



Contents lists available at ScienceDirect

# Journal of Computational and Applied Mathematics

journal homepage: [www.elsevier.com/locate/cam](http://www.elsevier.com/locate/cam)

## Surface skinning using periodic $T$ -spline in semi-NURBS form

Yusha Li<sup>a</sup>, Wenyu Chen<sup>b,\*</sup>, Yiyu Cai<sup>c,d</sup>, Ahmad Nasri<sup>e</sup>, Jianmin Zheng<sup>a,d</sup><sup>a</sup> School of Computer Engineering, Nanyang Technological University, Singapore<sup>b</sup> Institute for Infocomm Research, A\*STAR, Singapore<sup>c</sup> School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore<sup>d</sup> Institute for Media Innovation, Nanyang Technological University, Singapore<sup>e</sup> ASHA Computer Graphics & Animation Lab, American University of Beirut, Lebanon

### ARTICLE INFO

#### Article history:

Received 2 July 2012

Received in revised form 31 May 2014

#### Keywords:

Periodic  $T$ -spline

Surface skinning

Local modification

Incompatible cross sections

### ABSTRACT

NURBS skinning is a powerful and effective process in Computer Aided Geometric Design (CAGD). It constructs a surface by interpolating a set of cross sectional NURBS curves. These curves however may not be compatible, i.e., they have different knot vectors. This incompatibility is conventionally solved by knot refinement bringing all curves to share the same knot vector, which leads to an explosion in the number of control points defining the skinned surface. Another disadvantage of NURBS skinning is the difficulty of local modification: adjusting one cross section may result in a global change of the surface. In this paper, periodic  $T$ -spline in semi-NURBS form is discussed. Surface skinning using such  $T$ -splines is able to handle closed cross sections, to support local modifications and to control smoothness along the cross sectional curves. We provide explicit formulae for constructing such  $T$ -spline skinned surfaces, which avoid solving a large system of equations. Experimental results and theoretical analysis confirm that our approach is better than NURBS skinning as it generates surfaces with fewer control points.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Skinning, Boolean, and trimming are essential operations in many modeling systems for geometric design and industrial manufacturing. Skinning or lofting is the process of constructing a surface that interpolates a series of cross sectional curves arranged in 3D space [1]. In reverse engineering, a set of rows of data points is initially sampled on an object where each row is fitted by a  $B$ -spline curve that is defined by its control polygon and knot vector. These data are taken as input to the skinning approach where a surface that either interpolates or approximates these curves is generated.

Currently, the representation of skinned surface is dominated by non-uniform rational  $B$ -spline surface (NURBS) with the cross sections as NURBS curves. Though NURBS surfaces are capable of compactly describing various shapes, skinned NURBS surfaces suffer from redundant points mainly due to the incompatibility of cross sections. Here the incompatibility of two cross sections refers to the differences in their numbers of control points and their knot vectors. The ideal scenario where all cross sections have the same number of control points and the same knot vector is far from practical. One approach solves this problem by knot insertion, a process that leads to an explosion of the number of control points representing the skinned surface. In general, if there are  $n$  cross sections with an average of  $m$  control points per curve, the total number of control points for the skinned surface is of  $O(mn^2)$  [2], which is unacceptable for a CAD model.

This incompatibility problem was addressed by many researchers. Piegl and Tiller [3] made use of the flexibility of the parameters and knots of the curves. They chose a candidate knot vector that contains enough knots with regard to the

\* Corresponding author. Tel.: +65 97230486.

E-mail addresses: [ciweiyu@gmail.com](mailto:ciweiyu@gmail.com), [chenw@i2r.a-star.edu.sg](mailto:chenw@i2r.a-star.edu.sg) (W. Chen).

largest row of data points, and checked whether the candidate knot vector can be used to interpolate other rows. If not, the candidate knot vector will be refined until all rows are interpolated. In their method, the performance strongly depends on the candidate knot vector.

Another rectification to this problem is by using approximation rather than interpolation. The approximation approach was investigated comprehensively [2,4–8]. Piegl and Tiller [5] proposed a method to reduce the number of control points in skinning by allowing the user to interpolate or approximate the initial cross sections using knot removal algorithm [9]. A significant number of control points can be eliminated under a given tolerance without sacrificing the surface quality too much. Piegl and Tiller presented the second approximation  $B$ -spline method in [2]. First, sample points from each cross section are obtained, and then approximated using a non-rational  $B$ -spline curve based on a candidate knot vector. After that, removable knots are eliminated if the corresponding error is small enough and new knots are inserted if the corresponding error exceeds the given tolerance. Similarly, Park et al. [6] proposed a method to refit the initial cross sections based on a common knot vector, and provided an algorithm to reduce the size of this vector. Besides, an energy function is adopted to stabilize the solution of curve fitting and to insure its smoothness. However, the performance is somewhat subjected to the sampling density. Another similar work was carried out by Park [7]. In their algorithm, parameters should be carefully chosen to avoid wiggles resulting from an improper knot vector. Typically, the approximation method involves an iteration procedure, which is memory ineffective and computationally expensive.

Another method to handle the incompatibility is by adopting other representations, such as subdivision surface. In Nasri et al.'s approach [10], a skinned Catmull–Clark subdivision surface is generated through incompatible cross sections of cubic  $B$ -spline curves. For each curve a polygonal complex is constructed. A polygonal complex is a mesh structure that under subdivision converges to a curve. Embodying such a configuration in the polyhedron (or control mesh) of a subdivision surface guarantees the interpolation of the corresponding curve by the limit surface. Then the set of complexes is connected to generate the final polyhedron defining the skinned surface which will automatically interpolate the cross sections. Their method can reduce the size of the mesh to  $O(mn)$ .

In addition to the incompatibility problem, NURBS skinning has other limitations. The control mesh of a NURBS skinned surface is usually obtained by solving a global linear system. Thus there does not exist any local modification property. Moreover, it is not easy to generate NURBS skinned surfaces with sharp feature, i.e. with  $C^0$  continuity across a curve section.

As introduced by Sederberg et al. in the two pioneering papers [11,12], the  $T$ -spline surface is a generalization of the NURBS surface with the advantage of local refinement in surface merging [13], surface fitting [14,15], and surface simplification [16,17]. Recently, Nasri et al. [18] proposed a local approach to the skinning problem using  $T$ -spline by solving a local system of equations rather than solving a global system of equations. Their approach generates a skinned surface that interpolates incompatible NURBS curves with fewer control vertices. However, their approach does not handle periodic curves which are needed in tubular surfaces in various applications. Normally, NURBS handles such curves by considering them as non-periodic with special treatment of knots at the end and the beginning. In  $T$ -spline skinning, the first knots, as well as the last knots, for all input cross sections should be the same, which is not the case for periodic  $T$ -spline skinning.

In this paper, we propose a method to address the issues mentioned above using periodic  $T$ -splines in semi-NURBS form, which are periodic  $T$ -splines [19] with particular properties making them suitable for surface skinning. The paper contributes in:

1. Using closed cross sections as input to construct a tubular shape,
2. Adopting a constrained periodic  $T$ -spline surface to interpolate incompatible cross sections,
3. Supporting local modifications on closed cross sections in surface skinning,
4. Controlling the smoothness of the skinned surface at each closed cross section, thus a cross sectional curve could be a sharp edge,
5. Deriving the intermediate cross sections using 4-point interpolatory subdivision scheme [20] instead of directly using the middle points used in [18] to reduce the wiggle effect (Section 4.2), and
6. Providing explicit formulae for periodic  $T$ -spline vertices to avoid solving any system of equations.

The paper is structured as follows. In Section 2, preliminaries for periodic  $T$ -spline are introduced and the advantages using periodic  $T$ -spline in semi-NURBS form for skinning are analyzed. Section 3 details the skinning scheme using the periodic  $T$ -spline in semi-NURBS form. Analysis of the scheme and comparison with NURBS skinning are presented in Section 4. After showing examples in Section 5, we will conclude our research in the last section.

## 2. Preliminaries

Periodic curves or surfaces refer to those curves and surfaces that are seamless, with no beginning or ending [21]. They are very useful in representing tubular surfaces closed in one or both directions. The latter is referred to as torus-like surfaces. This section introduces the basic ideas of periodic  $T$ -spline surfaces. For more details, the reader is referred to  $T$ -spline [11,12] and periodic  $T$ -spline [19].

### 2.1. Notations

Our problem could be formulated as follows:

Given: A set of  $n$  closed cubic NURBS curves

$$\mathbf{C}_q(s), \quad q \in \{0, 1, \dots, n - 1\},$$

and a set of values  $(f_q)$  indicating their corresponding desired type of cross-smoothness, i.e.  $C^0$  or  $C^2$ .

*Find:* A periodic  $T$ -spline  $\mathbf{S}(s, t)$  that interpolates the given curves  $(\mathbf{C}_q)$  with their corresponding cross-smoothness  $(f_q)$ .

Assume that each curve  $\mathbf{C}_q$  is defined by a set of control points  $\mathbf{P}_{i,q}$  and a knot vector  $\mathbf{K}_q = \{s_{i,q}\}$ . Each cross section will be considered as an iso-parametric curve corresponding to a certain parameter  $t_q$ . With each cross sectional curve, we also associate a value  $f_q$  to indicate the type of smoothness, i.e.  $C^0$  or  $C^2$ . For instance to interpolate a curve  $\mathbf{C}_q$  by the skinned surface  $\mathbf{S}(s, t)$  with a  $C^2$  cross-smoothness, we should have  $\mathbf{S}(s, t_q) = \mathbf{C}_q(s)$ , and  $f_q = C^2$ .

Our output should be a periodic  $T$ -spline surface  $\mathbf{S}(s, t)$  that interpolates the given curves  $(\mathbf{C}_q)$  with their corresponding cross-smoothness  $(\mathbf{f}_q)$ . Let us assume that the surface  $\mathbf{S}$  is defined by a  $T$ -mesh  $\mathbf{P}$  of control vertices  $\mathbf{P}_{i,q}$ , and two knot vectors:  $\mathbf{K}_s$  in  $s$  direction, and  $\mathbf{K}_t$  in  $t$  direction. The knot vector  $\mathbf{K}_s$  will be obtained from the union of the knot vectors of all given curves, whereas the knot vector  $\mathbf{K}_t$  can be considered as input or calculated using the cumulative chord-length method [22].

### 2.2. Periodic NURBS curves and surfaces

NURBS systems are primarily developed to handle open curves/surfaces. For periodic curves, the traditional approach, called *repeating approach*, for representing such curves consists of repeating the first few control vertices and knot intervals [23]. For a degree  $k$  B-spline curve, the first  $k$  control vertices should coincide with the last  $k$  ones, and the corresponding knot intervals should also be equal. This technique is adopted by most of the commercial CAD software. The advantage is to avoid treating closed curves as special cases. Fig. 1 shows an example of constructing a cubic periodic spline. In Fig. 1(b) the last three control vertices coincide with the first three ones.

Generally, a knot vector  $\mathbf{K} = [s_{-k}, \dots, s_{-1}, s_0, \dots, s_m, s_{m+1}, \dots, s_{m+k}]$  is periodic if it satisfies

$$s_{m+i} - s_{m+i-1} = s_i - s_{i-1}, \quad i = -k + 1, \dots, k.$$

The period of  $\mathbf{K}$  is then given by  $T = s_m - s_0$ . For  $m$  distinct control vertices  $(\mathbf{P}_0, \dots, \mathbf{P}_{m-1})$ , a periodic curve can be defined by

$$\mathbf{C}(s) = \sum_{i=0}^{m+k-1} \mathbf{P}_{i \bmod m} N_{i \bmod m}^k(s), \quad s \in [s_0, s_m], \tag{1}$$

where ‘ $i \bmod m$ ’ is the abbreviation of ‘ $i$  modulo  $m$ ’, with ‘ $i \bmod m$ ’  $\in \{0, 1, \dots, m - 1\}$ .

The *repeating approach* can be extended to NURBS surfaces. The control vertices of first few columns should be repeated as well as their corresponding knot intervals. Fig. 2(a) illustrates this technique where vertices between columns  $c_0$  and  $c_5$  are connected by the yellow edges, which correspond to the edges between  $c_5$  and  $c_6$  in the preimage Fig. 2(b).

### 2.3. Periodic T-spline surfaces

The repeating approach works well for periodic NURBS surfaces since the repeated knots are the same for the different rows of control vertices. However, this cannot be applied in the case of periodic  $T$ -spline surfaces where different rows may have different knot vectors, see Fig. 3. To solve this problem, periodic  $T$ -spline [19] uses periodic basis functions as outlined below.

#### 2.3.1. Periodic basis functions

Typically, B-spline basis functions are piecewise polynomials defined over a limited knot vector. They are defined recursively as

$$N_i^0(s) = \begin{cases} 1, & s \in [s_i, s_{i+1}), \\ 0, & \text{otherwise,} \end{cases}$$

$$N_i^k(s) = \frac{s - s_i}{s_{i+k} - s_i} N_i^{k-1}(s) + \frac{s_{i+k+1} - s}{s_{i+k+1} - s_{i+1}} N_{i+1}^{k-1}(s).$$

A B-spline basis function of degree  $k$  is non-zero only over  $k + 1$  intervals.

Consider the cubic B-spline basis function  $N_T[\mathbf{s}_i](s)$  associated with a knot vector  $\mathbf{s}_i = [s_{i,0}, s_{i,1}, s_{i,2}, s_{i,3}, s_{i,4}]$ . A corresponding periodic basis function is defined by

$$\tilde{N}_T[\mathbf{s}_i](s) = \sum_{j=-\infty}^{+\infty} N[\mathbf{s}_i](s + jT), \tag{2}$$

where  $N[\mathbf{s}_i](s + jT)$  is the usual B-spline basis function associated with the knot vector  $[s_{i,0} - jT, s_{i,1} - jT, s_{i,2} - jT, s_{i,3} - jT, s_{i,4} - jT]$ .  $\tilde{N}_T[\mathbf{s}_i](s)$  is a sum of infinite number of B-spline basis functions, however there is only limited number of terms that do not vanish at any given  $s$  value. As such,  $\tilde{N}_T[\mathbf{s}_i](s)$  is a periodic function with periodic  $T$ . If  $T < s_{i,4} - s_{i,0}$ , then the

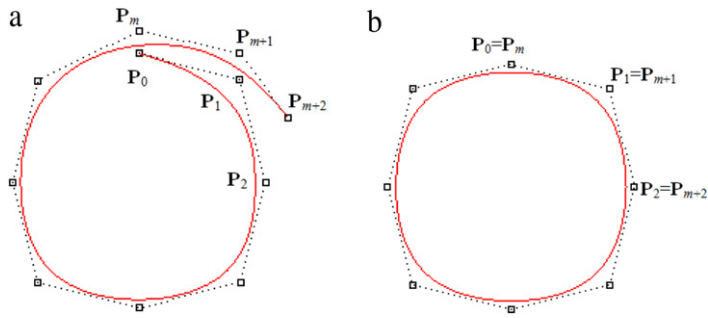


Fig. 1. Constructing a cubic periodic curve: (a) a non-periodic curve; and (b) repeating points to construct a periodic curve.

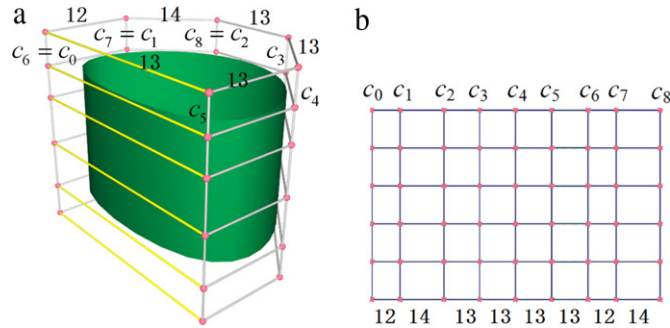


Fig. 2. A periodic cubic NURBS surface: (a) repeating three columns of control points and intervals; and (b) the preimage. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

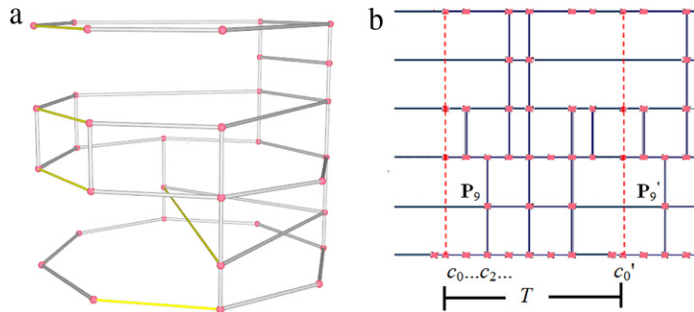


Fig. 3. Different repeated patterns in a periodic  $T$ -spline surface: (a) the  $T$ -mesh; and (b) the preimage. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

basis functions  $N[\mathbf{s}_i](s + jT)$  overlap with each other, thus  $\tilde{N}_T[\mathbf{s}_i](s) > 0$ , see Fig. 4(a). In this paper, the periodicity  $T$  will always satisfy  $T \geq s_{i,4} - s_{i,0}$ , which indicates (see Fig. 4(b))

$$\tilde{N}_T[\mathbf{s}_i](s) = N[\mathbf{s}_i](s + jT), \quad s \in [s_{i,0} - jT, s_{i,4} - jT].$$

With periodic basis functions, the periodic NURBS curve in Eq. (1) can be rewritten as

$$\mathbf{C}(s) = \sum_{i=0}^{m-1} \mathbf{P}_i \tilde{N}_T[\mathbf{s}_i](s), \quad s \in (-\infty, +\infty). \tag{3}$$

Similarly, a NURBS surface with  $s$ -periodic is

$$\mathbf{N}(s, t) = \sum_i \sum_j \mathbf{P}_{i,j} \tilde{N}_i^{k_1}(s) N_j^{k_2}(t). \tag{4}$$

Periodic surfaces use such basis functions in one or two parameter direction(s). In our case, we focus on the  $s$ -periodic surfaces that are homeomorphic to a cylinder. Such shapes are quite common in mechanical/medical engineering.

For a cubic NURBS curve in Eq. (3), since each control vertex  $\mathbf{P}_i$  is associated with a basis function of knots  $\mathbf{s}_i$ , we denote by  $\mathbf{P}_i[\mathbf{s}_i]$  the control vertex at the center of  $\mathbf{s}_i$ . Using this notation, every control vertex of the curve is associated with a

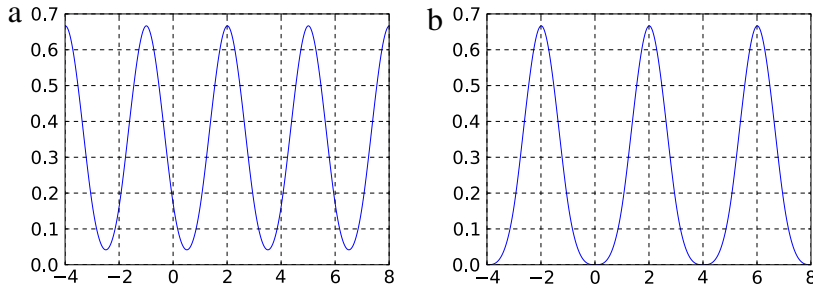


Fig. 4. Periodic basis functions with  $s_{i,4} - s_{i,0} = 4$ : (a) a periodic basis function with  $T = 3$ ; and (b) a periodic basis function with  $T = 4$ .

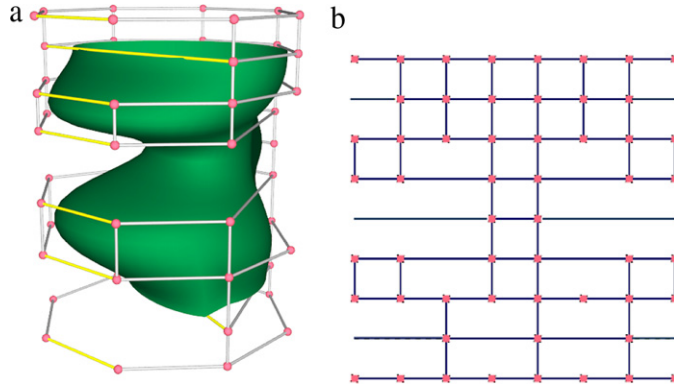


Fig. 5. A periodic  $T$ -spline surface: (a) the surface and the  $T$ -mesh; and (b) the preimage.

knot value belonging to  $\mathbf{K}$ . On the other hand, given  $s \in \mathbf{K}$ , we can find the associated control vertex  $\mathbf{P}[s]$ . Otherwise, knot refinement could be used to insert  $s$  into  $\mathbf{K}$ . Then, the associated control vertex  $\mathbf{P}[s]$  could also be found. In the latter case, after the vertex  $\mathbf{P}[s]$  is obtained, the curve is still represented using the original control vertices before the refinement, keeping the number of the control vertices unchanged.

2.3.2. Periodic  $T$ -spline surfaces

In this paper, cubic periodic basis functions are used for periodic  $T$ -spline, i.e.  $k = 3$ . A periodic  $T$ -spline with periodicity  $T$  in  $s$  direction can be formulated as follows:

$$\mathbf{S}(s, t) = \sum_{i=0}^n \mathbf{P}_i B_i^*(s, t), \tag{5}$$

where  $\mathbf{P}_i \in \mathbb{R}^4$  are weighted control vertices and  $B_i^*$  are periodic basis functions in  $s$  direction and given by

$$B_i^*(s, t) = \tilde{N}_T[\mathbf{s}_i](s)N[\mathbf{t}_i](t), \tag{6}$$

associated with  $\mathbf{s}_i = [s_{i,0}, s_{i,1}, s_{i,2}, s_{i,3}, s_{i,4}]$  and  $\mathbf{t}_i = [t_{i,0}, t_{i,1}, t_{i,2}, t_{i,3}, t_{i,4}]$ .

Compared to normal  $T$ -spline, a periodic  $T$ -spline has different domains in the  $s$  direction. In the preimage of a normal  $T$ -spline illustrated in Fig. 6(a), the domain is outlined by the filled rectangle. According to the definition of  $T$ -spline [11,12], in order to define the basis functions for the control vertices on the boundary, a virtual boundary surrounding the preimage is needed to guarantee the validity of the  $T$ -spline surface. However, a periodic  $T$ -spline has  $s \in (-\infty, +\infty)$  as depicted in Fig. 6(b) where no virtual boundary is required in the  $s$  direction. In Fig. 6(b), the parameters  $s = 8.8 - T$ ,  $s = 0$ , and  $s = T$  are associated with the columns  $c_{-1}$ ,  $c_0$  and  $c'_0$ , respectively. The two columns  $c_0$  and  $c'_0$  on the preimage correspond to the same column on the  $T$ -mesh.

In a NURBS surface,  $\mathbf{s}_i$  and  $\mathbf{t}_i$  can easily be determined from the global knot list of the surface. For  $T$ -spline,  $\mathbf{s}_i$  and  $\mathbf{t}_i$  are calculated under some rules as stated in [11,12]. A periodic  $T$ -spline differs in calculating  $\mathbf{s}_i$  when reaching the boundary. To find  $s_{i,3}$  and  $s_{i,4}$ , we cast a ray from  $\mathbf{P}_i$  in the parameter domain:  $R(x) = (s_{i,2} + x, t_{i,2})$ ,  $x > 0$ . Then  $s_{i,3}$  is the  $s$  parameter of the next control vertex or vertical edge intersecting the ray. Similarly,  $s_{i,4}$  can be obtained. When the ray crosses the right boundary, it continues to search for knots from the left boundary, and a period  $T$  is added to the knot obtained. For  $s_{i,0}$  and  $s_{i,1}$ , a period  $T$  should be subtracted when there is a jump from left boundary to right boundary.

Unfolding the periodic  $T$ -mesh in Fig. 3(a) at the knot interval indicated by yellow lines gives the preimage in Fig. 3(b) with columns  $c_i = c'_i$ , and vertices  $\mathbf{P}_i = \mathbf{P}'_i$ . In the  $T$ -mesh grid, knot intervals are constrained by the relationship that the

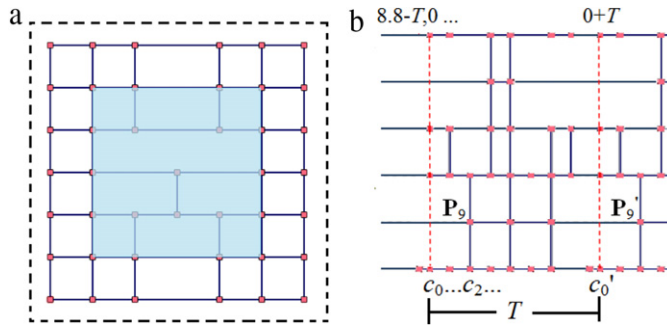


Fig. 6. Exacting knot vectors: (a) a non-periodic preimage; and (b) a periodic preimage.

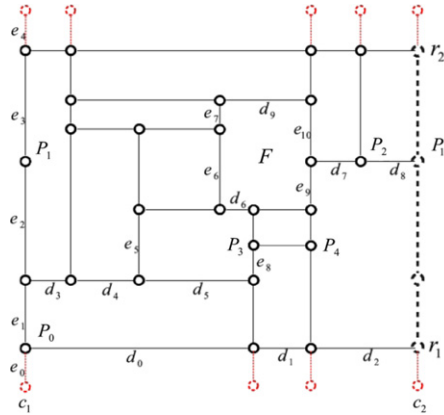


Fig. 7. Pre-image for a periodic  $T$ -spline surface (Fig. 8 from [19]).

sum of all knot intervals along one side of any face must be equal to the sum of knot intervals on the opposite side. The sum of all knot intervals between  $c_i$  and  $c_i'$  is the period  $T$ .

2.3.3. Periodic  $T$ -spline in semi-NURBS form

The  $T$ -mesh of a periodic  $T$ -spline, called periodic  $T$ -mesh, could have quite a flexible structure. For example, in Fig. 7, the control vertex  $P_2$  at the fourth row connects to a control vertex at the first row. Each row of the  $T$ -mesh corresponds to a periodic curve. Each control vertex on this row is connected to its neighboring control vertices on the same row (Fig. 3(b)). Since our aim is to provide a solution to skinning a set of closed sections, we only consider a special case of the periodic  $T$ -spline, i.e. if a control vertex at  $i$ th row connects to other vertices at different rows, then it can only connect to control vertices at the  $(i + 1)$ -th/ $(i - 1)$ -th row (Fig. 3). Such a constraint enables the periodic  $T$ -spline to support the skinning of closed sections.

A periodic  $T$ -spline could have an arbitrary number of control points along one periodic row as illustrated in Fig. 5. Furthermore, in a periodic  $T$ -mesh, there can be no vertical connection between two neighboring rows (the first two rows in Fig. 8). Let us assume that  $\mathbf{I}_\alpha$  consists of all control points' indices of one row on a periodic  $T$ -mesh, then we have the following definition.

**Definition 1** (Periodic  $T$ -spline in Semi-NURBS Form). A periodic  $T$ -spline in Eq. (5) is said to be in semi-NURBS form if all control vertices on the same row have the same basis function in the  $t$  direction, i.e.

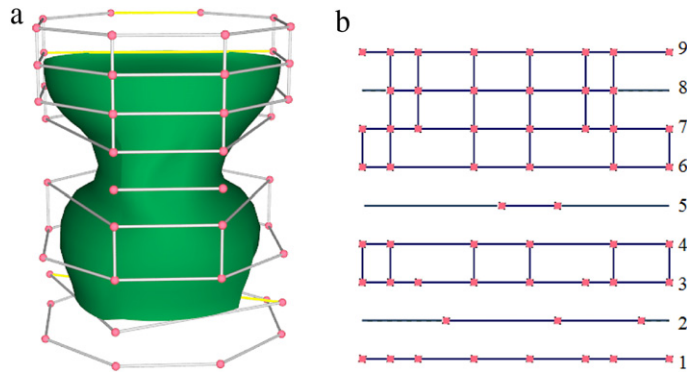
$$N[\mathbf{t}_i](t) = N[\mathbf{t}_j](t) = N_\alpha(t), \quad i, j \in \mathbf{I}_\alpha.$$

Such periodic  $T$ -spline can be rewritten in semi-NURBS form as

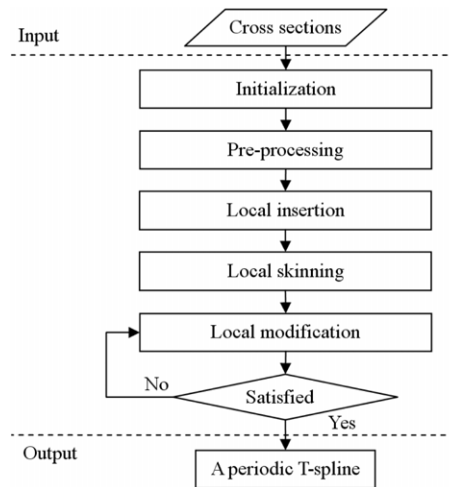
$$\mathbf{S}(s, t) = \sum_{\alpha} N_{\alpha}(t) \sum_{i \in \mathbf{I}_{\alpha}} \mathbf{P}_i \tilde{N}_T[\mathbf{s}_i](s). \tag{7}$$

For a parameter  $s$ ,  $s^-$  direction is denoted as the direction along which the parameter  $s$  decreases and  $s^+$  direction is the direction along which the parameter  $s$  increases. Non-periodic  $T$ -meshes can have  $I$ -junctions,  $L$ -junctions and/or  $T$ -junctions, whereas periodic  $T$ -meshes can only have  $T$ -junctions:

**Property 1.** Each control vertex in a periodic  $T$ -mesh has one  $s$ -edge connecting to the next vertex in the  $s^-$  direction and one  $s$ -edge connecting to the next vertex in the  $s^+$  direction. Thus, there are no  $L$ -junctions and  $I$ -junctions.



**Fig. 8.** A periodic  $T$ -mesh with partial vertical connection: (a) the  $T$ -mesh and the surface; and (b) the preimage (the number  $i$  indicates the  $i$ th row).



**Fig. 9.** Flowchart for periodic  $T$ -spline local skinning.

In the rest of this paper, periodic  $T$ -spline surfaces refer to periodic  $T$ -spline surfaces in semi-NURBS form.

### 3. Periodic $T$ -spline surface skinning

Given a series of closed cross sections, skinning aims to exactly reconstruct a surface interpolating these curves. NURBS skinning cannot handle the incompatible cross sections directly whereas periodic  $T$ -spline skinning can easily resolve the incompatibility problem due to its intrinsic property.

As already established, NURBS skinning derives the control vertices of the skinned surface by globally solving a linear system of equations and the non-periodic  $T$ -spline skinning [18] obtains the control vertices by locally solving a linear system of equations. Periodic  $T$ -spline skinning provides another local solution with explicit formulae to obtain the control vertices row by row. This enables users to perform local modifications and to smooth out the shape of the skinned surface, two typical problems in any interpolation process.

#### 3.1. Skinning flowchart

In this section we describe the basic steps in the flowchart to generate a periodic skinned  $T$ -spline surface as illustrated in Fig. 9.

**Initialization** consists of computing the knot vectors and creating an initial  $T$ -mesh. For each cross section there corresponds one row of the  $T$ -mesh.

**Pre-processing** performs refinements on some of the cross sections to guarantee that the number of  $t$ -edges connecting one row to its neighboring row is not less than a pre-defined minimal number.

**Local insertion** inserts *intermediate* cross sections and thus adds more rows to the initial  $T$ -mesh.

**Local skinning** produces a periodic  $T$ -spline surface by locally interpolating each input cross section.

**Local modification** allows users to perform local modifications to adjust the shape of the skinned surface.

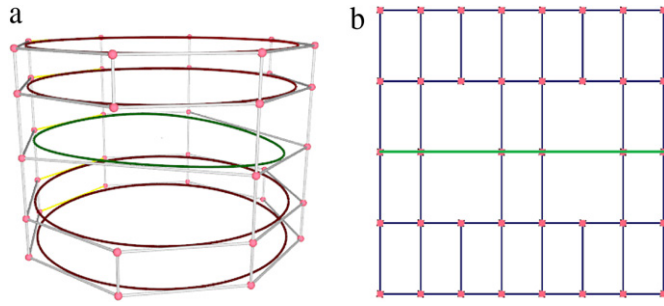


Fig. 10. A  $T$ -mesh after the initialization: (a) the  $T$ -mesh and the cross sections; and (b) the preimage after initialization.

### 3.2. Initialization

In the *initialization* step, we use the control vertices of the input cross sections to create an initial  $T$ -mesh and two initial knot vectors  $\mathbf{K}_s$  and  $\mathbf{K}_t$  for the skinned surface  $\mathbf{S}(s, t)$ . Since each given curve will be considered as an iso-parametric curve corresponding to a certain  $t$  knot value, the knot vector  $\mathbf{K}_t$  can be obtained from the union of these knot values. However, in periodic  $T$ -spline skinning there is no need to refine the curves in the  $s$  direction as the case in NURBS skinning. It is noteworthy to mention that although different  $\mathbf{K}_t$  may lead to different skinned surfaces, this, however, does not affect the feasibility of our algorithm.

As for the initial  $T$ -mesh, we arrange the vertices  $\mathbf{P}_{i,q}$  row by row to form a  $T$ -mesh. If there are common knots in two adjacent rows, then we connect them by  $t$ -edges. As illustrated in Fig. 10, a  $B$ -spline curve is shown with its control points highlighted in the pre-image. Six vertices  $\mathbf{P}_{i,q}$  of this curve serve as one row in the  $T$ -mesh where twelve  $t$ -edges are used to connect vertically to this row.

### 3.3. Pre-processing

As mentioned in Section 2.3, periodic  $T$ -spline allows no  $t$ -edges between two adjacent rows. For instance, there is no edge between the fourth and the fifth row as depicted in Fig. 8. Such a  $T$ -mesh becomes disconnected which is not suitable for intuitive editing. For skinning purposes, the number of  $t$ -edges between two adjacent rows should not be less than a pre-define number, at least 4 in our implementation. Otherwise, additional  $t$ -edges will be created by applying refinement to some of the input cross sections. Users can interactively refine a curve according to their requirements through our system. The influence of the pre-processing will be illustrated in Section 4.2.

### 3.4. Local insertion

As stated in Section 2.1, each skinned curve  $\mathbf{C}_q$  has a feature value  $f_q$  which determines its type of cross-smoothness by which the curve is interpolated. In our implementation, the first and last cross sections are considered as sharp features, whereas all other curves are initially given the default value  $f_q = C^2$  which means that they are interpolated with  $C^2$  continuity. The user could interactively change any of these values to  $C^0$  to turn the corresponding curve into a sharp edge, as illustrated in Section 4.3.

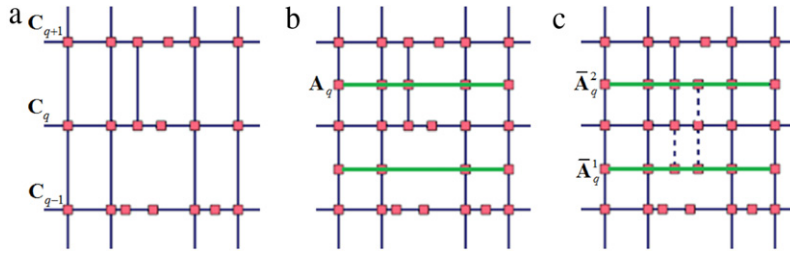
Generally, a pair of adjacent cross sections  $\mathbf{C}_q$  and  $\mathbf{C}_{q+1}$ , corresponding to two knot lines  $t_q$  and  $t_{q+1}$ , respectively, may have some common knots. For each of these knots, a  $t$ -edge is created vertically connecting two control points, one from each cross section, as shown in Fig. 11(a). Local insertion adds a periodic intermediate cross section between  $\mathbf{C}_q$  and  $\mathbf{C}_{q+1}$  according to the values  $f_q$  and  $f_{q+1}$ .

**Definition 2 (Local Insertion).** If  $f_q = f_{q+1} = C^0$ , two intermediate cross sections  $\mathbf{A}_{q1}$  and  $\mathbf{A}_{q2}$  with respective knots  $x_{q1}$  and  $x_{q2}$  are inserted. Otherwise, one cross section  $\mathbf{A}_q$  is inserted corresponding to a knot  $x_q$ .

Let us assume that there are  $k_c$  common knots between two given cross sections  $\mathbf{C}_q$  and  $\mathbf{C}_{q+1}$  then the corresponding intermediate cross section  $\mathbf{A}_q$  consists of  $k_c$  control vertices  $\mathbf{A}_{i,q}$ . The weights for the vertices of  $\mathbf{A}_{i,q}$  are taken as 1 and their positions are calculated using the 4-point interpolatory subdivision scheme [20] as

$$\mathbf{A}_{i,q} = \frac{\mathbf{P}_{i,q} + \mathbf{P}_{i,q+1}}{2} + \alpha \left( \frac{\mathbf{P}_{i,q} + \mathbf{P}_{i,q+1}}{2} - \frac{\mathbf{P}_{i,q-1} + \mathbf{P}_{i,q+2}}{2} \right),$$

$$x_q = \begin{cases} \frac{t_q + t_{q+1}}{2}, & f_q = f_{q+1} = C^2, \\ t_q, & f_q = C^0, f_{q+1} = C^2, \\ t_{q+1}, & f_q = C^2, f_{q+1} = C^0, \end{cases} \tag{8}$$



**Fig. 11.** Surface skinning scheme: (a) the  $T$ -mesh after initialization; (b) the  $T$ -mesh after intermediate insertion; and (c) temporary points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where  $\mathbf{P}_{i,j}$ ,  $j \in \{q - 1, q, q + 1, q + 2\}$  are the control vertices, corresponding to the same parameter  $s$ , of the four cross sections  $\mathbf{C}_j$ ,  $j \in \{q - 1, q, q + 1, q + 2\}$ , and the parameter  $\alpha$  is given a default value of 0.2. This could be adjusted interactively.

If  $f_q = f_{q+1} = C^0$ , then two intermediate cross sections will be inserted as

$$\begin{aligned} \mathbf{A}_{i,q1} &= \frac{\mathbf{A}_{i,q} + \mathbf{P}_{i,q}}{2}, \\ \mathbf{A}_{i,q2} &= \frac{\mathbf{A}_{i,q} + \mathbf{P}_{i,q+1}}{2}, \\ x_{q1} &= t_q, \\ x_{q2} &= t_{q+1}. \end{aligned} \tag{9}$$

To illustrate, consider the  $T$ -mesh in Fig. 11(a) where the three rows correspond to three cross sections  $\mathbf{C}_{q-1}$ ,  $\mathbf{C}_q$ , and  $\mathbf{C}_{q+1}$ . There are 4 common knots between  $\mathbf{C}_{q-1}$  and  $\mathbf{C}_q$ , and 5 common knots between  $\mathbf{C}_q$  and  $\mathbf{C}_{q+1}$ . By default, local insertion will insert two intermediate cross sections, the two green curves in Fig. 11(b).

These intermediate cross sections correspond to periodic curves with the same period as the input cross sections. Moreover, we have

**Corollary 1.** Given  $n$  cross sections, if  $n_0$  neighbor-pairs satisfy  $f_q = f_{q+1} = C^0$  ( $n_0 = 0, 1, \dots, n - 1$ ), the number of intermediate cross sections inserted will be  $n + n_0 - 1$ , and the total number of cross sections becomes  $2n + n_0 - 1$ .

**Corollary 2.** For an interior cross section  $\mathbf{C}_q[t_q]$  with a smoothness value  $f_q$ , there are two neighboring intermediate cross sections denoted by  $\mathbf{A}_q^1[x_q^1]$  and  $\mathbf{A}_q^2[x_q^2]$  with knot values  $x_q^1$  and  $x_q^2$ , respectively, such that

$$\begin{cases} x_q^1 < t_q < x_q^2, & \text{if } f_q = C^2, \\ x_q^1 = t_q = x_q^2, & \text{if } f_q = C^0. \end{cases}$$

After inserting the necessary intermediate cross sections, the skinned surface is defined by the following

- A  $T$ -mesh consisting of the rows  $\mathbf{P}_{i,q}$  of the given cross sections and the rows of the intermediate cross sections  $\mathbf{A}_{i,q}$ ,
- the knot vector  $\mathbf{K}_s$ , and
- the knot vector  $\mathbf{K}_t$  refined to contain the parameters  $\{x_q\}$  for intermediate cross sections.

It is not necessary for the skinned surface  $\mathbf{S}(s, t)$  to interpolate the intermediate cross sections. However, in order to interpolate the given curves, we fix the vertices  $\mathbf{A}_{i,q}$  of the intermediate cross sections and adjust the vertices  $\mathbf{P}_{i,q}$  of the given sections.

Typically,  $\mathbf{C}_q$  could be more refined than  $\mathbf{A}_q^1$  and  $\mathbf{A}_q^2$ . However,  $\mathbf{A}_q^1$  and  $\mathbf{A}_q^2$  can be refined to have the same knot vector as  $\mathbf{C}_q$ . Let us call the refined cross sections  $\bar{\mathbf{A}}_q^1$  and  $\bar{\mathbf{A}}_q^2$ , respectively. Fig. 11(c) shows the result of replacing  $\mathbf{A}_q^1$  and  $\mathbf{A}_q^2$  with their corresponding refined  $\bar{\mathbf{A}}_q^1$  and  $\bar{\mathbf{A}}_q^2$ , respectively. The three cross sections  $\bar{\mathbf{A}}_q^1$ ,  $\mathbf{C}_q$ , and  $\bar{\mathbf{A}}_q^2$  locally form a regular structure, thus it is convenient for updating  $\mathbf{P}_{i,q}$ . The final preimage is given in Fig. 11(b) where the temporary structure for updating  $\mathbf{C}_q$  is given in Fig. 11(c). As for updating  $\mathbf{C}_{q+1}$ , the control polygons  $\mathbf{A}_{q+1}^1 = \mathbf{A}_q^2$  will be refined to form a different temporary and corresponding structure.

### 3.5. Local skinning

According to Corollary 2, for each cross section  $\mathbf{C}_q$ , there are three rows on the  $T$ -mesh  $\mathbf{P}$  that correspond to  $\mathbf{A}_q^1$ ,  $\mathbf{C}_q$ , and  $\mathbf{A}_q^2$ . Suppose that  $\mathbf{I}_{q-1}$ ,  $\mathbf{I}_q$ , and  $\mathbf{I}_{q+1}$  are the set of indices of their control vertices, respectively. During local skinning, all intermediate cross sections remain unchanged, i.e.  $\mathbf{I}_{q-1}$  and  $\mathbf{I}_{q+1}$  do not change and only  $\mathbf{I}_q$  will be modified.

**Corollary 3.** The equality  $\mathbf{S}(s, t_q) = \mathbf{C}_q$  is not satisfied all the time. As such, the surface  $\mathbf{S}(s, t)$  may not generally interpolate the cross section.

**Proof.** Referring to Fig. 11(b), the shape of the iso-parametric curve  $\mathbf{S}(s, t_q)$  depends on  $\mathbf{I}_{q-1}$ ,  $\mathbf{I}_q$ , and  $\mathbf{I}_{q+1}$ . According to Definition 1, the iso-parametric curve can be formulated as follows:

$$\mathbf{S}(s, t_q) = \sum_{\alpha \in \{q-1, q, q+1\}} N_\alpha(t_q) \sum_{i \in \mathbf{I}_\alpha} \mathbf{P}_i \tilde{N}_T[\mathbf{s}_i](s). \tag{10}$$

Let

$$a = N_{q-1}(t_q), \quad b = N_q(t_q), \quad c = N_{q+1}(t_q), \tag{11}$$

then Eq. (10) becomes

$$\begin{aligned} \mathbf{S}(s, t_q) &= a \sum_{i \in \mathbf{I}_{q-1}} \mathbf{P}_i \tilde{N}_T[\mathbf{s}_i](s) + b \sum_{i \in \mathbf{I}_q} \mathbf{P}_i \tilde{N}_T[\mathbf{s}_i](s) + c \sum_{i \in \mathbf{I}_{q+1}} \mathbf{P}_i \tilde{N}_T[\mathbf{s}_i](s) \\ &= a\mathbf{A}_q^1 + b\mathbf{C}_q + c\mathbf{A}_q^2. \end{aligned} \tag{12}$$

Since  $a + b + c = 1$ ,  $\mathbf{S}(s, t_q) = \mathbf{C}_q$  is not always valid.  $\square$

$\bar{\mathbf{A}}_q^1$  and  $\bar{\mathbf{A}}_q^2$  can be refined to have the same knot vector as  $\mathbf{C}_q$ .

$$\begin{aligned} \bar{\mathbf{A}}_q^1 &= \sum_{i \in \mathbf{I}_{q-1}} \mathbf{P}_i \tilde{N}_T[\mathbf{s}_i](s) \\ &= \sum_{i \in \mathbf{I}_q} \bar{\mathbf{P}}_i^1 \tilde{N}_T[\mathbf{s}_i](s), \\ \bar{\mathbf{A}}_q^2 &= \sum_{i \in \mathbf{I}_{q+1}} \mathbf{P}_i \tilde{N}_T[\mathbf{s}_i](s) \\ &= \sum_{i \in \mathbf{I}_q} \bar{\mathbf{P}}_i^2 \tilde{N}_T[\mathbf{s}_i](s), \end{aligned} \tag{13}$$

where  $\mathbf{P}_i^1$  and  $\mathbf{P}_i^2$  form the control polygons of the intermediate cross sections, and  $\bar{\mathbf{P}}_i^1, \bar{\mathbf{P}}_i^2$  are their respective refined control polygons. These control polygons are illustrated in green lines in Fig. 11(b) and (c), respectively.

**Definition 3 (Local Skinning).** Local skinning consists of replacing all points  $\mathbf{P}_i$  on the row  $\mathbf{I}_q$  by  $\mathbf{W}_i$  with

$$\mathbf{W}_i = \frac{\mathbf{P}_i - a\bar{\mathbf{P}}_i^1 - c\bar{\mathbf{P}}_i^2}{b}, \tag{14}$$

where  $a, b, c$  are defined in Eq. (11). All other rows remain unchanged.

**Theorem 1.** Local skinning gives  $\mathbf{S}(s, t_q) = \mathbf{C}_q$ . As such, the surface interpolates the corresponding cross section.

**Proof.** Performing a local skinning process, and substituting the new vertices in Eq. (14), the iso-parametric curve becomes

$$\begin{aligned} \mathbf{S}(s, t_q) &= a\bar{\mathbf{A}}_q^1 + b \sum_{i \in \mathbf{I}_q} \mathbf{W}_i \tilde{N}_T[\mathbf{s}_i](s) + c\bar{\mathbf{A}}_q^2 \\ &= \sum_{i \in \mathbf{I}_q} (a\bar{\mathbf{P}}_i^1 + b\mathbf{W}_i + c\bar{\mathbf{P}}_i^2) \tilde{N}_T[\mathbf{s}_i](s) \\ &= \sum_{i \in \mathbf{I}_q} \mathbf{P}_i \tilde{N}_T[\mathbf{s}_i](s). \end{aligned}$$

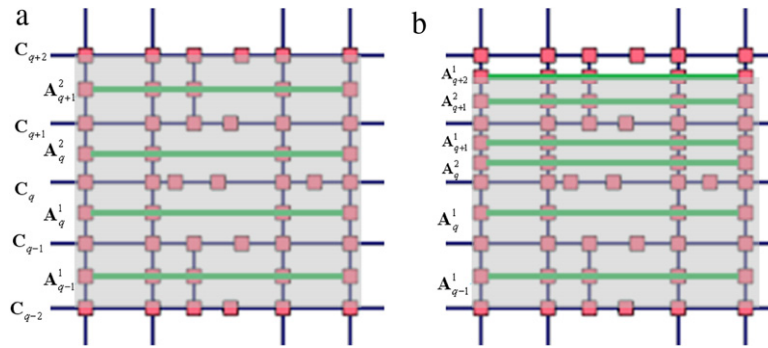
This is equivalent to  $\mathbf{S}(s, t_q) = \mathbf{C}_q$ .  $\square$

On the  $T$ -mesh, some rows correspond to input cross sections and others correspond to intermediate cross sections. Local skinning is performed on all rows corresponding to input cross sections. As such, the skinned surface  $\mathbf{S}(s, t)$  interpolates all the input cross sections.

### 3.6. Local modification

After a local skinning operation, the users could interactively perform local modifications on any cross section  $\mathbf{C}_q$  in two different manners:

- Adjusting the control points  $\mathbf{P}_i$  of that cross section, and/or
- Changing the feature value  $f_q$  of that cross section.



**Fig. 12.** Local modification: (a) one intermediate cross section between  $C_{q+2}$  and  $C_{q+1}$ ; and (b) two intermediate cross sections between  $C_{q+2}$  and  $C_{q+1}$ .

Given three neighboring input cross sections  $C_{q-1}$ ,  $C_q$  and  $C_{q+1}$ , we define *local modification* as follows:

**Definition 4** (*Local Modification*). Local modification is the process of editing a cross section  $C_q$  by modifying the  $T$ -mesh using two processes: the intermediate update process and the cross update process outlined below

*Intermediate update* computes intermediate cross sections between  $C_{q-1}$  and  $C_{q+1}$  using Eqs. (9) and (8), then updates the corresponding rows on the  $T$ -mesh.

*Cross update* applies local skinning on  $C_q$  using Eq. (14), then updates the corresponding three rows  $C_{q-1}$ ,  $C_q$  and  $C_{q+1}$  on the  $T$ -mesh.

Therefore, editing a cross section will only update few rows in the  $T$ -mesh. For instance, in Fig. 12(a) and (b), modifying  $C_q$  will update 5 rows, and 6 rows, respectively.

**Theorem 2.** Editing  $C_q$  locally affects the surface between the rows  $C_{q-2}$  and  $C_{q+2}$ .

**Proof.** The  $i$ th row of a periodic  $T$ -mesh controls the surface shape between the row  $(i-2)$  and the row  $(i+2)$ . Referring to Fig. 12, all modified rows are bounded by  $C_{q-1}$  and  $C_{q+1}$ . There is at least one intermediate cross section between two input cross sections. Thus, the surface modification is bounded by the rows between  $C_{q-2}$  and  $C_{q+2}$ .  $\square$

In Fig. 12(a), modifying  $C_q$  will affect the shape of the surface between  $C_{q-2}$  and  $C_{q+2}$ . Nevertheless, editing  $C_q$  in Fig. 12(b) modifies the surface between  $C_{q-2}$  and  $A_{q+2}^1$ .

## 4. Discussions and analysis

In this section, we compare our results with NURBS skinning, discuss the advantages over the non-periodic  $T$ -spline skinning [18], and analyze the implementation of our approach.

### 4.1. Local skinning and local modification

Usually, NURBS skinning solves a global linear system to compute the positions of the control vertices. As such, local modification of an input cross section is computationally expensive. Therefore, for NURBS skinning, both the skinning procedure and the modification are global. Non-periodic  $T$ -spline skinning [18] depends on solving a local linear system of equations without considering the local modifications.

Contrary to NURBS skinning, periodic  $T$ -spline skinning supports both local skinning and local modification. Without solving a linear system, periodic  $T$ -spline skinning adopts the explicit formula Eq. (14) to locally perform the skinning. Moreover, modification of any cross section has local influence on the skinned surface around this cross section. Fig. 13 shows an example of skinning a C-shape. Local skinning is applied to interpolate the input cross sections one by one (Fig. 13(a)). The skinned surface can be locally modified by moving any cross section. For instance, moving  $C_q$  to the left hand side affects the region between  $C_{q-2}$  and  $C_{q+2}$  only, as depicted in Fig. 13(b).

The local modification and the local skinning isolate each cross section from the others by inserting the intermediate cross sections. These are incompatible cross sections. If an intermediate cross section needs to be inserted in NURBS skinning, then it should be compatible with the input cross sections. Thus, the local modification and the local skinning are not suitable for NURBS skinning. Non-periodic  $T$ -spline skinning [18] needs to solve a local system of equations. Periodic  $T$ -spline skinning can perform better by adopting the explicit formula Eq. (14) without requiring to solve linear equations.

### 4.2. Pre-processing and local insertion

Given an input cross section, the positions of its vertices after skinning depend on the vertices of the two intermediate rows located above and below this cross section. Although in principle, very few vertices are needed to guarantee the

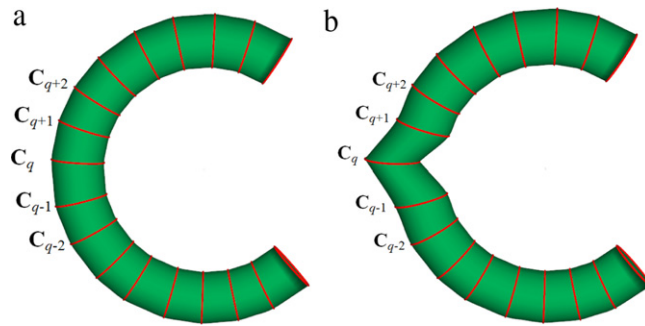


Fig. 13. Local skinning and local modification on a C-shape: (a) local skinning; and (b) local modification.

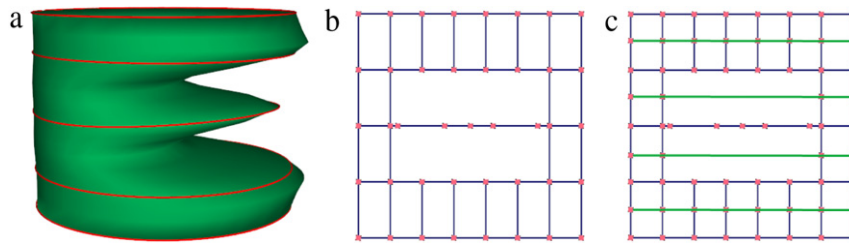


Fig. 14. Too few common knots result in sunk surface: (a) the skinned surface and the cross sections; (b) the preimage after initialization; and (c) the preimage after skinning.

interpolation of the input section, the shape of the surface depends on the number and the positions of these vertices. For example in Fig. 14(c), the second intermediate cross section (the fourth row) has only four control vertices which are closely positioned. Due to the convex hull property, the skinned surface will show some undesirable wiggles, as depicted in Fig. 14(a). As such, the intermediate cross sections should be constructed carefully. An intermediate cross section should neither have too few control points nor have the control points gathered in a small area. These can be avoided by refining two neighboring input cross sections during a pre-processing phase after which the shape of the surface is quietly improved, as depicted in Figs. 14(a) and 15(a).

The non-periodic  $T$ -spline skinning uses the middle point between two vertices to construct the intermediate cross sections. The wiggle effect will be more visible. In our implementation, the intermediate control vertices are explicitly computed using Eqs. (9) and (8) to achieve better visual results. Moreover, other formulations could be investigated where additional conditions such as fairness could be used.

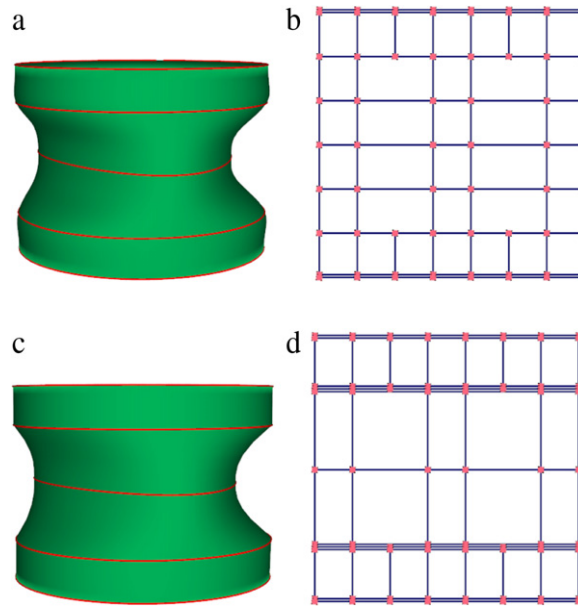
### 4.3. Surface features

Typically, each input cross section  $C_i$  corresponds to a knot line  $t_i$  and its two neighbor intermediate cross sections correspond to two different knot lines, say  $t_{i-1}$  and  $t_{i+1}$ . Normally, these knots are different and the surface is  $C^2$  across the input and the intermediate cross sections. If we make  $t_{i-1} = t_{i+1} = t_i$ , then the surface becomes  $C^0$  continuous along the input cross section  $C_i$  which becomes a crease as depicted in Fig. 15. In this regard, the introduction of the intermediate rows not only allows a local solution for the skinning problem but also makes it possible to generate various features along the skinned curves. In current implementations, we have supported  $C^0$  and  $C^2$  cross-smoothness. Actually,  $C^1$  cross-smoothness can also be included by setting  $t_{i+1} = t_i > t_{i-1}$  or  $t_{i+1} > t_i = t_{i-1}$ . Furthermore, fairness conditions and other derivative constraints, such as curvature or tangent plane conditions, could be implemented providing local control modification of the skinned surface.

### 4.4. Number of control points

Periodic  $T$ -spline skinning provides a solution to handle the incompatibility problem and supports local modification along skinned curves. However, we do not claim to reduce the total number of control points in all cases. Applying NURBS skinning in Fig. 15 may result in a smaller number of control vertices compared to using periodic  $T$ -spline skinning. The number of control points on final  $T$ -mesh depends on the degree of incompatibility of the input cross sections. If most of these cross sections are incompatible, then certainly the  $T$ -spline approach will generate much less control points on the skinned surface.

With  $n$  cross sections as input, there are  $n$  knot vectors. In the NURBS setting, one incompatible knot will introduce up to  $n - 1$  control vertices. As the incompatibility increases, the number of control vertices required in the NURBS skinning



**Fig. 15.** Feature generation at the second and the fourth cross section: (a) surface with  $C^2$  feature; (b) the preimage; (c) surface with  $C^0$  feature; and (d) the preimage.

**Table 1**

The number of control points using periodic  $T$ -spline skinning and NURBS skinning.

	Input	Periodic $T$ -spline	NURBS
Fig. 15	38	66	40
Fig. 16	195	360	702
Fig. 17	346	624	1056
Fig. 18	563	1061	1677
Fig. 19	655	1236	1608

will increase dramatically. According to [2], if the average number of control points among all cross sections is  $m$ , then the number of control vertices can be of  $O(mn^2)$ . Periodic  $T$ -spline skinning does not need to refine the cross sections to make them compatible. Thus, the number of control points can be reduced to  $O(mn)$ . Table 1 lists the number of control points for some examples in this paper. The number of control points using periodic  $T$ -spline skinning is almost twice as many as the input, a reduction of almost 50% as compared to NURBS skinning.

## 5. Examples

The following figures are examples for periodic  $T$ -spline skinning. The skinned surface is not restricted to be a revolution surface. Fig. 16 gives an example of a bottle model. Skinning is carried along the  $t$  vertical direction as shown in Fig. 16(b). Since the first and last cross sections are interpolated, the  $t$  values for the first two rows, and the last two ones, are equal, respectively. The input cross sections have a total of 195 control points. Periodic  $T$ -spline skinning will almost double the number of control points. However, the number of control points required for NURBS skinning increases to almost four times.

Figs. 17 and 18 show two chess models with 22 and 39 cross sections, respectively. As the incompatibility of the input cross sections increases, periodic  $T$ -spline skinning performs better than NURBS skinning. In Fig. 17, the fifth cross section is set to have a sharp feature  $C^0$ , which is difficult for NURBS skinning.

In previous examples, the input cross sections are all planar curves. Fig. 19 presents a head model where the skinned surface is generated from a set of non-planar cross sections.

Skinned surfaces can be used for human liver modeling. Liver modeling from CT images is important for liver transplantation and liver resection. The visualization of modeling result can help surgeons to better understand the shape of the patient's liver during the surgery planning. Usually, 1 mm CT scan of a human liver may include up to more than one hundred slices. Each slice can provide a cross section. Compared with NURBS skinning,  $T$ -spline skinning can help to reduce the total number of control points in liver modeling. Fig. 20 shows the liver by skinning 44 cross sections.

Our method can also be extended to model the torus-like surfaces. Fig. 13(a) is an example of the  $s$ -periodic  $T$ -splines. Cross section  $C_q$  is a periodic curve with parameter  $s$ . If we extend the C-shape surface by connecting the first and the last cross section, it forms an O-shape surface, i.e. a torus-like surface. Such a surface is both  $s$ -periodic and  $t$ -periodic.

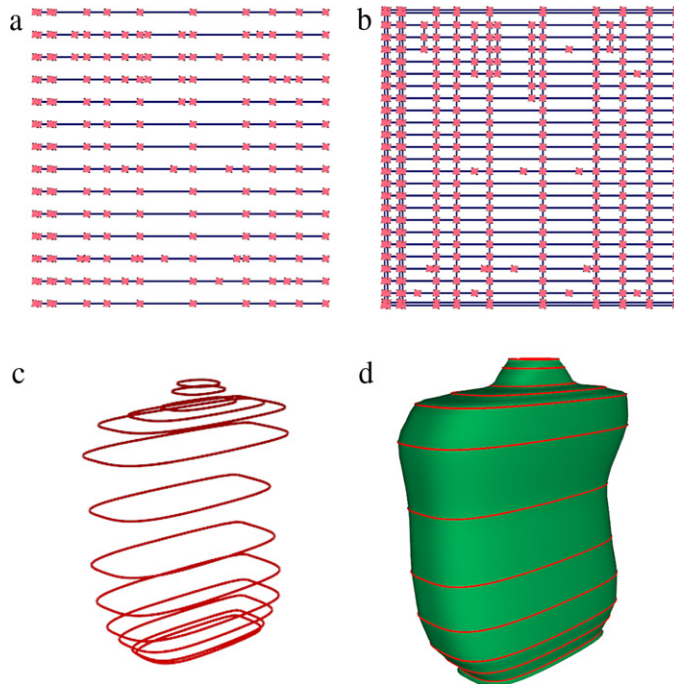


Fig. 16. *T*-spline skinned bottle: (a) the knot vectors for the cross sections; (b) the preimage; (c) the cross sections; and (d) the skinned surface.

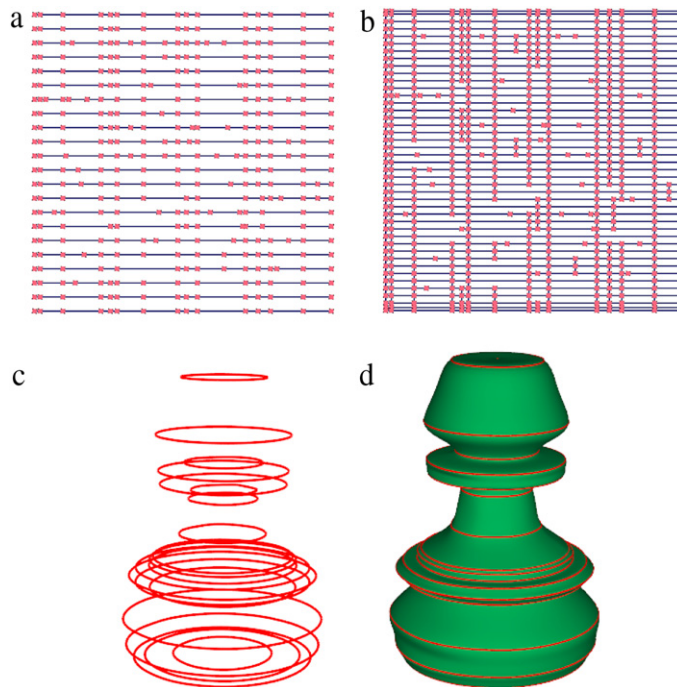
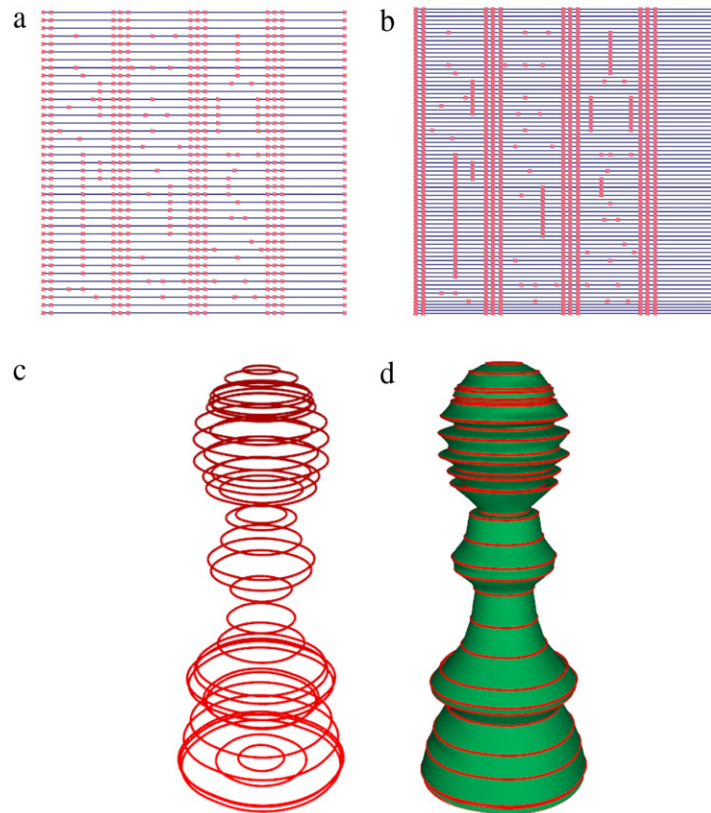


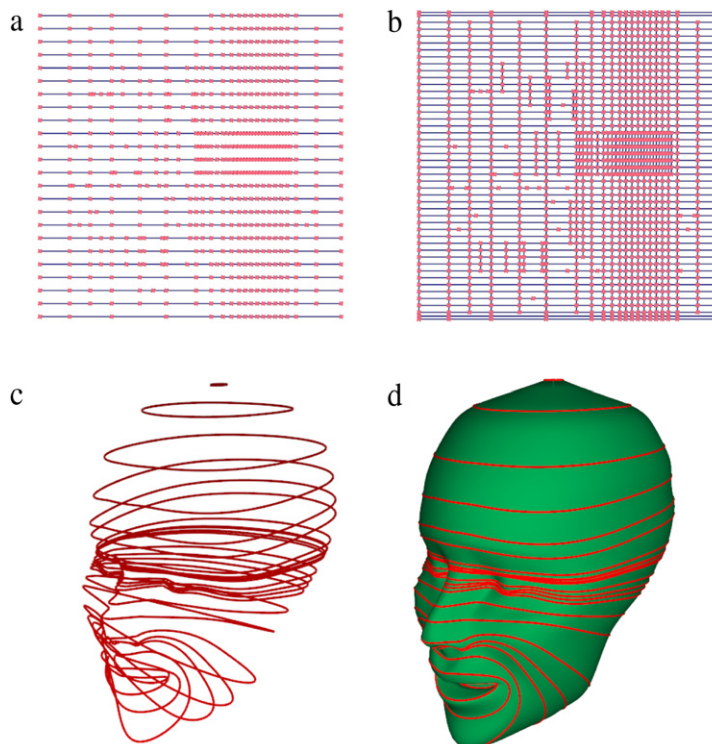
Fig. 17. A pawn model: (a) the knot vectors for the cross sections; (b) the preimage; (c) the cross sections; and (d) the skinned surface.

## 6. Conclusion and future work

Periodic *T*-splines in semi-NURBS form provide a solution to the skinning problem of closed NURBS curves. In practice, the input curves are naturally incompatible. Even after skinning, the designer may need to modify some specific input curves, thus increasing the degree of incompatibility of these curves. Compared with traditional NURBS skinning, periodic *T*-spline skinning can handle such incompatibilities even with less control vertices. Different from the non-periodic *T*-spline skinning



**Fig. 18.** A bishop model: (a) the knot vectors for the cross sections; (b) the preimage; (c) the cross sections; and (d) the skinned surface.



**Fig. 19.** T-spline skinned human head: (a) the knot vectors for the cross sections; (b) the preimage; (c) the cross sections; and (d) the skinned surface.

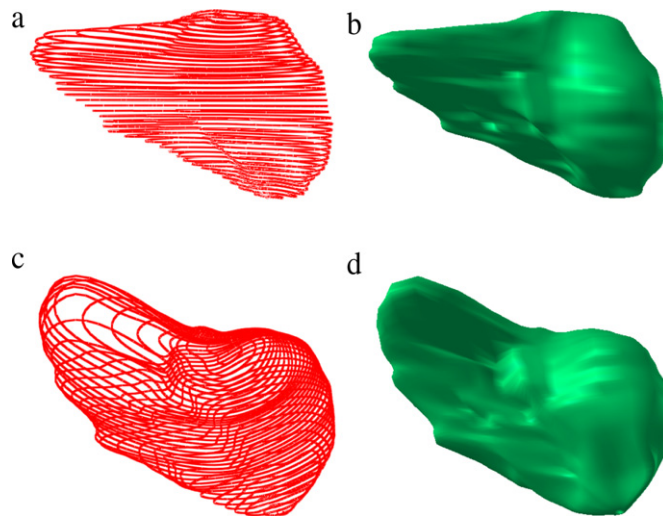


Fig. 20. *T*-spline skinned human liver: (a, c) the cross sections at different angles; and (b, d) the skinned surface at different angles.

method [18] and traditional NURBS skinning methods, our approach generates tubular skinned surfaces by using the explicit formulae without solving any local or global system of equations. As such, local modification is possible and various features can be generated along skinned curves.

In the current implementation, the 4-point interpolatory subdivision scheme is used for the intermediate cross sections. Future work will consider intermediate cross sections based on some criteria such as smoothing or fairing. Moreover, additional features such as pre-defined tangent plane or curvature values could be specified by the designers.

## Acknowledgments

This work was initiated during a visit by Ahmad Nasri to Nanyang Technological University. It is partially supported by the ARC9/09 Grant (MOE2008-T2-1-075) and AcRF RG 10/12 Grant (M4011022.050) of Singapore, a research grant (JCOAG03-FG05\_2009) and SERC grant (1225100006) from A\* STAR of Singapore, and a URB grant (URB-2009-2010) from the American University of Beirut.

## References

- [1] C.D. Woodward, Skinning techniques for interactive *B*-spline surface interpolation, *Comput.-Aided Des.* 20 (8) (1988) 441–451.
- [2] L.A. Piegl, W. Tiller, Surface skinning revisited, *Vis. Comput.* 18 (4) (2002) 273–283.
- [3] L.A. Piegl, W. Tiller, Reducing control points in surface interpolation, *IEEE Comput. Graph. Appl.* 20 (5) (2000) 70–74.
- [4] H. Park, K. Kim, Smooth surface approximation to serial cross-sections, *Comput.-Aided Des.* 28 (12) (1996) 995–1005.
- [5] L. Piegl, W. Tiller, Algorithm for approximate NURBS skinning, *Comput.-Aided Des.* 28 (9) (1996) 699–706.
- [6] H. Park, K. Kim, S.C. Lee, Method for approximate NURBS curve compatibility based on multiple curve refitting, *Comput.-Aided Des.* 32 (4) (2000) 237–252.
- [7] H. Park, Lofted *B*-spline surface interpolation by linearly constrained energy minimization, *Comput.-Aided Des.* 35 (14) (2003) 1261–1268.
- [8] W.K. Wang, H. Zhang, H. Park, J.H. Yong, J.C. Paul, J.G. Sun, Reducing control points in lofted *B*-spline surface interpolation using common knot vector determination, *Comput.-Aided Des.* 40 (10–11) (2008) 999–1008.
- [9] W. Tiller, Knot-removal algorithms for NURBS curves and surfaces, *Comput.-Aided Des.* 24 (8) (1992) 445–453.
- [10] A. Nasri, A. Abbas, I. Hasbini, Skinning Catmull–Clark subdivision surfaces with incompatible cross-sectional curves, in: *Proceedings of 11th Pacific Conference on Computer Graphics and Applications, 2003*, pp. 102–111.
- [11] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, *T*-splines and *T*-NURCCs, *ACM Trans. Graph.* 22 (2003) 477–484.
- [12] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, T. Lyche, *T*-spline simplification and local refinement, *ACM Trans. Graph.* 23 (2004) 276–283.
- [13] T.W. Sederberg, G.T. Finnigan, X. Li, H. Lin, H. Ipson, Watertight trimmed NURBS, *ACM Trans. Graph.* 27 (3) (2008) 1–8.
- [14] J. Zheng, Y. Wang, H. Seah, Adaptive *T*-spline surface fitting to *Z*-map models, in: *Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, ACM, 2005*, pp. 405–411.
- [15] Y. Wang, J. Zheng, Adaptive *T*-spline surface approximation of triangular meshes, in: *6th International Conference on Information, Communications & Signal Processing, IEEE, 2007*, pp. 1–5.
- [16] Y. Wang, J. Zheng, Control point removal algorithm for *T*-spline surfaces, *Geom. Model. Process.* (2006) 385–396.
- [17] Y. Li, W. Chen, J. Zheng, Y. Cai, *T*-spline multi-point removal, in: *Asian Conference on Design and Digital Engineering, 2011*, pp. 266–271.
- [18] A. Nasri, K. Sinno, J. Zheng, Local *T*-spline surface skinning, *Vis. Comput.* 28 (6–8) (2012) 787–797.
- [19] J. Zheng, Y. Wang, Periodic *T*-splines and tubular surface fitting, in: *Curves and Surfaces 2011*, in: LNCS, 6920, 2011, pp. 731–746.
- [20] N. Dyn, D. Levin, J. Gregory, A 4-point interpolatory subdivision scheme for curve design, *Comput. Aided Geom. Design* 4 (4) (1987) 257–268.
- [21] G. Farin, D. Hansford, *The Essentials of CAGD*, AK Peters, 2000.
- [22] E. Lee, Choosing nodes in parametric curve interpolation, *Comput.-Aided Des.* 21 (6) (1989) 363–370.
- [23] F. Jaillet, B. Shariat, D. Vandorpe, Periodic *B*-spline surface skinning of anatomic shapes, in: *Proceedings of 9th Canadian Conference in Computational Geometry, 1997*, pp. 199–210.