Available online at www.sciencedirect.com
ScienceDirect
Computer-Aided Design 39 (2007) 439–451
COMPUTER-AIDED DESIGN
www.elsevier.com/locate/cad

B-spline curve fitting based on adaptive curve refinement using dominant points

Hyungjun Park[a,*], Joo-Haeng Lee[b]

[a] Department of Industrial Engineering, Chosun University, 375 Seosuk-Dong, Dong-Gu, Gwangju 501-759, South Korea
[b] Digital Actor Team, Electronics and Telecommunications Research Institute, 161 Gajeong-Dong, Yuseong-Gu, Daejeon 305-350, South Korea

Received 29 July 2006; accepted 18 December 2006

Abstract

In this paper, we present a new approach of B-spline curve fitting to a set of ordered points, which is motivated by an insight that properly selected points called *dominant* points can play an important role in producing better curve approximation. The proposed approach takes four main steps: parameterization, dominant point selection, knot placement, and least-squares minimization. The approach is substantially different from the conventional approaches in knot placement and dominant point selection. In the knot placement, the knots are determined by averaging the parameter values of the dominant points, which basically transforms B-spline curve fitting into the problem of dominant point selection. We describe the properties of the knot placement including the property of local modification useful for adaptive curve refinement. We also present an algorithm for dominant point selection based on the adaptive refinement paradigm. The approach adaptively refines a B-spline curve by selecting fewer dominant points at flat regions but more at complex regions. For the same number of control points, the proposed approach can generate a B-spline curve with less deviation than the conventional approaches. When adopted in error-bounded curve approximation, it can generate a B-spline curve with far fewer control points while satisfying the desired shape fidelity. Some experimental results demonstrate its usefulness and quality.

Keywords: B-spline curve fitting; Knot placement; Dominant point selection; Adaptive curve refinement; Local modification

1. Introduction

B-spline curve approximation is one of the classic problems that have been established in the field of computer aided geometric design [1,2]. Nonetheless B-spline curve approximation is still an essential operation in many applications. For example, large amounts of data created in various processes of engineering design must be approximated by smooth B-spline curves.

In B-spline curve approximation to a sequence of points, a B-spline curve is sought that approximates the points in a way that the curve satisfies a criterion of approximation quality. In practical applications, a tolerance is specified in order to obtain a B-spline curve satisfying that the distance between the curve and the given points should be less than the tolerance. The resulting curve is called error-bounded. The important issue in B-spline curve approximation is to reduce the number of knots and the number of control points while keeping the desired accuracy. As it is usually not known in advance how many control points are required to obtain the accuracy, the problem generally requires an iterative process that adjusts the number of control points, the parameter values, and the knots to maintain an error bound [1–5]. Least-squares curve fitting is often used as one step in this iterative process.

In this paper, we present a new approach for B-spline curve fitting to a set of ordered points. This work stems from an insight that some points properly selected from the given points play an important role in yielding better approximations. These points are called *dominant* points. Fig. 1 is an example showing a comparison between the conventional approach and the proposed approach. Fig. 1(a) shows 101 input points sampled from the side profile of the human facial shape. Their parameter values are drawn below the point set. Fig. 1(b) shows a cubic B-spline curve obtained by the traditional approach with 15 control points. Its knot vector is drawn below the

* Corresponding author. Tel.: +82 62 230 7039; fax: +82 62 230 7128.
E-mail address: hzpark@chosun.ac.kr (H. Park).

0010-4485/$ - see front matter
doi:10.1016/j.cad.2006.12.006

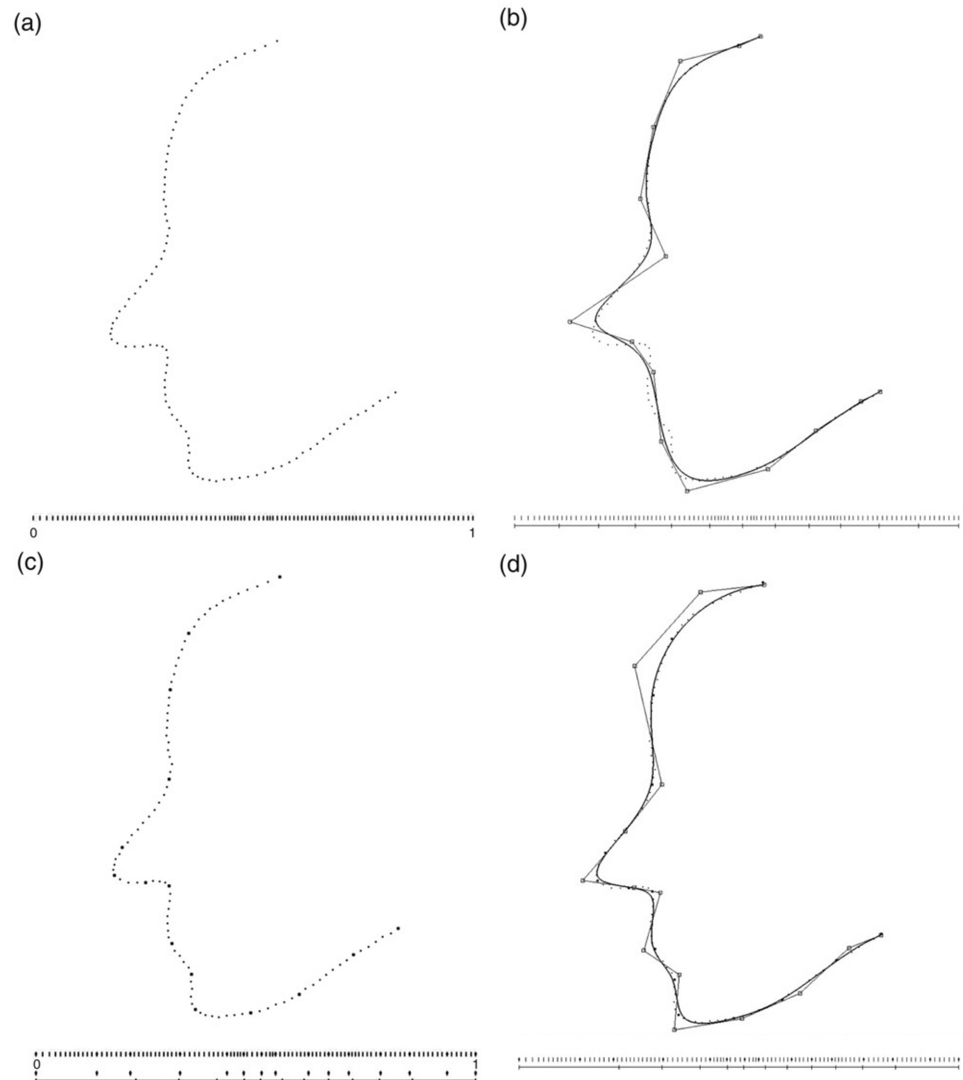

Fig. 1. Traditional approach vs. proposed approach for B-spline curve fitting: (a) Input points; (b) cubic B-spline curve obtained by the traditional approach with 15 control points; (c) 15 dominant points chosen from the input points; (d) cubic B-spline curve obtained by the proposed approach with 15 control points.

curve. Note that significant gaps between the curve and the point set are found around the nose and the mouth. Fig. 1(c) shows 15 dominant points chosen from the point set. Big solid dots denote dominant points. The parameter values of the dominant points and the knots are drawn in the lowest part of the figure. Fig. 1(d) shows a cubic B-spline curve obtained by the proposed approach with 15 control points. Its knot vector is drawn below the curve. Note that the gaps between the curve and the point set are reduced significantly. Compare the knot vectors obtained by the two approaches.

The proposed approach for B-spline curve fitting takes four main steps: parameterization, dominant point selection, knot placement, and least-squares minimization. The approach is substantially different from the conventional approaches in knot placement and dominant point selection. In the knot placement, the knots are determined by averaging the parameter values of the dominant points. Thus the approach basically transforms B-spline curve fitting into the problem of selecting dominant points among the given points. We present the properties of the knot placement including the property of local modification useful for adaptive refinement of a B-spline curve. We also present an algorithm for proper selection of dominant points based on the adaptive refinement paradigm.

With the knot placement and the dominant point selection presented, the proposed approach can realize the concept of adaptive curve refinement that fewer curve segments are generated at flat regions but more at complex regions. Consequently, for the same number of control points, the approach can generate a B-spline curve with less deviation from the point set than the conventional approaches presented in previous works [1–3,6]. The proposed approach can be adopted as a key ingredient for the iterative process of error-bounded B-spline curve approximation, playing an important role in generating a B-spline curve with far fewer control points while satisfying the desired shape fidelity.

The rest of the paper is organized as follows: In Section 2, related works on B-spline curve fitting are briefly described. In Section 3, details on the proposed approach for B-spline curve fitting are described. In Section 4, experimental results are given to demonstrate the usefulness and quality of the approach. In Section 5, the proposed approach is further discussed. Section 6 closes the paper.

2. Related works

We assume that the reader is familiar with the concepts of B-spline curves [1,2]. A parametric B-spline curve $\mathbf{C}(t)$ of order (degree + 1) p is defined by

$$\mathbf{C}(t) = \sum_{j=0}^{n} N_{j,p}(t)\mathbf{b}_j \quad (t_{p-1} \leq t \leq t_{n+1}) \tag{1}$$

where $\mathbf{b}_j$ $(j = 0, \dots, n)$ are control points and $N_{j,p}(t)$ are the normalized B-spline functions of order p defined on a knot vector $\mathbf{T} = \{t_0, t_1, \dots, t_{n+p-1}, t_{n+p}\}$. The knot vector $\mathbf{T}$ consists of non-decreasing real-valued knots and mostly its first and last knots are repeated with multiplicity equal to order p as follows: $t_0 = \dots = t_{p-1} = 0, t_{n+1} = \dots = t_{n+p} = 1$.

In this paper, we consider the following problem: Given a set of ordered points $\mathbf{p}_i$ $(i = 0, \dots, m)$ as input data, compute a B-spline curve to fit the points. We assume that, when the given points are not ordered, the ordering of the points can be determined with ease. For the case that it is very difficult or impossible to order data points, we can refer to some notable works on B-spline curve fitting to unorganized points [7,8].

It is possible to construct an optimal B-spline curve from the given points by solving a nonlinear optimization problem with the number of control points, the control points, the parameter values, and the knots as unknowns. We can refer to some notable works based on the optimization approach [4,9,10], but there still remain many challenges in this approach.

As a widely used tool for computing a B-spline curve $\mathbf{C}(t)$ from the ordered points $\mathbf{p}_i$, least-squares B-spline curve fitting takes three steps: parameterization, knot placement, and least-squares minimization. In parameterization, we select the parameter values $\bar{t}_i$ of the points $\mathbf{p}_i$. In knot placement, we determine a knot vector $\mathbf{T}$ after specifying the order p and the number of points n. In least-squares minimization, we determine the control points $\mathbf{b}_j$ $(j = 0, \dots, n)$ of a B-spline curve $\mathbf{C}(t)$ by minimizing the least-squares error defined as

$$E(\mathbf{b}_0, \dots, \mathbf{b}_n) = \sum_{i=0}^{m} \|\mathbf{C}(\bar{t}_i) - \mathbf{p}_i\|^2. \tag{2}$$

Generally, n is not greater than m. When linear constraints such as tangents, normals, or higher derivatives are given, they can be incorporated into Eq. (2) using Lagrange multipliers [2]. The least-squares minimization problem is transformed into that of solving a linear system. The resulting curve is a piecewise B-spline curve of order (degree + 1) p with C^{p-2} continuity at knots.

Selection of the parameter values $\bar{t}_i$ and the knot vector $\mathbf{T}$ is critical to the quality of the fitted curve $\mathbf{C}(t)$. The approach widely adopted takes two steps: determination of the parameter values and placement of the knots. Mostly the parameter values are computed using the chord length or centripetal methods [1,2,5]. A knot vector $\mathbf{T}$ must be selected according to the parameter values $\bar{t}_i$, the number n of control points, and the number m of the given points. Mostly the knots are determined to reflect the distribution of the parameter values and to guarantee that every knot span contains at least one parameter value. The interior knots t_i of the knot vector $\mathbf{T}$ can be spaced as follows [2]:

$$t_{p+i-1} = \begin{cases} \dfrac{1}{p-1} \displaystyle\sum_{j=i}^{i+p-2} \bar{t}_j \Leftarrow m = n \\ (1-u)\bar{t}_{k-1} + u\bar{t}_k \Leftarrow m > n \end{cases} \quad (i = 1, \dots, n-p+1) \tag{3}$$

where $k = \text{int}(i \times d), d = (m+1)/(n-p+2)$, and $u = i \times d - k$. This is called the averaging technique (AVG) for $m = n$, and the knot placement technique (KTP) for $m > n$. Several methods [1,3,11,12] for parameterization or parameter correction can be adopted in order to further reduce the fitting errors and improve the quality of the resulting curve.

The techniques (AVG and KTP) give a stable system of equations with a diagonally dominant coefficient matrix that has semi-bandwidth less than p [1,2,13]. Nonetheless, not all stable equations yield acceptable curves [2,6]. When m is nearly greater than n ($|m - n|$ is small), it is sensitive to the distribution or parameter values of the points $\mathbf{p}_i$ and liable to generate undesirable results where curves can be fairly wiggly, especially toward endpoints. See Fig. 2(a). The stability of B-spline curve fitting can be more or less improved by combining energy terms with the least-squares error criterion [14–16], but the stability problem still exists in B-spline curve fitting with the knots computed by KTP. To avoid this problem, Piegl and Tiller [6] recently suggested another knot placement technique (NKTP) that is a generalization of the averaging technique used for interpolation. See Fig. 2(b). The knot placement using dominant points does not suffer from this problem and has interesting properties, which will be described in the following section. See Fig. 2(c).

Knot removal can be used for B-spline curve approximation by starting with a B-spline curve that interpolates all the points and by progressively removing the knots with less significance from the curve until the number of control points is reached [5,17,18]. This knot removal approach (KRM) is straightforward and can be designed to be error-bounded, but often suffers from noisy data. In recent works [19,20], the shape information of the given point set was used in knot selection. Razdan [19] dealt with knot selection for B-spline curve approximation of a given curve and suggested a simple but insightful algorithm that selects node points on the curve and interpolates them with a cubic B-spline curve. He adopted cubic B-spline curve interpolation with end-conditions and made the knots identical to the parameter values of the selected points. For reasonable selection of node points, he considered shape information such as arc length and curvature distribution of the input data. However, his algorithm is sensitive to noisy data because of the interpolation. Moreover, its ability to deal with error-bounded approximation is not addressed. Li et al. [20] also

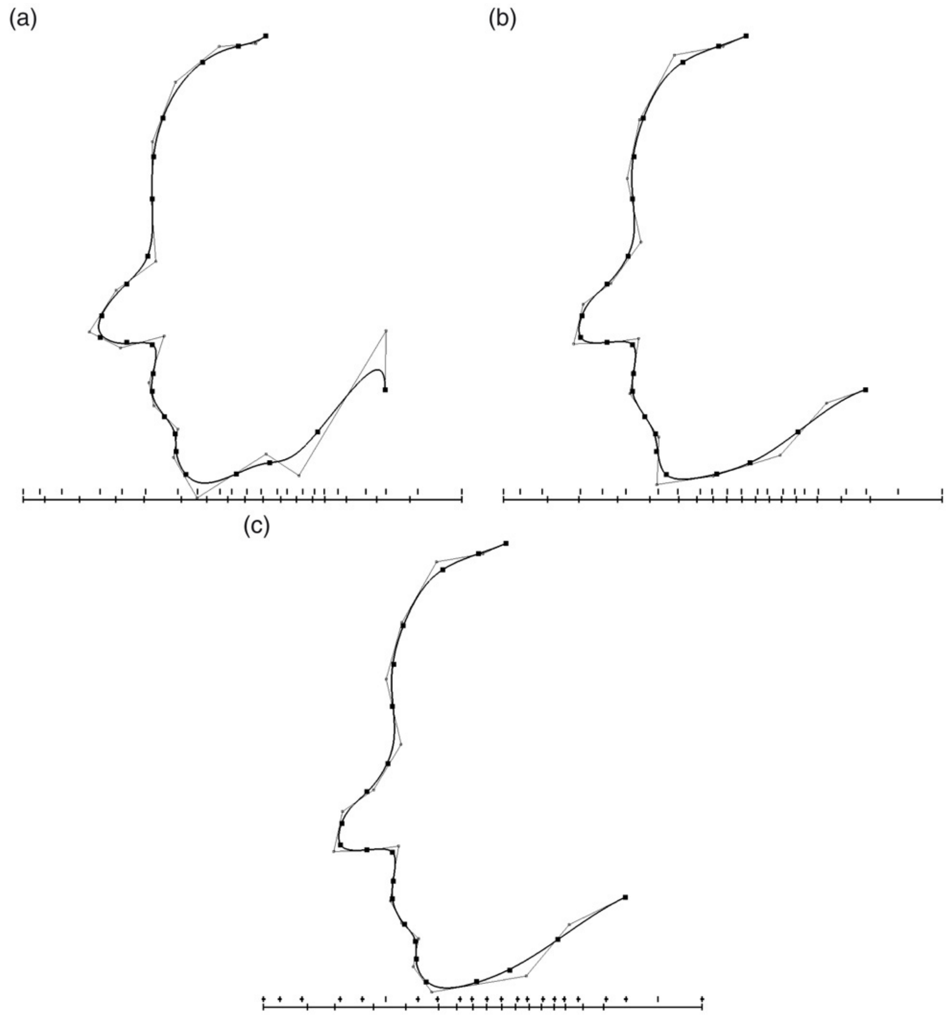

Fig. 2. Cubic B-spline curve fitting for $m = 21$ points and $n = 19$ control points: (a) Curve obtained by KTP; (b) curve obtained by NKTP; (c) curve obtained by the proposed approach.

proposed a similar algorithm for knot selection for B-spline curve approximation. They adopted least-squares curve fitting where the knots are identical to the parameter values of the selected points. This knot placement possibly suffers from the stability problem when m is nearly equal to n. Their algorithm was not designed to be error-bounded. Moreover, their heuristic rule for knot placement tends to require more knots than is expected. The approach proposed in this paper is similar to those of [19,20] in that shape information is used in knot placement, but its strength lies in stability, robustness to noise, and error-boundedness.

3. Proposed approach

Based on the insight illustrated in Fig. 1, we propose a novel approach for B-spline curve fitting, which takes the following steps:

(1) Compute the parameter values of all the given points.
(2) Select a specified number of dominant points from the points.
(3) Determine knots by using the parameter values of the selected dominant points.
(4) Using Eq. (2), perform the least-squares minimization in order to obtain a B-spline curve that approximates all the points.

Throughout the paper, the proposed approach is denoted as DOM. In this work, we adopt the chord length or centripetal methods [1,2] to compute the parameter values. The number of control points of a resulting B-spline curve is equal to that of the dominant points. The proposed approach takes well-known steps such as parameterization and least-squares minimization. However, its uniqueness lies in two steps: dominant point selection and knot placement. For convenience, we first describe how to determine knots. Then, we describe how to select dominant points.

3.1. Determination of knots

For the time being, assume that dominant points $\mathbf{d}_j$ ($j = 0, \ldots, n$) have been selected among the given points $\mathbf{p}_k$ ($k = 0, \ldots, m$). Without loss of generality, n is not greater than m ($n \leq m$). With the dominant points $\mathbf{d}_j$, interior knots t_i can be computed as follows:

$$t_{p+i-1} = \frac{1}{p-1} \sum_{j=i}^{i+p-2} \bar{t}_{f(j)} \quad (i = 1, \ldots, n-p+1) \tag{4}$$

where $\bar{t}_k$ are the parameter values of the points $\mathbf{p}_k$ and $f(j)$ is a simple function that returns the index of the point $\mathbf{p}_k$ corresponding to a dominant point $\mathbf{d}_j$.

3.1.1. Properties of knot placement using dominant points

Note that the above knots are computed by averaging the parameter values of the dominant points, so the resulting knot vector coincides with the one for B-spline curve interpolation to the dominant points. Knot placement using dominant points has the following important properties:

- It yields a nonsingular stable system matrix during the least squares minimization.
- Even if m is nearly equal to n, it yields curves of good quality.
- If $m = n$, the knot vector coincides with the one for interpolation.
- A dominant point newly included changes the knot vector only in at most $(p-1)$ knots near its parameter value.

The knot placement using dominant points is basically the same as the averaging technique AVG in [1,2]. Thus, it gives a nonsingular stable system matrix according to the Schoenberg–Whitney theorem [1,2,9]. Moreover, knot placement using dominant points does not cause undesirable results when $|m-n|$ is small. Fig. 2 shows cubic B-spline curve fitting for $m = 21$ and $n = 19$. Fig. 2(a) shows a B-spline curve obtained by KTP. Note that the curve is fairly wiggly toward the endpoints. Fig. 2(b) shows a B-spline curve obtained by NKTP recently suggested by Piegl and Tiller [6]. Fig. 2(c) shows a B-spline curve obtained by DOM, where the 6th and the 21st points are not used as dominant points. Control polylines are overlayed on the curves. Parameter values and knot vectors are drawn at the bottom of the figures. The sign '+' is used to denote dominant points. Note that the curves in Fig. 2(b) and (c) are of good quality. It is obvious that, when $m = n$, knot placement based on dominant points yields a knot vector coinciding with the one for interpolation.

The fourth property, called the property of *local modification*, is very useful for adaptive refinement of a B-spline curve, which adaptively selects new dominant points near the region to be refined while keeping the other regions. See Fig. 4 for two cases of cubic B-spline curve fitting by knot placement using dominant points. As the knot placement techniques KTP and NKTP partition the parameter values according to the number of control points, they change all interior knots of the knot vector even though we increase the number of control points by one. Consequently, KTP and NKPT have the property of *global modification*. The knot placement NKTP and the knot placement using dominant points have the first three properties in common, but they are different in the fourth property.

Writing Eq. (4) with $(n+1)$ dominant points $\mathbf{d}_j^{\text{cur}}$ $(j = 0, \ldots, n)$, we have current interior knots t_i^{cur} given as follows:

$$t_{p+i-1}^{\text{cur}} = \frac{1}{p-1} \sum_{j=i}^{i+p-2} \bar{t}_j^{\text{cur}} \quad (i = 1, \ldots, n-p+1) \tag{5}$$

where $\bar{t}_j^{\text{cur}}$ are the parameter values of the dominant points $\mathbf{d}_j^{\text{cur}}$. Let $\mathbf{d}_w$ be a dominant point newly inserted and let $\bar{t}_w$ be its parameter value such that $\bar{t}_k^{\text{cur}} < \bar{t}_w < \bar{t}_{k+1}^{\text{cur}}$. Now we have $(n+2)$ dominant points $\mathbf{d}_j^{\text{new}}$ given as follows:

$$\mathbf{d}_j^{\text{new}} = \begin{cases} \mathbf{d}_j^{\text{cur}} & \text{for } 0 \le j \le k \\ \mathbf{d}_w & \text{for } j = k+1 \\ \mathbf{d}_{j-1}^{\text{cur}} & \text{for } k+2 \le j \le n+1. \end{cases} \tag{6}$$

Obviously, the parameter values $\bar{t}_j^{\text{new}}$ of the dominant points $\mathbf{d}_j^{\text{new}}$ are given as follows:

$$\bar{t}_j^{\text{new}} = \begin{cases} \bar{t}_j^{\text{cur}} & \text{for } 0 \le j \le k \\ \bar{t}_w & \text{for } j = k+1 \\ \bar{t}_{j-1}^{\text{cur}} & \text{for } k+2 \le j \le n+1. \end{cases} \tag{7}$$

Then, new interior knots t_i^{new} are defined as follows:

$$t_{p+i-1}^{\text{new}} = \frac{1}{p-1} \sum_{j=i}^{i+p-2} \bar{t}_j^{\text{new}} \quad (i = 1, \ldots, n-p+2). \tag{8}$$

By Eq. (7), the interior knots t_{p+i-1}^{new} $(i = 1, \ldots, k-p+2)$ are expressed as follows:

$$t_{p+i-1}^{\text{new}} = \frac{1}{p-1} \sum_{j=i}^{i+p-2} \bar{t}_j^{\text{new}} = \frac{1}{p-1} \sum_{j=i}^{i+p-2} \bar{t}_j^{\text{cur}} = t_{p+i-1}^{\text{cur}}. \tag{9}$$

Similarly, the interior knots t_{p+i-1}^{new} $(i = k+2, \ldots, n-p+2)$ are expressed as follows:

$$t_{p+i-1}^{\text{new}} = \frac{1}{p-1} \sum_{j=i}^{i+p-2} \bar{t}_j^{\text{new}} = \frac{1}{p-1} \sum_{j=i}^{i+p-2} \bar{t}_{j-1}^{\text{cur}} = t_{p+i-2}^{\text{cur}}. \tag{10}$$

Finally, we have the new interior knots t_i^{new} that can be expressed as follows:

$$t_{p+i-1}^{\text{new}} = \begin{cases} t_{p+i-1}^{\text{cur}} & \text{for } i = 1, \ldots, k-p+2 \\ \dfrac{1}{p-1} \displaystyle\sum_{j=i}^{i+p-2} \bar{t}_j^{\text{new}} & \text{for } i = k-p+3, \ldots, k+1 \\ t_{p+i-2}^{\text{cur}} & \text{for } i = k+2, \ldots, n-p+2. \end{cases} \tag{11}$$

From Eqs. (5) to (11), we have found that a dominant point newly inserted yields a knot vector in which at most $(p-1)$ knots are modified and the others are unchanged.

The knot placement KTP selects the knots in a manner such that each knot span contains almost the same number of parameter values. The knot placement NKTP shows a similar trend except at the first and the last knot spans. Combined with a proper selection of dominant points, the knot placement using dominant points adaptively selects the knots such that fewer knots are selected at flat regions but more at complex regions. Therefore, the overall shape of the given point set can be reflected upon a knot vector if dominant points are properly selected based on the shape information.

3.2. Selection of dominant points

In this paper, the problem of B-spline curve fitting is basically transformed into the one of dominant point selection, where dominant points $\mathbf{d}_j$ $(j = 0, \ldots, n)$ should be selected among the points $\mathbf{p}_i$ $(i = 0, \ldots, m)$. Once dominant points are chosen, a B-spline curve can be generated with ease by performing the knot placement in Eq. (4) and the least-squares curve fitting to all the points. Eventually, the quality of a resulting curve depends on how to select dominant points. The dominants points can be determined by combinatorial optimization [21]. However, this approach is too time-consuming to be used for practical applications. We can devise an algorithm based on the adaptive refinement paradigm, realizing that fewer dominant points are chosen at flat regions but more at complex regions. Its main steps are given as follows:

(1) Set $n_d = -1$ where n_d denotes the highest index of current dominant points.
(2) Select *seed* points from the point set, and store them in a list SEED in decreasing order of *significance*. While SEED is not empty AND $n_d < n$, do the following steps: pop a point from SEED, make it a *new* dominant point, and $n_d \leftarrow n_d + 1$.
(3) While $n_d < n$, select dominant points $\mathbf{d}_j$ adaptively as follows:
    (3.1) Find the region defined between two points $\mathbf{d}_k = \mathbf{p}_s$ and $\mathbf{d}_{k+1} = \mathbf{p}_e$ $(|e-s| > 1)$ with the largest deviation.
    (3.2) Choose a point $\mathbf{p}_w$ $(s < w < e)$ as a *new* dominant point, and $n_d \leftarrow n_d + 1$.

3.2.1. Selection of seed points

Various kinds of feature points characterizing the shape of the given point set can be used as seed points. These include local curvature maximum (LCM) points, inflection points and sharp bends in curvature [19]. Deciding which kind of feature points to use as seed points depends on the easiness and robustness of their determination. Feature points are determined mostly based on the curvature information. When the given points are noisy, it not easy to estimate their curvatures accurately and reliably, which disturbs the detection of some feature points. Compared to the others, LCM points can be determined more easily and robustly. In this work, we consider two kinds of seed points: two end points and LCM points. From the curvatures k_i estimated at the given points, we can detect LCM points using the following rules: if $k_i > k_{i-1}$ and $k_i > k_{i+1}$, the point $\mathbf{p}_i$ is considered as an LCM point. These points correspond to peaks in the curvature plot. After searching all LCM points, we exclude the points with curvatures smaller than the lower curvature bound k_{low}. In this work, we set $k_{\text{low}} = k_{\text{avg}}/4$ where k_{avg} is the average of all the discrete curvatures. Two end points are the most significant. For LCM points, the greater their curvatures are, the more significant they are.

Curvatures of the given points provide useful information to describe the shape of the points. We can estimate the curvatures either by local methods [22,23] or by curve fitting methods [1,2,5,16]. When the points are noisy, curvatures computed with local methods contain more severe noise. If the input points are free of noise, we take the former approach. Otherwise, we take the latter approach to obtain a cubic B-spline curve, called the *base* curve, which captures the overall shape of the point set but produces curvatures varying smoothly. For the base curve, we adopt the bisection method [5,16] for error-bounded B-spline curve approximation with a roughly estimated tolerance. In this work, we use 1%–5% of the longest edge length of the bounding box of the point set as the tolerance.

When a curve $\mathbf{C}(t)$ is given, the curvature k_i at the point $\mathbf{p}_i$ is easily obtained as follows [1,2]:

$$k_i = \|\dot{\mathbf{C}}(\bar{t}_i) \times \ddot{\mathbf{C}}(\bar{t}_i)\| / \|\dot{\mathbf{C}}(\bar{t}_i)\|^3 \tag{12}$$

where $\dot{\mathbf{C}}(\bar{t}_i)$ and $\ddot{\mathbf{C}}(\bar{t}_i)$ are the first and the second derivatives of the curve $\mathbf{C}(t)$ at the parameter $\bar{t}_i$, respectively. Eq. (12) is used to estimate the initial curvatures of the given points from the base curve and to update the curvatures as the iteration proceeds. When the initial curvatures are not accurate enough, some seed points may be missed. But this missing is mostly compensated by the iterative selection of new dominant points.

Fig. 3 shows an example of selecting seed points. Fig. 3(a) shows 251 points sampled from a human face profile of size 84×126. Fig. 3(b) shows a base curve computed from the point set. It is a cubic B-spline curve defined by 16 control points. Fig. 3(c) shows the curvature plot of the given points computed from the base curve. The dotted horizontal line denotes the average curvature. From the curvature plot, we find 10 seed points: 2 end points and 8 curvature maximum points (∇). These seed points become initial dominant points as shown with solid rectangles in Fig. 3(d).

3.2.2. Choice of a new dominant point

With the dominant points selected so far, we can generate a curve $\mathbf{C}(t)$ by knot placement and least-squares minimization, and update the discrete curvatures using the curve. After building up each segment $\mathbf{S}_{s,e}$ with two successive dominant points $\mathbf{d}_j = \mathbf{p}_s$ and $\mathbf{d}_{j+1} = \mathbf{p}_e$, we consider how to select from the current segments one segment to be refined. We first gather the segments that consist of at least three points $(|e - s| > 1)$. Among the points of these segments, we then find the point $\mathbf{p}_c$ at which its deviation to the current curve $\mathbf{C}(t)$ is the largest. The deviation of a point $\mathbf{p}_i$ is defined as $\|\mathbf{C}(\bar{t}_i) - \mathbf{p}_i\|$. We then choose the segment containing the point $\mathbf{p}_c$ to be refined.

Now consider how to choose a *new* dominant point $\mathbf{p}_w$ from the segment $\mathbf{S}_{s,e}$ at which the segment is subdivided into two sub-segments $\mathbf{S}_{s,w}$ and $\mathbf{S}_{w,e}$. Note that the segment $\mathbf{S}_{s,e}$ has at least three points and two dominant points $\mathbf{d}_j = \mathbf{p}_s$ and $\mathbf{d}_{j+1} = \mathbf{p}_e$. Two intuitive approaches can be considered. One is to choose the median of the points between the two dominant points, and the other is to choose the point $\mathbf{p}_c$ if it lies between the two dominant points. It has been experimentally found

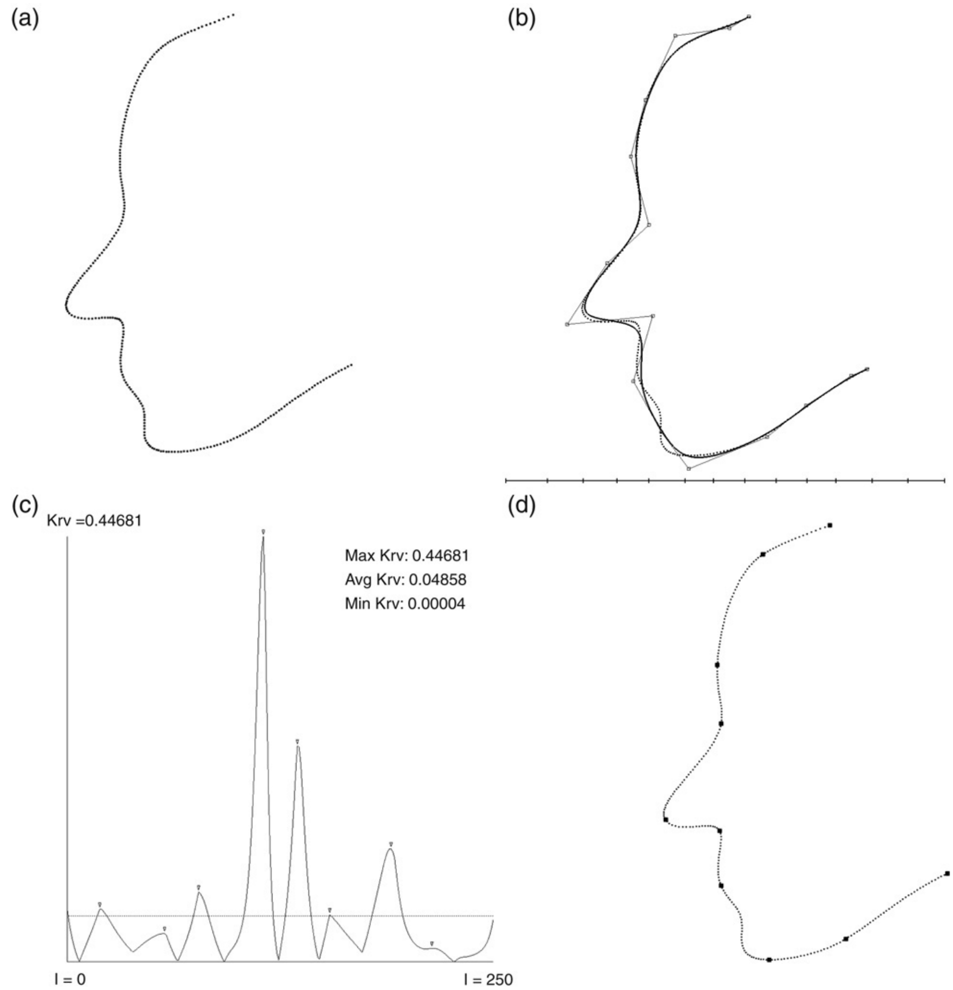

Fig. 3. Selection of seed points: (a) 251 input points; (b) base curve represented by a cubic B-spline curve with 16 control points; (c) discrete curvature plot; (d) 10 initial dominant points.

that these intuitive approaches create worse results than those obtained by an approach of selecting the point $\mathbf{p}_w$ based on the shape information of segments. The first one tends to create curves with more control points. The second one tends to make curves severely affected by the point $\mathbf{p}_c$, yielding less smooth curves. The point $\mathbf{p}_c$ is more useful for finding the segment to be refined.

Consider how to use the shape information for selecting a more promising point $\mathbf{p}_w$. Let $\mathbf{S}_{s,e}$ be a segment that consists of points $\mathbf{p}_k$ $(k = s, \ldots, e)$. The *shape index* $\lambda_{s,e}$ of the segment $\mathbf{S}_{s,e}$ is a measure of how complex the segment is in shape and can be estimated as follows:

$$\lambda_{s,e} = r\frac{K_{s,e}}{K_{0,m}} + (1-r)\frac{L_{s,e}}{L_{0,m}} \tag{13}$$

where $0 \leq r \leq 1$. The values $K_{s,e}$ and $L_{s,e}$ denote the total curvature and the arc length of the segment $\mathbf{S}_{s,e}$, respectively. Using the Trapezoidal rule [24], the total curvature is approximated as $K_{s,e} = \sum_{i=s}^{e-1}(|k_i| + |k_{i+1}|)(\bar{t}_{i+1} - \bar{t}_i)/2$. The denominator $K_{0,m}$ denotes the approximant of the total curvature of the given point set. The arc length is approximated as $L_{s,e} = \sum_{i=s}^{e-1} \|\mathbf{p}_{i+1} - \mathbf{p}_i\|$. The denominator $L_{0,m}$ denotes the approximant of the arc length of the given point set. The weight $r = 0$ corresponds to equidistribution of the arc length (see [19]), and $r = 1$ corresponds to equidistribution of the total curvature (see [19,20]). The arc length term is useful either when the point set is noisy or when discrete curvatures are nearly the same or very small. It is not easy to automate the choice of the weight r, but an empirical guideline is to choose a smaller one for a more noisy point set. Using Eq. (13), we can select a new dominant point $\mathbf{p}_w$ that minimizes the difference of the shape indices of two sub-segments $\mathbf{S}_{s,w}$ and $\mathbf{S}_{w,e}$, that is, $\min_w |\lambda_{s,w} - \lambda_{w,e}|$ where $s < w < e$. Fig. 4 shows two cubic B-spline curves obtained using 10 and 13 dominant points. In Fig. 4(a), the initial dominant points in Fig. 3(d) are used. In Fig. 4(b), 3 more dominant points make the curve rapidly converge to the point set.

3.3. Distance metric for curve fitting

The distance between the given point set $\mathbf{P} = \{\mathbf{p}_i | i = 0, \ldots, m\}$ and the fitted curve $\mathbf{C}(t)$ can be defined in various ways. Two definitions of the distance are considered in this paper. When the degree of mismatch between the curve $\mathbf{C}(t)$ and

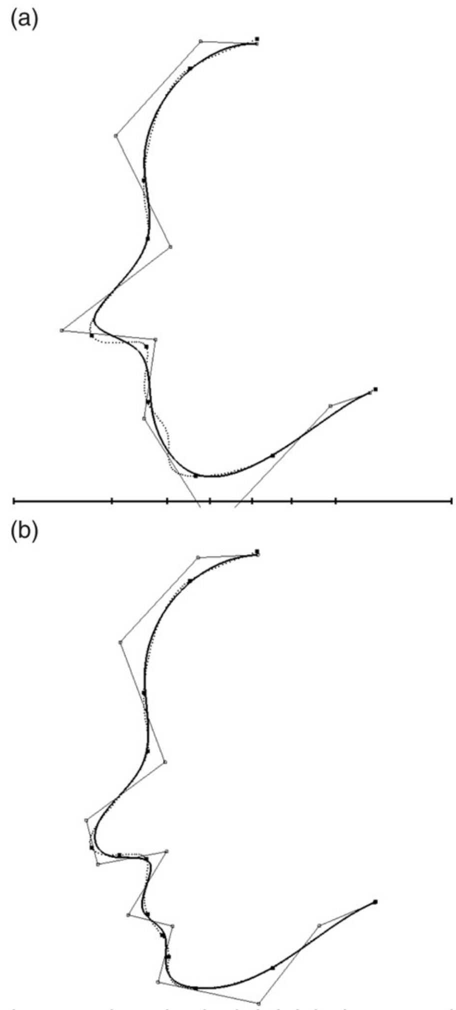

Fig. 4. Cubic B-spline curve fitting by the knot placement using dominant points: (a) 10 dominant points; (b) 13 dominant points.

the discrete points $\mathbf{p}_i$ is considered, the distance is defined as

$$\mathrm{dist}(\mathbf{C}, \mathbf{P}) = \max_i \mathrm{dist}(\mathbf{C}, \mathbf{p}_i) = \max_i \|\tilde{\mathbf{p}}_i - \mathbf{p}_i\| \tag{14}$$

where $\mathrm{dist}(\mathbf{C}, \mathbf{p}_i)$ is the distance from the point $\mathbf{p}_i$ to the curve $\mathbf{C}(t)$ and the point $\tilde{\mathbf{p}}_i$ is the orthogonal projection of the point $\mathbf{p}_i$ onto the curve $\mathbf{C}(t)$. When the point set $\mathbf{P}$ forms the vertices of a polyline and the degree of mismatch between the curve $\mathbf{C}(t)$ and all the points on the polyline $\mathbf{P}$ is considered, the distance is defined as

$$\mathrm{dist}(\mathbf{C}, \mathbf{P}) = H(\mathbf{C}, \mathbf{P}) = \max(h(\mathbf{C}, \mathbf{P}), h(\mathbf{P}, \mathbf{C})) \tag{15}$$

where $h(\mathbf{C}, \mathbf{P}) = \max_{\mathbf{a}\in\mathbf{C}} \min_{\mathbf{b}\in\mathbf{P}} \|\mathbf{a} - \mathbf{b}\|$. This distance $H(\mathbf{C}, \mathbf{P})$ is called the Hausdorff distance [5]. The function $h(\mathbf{C}, \mathbf{P})$ is called the directed Hausdorff distance from $\mathbf{C}$ to $\mathbf{P}$. Intuitively, if the Hausdorff distance is d, then every point of $\mathbf{C}$ must be within a distance d of some point of $\mathbf{P}$ and vice versa [5]. It is not easy to compute the distance $H(\mathbf{C}, \mathbf{P})$ directly, but we can check if the distance is smaller than a tolerance [5].

The distance based on Eq. (14) measures how far the given points deviate from the curve, but not how far the curve deviates from the points. The Hausdorff distance measures the dissimilarity between two geometric entities. For a curve passing through all the vertex points of a polyline, the distance based on Eq. (14) is equal to zero, but the Hausdorff distance is not zero in most cases. The more wiggles the curve has, the greater the Hausdorff distance is.

4. Experimental results

We implemented the proposed approach (DOM) with the conventional approaches to B-spline curve fitting using the knot placement techniques KTP and NKTP. The implementation was performed in the C language on an IBM compatible personal computer with an Intel Pentium IV processor. Using various point sets, we firstly compared the proposed approach with the approaches using KTP and NKTP by the deviation incurred with a given number of control points. The results have shown that the proposed approach mostly creates a B-spline curve with less deviation. Fig. 5 shows cubic B-spline curve fitting to the point set in Fig. 3(a). Fig. 5(a) shows the point set with 31 dominant points. Fig. 5(b) shows a cubic B-spline curve obtained by DOM. In this example, $r = 0.8$ is used in Eq. (13). Note that fewer dominant points (curve segments) are generated at flat regions but more at complex regions. Fig. 5(c)–(d) show cubic B-spline curves obtained by KTP and NKTP with 31 control points, respectively. Fig. 6 shows the deviation caused by B-spline curve fitting in Fig. 5. For each approach, we compute the deviation as we increase the number of control points from 16 to 251. The deviation is defined by Eq. (14). Deviations for more than 101 control points were omitted from the plot since they are not differentiable visually. The results show that the deviation caused by DOM is 69% less than the one caused by KTP, and that the deviation caused by NKTP is nearly the same as (actually 3% greater than) the one caused by KTP. These rates are computed as follows. Let d_i^M be the deviation incurred by the curve consisting of i control points that is obtained by a method M. $R_{B/A} = \frac{1}{N}\sum_i \frac{d_i^A - d_i^B}{d_i^A}$ denotes the average reduction rate of deviation caused by a method B with respect to a method A. $R_{\mathrm{DOM/KTP}} = 0.69$ and $R_{\mathrm{NKTP/KTP}} = -0.03$.

Secondly, we compared the proposed approach with the others by the number of control points and by the computational time required for error-bounded B-spline curve approximation. In the error-bounded approximation, given a point set and a tolerance, we seek a B-spline curve that approximates the point set so that the distance between the curve and the point set should be smaller than the tolerance. An important concern in this problem is to reduce the number of B-spline control points while keeping the accuracy in the resulting B-spline curve. Several methods have been presented for error-bounded B-spline curve approximation [2,5,17]. Basically, the error-bounded curve approximation can be achieved by one of two incremental methods which either start with few control points, fit, check deviation, and add control points if necessary; or start with many control points, fit, check deviation, and remove control points if possible [2,5]. Forward incremental methods

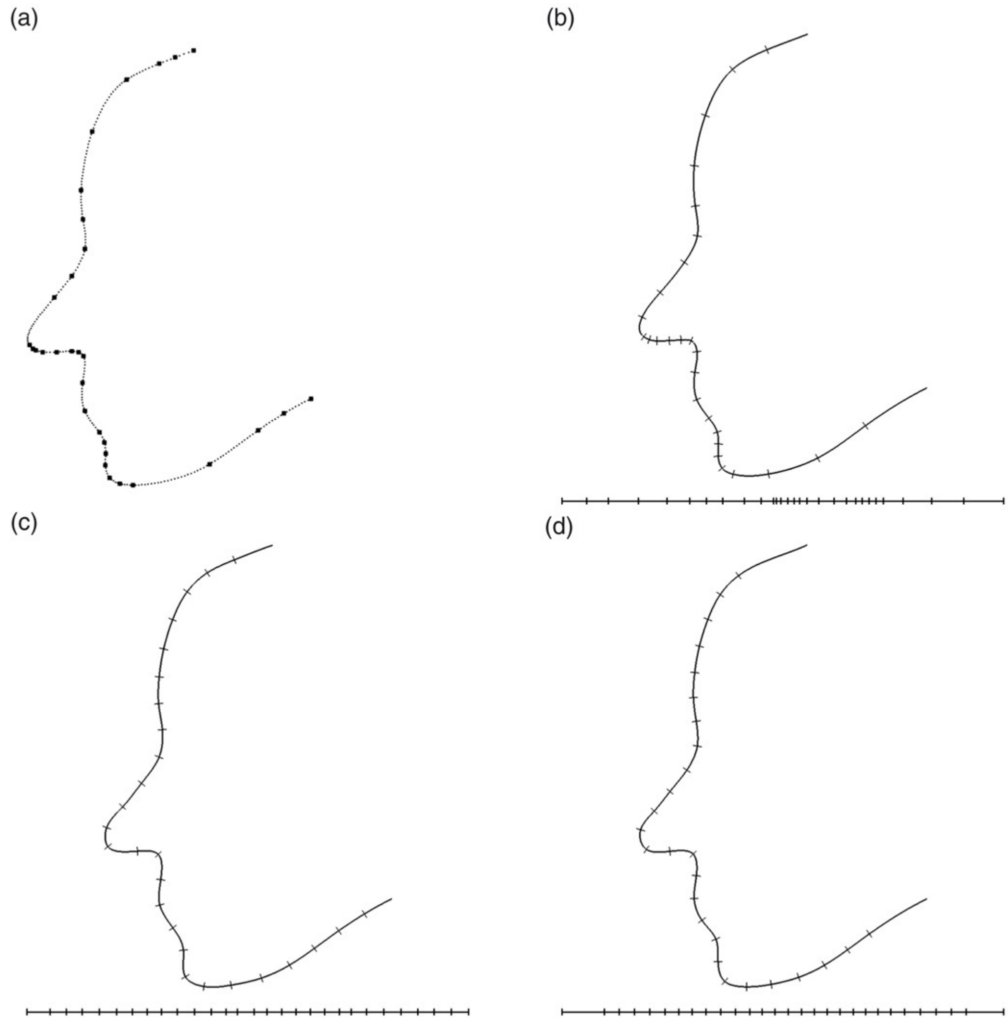

Fig. 5. Cubic B-spline curve fitting with 31 control points ($p = 4, n = 30$): (a) Point set with 31 dominant points; (b) cubic curve obtained with the dominant points; (c) cubic curve obtained by KTP; (d) cubic curve obtained by NKTP.

Table 1
Main steps of iterative methods for error-bounded approximation

	Incremental method using KTP or NKTP	Knot removal method (KRM)	Incremental method using DOM
Preprocessing	Parameterization	Parameterization Interpolation of all points	Parameterization Selection of seed points
Iteration process	Knot placement Least-squares minimization Deviation check	Selection of a candidate knot Deviation check Knot removal	Knot placement Least-squares minimization Deviation check Selection of a new dominant point

usually tend to require less control points than backward incremental methods. A backward incremental method based on knot removal [17] can be adopted for the error-bounded approximation. Starting with a B-spline curve that interpolates all the points, the knot removal method (KRM) progressively removes the knots with small weights from the current B-spline curve while keeping the distance between the point set and the curve smaller than the tolerance.

Table 1 summarizes the main steps of iterative methods for error-bounded approximation using KTP, NKTP, KRM, and DOM. The forward incremental method using Razdan [19] or Li et al. [20] takes similar steps like the one using DOM. We implemented the iterative methods in Table 1. The KRM is based on the work done by Lyche and Mørken [17,18]. Eq. (14) was used for computing the deviation. We compared the four methods by the number of control points and by the computational time required to approximate a given point set within a tolerance δ. Fig. 7 shows the comparison results where the point set in Fig. 8(a) is used. Note that the iterative method using DOM outperforms the iterative methods using the others in both data reduction and computational time. Fig. 8 shows cubic B-spline curve approximation to a point set ($\delta = 0.1$).

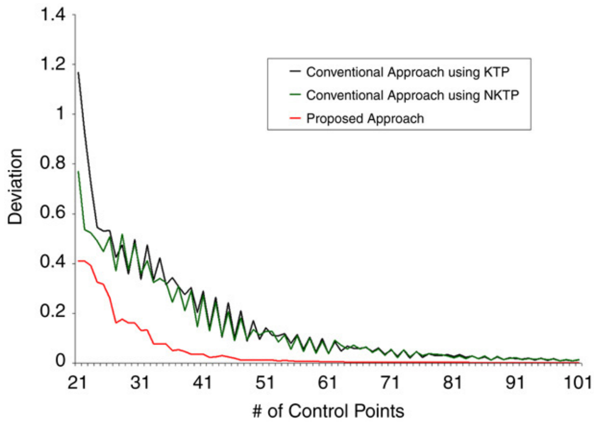

Fig. 6. Plot of deviation incurred with the given number of control points required for curve fitting of the point set in Fig. 3(a).

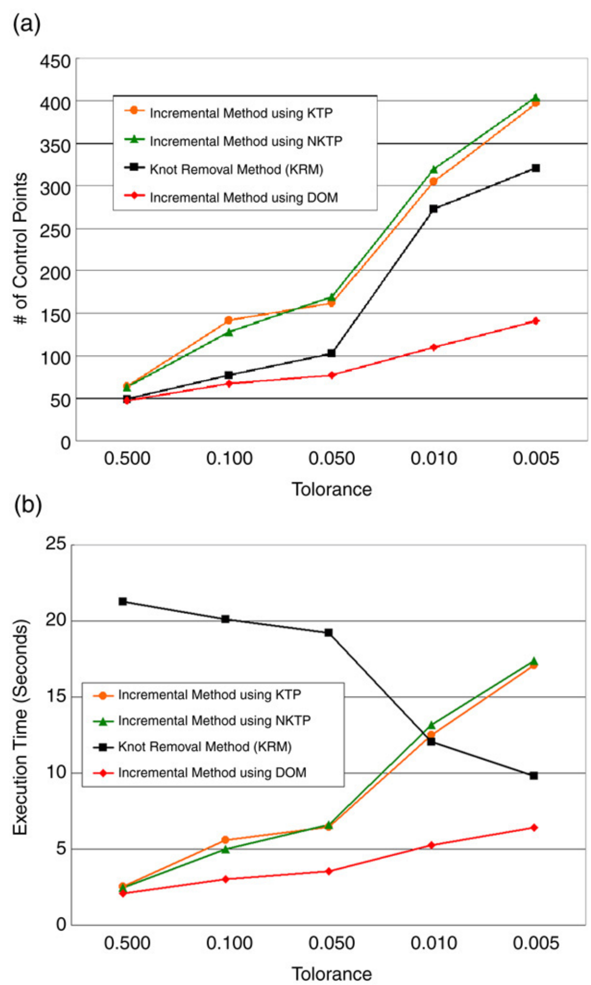

Fig. 7. Comparison results for error-bounded curve fitting of the point set in Fig. 8(a): (a) Plot of the number of control points required; (b) plot of the computational time required.

The point set, enclosed in a 64×78 rectangle, consists of 501 points with some noise. Fig. 8(a) shows the point set with 67 dominant points. Fig. 8(b) shows a B-spline curve obtained by the incremental method using KTP, which requires 128 control points. Fig. 8(c) shows a B-spline curve obtained by the incremental method using NKTP, which requires 128 control points. Fig. 8(d) shows a B-spline curve obtained by the knot removal method (KRM), which requires 77 control points. Fig. 8(e) shows the point set with 67 dominant points. Fig. 8(f) shows a B-spline curve obtained by the incremental method using DOM, which requires 67 control points to maintain the tolerance. In this example, we also set $r = 0.8$ in Eq. (13).

Table 2
Results of vector font reconstruction

Curve index	# of points	# of control points required			
		KTP	NKTP	KRM	DOM
0	207	4	4	123	4
1	287	29	50	173	17
2	24	4	4	13	4
3	166	6	6	39	6
4	292	26	40	193	9
5	1112	47	43	748	24
6	235	4	4	123	4
7	65	4	4	27	4
8	20	4	4	12	4
9	198	5	5	73	5
10	21	4	4	12	4
11	483	12	17	181	8
12	655	23	21	342	17
Total	3765	172	206	2059	110

Fig. 9 shows an example of vector font reconstruction using B-spline curve approximation. For a binary image (600×620 pixels) of a character K, its boundary is extracted and divided into 13 polylines at sharp corner points as shown in Fig. 9(a). Due to image resolution, some noise is inevitably included in the boundary. The error-bounded approximation is applied to obtain cubic B-spline curves from the polylines ($\delta = 1$ pixel). The vector font is then acquired by converting approximated B-spline curves into cubic Bezier curves. Fig. 9(b)–(c) show B-spline curves obtained by the incremental method using KTP and DOM, respectively. In this example, $r = 0.8$ is used for DOM. Table 2 summarizes the results of vector font reconstruction showing how many control points are required by four error-bounded approximation methods. Note that KRM produced unsatisfactory results. This is because the noise size is nearly equal to the given tolerance and the initial curves interpolating all the points have many wiggles over the entire region. Note that KTP, NKTP, and DOM require nearly the same number of control points for the point sets with flat shapes, but that DOM requires a much smaller number of control points for the point sets with complex shapes.

5. Discussion

The proposed approach is discussed in computational performance and robustness to noise, and it is briefly described how to combine the approach with the others for further improvement of the quality of a resulting curve.

5.1. Computational performance

The proposed approach (DOM) can be compared in computational performance with the others including KTP,

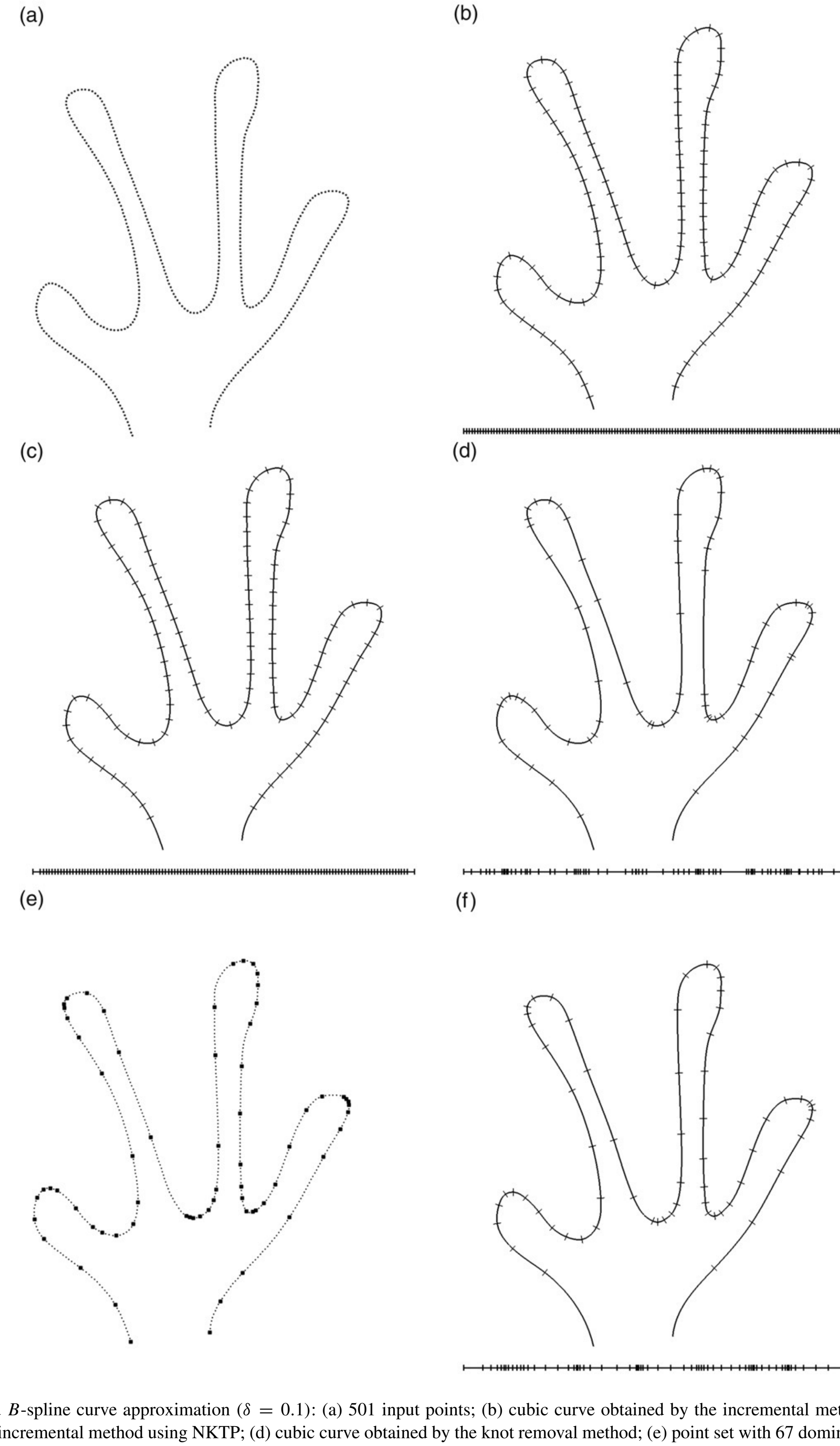

Fig. 8. Error-bounded B-spline curve approximation ($\delta = 0.1$): (a) 501 input points; (b) cubic curve obtained by the incremental method using KTP; (c) cubic curve obtained by the incremental method using NKTP; (d) cubic curve obtained by the knot removal method; (e) point set with 67 dominant points; (f) cubic curve obtained by the incremental method using DOM.

NKTP, and KRM for two cases: one is the case of specifying the number of control points, and the other is the case of specifying a tolerance. In case that the number of control points is specified, KTP and NKTP do not need any iterative process, so they are faster than the other approaches using iterative processes such as DOM, KRM, Razdan [19], and Li et al. [20].

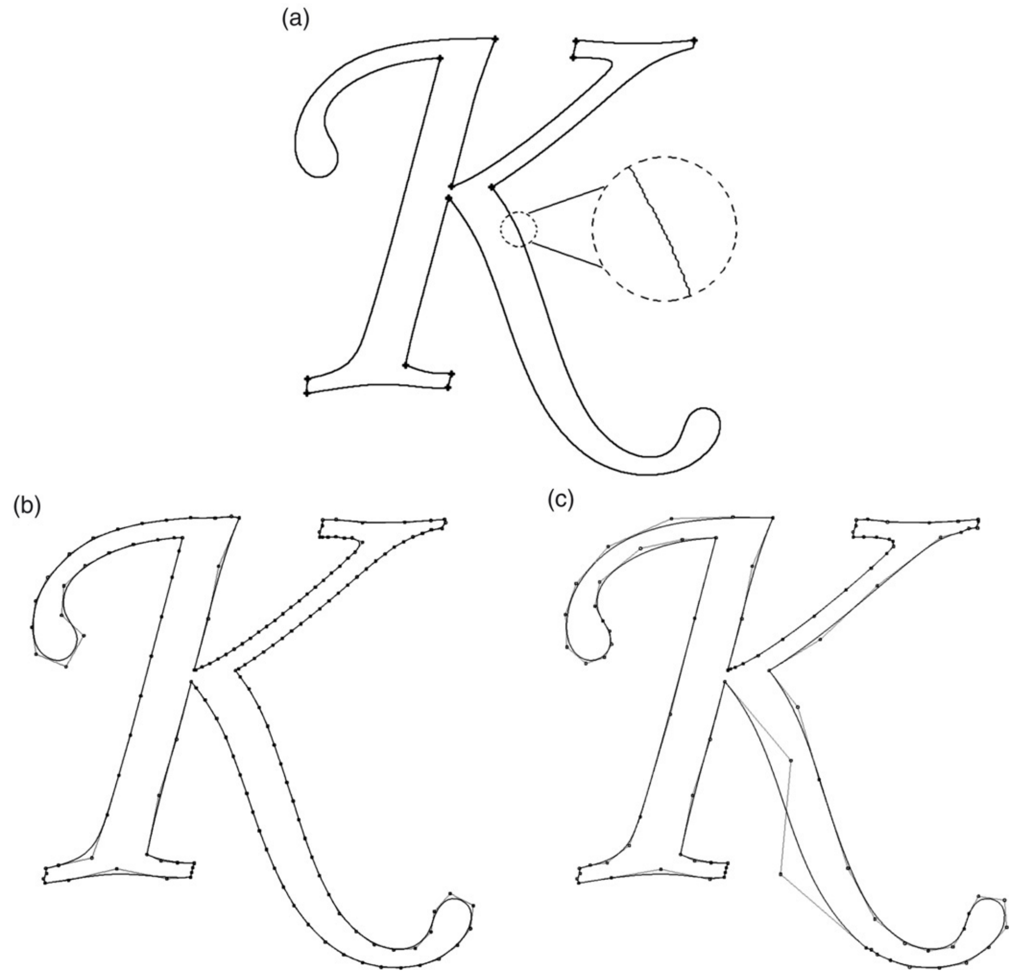

Fig. 9. Vector font reconstruction ($\delta = 1.0$): (a) Boundary extracted from the binary image of a character K; (b) B-spline curves obtained by the incremental method using KTP (172 control points in total); (c) B-spline curves obtained by the incremental method using DOM (110 control points in total).

In DOM, the increased computation leads to the improved quality of resulting curves.

In the case that a tolerance is specified, the approaches KTP, NKTP, KRM, and DOM should be incorporated into the iterative methods for error-bounded approximation as shown in Table 1. The computational performance of each iterative method depends on the computational load required for preprocessing, the number of iterations, and the computational load of each iteration. Parameterization has no influence on the computational comparison since it is invoked just once and equally applied to all the methods. The selection of seed points requires a few times more computation than the interpolation of all the points. Thus, the computational load of preprocessing increases in order as follows: KTP/NKTP, KRM, DOM. The knot placement of DOM has the same order of computational complexity as the ones of KTP or NKTP. The selection of a new dominant point needs a few times more computation than parameterization. Least-squares minimization and/or deviation checking take a large portion of the computation required for each iteration. The computational load of each iteration increases in order as follows: KRM, KTP/NKTP, DOM.

For the same number of iterations, the computational performance decreases in order as follows: KRM, KTP, NKTP, DOM. The methods using KTP and NKTP usually show similar computational performance. The method using DOM takes more computational time both in preprocessing and each iteration than the others, but it usually requires the smallest number of iterations, which makes it outperform the others in computation time (see Fig. 7).

5.2. Robustness to noise

The proposed approach (DOM) is as robust to noise as KTP, NKTP, and Li et al. [20], but it is more robust to noise than the knot removal approach (KRM) and the approach proposed by Razdan [19]. When the given points are noisy, KRM usually stops in a few iterations and creates a resulting curve with many wiggles as seen in Fig. 9(b). This is mainly because KRM starts with an initial curve that interpolates all the points. The approach proposed by Razdan [19] is based on interpolation of node points, so it often yields undesirable results since it lacks a process to exclude noisy node points.

5.3. Further improvement

We can improve the quality of a resulting B-spline curve by incorporating parameter correction [1,3,11,12] into the proposed approach. Moreover, further improvement can be acquired by using the resulting curve as the initial solution of global methods [4,9,10] for construction of an optimal B-spline curve where the choice of good initial solutions is very important.

6. Concluding remarks

We have proposed a new approach for B-spline curve fitting to a sequence of points. Once dominant points are selected, knots are determined using their parameters, and a B-spline curve is easily obtained by least squares minimization. We described the interesting properties of the knot placement using dominant points. Among them, the property of local modification has been found to be very useful for adaptive curve refinement. We also presented an algorithm for dominant point selection based on the adaptive refinement paradigm. The algorithm basically searches a segment with the largest deviation and chooses a new dominant point to subdivide the segment into two sub-segments with nearly the same shape complexity. With the knot placement and the dominant point selection, the proposed approach can realize the concept of adaptive curve refinement that fewer curve segments are generated at flat regions but more at complex regions. Consequently, the approach can generate a B-spline curve of good quality with less deviation. Adopted in error-bounded curve approximation, it can play an important role in generating B-spline curves with far fewer control points.

The proposed approach can be applied to points in space, not confined to planar points. It can be adopted as a key ingredient for B-spline curve approximation of freeform curves [5]. For future works, we are devising non-deterministic algorithms such as genetic algorithms [25,26] for the optimal selection of dominant points. We are also expanding this approach for B-spline surface fitting of grid points or a sequence of profiles [1, 2]. We expect that, when applied in B-spline surface fitting or skinning [2,6], the expanded approach can play an important role in reducing the number of control points of a resulting B-spline surface.

References

[1] Hoschek J, Lasser D. Fundamentals of computer aided geometric design. London: AK Peters; 1993.
[2] Piegl L, Tiller W. The NURBS book. New York: Springer-Verlag; 1995.
[3] Rogers DF, Fog NG. Constrained B-spline curve and surface fitting. Computer-Aided Design 1989;21(10):641–8.
[4] Sarkar B, Menq CH. Parameter optimization in approximating curves and surfaces to measurement data. Computer Aided Geometric Design 1991; 8(4):267–90.
[5] Park H. An error-bounded approximate method for representing planar curves in B-splines. Computer Aided Geometric Design 2004;21(5): 479–97.
[6] Piegl L, Tiller W. Surface approximation to scanned data. The Visual Computer 2000;16(7):386–95.
[7] Wang W, Pottmann H, Liu Y. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. ACM Transactions on Graphics 2006;25(2):214–38.
[8] Fang L, Gossard DC. Multidimensional curve fitting to unorganized data points by nonlinear minimization. Computer-Aided Design 1995;27(1): 48–58.
[9] Laurent-Gengoux P, Mekhilef M. Optimization of a NURBS representation. Computer-Aided Design 1993;25(11):699–710.
[10] Goldenthal R, Bercovier M. Spline curve approximation and design by optimal control over the knots. Computing 2004;72:53–64.
[11] Hoschek J. Intrinsic parameterization for approximation. Computer Aided Geometric Design 1988;5(1):27–31.
[12] Speer T, Kuppe M, Hoschek J. Global reparametrization for curve approximation. Computer Aided Geometric Design 1998;15(9):869–77.
[13] De Boor C. A practical guide to splines. Berlin: Springer; 1978.
[14] Vassilev TI. Fair interpolation and approximation of B-splines by energy minimization and points insertion. Computer-Aided Design 1996;28(9): 753–60.
[15] Celniker G, Gossard D. Deformable curve and surface finite elements for free-form shape design. Computer Graphics 1991;25(4):257–66.
[16] Park H, Kim K, Lee SC. A method for approximate NURBS curve compatibility based on multiple curve refitting. Computer-Aided Design 2000;32(4):237–52.
[17] Lyche T, Mørken K. Knot removal for parametric B-spline curves and surfaces. Computer Aided Geometric Design 1987;4(3):217–30.
[18] Lyche T, Mørken K. A data-reduction strategy for splines with applications to the approximation of functions and data. IMA Journal of Numerical Analysis 1988;8:185–208.
[19] Razdan A. Knot placement for B-spline curve approximation. Report 1999. Arizona State University. http://citeseer.ist.psu.edu/398077.html.
[20] Li W, Xu S, Zhao G, Goh LP. Adaptive knot placement in B-spline curve approximation. Computer-Aided Design 2005;37(8):791–7.
[21] Cook WJ, Cunningham WH, Pulleyblank WR, Schrijver A. Combinatorial optimization. New York: John Wiley and Sons; 1998.
[22] Hamann B, Chen JL. Data point selection for piecewise linear curve approximation. Computer Aided Geometric Design 1994;11(3):289–301.
[23] Liu GH, Wong YS, Zhang YF, Loh HT. Adaptive fairing of digitized data with discrete curvature. Computer-Aided Design 2002;34(4):309–20.
[24] Boehm W, Prautzch H. Numerical methods. Massachusetts: AK Peters; 1993.
[25] Yoshimoto F, Harada T, Yoshimoto Y. Data fitting with a spline using a real-coded genetic algorithm. Computer-Aided Design 2003;35(8): 751–60.
[26] Sarfraz M. Representing shapes by fitting data using an evolutionary approach. Computer-Aided Design & Applications 2004;1:179–86.

Hyungjun Park received his BS, MS, and PhD in Industrial Engineering from POSTECH (Pohang University of Science and Technology), Korea, in 1991, 1993, and 1996, respectively. From 1996 to 2001, he worked as a senior researcher at Samsung Electronics, Korea. He involved in developing commercial CAD/CAM software and in-house software for modeling and manufacturing aspheric lenses used in various optical products. Since 2001, he has been a faculty member of Industrial Engineering at Chosun University, Korea. His current research interests include geometric modeling, virtual prototyping of engineered products, 3D shape reconstruction using reverse engineering, bio-medical engineering, and CAD/CAM/CG applications.

Joo-Haeng Lee received his BS, MS, and PhD in Computer Science from POSTECH, Korea, in 1994, 1996 and 1999, respectively. He joined ETRI (Electronics and Telecommunications Research Institute), Korea in 1999, and is a Senior Researcher at Digital Actor Team, Digital Contents Division. His research interests include geometric modeling and processing algorithms for computer graphics, CAD, and robotics. He is also interested in embedding intelligence for computer graphics, CAD, and robotics applications.