Available online at www.sciencedirect.com

ScienceDirect

Computer-Aided Design 39 (2007) 1134–1142

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Adaptive patch-based mesh fitting for reverse engineering

Hongwei Lin*, Wei Chen, Hujun Bao

State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310027, China

Received 13 May 2007; accepted 1 October 2007

Abstract

In this paper, we propose a novel adaptive mesh fitting algorithm that fits a triangular model with G^1 smoothly stitching bi-quintic Bézier patches. Our algorithm first segments the input mesh into a set of quadrilateral patches, whose boundaries form a quadrangle mesh. For each boundary of each quadrilateral patch, we construct a normal curve and a boundary-fitting curve, which fit the normal and position of its boundary vertices respectively. By interpolating the normal and boundary-fitting curves of each quadrilateral patch with a Bézier patch, an initial G^1 smoothly stitching Bézier patches is generated. We perform this patch-based fitting scheme in an adaptive fashion by recursively subdividing the underlying quadrilateral into four sub-patches. The experimental results show that our algorithm achieves precision-ensured Bézier patches with G^1 continuity and meets the requirements of reverse engineering.

Keywords: Reverse engineering; Adaptive fitting; Fitting error; G^1 continuity; Bézier patch; Triangular mesh

1. Introduction

In reverse engineering, the reconstructed triangular mesh from a point cloud contains many vertices and needs to be fitted with a smooth surface, e.g., smooth parametric patches. The triangular mesh is first segmented into a sequence of quadrilateral patches (see Fig. 1(a)), each of which is a triangular mesh with four boundaries. A set of Bézier (B-spline) patches are then built to fit the mesh vertices of these quadrilateral patches with two requirements, i.e., the fitting error of each patch and the G^1 continuity between each pair of neighboring patches. Satisfying these two requirements simultaneously can be achieved with an adaptive fitting scheme. That is, if the fitting error of one Bézier (B-spline) patch is beyond a given tolerance, its corresponding quadrilateral patch must be subdivided into four patches, which are further fitted with four smoothly stitched Bézier (B-spline) patches. This procedure proceeds until the fitting error of each Bézier (B-spline) patch reaches a given tolerance.

The continuous stitching problem among neighboring patches arises in interpolating a quadrangle curve mesh (see Fig. 1(b)) with parametric patches. Normally, the quadrangle curve mesh is manually generated by users, for instance, during designing a car body or ship hull. It consists of a set of smooth curves, and has fewer vertices than the reconstructed triangular mesh in reverse engineering. Here, the Bézier (B-spline) patches that interpolate these smooth curves are only required to be G^1 continuously stitched. Adaptive fitting is unnecessary in this case. Therefore, we will present the related work in Section 1.1 mainly on the continuity problem.

In fact, there has been much work which cares about the fitting error, but results in surfaces with weak continuity. For instance, Milroy et al. [1] proposed a B-spline surface fitting approach that leads to an uncomfortable visual appearance due to the lack of smoothness. Eck and Hoppe [2] developed an automatic method for fitting irregular meshes using bi-cubic Bézier patches. Its resulting surface has ε–G^1 continuity. Although the algorithm proposed by Krishnamurthy and Levoy [3] can fit B-spline surfaces with arbitrary topology, there is little discussion on the continuity of the resulting B-spline surfaces. Shi and Wang [4] introduced a local scheme for constructing convergent G^1 (not true G^1) smooth bi-cubic B-spline patches with interior single knots over a given arbitrary quadrangular partition of a polygonal model. Note that all these methods are incapable of adaptive fitting, and their fitting error cannot be guaranteed. However, in reverse engineering,

[*] Corresponding author. Tel.: +86 571 88206681x518; fax: +86 571 88206680.

E-mail address: hwlin@cad.zju.edu.cn (H. Lin).

0010-4485/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2007.10.002

precision-ensured fitting is important to the downstream CAD applications.

Till now, all work with adaptive fitting capability either makes use of triangular Bézier–Bernstein patches [5,6], or employs T-spline surfaces [7]. Whereas the quadrilateral Bézier patch is more preferred because it satisfies the NURBS standard, and is more popular than the triangular patch and T-spline in reverse engineering.

In this paper, a new *patch-by-patch* scheme is proposed to construct G^1 continuously stitching Bézier patches for the purpose of adaptively fitting the mesh vertices of the quadrilateral patches with precision-ensured results. We compute the normal vectors at the mesh vertices of the triangular mesh by averaging the normal vectors of their adjacent triangular patches. Along each boundary of each quadrilateral patch, the normal vectors form a fence which encloses and separates each quadrilateral patch (see Fig. 1(c)). This is different from the quadrangle curve mesh (Fig. 1(b)). We also generate a *normal curve* for each boundary by fitting the normal vectors with a quadratic Bézier curve. The normal curves make it possible to construct G^1 continuously stitching Bézier patches in a patchwise way. The key to the feasibility of the *patch-by-patch* scheme lies in that each fitting Bézier patch interpolates the normal curves on its four boundaries. Thus, adjacent Bézier patches share the same normal vectors on the common boundary, and they are tangent plane continuous (that is, G^1 continuous) along the boundary [8].

More importantly, the *patch-by-patch* scheme makes the adaptive fitting feasible, because the G^1 continuity of the patch stitching on the T-conjunctions which are caused by recursive subdivision, can be achieved by interpolating the normal curve on each boundary of every sub-quadrilateral patch. Surely, the fitting error is improved in an adaptive fashion.

The rest of this paper is organized as follows. In Section 1.1, we briefly review the related work. The overview of our approach is presented in Section 2. In Section 3, we introduce how to construct the normal curve mesh and boundary-fitting curve mesh. We present our *patch-by-patch* scheme, and show how to construct initial Bézier patches over the quadrilateral patches in Section 4. The adaptive fitting approach is described in Section 5. The experimental results are given in Section 6. Finally, we conclude the whole paper in Section 7.

1.1. Related work

Here, we briefly review the related work on interpolating a quadrangle curve mesh using G^1 smoothly stitching parametric patches. In general, these methods pay more attention on continuously stitching, and neglect the capability of ensuring the fitting error. They can be roughly classified into two categories according to the patch type: Bézier patch and B-spline patch.

One pioneering work using Bézier patch was proposed by van Wijk [9] for generating a smooth surface over a non-rectangular mesh with bi-cubic patches. Shirman and Séquin [10,11] employed five bi-cubic patches to interpolate each quadrilateral in a mesh of cubic curves. Peters [12] proposed a local ε–G^1 continuity scheme, to construct a smooth spline surface using bi-quadratic and bi-cubic Bézier patches. Later, Peters [13] constructed a G^1 smooth bi-quartic Bézier surface over a refined network of quadrilateral sub-cells generated by the midpoint mesh refinement technique. By subdividing each Bézier patch into nine small patches, Ma and Peng [14] obtained a G^1 smooth surface. Reif [15] generated G^1 smooth surfaces using bi-quadratic rectangular Bézier patches over semi-regular meshes. Ye and Nowacki [16] employed G^1 smooth bi-quintic Bézier patches to interpolate rectangular cubic curve meshes. Ye [17] also extended this method for constructing G^2 Bézier surfaces by interpolating a given G^2 quintic curve meshes.

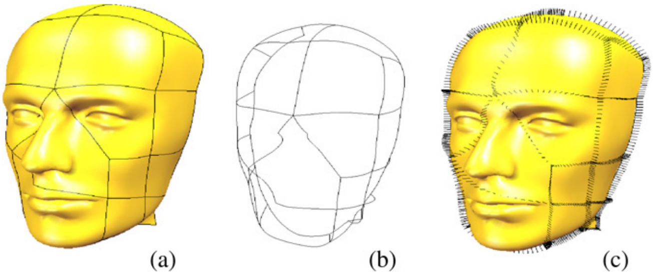

Fig. 1. (a) A triangular mesh is segmented into a set of quadrilateral patches. (b) The quadrangle curve mesh formed by the boundaries of the quadrilateral patches. (c) The normal curves enclose and separate all quadrilateral patches like a fence.

In terms of B-spline patches, Peters [18] constructed G^1 smooth bi-cubic B-spline patches with interior double knots generated by Catmull-Clark subdivision. Shi and Wang [19] developed a local processing scheme for constructing G^1 smooth bi-cubic B-spline surfaces with at least two pairs of interior double knots. Further, Shi et al. [20] improved this method to construct G^1 smooth bi-quartic B-spline patches with one pair of interior double knots. More recently, Shi et al. [21] proposed to constructs G^1 smooth B-spline surfaces with single interior knots over arbitrary topology. Another work by Kruth and Kerstens [22] incorporates positional, tangential or curvature continuity conditions with non-uniform rational B-splines in the CAD modeling of free-form surfaces.

2. Overview

Suppose that there is an oriented triangular mesh M. The normal vector at each vertex is computed by normalizing the average of the normal vectors of its adjacent triangular faces. The triangular mesh M is segmented into a set of quadrilateral patches, which are triangular mesh patches with four boundaries (Fig. 2(a)). The boundaries of all patches compose a quadrangle curve mesh Q (see Fig. 1(b)).

The adaptive mesh fitting approach includes the following steps:

1. Fit the normal vectors at the mesh vertices on each boundary of the quadrangle curve mesh Q with a quadratic Bézier curve, called a *normal curve* (see Fig. 2(b)).
2. Fit the mesh vertices on each boundary of the quadrangle curve mesh Q with a quintic Bézier curve, called a

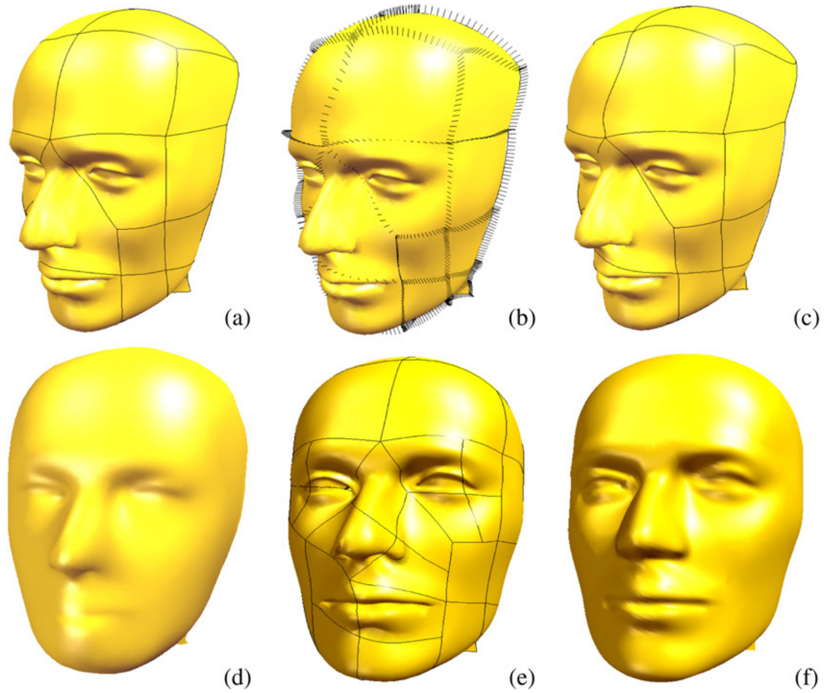

Fig. 2. (a) The quadrilateral patches. (b) The normal curve mesh. (c) The boundary-fitting curve mesh. (d) The initial G^1 continuously stitched Bézier patches that fit the quadrilateral patches of (a). (e) The final curve mesh with the adaptive fitting scheme. (f) The final result with the given fitting precision 2.1×10^4.

boundary-fitting curve, which is perpendicular to the normal curve on the boundary (see Fig. 2(c)).

3. Fit each quadrilateral patch with a bi-quintic Bézier patch in a patchwise fashion, by interpolating the boundary-fitting curves and the normal curves on four boundaries. Adjacent Bézier patches have the same normal vectors on their common boundary, and they are G^1 continuity along the boundary [8] (Fig. 2(d)).
4. Perform fitting adaptively till the fitting error is below a given tolerance. That is, if the fitting error of one Bézier patch is larger than the given tolerance, the corresponding quadrilateral patch is subdivided into four quadrilateral patches. A set of new normal curves and boundary-fitting curves are computed on the newly generated boundaries and the fitting is carried out on the new quadrilateral patches recursively (Fig. 2(e) and (f)).

The procedure of the adaptive mesh fitting is illustrated in Fig. 2. Fig. 2(a) depicts the quadrilateral patches, whose boundaries form the quadrangle curve mesh Q. Fig. 2(b) shows the normal curve mesh. For the purposes of display, each normal curve is sampled into a list of normal vectors, and placed on the corresponding boundary. Fig. 2(c) shows the boundary-fitting curve mesh. The initial G^1 continuously stitching Bézier patches after the first fitting is shown in Fig. 2(d). After recursively fitting, we get a complete curve mesh (Fig. 2(e)), and G^1 smoothly stitching Bézier patches (Fig. 2(f)) with the fitting error 2.1×10^{-4}.

It should be pointed out that mesh fitting requires to segment the mesh into pure quadrilateral patches, whose boundaries should pass the feature edges. This is indeed a strict requirement. Although recent advances [23] make it amenable to segment the mesh into pure quadrilateral patches, the boundaries of the resultant patches fail to pass the feature edges. For more details on segmentation, please refer to [23]. Our method is semi-automatic and yields the desirable segmentation. It produces a boundary between two user-selected corners, and the boundary passes the feature edges while making the total length as short as possible.

3. The normal and boundary-fitting curves

The boundaries of the quadrilateral patches segmented from M compose a quadrangle curve mesh Q. Each boundary of Q links two corners $\boldsymbol{p}_0$ and $\boldsymbol{p}_m$ of Q. We denote the normalized normal vectors at $\boldsymbol{p}_0$ and $\boldsymbol{p}_m$ as $\boldsymbol{n}_0$ and $\boldsymbol{n}_m$. Suppose the boundary consists of a list of vertices $\{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_m\}$, which have been parameterized with the normalized accumulated chord length method, i.e., the parameter of $\boldsymbol{p}_0$ is 0, and the parameter of $\boldsymbol{p}_m$ is 1.0. The corresponding normalized normal vectors are denoted as $\{\boldsymbol{n}_0, \boldsymbol{n}_1, \ldots, \boldsymbol{n}_m\}$.

3.1. The construction of the normal curve

We fit the normalized normal vectors along each boundary with a quadratic Bézier curve $\boldsymbol{N}(t) = \sum_{i=0}^{2} N_i B_i^2(t), t \in [0, 1]$. The resultant normal curve has three control vectors, $\boldsymbol{N}_0$, $\boldsymbol{N}_1$, and $\boldsymbol{N}_2$, which can be solved by:

$$\min \sum_{j=0}^{m} \left\| \sum_{i=0}^{2} N_i B_i^2(t_j) - \boldsymbol{n}_j \right\|^2, \tag{1}$$

with two additional linear constraints:

(1.1) The curve interpolates $\boldsymbol{n}_0$ and $\boldsymbol{n}_m$ at its two corners, so that $\boldsymbol{N}_0 = \boldsymbol{n}_0$, and $\boldsymbol{N}_2 = \boldsymbol{n}_m$.

(1.2) Three control vectors $\boldsymbol{N}_0$, $\boldsymbol{N}_1$, and $\boldsymbol{N}_2$ are linear independent, namely, the 3×3 determinant formed by them is nonzero.

When $\boldsymbol{n}_0$ and $\boldsymbol{n}_m$ are colinear, the constraint (1.2) no longer holds. In this case, we make a slight disturbance to one of them.

Note that the normal curve can be constructed in a piecewise way. In addition, the degree of the normal curve influences the degree of the boundary-fitting curve presented in Section 3.2. The lower the degree of the normal curve is, the lower that of the boundary-fitting curve.

3.2. The construction of the boundary-fitting curve

The boundary-fitting curve, $\boldsymbol{C}(t) = \sum_{i=0}^{5} \boldsymbol{C}_i B_i^5(t)$, is constructed after computing the normal curve mesh, by fitting the mesh vertices on each boundary of Q, with the following least-square optimization:

$$\min \sum_{j=0}^{m} \left\| \sum_{i=0}^{5} \boldsymbol{C}_i B_i^5(t_j) - \boldsymbol{p}_j \right\|^2. \tag{2}$$

Besides, there are three additional linear constraints:

(2.1) The curve interpolates the two corners of the boundary.
(2.2) The curve is perpendicular to the normal curve on the boundary.
(2.3) The twist constraint at corners (also called twist compatibility conditions).

The constraint (2.2) means that the dot product between the normal curve and the derivative of the boundary-fitting curve equals zero:

$$\boldsymbol{N}(t) \cdot \boldsymbol{C}'(t) = \sum_{k=0}^{6} \frac{5B_k^6(t)}{\binom{6}{k}} \sum_{i+j=k} \binom{4}{i} \binom{2}{j} (\boldsymbol{C}_{i+1} - \boldsymbol{C}_i) \cdot \boldsymbol{N}_j = 0, \quad t \in [0, 1]. \tag{3}$$

Here, $\binom{i}{j}$ is the coefficient of the binomial. Due to the linear independence of the Bernstein basis, it is identical to:

$$\sum_{i+j=k} \binom{4}{i} \binom{2}{j} (\boldsymbol{C}_{i+1} - \boldsymbol{C}_i) \cdot \boldsymbol{N}_j = 0, \quad k = 0, 1, \ldots, 6. \tag{4}$$

The twist constraint (2.3) is introduced to solve the vertex enclosure problem [24] which will lead to the tangent constraint and the twist constraint. Because the tangent constraint is contained in the constraint (2.2) (see Eq. (4)), we only need to handle the twist constraint. We will explain the derivation of the twist constraint in Section 4.2. For more details for the twist vector, please refer to ([25], pp. 247–248).

The twist constraint is given at each corner of Q. All corners fall into two categories: the open corners and the closed corners. The former lies on the border of Q, and the latter lies inside the mesh. As illustrated in Fig. 3(a), the closed corner $\boldsymbol{C}$ is adjacent to n pieces of normal curves and n pieces of boundary-fitting curves. Suppose $\{\boldsymbol{N}_1^i, \boldsymbol{C}_1^i, i = 0, 1, \ldots, n-1\}$ are the control points of the normal and boundary-fitting curves respectively, and are the nearest ones to $\boldsymbol{C}$. Then, the twist constraint at the closed corner $\boldsymbol{C}$ is defined as:

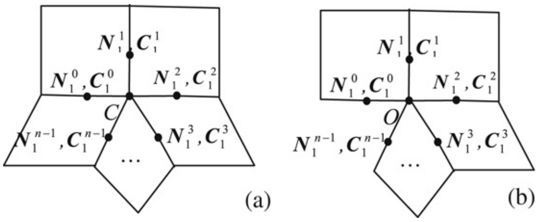

Fig. 3. (a) A closed corner C. (b) An open corner O.

$$(\boldsymbol{C}_1^i - \boldsymbol{C}) \cdot \boldsymbol{N}_1^{(i+1) \bmod n} = (\boldsymbol{C}_1^{(i+1) \bmod n} - \boldsymbol{C}) \cdot \boldsymbol{N}_1^i, \quad i = 0, 1, \ldots, n-1, \tag{5}$$

where $((i + 1) \bmod n)$ means the result of $(i + 1)$ modulo n. There are n equations in Eq. (5), each of which corresponds to one angle formed by consecutive boundary-fitting curves. Likewise, at the open corner $\boldsymbol{O}$, which is adjacent to n pieces of normal curves and n pieces of boundary-fitting curves, the twist constraint contains $n - 1$ equations (Fig. 3(b)):

$$(\boldsymbol{C}_1^i - \boldsymbol{O}) \cdot \boldsymbol{N}_1^{i+1} = (\boldsymbol{C}_1^{i+1} - \boldsymbol{O}) \cdot \boldsymbol{N}_1^i, \quad i = 0, 1, \ldots, n-2. \tag{6}$$

The twist constraint (5) or (6) makes the consecutive boundary-fitting curves correlative, and hence the boundary-fitting curve mesh has to be constructed globally. If the fitting error of the boundary-fitting curves is large or beyond a predefined threshold, the quadrangle mesh should be subdivided. The boundaries of the new quadrangle mesh are fitted again to improve the fitting error. Another way to improve the fitting error is to raise the degrees of the normal curve and the boundary-fitting curve.

It is well known that a linear system of equations is consistent if and only if both of its coefficient matrix and its augment matrix have the same rank. Suppose the quadrangle mesh Q has l pieces of boundaries, we will show in Appendix A that the coefficient matrix of the linear system which contains l groups of (2.1)–(2.3) is of full rank.

3.3. The initial values

To compute the boundary-fitting curve, appropriate initial values are required for the least square fitting with the linear constraints. We first fit the mesh vertices at each boundary using a quintic Bézier curve, solely with constraint (2.1). The control points of the fitted curve are employed as the initial values to solve the fitting problems with the linear constraints (2.1)–(2.3). With this scheme, reasonable boundary fitting curves can be easily computed as shown in Fig. 4. However, in the case that the initial values violate the constraints, the constrained optimization problem (2) is solved directly without the initial values.

4. Patch-by-patch fitting

Before performing the fitting, the vertices of each quadrilateral patch $\{\boldsymbol{p}_{ij}\}$ are parameterized by the parameterization

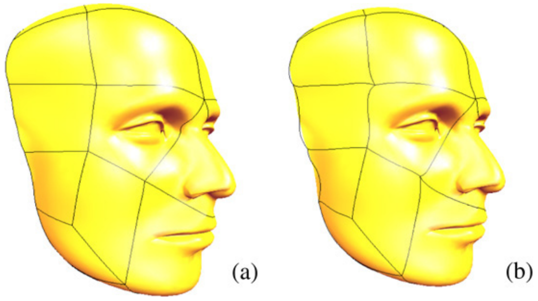

Fig. 4. (a) The curve mesh computed by solving the least square fitting problem (2) with the constraint (2.1). (b) The boundary fitting-curve mesh computed by using the control points of (a) as the initial values.

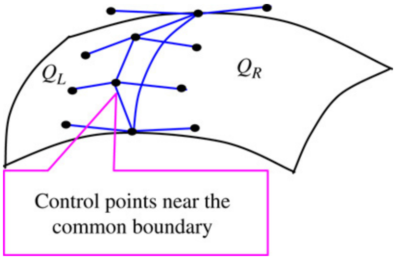

Fig. 5. With the pair-by-pair scheme, the control points of two adjacent Bézier patches must be determined simultaneously.

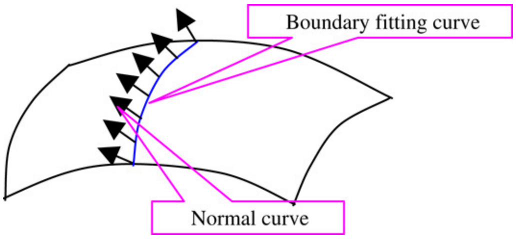

Fig. 6. With our patch-by-patch approach, Bézier patches can be constructed individually, each of which interpolates a shared boundary-fitting curve and normal curve.

method proposed in [26], which produces two groups of nearly orthogonal iso-parameter lines.

4.1. The patch-by-patch scheme

To construct G^1 smoothly stitching Bézier (B-spline) patches over a quadrangle curve mesh, previous methods determine the control points of two adjacent Bézier (B-spline) patches simultaneously (Fig. 5). This is because traditional G^1 continuous condition between Bézier (B-spline) patches is developed in a pairwise way. We call this way a *pair-by-pair scheme*.

Note that two adjacent smooth patches are G^1 continuous, or equivalently, tangent plane continuous, if and only if they have the same boundary curve, and the same normal vectors on the boundary [8]. In other words, if two smooth patches interpolate both the same boundary curve and the same normal vector curve, they are G^1 continuous along the common boundary (see Fig. 6). Based on this investigation, we propose a new *patch-by-patch* scheme for constructing G^1 smoothly stitching Bézier patches. Our key idea is to individually construct a Bézier patch which interpolates both the boundary-fitting curves and the normal curves on its four boundaries. Because two adjacent Bézier patches have the same normal vectors along their common boundary, they are G^1 continuous along the boundary.

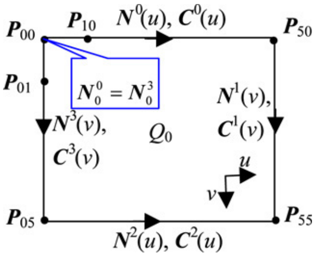

Fig. 7. Sketch of a quadrilateral patch and the normal curves and boundary-fitting curves on its four boundaries.

4.2. Patch-based fitting

Suppose we have four normal curves and four boundary-fitting curves on the four boundaries of the quadrilateral patch Q_0, as illustrated in Fig. 7:

$$\begin{aligned} \boldsymbol{N}^k(t) &= \sum_{i=0}^{2} \boldsymbol{N}_i^k B_i^2(t), \\ \boldsymbol{C}^k(t) &= \sum_{i=0}^{5} \boldsymbol{C}_i^k B_i^5(t), \quad k = 0, 1, 2, 3, t \in [0, 1]. \end{aligned} \tag{7}$$

We construct a bi-quintic Bézier patch over the quadrilateral patch $Q_0 = \{\boldsymbol{p}_{ij}\}$:

$$\boldsymbol{P}_0(u, v) = \sum_{i=0}^{5} \sum_{j=0}^{5} \boldsymbol{P}_{ij} B_i^5(u) B_j^5(v), \tag{8}$$

by solving a least square fitting problem:

$$\min \sum_{ij} \left\| \boldsymbol{P}_0(u_i, v_j) - \boldsymbol{p}_{ij} \right\|^2, \tag{9}$$

with the following linear constraints:

(3.1) The patch interpolates four boundary fitting curves.
(3.2) The patch interpolates four normal curves.

To satisfy the constraint (3.1), we only need to set the control points of four boundary-fitting curves to be the corresponding boundary control points of P_0.

However, it is not straightforward to meet the constraint (3.2). Remember that the boundary- fitting curve, e.g., $\boldsymbol{C}^0(u)$ in Fig. 7, is perpendicular to the normal curve, e.g., $\boldsymbol{N}^0(u)$. This means that the dot product between the u-directional partial-derivative vector curve at the top boundary $\{v = 0, 0 \le u \le 1\}$ and the normal curve $\boldsymbol{N}^0(u)$ equals zero. According to the constraint (3.2), the dot product between the v-directional partial-derivative vector curve at the top boundary and the normal curve $\boldsymbol{N}^0(u)$ should be zero too. Based on the linear independence of the Bernstein basis, it yields:

$$\sum_{i+j=k} \binom{5}{i} \binom{2}{j} (\boldsymbol{P}_{i1} - \boldsymbol{P}_{i0}) \cdot \boldsymbol{N}_j^0 = 0, \quad k = 1, 2, \ldots, 6. \tag{10}$$

Similarly, we have:

$$\sum_{i+j=k} \binom{5}{i}\binom{2}{j}(\boldsymbol{P}_{i5}-\boldsymbol{P}_{i4})\cdot \boldsymbol{N}_j^2 = 0, \qquad k=1,2,\ldots,6; \quad (11)$$

$$\sum_{i+j=k} \binom{5}{i}\binom{2}{j}(\boldsymbol{P}_{1i}-\boldsymbol{P}_{0i})\cdot \boldsymbol{N}_j^3 = 0, \qquad k=1,2,\ldots,6; \quad (12)$$

$$\sum_{i+j=k} \binom{5}{i}\binom{2}{j}(\boldsymbol{P}_{5i}-\boldsymbol{P}_{4i})\cdot \boldsymbol{N}_j^1 = 0, \qquad k=1,2,\ldots,6. \quad (13)$$

To deduce the twist constraint (2.3) introduced in Section 3.2, we let $k = 1$ in Eqs. (10) and (12), yielding:

$$5(\boldsymbol{P}_{11}-\boldsymbol{P}_{10})\cdot\boldsymbol{N}_0^0 + 2(\boldsymbol{P}_{01}-\boldsymbol{P}_{00})\cdot\boldsymbol{N}_1^0 = 0,$$
$$\text{and} \quad 5(\boldsymbol{P}_{11}-\boldsymbol{P}_{01})\cdot\boldsymbol{N}_0^3 + 2(\boldsymbol{P}_{10}-\boldsymbol{P}_{00})\cdot\boldsymbol{N}_1^3 = 0. \quad (14)$$

According to the constraint (3.1), the boundary control vectors $\boldsymbol{P}_{00}$, $\boldsymbol{P}_{10}$, and $\boldsymbol{P}_{01}$ are known. Two equations in (14) have the same unknown twist vector $\boldsymbol{P}_{11}$ and have the same coefficients $5\boldsymbol{N}_0^0 = 5\boldsymbol{N}_0^3$ (see Fig. 7). Because these two equations belong to a linear system, their constant items must be identical to make the linear system compatible, resulting in:

$$2(\boldsymbol{P}_{01}-\boldsymbol{P}_{00})\cdot\boldsymbol{N}_1^0 - 5\boldsymbol{P}_{10}\cdot\boldsymbol{N}_0^0$$
$$= 2(\boldsymbol{P}_{10}-\boldsymbol{P}_{00})\cdot\boldsymbol{N}_1^3 - 5\boldsymbol{P}_{01}\cdot\boldsymbol{N}_0^3. \quad (15)$$

Note that $(\boldsymbol{P}_{10}-\boldsymbol{P}_{00})\cdot\boldsymbol{N}_0^0 = (\boldsymbol{P}_{01}-\boldsymbol{P}_{00})\cdot\boldsymbol{N}_0^3 = 0$. By adding $5\boldsymbol{P}_{00}\cdot\boldsymbol{N}_0^0$ and $5\boldsymbol{P}_{00}\cdot\boldsymbol{N}_0^3$ to both sides of Eq. (15), it becomes:

$$(\boldsymbol{P}_{01}-\boldsymbol{P}_{00})\cdot\boldsymbol{N}_1^0 = (\boldsymbol{P}_{10}-\boldsymbol{P}_{00})\cdot\boldsymbol{N}_1^3. \quad (16)$$

Eq. (16) implies that two equations in (14) are identical. Thus we only keep one of them in the linear constraint system. As illustrated in Fig. 7, Eq. (16) corresponds to the angle formed by the boundary fitting curves $\boldsymbol{C}^0(u)$ and $\boldsymbol{C}^3(v)$. By constructing equations corresponding to other angles adjacent to $\boldsymbol{P}_{00}$ in a similar way, we can get the twist constraint (2.3) at $\boldsymbol{P}_{00}$, as described in Section 3.2 (see Fig. 3 and Eqs. (5) and (6)).

There are four pairs of identical equations in Eqs. (10)–(13) for the twist constraint (2.3), that is, Eq. (10) with $k = 1$ and Eq. (12) with $k = 1$, Eq. (10) with $k = 6$ and Eq. (13) with $k = 1$, Eq. (11) with $k = 1$ and Eq. (12) with $k = 6$, and Eq. (11) with $k = 6$ and Eq. (13) with $k = 6$. In each pair, only one equation is kept in the linear constraint system (3.2). In Appendix B, we will show that the coefficient matrix for the linear constraint system (3.1) and (3.2) is of full rank, and the linear system is compatible.

As stated in Section 3.3, the initial values are important to solve the least square fitting problem with the linear constraints. By solving the fitting problem (9) with the sole constraint (3.1), we get a bi-quintic patch. The control points of the patch are taken as the initial value. However, if the initial value violates the constraints, it is discarded. The fitting problem (9) is solved directly without the initial value.

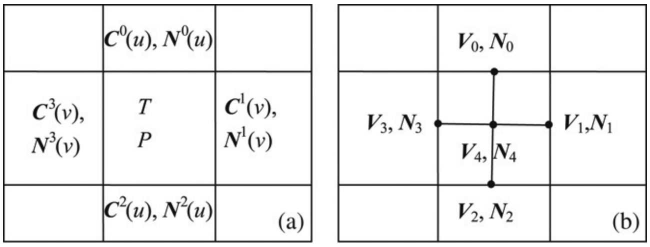
$\boldsymbol{C}^0(u), \boldsymbol{N}^0(u)$
$\boldsymbol{C}^3(v)$,
$\boldsymbol{N}^3(v)$
T
P
$\boldsymbol{C}^1(v)$,
$\boldsymbol{N}^1(v)$
$\boldsymbol{C}^2(u), \boldsymbol{N}^2(u)$
(a)
$\boldsymbol{V}_0, \boldsymbol{N}_0$
$\boldsymbol{V}_3, \boldsymbol{N}_3$
$\boldsymbol{V}_1, \boldsymbol{N}_1$
$\boldsymbol{V}_4, \boldsymbol{N}_4$
$\boldsymbol{V}_2, \boldsymbol{N}_2$
(b)

Fig. 8. (a). The bi-quintic Bézier patch P associated with the quadrilateral patch T. (b) T is subdivided into four patches. $\boldsymbol{N}^i$ and $\boldsymbol{C}^i$, $i = 0, 1, 2, 3$, are the normal and boundary-fitting curves, respectively.

5. Adaptive fitting

In reverse engineering, the fitting error is of essential importance. Normally, fitting triangular mesh within given tolerance requires that the fitting method has adaptive fitting capability. Our *patch-by-patch* fitting scheme allows us to perform adaptive fitting till the fitting error of every Bézier patch meets the tolerance requirement.

Suppose that the fitting error of a bi-quintic Bézier patch P, associated with a quadrilateral patch T, is larger than the pre-defined tolerance, as shown in Fig. 8(a). P interpolates four boundary-fitting curves $\boldsymbol{C}^i$, $i = 0, 1, 2, 3$ and four normal curves $\boldsymbol{N}^i$, $i = 0, 1, 2, 3$. To perform adaptive fitting on T, we first uniformly subdivide each boundary fitting curve and each normal curve into four curves with the de Casteljau algorithm [27]. It is apparent that each new boundary fitting curve is orthogonal to the corresponding new normal curve, because only the algebraic representation is changed, without touching their geometric positions.

We denote the middle points of four boundary fitting curves $\boldsymbol{C}^i$ as $\boldsymbol{V}_i$, and the middle points of four normal curves $\boldsymbol{N}^i$ as $\boldsymbol{N}_i$ $(i = 0, 1, 2, 3)$ (see Fig. 8(b)). By connecting two pairs of points, $\boldsymbol{V}_0$ and $\boldsymbol{V}_2$, $\boldsymbol{V}_1$ and $\boldsymbol{V}_3$ with two lines, and projecting them onto the quadrilateral patch T, we generate two projecting curves on the patch. The two projecting curves intersect at the center corner $\boldsymbol{V}_4$ with the normal vector $\boldsymbol{N}_4$, and can be subdivided into four curves. Meanwhile, they also segment T into four quadrilateral patches (see Fig. 8(b) and Fig. 2(e)).

Thereafter, the normal curves on the newly generated four curves are constructed by the algorithm presented in Section 3.1, by interpolating $\boldsymbol{N}_0$ and $\boldsymbol{N}_4$, $\boldsymbol{N}_1$ and $\boldsymbol{N}_4$, $\boldsymbol{N}_2$ and $\boldsymbol{N}_4$, $\boldsymbol{N}_3$ and $\boldsymbol{N}_4$, respectively. And the boundary-fitting curves are constructed with the method introduced in Section 3.2, by interpolating $\boldsymbol{V}_0$ and $\boldsymbol{V}_4$, $\boldsymbol{V}_1$ and $\boldsymbol{V}_4$, $\boldsymbol{V}_2$ and $\boldsymbol{V}_4$, $\boldsymbol{V}_3$ and $\boldsymbol{V}_4$, respectively. In addition, four bi-quintic Bézier patches fitting four new quadrilateral patches are generated individually using our *patch-by-patch* scheme. In this way, not only the fitting error is improved, but also the new generated patches are G^1 smoothly stitched.

It should be pointed out that, in the quadrangle mesh, if one boundary is adjacent to one of its siblings and a descending quadrilateral patch, the normal curve and boundary-fitting curve along the boundary must be kept unchanged. However, if both of its adjacent quadrilateral are subdivided, the normal curve and boundary-fitting curve would be re-computed, which improves the fitting error. Accordingly, the

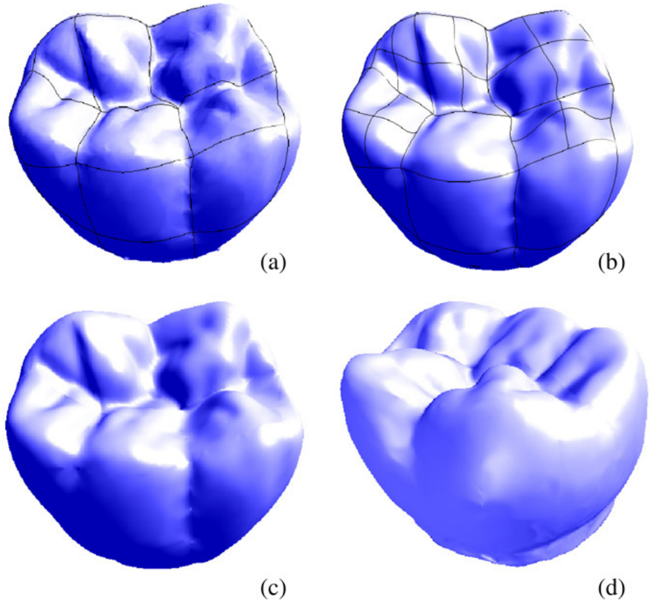

Fig. 9. (a) The input Tooth model with segmented quadrilateral patches. (b) The final fitted quadrilateral patches with boundary-fitting curves, where five quadrilateral patches are subdivided. (c) The fitted Bézier patches from one view; (d) from another view.

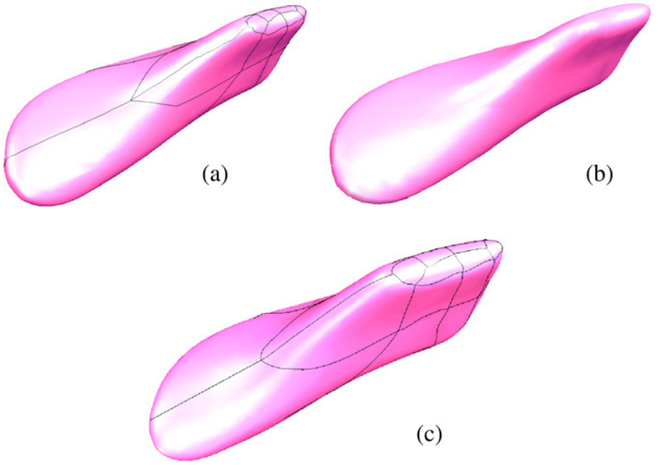

Fig. 10. (a) The input Shoe model and the initial quadrilateral patches. (b) The fitted Bézier patches. (c) The fitted Bézier patches with boundary-fitting curves.

four adjacent quadrilaterals should be re-fitted for improving the precision as well.

We perform the adaptive fitting recursively till the fitting error of every fitting patch achieves the pre-defined tolerance. Finally, the triangular mesh is fitted with a G^1 smooth surface with the pre-defined fitting error.

6. Results

We have implemented the proposed approach with Visual C++ 6.0 by C++ and Matlab mixed programming, on a PC with 2.0 GHz CPU and 512 MB memory. That is, C++ program calls Matlab function *lsqlin*, which solves the optimization problem. We have tested several practical examples to validate the efficiency and effectiveness of our method. It should be pointed out that the quality of the triangular mesh affects the fitting result. Higher quality mesh leads to the better fitting result.

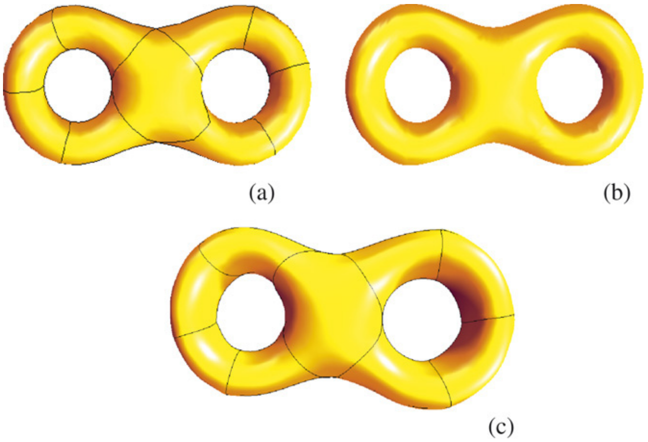

Fig. 11. (a) The input Eight model and the initial quadrilateral patches. (b) The fitted Bézier patches. (c) The fitted Bézier patches with boundary-fitting curves.

The first example is a Head model shown in Fig. 2. Beginning from an initial patch subdivision with 20 pieces, our adaptive fitting approach achieves a G^1 continuous surface with given fitting error. The quadrilateral patches near the nose part are subdivided into a sequence of small quadrilateral patches.

Other examples are illustrated in Figs. 9–11. In Fig. 9, the Tooth model is subdivided into 26 pieces of quadrilateral patches. Because the top of the dataset contains fine details, five quadrilateral patches are further divided and fitted. On the other hand, the Shoe and Eight models shown in Figs. 10 and 11 are smooth, and no subdivision occurs in the final results. Specifically, the Eight model is topologically complex with genus 2 and is segmented into 18 pieces of quadrilateral patches. We summarize the model configuration, timing statistics and fitting error in Table 1. The fourth and fifth columns list the patch numbers of the initial segmentation and the fitting results. The times spent on curve mesh construction and patch fitting are separately reported in the sixth and seventh columns. The last column lists the sizes of the bounding box of all models.

7. Conclusions

In this paper, we propose an adaptive fitting approach for a triangular mesh with arbitrary topology. We introduce a new patch-by-patch scheme to construct G^1 smoothly stitching bi-quintic Bézier patches by exploiting the normal curves and boundary fitting curves along the patch boundaries. This scheme facilitates the adaptive fitting for the purpose of improving the fitting error. Consequently, our results are not only G^1 smoothly stitching, but also capable of satisfying the user-defined fitting error. This capability makes our approach quite useful for reverse engineering.

Acknowledgements

This work is supported by 973 program of China (no. 2004CB719400) and NSF of China (nos. 60503057, 60503056).

Table 1
Experimental configurations and results

Model	#Vertices	#Face	#Input patches	#Final patches	Curve mesh	Patch fitting	Fitting error	Bounding box ($L \times W \times H$)
Head	7579	15092	20	29	2 s	5 s	2.1×10^{-4}	$0.62 \times 0.83 \times 1.00$
Tooth	7610	15216	26	41	2 s	5 s	5.2×10^{-4}	$1.00 \times 0.81 \times 0.98$
Shoe	4569	9134	28	28	1 s	3 s	9.5×10^{-5}	$1.00 \times 0.45 \times 0.50$
Eight	931	1866	18	18	1 s	2 s	1.6×10^{-4}	$1.00 \times 0.48 \times 0.20$

Appendix A. The consistence of linear constraint system in boundary-curve fitting

Suppose the quadrangle mesh Q has l pieces of boundaries, the linear constraint system contains l groups of conditions (2.1)–(2.3). For simplicity, we integrate each constraint (2.1) into the corresponding (2.2) and (2.3), yielding the following coefficient matrix of the linear system:

$$\begin{bmatrix} A^1 & & & & \\ & \ddots & & & \\ & & A^i & & \\ & & & \ddots & \\ & & & & A^l \\ B^1 & & & & \\ & \ddots & & & \\ & & B^i & & \\ & & & \ddots & \\ & & & & B^l \end{bmatrix}, \tag{17}$$

Here, A^i is a 7×12 matrix, and B^i is a 4×12 matrix,

$$A^i = \begin{bmatrix} \boldsymbol{N}_0^i & & & \\ \boldsymbol{N}_1^i - 2\boldsymbol{N}_0^i & 2\boldsymbol{N}_0^i & & \\ \boldsymbol{N}_2^i - 8\boldsymbol{N}_1^i & 8\boldsymbol{N}_1^i - 6\boldsymbol{N}_0^i & 6\boldsymbol{N}_0^i & \\ -\boldsymbol{N}_2^i & \boldsymbol{N}_2^i - 3\boldsymbol{N}_1^i & 3\boldsymbol{N}_1^i - \boldsymbol{N}_0^i & \boldsymbol{N}_0^i \\ & -6\boldsymbol{N}_2^i & 6\boldsymbol{N}_2^i - 8\boldsymbol{N}_1^i & 8\boldsymbol{N}_1^i - \boldsymbol{N}_0^i \\ & & -2\boldsymbol{N}_2^i & 2\boldsymbol{N}_2^i - \boldsymbol{N}_1^i \\ & & & -\boldsymbol{N}_2^i \end{bmatrix}, \tag{18}$$

$$B^i = \begin{bmatrix} \boldsymbol{N}_1^{ai} & \vec{0} & \vec{0} & \vec{0} \\ -\boldsymbol{N}_1^{bi} & \vec{0} & \vec{0} & \vec{0} \\ \vec{0} & \vec{0} & \vec{0} & \boldsymbol{N}_1^{ci} \\ \vec{0} & \vec{0} & \vec{0} & -\boldsymbol{N}_1^{di} \end{bmatrix},$$

where $\boldsymbol{N}_j^i = (x_j^i, y_j^i, z_j^i)$ $(i = 1, 2, \ldots, l,\ j = 0, 1, 2)$ denotes the j-th control point of the i-th normal curve, and $\vec{0} = (0, 0, 0)$ is a null vector.

The matrix (17) can be separated into two sub-matrices. The top one is quasi-diagonal and corresponds to the constraint (2.2). Each block A^i is a coefficient matrix for Eq. (4), where the two end control points are known by integrating the constraint (2.1). The bottom sub-matrix corresponds to the constraint (2.3).

Suppose the matrix (17) has $k = m + n$ row vectors $\{\boldsymbol{r}_i, i = 0, 1, \ldots, m + n\}$, of which m vectors belong to the top sub-matrix, and n vectors lie in the bottom sub-matrix. Suppose that there are also k $\{a_i, i = 1, 2, \ldots, k\}$ real numbers, such that:

$$\sum_{i=1}^{k} a_i \boldsymbol{r}_i = \sum_{i=1}^{m} a_i \boldsymbol{r}_i + \sum_{i=m+1}^{k} a_i \boldsymbol{r}_i = 0. \tag{19}$$

Because the top sub-matrix is quasi-diagonal, we first consider its first block, which consists of seven row vectors. Note that non-zero entries from the fourth column to the ninth column in the matrix (17) only lie in the first block, or say, from the second row to the fifth row. Therefore, the system of Eq. (19) contains the following equations:

$$\begin{cases} 2a_2\boldsymbol{N}_0^0 + a_3(8\boldsymbol{N}_1^0 - 6\boldsymbol{N}_0^0) + a_4(\boldsymbol{N}_2^0 - 3\boldsymbol{N}_1^0) - 6a_5\boldsymbol{N}_2^0 = 0, \\ 6a_3\boldsymbol{N}_0^0 + a_4(3\boldsymbol{N}_1^0 - \boldsymbol{N}_0^0) + a_5(6\boldsymbol{N}_2^0 - 8\boldsymbol{N}_1^0) - 2a_6\boldsymbol{N}_2^0 = 0. \end{cases} \tag{20}$$

It is equivalent to:

$$\begin{cases} (2a_2 - 6a_3)\boldsymbol{N}_0^0 + (8a_3 - 3a_4)\boldsymbol{N}_1^0 + (a_4 - 6a_5)\boldsymbol{N}_2^0 = 0, \\ (6a_3 - a_4)\boldsymbol{N}_0^0 + (3a_4 - 8a_5)\boldsymbol{N}_1^0 + (6a_5 - 2a_6)\boldsymbol{N}_2^0 = 0. \end{cases} \tag{21}$$

Because the control vectors of the normal curve $\{\boldsymbol{N}_0^0, \boldsymbol{N}_1^0, \boldsymbol{N}_2^0\}$ are independent according to the constraint (1.2) in Section 3.1, the coefficients of Eqs. (21) are all zero. It implies that $a_2 = a_3 = \cdots = a_6 = 0$.

Similarly, it can be proved that the corresponding coefficients of other blocks are all zero, that is, $a_{7i+2} = a_{7i+3} = \cdots = a_{7i+6} = 0$, $(i = 1, 2, \ldots, l - 1)$.

The remaining vectors in Eq. (19), and accordingly, the corresponding entries in the matrix (17), can be re-organized into a quasi-diagonal matrix by applying an elementary transformation. Each block corresponds to the constraint (2.3) at a corner. For example, the block corresponding to the constraints in Eq. (5) at the corner $\boldsymbol{C}$ (see Fig. 3(a)) is,

$$\begin{bmatrix} \boldsymbol{N}_0 & & & & \\ & \boldsymbol{N}_0 & & & \\ & & \ddots & & \\ & & & \boldsymbol{N}_0 & \\ & & & & \boldsymbol{N}_0 \\ \boldsymbol{N}_1^1 & -\boldsymbol{N}_1^0 & & & \\ & \boldsymbol{N}_1^2 & -\boldsymbol{N}_1^1 & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{N}_1^{n-1} & -\boldsymbol{N}_1^{n-2} \\ -\boldsymbol{N}_1^{n-1} & & & & \boldsymbol{N}_1^0 \end{bmatrix}, \tag{22}$$

where $\boldsymbol{N}_0^0 = \boldsymbol{N}_0^1 = \cdots = \boldsymbol{N}_0^{n-1} = \boldsymbol{N}_0$ are the normal vectors at the corner $\boldsymbol{C}$ (see Fig. 3(a)). In addition, because the control vectors of a normal curve are independent, the set of vectors $\{N_0, N_1^1, N_1^{n-1}\}$, $\{N_0, N_1^0, N_1^2\}, \ldots, \{N_0, N_1^{n-2}, N_1^0\}$ are independent. Therefore, the matrix (22) is of full rank. Similarly, all blocks in the remaining quasi-diagonal matrix are of full rank, and hence the remaining vectors in Eq. (19) are independent. This implies that the remaining coefficients of Eq. (19) are all zero. In conclusion, the coefficients in Eq. (19) are all zero, and the matrix (17) is of full rank.

Appendix B. The consistence of linear constraint system in patch fitting

Now we show that the coefficient matrix of the linear constraint system (3.1) and (3.2) is of full rank. Without loss of generality, four equations are removed from the system of equations (10)–(13), that is, Eq. (10) when $k = 1$, Eq. (11) when $k = 6$, Eq. (12) when $k = 6$, and Eq. (13) when $k = 1$.

After integrating the constraint (3.1) into (3.2), the boundary control points of the Bézier patch are known, leading to the coefficient matrix of the linear system:

$$\begin{bmatrix} A^1 & & & \\ & B^2 & & \\ & & B^3 & \\ & & & A^4 \end{bmatrix}. \tag{23}$$

Here,

$$A^i = \begin{bmatrix} 10N_1^i & 10N_0^i & & \\ N_2^i & 4N_1^i & 2N_0^i & \\ & 2N_2^i & 4N_1^i & N_0^i \\ & & 10N_2^i & 10N_1^i \\ & & & 5N_2^i \end{bmatrix}, \qquad B^j = \begin{bmatrix} 5N_0^j & & & \\ 10N_1^j & 10N_0^j & & \\ N_2^j & 4N_1^j & 2N_0^j & \\ & 2N_2^j & 4N_1^j & N_0^j \\ & & 10N_2^j & 10N_1^j \end{bmatrix}, \tag{24}$$

$i = 0, 3, \; j = 1, 2.$

And $\{N_0^i, N_1^i, N_2^i, i = 0, 1, 2, 3\}$ are the control points of four normal curves on the boundaries of the Bézier patch. The matrix (23) is also a quasi-diagonal matrix and each block corresponds to a coefficient matrix of the systems in Eqs. (10)–(13), respectively.

Because $\{N_0^i, N_1^i, N_2^i, i = 0, 1, 2, 3\}$ are linear independent, each block in the matrix (23) is of full rank, and thus the matrix (23) is of full rank.

References

[1] Milroy MJ, Bradley C, Vickera GW, Weir DJ. G1 continuity of B-spline surface patches in reverse engineering. Computer Aided Design 1995;27:471–8.

[2] Eck M, Hoppe H. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In: ACM Computer Graphics, SIGGRAPH. 1996. p. 325–34.

[3] Krishnamurthy V, Levoy M. Fitting smooth surfaces to dense polygon meshes. In: ACM Computer Graphics, SIGGRAPH. 1996. p. 313–24.

[4] Shi X, Wang T, Wu P, Liu F. Reconstruction of convergent G1 smooth B-spline surfaces. Computer Aided Geometric Design 2004;21:893–913.

[5] Gonzalez-Ochoa C, Peters J. Localized-hierarchy surface splines (LeSS). In: Proceedings of the 1999 Symposium on Interactive 3D Graphics. Atlanta (Georgia, United States): ACM Press; 1999. p. 7–15.

[6] Alex Y, Stefanie H, Georges-Pierre B. Smooth adaptive fitting of 3D models using hierarchical triangular splines. In: International conference on shape modeling and applications. MIT Boston: IEEE Computer Society Press; 2005. p. 13–22.

[7] Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-spline simplification and local refinement. ACM Transactions on Graphics 2004;23:276–83.

[8] Farin G. Curves and surfaces for computer aided geometric design. 4th ed. San Diego: Academic Press; 2000. p. 308.

[9] van Wijk J. Bicubic patches for approximating non-rectangular control-point meshes. Computer Aided Geometric Design 1986;3:1–13.

[10] Shirman L, Séquin C. Local surface interpolation with Bézier patches. Computer Aided Geometric Design 1987;4:279–95.

[11] Shirman L, Séquin C. Local surface interpolation with Bézier patches: Errata and improvements. Computer Aided Geometric Design 1991;8:217–22.

[12] Peter J. Constructing C1 surface of arbitrary topology using biquadratic and bicubic aplines. In: Sapidis N, editor. Designing fair curves and surfaces. SIAM; 1994. p. 277–93.

[13] Peter J. Biquartic C1-surface splines over irregular meshes. Computer-Aided Design 1995;27:895–903.

[14] Ma L, Peng Q. Smoothing of free-form surfaces with Bézier patches. Computer Aided Geometric Design 1995;12:231–49.

[15] Reif U. Biquadratic G-spline surfaces. Computer Aided Geometric Design 1995;12:193–205.

[16] Ye X, Nowacki H. G1 interpolation of rectangular cubic curve meshes using biquintic Bézier patches. In: Mullineux G, editor. The sixth IMA conference on mathematics of surfaces. UK, Brunel University: Oxford University Press; 1994. p. 429–52.

[17] Ye X. Curvature continuous interpolation of curve meshes. Computer Aided Geometric Design 1997;14:169–90.

[18] Peter J. Patching Catmull–Clark meshes. In: ACM Computer Graphics, SIGGRAPH. 2000. p. 255–8.

[19] Shi X, Wang T. G1 continuous conditions of bicubic B-spline surfaces with interior double knots. Raindrop geomagic research report, also appeared in NSF SBIR Phase II Project 'Automation creation of NURBS patches from triangulated surface' (Award# 9901627), Report 4, Appendix 2. 2001.

[20] Shi X, Yu P, Wang T. G1 continuity conditions of biquartic B-spline surfaces. Journal of Computational and Applied Mathematics 2002;144:251–62.

[21] Shi X, Wang T, Yu P. A practical construction of G1 smooth bi-quintic B-spline surfaces over arbitrary topology. Computer-Aided Design 2004;34:413–24.

[22] Kruth J-P, Kerstens A. Reverse engineering modelling of free-form surfaces from point clouds subject to boundary conditions. Journal of Materials Processing Technology 1998;76:120–7.

[23] Dong S, Bremer P-T, Garland M, Pascucci V, Hart JC. Spectral surface quadrangulation. ACM Transactions on Graphics 2006;25:1057–66.

[24] Peters J. Smooth interpolation of a mesh of curves. Constructive Approximation 1991;7:221–46.

[25] Farin G. Curves and surfaces for computer aided geometric design, a practical guide. Academic Press; 1997.

[26] Lin H, Wang G, Liu L, Bao H. Parameterization for fitting triangular mesh. Progress in Natural Science 2006;16:1214–21.

[27] Lane JM, Reisenfeld RF. A theoretical development for the computer generation and display of piecewise polynomial surface. IEEE Transactions on Pattern Analysis and Machine Intelligence 1980;2:35–46.