Computer-Aided Design 44 (2012) 1269–1276

Contents lists available at SciVerse ScienceDirect

Computer-Aided Design

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

Approximate *T*-spline surface skinning

Xunnian Yang [a,*], Jianmin Zheng [b]

[a] *Department of Mathematics, Zhejiang University, China*

[b] *School of Computer Engineering, Nanyang Technological University, Singapore*

ARTICLE INFO

Article history:
Received 29 November 2011
Accepted 9 July 2012

Keywords:
T-spline surfaces
B-spline surfaces
Surface skinning
Curve interpolation

ABSTRACT

This paper considers the problem of constructing a smooth surface to fit rows of data points. A special class of *T*-spline surfaces is examined, which is characterized to have a global knot vector in one parameter direction and individual knot vectors from row to row in the other parameter direction. These *T*-spline surfaces are suitable for lofted surface interpolation or approximation. A skinning algorithm using these *T*-spline surfaces is proposed, which does not require the knot compatibility of sectional curves. The algorithm consists of three main steps: generating sectional curves by interpolating data points of each row by a *B*-spline curve; computing the control curves of a skinning surface that interpolates the sectional curves; and approximating each control curve by a *B*-spline curve with fewer knots, which results in a *T*-spline surface. Compared with conventional *B*-spline surface skinning, the proposed *T*-spline surface skinning has two advantages. First, the sectional curves and the control curves of a *T*-spline surface can be constructed independently. Second, the generated *T*-spline skinning surface usually has much fewer control points than a lofted *B*-spline surface that fits the data points with the same error bound. Experimental examples have demonstrated the effectiveness of the proposed algorithm.

1. Introduction

Surface skinning or lofting is a powerful method for surface generation from a set of sectional curves. It can be used for surface design based on a set of input curves [1–5], and it can be used to reconstruct smooth surfaces from rows of scanned data too [6–8]. Surface skinning for rows of data points often consists of two main steps, fitting a set of curves to rows of data points and generating a surface that passes through the sectional curves exactly or approximately.

In CAD, NURBS surfaces have become an industry standard for representing freeform shapes and it is a common process to fit lofted NURBS or *B*-spline surfaces to sectional curves or sampled data points. Usually, the sectional curves represented by *B*-splines have independent knot vectors. These knot vectors should be made compatible to permit a passing *B*-spline surface. However, a *B*-spline surface that passes through the sectional curves may have explosive knots and thus a huge number of control points as well by naively merging all knot vectors of the sectional curves to have a common knot vector. How to reduce the number of knots in the common knot vector while assuring the quality of sectional curves is a challenging problem for *B*-spline surface skinning [9–11].

As a generalization of *B*-spline surfaces, a *T*-spline surface permits *T*-junctions in its control mesh [12], which enables local refinement of knots or control points of the surface [13]. These properties make *T*-splines very flexible in representing complex surface shapes with much fewer control points than *B*-spline surfaces. To approximate dense regular grid data or even irregular surface meshes, a *T*-spline surface can be obtained by least squares fitting with adaptive knot insertion [14,15] or by a hierarchical subdivision of the parameter domain [16].

In this paper we propose to use *T*-splines for surface skinning to overcome the knot compatibility problem. We examine a special class of *T*-spline surfaces that can be regarded as the blending of some control curves using *B*-spline basis functions. The control curves themselves are also *B*-spline curves parameterized by another parameter. Differently from tensor-product *B*-spline surfaces of which the control curves are defined on a common knot vector, the control curves of the *T*-spline surfaces may have independent knot vectors. These *T*-spline surfaces have many advantages of *B*-spline surfaces. For example, evaluation of points or derivatives on a *T*-spline surface can be reduced to point or derivative evaluation on *B*-spline curves. Meanwhile, they also have flexibility in representing complex surfaces with as few control points as possible, especially for the approximation of rows of sampled points or sectional curves with independent knot vectors. A *T*-spline surface skinning algorithm is then proposed, which does not require compatibility of sectional curves. Specifically, for each row of ordered data points, the algorithm

* Corresponding author. Tel.: +86 571 87951609; fax: +86 571 87951428.
E-mail addresses: yxn@zju.edu.cn (X. Yang), ASJMZheng@ntu.edu.sg (J. Zheng).

0010-4485/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2012.07.003

first interpolates the data points by an independent B-spline curve. Then the B-spline curves are used as sectional curves of a parametric surface and the control curves of the surface are computed as linear combinations of the sectional curves. Finally each control curve is approximated by a B-spline curve with fewer knots. As a result, the lofted T-spline surface has much fewer control points than its B-spline counterpart.

The paper is structured as follows. Section 2 briefly reviews B-spline curve interpolation and Section 3 describes a class of T-spline surfaces, which will be used for surface skinning. Then in Section 4, we present an algorithm for constructing a T-spline surface that approximates rows of data points within a given tolerance bound. Experimental examples are provided in Section 5, which demonstrate the effectiveness of the proposed algorithm. Section 6 concludes the paper.

2. Review of B-spline curve interpolation

A B-spline curve of order k (or degree $k-1$) can be represented by

$$\mathbf{C}(u) = \sum_{i=0}^{I} \mathbf{X}_i N_{i,k}(u), \tag{1}$$

where $\mathbf{X}_i$ are the control points of the curve and $N_{i,k}(u)$ are the B-spline basis functions defined on a given knot vector $\mathbf{u} = \{u_0, u_1, \ldots, u_{I+k}\}$. The knots $\{u_i\}$ are non-decreasing and no k knots are the same except for at two ends. For any $u \in [u_{k-1}, u_{I+1}]$, $N_{i,k}(u)$ can be recursively evaluated by

$$N_{i,1}(u) = \begin{cases} 1, & \text{if } u \in [u_i, u_{i+1}]; \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,k}(u) = \frac{u - u_i}{u_{i+k-1} - u_i} N_{i,k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} N_{i+1,k-1}(u)$$

with $0/0 = 0$ if $u_i = u_{i+1}$.

Assume $\mathbf{P}_i$, $i = 0, 1, \ldots, K$, are a set of ordered points, $\mathbf{d}_0$ and $\mathbf{d}_K$ are two given boundary derivatives. If the knots and nodes are also given in advance, a cubic B-spline curve can be uniquely determined to interpolate the input data [17].

When the interpolating curves are to be used as sectional curves of an interpolating surface, the chord length parametrization of the input points can help to reduce the distortion of interpolating surfaces [7]. Let $d_0 = 0$, $d_i = d_{i-1} + |\mathbf{P}_i - \mathbf{P}_{i-1}|$ for $i = 1, 2, \ldots, K$, and $u_i = d_i/d_K$ for $i = 0, 1, \ldots, K$. An interpolating B-spline curve can be represented by $\mathbf{C}(u) = \sum_{i=0}^{K+2} \mathbf{X}_i N_{i,4}(u)$ with knot vector $\mathbf{u} = \{u_0, u_0, u_0, u_0, u_1, u_2, \ldots, u_{K-1}, u_K, u_K, u_K, u_K\}$. The control points $\mathbf{X}_i$ for the curve are determined by the following equations

$$\mathbf{C}(u_l) = \sum_{i=0}^{K+2} \mathbf{X}_i N_{i,4}(u_l) = \mathbf{P}_l, \quad l = 0, 1, \ldots, K \tag{2}$$

$$\mathbf{C}'(u_0) = \frac{3}{u_1 - u_0}(\mathbf{X}_1 - \mathbf{X}_0) = \mathbf{d}_0 \tag{3}$$

$$\mathbf{C}'(u_K) = \frac{3}{u_K - u_{K-1}}(\mathbf{X}_{K+2} - \mathbf{X}_{K+1}) = \mathbf{d}_K. \tag{4}$$

It is clear that $\mathbf{X}_0 = \mathbf{P}_0$ and $\mathbf{X}_{K+2} = \mathbf{P}_K$. The remaining control points can be solved from the following linear system (see Box I).

It is easily verified that the coefficient matrix of the above linear system is diagonally dominated, and the system can be solved efficiently by numerical algorithms such as LU decomposition [17].

When only sampled points are known, the end derivatives $\mathbf{d}_0$ and $\mathbf{d}_K$ can be estimated by two interpolating parabolic curves to the first three or the last three point data, respectively. It yields

$$\mathbf{d}_0 = \gamma_0 \mathbf{P}_0 + \gamma_1 \mathbf{P}_1 + \gamma_2 \mathbf{P}_2$$
$$\mathbf{d}_K = \bar{\gamma}_0 \mathbf{P}_K + \bar{\gamma}_1 \mathbf{P}_{K-1} + \bar{\gamma}_2 \mathbf{P}_{K-2}$$

where $\gamma_0 = -\left(\frac{1}{u_1 - u_0} + \frac{1}{u_2 - u_0}\right)$, $\gamma_1 = \frac{1}{u_1 - u_0} + \frac{1}{u_2 - u_1}$, $\gamma_2 = \frac{1}{u_2 - u_0} - \frac{1}{u_2 - u_1}$ and $\bar{\gamma}_0 = \frac{1}{u_K - u_{K-1}} + \frac{1}{u_K - u_{K-2}}$, $\bar{\gamma}_1 = -\left(\frac{1}{u_K - u_{K-1}} + \frac{1}{u_{K-1} - u_{K-2}}\right)$, $\bar{\gamma}_2 = -\left(\frac{1}{u_K - u_{K-2}} - \frac{1}{u_{K-1} - u_{K-2}}\right)$.

Besides the parabolic ends stated above, the natural ends for cubic B-spline curve interpolation are also a popular choice, which assumes that the second order derivatives of an interpolating curve vanish at two ends, i.e., $\mathbf{C}''(u_0) = 0$ and $\mathbf{C}''(u_K) = 0$. Eqs. (3) and (4) should then be replaced by the following two equations

$$\frac{2\Delta_0 + \Delta_1}{\Delta_0 + \Delta_1}\mathbf{X}_1 - \frac{\Delta_0}{\Delta_0 + \Delta_1}\mathbf{X}_2 = \mathbf{X}_0 \tag{5}$$

$$\frac{-\bar{\Delta}_0}{\bar{\Delta}_0 + \bar{\Delta}_1}\mathbf{X}_K + \frac{2\bar{\Delta}_0 + \bar{\Delta}_1}{\bar{\Delta}_0 + \bar{\Delta}_1}\mathbf{X}_{K+1} = \mathbf{X}_{K+2} \tag{6}$$

where $\Delta_0 = u_1 - u_0$, $\Delta_1 = u_2 - u_1$, $\bar{\Delta}_0 = u_K - u_{K-1}$ and $\bar{\Delta}_1 = u_{K-1} - u_{K-2}$. The coefficient matrix of the linear system consisting of Eqs. (2), (5) and (6) is also a diagonal-dominated matrix, and the system can be solved in the same way as the system with given boundary derivatives.

3. T-spline surfaces with control curves

In this section we describe a special class of T-spline surfaces that are defined by the following equation:

$$\mathbf{S}(u, v) = \sum_{j=0}^{J} \sum_{i=0}^{I_j} \mathbf{V}_{ij} N_{i,k_1}^{(j)}(u) N_{j,k_2}(v), \tag{7}$$

where $\mathbf{V}_{ij}$ are the control points and $N_{i,k_1}^{(j)}(u)N_{j,k_2}(v)$ are the blending functions. In particular, $N_{i,k_1}^{(j)}(u)$ are the B-spline basis functions of order k_1 defined on knot vectors $\mathbf{u}_j = \{u_0^{(j)}, u_1^{(j)}, \ldots, u_{I_j+k_1}^{(j)}\}$ for $j = 0, 1, \ldots, J$, and $N_{j,k_2}(v)$ are the B-spline basis functions of order k_2 defined on knot vector $\mathbf{v} = \{v_0, v_1, \ldots, v_{J+k_2}\}$. The knot vectors $\mathbf{u}_j$ also satisfy the end constraints, $u_{k_1-1}^{(0)} = u_{k_1-1}^{(1)} = \cdots = u_{k_1-1}^{(J)} = u_a$ and $u_{I_0+1}^{(0)} = u_{I_1+1}^{(1)} = \cdots = u_{I_J+1}^{(J)} = u_b$.

The equation of the T-spline surface by Eq. (7) can be reformulated as

$$\mathbf{S}(u, v) = \sum_{j=0}^{J} \mathbf{Q}_j(u) N_{j,k_2}(v), \quad (u, v) \in [u_a, u_b] \times [v_{k_2-1}, v_{J+1}] \tag{8}$$

where $\mathbf{Q}_j(u) = \sum_{i=0}^{I_j} \mathbf{V}_{ij} N_{i,k_1}^{(j)}(u)$ are a set of B-spline curves. These B-spline curves are called the *control curves* for the surface, just like the control points for B-spline curves. Such T-splines are called *T-spline surfaces with control curves*.

Except for the end constraints, the knot vectors of the control curves of surface $\mathbf{S}(u, v)$ basically are independent. Then a T-spline surface with control curves is characterized to have a global knot vector $\mathbf{v}$ and a set of individual knot vectors $\mathbf{u}_j$. When all the individual knot vectors happen to be the same, i.e., $\mathbf{u}_0 = \mathbf{u}_1 = \cdots = \mathbf{u}_J$, the T-spline surface of (7) reduces to a B-spline surface.

Proposition. *The blending functions of a T-spline surface with control curves are linearly independent and form a partition of unity.*

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ N_{1,4}(u_1) & N_{2,4}(u_1) & N_{3,4}(u_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & N_{K-1,4}(u_{K-1}) & N_{K,4}(u_{K-1}) & N_{K+1,4}(u_{K-1}) \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_K \\ \mathbf{X}_{K+1} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_0 + \dfrac{u_1 - u_0}{3}\mathbf{d}_0 \\ \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_{K-1} \\ \mathbf{P}_K - \dfrac{u_K - u_{K-1}}{3}\mathbf{d}_K \end{pmatrix}.$$

Box I.

Proof. Assume the T-spline surface is defined by Eq. (7). To prove the blending functions are linearly independent, we assume $\sum_{j=0}^{J}\sum_{i=0}^{I_j} c_{ij}N_{i,k_1}^{(j)}(u)N_{j,k_2}(v) \equiv 0$ with some constants c_{ij}. Let $c_j(u) = \sum_{i=0}^{I_j} c_{ij}N_{i,k_1}^{(j)}(u)$. The identity becomes $\sum_{j=0}^{J} c_j(u)N_{j,k_2}(v) \equiv 0$, which gives $c_j(u) \equiv 0$ for $j = 0, 1, \ldots, J$ due to the linear independence of B-spline basis functions $N_{j,k_2(v)}, j = 0, 1, \ldots, J$. Similarly, since $N_{i,k_1}^{(j)}(u); i = 0, 1, \ldots, I_j$ are B-spline basis functions, from $c_j(u) = \sum_{i=0}^{I_j} c_{ij}N_{i,k_1}^{(j)}(u) \equiv 0$, we conclude that $c_{ij} = 0$ for $i = 0, 1, \ldots, I_j$. Therefore $N_{i,k_1}^{(j)}(u)N_{j,k_2}(v)$ are linearly independent.

Next, since B-spline basis functions form a partition of unity, $\sum_{i=0}^{I_j} N_{i,k_1}^{(j)}(u) \equiv 1$ and thus $\sum_{j=0}^{J}\sum_{i=0}^{I_j} N_{i,k_1}^{(j)}(u)N_{j,k_2}(v) \equiv \sum_{j=0}^{J} N_{j,k_2}(v) \equiv 1$. □

While T-spline surfaces with control curves have flexibility in knot vectors and numbers of control points of the control curves, they can also be evaluated as simply as B-spline surfaces. In fact, the evaluation of points or partial derivatives of a T-spline surface can be reduced to the evaluation of points or derivatives of B-spline curves. For example, the partial derivatives of the surface can be obtained as

$$\frac{\partial \mathbf{S}(u, v)}{\partial u} = \sum_{j=0}^{J} \mathbf{Q}_j'(u)N_{j,k_2}(v)$$

$$\frac{\partial \mathbf{S}(u, v)}{\partial v} = \frac{d}{dv}\sum_{j=0}^{J} \mathbf{Q}_j^u N_{j,k_2}(v)$$

where $\mathbf{Q}_j'(u)$ is the derivative of $\mathbf{Q}_j(u)$ with respect to parameter u and $\mathbf{Q}_j^u = \mathbf{Q}_j(u)$.

From Eq. (8) we can see that the insertion or deletion of a u knot can be implemented completely locally while only the insertion or deletion of a v knot should be implemented across the whole parameter domain. The local refinement of u knots permits construction of control curves with adaptive knot vectors for a T-spline surface. This property makes T-spline surfaces with control curves an appropriate tool for lofted surface interpolation or approximation.

4. Approximate T-spline surface skinning

The problem we consider here can be stated as follows: given a set of input points $\{\mathbf{P}_{ij} : 0 \le i \le m_j; 0 \le j \le n\}$ that are arranged in $n + 1$ rows and points in each row are also arranged in order, construct a T-spline surface $\widetilde{\mathbf{S}}(u, v) = \sum_{j=0}^{n+2} \widetilde{\mathbf{Q}}_j(u)N_{j,k_2}(v)$ approximating the points within a prescribed tolerance, where $\widetilde{\mathbf{Q}}_j(u) = \sum_{i=0}^{I_j} \mathbf{V}_{ij}N_{i,4}^{(j)}(u)$.

Following the fashion of B-spline surface skinning, we construct an approximating T-spline surface by performing a series of B-spline curve interpolation. Basically, the T-spline surface skinning procedure consists of three steps:

(1) Interpolate data points of each row by a B-spline curve $\mathbf{R}_j(u)$, which results in sectional curves.
(2) Compute the control curves $\mathbf{Q}_j(u)$ of an interpolating surface $\mathbf{S}(u, v)$, which are linear combinations of the sectional curves.
(3) Approximate each control curve $\mathbf{Q}_j(u)$ by $\widetilde{\mathbf{Q}}_j(u)$, a B-spline curve with fewer knots, which results in the desired T-spline surface $\widetilde{\mathbf{S}}(u, v)$.

The first step can be accomplished using the algorithms explained in Section 2. Particularly, for the jth row of data points $\mathbf{P}_{ij} : 0 \le i \le m_j$, we normalize the chord length parameters and obtain the knot vector $\mathbf{s}_j = \{0, 0, 0, 0, u_1^{(j)}, u_2^{(j)}, \ldots, u_{m_j-1}^{(j)}, 1, 1, 1, 1\}$ with $u_0^{(j)} = 0$ and $u_{m_j}^{(j)} = 1$. A cubic interpolating B-spline curve $\mathbf{R}_j(u)$ is constructed, which satisfies $\mathbf{R}_j(u_i^{(j)}) = \mathbf{P}_{ij}$ for $0 \le i \le m_j$. After all the interpolating curves are generated, they will be used as sectional curves for surface skinning process.

The second and third steps are elaborated below.

4.1. Computation of control curves for an interpolating surface $\mathbf{S}(u, v)$

To determine an interpolating surface $\mathbf{S}(u, v)$, we choose a knot vector for parameter v. Let $v_0 = 0$, $v_j = v_{j-1} + |\mathbf{P}_{0,j} - \mathbf{P}_{0,j-1}| + |\mathbf{P}_{m_j,j} - \mathbf{P}_{m_{j-1},j-1}|$ for $j = 1, 2, \ldots, n$. A set of B-spline basis functions $N_{j,4}(v)$ are defined on knot vector $\mathbf{v} = \{v_0, v_0, v_0, v_0, v_1, v_2, \ldots, v_{n-1}, v_n, v_n, v_n, v_n\}$.

Let $\mathbf{S}(u, v) = \sum_{j=0}^{n+2} \mathbf{Q}_j(u)N_{j,4}(v)$. We want to find the control curves $\mathbf{Q}_j(u)$, which satisfy the following interpolation conditions:

$$\mathbf{S}(u, v_l) = \sum_{j=0}^{n+2} \mathbf{Q}_j(u)N_{j,4}(v_l) = \mathbf{R}_l(u), \quad l = 0, 1, \ldots, n \tag{9}$$

$$\left.\frac{\partial^2 \mathbf{S}(u, v)}{\partial v^2}\right|_{v=v_0} = 0 \tag{10}$$

$$\left.\frac{\partial^2 \mathbf{S}(u, v)}{\partial v^2}\right|_{v=v_n} = 0. \tag{11}$$

Here we use the natural end conditions in the v direction. The other end conditions can be analyzed in a similar way. For notational simplicity, we may just use $\mathbf{Q}_j$ for $\mathbf{Q}_j(u)$ and use $\mathbf{R}_l$ for $\mathbf{R}_l(u)$, without causing ambiguity.

Similar to Eqs. (2)–(4) and Eqs. (9)–(11) can be represented in matrix form too. In fact, we have

$$\mathbf{M}_{(n+3)\times(n+3)} \begin{pmatrix} \mathbf{Q}_0 \\ \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \vdots \\ \mathbf{Q}_n \\ \mathbf{Q}_{n+1} \\ \mathbf{Q}_{n+2} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{n-1} \\ \mathbf{R}_n \\ \mathbf{R}_n \end{pmatrix} \tag{12}$$

where

$$\mathbf{M}^{-1}=\begin{pmatrix} p_{00} & p_{01} & q_{02} & \cdots & q_{0n} & p_{0,n+1} & p_{0,n+2}\\ p_{10} & p_{11} & q_{12} & \cdots & q_{1n} & p_{1,n+1} & p_{1,n+2}\\ p_{20} & p_{21} & q_{22} & \cdots & q_{2n} & p_{2,n+1} & p_{2,n+2}\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\ p_{n0} & p_{n1} & q_{n2} & \cdots & q_{nn} & p_{n,n+1} & p_{n,n+2}\\ p_{n+1,0} & p_{n+1,1} & q_{n+1,2} & \cdots & q_{n+1,n} & p_{n+1,n+1} & p_{n+1,n+2}\\ p_{n+2,0} & p_{n+2,1} & q_{n+2,2} & \cdots & q_{n+2,n} & p_{n+2,n+1} & p_{n+2,n+2} \end{pmatrix},$$

Box II.

$$\mathbf{M}=\begin{pmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0\\ a_1 & b_1 & c_1 & 0 & \cdots & 0 & 0\\ 0 & a_2 & b_2 & c_2 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\ 0 & 0 & \cdots & a_n & b_n & c_n & 0\\ 0 & 0 & \cdots & 0 & a_{n+1} & b_{n+1} & c_{n+1}\\ 0 & 0 & \cdots & 0 & 0 & a_{n+2} & b_{n+2} \end{pmatrix}.$$

The elements of matrix $\mathbf{M}$ are

$$b_0=b_{n+2}=1,\qquad c_0=a_{n+2}=0,$$
$$a_1=0,\qquad c_1=-\frac{v_1-v_0}{v_2-v_0},\qquad b_1=1-c_1,$$
$$a_j=N_{j-1,4}(v_{j-1}),\qquad b_j=N_{j,4}(v_{j-1}),$$
$$c_j=N_{j+1,4}(v_{j-1}),\quad j=2,3,\ldots,n$$
$$a_{n+1}=-\frac{v_n-v_{n-1}}{v_n-v_{n-2}},\qquad b_{n+1}=1-a_{n+1},\qquad c_{n+1}=0.$$

Because matrix $\mathbf{M}$ is diagonally dominated, the linear system (12) can be solved robustly.

Differently from curve interpolation by which a set of control points are solved, the solution to system (12) gives a set of control curves. To obtain the control curves exactly, we compute the inverse matrix of $\mathbf{M}$ first. See Appendix for the computation of the inverse matrix of $\mathbf{M}$. Assuming (see Box II)

we let

$$\mathbf{C}=\begin{pmatrix} q_{01} & q_{02} & \cdots & q_{0n} & q_{0,n+1}\\ q_{11} & q_{12} & \cdots & q_{1n} & q_{1,n+1}\\ q_{21} & q_{22} & \cdots & q_{2n} & q_{2,n+1}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ q_{n1} & q_{n2} & \cdots & q_{nn} & q_{n,n+1}\\ q_{n+1,1} & q_{n+1,2} & \cdots & q_{n+1,n} & q_{n+1,n+1}\\ q_{n+2,1} & q_{n+2,2} & \cdots & q_{n+2,n} & q_{n+2,n+1} \end{pmatrix},$$

where $q_{i1}=p_{i0}+p_{i1}$ and $q_{i,n+1}=p_{i,n+1}+p_{i,n+2}$ for $i=0,1,\ldots,n+2$.

From Eq. (12), the control curves $\mathbf{Q}_j(u)$ can be obtained as

$$\begin{pmatrix} \mathbf{Q}_0\\ \mathbf{Q}_1\\ \mathbf{Q}_2\\ \vdots\\ \mathbf{Q}_n\\ \mathbf{Q}_{n+1}\\ \mathbf{Q}_{n+2} \end{pmatrix}=\mathbf{M}^{-1}\begin{pmatrix} \mathbf{R}_0\\ \mathbf{R}_0\\ \mathbf{R}_1\\ \vdots\\ \mathbf{R}_{n-1}\\ \mathbf{R}_n\\ \mathbf{R}_n \end{pmatrix}=\mathbf{C}\begin{pmatrix} \mathbf{R}_0\\ \mathbf{R}_1\\ \vdots\\ \mathbf{R}_{n-1}\\ \mathbf{R}_n \end{pmatrix}. \tag{13}$$

Thus, the control curves obtained by Eq. (13) can be represented by

$$\mathbf{Q}_j(u)=\sum_{l=0}^{n} q_{j,l+1}\mathbf{R}_l(u),\quad j=0,1,\ldots,n+2. \tag{14}$$

4.2. Computation of control curves for approximate T-spline skinning surface $\widetilde{\mathbf{S}}(u,v)$

Note that each $\mathbf{Q}_j(u)$ obtained by (14) is actually a cubic B-spline curve defined on knot vector $\mathbf{s}_{all}=\cup_{l=0}^{n}\mathbf{s}_l$ by knot insertion. This is because $\mathbf{R}_l(u)$ are a set of cubic B-spline curves defined on knot vector $\mathbf{s}_l=\{0,0,0,0,u_1^{(l)},u_2^{(l)},\ldots,u_{m_l-1}^{(l)},1,1,1,1\}$. Thus, the skinning surface with control curves $\mathbf{Q}_j(u)$ is just an interpolating B-spline surface. However, the unified knot vector may have a huge number of knots and the number of control points of curve $\mathbf{Q}_j(u)$ may be very large too. To obtain a skinning surface with fewer control points, we here present a method to replace $\mathbf{Q}_j(u)$ by an approximating curve $\widetilde{\mathbf{Q}}_j(u)$ that has an independent knot vector with much fewer knots.

Recall that $\mathbf{S}(u_i^{(l)},v_l)=\sum_{j=0}^{n+2}\mathbf{Q}_j(u_i^{(l)})N_{j,k_2}(v_l)=\mathbf{P}_{il}$ for $0\le i\le m_l$, $0\le l\le n$. For a given tolerance ε, if $|\widetilde{\mathbf{Q}}_j(u_i^{(l)})-\mathbf{Q}_j(u_i^{(l)})|<\varepsilon$ hold for all knots $u_i^{(l)}\in\mathbf{S}_{all}$, the distance from any point $\mathbf{P}_{il}$ to the approximate T-spline surface with control curves $\widetilde{\mathbf{Q}}_j(u)$, which is defined by $\widetilde{\mathbf{S}}(u,v)=\sum_{j=0}^{n+2}\widetilde{\mathbf{Q}}_j(u)N_{j,4}(v)$, is less than ε too. Therefore we only need to construct approximate control curves $\widetilde{\mathbf{Q}}_j(u)$ such that the distance between $\widetilde{\mathbf{Q}}_j(u)$ and $\mathbf{Q}_j(u)$ at the knots is less than ε.

To construct the approximate control curve $\widetilde{\mathbf{Q}}_j(u)$, we first choose knots from $\mathbf{s}_{all}$ to form a selected knot vector $\bar{\mathbf{u}}_j$. Second, we start with a subset of $\bar{\mathbf{u}}_j$ as the current knot vector and find a cubic B-spline curve $\widetilde{\mathbf{Q}}_j(u)$ that interpolates the points sampled on the curve $\mathbf{Q}_j(u)$ at the knots in the current knot vector. Third, we check the error $|\widetilde{\mathbf{Q}}_j(u_i^{(l)})-\mathbf{Q}_j(u_i^{(l)})|$ for all $u_i^{(l)}$ in the selected knot vector. If the maximum error is larger than ε, the corresponding knot will be inserted into the current knot vector and a new cubic B-spline curve will be constructed to interpolate the sampled points corresponding to the updated current knot vector. We continue this process until the maximum error is less than ε. Then the curve $\widetilde{\mathbf{Q}}_j(u)$ will be used as a control curve for the lofted T-spline surface. Fig. 1 illustrates the knot vectors of control curves of a T-spline surface as well as the knot vectors for those original sectional curves.

Considering the fact that farther sectional curves $\mathbf{R}_l(u)$ have less influence on a particular control curve $\mathbf{Q}_j(u)$, we only choose and check knots from sectional curves that are close to a control curve $\mathbf{Q}_j(u)$. To compute an approximate control curve $\widetilde{\mathbf{Q}}_j(u)$, the selected knot vector $\bar{\mathbf{u}}_j$ are chosen as follows

$$\bar{\mathbf{u}}_0=\bar{\mathbf{u}}_1=\mathbf{s}_0\cup\mathbf{s}_1$$
$$\bar{\mathbf{u}}_j=\mathbf{s}_{j-2}\cup\mathbf{s}_{j-1}\cup\mathbf{s}_j,\quad j=2,3,\ldots,n$$
$$\bar{\mathbf{u}}_{n+1}=\bar{\mathbf{u}}_{n+2}=\mathbf{s}_{n-1}\cup\mathbf{s}_n.$$

As the number of knots in $\bar{\mathbf{u}}_j$ is usually much smaller than the number of knots in $\mathbf{s}_{all}$, it will save a lot of computational cost for the computation of approximate control curve $\widetilde{\mathbf{Q}}_j(u)$ by choosing and checking knots in $\bar{\mathbf{u}}_j$ rather than in $\mathbf{s}_{all}$. We note that even though we have chosen and checked such small sets of candidate

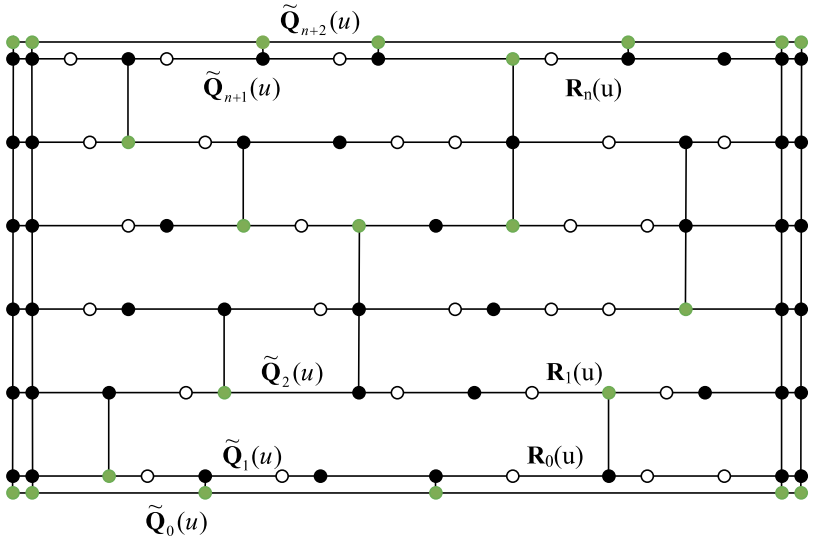

Fig. 1. Pre-image of a lofted T-spline surface. The hollow and black bullets represent knots of sectional curve $\mathbf{R}_j(u)$ and the solid bullets (black and green) represent knots of control curves $\widetilde{\mathbf{Q}}_j(u)$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

knots for the computation of control curves, the final T-spline surface still approximates the initial points within a prescribed tolerance in our experiments.

If the selected knot vector is $\bar{\mathbf{u}}_j = \{0, 0, 0, 0, \bar{u}_1^{(j)}, \bar{u}_2^{(j)}, \ldots, \bar{u}_{L_j}^{(j)}, 1, 1, 1, 1\}$, we let the initial current knot vector for the approximate control curve $\widetilde{\mathbf{Q}}_j(u)$ be $\mathbf{u}_j = \{0, 0, 0, 0, 1, 1, 1, 1\}$.

Now suppose a current knot vector is denoted by $\mathbf{u}_j = \{u_0^{(j)}, u_0^{(j)}, u_0^{(j)}, u_0^{(j)}, u_1^{(j)}, \ldots, u_{K-1}^{(j)}, u_K^{(j)}, u_K^{(j)}, u_K^{(j)}, u_K^{(j)}\}$ with $u_0^{(j)} = 0$ and $u_K^{(j)} = 1$. We sample points and boundary derivatives from the original control curve $\mathbf{Q}_j(u)$:

$$\mathbf{G}_{ij} = \mathbf{Q}_j(u_i^{(j)}) = \sum_{l=0}^{n} q_{j,l+1}\mathbf{R}_l(u_i^{(j)}), \quad i = 0, 1, \ldots, K$$

$$\mathbf{d}_0 = \mathbf{Q}_j'(u_0^{(j)}) = \sum_{l=0}^{n} q_{j,l+1}\mathbf{R}_l'(u_0^{(j)})$$

$$\mathbf{d}_K = \mathbf{Q}_j'(u_K^{(j)}) = \sum_{l=0}^{n} q_{j,l+1}\mathbf{R}_l'(u_K^{(j)}).$$

Using the algorithm described in Section 2, we obtain a B-spline curve $\widetilde{\mathbf{Q}}_j(u)$ by interpolating points $\mathbf{G}_{ij}$ and end derivatives $\mathbf{d}_0$, $\mathbf{d}_K$ with the current knot vector $\mathbf{u}_j$.

We compute the distances between curves $\mathbf{Q}_j(u)$ and $\widetilde{\mathbf{Q}}_j(u)$ at knots in the selected knot vector $\bar{\mathbf{u}}_j$. If

$$\max_{1\le i\le L_j} |\mathbf{Q}_j(\bar{u}_i^{(j)}) - \widetilde{\mathbf{Q}}_j(\bar{u}_i^{(j)})| < \varepsilon,$$

output $\widetilde{\mathbf{Q}}_j(u)$ as a control curve of the lofted T-spline surface. Otherwise, the knot corresponding to the maximum error is inserted into the current knot vector $\mathbf{u}_j$, and a new control curve is generated for the updated sampled points and current knot vector. This process is repeated until a control curve within the permitted error bound at all knots in the selected knot vector has been found. If all knots in the selected knot vector $\bar{\mathbf{u}}_j$ have been in the current knot vector, we have $\mathbf{u}_j = \bar{\mathbf{u}}_j$ and the maximum error is zero. This implies that the iteration process terminates in a finite number of steps.

To sum up, we outline the algorithm for approximate control curve computation below.

Algorithm 1. Approximate control curve computation

input: $\mathbf{R}_l(u)$, $q_{j,l+1}$, $l = 0, 1, \ldots, n$ and ε
output: a cubic B-spline curve $\widetilde{\mathbf{Q}}_j(u)$

1. Choose a selected knot vector $\bar{\mathbf{u}}_j$;
2. Initialize the current knot vector $\mathbf{u}_j = \{0, 0, 0, 0, 1, 1, 1, 1\}$;
3. Compute end derivatives and sampled points of $\mathbf{Q}_j(u)$ at all knots in $\mathbf{u}_j$;
4. Interpolate a cubic B-spline curve $\widetilde{\mathbf{Q}}_j(u)$ with knot vector $\mathbf{u}_j$ to the sampled data;
5. Check distances between pairs of points $\widetilde{\mathbf{Q}}_j(u)$ and $\mathbf{Q}_j(u)$ sampled at knots in $\bar{\mathbf{u}}_j$;
6. If the maximum error is greater than ε, refine knot vector $\mathbf{u}_j$ and go to step 3;
7. Output $\widetilde{\mathbf{Q}}_j(u)$ as a control curve.

5. Experimental examples

In this section we present several examples to show the usefulness and effectiveness of the proposed T-spline surface skinning. The comparison between the resulting T-spline surfaces and the lofted B-spline surfaces generated by the approximate B-spline surface skinning technique proposed by Piegl and Tiller in [3] is also given.

First, we present an example of dustpan shape modeling by T-spline surface skinning. To model the surface, a set of data points have been sampled on 11 sectional curves which are composed of smoothly jointed line segments and circular arcs; see Fig. 2(a). There are 222 sampled points in total and each row has at least 10 and at most 30 points. The interpolating B-spline curves with chord length parametrization are illustrated in Fig. 2(b). Fig. 2(c) shows the lofted T-spline surface with an error bound $\varepsilon = 10^{-5}$, where 1 is the diagonal diameter of the bounding box of the original sampled data. From the plot of iso-parameter curves in Fig. 2(d) we can see that curve interpolation using chord length parameters arouses no distortion for surface approximation. The control mesh of the lofted T-spline surface is depicted in Fig. 2(e). Because the skinning B-spline surface with the same error bound looks almost the same as the T-spline surface, we just plot the control mesh for the lofted B-spline surface in Fig. 2(f). The number of control points of one control curve and the total number of control points of a T-spline surface or a B-spline surface are given in Table 1.

Table 1
Numbers of control points of the skinning *B*-spline surfaces or the *T*-spline surfaces.

Example	Tolerance (ε)	*B*-spline control points		*T*-spline control points	
		One control line	Total	One control line (min/max)	Total
Dustpan	10^{-5}	45	585	4/38	334
Wave	0.5×10^{-3}	168	1,176	15/157	577
Screwdriver handle	10^{-4}	52	728	20/49	463
Head	0.5×10^{-3}	181	18,462	14/136	6311

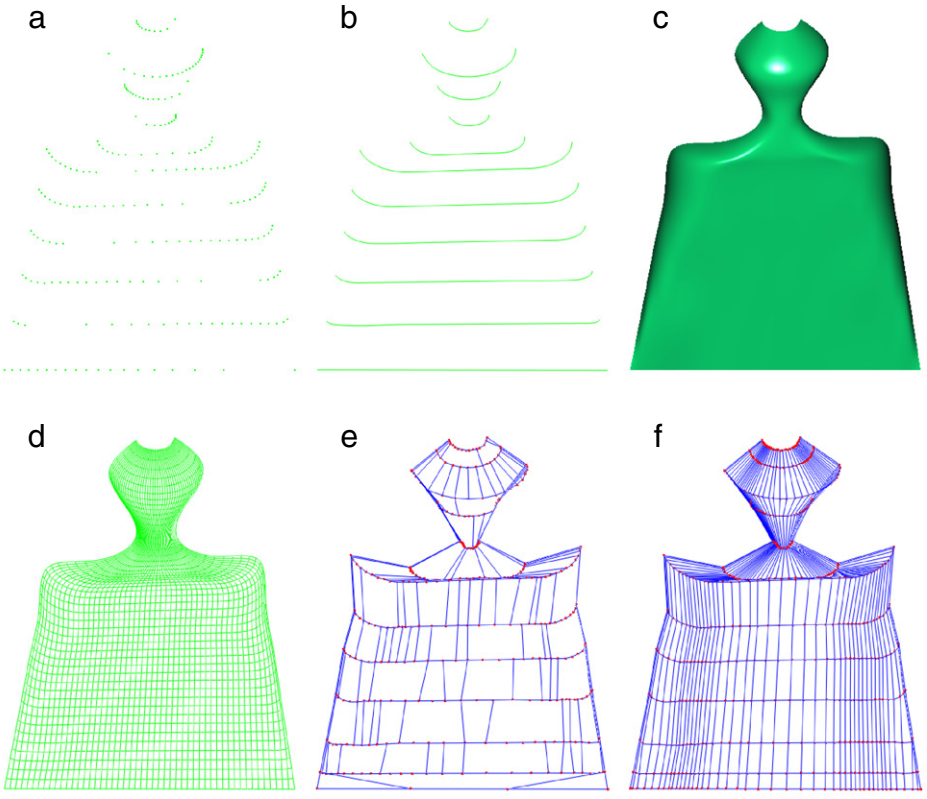

Fig. 2. Dustpan surface modeling by approximate surface skinning: (a) the sampled data; (b) the interpolating *B*-spline curves; (c) the lofted *T*-spline surface; (d) the iso-parameter curves of the *T*-spline surface; (e) the *T*-spline control mesh; (f) the *B*-spline control mesh.

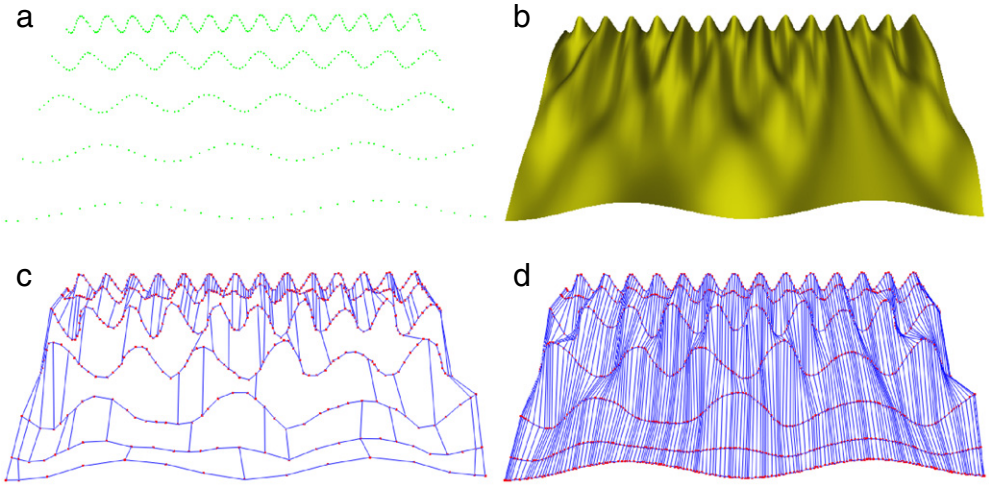

Fig. 3. Wave surface modeling by approximate *T*-spline surface skinning: (a) the sampled data; (b) the lofted *T*-spline surface; (c) the *T*-spline control mesh; (d) the *B*-spline control mesh.

Second, we model a wave shape surface using surface skinning techniques, too. The initial points are sampled from five wave shape curves that have different wave lengths. Due to the variances of the curve shapes, the points are sampled adaptively on each

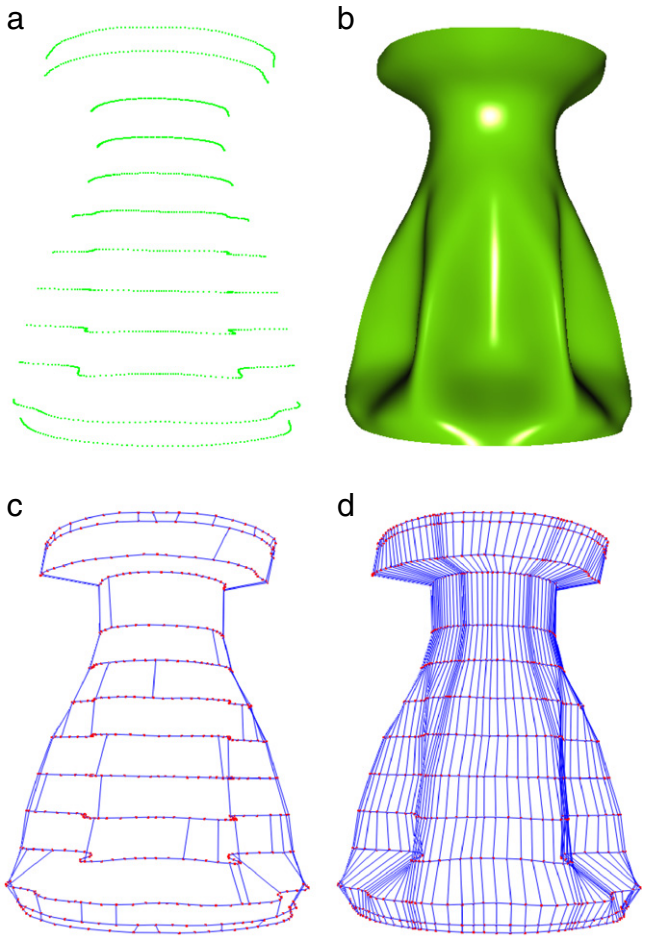

Fig. 4. Lofted T-spline surface approximation to a set of sampled data on a screwdriver's handle: (a) the sampled data; (b) the lofted T-spline surface; (c) the T-spline control mesh; (d) the B-spline control mesh.

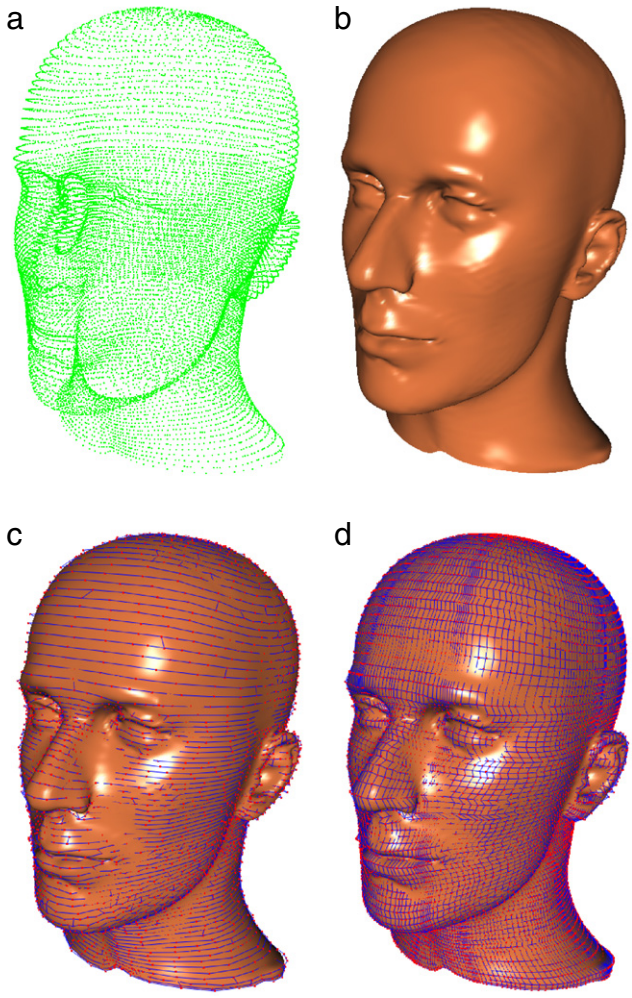

Fig. 5. Lofted T-spline surface approximation to a set of sampled data on a head model: (a) the sampled data; (b) the lofted T-spline surface; (c) the T-spline control mesh; (d) the lofted B-spline surface and its control mesh.

curve and there are totally 504 sampled points, as shown in Fig. 3(a). After interpolating each row of data points by a B-spline curve, a T-spline surface that approximates the sampled data within an error bound $\varepsilon = 0.5 \times 10^{-3}$ is constructed; see Fig. 3(b) for the T-spline surface. Fig. 3(c) shows the control mesh of the T-spline surface and Fig. 3(d) is the control mesh of the approximate skinning B-spline surface. Because the control curves of the T-spline surface are constructed with adaptively chosen knot vectors, every control curve of the T-spline surface has fewer control points than one control curve of the B-spline surface. Consequently, the total number of control points of the T-spline surface is less than half the number of control points of the skinning B-spline surface; see Table 1 for the results.

Third, we present an example of reconstructing the shape of a screwdriver handle by surface skinning. We have sampled 12 rows, totally 1224 points, from an original screwdriver model and each row contains 90–150 sampled points; see Fig. 4(a) for the sampled points. In the same way as the above two examples, a set of B-spline curves have been constructed independently to interpolate the original sampled data using chord length parametrization. By choosing a relative error bound as $\varepsilon = 10^{-4}$, we obtain the lofted T-spline surface shown in Fig. 4(b). From Table 1, we know that the approximate T-spline surface has much fewer control points than a B-spline surface; see Fig. 4(c) and (d) for the control mesh of the T-spline surface or the control mesh of the approximate B-spline surface, respectively.

Finally, we present an example of reconstructing a more complex shape by approximate T-spline surface skinning. The original model is a triangular mesh. A harmonic scalar field is computed on the mesh surface using the algorithm proposed in [18], and then 18,368 points are sampled on 100 iso-parameter curves of the field; see Fig. 5(a) for the sampled points. An approximate T-spline surface with control curves is obtained using a tolerance $\varepsilon = 0.5 \times 10^{-3}$, where 1 is the length of the diagonal diameter of the bounding box of the input points. Fig. 5(b) displays the T-spline surface. The control mesh of the T-spline surface has 6631 control points; see Fig. 5(c) for the control mesh of the T-spline surface. When the sampled data is fitted by a B-spline surface using the same error bound, the control mesh has about two times more control points than the T-spline surface; see Fig. 5(d) for the lofted B-spline surface and its control mesh.

From the above examples we can see that the shape of an approximate T-spline surface or a B-spline surface depends greatly on the shape or the quality of the input data. If the original points are sampled from a smooth surface or the sampled curves are smooth and sparse spaced, the resulting surfaces are usually smooth. On the other hand, if the points are sampled from some wavy curves or the points contain noise, the approximate skinning surfaces may have local undulations when the tolerance is chosen to be a small number. This suggests that one can construct smooth skinning surfaces using filtered points if the input data contain noise.

6. Conclusion

In this paper we propose to use a special class of T-spline surfaces for surface skinning. The T-spline surfaces in this class can have control curves with independent knot vectors offering flexibility in surface skinning and can be evaluated as simply as B-spline surfaces. A T-spline surface skinning algorithm has been presented, which generates a T-spline surface approximating the input rows of data points within a prescribed tolerance. The control curves of the T-spline surface are obtained by fitting B-spline curves with adaptively chosen knot vectors. The lofted T-spline surface with adaptively constructed control curves usually has much fewer control points than a lofted B-spline surface when they are approximating the same set of data points within the same tolerance bound. The algorithm can overcome the knot compatibility problem existing in conventional B-spline surface skinning.

Acknowledgments

We owe thanks to the anonymous referees for their helpful comments and suggestions. This work is supported by NNSF of China grant (60970077) and the ARC 9/09 Grant (MOE2008-T2-1-075) of Singapore.

Appendix. Inverse matrix computation

We now show how to compute the inverse matrix of coefficient matrix $\mathbf{M}$ of Eq. (12). Let $\beta_0 = b_0, \alpha_i = a_i/\beta_{i-1}$ and $\beta_i = b_i - \alpha_i c_{i-1}$ for $i = 1, 2, \ldots, n+2$. Let $\gamma_i = c_i/\beta_i$ for $i = 0, 1, \ldots, n+1$. Matrix $\mathbf{M}$ can then be decomposed into

$$\mathbf{M} = \mathbf{M}_\alpha \mathbf{M}_\beta \mathbf{M}_\gamma, \tag{A.1}$$

where

$$\mathbf{M}_\alpha = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \alpha_1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & \alpha_2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \alpha_{n+1} & 1 & 0 \\ 0 & 0 & \cdots & 0 & \alpha_{n+2} & 1 \end{pmatrix},$$

$\mathbf{M}_\beta = \mathtt{Diag}\{\beta_0, \beta_1, \ldots, \beta_{n+2}\}$ and

$$\mathbf{M}_\gamma = \begin{pmatrix} 1 & \gamma_0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \gamma_1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & \gamma_{n+1} \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}.$$

The inverse matrix of $\mathbf{M}_\beta$ is $\mathbf{M}_\beta^{-1} = \mathtt{Diag}\left\{\frac{1}{\beta_0}, \frac{1}{\beta_1}, \ldots, \frac{1}{\beta_{n+2}}\right\}$. The inverse matrices $\mathbf{M}_\alpha^{-1}$ and $\mathbf{M}_\gamma^{-1}$ are

$$\mathbf{M}_\alpha^{-1} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ g_{10} & 1 & 0 & \cdots & 0 & 0 \\ g_{20} & g_{21} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{n+1,0} & g_{n+1,1} & \cdots & g_{n+1,n} & 1 & 0 \\ g_{n+2,0} & g_{n+2,1} & \cdots & g_{n+2,n} & g_{n+2,n+1} & 1 \end{pmatrix}$$

with $g_{ij} = \Pi_{l=j+1}^{i}(-\alpha_l)$ and

$$\mathbf{M}_\gamma^{-1} = \begin{pmatrix} 1 & r_{01} & r_{02} & \cdots & r_{0,n+1} & r_{0,n+2} \\ 0 & 1 & r_{12} & \cdots & r_{1,n+1} & r_{1,n+2} \\ 0 & 0 & 1 & \cdots & r_{2,n+1} & r_{2,n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & r_{n+1,n+2} \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

with $r_{ij} = \Pi_{l=i}^{j-1}(-\gamma_l)$. Thus the inverse matrix of $\mathbf{M}$ is obtained as $\mathbf{M}^{-1} = \mathbf{M}_\gamma^{-1}\mathbf{M}_\beta^{-1}\mathbf{M}_\alpha^{-1}$.

References

[1] Woodward C. Cross-sectional design of B-spline surfaces. Computers and Graphics 1987;11(2):193–201.
[2] Woodward C. Skinning techniques for interactive B-spline surface interpolation. Computer-Aided Design 1988;20(8):441–51.
[3] Piegl LA, Tiller W. Surface skinning revisited. The Visual Computer 2002;18:273–83.
[4] Slabaugh G, Rossignac J, Whited B, Fang T, Unal G. 3D ball skinning using PDEs for generation of smooth tubular surfaces. Computer-Aided Design 2010;42:18–26.
[5] Piegl L, Tiller W. Algorithm for approximate NURBS skinning. Computer-Aided Design 1996;28(9):699–706.
[6] Park H, Kim K. Smooth surface approximation to serial cross-sections. Computer-Aided Design 1996;28(12):995–1005.
[7] Piegl LA, Tiller W. Surface approximation to scanned data. The Visual Computer 2000;16:386–95.
[8] Yano K, Harada K. Reconstruction of B-spline skinning surface from generalized cylinder mesh. The Visual Computer 2010;26:31–40.
[9] Piegl L, Tiller W. Reducing control points in surface interpolation. IEEE Computer Graphics and Applications 2000;20(5):70–4.
[10] Park H. Lofted B-spline surface interpolation by linearly constrained energy minimization. Computer-Aided Design 2003;35:1261–8.
[11] Wang W-K, Zhang H, Park H, Yong J-H, Paul J-C, Sun J-G. Reducing control points in lofted B-spline surface interpolation using common knot vector determination. Computer-Aided Design 2008;40:999–1008.
[12] Sederberg TW, Zheng J, Bakenov A, Nasri AH. T-splines and t-nurccs. ACM Transactions on Graphics (SIGGRAPH Proceedings) 2003;22(3):477–84.
[13] Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-spline simplification and local refinement. ACM Transactions on Graphics (SIGGRAPH Proceedings) 2004;23(3):276–83.
[14] Zheng J, Wang Y, Seah HS. Adaptive T-spline surface fitting to z-map models. Proceedings of GRAPHITE '05. 2005. pp. 405–411.
[15] Wang Y, Zheng J. Adaptive T-spline surface approximation of triangular meshes. in: Proceedings of the 6th international conference on information, communications & signal processing. Singapore. December 2007.
[16] Deng J, Chen F, Li X, Hu C, Tong W, Yang Z, Feng Y. Polynomial splines over hierarchical T-meshes. Graphical Models 2008;70:76–86.
[17] Farin G. Curves and surfaces for computer aided geometric design: a practical guide. 5th ed. Academic Press; 2002.
[18] Dong S, Kircher S, Garland M. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. Computer Aided Geometric Design 2005;22:392–423.