

Virtual knot technique for curve fitting of rapidly varying data

Hsun-Chang Hsieh, Wen-Tong Chang*

Department of Communication Engineering & Signal and Image Processing Laboratory, National Chiao Tung University, Hsinchu, Taiwan, ROC

Received September 1991; revised December 1992

Abstract

Curve fitting with piecewise splines for rapidly varying data has been a difficult problem. Often, the curve exhibits unwanted "wiggles" around those data points whose locations change rapidly in comparison to their neighboring data points. Many methods were proposed to solve this problem using non-homogeneous tension spline or fairing process. In this paper, a new approach using the concept of virtual knots is presented. Since the occurrence of the problem is due to the inconsistency between the parametric spans of the knots and the geometric spans of the data points, a number of virtual knots are inserted into the original knot sequence such that the parametric spans are made more consistent with the geometric spans of the data. With this parametric adjustment, a regularization process is used to find the desired curve. Experimental results show that this technique is efficient to adjust the fairness of the desired curve. This technique can be applied to different spline spaces. In this paper, the usages of this technique in linear B-spline and cubic B-spline spaces are demonstrated.

Key words: Curve fitting; Fairness; Virtual knot technique; Computer-aided design; Interpolation; Regularization; Non-uniform parameterization

1. Introduction

Curve fitting has extensive applications in science and engineering such as computer-aided geometric design (Wu et al., 1977; Forrest, 1980; Hartley and Judd, 1980; Cohen and Lyche, 1980), computer graphics, image processing, computer vision (Medioni and Yasumoto, 1987; Namane and Sid-Ahmed, 1990; Lane and Riesenfeld, 1980), computer-aided animation (Yang et al., 1986), data analysis (Prenter and Westwater, 1986; Hou and Andrews, 1978; Keys, 1981; Carlson and Fritsch, 1985) and so on.

*Corresponding author. Email: wtchang@twncu01.bitnet.

Hence, various efficient methods have been proposed in the literature and excellent results have been shown. However, for data points with non-uniform span, the *fairness* (Farin et al., 1987) of the curve is difficult to control due to the inconsistency between the parametric spans and the geometric spans of data points. Therefore, many methods were proposed to solve this problem (Nielson, 1974; Salkauskas, 1974; Foley, 1987). In this paper, a new technique which can be applied to various spline spaces to obtain fairer curve is proposed.

The interpolant of uniform parameterization and homogeneous tension are not effective if used to fit a sequence of data that changes rapidly. Some unwanted “wiggles” are often generated around those data points whose locations change rapidly in comparison with their neighboring data points. This gives the curve an unpleasant look. To overcome this problem, many methods using non-homogeneous interpolant by changing the tension, such as the weighted spline (Lancaster and Salkauskas, 1986, Ch. 4; Salkauskas, 1974; Foley, 1987), ν -splines (Nielson, 1974), tensioned splines (Cohen, 1987; Foley and Ely, 1989), β -splines and splines with shape parameters (DeRose and Barsky, 1988), have been proposed. In our approach, instead of changing the tension of the splines, we change the parameterization by inserting additional *virtual knots*. (For convenience, the “knot” is used to indicate the data points in this paper.) The virtual knots are pseudo-knots added to the original knot sequence to alter the parametric distance between the actual knots. They are inserted between actual knots to adjust the parametric spans. By this, the parameter spans are made more consistent with the geometric spans such that interpolant with uniform parameterization and homogeneous tension can be useful to fit rapidly varying data.

This method is general in that it can be applied to various spline spaces. In the formulation, the desired curve is expressed as the weighted sum of a set of spline bases. In conventional methods, one spline basis is assigned for each data point. In the proposed method, in addition to the actual data points, spline bases are also assigned for the virtual knots. This makes the system under-determined. Thus, a regularization process (Lee and Pavlidis, 1988; Terzopoulos, 1986; Groetsch, 1984) is then used to constrain the formulation. The objective of the regularization is to confine the solution to be unique and to determine the corresponding weighting variables for all the splines bases. In our method, smoothness is used as the constraint in designing the regularization operator. An objective function is introduced to measure the smoothness of the desired curve. This desired curve is solved by minimizing this objective function. With this approach, a fair (or pleasant) curve that fits the rapidly varying data can be obtained.

In the following section, we give a curve generated from a set of rapidly varying data by conventional cubic B-splines to illustrate the need for fairness adjustment. Several popular methods that were proposed to solve this problem are also briefly reviewed. Then, the idea of virtual knot insertion is introduced and the curve generated by this proposed method is shown for comparison. In Section 3, the derivation of the proposed method and the process of regularization is discussed in details. Also, the insertion schemes of the virtual

knot are introduced. In Section 4, applications of this technique with cubic and linear B-spline interpolants are presented and several experimental results are shown. Finally, a conclusion is given in Section 5.

2. The fitting of rapidly varying data

The need to fair a curve in fitting a set of rapidly varying data is discussed in this section. The fairing process is to modify a curve such that the new one is fairer than the old one (Farin et al., 1987; Lee, 1990). Then the idea of virtual knot insertion for generating fairer curve is introduced.

2.1. An example

Let an n -dimensional curve fitting problem be stated as follows.

Given a set of data $\{\mathbf{p}_i \mid i = 0, 1, \dots, K-1\}$, to find a curve $\mathbf{f}(s)$ such that $\mathbf{f}(s_i)$ passes through \mathbf{p}_i for $i = 0, 1, \dots, K-1$.

where $\mathbf{p}_i = [p_{i1}, p_{i2}, \dots, p_{in}]$ is an n -dimensional vector and $\mathbf{f}(s) = [f_1(s), f_2(s), \dots, f_n(s)]$ is an n -dimensional curve function. In planar curve fitting, the dimension is two. Throughout this paper, these known data points \mathbf{p}_i are referred to as *actual knots*. The parameter s is a scalar parameter along the curve. In curve fitting, parametric representation is widely used. If the parameter s is chosen to be uniformly distributed along knots (i.e. $s_{i+1} - s_i = \text{constant}$), it is called *uniform parameterization* and the knots are said to uniformly span over the *parameter space* of s .

It is found that uniform parameterization may render undesirable fluctuation (Foley, 1987) when the parameterization is not consistent with the geometric spans. This makes the resulted curve look “unpleasant”. The following example, which fits a set of *rapidly varying data*, demonstrates this phenomenon (Salkauskas, 1974; Foley, 1987).

Find a curve $\mathbf{f}(s)$ to fit the data set $\{\mathbf{p}_0 = (0, 0), \mathbf{p}_1 = (1, 0), \mathbf{p}_2 = (2, 0), \mathbf{p}_3 = (3, 0), \mathbf{p}_4 = (4, 0), \mathbf{p}_5 = (5, 4), \mathbf{p}_6 = (6, 4), \mathbf{p}_7 = (7, 4), \mathbf{p}_8 = (8, 4), \mathbf{p}_9 = (9, 4)\}$.

When the data are fitted by the uniform cubic B-spline (Ballard and Brown, 1982, Section 8.2.6) with the second derivative of $\mathbf{f}(s)$ outside the two end points set to zero as the end conditions, fluctuations happen. The curve is plotted as dashed line in Fig. 1(a). The dashed curve exhibits fluctuation (or wiggle) around the points where the geometric spans between knots change abruptly (between knot 4 and 5). Here, the *geometric span* is defined as the Euclidean distance between two adjacent knots.

In (Lee, 1990), it has been pointed out that the parametric smoothness does not imply geometric smoothness. Although the cubic B-spline is smooth in parametric space, it does not always offer geometric smoothness. Therefore, parametric smoothness does not guarantee a fair or pleasant curve. A curve is

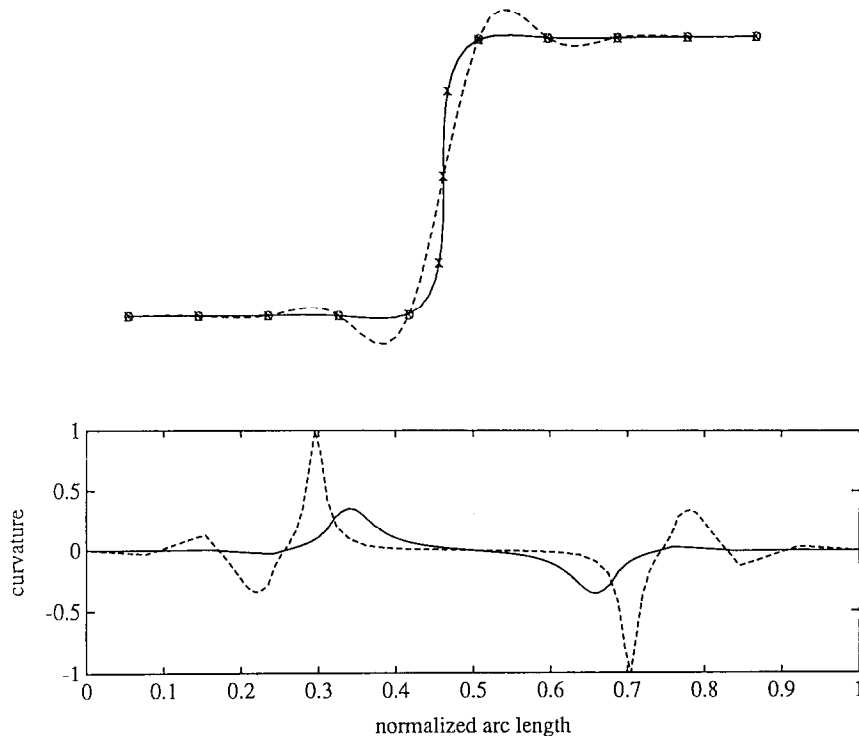


Fig. 1. An example of curve fitting. Circles indicate the actual knots and crosses indicate the new knots after virtual knot insertion. (top) The curve fitted with the classical cubic B-spline method is plotted in dash line. The curve fitted with the virtual knot technique using cubic B-spline basis is plotted in solid line. (bottom) The corresponding curvature plots of both curves.

considered to be “fair” if its curvature is nearly piecewise linear and continuous. In (Farin et al., 1987), the curvature plot is used to examine the fairness of curves successfully. For comparison, the curvature plot of the above curve is plotted in Fig. 1(b). It can be found that the curvature is far from smooth. The problem of fitting *rapidly changing data* (Salkauskas, 1974; Foley, 1987) or *extraneous inflection data* (Barsky, 1984) was addressed by many researchers (Salkauskas, 1974; Barsky, 1984; Foley, 1987). In the following subsection, we will briefly review some of useful methods which had been proposed to fit this kind of data.

2.2. Reviews

For fitting rapidly varying data, various methods had been proposed. In (Salkauskas, 1974), a weighted spline is used to interpolate rapidly varying data. Foley (1987) extended the weighted spline to the so called “weighted bicubic spline” for three-dimensional data. By modifying the natural spline,

they used the cubic spline to minimize the functional

$$\int_a^b w(s) \left[\frac{\partial^2 f(s)}{\partial s^2} \right]^2 ds, \tag{1}$$

where $f(s)$ is the desired curve and $[a, b]$ is the range of the parameter s . The weighting $w(s)$ is not all zero over the domain of interest and $w(s) \geq 0$ for all s . With this design, the solution space would no longer be confined to class $C^2[a, b]$. They used a piecewise cubic interpolant from class $C^1[a, b]$. Though this weighted spline can attack the problem of oscillation, it lacks the continuity of the second derivative of the spline (Foley, 1987).

Another class of splines to remove oscillation is the *tensioned spline* with adjustable tension parameters. For example, Nielson developed the ν -spline (Nielson, 1974) with the following objective functional:

$$\int_a^b \left[\frac{\partial^2 f(s)}{\partial s^2} \right]^2 ds + \sum_{i=0}^{K-1} t_i \left[\frac{\partial f(s)}{\partial s} \right]^2. \tag{2}$$

where parameter t_i is the tension value on the i th knot. In (Barsky, 1984), the ν -spline was juxtaposed with the exponential-based tensioned spline which minimizes

$$\int_a^b \left[\frac{\partial^2 f(s)}{\partial s^2} \right]^2 ds + \sum_{i=0}^{K-1} t_i \left[\int_{s_i}^{s_{i+1}} \frac{\partial f(s)}{\partial s} \right]^2 ds \tag{3}$$

where t_i is the tension value of the i th curve segment over $s = [s_i, s_{i+1}]$. Different from the ν -spline, this formulation introduces tension parameter to each curve segment rather than to each knot.

The common idea of these methods is to change the tension by introducing additional parameters into the spline. Based on this idea, many methods with generalized shape parameters have been proposed (Cohen, 1987; DeRose and Barsky, 1988; Foley and Ely, 1989; Schaback, 1989). Basically, in those methods, the spline is designed to be non-homogeneous. Thus, the solution is found in a non-homogeneous spline space characterized by shape parameters. Beside these, Farin et al. (1987) presented local fairing algorithms to fair a cubic B-spline curves. Also, the smoothing method proposed by Kjellander is useful to improve the fairness of a curve (Kjellander, 1983).

2.3. Insertion of virtual knots

Above we have briefly mentioned the use of non-homogeneous splines for fitting rapidly changing data. As indicated above, an unpleasant curve resulted when interpolant with uniform parameterization and homogeneous tension is used to fit data with non-uniform geometric spans. Instead of adjusting the tension of the interpolant, we propose a new approach by adjusting the parameterization of the interpolant. The parametric span is made consistent

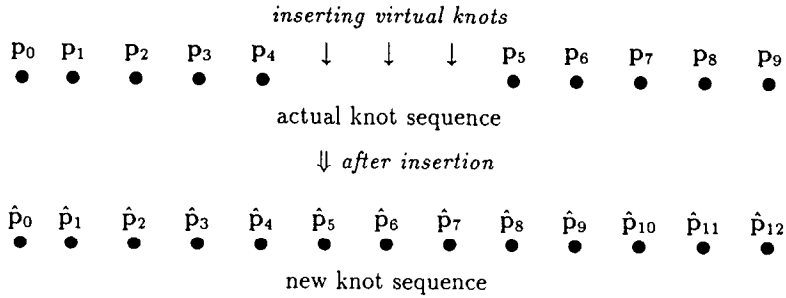


Fig. 2. An example of virtual knot insertion.

with the geometric span of the given data. The parametric adjustment is achieved by inserting virtual knots into the actual knot sequence. With uniform parameterization along the new knot sequence (including virtual knots and actual knots), the parameterization along the actual knot sequence is made non-uniform. Note that the *actual knot* refers to the knot of the original data and the *virtual knot* refers to the additional knot inserted. Since the parameterization is kept uniform in the new knot sequence, the simplicity of those methods with uniform parameterization is preserved. Moreover, the approach only adds additional knots into the original knot sequence, so it can be applied to various interpolation spaces.

To outline the idea of virtual knot, the above mentioned example is used again for illustration. Since the geometric span between actual knots 4 and 5 is large, virtual knots are inserted into this span. The Euclidean distance of this large span is approximately four times larger than that of other spans, so three virtual knots are inserted into this span. Fig. 2 gives a graphic illustration of the insertion of virtual knots.

Let $\{\hat{p}_i = (f_1(\hat{s}_i), f_2(\hat{s}_i))\}$ denote the new knot sequence after the insertion of virtual knots. For the above example, $\hat{p}_i = p_i$ for $i = 0, 1, 2, 3, 4$ and $\hat{p}_{i+3} = p_i$ for $i = 5, 6, 7, 8, 9$, \hat{p}_5 to \hat{p}_7 are inserted virtual knots. If cubic B-spline is used to interpolate these knots, the solution curve can be written as

$$f(s) = [f_1(s), f_2(s)] = \left[\sum_{l=0}^{12} v_{1l} \Psi^c_l(s), \sum_{l=0}^{12} v_{2l} \Psi^c_l(s) \right].$$

where the $\Psi^c_l(s)$ are cubic B-spline bases and the v_{1l} and v_{2l} are the corresponding weighting coefficients. After substituting the bases, the following equations (Ballard and Brown, 1982, Sec. 8.2.6)

$$\frac{1}{6} [v_{k,i-1} + 4v_{k,i} + v_{k,i+1}] = f_k(s_i)$$

for $i = 1, 2, 3, 4, 8, 9, 10, 11$ and $k = 1, 2$ can be obtained. With zero curvature as the end conditions, the following two equations result.

$$v_{k,0} = f_k(s_0), \quad v_{k,12} = f_k(s_{12})$$

for $k = 1, 2$. In total, there are 10 equations with 13 unknowns. These extra unknowns are due to the bases associated with the virtual knots. The locations of virtual knots are left unspecified first and are to be determined by constraints to ensure the fairness of the resultant curve.

Therefore, the control of the fairness of the curve is dependent upon the design of this constraint. A reasonable choice is to maximize the smoothness of the solution curve. The solution can be defined to be the one that minimizes a quantity \mathcal{S} inversely proportional to the smoothness. By minimizing the quantity \mathcal{S} , three additional equations can be obtained by the Euler equations:

$$\frac{\partial \mathcal{S}}{\partial v_{ki}} = 0 \tag{4}$$

for $i = 5, 6, 7$ and $k = 1, 2$. Totally, these equations are used to determine all the weighting coefficients. Subsequently, the locations of the inserted virtual knots can then be determined. The curve obtained by this criterion is shown in Fig. 1(a) (the solid curve). The knots are marked with an "x" (including the original knots and the virtual knots). For comparison, the curvature plot of the curve obtained by this proposed approach is also shown in Fig. 1(b). One can find that it is much smoother than that of the previous curve.

3. The virtual knot technique

The above example illustrates the basic idea of the virtual knot technique. The general formulation of this technique will be discussed in this section.

3.1. Regularization

For interpolation with a set of actual knots $\{\mathbf{p}_i \mid i = 0, 1, \dots, K - 1\}$, assume that $N - K$ virtual knots are to be added into the actual knot sequence. Let the new sequence of knots be denoted as $\{\hat{\mathbf{p}}_i \mid i = 0, 1, \dots, N - 1\}$. To indicate whether a knot is an actual knot or a virtual knot, a corresponding *mark vector* $\mathbf{m} = [m_0, m_1, \dots, m_{N-1}]^T$ is defined.

$$m_k \triangleq \begin{cases} 1, & \hat{\mathbf{p}}_k \text{ is an actual knot,} \\ 0, & \hat{\mathbf{p}}_k \text{ is a virtual knot.} \end{cases} \tag{5}$$

In conventional interpolation, the degree of freedom of the solution space is determined by the number of data points. After inserting the virtual knots, the degree of freedom of the solution space is increased. Therefore, the problem becomes under-determined or ill-posed. Thus, an additional constraint is needed in order to obtain a unique curve. The purpose of the regularization is to form such additional constraint (Terzopoulos, 1986; Lee and Pavlidis, 1988) in order to transform this ill-posed problem into a well-posed one. By introducing an energy functional into the under-determined problem, the solution is defined to be the one that minimizes the value of the energy functional.

The energy functional is defined as the following (Terzopoulos, 1986; Lee and Pavlidis, 1988).

$$\mathcal{E}(f) = \mathcal{S}(f) + \beta \mathcal{C}(f, g) \quad (6)$$

where f is the desired function and g is the known data. The stabilizing functional $\mathcal{S}(f)$ is a measure of the properties of f such as smoothness or curvature and $\mathcal{C}(f, g)$ is a cost functional describing the discrepancy between f and g . The parameter β is a weighting factor for balance.

For our purpose, the insertion of the virtual knots is for the fairness of the curve. Therefore, the stabilizing functional is used as an mechanism to model the curvature of the desired curve. The cost functional is used to control the closeness between the curve and the given data.

For this, we define the stabilizing functional as

$$\mathcal{S}(f) = \int_{\Omega} \left\| \frac{\partial^2 f(s)}{\partial s^2} \right\|^2 ds. \quad (7)$$

where Ω is the domain of interest and s is a parameter along the curve $f(s)$. Note that \mathcal{S} will become the integral of the curvature if f is a two-dimensional curve and s is the displacement along the curve.

For the *cost functional*, a weighted square sum of the distance from the knots to the desired curve is used. That is,

$$\mathcal{C}(f) = \sum_{k=0}^{N-1} m_k [(f(s_k) - \hat{p}_k)]^2 \quad (8)$$

where s_k is the parameter value corresponding to the knot \hat{p}_k . One may notice that the location of the virtual knot \hat{p}_k is unknown until the curve is solved. Therefore, the corresponding m_k of the virtual knot is zero. That is, the distances from those virtual knots to the curve are not taken into account in this formulation.

To minimize the energy functional, the smoothness of the curve and the closeness between the curve and data points are both emphasized. The weighting factor β can be used to control the balance. When β approaches infinity, the curve exactly interpolates the data points. In such a situation, the cost functional should be considered separately and the minimization problem becomes a constrained minimization one (see Appendix A). The advantage to consider both smoothness and closeness together in the energy functional is that a more general system equation can be derived. For interpolation, β is chosen large to constrain the curve to pass through the data points. For approximation, β can be adjusted for desired solution when the data are corrupted with noise.

For a bivariate curve $f(s) = (x(s), y(s))$ with knots $\hat{p}_k = (\hat{x}_k, \hat{y}_k)$, the energy functional can be expressed as

$$\mathcal{E}(f) = \int_{\Omega} \left[\left(\frac{\partial^2 x(s)}{\partial s^2} \right)^2 + \left(\frac{\partial^2 y(s)}{\partial s^2} \right)^2 \right] ds$$

$$\begin{aligned}
 & + \beta \sum_{k=0}^{N-1} m_k [(x(s_k) - \hat{x}_k)^2 + (y(s_k) - \hat{y}_k)^2] \\
 & = \mathcal{E}_x(x) + \mathcal{E}_y(y)
 \end{aligned} \tag{9}$$

where

$$\mathcal{E}_x(x) = \int_{\Omega} \left(\frac{\partial^2 x(s)}{\partial s^2} \right)^2 ds + \beta \sum_{k=0}^{N-1} m_k [x(s_k) - \hat{x}_k]^2, \tag{10}$$

$$\mathcal{E}_y(y) = \int_{\Omega} \left(\frac{\partial^2 y(s)}{\partial s^2} \right)^2 ds + \beta \sum_{k=0}^{N-1} m_k [y(s_k) - \hat{y}_k]^2. \tag{11}$$

The functionals \mathcal{E}_x and \mathcal{E}_y are called the x -component and the y -component of the energy functional, respectively. Since both of them are non-negative and independent, they can be minimized separately. Therefore, the following discussion will focus on the one-dimensional case (e.g. the minimization of \mathcal{E}_x). Extension to multi-dimensional cases can be easily done.

3.2. The system equation

With regularization, the curve fitting problem can be converted into an optimization problem, i.e., to minimize an objective functional. Let the solution space be spanned by the set of bases $\{\phi_i(s) \mid i = 0, 1, \dots, L-1\}$; then, the function $x(s)$ can be expressed as

$$x(s) = \sum_{l=0}^{L-1} v_l \phi_l(s). \tag{12}$$

where v_l is the weighting coefficient related to ϕ_l .

When the number of bases of the solution space then the numbers of knots after insertion are equal (i.e., $L = N$). With (12), we get

$$\begin{aligned}
 \mathcal{E}_x(x(s)) & = \int_{\Omega} \left[\sum_{i=0}^{N-1} v_i \phi_i''(s) \right] \left[\sum_{j=0}^{N-1} v_j \phi_j''(s) \right] ds \\
 & + \beta \sum_{k=0}^{N-1} m_k \left[\sum_{i=0}^{N-1} v_i \phi_i(s_k) - \hat{x}_k \right] \left[\sum_{j=0}^{N-1} v_j \phi_j(s_k) - \hat{x}_k \right] \\
 & = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} v_i v_j \int_{\Omega} \phi_i''(s) \phi_j''(s) ds \\
 & + \beta \left[\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} v_i v_j \sum_{k=0}^{N-1} m_k \phi_i(s_k) \phi_j(s_k) \right. \\
 & \quad \left. - 2 \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} v_i \phi_i(s_k) m_k \hat{x}_k + \sum_{k=0}^{N-1} m_k \hat{x}_k^2 \right].
 \end{aligned} \tag{13}$$

To simplify the expression, we define the following quantities:

$$t_{ij} = \int_{\Omega} \phi_i''(s) \phi_j''(s) ds, \quad (14)$$

$$u_{ki} = \phi_i(s_k). \quad (15)$$

The quantity t_{ij} is called the *relation coefficient* between variables v_i and v_j and u_{ki} is called the *couple coefficient* of basis ϕ_i at knot k . Also, we define the following vectors and matrices:

$$\mathbf{v} = [v_0, v_1, \dots, v_{N-1}]^T,$$

$$\hat{\mathbf{x}} = [\hat{x}_0, \hat{x}_1, \dots, \hat{x}_{N-1}]^T,$$

$$\mathbf{T} = \begin{bmatrix} t_{00} & t_{01} & t_{02} & \cdots & t_{0,N-1} \\ t_{10} & t_{11} & t_{12} & \cdots & t_{1,N-1} \\ t_{20} & t_{21} & t_{22} & \cdots & t_{2,N-1} \\ t_{30} & t_{31} & t_{32} & \cdots & t_{3,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{N-1,0} & t_{N-1,1} & t_{N-1,2} & \cdots & t_{N-1,N-1} \end{bmatrix},$$

$$\mathbf{U} = \begin{bmatrix} u_{00} & u_{01} & u_{02} & \cdots & u_{0,N-1} \\ u_{10} & u_{11} & u_{12} & \cdots & u_{1,N-1} \\ u_{20} & u_{21} & u_{22} & \cdots & u_{2,N-1} \\ u_{30} & u_{31} & u_{32} & \cdots & u_{3,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{N-1,0} & u_{N-1,1} & u_{N-1,2} & \cdots & u_{N-1,N-1} \end{bmatrix},$$

$$\mathbf{M} = \begin{bmatrix} m_0 & 0 & 0 & \cdots & 0 \\ 0 & m_1 & 0 & \cdots & 0 \\ 0 & 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & m_{N-1} \end{bmatrix}.$$

The matrix \mathbf{T} is called *relation matrix* and \mathbf{U} is called *couple matrix*. With these definitions, the energy \mathcal{E}_x can be expressed as

$$\mathcal{E}_x = \mathbf{v}^T \mathbf{T} \mathbf{v} + \beta [\mathbf{v}^T \mathbf{U}^T \mathbf{M} \mathbf{U} \mathbf{v} - 2\mathbf{v}^T \mathbf{U}^T \mathbf{M} \hat{\mathbf{x}} + \hat{\mathbf{x}}^T \mathbf{M} \hat{\mathbf{x}}]. \quad (16)$$

This is a quadratic form. To minimize the energy, the Euler–Lagrange condition must be satisfied, that is

$$\frac{\partial \mathcal{E}_x}{\partial v_i} = 0 \quad (17)$$

for $i = 0, 1, \dots, N-1$. This results in a system of linear equations:

$$(\mathbf{T} + \beta \mathbf{U}^T \mathbf{M} \mathbf{U}) \mathbf{v} = \beta \mathbf{U}^T \mathbf{M} \hat{\mathbf{x}}. \quad (18)$$

This is the general system equation of curve fitting with virtual knots. Note that this is only a sufficient condition for the minimization of \mathcal{E}_x . However, in practice, when proper basis functions are chosen, it is usually a necessary and sufficient condition. That is, the solution of the above equation will not only be a local minimum but also the unique global minimum.

As indicated above, exact interpolation can be obtained only when β is set to infinity. When $\beta \rightarrow \infty$ and all $m_k = 1$ (i.e., no virtual knots are added), the system equation will degenerate to the equation:

$$Uv = \hat{x}, \tag{19}$$

which can be expanded to the following equations:

$$\sum_{k=0}^{N-1} \phi_k(s_i)v_k = \hat{x}_i, \tag{20}$$

for $i = 0, 1, \dots, N - 1$. This is the system equation for classical interpolation. When $\beta \rightarrow \infty$ and not all $m_k = 1$, the situation corresponds to exact interpolation with virtual knots. The system equation for this case is shown in Appendix B. However, from the experimental results, it is found that when β is chosen to be large enough the resulted curve will be almost identical to that obtained by exact interpolation. Therefore, in this work, we choose to define the energy functional as (6), since it is more general.

3.3. Insertion schemes

Inserting virtual knots is equivalent to defining the mark vector. To obtain a fair curve, the virtual knots are usually inserted in such a way that the parametric span is made consistent with the geometric span. In the following, two insertion schemes are introduced and a quantity called *relative insertion density* is defined for the description of these schemes. The relative insertion density d_i is related to the geometric span between actual knots p_i and p_{i+1} . The higher the value of the relative insertion density, the more the number of the virtual knots are inserted into the corresponding span. The number of virtual knots inserted into span between actual knots p_i and p_{i+1} is chosen to be

$$n_i = \left\lfloor \frac{d_i}{\sum d_i} N_r \right\rfloor - 1. \tag{21}$$

where $\lfloor \cdot \rfloor$ is the largest integer smaller than the value bracketed. N_r is the desired total number of knots after insertion. Because of the truncation, the total knot number will be $K + \sum n_i$ which may not equal N_r .

The proposed two insertion schemes are

1. distance ratio insertion (or distance insertion),
2. arc length ratio insertion (or arc insertion).

The difference between these two schemes is the methods used to calculate the relative insertion density. In the *distance ratio insertion*, the distance of two

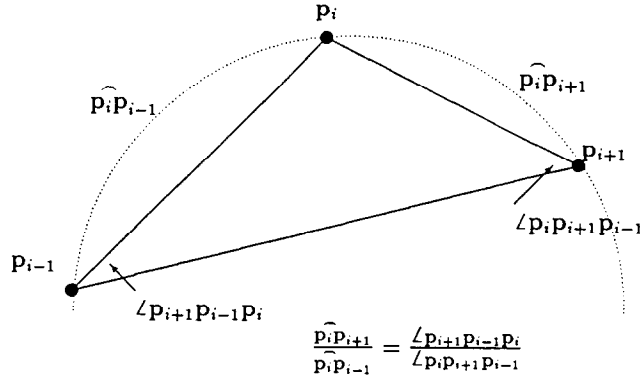


Fig. 3. The insertion density of arc length ratio insertion method are computed according to the ratio of arc length.

consecutive knots is calculated first. Then, the virtual knots are inserted into each span according to the ratio of these distances. The larger the span, the more the virtual knots are inserted. The relative insertion density is defined as

$$d_i = \frac{\|P_{i+1} - P_i\|}{\|P_i - P_{i-1}\|} d_{i-1} \tag{22}$$

for $i = 1, 2, \dots, K$ with $d_0 = 1$.

The *arc ratio insertion* uses the arc length ratio to determine the insertion density. Let P_{i-1} , P_i and P_{i+1} be consecutive actual knots. There exists a circle that will pass through all the three points. The insertion density is calculated according to arc lengths of the arcs $P_i\hat{P}_{i-1}$ and $P_i\hat{P}_{i+1}$ (see Fig. 3). That is

$$d_i = \frac{P_i\hat{P}_{i+1}}{P_i\hat{P}_{i-1}} d_{i-1} = \frac{\angle P_{i+1}P_{i-1}P_i}{\angle P_iP_{i+1}P_{i-1}} d_{i-1} \tag{23}$$

where

$$\angle P_iP_{i+1}P_{i-1} = \cos^{-1} \left(\frac{(P_i - P_{i+1}) \cdot (P_{i-1} - P_{i+1})}{\|P_i - P_{i+1}\| \|P_{i-1} - P_{i+1}\|} \right),$$

$$\angle P_{i+1}P_{i-1}P_i = \cos^{-1} \left(\frac{(P_{i+1} - P_{i-1}) \cdot (P_i - P_{i-1})}{\|P_{i+1} - P_{i-1}\| \|P_i - P_{i-1}\|} \right),$$

and $d_0 = 1$.

Note that when these three consecutive actual knots are collinear, the insertion density can be calculated using the distance ratio. In practice, if one of the angles $\angle P_iP_{i+1}P_{i-1}$ or $\angle P_{i+1}P_{i-1}P_i$ is less than a small threshold, the three knots are treated as “collinear”.

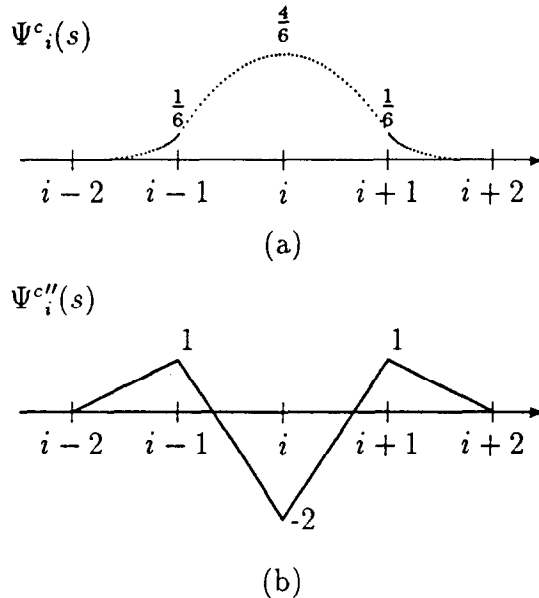


Fig. 4. (a) The basis function of the cubic B-spline. (b) The second derivatives of the cubic B-spline basis.

4. Illustrations and experiments

From the above discussion, one can see that a set of simultaneous equations should be solved for the curve fitting with the virtual knots. Since the system matrices are sparse and banded, to solve the equations is easy. With the conventional matrix methods, its computation complexity and the memory requirement is only in the order of $O(N)$. This simplicity is due to the local support of the bases. For illustration, both the cubic B-spline and linear B-spline bases are used in the following discussion.

4.1. Virtual knot technique in cubic B-spline space

In the cubic B-spline space, the basis functions are expressed as

$$\Psi_i^c(s) \triangleq \begin{cases} \frac{1}{6}[s - (i - 2)]^3, & i - 2 \leq s < i - 1, \\ \frac{1}{6}[4 - 6(s - i)^2 - 3(s - i)^3], & i - 1 \leq s < i, \\ \frac{1}{6}[4 - 6(s - i)^2 + 3(s - i)^3], & i \leq s < i + 1, \\ \frac{1}{6}[(i - 2) - s]^3, & i + 1 \leq s < i + 2, \\ 0 & \text{otherwise,} \end{cases} \quad (24)$$

as plotted in Fig. 4(a).

To apply the virtual knot technique, the bases ϕ_l in (12) are substituted by cubic B-spline bases. Then, the relation matrix and the couple matrix can be determined accordingly.

When a closed curve is considered, the periodic end condition:

$$v_i = v_{i \bmod N} \tag{25}$$

is used and the couple matrix is

$$U = \frac{1}{6} \begin{bmatrix} 4 & 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 4 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & & & \vdots \\ 0 & \dots & 0 & 1 & 4 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 & 4 & 1 \\ 1 & 0 & \dots & 0 & 0 & 1 & 4 \end{bmatrix}. \tag{26}$$

For an open curve, the following end conditions are used.

$$v_{-1} = 2v_0 - v_1 \quad \text{and} \quad v_N = 2v_{N-1} - v_{N-2}. \tag{27}$$

They are obtained by setting the second derivative at the ends to be zero. The resulted couple matrix is

$$U = \frac{1}{6} \begin{bmatrix} 6 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & & & \vdots \\ 0 & \dots & 0 & 1 & 4 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 & 4 & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 & 6 \end{bmatrix}. \tag{28}$$

To find the relation matrix, the second derivatives of these bases are used. They are

$$\Psi_i^{c''}(s) \triangleq \begin{cases} s - (i - 2), & i - 2 \leq s < i - 1, \\ -2 - 3(s - i), & i - 1 \leq s < i, \\ -2 + 3(s - i), & i \leq s < i + 1, \\ (i - 2) - s, & i + 1 \leq s < i + 2, \\ 0, & \text{otherwise,} \end{cases} \tag{29}$$

as plotted in Fig. 4(b). With this, the relation coefficients t_{ij} can be computed with (14). For the case of closed curve, the relation matrix is

$$T = \frac{1}{6} \begin{bmatrix} 16 & -9 & 0 & 1 & 0 & \dots & 0 & 1 & 0 & -9 \\ -9 & 16 & -9 & 0 & 1 & 0 & \dots & 0 & 1 & 0 \\ 0 & -9 & 16 & -9 & 0 & 1 & \dots & 0 & 0 & 1 \\ 1 & 0 & -9 & 16 & -9 & 0 & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & \ddots & & & & \ddots & & \vdots \\ 1 & 0 & 0 & \dots & 1 & 0 & -9 & 16 & -9 & 0 \\ 0 & 1 & 0 & \dots & 0 & 1 & 4 & -9 & 16 & -9 \\ -9 & 0 & 1 & 0 & \dots & 0 & 1 & 0 & -9 & 16 \end{bmatrix}. \tag{30}$$

For the case of an open curve, the end conditions are obtained by setting the second derivative beyond the domain of interest to be zero. It implies

$$\begin{aligned}
 x_{-3} &= 2x_{-2} - x_{-1} = 4x_0 - 3x_1, \\
 x_{-2} &= 2x_{-1} - x_0 = 3x_0 - 2x_1, \\
 x_{-1} &= 2x_0 - x_1, \\
 x_N &= 2x_{N-1} - x_{N-2}, \\
 x_{N+1} &= 2x_N - x_{N-1} = 3x_{N-1} - 2x_{N-2}, \\
 x_{N+2} &= 2x_{N+1} - x_N = 4x_{N-1} - 3x_{N-2}.
 \end{aligned}
 \tag{31}$$

With these, the relation matrix can be computed as follows.

$$T = \frac{1}{6} \begin{bmatrix} 2 & -3 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ -6 & 14 & -9 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 2 & -10 & 16 & -9 & 0 & 1 & \dots & 0 & 0 & 0 \\ 1 & 0 & -9 & 16 & -9 & 0 & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & & \ddots & & \ddots & & \vdots & \\ 0 & 0 & 0 & \dots & 1 & 0 & -9 & 16 & -10 & 2 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 & -9 & 14 & -6 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & -3 & 2 \end{bmatrix}.
 \tag{32}$$

With these matrices, the curve can be easily solved with (18).

4.2. Virtual knot technique in linear B-spline space

Likewise, the virtual knot technique can be applied to the linear B-spline space. These basis functions, called *tent functions*, are defined as

$$\Psi_i^1(s) \triangleq \begin{cases} s - i + 1, & i - 1 \leq s < i, \\ i + 1 - s, & i \leq s < i + 1, \\ 0 & \text{otherwise.} \end{cases}
 \tag{33}$$

as plotted in Fig. 5(a).

To apply the virtual knot technique to the linear B-spline space, the stabilizing functional must be generalized, since the derivatives of tent function are not well defined. To generalize the stabilizing functional, an alternative operator \mathcal{D}_s is used to replace the differential operator $\partial/\partial s$. The operator \mathcal{D}_s is defined as follows.

- When $x(s)$ is continuous but its derivative is not well-defined at s_i , define

$$\mathcal{D}_s x = \begin{cases} \frac{\partial x(s_i^+)}{\partial s} & \text{for } s = s_i, \\ \frac{\partial x(s)}{\partial s} & \text{otherwise.} \end{cases}
 \tag{34}$$

- When x has discontinuities, define

$$\mathcal{D}_s x = x(s + \frac{1}{2}) - x(s - \frac{1}{2}).
 \tag{35}$$

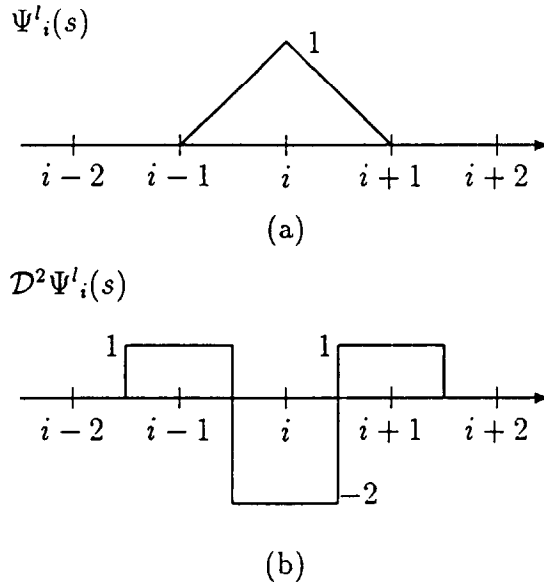


Fig. 5. (a) The basis function of the linear B-spline. (b) An approximated function of the "second derivative" of the linear B-spline basis.

With this, the functions $\mathcal{D}_s^2 \Psi_i^l(s)$ are expressed as

$$\mathcal{D}_s^2 \Psi_i^l(s) \triangleq \begin{cases} 1, & i - \frac{3}{2} \leq s < i - \frac{1}{2}, \\ -2, & i - \frac{1}{2} \leq s < i + \frac{1}{2}, \\ 1, & i + \frac{1}{2} \leq s < i + \frac{3}{2}, \\ 0, & \text{otherwise.} \end{cases} \quad (36)$$

as plotted in Fig. 5(b). The function $\mathcal{D}_s^2 x(s)$ is a step function. It can be verified that the amplitude of $\mathcal{D}_s^2 x(s)$ in interval $[i - \frac{1}{2}, i + \frac{1}{2})$ is the difference between the slopes of $x(s)$ in the two adjacent intervals $[i - 1, i)$ and $[i, i + 1)$. Therefore, the generalized stabilizing functional is still a meaningful measure of the variation of the linear B-spline curve.

The relation coefficient t_{ij} can be computed by substituting $\mathcal{D}_s^2 \Psi_i^l(s)$ for ϕ_i'' in (14). For the case of a closed curve, the relation matrix for the linear spline space is

$$T = \frac{1}{6} \begin{bmatrix} 6 & -4 & 1 & 0 & 0 & \dots & 0 & 1 & -4 \\ -4 & 6 & -4 & 1 & 0 & \dots & 0 & 0 & 1 \\ 1 & -4 & 6 & -4 & 1 & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & & \ddots & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & -4 & 6 & -4 & 1 \\ 1 & 0 & 0 & \dots & 0 & 1 & -4 & 6 & -4 \\ -4 & 1 & 0 & \dots & 0 & 0 & 1 & -4 & 6 \end{bmatrix}. \quad (37)$$

For the case of an open curve, the end conditions stated in (31) is applied again and the relation matrix is

$$T = \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -2 & 5 & -4 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & & \ddots & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -4 & 5 & -2 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & -2 & 1 \end{bmatrix}. \tag{38}$$

Moreover, because of the *cardinal property*

$$\Psi_i^1(j) \triangleq \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \tag{39}$$

the couple matrix U is an identity matrix. With these, the system equation using linear B-spline basis is

$$(T + \beta M)v = \beta Mx. \tag{40}$$

Due to the cardinal property of the linear B-spline basis, this equation is simpler than that of the cubic B-spline. In the same way, this technique can be applied with quadratic B-spline bases. The application in the quadratic B-spline space is shown in the appendix.

4.3. Experiments

To show the usefulness of the virtual knot technique in fairness control, several examples are shown in this section. In the first example, an “olive” shape (a closed curve) is fitted with ten actual knots. The curve fitted by the classical cubic B-spline method and its corresponding curvature plot are shown in Fig. 6. Wiggles near the two acute apexes are apparent. These wiggles correspond to the discontinuity in the curvature plot. When virtual knot technique is applied (with distance ratio insertion scheme and $N_r = 18$), the resultant curve is seen to be fairer. (see Fig. 6(a)). The corresponding curvature plot is also made smoother.

In the second example, a “half-circle” shape with ten actual knots is fitted. In this case, the arc ratio insertion scheme is used. The results are shown in Fig. 7. Again, unwanted wiggles can be seen in the curve generated by the classical cubic B-spline method. And a fairer curve is generated by inserting virtual knots.

In the third example, virtual knot technique is applied to fit a “cat-ear” shape with nine actual knots. Three curves are shown in Fig. 8. They are generated by the classical cubic B-spline fitting, virtual knot technique with linear B-spline as basis with 10 and 30 virtual knots, respectively. The virtual knots are inserted by the arc insertion method. The curves constructed by the virtual knot technique look fairer than The curve constructed by the classical cubic

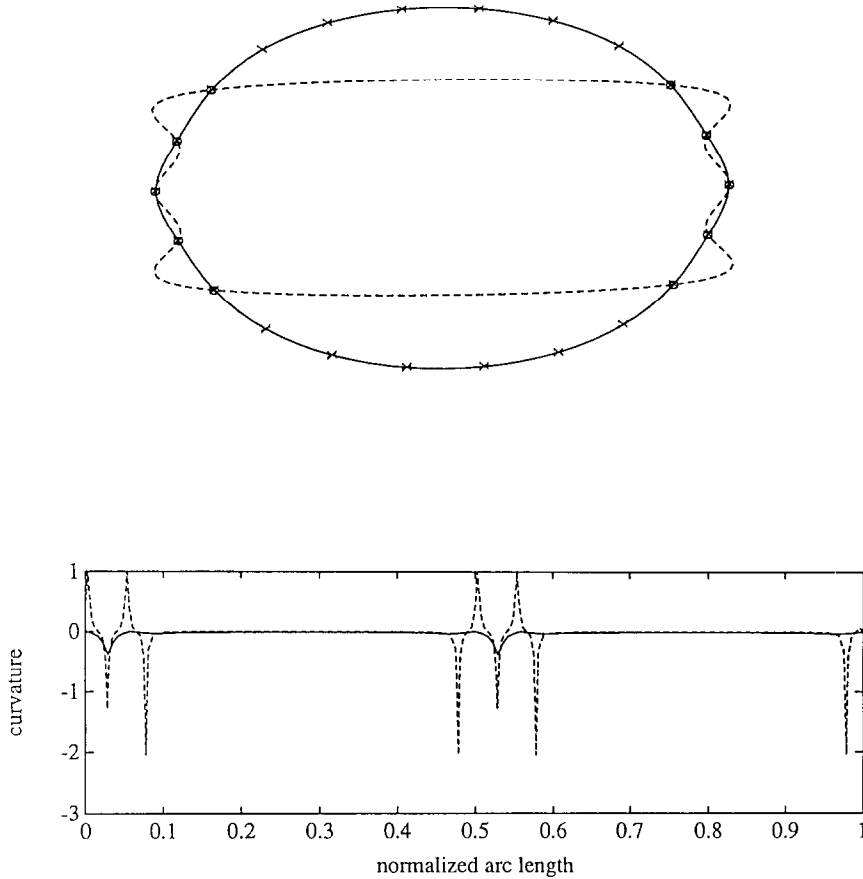


Fig. 6. An "olive" shape with ten actual knots (indicated by circles) is fitted. Circles indicate the actual knots and crosses indicate the new knots after virtual knot insertion. (*top*) The curve fitted by the classical cubic B-spline method is plotted as a dashed line and the curve fitted by the virtual knot technique using cubic B-spline basis with distance ratio insertion is plotted as a solid line. (*bottom*) The corresponding curvature plots of the curve fitted with the classical cubic B-spline method (dash) and the curve fitted with the virtual knot technique (solid).

B-spline method. The curve interpolated by the linear B-spline is a linear piece-wise polygon. This linear piece-wise polygon can approach the desired curve as long as the dimension of the interpolation space is increased. In classical linear interpolation, the dimension is usually limited to the number of actual knots. With the virtual knot technique, the dimension of the interpolation space can be increased arbitrarily such that the piece-wise polygon can approach a smooth curve.

In the last example, an open curve is fitted. The proposed method is compared with both the classical B-spline method and the Salkauskas-Foley method

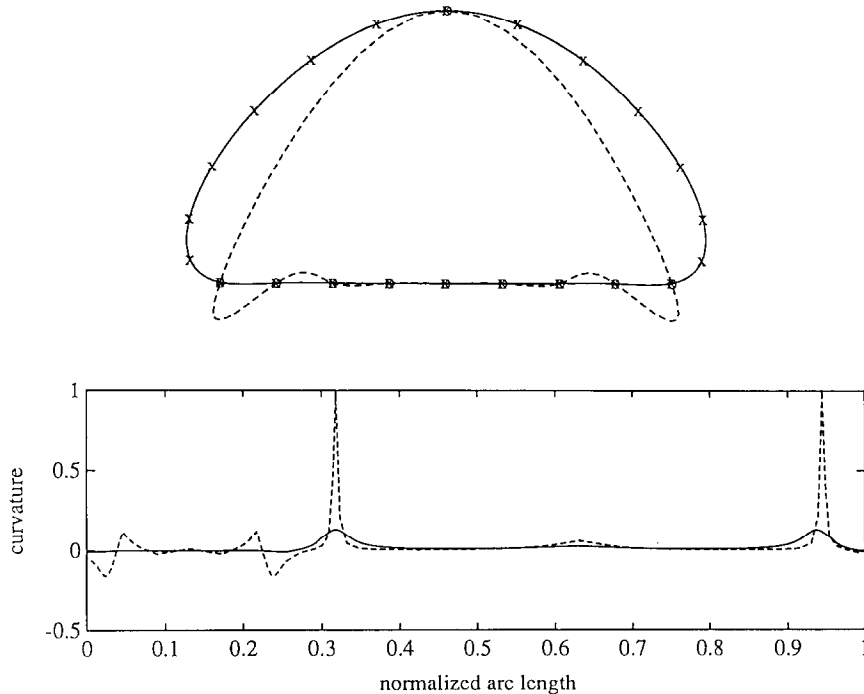


Fig. 7. A “half-circle” shape with ten actual knots is fitted. Circles indicate the actual knots and crosses indicate the new knots after virtual knot insertion. (top) The curve fitted by the classical cubic B-spline method is plotted as a dashed line and the curve fitted by the virtual knot technique using cubic B-spline with arc ratio insertion is plotted in solid line. (bottom) The corresponding curvature plots of both curves.

(Foley, 1987; Salkauskas, 1974). With fifteen data points, the resultant curves are shown in Fig. 9. The curve fitted by the classical cubic B-spline is shown in dashes. The curves fitted by the proposed method and the Salkauskas–Foley method are shown in solid and dash/dot, respectively. In this case, eight virtual knots are inserted and cubic B-spline bases are used in the proposed method. In the Salkauskas–Foley method, the weight control function $w(s)$ is chosen to be piecewise constant and is set to 5 for rapidly varying regions and 1 for the rest. It can be seen that the result generated by the our technique is compatible with that of the Salkauskas–Foley approach. However, the resulted curve representation of our method is simpler. No tension parameters is needed. When all weighting variables are solved, the representation of the resultant curve will be as simple as that of a classical spline curve. This simple representation is useful in many applications.

5. Conclusion

Fitting of rapidly varying data has long been a topic under active study. In this paper, we have studied how the insertion of virtual knots can be used to adjust the parametric spans of the actual knots such that they can be made

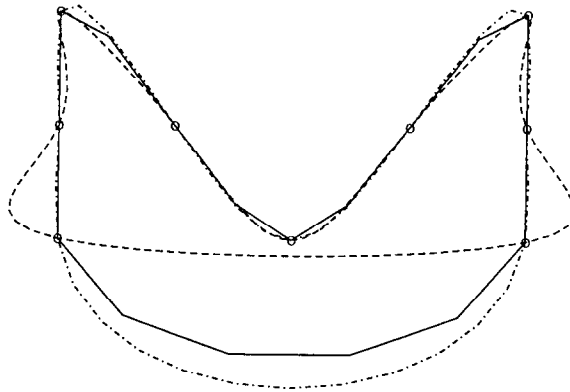


Fig. 8. A “cat-ear” shape with nine actual knots is fitted. The curve fitted by the classical cubic B-spline method is plotted as a dashed line. The curve fitted with the virtual knot technique using linear B-splines with arc ratio insertion is plotted in solid line (with 10 virtual knots) and dot line (with 30 virtual knots).

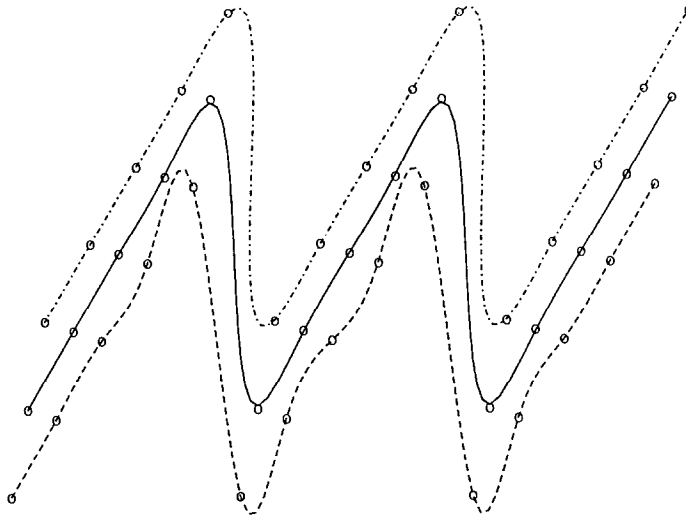


Fig. 9. Fitting of an open curve: three curves are generated by classical B-spline method (dash), the proposed method (solid) and the Salkauskas–Foley method (dash/dot), respectively.

consistent with the geometric spans to maintain the fairness even when the data is rapidly varying. Although various approaches have been proposed to attack the problem, most of them focus on the non-homogeneous interpolant. The proposed technique use non-uniform parameterization instead of non-homogeneous tension. This non-uniform parameterization is achieved by only inserting virtual knots. Although the parameterization along the actual knot sequence is non-uniform, the parameterization along the new knot sequence is kept uniform. By this, the simplicity of uniform interpolation is preserved. Therefore, the proposed method is a simple and convenient technique.

Our technique can be applied to different kinds of interpolation spaces. In this paper, both the cubic B-spline and the linear B-spline have been chosen for demonstration. For other interpolation spaces, the general system equation derived in Section 3 is also applicable. With the proposed technique, a fairer curve can be obtained with slight increase in computation complexity in comparison with the classical spline methods. However, the extra computational complexity with inserting virtual knots is kept minimum. Since, with finite support basis, the matrices T and U are kept banded. Therefore, the computation complexity is still kept linearly proportional to the number of knots. Moreover, the resultant curve representation is as simple as that of the classical uniform interpolation method.

Appendix A. Virtual knot technique with quadratic B-splines

In this appendix, the application of virtual knot technique to quadratic B-spline is discussed. This is an illustration for the usage of the proposed technique to even order B-splines. For higher order B-splines, similar results can be obtained. To interpolate a set of data points by quadratic B-splines, the resulted system equation can be unsolvable when the join-points of the piecewise quadratic curve are chosen as the data points. In (Pham, 1989), a smart modification is proposed to overcome the problem. Pham adjusted each join-point of the piecewise quadratic curve as the midpoint between every two data points to make the system equation become well-conditioned. In our approach, the join-point of the piecewise quadratic curve is chosen to be the midpoint between the new knot sequence. With this, the quadratic B-spline bases are expressed in the form

$$\Psi_i^q(s) \triangleq \begin{cases} \frac{1}{2} [(s-i) + \frac{3}{2}]^2, & i - \frac{3}{2} \leq s < i - \frac{1}{2}, \\ \frac{1}{2} [-2(s-i)^2 + \frac{3}{2}], & i - \frac{1}{2} \leq s < i + \frac{1}{2}, \\ \frac{1}{2} [-(s-i) + \frac{3}{2}]^2, & i + \frac{1}{2} \leq s < i + \frac{3}{2}, \\ 0, & \text{otherwise.} \end{cases}$$

and the relation matrix and couple matrix are calculated as

$$T = \frac{1}{6} \begin{bmatrix} 6 & -4 & 1 & 0 & 0 & \dots & 0 & 1 & -4 \\ -4 & 6 & -4 & 1 & 0 & \dots & 0 & 0 & 1 \\ 1 & -4 & 6 & -4 & 1 & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & & \ddots & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & -4 & 6 & -4 & 1 \\ 1 & 0 & 0 & \dots & 0 & 1 & -4 & 6 & -4 \\ -4 & 1 & 0 & \dots & 0 & 0 & 1 & -4 & 6 \end{bmatrix}$$

and

$$U = \frac{1}{8} \begin{bmatrix} 6 & 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 6 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 6 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & \dots & 0 & 1 & 6 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 & 6 & 1 \\ 1 & 0 & \dots & 0 & 0 & 1 & 6 \end{bmatrix}$$

for the case of closed curve.

For the open curve case, the corresponding relation matrix and couple matrix can be calculated with the above-mentioned end conditions.

$$T = \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -2 & 5 & -4 & 1 & 0 & \dots & 0 & 0 & 1 \\ 1 & -4 & 6 & -4 & 1 & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & & \ddots & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -4 & 5 & -2 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

and

$$U = \frac{1}{8} \begin{bmatrix} 8 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 6 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 6 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & \dots & 0 & 1 & 6 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 & 6 & 1 \\ 0 & 0 & \dots & 0 & 0 & 0 & 8 \end{bmatrix}.$$

Appendix B. The system equation for exact interpolation

In Section 3, it has been discussed that the system equation for the proposed technique will degenerate to the system equation of classical interpolation when all $m_k = 1$ (i.e., no virtual knots are added) and $\beta \rightarrow \infty$. In this appendix, another special case when $\beta \rightarrow \infty$ and not all $m_k = 1$ will be discussed. In this case, the fitting process is an exact interpolation with virtual knots. The problem will become a constrained optimization problem which minimize the stabilizing functional $S(f(s))$ subject to $f(s_k) = \hat{p}_k$ when $m_k = 1$. There are various convenient methods to solve a constrained optimization problem (Van de Panne, 1975).

Recall the system equation in (18), the matrix equation can be expanded into a set of equations:

$$\sum_{j=0}^{N-1} t_{ij}v_j + \beta \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} u_{ki}m_k u_{kj}v_j = \beta \sum_{j=0}^{N-1} u_{ji}m_j \hat{x}_j,$$

for $i = 0, 1, \dots, N - 1$. It can be written in the form

$$\sum_{j=0}^{N-1} t_{ij}v_j + \beta \sum_{j=0}^{N-1} a_{ij}v_j = \beta \sum_{j=0}^{N-1} b_{ij}\hat{x}_j$$

where

$$a_{ij} = \sum_{k=0}^{N-1} u_{ki}m_k u_{kj}, \quad b_{ij} = u_{ji}m_j.$$

If β approaches to infinity, these equations become

$$\begin{cases} \sum_{j=0}^{N-1} t_{ij}v_j = 0 & \text{if } a_{i0} = a_{i1} = \dots = a_{i,N-1} \\ & = b_{i0} = b_{i1} = \dots = b_{i,N-1} = 0, \\ \sum_{j=0}^{N-1} a_{ij} = \sum_{j=0}^{N-1} b_{ij}\hat{x}_j & \text{otherwise,} \end{cases}$$

for $i = 0, 1, \dots, N - 1$. Solving this set of equation, an exact interpolating curve with virtual knots can be obtained. Theoretically, exact interpolation can only achieved by setting β to infinity. However, in practice, if β is chosen to be large enough the resulted curve will be almost identical to that obtained by exact interpolation. Therefore, the system equation in (18) can also be applied for this case by choosing a large β (e.g., $\beta = 100$).

Acknowledgment

The authors wish to thank the reviewers for their most helpful comments.

References

- Ballard, D.H. and Brown, C.M. (1982), *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ.
- Barsky, B.A. (1984), Exponential and polynomial methods for applying tension to an interpolating spline curve, *Comput. Vision Graphics Image Process.* 27, 1-18.
- Boehm, W. (1980), Inserting new knots into B-spline curves, *Comput. Aided Design* 12, 199-201.
- Boehm, W. (1982), On cubics: a survey, *Comput. Graphics Image Process.* 19, 201-226.
- Carlson, R.E. and Fritsch, F.N. (1985), Monotone piecewise bicubic interpolation, *SIAM J. Numer. Anal.* 22, 386-400.
- Cohen, E. (1987), A new local basis for designing with tensioned splines, *ACM Trans. Graphics* 6, 81-122.

- Cohen, E. and Lyche, T. (1980), Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics, *Comput. Graphics Image Process.* 14, 87-111.
- de Boor, C. (1978), *A Practical Guide to Splines*, Springer, New York.
- DeRose, T.D. and Barsky, B.A. (1988), Geometric continuity, shape parameters, and geometric constructions for Catmull-Rom splines, *ACM Trans. Graphics* 7, 1-41.
- Farin, G., Rein, G. Sapidis, N. and Worsey, A.J. (1987), Fairing cubic B-spline curves, *Computer Aided Geometric Design* 4, 91-103.
- Foley, T.A. (1987), Weighted bicubic spline interpolation to rapidly varying data, *ACM Trans. Graphics* 6, 1-18.
- Foley, T.A. and Ely, H.S. (1989), Surface interpolation with tension controls using cardinal bases, *Comput. Aided Geometric Design* 6, 97-109.
- Forrest, A.R. (1980), The twisted cubic curve: a computer-aided geometric design approach, *Computer Aided Design* 12, 165-172.
- Groetsch, C.W. (1984), *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*, Pitman, Boston, MA.
- Gunther, O. and Wong, E. (1990), The arc tree: an approximation scheme to represent arbitrary curved shapes, *Comput. Vision Graphics Image Process.* 51, 313-337.
- Hartley, P.J. and Judd, C.J. (1980), Parametrization and shape of B-spline curves for computer aided design, *Computer Aided Design* 12, 235-238.
- Hou, H.S. and Andrews, H.C. (1978), Cubic splines for image interpolation and digital filtering, *IEEE Trans. Acoust., Speech, Signal Process.* 26, 508-517.
- Keys, R.G. (1981), Cubic convolution interpolation for digital image processing, *IEEE Trans. Acoust., Speech, Signal Process.* 29, 1153-1160.
- Kjellander, J.A. (1983), Smoothing of cubic parametric splines, *Computer Aided Design* 15, 288-294.
- Lancaster, P. and Salkauskas, K. (1986), *Curve and Surface Fitting*, Academic Press, New York.
- Lane, J.M. and Riesenfeld, R.F. (1980), A theoretical development for the computer generation and display of piecewise polynomial surfaces, *IEEE Trans. Pattern Anal. Mach. Intell.* 2, 35-46.
- Langridge, D.J. (1982), Curve encoding and the detection of discontinuities, *Comput. Graphics Image Process.* 20, 58-71.
- Lee, E.T.Y. (1990), Energy, fairness, and counterexample, *Computer Aided Design* 22, 37-40.
- Lee, D. and Pavlidis, D. (1988), One-dimensional regularization with discontinuities, *IEEE Trans. Pattern Anal. Mach. Intell.* 10, 822-829.
- Medioni, G. and Yasumoto, Y. (1987), Corner detection and curve representation using cubic B-splines, *Comput. Vision Graphics Image Process.* 38, 267-278.
- Namane, A. and Sid-Ahmed, M.A. (1990), Character scaling by contour method, *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 600-606.
- Nielson, G.M. (1974), Some piecewise polynomial alternatives to splines under tension, in: Barnhill, R.E. and Riesenfeld, R.F., eds., *Computer Aided Geometric Design*, Academic Press, New York, 209-235.
- Park, S.K. and Schowengerdt, R.A. (1983), Image reconstruction by parametric cubic convolution, *Comput. Vision Graphics Image Process.* 23, 258-272.
- Pham, B. (1989), Quadratic B-splines for automatic curve and surface fitting, *Comput. & Graphics* 13, 471-475.
- Prenter, P.M. and Westwater, E.R. (1986), Three adaptive discrete least squares cubic spline procedures for the compression of data, *Comput. Vision Graphics Image Process.* 33, 327-345.
- Salkauskas, K. (1974), C^1 splines for interpolation of rapidly varying data, *Rocky Mountain J. Math.* 14, 239-250.
- Schaback, R. (1989), Interpolation with piecewise quadratic visually C^2 Bézier polynomials, *ACM Trans. Graphics* 6, 219-233.
- Terzopoulos, D. (1986), Regularization of inverse visual problems involving discontinuities, *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 413-424.
- Tiller, W. (1983), Rational B-spline for curve and surface representation, *IEEE Comput. Graphics Appl.*, 61-69.
- Van de Panne, C. (1975), *Methods for Linear and Quadratic Programming*, North-Holland, Oxford.

- Wu, S.C., Abel, J.F., and Greenberg, D.P. (1977), An interactive computer graphics approach to surface representation, *Comm. ACM* 20, 703–712.
- Yang, M.C.K., Kim, C.K., Cheng, K.Y., Yang, C.C. and Liu, S.S. (1986), Automatic curve fitting with quadratic B-spline functions and its application to computer-assisted animation, *Comput. Vision Graphics Image Process.* 33, 346–363.