Computer-Aided Design 43 (2011) 258–264

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

B-spline surface fitting based on adaptive knot placement using dominant columns

Hyungjun Park *

Department of Industrial Engineering, Chosun University, 375 Seosuk-Dong, Dong-Gu, Gwangju 501-759, South Korea

ARTICLE INFO

Article history:
Received 19 October 2009
Accepted 1 December 2010

Keywords:
B-spline surface fitting
Approximate B-spline surface lofting
Multiple B-spline curve fitting
Adaptive knot placement
Dominant columns

ABSTRACT

By expanding the idea of B-spline curve fitting using dominant points (Park and Lee 2007 [13]), we propose a new approach to B-spline surface fitting to rectangular grid points, which is based on adaptive knot placement using dominant columns along u- and v-directions. The approach basically takes approximate B-spline surface lofting which performs adaptive multiple B-spline curve fitting along and across rows of the grid points to construct a resulting B-spline surface. In multiple B-spline curve fitting, rows of points are fitted by compatible B-spline curves with a common knot vector whose knots are computed by averaging the parameter values of dominant columns selected from the points. We address how to select dominant columns which play a key role in realizing adaptive knot placement and thereby yielding better surface fitting. Some examples demonstrate the usefulness and quality of the proposed approach.

1. Introduction

B-splines have been one of the de facto industrial standards for geometric representation schemes for more than two decades [1–3]. B-spline surface fitting to rectangular grid points is still considered as an essential operation for surface design and modeling in many applications such as CAD/CAM, computer graphics, and reverse engineering. Although there have been numerous works [1–8] on such B-spline surface fitting, one of the most common approaches takes the following straightforward steps: determination of parameter values, choice of the number of control points, placement of knots along each parametric direction, and computation of the control points via least-squares minimization [1,2,4]. However, the results of B-spline surface fitting are severely affected by how to select the parameter values and the knot values. Especially, in most of the previous knot placement techniques for B-spline surface fitting, knots are selected in a manner such that each knot span contains almost the same number of parameter values, which makes it difficult to realize adaptive knot placement that places fewer knots at flat regions but more at complex regions. Consequently, it is not efficient enough to construct a compact but accurate B-spline surface, especially when the point data has sharp peaks considered as not outliers but features.

In this paper, we propose a new approach to B-spline surface fitting to rectangular grid points, which is based on adaptive knot placement using dominant columns along u- and v-directions. The approach basically takes approximate B-spline surface lofting in which the problem of B-spline surface fitting is decomposed into the sub-problems of multiple B-spline curve fitting [9–12] to serial rows of points running along each parametric direction. In multiple B-spline curve fitting, the rows of points are fitted by compatible B-spline curves whose knots are computed by the adaptive knot placement using dominant columns selected from the points. We address how to select the dominant columns which play a key role in realizing adaptive knot placement and thereby yielding better results for multiple B-spline curve fitting and, correspondingly, for B-spline surface fitting. It is shown by experimental results that the proposed approach can yield better approximation than conventional approaches based on trivial knot selection strategies.

2. Previous work

We assume that the reader is familiar with the concepts of B-spline curves and surfaces. A parametric B-spline curve of order (degree + 1) p is defined as $\mathbf{C}(t) = \sum_{i=0}^{n} N_{i,p}(t)\mathbf{b}_i$ where $\mathbf{b}_i$ are control points and $N_{i,p}(t)$ are B-spline basis functions defined over a knot vector $\mathbf{T} = \{t_0, \ldots, t_{n+p}\}$. A parametric tensor product B-spline of orders p and q is defined as $\mathbf{S}(u, v) = \sum_{i=0}^{m}\sum_{j=0}^{n} N_{i,p}(u)M_{j,q}(v)\mathbf{v}_{i,j}$ where $\mathbf{v}_{i,j}$ are control points, and $N_{i,p}(u)$ and $M_{j,q}(v)$ are B-spline basis functions defined over $\mathbf{U} = \{u_0, \ldots, u_{m+p}\}$ and $\mathbf{V} = \{v_0, \ldots, v_{n+q}\}$.

In the problem of B-spline surface fitting to grid points $\mathbf{p}_{i,j}$ ($i = 0, \ldots, r; j = 0, \ldots, s$), we want each point $\mathbf{p}_{i,j}$ to be a B-spline surface point at specific parameter values $(\bar{u}_{i,j}, \bar{v}_{i,j})$, that is, $\mathbf{S}(\bar{u}_{i,j}, \bar{v}_{i,j}) = \mathbf{p}_{i,j}$. With a common assumption that the grid points running along each row or column have the same parameter values, i.e., $(\bar{u}_{i,j}, \bar{v}_{i,j}) = (\bar{u}_i, \bar{v}_j)$, the control points $\mathbf{v}_{i,j}$ of the surface

* Tel.: +82 62 230 7039; fax: +82 62 230 7128.
E-mail address: hzpark@chosun.ac.kr.

0010-4485/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2010.12.001

$S(u, v)$ can be determined by minimizing the least-squares error $E(\mathbf{v}_{0,0}, \dots, \mathbf{v}_{m,n}) = \sum_{i=0}^{r}\sum_{j=0}^{s}\left\|\mathbf{S}(\bar{u}_i, \bar{v}_j) - \mathbf{p}_{i,j}\right\|^2$. This minimization problem is equivalent to solving a linear system. For computational efficiency in solving the minimization problem, however, approximate B-spline surface lofting (or skinning) has been widely adopted [9–12] where the control points are obtained by successively applying multiple B-spline curve fitting to a set of polylines along each parametric direction.

By introducing *intermediate* points $\mathbf{g}_{i,b}$, we can simplify the equation $\mathbf{S}(\bar{u}_i, \bar{v}_j) = \mathbf{p}_{i,j}$ as follows: $\mathbf{p}_{i,j} = \sum_{b=0}^{n} N_{b,q}(\bar{v}_j)\mathbf{g}_{i,b} = \mathbf{C}_i(\bar{v}_j)$ for $i \in [0, \dots, r]$ where $\mathbf{g}_{i,b} = \sum_{a=0}^{m} N_{a,p}(\bar{u}_i)\mathbf{v}_{a,b}$. This means multiple B-spline curve fitting to $(r + 1)$ polylines each of which consists of $(s + 1)$ points passing along the v-direction. We can obtain the control points $\mathbf{g}_{i,b}$ for $(i, b) \in [0, \dots, r] \times [0, \dots, n]$ by solving the multiple B-spline curve fitting where n, q, $\bar{v}_j$, and $\mathbf{V}$ are given. The intermediate points $\mathbf{g}_{i,b}$ are then expressed as follows: $\mathbf{g}_{i,b} = \sum_{a=0}^{m} N_{a,p}(\bar{u}_i)\mathbf{v}_{a,b} = \mathbf{C}_b(\bar{u}_i)$ for $b \in [0, \dots, n]$. Similarly, this means multiple B-spline curve fitting to $(n + 1)$ polylines each of which consists of $(r + 1)$ points passing along the u-direction. We can finally obtain the control points $\mathbf{v}_{a,b}$ for $(a, b) \in [0, \dots, m] \times [0, \dots, n]$ by solving the multiple B-spline curve fitting where m, p, $\bar{u}_i$, and $\mathbf{U}$ are given.

The multiple B-spline curve fitting is a key ingredient of approximate B-spline surface lofting. It approximates a given set of polylines by *compatible* B-spline curves which are simultaneously defined on the same knot vector with the same B-spline order and the same number of control points [9–12]. Consider polylines $\mathbf{P}_i = \{\mathbf{p}_{i,j} | j = 0, \dots, m_c\}$ $(i \in [0, \dots, m_r])$ to make rows of points running along a parametric direction. We seek B-spline curves $\mathbf{C}_i(t)$ defined over a common knot vector $\mathbf{T}$ satisfying $\mathbf{C}_i(\bar{t}_j) = \mathbf{p}_{i,j}$ in the sense of approximation, so the control points $\mathbf{b}_{i,k}$ $(k \in [0, \dots, n])$ of the curves can be determined by minimizing the following least-squares errors:

$$E(\mathbf{b}_{i,0}, \dots, \mathbf{b}_{i,n}) = \sum_{j=0}^{m_c}\left\|\mathbf{C}_i(\bar{t}_j) - \mathbf{p}_{i,j}\right\|^2 \quad \text{for } i \in [0, \dots, m_r]. \tag{1}$$

Each minimization problem is equivalent to solving a linear system. The quality of the fitted curves heavily depends on how to select the parameter values and the knots. The widely adopted approach first determines the parameter values and then selects the knots by taking account of them. The parameter values are mostly computed using the chord length or centripetal methods [1,2]. Several ways of placing the interior knots of the knot vector $\mathbf{T}$ have been suggested including the averaging technique (AVG) for $m_c = n$, and the knot placement technique (KTP) for $m_c > n$ [1,2]. When m_c is nearly greater than n ($|m_c - n|$ is small), it often generates undesirable results [13,10,12]. To avoid this, Piegl and Tiller [12] suggested another knot placement technique (NKTP). However, these previous knot placement techniques select the knots in a simple and trivial manner that each knot span contains almost the same number of parameter values, and that all interior knots are changed even though the number n increases by one, which makes it difficult to realize adaptive fitting. Recently, Park and Lee [13] presented a new approach to knot placement for B-spline curve fitting. After selecting dominant points from the given points, it determines the interior knots by averaging the parameter values of the dominant points. This knot placement results in a stable system matrix which is not singular, and supports local modification realizing adaptive curve fitting that fewer knots are placed at flat regions but more at complex regions.

3. Proposed approach

By expanding the idea of B-spline curve fitting using dominant points [13], we have developed adaptive B-spline surface fitting using dominant columns along u- and v-directions. The overall procedure for such B-spline surface fitting is summarized as follows:

(Procedure 1) Adaptive B-spline surface fitting using dominant columns

(1) Specify B-spline orders (p, q) and determine parameter values $\bar{u}_i$, $\bar{v}_j$ of points $\mathbf{p}_{i,j}$.
(2) Do v-directional multiple B-spline curve fitting, which converts the points $\mathbf{p}_{i,j}$ to intermediate points $\mathbf{g}_{i,b}$.
  [a] Extract v-directional polylines from the points $\mathbf{p}_{i,j}$.
  [b] Select $(n + 1)$ dominant columns from the v-directional polylines, and obtain a knot vector $\mathbf{V}$ by using the parameter values $\bar{v}_j$ of the dominant columns.
  [c] Compute the points $\mathbf{g}_{i,b}$ by solving the least-squares minimization in Eq. (1).
(3) Similarly, do u-directional multiple B-spline curve fitting, which converts the points $\mathbf{g}_{i,b}$ to final control points $\mathbf{v}_{a,b}$.
  [a] Extract u-directional polylines from the points $\mathbf{g}_{i,b}$.
  [b] Select $(m + 1)$ dominant columns from the u-directional polylines, and obtain a knot vector $\mathbf{U}$ by using the parameter values $\bar{u}_i$ of the dominant columns.
  [c] Compute control points $\mathbf{v}_{a,b}$ by solving the least-squares minimization in Eq. (1).
(4) Make a B-spline surface $\mathbf{S}(u, v)$ with (p, q), (m, n), $(\mathbf{U}, \mathbf{V})$, and $\mathbf{v}_{a,b}$.

We determine the parameter values $(\bar{u}_i, \bar{v}_j)$ according to Piegl and Tiller [2]. There are usually two criteria of quality in B-spline surface fitting to a point set. First, we seek a B-spline surface with a fixed number (i.e., m and n) of control points that minimizes its deviation to the data set. Second, we seek a B-spline surface with a minimal number of control points that keeps its deviation smaller than a specified tolerance. In either case, we can obtain the surface by successively applying multiple B-spline curve fitting as described in Procedure 1. We can also improve the quality of multiple B-spline curves and, correspondingly, a resulting B-spline surface by adjusting the parameter values and repeating the multiple curve fitting. This is called parameter correction [1,14].

Especially, in the second case, the resulting surface is called *error bounded*. We can obtain such a surface by equally dividing the tolerance into two and applying the error-bounded multiple B-spline curve fitting along each parametric direction. As it is not known in advance how many control points are required to construct a set of compatible B-spline curves which are within the desired accuracy, it is common to take an iterative process which repeats selection of dominant columns, computation of B-spline curves, checking deviation, and adjusting the number of dominant columns.

Compared to the conventional approaches [1–8] based on trivial knot selection strategies, the proposed approach is substantially different in knot placement and dominant column selection, which will be described in the following subsections.

3.1. Knot determination using dominant columns

For a set of grid points $\mathbf{p}_{i,j}$ $(i = 0, \dots, m_r; j = 0, \dots, m_c)$, we have $(m_r + 1)$ polylines $\mathbf{P}_i = \{\mathbf{p}_{i,j} | j = 0, \dots, m_c\}$ and $(m_c + 1)$ columns $\mathbf{Q}_j = \{\mathbf{p}_{i,j} | i = 0, \dots, m_r\}$. For the time being, assume that dominant columns $\mathbf{D}_j$ for $j \in [0, \dots, n]$ are selected from the columns $\mathbf{Q}_k$. That is, $\mathbf{Q}_{f(j)} = \mathbf{D}_j$ where $f(j)$ is a monotonically increasing function that returns the index of the column corresponding to the jth dominant column. Given B-spline order p and the parameter values $\bar{t}_j$ of the columns, we determine interior knots by averaging the parameter values of the dominant columns as follows:

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

With these knots, we perform the least-squares curve fitting to all the polylines (i.e., solve the least-squares minimization in Eq. (1)) in order to compute compatible B-spline curves.

3.2. Selection of dominant columns

In this work, the quality of multiple B-spline curve fitting depends on how to select dominant columns from a set of polylines. We can possibly determine the dominant columns by solving a combinatorial optimization problem with some objective function, but this approach is too time consuming to be used for practical applications. We tackle this with an algorithm based on adaptive refinement paradigm as follows. Let $\mathbf{S}^{s,e} = \{\mathbf{Q}_j | j = s, \ldots, e\}$ be a column subset whose two end columns $\mathbf{Q}_s$ and $\mathbf{Q}_e$ have been already chosen as dominant columns. Starting with a subset $\mathbf{S}^{0,m_c}$ and two initial dominant columns $\mathbf{Q}_0$ and $\mathbf{Q}_{m_c}$, we perform the iterative segmentation of a sequence of column subsets by selecting the subset with the largest deviation and splitting it into two. Any column subset $\mathbf{S}^{s,e}$ to be selected should have at least three columns in it ($|e - s| > 1$). For a column $\mathbf{Q}_w$ ($s < w < e$) determined as a new dominant column, we divide the subset $\mathbf{S}^{s,e}$ into two subsets $\mathbf{S}^{s,w}$ and $\mathbf{S}^{w,e}$. Iteration continues until we select $(n + 1)$ dominant columns $\mathbf{D}_j$ for $j \in [0, \ldots, n]$. As iteration proceeds, the algorithm chooses new dominant columns without changing the other dominant columns. This means that the algorithm supports the local refinement of knots which is useful for adaptive multiple curve fitting [13].

3.2.1. Selection of the column subset with the largest deviation

Let n_d denote the highest index of current dominant columns $\mathbf{D}_k$ ($k = 0, \ldots, n_d$). This implies that we have n_d column subsets. The adaptive local refinement necessitates a proper selection of a column subset in which a new dominant column is determined. In this work, this is simply completed by finding the subset with the largest deviation which is either the largest chord height at initial iterations or the largest curve fit deviation later. The deviation $D^{s,e}$ of a subset $\mathbf{S}^{s,e}$ is expressed as

$$D^{s,e} = \begin{cases} \max\limits_{j=s}^{e} \left(\max\limits_{i=0}^{m_r} \left\| \mathbf{p}_{i,j} - \tilde{\mathbf{p}}_{i,j} \right\| \right) & \text{if } \dfrac{n_d}{m_c} \le \alpha \\ \max\limits_{j=s}^{e} \left(\max\limits_{i=0}^{m_r} \left\| \mathbf{p}_{i,j} - \mathbf{C}_i^{\mathrm{cur}}(\bar{t}_j) \right\| \right) & \text{otherwise} \end{cases} \tag{3}$$

where $\tilde{\mathbf{p}}_{i,j}$ is the chord height of the $\mathbf{p}_{i,j}$ with respect to the line segment $\overline{\mathbf{p}_{i,s}\mathbf{p}_{i,e}}$ and $\mathbf{C}_i^{\mathrm{cur}}(t)$ is the B-spline curve obtained with the dominant columns $\mathbf{D}_k$. After determining a common knot vector $\mathbf{T}$ as described in Eq. (2), we obtain compatible B-spline curves $\mathbf{C}_i^{\mathrm{cur}}(t)$ of order p_d by solving the least-squares minimization in Eq. (1) where $p_d = \min(n_d+1, p)$. Starting with $p_d = 2$ (i.e., linear curves), the order increases to p as the iteration proceeds.

Using the curve fit deviation term in Eq. (3) is more precise as a metric of the deviation of a column subset in view of multiple B-spline curve fitting, but less efficient in computation than using the chord height term. If we use only the curve fit deviation in Eq. (3) (i.e., $\alpha = 0$), dominant columns tend to be selected densely at complex regions in a very few iterations, which may cause highly uneven distribution of knots and, correspondingly, deteriorate the quality of resulting B-spline curves, though such a deterioration is relieved gradually with additional dominant columns. It is empirically recommended to start with the chord height term and switch over to the curve fit deviation term. By giving a small fraction to the α (e.g., $0 \le \alpha \le 10^{-1}$) in Eq. (3), we can use the chord height term at early iterations. This is useful especially when the point set has very sharp peaks.

3.2.2. Choice of new dominant columns

Once the column subset with the largest deviation has been found, the next step is to select a new dominant column among its columns in a similar manner described in [13]. A simple approach is to choose the median of the columns (i.e., $w = (s+e)/2$ for a given subset $\mathbf{S}^{s,e}$). However, it is empirically recommended to choose a new dominant column by considering the geometry of the subset.

In this work, we use the shape index $\Omega^{s,e}$ of a subset $\mathbf{S}^{s,e}$ to estimate its shape complexity as follows:

$$\Omega^{s,e} = \frac{1}{m_r + 1} \sum_{i=0}^{m_r} \lambda_i^{s,e} \tag{4}$$

where $\lambda_i^{s,e}$ denotes the shape index of a sub-polyline $\mathbf{P}_i^{s,e} = \{\mathbf{p}_{i,j} | j \in s, \ldots, e\}$. As described in [13], the $\lambda_i^{s,e}$ is defined as a weighted sum of the normalized total curvature and the normalized arc length of the sub-polyline, which is expressed in detail as follows:

$$\lambda_i^{s,e} = \beta \frac{K_i^{s,e}}{K_i^{0,m_c}} + (1 - \beta) \frac{L_i^{s,e}}{L_i^{0,m_c}} \tag{5}$$

where $0 \le \beta \le 1$. The values $K_i^{s,e}$ and $L_i^{s,e}$ denote the total curvature and the arc length of the $\mathbf{P}_i^{s,e}$, respectively. Using Trapezoidal rule [15], the total curvature is approximated as $K_i^{s,e} = \sum_{j=s}^{e-1} \left(|k_{i,j}| + |k_{i,j+1}|\right) \left(\bar{t}_{j+1} - \bar{t}_j\right)/2$ where $k_{i,j}$ is the curvature at the jth point $\mathbf{p}_{i,j}$ of the ith polyline $\mathbf{P}_i$. The denominator K_i^{0,m_c} denotes the approximant of the total curvature of the polyline. The arc length is approximated as $L_i^{s,e} = \sum_{j=s}^{e-1} \left\| \mathbf{p}_{i,j+1} - \mathbf{p}_{i,j} \right\|$. The denominator L_i^{0,m_c} denotes the approximant of the arc length of the polyline. The arc length term is useful when the points of sub-polylines have nearly the same curvatures. An empirical guideline for setting the value β is to start with a small value and increase it as the iteration proceeds. In this work, we set the value as $\beta = (\frac{n_d}{m_c})^2$.

The curvature $k_{i,j}$ at $\mathbf{p}_{i,j}$, which can be estimated either by local methods [16,17] or by curve fitting methods [1,2,10,18], is defined as $k_{i,j} = \left\| \dot{\mathbf{C}}_{i,j} \times \ddot{\mathbf{C}}_{i,j} \right\| / \left\| \dot{\mathbf{C}}_{i,j} \right\|^3$ where $\dot{\mathbf{C}}_{i,j}$ and $\ddot{\mathbf{C}}_{i,j}$ are the first and the second derivatives at $\mathbf{p}_{i,j}$, respectively. We can use the B-spline curve $\mathbf{C}_i^{\mathrm{cur}}(t)$ to directly compute the derivatives $\dot{\mathbf{C}}_{i,j} = \dot{\mathbf{C}}_i^{\mathrm{cur}}(\bar{t}_j)$ and $\ddot{\mathbf{C}}_{i,j} = \ddot{\mathbf{C}}_i^{\mathrm{cur}}(\bar{t}_j)$. As the curve is not usually accurate at early iterations, it is recommended to use the discrete derivatives before a sufficient number of dominant columns are selected. The discrete derivatives, which can be estimated as the average of the left and right discrete derivatives, are defined as follows:

$$\dot{\mathbf{C}}_{i,j} = 0.5 \frac{\mathbf{p}_{i,j} - \mathbf{p}_{i,j-1}}{\bar{t}_j - \bar{t}_{j-1}} + 0.5 \frac{\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j}}{\bar{t}_{j+1} - \bar{t}_j},$$

$$\ddot{\mathbf{C}}_{i,j} = 0.5 \frac{\dot{\mathbf{C}}_{i,j} - \dot{\mathbf{C}}_{i,j-1}}{\bar{t}_j - \bar{t}_{j-1}} + 0.5 \frac{\dot{\mathbf{C}}_{i,j+1} - \dot{\mathbf{C}}_{i,j}}{\bar{t}_{j+1} - \bar{t}_j}.$$

For $j = 0$ ($j = m_c$), only the right (left) discrete derivatives are used.

Using Eqs. (4) and (5), we can select among the columns of the subset $\mathbf{S}^{s,e}$ a new dominant column $\mathbf{Q}_w$ that minimizes the difference of the shape indices of two subsets $\mathbf{S}^{s,w}$ and $\mathbf{S}^{w,e}$, that is, $\min_w |\Omega^{s,w} - \Omega^{w,e}|$ where $s < w < e$. When the given polylines $\mathbf{P}_i$ are symmetric with respect to the column direction, we can keep this symmetry by selecting a column $\mathbf{Q}_{m_c - w}$ as another new dominant column.

4. Examples

Not only the proposed approach but also conventional approaches using trivial knot selection strategies such as the KTP [1,2] and the NKTP [12] have been implemented in the C language on an IBM PC compatible computer with an Intel Core 2 Duo processor T7500 running MS Windows XP. For various data sets, the proposed approach was tested and compared with the

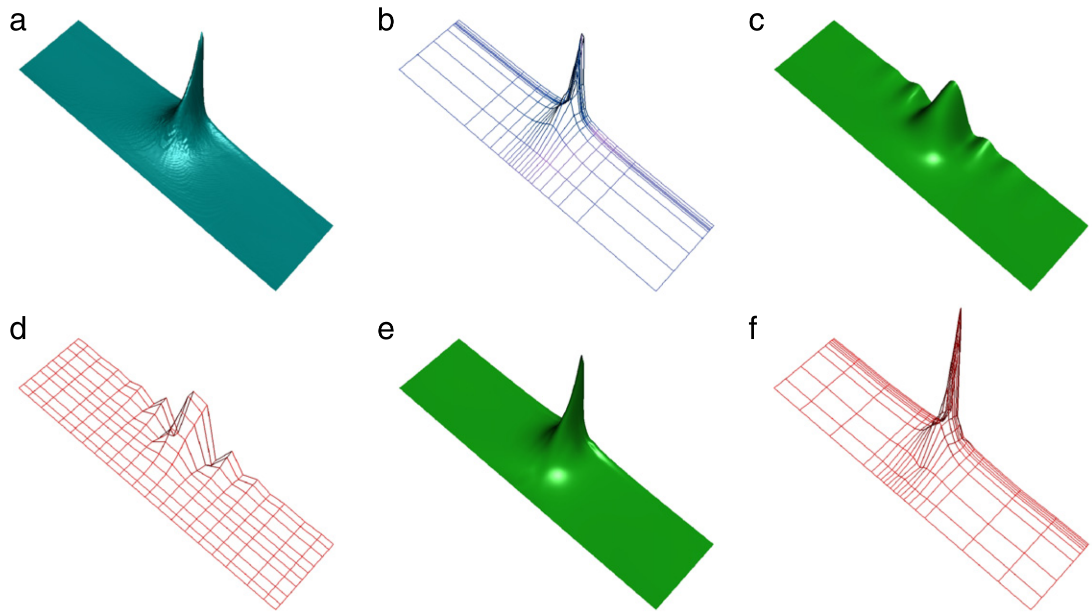

Fig. 1. Conventional approach vs. proposed approach for B-spline surface fitting to a sharply-peaked point data: (a) input point mesh; (b) 20 × 10 dominant mesh; (c) bicubic B-spline surface obtained by a conventional approach; (d) its 20 × 10 control points; (e) bicubic B-spline surface obtained using the dominant mesh; (f) its 20 × 10 control points.

conventional approaches. In this paper, two data sets (sharply peaked point data and mask-shaped point data) are included to demonstrate the usefulness and quality of the proposed approach. As B-spline surface fitting using the trivial knot selection strategies behave in a similar way, B-spline surface fitting using the NKTP is used as a reference of comparison with the proposed approach (hereafter called DOM).

Fig. 1 shows B-spline surface fitting using the NKTP and the DOM. Fig. 1(a) shows a sharply peaked point data (361 × 91 points) acquired from optical measurement and enclosed in a box of size 1.6 × 6.3 × 2.8. The parameter values of the points are constrained to be uniform. Fig. 1(b) shows a 20 × 10 dominant mesh, which contains the information about which columns are used as dominant columns in multiple B-spline curve fitting along u- and v-directions. Fig. 1(c) shows a bicubic B-spline surface obtained by the NKTP which places knots in a way that each knot span has almost the same number of parameter values. Fig. 1(d) shows 20 × 10 control points of the surface in Fig. 1(c). Note that significant gaps between the surface and the point data are found especially around the sharp peak. Shown in Fig. 1(e) is a bicubic B-spline surface which is obtained by the DOM using the dominant mesh. Fig. 1(f) shows 20×10 control points of the surface in Fig. 1(e). Note that, with the same number of control points, the gaps are significantly reduced since control points are placed adaptively (i.e., more at complex regions and fewer at flat regions).

Fig. 2 shows multiple B-spline curve fitting using the conventional knot placement (NKTP) and the proposed knot placement (DOM) with different α and β values. For clear explanation, we use a set of 151 × 11 points sampled from the point set in Fig. 1(a). Fig. 2(a) shows 11 polylines. Fig. 2(b) shows compatible B-spline curves (obtained using the NKTP) and their control polygons, each of which has 13 control points. The curves are drawn in black color and the control polygons in red color. Fig. 2(c), (e), (g), and (i) show dominant columns selected with four combinations of (α, β). Fig. 2(d), (f), (h), and (j) show compatible B-spline curves and their control polygons (with same number of control points) obtained using the DOM. As shown in Fig. 2, the values α and β usually affect the distribution of dominant columns near complex regions as the iteration proceeds. However, the multiple B-spline curve fitting using dominant columns has a notable tendency to generate fewer control points at flat regions but more at complex regions. Thus, it produces better results than the multiple B-spline curve fitting using simple but crude knot placement techniques like the NKTP.

Fig. 3 shows how the DOM approach creates dominant meshes from grid points. Fig. 3(a) shows a mask-shaped point data which consists of 201×201 points enclosed in a box of size 7.2×4.1×6.3. The chord length method [1,2] is used to determine the parameter values of their points. Fig. 2(b)–(d) show three dominant meshes of different size, which are all determined from the input points for $\alpha = \frac{1}{20}$ and $\beta = (\frac{n_d}{m_c})^2$. Notice that dominant columns are adaptively placed in the dominant meshes, which makes them rapidly converge to the given point set as the size increases.

In Fig. 4, the DOM and the NKTP were compared by the average deviations between the two data sets and their bicubic B-spline surfaces which are fitted with different numbers of control points. The results have shown that the DOM creates B-spline surfaces with smaller deviations than the NKTP. A similar trend was found in the largest deviations.

The DOM and the NKTP were also compared by the number of control points required for error-bounded B-spline surface fitting which generates a bicubic B-spline surface approximation of a data set within a tolerance. As the data sets have different geometric sizes, the rate $(0 < \text{rate} < 1)$ is specified to determine the absolute tolerance for each data set as tol $= MR \cdot$ rate where MR is the longest edge length of the enclosing box of the data set. Fig. 5 shows an example of the error-bounded B-spline surface fitting applied to the mask-shaped point data. Fig. 6 shows the number of control points required for B-spline surface fitting to the two data sets for different tolerances. The results have shown that the DOM requires much fewer control points to construct error-bounded surfaces.

5. Concluding remarks

We have proposed a new approach to B-spline surface fitting to rectangular grid points, which constructs a resulting B-spline surface by the successive application of multiple B-spline curve fitting based on adaptive knot placement using dominant columns

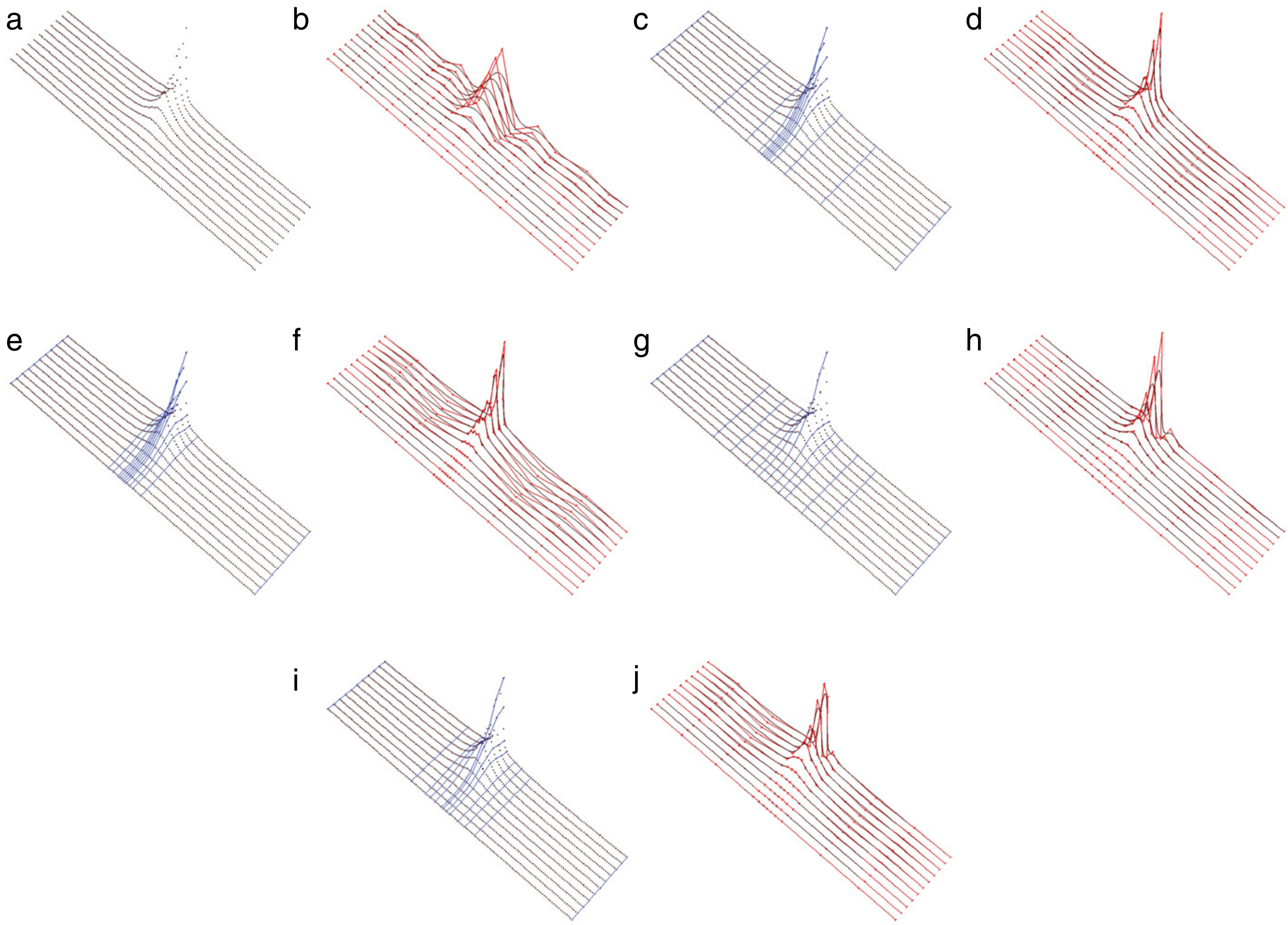

Fig. 2. Multiple B-spline curve fitting: (a) 11 polylines each of which consists of 121 points; (b) B-spline curves (obtained using the NKTP) and their control polygons, each of which has 13 control points; (c) 13 dominant columns for $\alpha = \beta = 0$; (d) B-spline curves and their control polygons (with same number of control points) obtained using the dominant columns in (c). (e) 13 dominant columns for $\alpha = 0, \beta = 1$; (f) B-spline curves and their control polygons obtained using the dominant columns in (e); (g) 13 dominant columns for ($\alpha = \frac{1}{10}, \beta = 0$); (h) B-spline curves and their control polygons obtained using the dominant columns in (g); (i) 13 dominant columns for ($\alpha = \frac{1}{10}$, $\beta = 1$); (j) B-spline curves and their control polygons obtained using the dominant columns in (i).

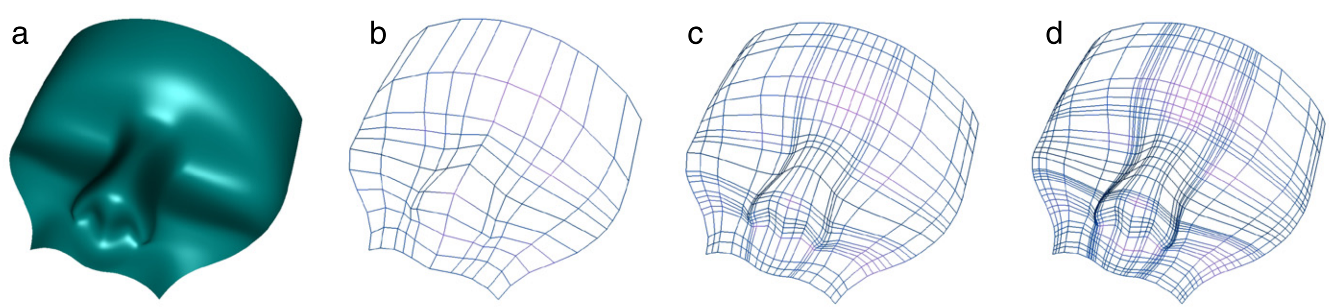

Fig. 3. Dominant meshes determined from a mask-shaped point data ($\alpha = \frac{1}{20}, \beta = (\frac{n_d}{m_c})^2$): (a) 201 × 201 mesh of input points; (b) 11 × 11 dominant mesh; (c) 21 × 21 dominant mesh; (d) 31 × 31 dominant mesh.

along u- and v-directions. Although the proper selection of dominant columns causes some increase in computational time, it plays an important role in making the fitting process adaptive in the sense that more control points are placed at complicated regions and fewer at simple regions. Due to this adaptiveness, the proposed approach can yield better approximation than conventional approaches based on trivial knot selection strategies. For the same number of control points, the approach can generate an accurate B-spline surface with less deviation. When the tolerance is specified for error-bounded fitting, it can compute a compact B-spline surface with fewer control points.

The proposed approach is very general in that it can create B-spline surfaces of any order with non-uniform knots. In the case of $(p, q) = (2, 2)$, it produces a linear approximation of the given points. This makes the approach advantageous over previous methods [19–21] for adaptive or hierarchical surface fitting, which use specific B-spline orders, uniform knots, or other basis functions like Bezier or T-spline functions. Moreover, the proposed approach is very effective to produce compact B-spline surfaces when the point data has sharp peaks considered as not outliers but features, which occurs frequently in scientific data fitting. On the other hand, the approach is not symmetric with respect to swapping

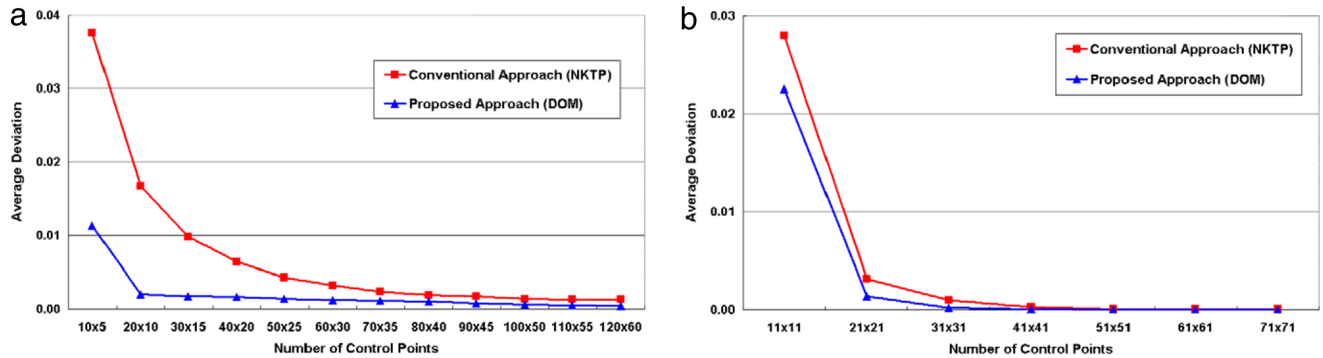

Fig. 4. Plots of deviation errors occurred in B-spline surface fitting: (a) the sharply-peaked point data; (b) the mask-shaped point data.

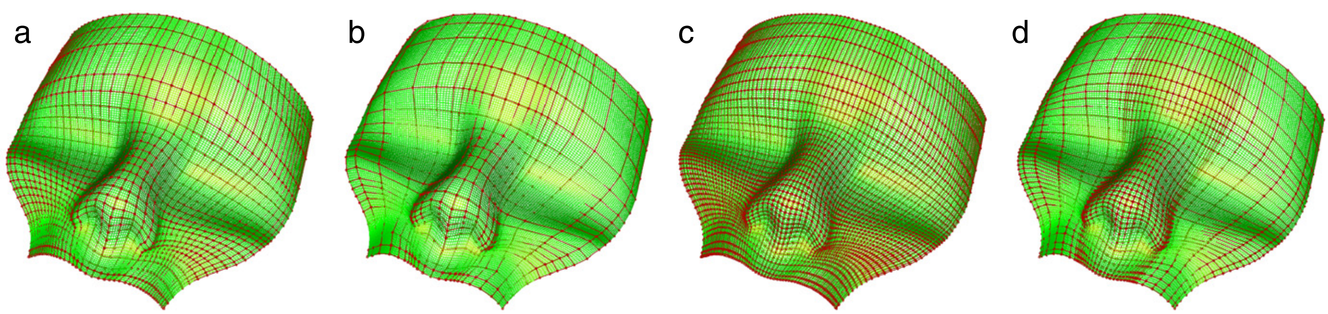

Fig. 5. Error-bounded B-spline surface fitting applied to the mask-shaped point data ($\alpha = \frac{1}{20}$, $\beta = (\frac{n_d}{m_c})^2$): (a) bicubic B-spline surface and its 43×26 control points obtained by the NKTP approach with tol $= 10^{-3}$; (b) bicubic B-spline surface and its 23×22 control points obtained by the DOM approach with tol $= 10^{-3}$; (c) bicubic B-spline surface and its 81×39 control points obtained by the NKTP approach with tol $= 10^{-4}$; (d) bicubic B-spline surface and its 37×35 control points obtained by the DOM approach with tol $= 10^{-4}$.

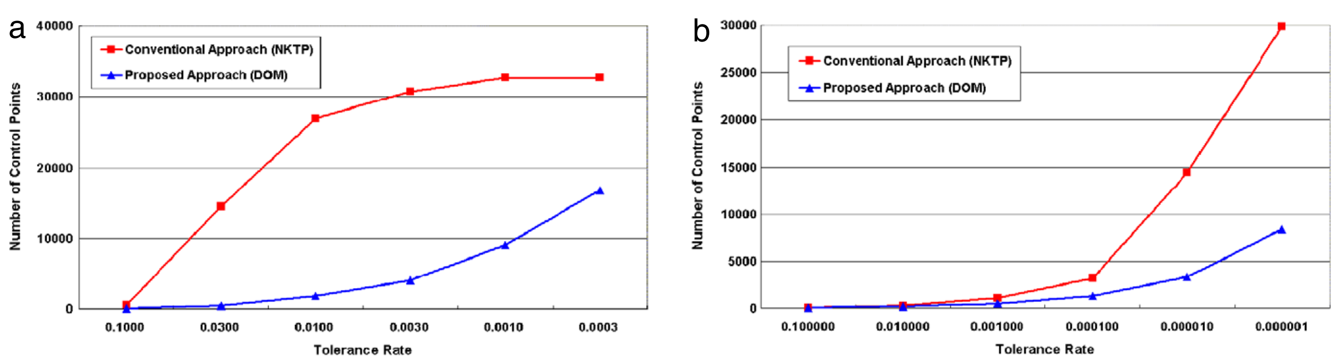

Fig. 6. Plot of the number of control points required for error-bounded B-spline surface fitting: (a) the sharply-peaked data; (b) the mask-shaped data.

rows and columns in grid points, so it usually creates different results before and after the swapping. Nonetheless, it keeps the same adaptiveness in the results.

For future works, we need to do more research on the following issues. First, it is very interesting to devise non-deterministic algorithms such as genetic algorithms [22] for the optimal selection of dominant columns and to investigate how far the knots obtained from dominant columns are away from optimal knots [7,8]. Secondly, the proposed approach can be extended to improve B-spline surface fitting either to serial contours or to scattered points with four boundaries. Thirdly, B-spline surfaces obtained by the proposed approach can be made more compact if they are converted to T-spline surfaces [23], so it is meaningful to investigate how to accomplish such a conversion efficiently. Lastly, the approach can be extended to construct a multivariate B-spline model from high dimensional structured grid data which comes from scientific experiments and measurements.

Acknowledgement

This study was supported in part by research funds from Chosun University, 2009.

References

[1] Hoschek J, Lasser D. Fundamentals of computer aided geometric design. A.K. Peters; 1993.
[2] Piegl L, Tiller W. The NURBS book. Springer-Verlag; 1995.
[3] Farin G. Curves and surfaces for CAGD: a practical guide. Morgan-Kaufmann; 2002.
[4] Weiss V, Andor L, Renner G, Varady T. Advanced surface fitting techniques. Computer Aided Geometric Design 2002;19(1):19–42.
[5] Rogers DF, Fog NG. Constrained B-spline curve and surface fitting. Computer-Aided Design 1989;21(10):641–8.
[6] Sarkar B, Menq CH. Smooth-surface approximation and reverse engineering. Computer-Aided Design 1991;23(9):623–8.
[7] Sarkar B, Menq CH. Parameter optimization in approximating curves and surfaces to measurement data. Computer Aided Geometric Design 1991;8(4): 267–90.

[8] Laurent-Gengoux P, Mekhilef M. Optimization of a NURBS representation. Computer-Aided Design 1993;25(11):699–710.
[9] Piegl L, Tiller W. Algorithm for approximate NURBS skinning. Computer-Aided Design 1996;28(9):699–706.
[10] 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.
[11] Park H. An approximate lofting approach for B-spline surface fitting to functional surfaces. International Journal of Advanced Manufacturing Technology 2001;18(7):474–82.
[12] Piegl L, Tiller W. Surface approximation to scanned data. The Visual Computer 2000;16(7):386–95.
[13] Park H, Lee JH. B-spline curve fitting based on adaptive curve refinement using dominant points. Computer-Aided Design 2007;39(6):439–51.
[14] Hoschek J. Intrinsic parameterization for approximation. Computer Aided Geometric Design 1988;5(1):27–31.
[15] Boehm W, Prautzch H. Numerical methods. Massachusetts: A.K. Peters; 1993.
[16] Hamann B, Chen JL. Data point selection for piecewise linear curve approximation. Computer Aided Geometric Design 1994;11(3):289–301.
[17] Liu GH, Wong YS, Zhang YF, Loh HT. Adaptive fairing of digitized data with discrete curvature. Computer-Aided Design 2002;34(4):309–20.
[18] Park H. An error-bounded approximate method for representing planar curves in B-splines. Computer Aided Geometric Design 2004;21(5):479–97.
[19] Schmit FJM, Barsky BA, Du WH. An adaptive subdivision method for surface-fitting from sampled data. Computer Graphics 1986;20(4):179–88.
[20] Forsey DR, Bartels RH. Surface fitting with hierarchical splines. ACT Transactions on Graphics 1995;14(2):134–61.
[21] Zheng J, Wang Y, Seah HS. Adaptive T-spline surface fitting to z-map models. In: Proceedings of graphite. 2005. p. 405–11.
[22] 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.
[23] Wang Y, Zheng J, Seah HS. Conversion between T-splines and hierarchical B-splines. In: Proceedings of computer graphics and imaging. 2005. p. 8–13.