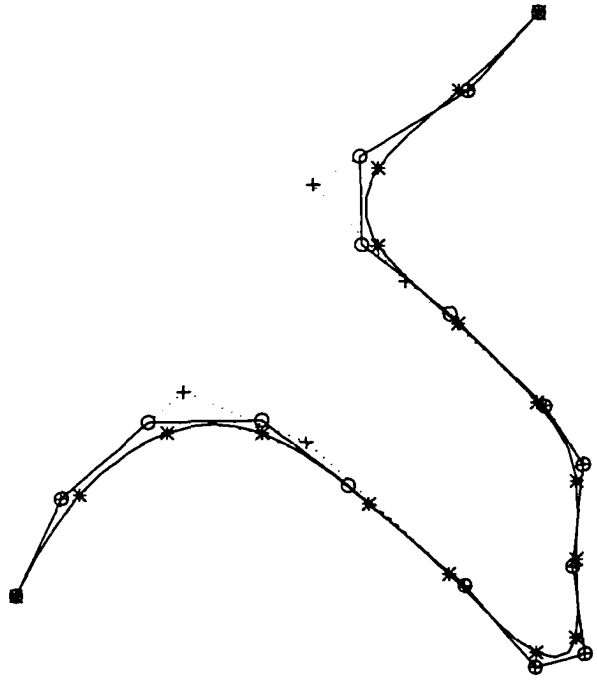


(b) Removing 2 knots

Figure 5.4: Removing knots on a curve via chordal parametrization
— : original curve and its control polygon, ··· : approximate curve and its control polygon,
o : control points for the original curve, and + : control points for the original curve

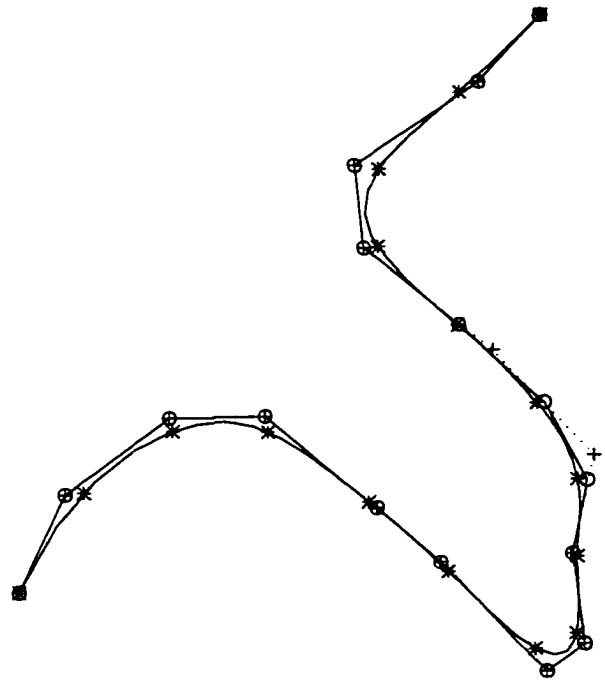


(a) Removing 1 knot

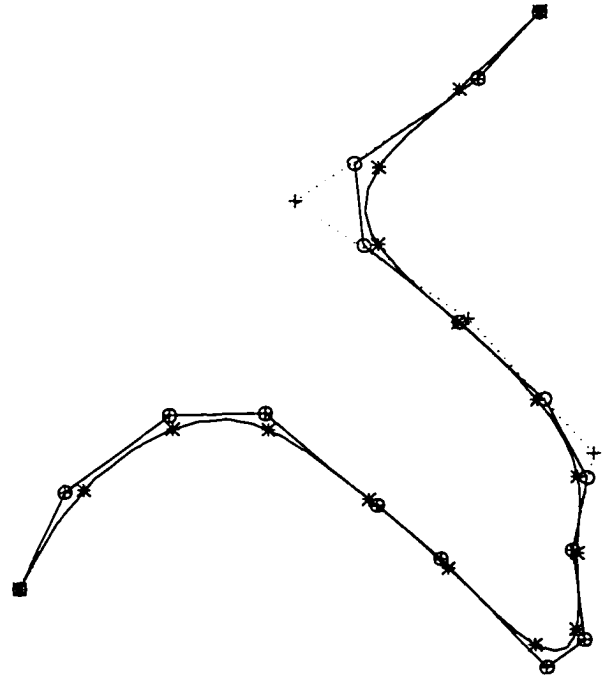


(b) Removing 2 knots

Figure 5.5: Removing knots on a curve via Foley parametrization
— : original curve and its control polygon, $\cdots$: approximate curve and its control polygon,
o : control points for the original curve, and + : control points for the original curve



(a) Removing 1 knot



(b) Removing 2 knots

Figure 5.6: Removing knots on a curve via universal parametrization
— : original curve and its control polygon, $\cdots$: approximate curve and its control polygon,
o : control points for the original curve, and + : control points for the original curve

Chapter 6
Conclusion

In this paper, we have tried to make use of the nature of B-splines $N_{i,k}(t)$ to improve the performance of the existing parametrizations. First, most parameter values are used as knot values. It gives us nice-looking curves. More specifically, small bulges are obtained between data points. The computation of B-spline interpolation is faster and simpler. However, this scheme doesn't work well in odd orders k.

On the other hand, universal parametrization works well in all orders k. It also gives us more natural looking curves while the curves are transformation invariant. Importantly, the new method has the semi-localness property with respect to data points. It means that the effect of an altered data point is mostly on its neighborhood. This is very critical in interactive modeling. The computation of interpolation is simpler and faster because the selection of parameter and knot values does not depend on data points at hand. Only deBoor points need to be recomputed but not $N_{i,k}(t)$. This scheme works well in any order k. It makes the new scheme even more natural. The effectiveness and efficiency of the scheme holds in closed curve cases as well. In this dissertation, we applied universal parametrization to a few examples of 2D curves and 3D surfaces. However, it worked well in most of our experimentations.

In summary, the new parametrization has some advantages over conventional parametrizations as follows.

1. The resulting curves and surfaces look more natural.
2. The curves are invariant under linear transformation.
3. It works well in any order k.
4. It has the semi-localness property with respect to data points.

5. The computation is efficient.

The fact that the resulting curves are more natural is more obvious from the fairness evaluation in Chapter 4. The curvature plots look very nice with the new parametrization. The fact was confirmed once again when knot removal was performed to the resulting curves in Chapter 5. With the new parametrization, we could obtain very close curves to original curves, removing the same number of knots.

Two major contributions for curve and surface fitting have been accomplished in this dissertation. One is the new universal parametrization which works wonderfully in most cases. The other is an efficient and effective knot removal method. It gives us very close curves to original curves.

The possibilities of future research can be listed as follows.

- Chordal parametrization seems to be able to provide us with curves of small total curvatures while universal parametrization gives us nice looking curves with normal curvature plots. It will be interesting to have a variation of chordal parametrization mixed with the idea of universal parametrization.
- There is a belief that if more data points are used, then better curves can be generated. However, the interpolation is not so efficient with the large amount of data points because the involved matrix is so large. It seems that the interpolation could be efficient with universal parametrization even with the large amount of data points. It is because the matrix A in (2.1) does not depend on given data points with the new method.
- The newly proposed method for knot removal needs to be expanded for B-spline surfaces.

- If we could develop an efficient interpolation with the large amount of data points, then an intergration is also possible to make a compact representation of the curve from a free form curve. This is a quite difficult problem without an efficient interpolation computation or an efficient knot removal method.

Bibliography

[1] D.H. Baliand. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1983.

[2] Phillip J. Barry. An introduction to blossoming. In Ronald N. Goldman and Tom Lyche, editors, *Knot Insertion and Deletion Algorithms for B-Spline Curves and Surfaces*, pages 1–10. SIAM, 1993.

[3] Wolfgang Böhm, Gerald Farin, and Jurgeo Kahmann. A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, 1:1–60, 1984.

[4] Mark Berman. Automated smoothing of image and other regularly spaced data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):460–468, 1994.

[5] W. Boehm. Efficient evaluation of splines. *Computing*, 33:171–177, 1984.

[6] Wolfgang Boehm. Inserting new knots into b-spline curves. *Computer Aided Design*, 12(4):199–201, 1980.

[7] J.C. Koua Brou. A knot removal strategy for spline curves. In P.J. Laurent, A. Le Méhauté, and L.L. Schumaker, editors, *Curves and Surfaces in Geometric Design*, pages 277–284. A K Peters, 1994.

[8] George Celniker and Dave Gossard. Deformable curve and surface finite-elements for free-form shape design. In *Proc. SIGGRAPH*, pages 257–266, 1991.

[9] Costanza Conti and Alessandra Sestini. Choosing nodes in parametric blending function interpolation. *Computer-Aided Design*, 28(2):135–143, 1995.

[10] M.G. Cox. The numerical evaluation of b-splines. *Journal of the Institute of Mathematics and Its Applications*, pages 134–149, 1972.

[11] Carl de Boor. On calculating with b-splines. *Journal of Approximation Theory*, 6:50–62, 1972.

[12] Carl de Boor. *A Practival Guide to Splines*. Springer-Verlag, 1978.

[13] W.L.F. Degen. Best approximations of parametric curves by splines. In Tom Lyche and Larry L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design II*, pages 171–184. Academic Press, 1992.

[14] J. Dill. An applcation of color graphics to the display of surface curvature. *Computer Graphics*, 15:153–161, 1981.

[15] M. Eck and J. Hadenfeld. A stepwise algorithm for converting b-splines. In P.J. Laurent, A. Le Méhauté, and L.L. Schumaker, editors, *Curves and Surfaces in Geometric Design*, pages 131–138. A K Peters, 1994.

[16] Matthias Eck and Jan Hadenfeld. Knot removal for b-spline curves. *Computer Aided Geometric Design*, 12:259–282, 1995.

[17] G. Farin, G. Rein, N. Sapidis, and A.J. Worsey. Fairing cubic b-spline curves. *Computer Aided Geometric Design*, 4:91–103, 1987.

[18] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design - A Practical Guide.* Academic Press, Inc., 1992.

[19] Thomas A. FOLEY. A shape preserving interpolant with tension controls. *Computer Aided Geometric Design*, 5:105–118, 1988.

[20] Thomas A. Foley and Gregory M. Nielson. Knot selection for parametric spline interpolation. In Tom Lyche and Larry L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 261–271. Academic Press, Inc., 1989.

[21] David R. Forsey and Richard H. Bartels. Surface fitting with hierachical splines. *ACM Transactions on Graphics*, 14(2):134–161, 1995.

[22] T.N.T. Goodman and K. Unsworth. Shape preserving interpolation by curvature continuous parameter curves. *Computer Aided Geometric Design*, 5:323–340, 1988.

[23] Hans Hagen and Guido Schulze. Automatic smoothing with geometric surface patches. *Computer Aided Geometric Design*, 4:231–235, 1987.

[24] Mark Halstead, Michael Kass, and Tony DeRose. Efficient, fair interpolation using catmull-clark surfaces. In *Proc. SIGGRAPH Annual Series (1993)*, pages 35–44, 1993.

[25] David Handscomb. Knot-elimination: Reversal of the oslo algorithm. *International Series of Numerical Mathematics*, 81:103–111, 1987.

[26] Francis S. Hill. *Computer Graphics.* Macmillan, 1990.

[27] Josef Hoschek. Bézier curves and surface patches on quadrics. In Tom Lyche and Larry L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design II*, pages 331–342. Academic Press, 1992.

[28] Josef Hoschek and Dieter Lasser. *Fundamentals of Computer Aided Geometric Design.* A K Peters, 1993.

[29] Hsieh S. Hou and Harry C. Andrews. Cubic splines for image interpolation and digital filtering. *IEEE Transactions on Accoustics, Speech, and Signal Processing*, 26(6):508–517, 1978.

[30] Peter Lancaster and Kestutis Salkauskas. *Curve and Surface Fitting : An Introduction.* Academic Press, Inc., 1986.

[31] E.T.Y. Lee. A simplified b-spline computation routine. *Computing*, 29:365–371, 1982.

[32] E.T.Y. Lee. Comments on some b-spline algorithms. *Computing*, 36:229–238, 1986.

[33] E.T.Y. Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 21(6):363–370, 1989.

[34] E.T.Y. Lee. Energy, fairness, and a counterexample. *Computer-Aided Design*, 22(1):37–40, 1990.

[35] Choong-Gyoo Lim. A universal parametrization in b-spline curve and surface interpolation. *Submitted for publication in Computer Aided Geometric Design*, 1997.

[36] Fenqiang Lin and W.T Hewitt. Expressing coons-gordon surfaces and NURBS. *Computer-Aided Design*, 26(2):145–155, 1994.

[37] Bradley J. Lucier. Wavelets and image compression. In Tom Lyche and Larry L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design II*, pages 391–400. Academic Press, 1992.

[38] T. Lyche and K. Morkon. A metric for parametric approximation. In P.J. Laurent, A. Le Méhauté, and L.L. Schumaker, editors, *Curves and Surfaces in Geometric Design*, pages 311–318. A K Peters, 1994.

[39] Tom Lyche and Knut Morken. Knot removal for parametric b-spline curves and surfaces. *Computer Aided Geometric Design*, 4:217–230, 1987.

[40] Tom Lyche and Knut Morken. A data-reduction strategy for splines with applications to the approximation of functions and data. *IMA Journal of Numerical Analysis*, 8:185–208, 1988.

[41] H. Meier and H. Nowacki. Interpolating curves with gradual changes in curvature. *Computer Aided Geometric Design*, 4:297–305, 1987.

[42] Michael E. Mortenson. *Geometric Modeling*. John Wiley & Sons, Inc., 1997.

[43] Gregory M. Nielson. Coordinate free scattered data interpolation. In C.K. Chui, Larry L. Schumaker, and F.I. Utreras, editors, *Topics in Multivariate Approximation*, pages 175–184. Academic Press, Inc., 1987.

[44] Gregory M. Nielson and Thomas A. Foley. A survey of applications of an affine invariant norm. In Tom Lyche and Larry L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 445–463. Academic Press, Inc., 1989.

[45] Les Piegl and Wayne Tiller. Software-engineering approach to degree elevation of b-spline. *Computer-Aided Design*, 24(1):17–28, 1994.

[46] Helmit Pottmann. The geometry of tchebycheffian splines. *Computer Aided Geometric Design*, 10:181–210, 1993.

[47] Lyle Ramshaw. Blossoms are polar forms. *Computer Aided Geometric Design*, 6:323–358, 1989.

[48] Paul Sablonniere. Discrete bezier curves and surfaces. In Tom Lyche and Larry L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design II*, pages 497–515. Academic Press, 1992.

[49] N. Sapidis and G. Farin. Automatic fairing algorithm for b-spline curves. *Computer-Aided Design*, pages 121–129, 1990.

[50] Hans-Peter Seidel. A new multiaffine approach to b-spline. *Computer Aided Geometric Design*, 6:23–32, 1989.

[51] Yohanes Stefanus and Ronal Goldman. Discrete convolution schemes. In Tom Lyche and Larry L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design II*, pages 585–596. Academic Press, 1992.

[52] B.-Q. Su and D.-Y. Liu. *Computational Geometry*. Academic Press, 1989.

[53] W. Tiller. Knot-removal algorithms for nurbs curves and surfaces. *Computer-Aided Design*, 24(8):445–453, 1992.

[54] Kazuo Toraichi, Sai Yang, Masaru Kamada, and Ryoichi Mori. Two-dimensional spline interpolation for image reconstruction. *Pattern Recognition*, 21(3):275–284, 1988.

[55] Michael Unser, Akram Aldroubi, and Murray Eden. Fast b-spline transforms for continuous image construction and interpolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):277–285, 1991.

[56] Michael Unser, Akram Aldroubi, and Murray Eden. B-spline signal processing: Part i - theory. *IEEE Transactions on Signal Processing*, 41(2):821–833, 1993.

[57] Michael Unser, Akram Aldroubi, and Murray Eden. B-spline signal processing: Part ii - efficient design and applications. *IEEE Transactions on Signal Processing*, 41(2):834–848, 1993.

[58] Tzvetomir Ivanov Vassilev. Fair interpolation and approximation of b-splines by energy minimization and points insertion. *Computer Aided Design*, 28(9):753–760, 1996.

Vita

Choong-Gyoo Lim received his bachelor of science and master of education degrees in Mathematics Education at Seoul National University, Seoul, Korea, in 1988 and 1990, respectively. He is now completing his doctoral studies in Computer Science at Louisiana State University, Baton Rouge, and will receive his degree of Doctor of Philosophy in May, 1998. His research interests include Computer Aided Geometric Modeling, Computer Graphics, Computer Vision, and Machine Learning.

DOCTORAL EXAMINATION AND DISSERTATION REPORT

Candidate: Choong-Gyoo Lim

Major Field: Computer Science

Title of Dissertation: A Universal Parametrization in B-Spline Curve and Surface Interpolation and Its Performance Evaluation

Approved:

Major Professor and Chairman

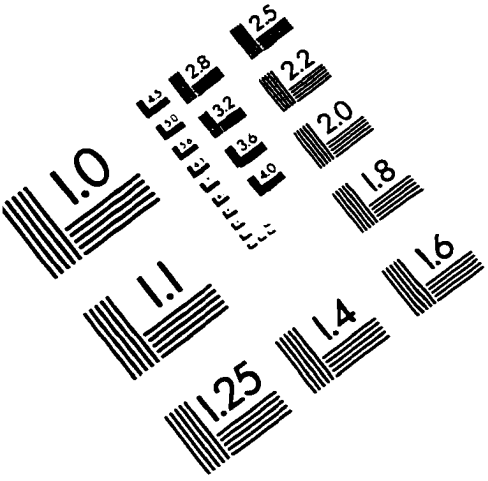
Dean of the Graduate School

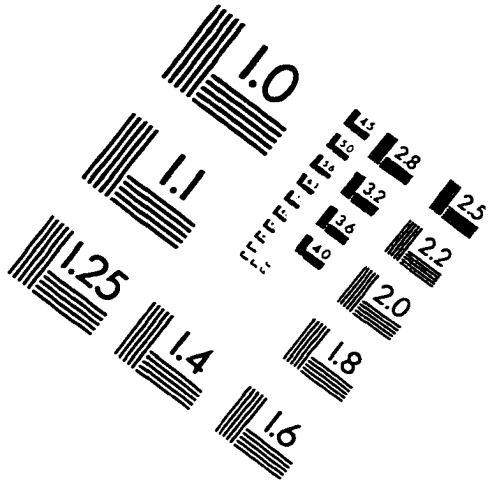
EXAMINING COMMITTEE:

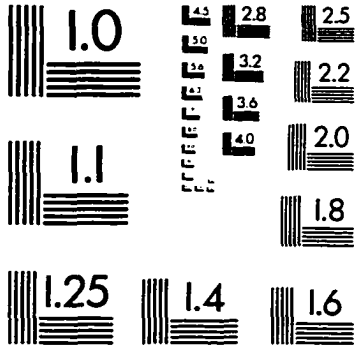
Date of Examination:

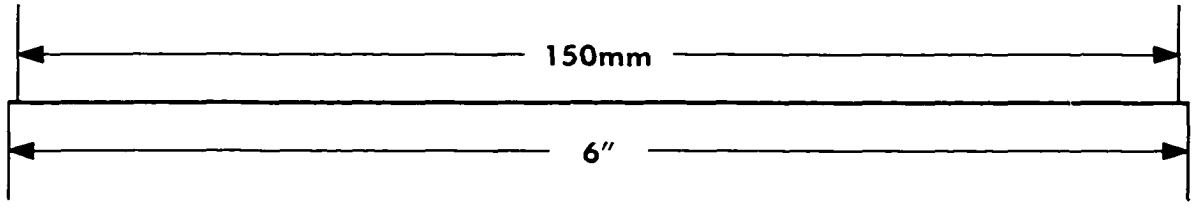
April 9, 1998

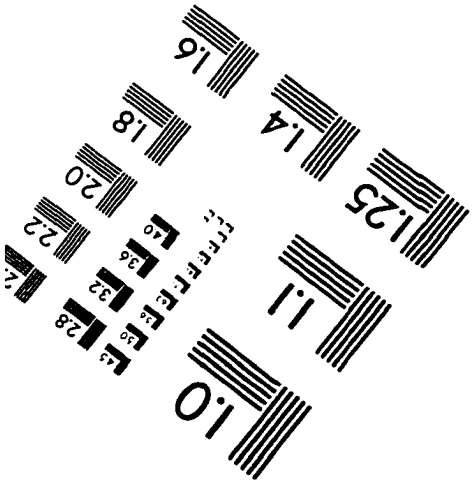
IMAGE EVALUATION
TEST TARGET (QA-3)

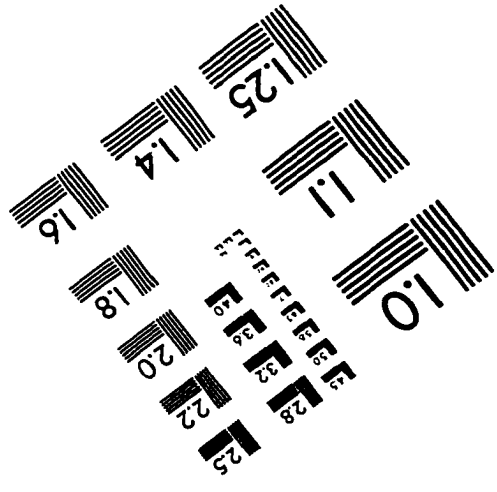












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989