Computer-Aided Design 44 (2012) 1127–1140

Contents lists available at SciVerse ScienceDirect

Computer-Aided Design

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

Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines

Wei-Cheng Xie [a], Xiu-Fen Zou [a,*], Jian-Dong Yang [b], Jie-Bin Yang [b]

[a] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China

[b] State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Wuhan 430072, China

ARTICLE INFO

Article history:
Received 19 July 2011
Accepted 15 May 2012

Keywords:
Surface reconstruction
NURBS fitting
Iteration of knot vector
Projected optimization algorithm
Unorganized points

ABSTRACT

Curve or surface reconstruction is a challenging problem in the fields of engineering design, virtual reality, film making and data visualization. Non-uniform rational B-spline (NURBS) fitting has been applied to curve and surface reconstruction for many years because it is a flexible method and can be used to build many complex mathematical models, unlike certain other methods. To apply NURBS fitting, there are two major difficult sub-problems that must be solved: (1) the determination of a knot vector and (2) the computation of weights and the parameterization of data points. These two problems are quite challenging and determine the effectiveness of the overall NURBS fit. In this study, we propose a new method, which is a combination of a hybrid optimization algorithm and an iterative scheme (with the acronym HOAAI), to address these difficulties. The novelties of our proposed method are the following: (1) it introduces a projected optimization algorithm for optimizing the weights and the parameterization of the data points, (2) it provides an iterative scheme to determine the knot vectors, which is based on the calculated point parameterization, and (3) it proposes the boundary-determined parameterization and the partition-based parameterization for unorganized points. We conduct numerical experiments to measure the performance of the proposed HOAAI with six test problems, including a complicated curve, twisted and singular surfaces, unorganized data points and, most importantly, real measured data points from the Mashan Pumped Storage Power Station in China. The simulation results show that the proposed HOAAI is very fast, effective and robust against noise. Furthermore, a comparison with other approaches indicates that the HOAAI is competitive in terms of both accuracy and runtime costs.

1. Introduction

In computer-aided design, curve and surface reconstruction are used in a broad set of applications in a variety of fields. The aim of reconstruction is to construct a mathematical model that approximates an unknown curve or surface from many data points. These points are usually the scanning points from two-dimensional (2D) slices of a three-dimensional (3D) shape or the iso-parametric curves on a surface. A few approaches, such as the polygonal mesh method [1], continuous global optimization [2] and interpolation [3], have been introduced; these approaches either produced low-accuracy results or required the data points to be relatively exact. In addition, they were time-consuming. In another class of reconstruction methods, the real shape is presented through an approximation technique with a free-form parametric mathematical model that is flexible and can accurately depict a wide variety of geometries.

Among the basic parametric functions, which are fundamental elements for constructing approximate curves or surfaces, B-spline and non-uniform rational B-spline (NURBS) are generally used. An approximation with B-splines is chosen for consideration because it contains a small number of unknown parameters and consumes a small amount of computational time.

Based on the B-spline function, reconstruction with multi-patch and subdivided surfaces is often considered due to the weak requirement of the structure of the provided points. Methods such as multi-patch network, parameterization on the patches [4], and quadtree-like subdivisions [5] have been reported. Viewing the curve or the surface reconstruction as an optimization problem is also a notable class of approaches, such as reconstruction with the Levenberg–Marquardt algorithm [6], the trust region algorithm [7], the Gauss–Newton optimization method [8], the two-step genetic algorithm [9], and other optimization schemes and algorithms [10,11].

However, a B-spline surface is not recommended for some surface reconstructions because it cannot accurately represent certain complicated geometries. In contrast, NURBS provides a sophisticated and promising choice to represent a wide variety of smooth objects, especially conical surfaces, due to its flexibility and versatility. Based on the NURBS surface, the reconstructions were performed by making use of evolutionary intelligence

* Corresponding author. Tel.: +86 27 68772958; fax: +86 27 68752256.
E-mail addresses: xfzou@whu.edu.cn, zouxiufen@yahoo.com (X.-F. Zou).

0010-4485/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2012.05.004

techniques [12–15] and particle swarm optimization (PSO) [16]. However, some artificial intelligence techniques may be somewhat time-consuming because they are based on random search strategies, while the real situation easily consists of thousands of reconstructions, making them low-efficiency algorithms. The reconstruction was presented as an optimization problem by minimizing a penalty function [17]. This method consisted of computing the derivatives of the penalty function with respect to (w.r.t.) the knot vectors, which made the optimization complex and time-consuming.

To improve the accuracy and to accelerate the convergence process, this study proposes a new method that is a combination of a hybrid optimization algorithm and an iterative scheme (namely, HOAAI). It brings together a new projected optimization algorithm and an iteration technique in NURBS fitting for the reconstruction of 3D curves and surfaces from clouds of complicated data points.

This paper is structured as follows. Section 2 introduces the basic definitions and notations for NURBS fitting. Section 3 provides detailed descriptions of the HOAAI, including the determination of initial solutions, the projected optimization algorithms for the weight coefficients, and the iterative scheme of the knot vectors. In Section 4, numerical experiments are conducted for test problems. Methods for parameterizing unorganized points and numerical experiments are presented in Section 5. Finally, our discussions and conclusions are provided in Sections 6 and 7, respectively.

2. Basic definitions and notations

2.1. NURBS functions

For a given nondecreasing real-number sequence $\mathcal{T} = \{t_0, t_1, \ldots, t_{p+k}, t_{p+k+1}\}$, which is called a knot vector, the i-th B-spline basis function $N_{i,k}(u)$ of order $k+1$ (or degree k) is defined by the Cox–De Boor formula [18]

$$N_{i,0}(u) := \begin{cases} 1, & \text{if } t_i \le u < t_{i+1}, \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $0 \le i \le p$, and

$$N_{i,k}(u) := \frac{u - t_i}{t_{i+k} - t_i} N_{i,k-1}(u) + \frac{t_{i+k+1} - u}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(u). \tag{2}$$

In Eq. (2), the convention $\frac{0}{0} = 0$ is applied if necessary. The interval $[t_i, t_{i+k+1})$ is called the support of $N_{i,k}(u)$ because $N_{i,k}(u) = 0$ for $u \notin [t_i, t_{i+k+1})$. This local support property can be employed to greatly simplify the computation of the NURBS fitting. In most practical uses, the knot vector is selected to be non-uniform; in other words, each knot t_i may appear more than once, or that the knots $\{t_i\}$ ($\{t_i\}$ are the simplification of $\{t_i, i = 0, \ldots, p+k+1\}$ without confusion) are not uniformly distributed. The most common non-uniform knot vector is constructed by simultaneously repeating the end knots the same number of times as the order of the B-spline and normalizing these values in the interval [0, 1]; in other words, $t_0 = \cdots = t_k = 0$ and $t_{p+1} = \cdots = t_{p+k+1} = 1$. Because curve reconstruction is a special case of surface reconstruction, we consider only the notations related to the surface reconstruction without loss of generality.

Given a set of 3D control points $\{D_{i,j} = (D^1_{i,j}, D^2_{i,j}, D^3_{i,j}), i = 0, \ldots, p; j = 0, \ldots, q\}$ where $(p+1)(q+1)$ is the number of control points, with the corresponding weights $\{w_{i,j}, i = 0, \ldots, p; j = 0, \ldots, q\}$ and the two knot vectors $\mathcal{T} = \{t_0, t_1, \ldots, t_{p+k}, t_{p+k+1}\}$, $\mathcal{S} = \{s_0, s_1, \ldots, s_{q+l}, s_{q+l+1}\}$, a $(k+1, l+1)$-order NURBS parametric surface is defined as follows:

$$P(u, v) = \frac{\sum_{i=0}^{p}\sum_{j=0}^{q} w_{i,j} D_{i,j} N_{i,k}(u) N_{j,l}(v)}{\sum_{i=0}^{p}\sum_{j=0}^{q} w_{i,j} N_{i,k}(u) N_{j,l}(v)} \tag{3}$$

where $\{w_{i,j}, i = 0, \ldots, p, j = 0, \ldots, q\} \subset [0, \infty)$ and the parameters u, v are normalized to the interval [0, 1].

2.2. Fitting and parameterization of the data points

For a given set of 3D data points $\{Q_{i,j} = (Q^1_{i,j}, Q^2_{i,j}, Q^3_{i,j}), i = 0, \ldots, m, j = 0, \ldots, n\}$ where $(m+1)(n+1)$, which is larger than $(p+1)(q+1)$, is the number of data points, we construct an associate parameter pair (u_i, v_j) for each of the data points $Q_{i,j}$, which is called a point parameterization. The least squares(LSQ) fitting error E_{ls} is calculated as follows:

$$\begin{aligned} E_{ls} &= \sum_{i=0}^{m}\sum_{j=0}^{n} \|Q_{i,j} - P(u_i, v_j)\|^2 \\ &= \sum_{i=0}^{m}\sum_{j=0}^{n} \left\| Q_{i,j} - \frac{\sum_{\alpha=0}^{p}\sum_{\beta=0}^{q} w_{\alpha,\beta} D_{\alpha,\beta} N_{\alpha,k}(u_i) N_{\beta,l}(v_j)}{\sum_{\alpha=0}^{p}\sum_{\beta=0}^{q} w_{\alpha,\beta} N_{\alpha,k}(u_i) N_{\beta,l}(v_j)} \right\|^2. \end{aligned} \tag{4}$$

Only two vectors $\mathbf{u} = \{u_0, \ldots, u_m\}$, $\mathbf{v} = \{v_0, \ldots, v_n\}$ are required to store the information related to the point parameterization. In general, if we increase the number of control points, then this fitting error can decrease; however, it will lead to an increase in computation time. Therefore, when setting the number of control points, the number chosen is a balance between the accuracy that is needed and the runtime cost. The control points $\{D_{\alpha,\beta}\}$ can be obtained by using LSQ algorithms to solve the system in (4).

2.3. Derivatives of the NURBS function with respect to the parameters

The derivatives of the NURBS function w.r.t. the weight coefficients are easily calculated as the following:

$$\frac{\partial P(u, v)}{\partial w_{i,j}} = \frac{N_{i,k}(u) N_{j,l}(v)(D_{i,j} swn - swdn)}{swn^2} \tag{5}$$

where

$$\begin{cases} swn = swn(u, v) \triangleq \sum_{\alpha=0}^{p}\sum_{\beta=0}^{q} w_{\alpha,\beta} N_{\alpha,k}(u) N_{\beta,l}(v), \\ swdn = swdn(u, v) \triangleq \sum_{\alpha=0}^{p}\sum_{\beta=0}^{q} w_{\alpha,\beta} D_{\alpha,\beta} N_{\alpha,k}(u) N_{\beta,l}(v). \end{cases}$$

The derivative of the NURBS function w.r.t. the parameter u can be computed as the following:

$$\frac{\partial P(u, v)}{\partial u} = \frac{swn \cdot swdnu - swdn \cdot swnu}{swn^2} \tag{6}$$

where

$$\begin{cases} swnu = swnu(u, v) \triangleq \sum_{\alpha=0}^{p}\sum_{\beta=0}^{q} w_{\alpha,\beta} \frac{dN_{\alpha,k}(u)}{du} N_{\beta,l}(v), \\ swdnu = swdnu(u, v) \\ \quad \triangleq \sum_{\alpha=0}^{p}\sum_{\beta=0}^{q} w_{\alpha,\beta} D_{\alpha,\beta} \frac{dN_{\alpha,k}(u)}{du} N_{\beta,l}(v) \end{cases}$$

where $\frac{dN_{\alpha,k}(u)}{du}$ is the derivative of $N_{\alpha,k}(u)$ w.r.t. the parameter u. Cox and de Boor [18] suggested a simple and fast formula to compute this derivative when given lower-order B-spline function values, as follows:

$$\frac{dN_{\alpha,k}(u)}{du} = k\left[\frac{N_{\alpha,k-1}(u)}{t_{\alpha+k} - t_\alpha} - \frac{N_{\alpha+1,k-1}(u)}{t_{\alpha+k+1} - t_{\alpha+1}}\right]. \tag{7}$$

The derivative of $P(u, v)$ w.r.t. the parameter v can be similarly derived and we omit it here.

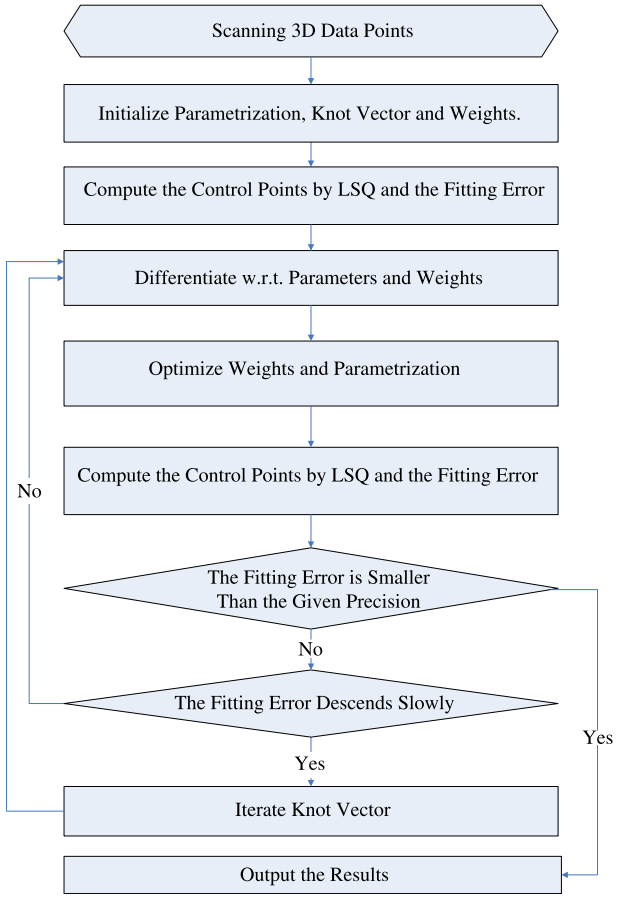

Fig. 1. Framework of the HOAAI.

3. The proposed HOAAI

In this section, we provide detailed descriptions of the proposed HOAAI, including the determination of the initial solution comprised of the parameterization and the knot vectors, the computation of the control points, the iterative scheme of the knot vectors, and the projected optimization algorithms for new weight coefficients and the new parameterization of the data points. The process is divided into the following eight steps. In the first step, the initial parameterization, the initial knot vector and the weight coefficients are computed; this step is presented in Sections 3.1 and 3.2. The second step is to apply the classical LSQ method using either the SVD or the LU technique to obtain the control points and the corresponding fitting error, which is shown in Section 3.3. The third step is to compute the derivatives of the fitting error w.r.t. all the parameters $\{u_i, v_j, w_{\alpha,\beta}\}$ shown in Section 3.4. For the fourth step, we employ the projected optimization algorithms in Section 3.5 in deriving the new point parameterization and weight coefficients. Based on these new parameter values, the control points and the fitting error are recalculated in the fifth step. The sixth step is to judge whether this new fitting error is smaller than the given precision. If it is, the results are output; otherwise, the seventh step is executed, which judges whether the fitting error descends slowly. If it does, the procedure proceeds to the eighth step to iterate the knot vectors, as presented in Section 3.6, then proceeds to the third step. Otherwise, the procedure goes directly to the third step. The entire framework of the HOAAI is presented in Fig. 1.

3.1. Parameterization of the data points

For the LSQ fitting problem (4), the initial point parameterization $\{(u_i, v_j)\}$ that is associated with the data points $\{Q_{i,j}\}$ can affect the fitting error. To obtain this initial parameterization, we simply utilize the arc length parameterization reported in the literature as uniform, centripetal and chord length parameterization for organized data points. A generalization of the method for the points $\{Q_{i,j}\}$ has been proposed by Lee [19] as follows:

$$\begin{cases} u_0^j = 0, \qquad v_i^0 = 0, \quad j = 0, \ldots, n, i = 0, \ldots, m \\ u_i^j = u_{i-1}^j + \dfrac{\|Q_{i,j} - Q_{i-1,j}\|^e}{\sum_{s=1}^{m} \|Q_{s,j} - Q_{s-1,j}\|^e}, \\ \quad i = 1, \ldots, m, j = 0, \ldots, n \\ v_i^j = v_i^{j-1} + \dfrac{\|Q_{i,j} - Q_{i,j-1}\|^e}{\sum_{t=1}^{n} \|Q_{i,t} - Q_{i,t-1}\|^e}, \\ \quad j = 1, \ldots, n, i = 0, \ldots, m \end{cases} \tag{8}$$

where $e \in [0, \infty)$. Each parameter pair (u_i, v_j) that corresponds to the data point $Q_{i,j}$ is then computed by averaging, as follows: $u_i = \frac{1}{n+1} \sum_{j=0}^{n} u_i^j$, $v_j = \frac{1}{m+1} \sum_{i=0}^{m} v_i^j$.

3.2. Initialization of the knot vector

To guarantee that the matrix $M^T \cdot M$ in the following Eq. (13) is non-singular or not ill-conditioned, the Schoenberg–Whitney conditions in [18] imply that equal numbers of data points should be allocated inside the support domains of all of the B-splines. Thus, in our proposed method, the initial knot vector $\{t_0, t_1, \ldots, t_{p+k}, t_{p+k+1}\}$ is computed according to the parameterization of the data points, as follows:

$$\begin{cases} t_0 = \cdots = t_k = u_0 = 0, \\ t_{p+1} = \cdots = t_{p+k+1} = u_m = 1, \\ t_{k+i} = \dfrac{1}{NT} \sum_{r=1}^{NT} u_{[i_c - r + 1]} - (1 - i_c + [i_c])(u_{[i_c]} - u_{[i_c - 1]}), \\ i_c \triangleq i \cdot \dfrac{m+1}{p-k+1}, \quad i = 1, \ldots, p-k. \end{cases} \tag{9}$$

where $NT \geq 1$ is an integer. The knot vector $S = \{s_0, s_1, \ldots, s_{q+l}, s_{q+l+1}\}$ can be similarly derived when considering the parameterization vector $\mathbf{v} = \{v_0, \ldots, v_n\}$. When $NT = 1$ and the data points are distributed uniformly, this computing method in Eq. (9) is equivalent to the uniform method, where the lengths of all of the intervals between the knot vectors are equal.

3.3. Computation of the control points and the fitting error

We first transform the fitting error (4) into the following matrix form:

$$E_{ls} = \|\mathbf{M} \cdot \boldsymbol{D} - \mathbf{Q}\|_2^2 = \sum_{i=0}^{m} \sum_{j=0}^{n} B_{i,j}^T B_{i,j} \tag{10}$$

where $B_{i,j} = P(u_i, v_j) - Q_{i,j}$ and matrices $\mathbf{D}$ and $\mathbf{Q}$ are, respectively, the vectorizations of the control points $\{D_{\alpha,\beta}\}$ and the data points $\{Q_{i,j}\}$ that are obtained by stacking one set of column-points on top of another, as follows:

$$\begin{cases} \mathbf{D} = (D_{0,0}, \ldots, D_{p,0}, \ldots, D_{\alpha,\beta}, \ldots, D_{0,q}, \ldots, D_{p,q})^T, \\ \mathbf{Q} = (Q_{0,0}, \ldots, Q_{m,0}, \ldots, Q_{i,j}, \ldots, Q_{0,n}, \ldots, Q_{m,n})^T. \end{cases} \tag{11}$$

The corresponding coefficient matrix is M, which is given in Eq. (12), where $R^{(i,j)}_{\alpha,\beta} = \frac{w_{\alpha,\beta}N_{\alpha,k}(u_i)N_{\beta,l}(v_j)}{\sum_{\theta=0}^{p}\sum_{\tau=0}^{q} w_{\theta,\tau}N_{\theta,k}(u_i)N_{\tau,l}(v_j)}$.

$$\mathbf{M} = \begin{pmatrix} R^{(0,0)}_{0,0} & \cdots & R^{(0,0)}_{p,0} & \cdots & R^{(0,0)}_{\alpha,\beta} & \cdots & R^{(0,0)}_{0,q} & \cdots & R^{(0,0)}_{p,q} \\ R^{(m,0)}_{0,0} & \cdots & R^{(m,0)}_{p,0} & \cdots & R^{(m,0)}_{\alpha,\beta} & \cdots & R^{(m,0)}_{0,q} & \cdots & R^{(m,0)}_{p,q} \\ & \cdots & & \cdots & & \cdots & & \cdots & \\ R^{(i,j)}_{\alpha,\beta} & \cdots & R^{(i,j)}_{\alpha,\beta} & \cdots & R^{(i,j)}_{\alpha,\beta} & \cdots & R^{(i,j)}_{\alpha,\beta} & \cdots & R^{(i,j)}_{\alpha,\beta} \\ & \cdots & & \cdots & & \cdots & & \cdots & \\ R^{(0,n)}_{0,0} & \cdots & R^{(0,n)}_{p,0} & \cdots & R^{(0,n)}_{\alpha,\beta} & \cdots & R^{(0,n)}_{0,q} & \cdots & R^{(0,n)}_{p,q} \\ R^{(m,n)}_{0,0} & \cdots & R^{(m,n)}_{p,0} & \cdots & R^{(m,n)}_{\alpha,\beta} & \cdots & R^{(m,n)}_{0,q} & \cdots & R^{(m,n)}_{p,q} \end{pmatrix}. \quad (12)$$

Therefore, the control points $\{D_{i,j}\}$ can be obtained through the minimization of the system (10) or, in other words, by solving the following linear system:

$$\mathbf{M}^T \cdot \mathbf{M} \cdot \mathbf{D} = \mathbf{M}^T \cdot \mathbf{Q}. \quad (13)$$

To evaluate the effectiveness of the fitting results, two error measures, the relative average error (RAE) and the relative maximum error (RME), are introduced as follows:

$$\begin{cases} RAE = \dfrac{1}{3}\displaystyle\sum_{k=1}^{3} \frac{\sum_{i=0}^{m}\sum_{j=0}^{n}|B^k_{i,j}|}{M_k \cdot (m+1)\cdot(n+1)}, \\ RME = \displaystyle\max_{k=1}^{3} \frac{\max_{i,j}|B^k_{i,j}|}{M_k} \end{cases} \quad (14)$$

where $M_k = \max_{i,j}|Q^k_{i,j}|$ ($Q_{i,j} = (Q^1_{i,j}, Q^2_{i,j}, Q^3_{i,j})$).

3.4. Derivatives of the fitting error

From Eqs. (5) and (6), we obtain the derivatives of the NURBS function w.r.t. the weight coefficients $\{w_{\alpha,\beta}\}$ and the parameter u. By viewing the fitting error E_{ls} as the multi-variable function of $\{w_{\alpha,\beta}\}$, $\{u_i\}$ and $\{v_j\}$, we need only to minimize E_{ls} by adjusting these variables.

The derivatives of the fitting error E_{ls} w.r.t. the weight coefficient $w_{\alpha,\beta}$ and u_r are computed as follows:

$$\frac{\partial E_{ls}}{\partial w_{\alpha,\beta}} = 2\sum_{i=0}^{m}\sum_{j=0}^{n}\frac{\partial P(u_i, v_j)}{\partial w_{\alpha,\beta}}\cdot B^T_{i,j}. \quad (15)$$

$$\frac{\partial E_{ls}}{\partial u_r} = 2\sum_{j=0}^{n}\frac{\partial P(u_r, v_j)}{\partial u_r}\cdot B^T_{r,j} \quad (16)$$

where $\frac{\partial P(u_i,v_j)}{\partial w_{\alpha,\beta}}$ and $\frac{\partial P(u_r,v_j)}{\partial u_r}$ are calculated as in Eqs. (5) and (6), respectively.

3.5. New projected algorithms for the point parameterization and weight coefficients

To obtain the optimal weight coefficients and the point parameterization, we employ the classical optimization method Levenberg–Marquardt (LM) or conjugate gradient (CG) that is associated with the gradient descent (GD) method [20], and the iterative directions are projected into the space of feasible directions [21,22].

The LM optimization algorithm has been widely adopted in addressing non-linear unconstrained LSQ problems. For our optimization problem, which is actually a non-linear constrained problem with linear constraints (17), not all the iterative directions of the LM algorithm are feasible:

$$\begin{cases} 0.5 \le w_{\alpha,\beta} \le 1.5, \quad \alpha = 0,\ldots,p,\ \beta = 0,\ldots,q, \\ u_i \le u_{i+1} - eps, \quad i = 0,\ldots,m-1, \\ v_j \le v_{j+1} - eps, \quad j = 0,\ldots,n-1. \end{cases} \quad (17)$$

In Ref. [17], the constraints of the parameters $\{u_i\}$, $\{v_j\}$ were set in the interval [0, 1]. In spite of its relative ease of use in handling constraints using a penalty function, this method may not yield a high approximation accuracy because it does not utilize the information implied by the relative positions of the data points. Moreover, the large number of monotonicity constraints in our optimization problem reduces the effectiveness of the penalty method.

In our algorithm, the projected gradient method is employed. In other words, the solution is iterated away from the infeasible region into the feasible region. Moreover, when the fitting error in the projected direction decreases slowly, implying that the effectiveness of the LM algorithm is greatly reduced, the projected GD iterative direction is adopted as an alternative. The iterative direction is projected into the space of feasible directions w.r.t. the current solution. When the iterative step length is quite small, this projection is implemented by setting the components of the iterative direction vector to zero where the constrained conditions are violated. This procedure is presented in Algorithm 1.

When the number of optimization variables is as large as the number of data points, we are likely to consider the CG algorithm which is more suitable for solving large-scale problems. Similar to Algorithm 1, the combination of the projected CG and projected GD can be used to solve our optimization problems, with step 4 and steps 20–26 in Algorithm 1 being altered as follows:

4: **if** $k = 0$ or $\frac{E_{ls}(x_{k-1})-E_{ls}(x_k)}{E_{ls}(x_k)} \le \rho$ **then**
$\Delta x_k = -J_k^T$
else $\beta_{k-1} = \frac{J_k^T J_k}{J_{k-1}^T J_{k-1}}$, $\Delta x_k = -J_k^T + \beta_{k-1}\cdot\Delta x_{k-1}$
end if

20-26: Find the optimal value α_{opt} in $[0, \alpha]$ through exact search algorithm, such as the golden section method [20]
$x_{k+1} = x_k + \alpha_{opt}\cdot\Delta x_k$
Compute $E_{ls}(x_{k+1})$

where J_k is the corresponding Jacobi vector, which records the derivatives of the error r_k w.r.t. all the variables; upper bound α and increment vector Δx_k are obtained through steps 5–19 in Algorithm 1; β_k is a scalar that makes the searching directions conjugate. This generated algorithm is simply denoted as PCGAGD.

3.6. Iterative scheme of the knot vector

It was reported in [11,23,24] that the knot vector can be adaptively placed according to the curvature or the performance of the fitting for the curve approximation. Weiss et al. [25] proposed a knot refinement procedure that depends on the shape of the fitting surface. Ma and Kruth [26] suggested computing the knot vectors according to the known point parameterization by averaging some terms of the components of the parameterization.

Obtaining the knot vectors through optimization, Laurent-Gengoux and Mekhilef [17] computed the derivatives of the fitting error w.r.t. the knot vectors by differentiating the Cox–De Boor formula (2). This differentiating procedure is complicated, especially for high-order B-spline functions, requiring us to perform approximately $(m+1)2^k(k+2)+(n+1)2^l(l+2)$ symbolic differential operations to compute the derivatives of the basis spline functions w.r.t. all the knots for a single iteration, where $k+2$ is the number of knots contained in every basis function $N_{i,k}(u)$ and $m+1$ is the number of basis functions in the u-parameter direction. This complexity is high when the degrees k, l of the B-spline functions are large, which may be one reason that only cubic B-spline functions are employed in NURBS reconstruction in [17]. Meanwhile, the numbers of the optimization variables and constraints increase when the knot vectors are added to the

Algorithm 1 The projected LM and GD algorithms

Initialize variable $x_0 = \{x_0^1, \ldots, x_0^N\}$ where $x = \{\{w_{\alpha,\beta}\}, \{u_i\}, \{v_j\}\}$, and parameters $\mu_0 = E_{ls}(x_0)$, $\beta = 10 \cdot \mu_0$, $\alpha = 1$, $\rho = 5 \cdot 10^{-3}$, $Eps1$, $Eps2$, $SF = 0.8$, $k = 0$;
Calculate the Jacobi vector $J_0 = J(x_0) = \nabla E_{ls}(x_0)$
while $\|J_k\| > Eps1$ or $k \leq MAXGEN$ **do**
  LM direction: $\Delta x_k = -(J_k^T J_k + \mu_k I)^{-1} J_k^T E_{ls}(x_k)$
  for $j = 1, \ldots, N$ **do**
    while $x_{k+1}^j = x_k^j + \alpha \Delta x_k^j$ do not satisfy the conditions in Eq. (17) **do**
      Scale α to a smaller number by $\alpha \leftarrow SF \cdot \alpha$
      if $\alpha < Eps2$ **then**
        $\alpha \leftarrow Eps$, $\Delta x_k^j = 0$
        for $r = j - 1, \ldots, 1$ **do**
          if condition violated **then**
            $\Delta x_k^r = 0$
          else
            break(jump out of the latest **for** loops)
          end if
        end for
      end if
    end while
  end for
  Compute $E_{ls}(x_{k+1})$
  if $\frac{E_{ls}(x_k) - E_{ls}(x_{k+1})}{E_{ls}(x_k)} > \rho$ **then**
    If $\mu_k > \beta$, $\mu_k \leftarrow \mu_k - \beta$
  else
    $\mu_k \leftarrow \mu_k + \beta$. Employ GD direction: $\Delta x_{k+1} = -J_{k+1}^T$
    $k \leftarrow k + 1$, $\alpha \leftarrow 1$. Goto step 5
  end if
  $k \leftarrow k + 1$, $\alpha \leftarrow 1$, calculate $J_k = J(x_k) = \nabla E_{ls}(x_k)$
end while
Output the fitting error and optimization variable values.

optimization, which places a heavy burden on the penalty function method.

In this study, the knot vectors are obtained indirectly by adjusting the positions of the data points inside every two knots. In other words, we use Eq. (9) to recompute the knot vector if the calculated fitting error cannot reach the given accuracy and descends very slowly. In this case, this procedure provides an easy and fast implementation for the iterative scheme for the knot vectors.

4. Numerical experiments

To verify the efficiency and effectiveness of the HOAAI, we have conducted several numerical experiments. Here, we choose a test problem on curve fitting, four test problems from [9] on surface reconstruction and one test problem on modeling a set of real-measured data points from the Mashan Pumped Storage Power Station in China. These test problems involve complicated curves and twisted and singular surfaces. We present the numerical results, discuss the impact of some of the other preferences on the performance of HOAAI, compare HOAAI with some other published approaches w.r.t. both the accuracy and the runtime cost and investigate the robustness of HOAAI against noise.

Test problem 1: a spiral curve. This parametric curve is given by the following:

$$\begin{cases} x = 2(1 - \sin\theta), \\ y = 2(1 - \cos\theta), \quad \theta \in [0, 4\pi] \\ z = 3\theta^2. \end{cases}$$

Test problem 2: a B-spline surface. For the convenience of comparison, we consider the same number, 14,400, of data points as in [9]. These points are obtained from a (5,4)-order B-spline surface with 8×6 control points.

Test problem 3: shell surface. This parameterized surface is defined as follows:

$$\begin{cases} x = \frac{1}{5}\left(1 - \frac{v}{2\pi}\right)\cos(2v)[1 + \cos(u)] + \frac{1}{10}\cos(2v), \\ y = \frac{1}{5}\left(1 - \frac{v}{2\pi}\right)\sin(2v)[1 + \cos(u)] + \frac{1}{10}\sin(2v), \\ \quad u, v \in [0, 2\pi] \\ z = \frac{v}{2\pi} + \frac{1}{5}\left(1 - \frac{v}{2\pi}\right)\sin(u). \end{cases}$$

Fitting this surface is not trivial because the surface is both twisted and concave and exhibits a singularity at the end of the surface.

Test problem 4: pisot surface. This complicated genus-one surface is given in parametric form by the following:

$$\begin{cases} x = A\cos(B + u)(2 + \cos(v)), \\ y = C\cos(D - u)(2 + E\cos(F + v)), \quad 0 \leq u, v \leq 2\pi, \\ z = E\cos(F + u)(2 + G\cos(H - v)). \end{cases}$$

where $A = 0.655866$, $B = 1.03002$, $C = 0.754878$, $D = 1.40772$, $E = 0.868837$, $F = 2.43773$, $G = 0.495098$ and $H = 0.377696$. This surface is chosen for our numerical study because it is a genus non-zero surface that can reflect the complexity of the model shape in practice.

Test problem 5: crescent surface. This intricate concave surface is given in parametric form by the following:

$$\begin{cases} x = [2 + \sin(2\pi u)\sin(2\pi v)]\sin(3\pi v), \\ y = [2 + \sin(2\pi u)\sin(2\pi v)]\cos(3\pi v), \quad u, v \in [0, 1], \\ z = \cos(2\pi u)\sin(2\pi v) + 4v - 2. \end{cases}$$

The chosen surface is challenging because it is closed, singular and highly distorted along the boundaries of the surface.

Test problem 6: pump surface. The problem addressed here is the construction of a mathematical model of the complete characteristic surface of the Mashan Pumped Storage Power Station in China through a set of real-measured data points with three variables: rotation speed, flow and moment (see the supplementary Excel table). For the pumped storage power station, the operating modes must be transferred from one to another to adapt to variations in the network load. To guarantee the operating safety, we must respond with the running conditions quickly and frequently; for this reason, the concrete mathematical relations between the characteristic parameters of the running pump turbine must first be established. Among these parameters, rotation speed, flow and moment are the most crucial, and only the relations of these three parameters must be obtained. However, the measured values of these three parameters for different openings of the wheels often contain a large amount of noise. Moreover, the curves that are obtained by linking all the measured points with the same opening are often crossed, collective and twisted, which implies further difficulties associated with this surface approximation. In this test problem, it is preferable to compute one of the two parameterization vectors $\mathbf{u}$, $\mathbf{v}$ by utilizing the given openings of the wheels.

4.1. Numerical results

We perform the HOAAI on a PC with a core processor operating at 2.8 GHz with 4 GB of RAM. Most of the computations were implemented in the popular scientific program Matlab, version 2009b, while the large matrix M in Eq. (13) is constructed using C++ with the interface function-mexFunction, as the construction procedure contains a large number of loops. The simulation results are presented in Table 1. For all test problems, we use the preferences

Table 1
The simulation results for six tested problems. NDP, NCP and OSF are the number of data points, the number of control points used and the order of B-spline function, respectively. RAE and RME are relatively average fitting error and relatively maximum fitting error, respectively. Notations m and s are the abbreviations for minute and second, respectively, which are adopted in the following tables.

Examples	NDP	NCP	OSF	RAE	RME	Time cost	Algorithm used
Spiral curve	126	30	12	6.5×10^{-5}	1.3×10^{-4}	83 s	Algorithm 1
B-spline surface	120×120	14×14	(6, 6)	3.1×10^{-6}	2.9×10^{-5}	7.4 m	PCGAGD
Shell surface	45×55	20×20	(9, 9)	1.03×10^{-7}	6.1×10^{-6}	2.7 m	Algorithm 1
Pisot surface	100×105	20×20	(8, 8)	1.27×10^{-6}	5.4×10^{-5}	11.3 m	PCGAGD
Crescent surface	116×124	30×30	(8, 8)	8.6×10^{-9}	1.1×10^{-8}	5.7 m	Algorithm 1
Pump surface	370×14	40×12	(16, 10)	8.8×10^{-4}	2.0×10^{-3}	7.3 m	Algorithm 1

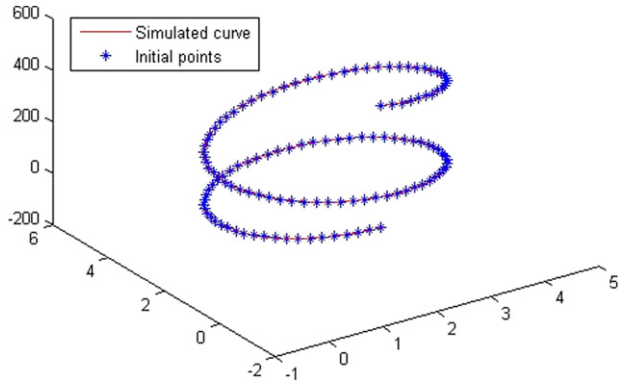

Fig. 2. Reconstruction of a spiral curve.

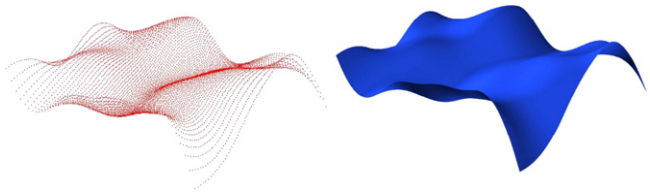

Fig. 3. Reconstruction of a B-spline surface. The left is data points, the right is the fitting surface.

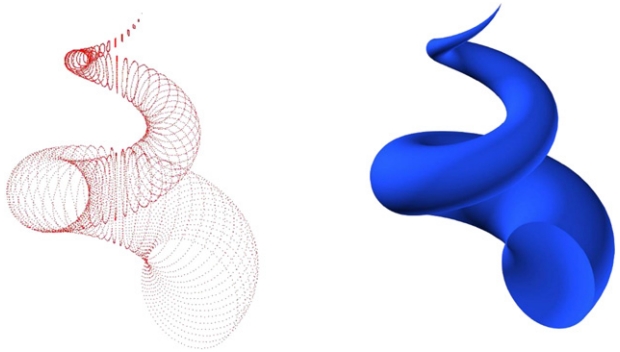

Fig. 4. Reconstruction of shell surface. The left is data points, the right is the fitting surface.

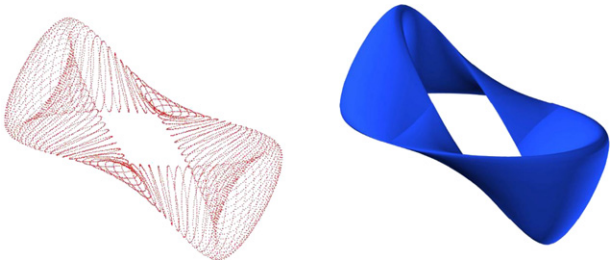

Fig. 5. Reconstruction of pisot surface. The left is data points, the right is the fitting surface.

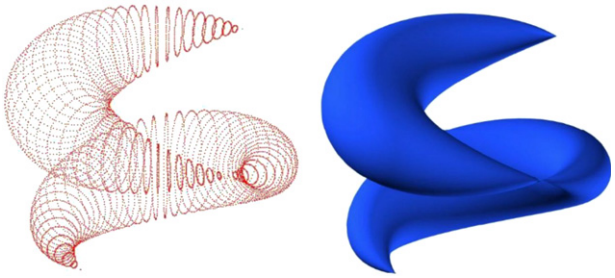

Fig. 6. Reconstruction of crescent surface. The left is data points, the right is the fitting surface.

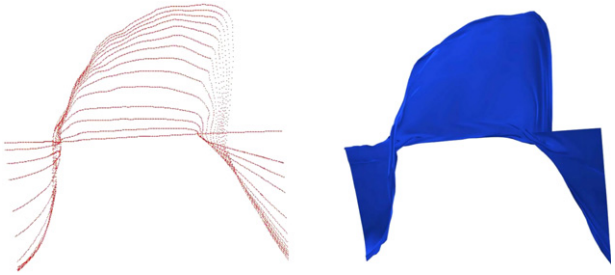

Fig. 7. NURBS fitting of pump surface. The left is data points, the right is the fitting surface.

$e = 1$ in Eq. (8) and $NT = 1$ in Eq. (9). The reconstruction curve or surfaces of the test problems are shown in Figs. 2–7.

Numerical results show that our proposed method can yield small fitting errors for a variety of complex surfaces in a relatively small period of time. For test problem 6, an average fitting error (RAE) from 5×10^{-3} to 1×10^{-3} is sufficient for real use, and the proposed HOAAI can reach an accuracy of 1.8×10^{-3} in 7 s, which demonstrates that our method responds rapidly to original or newly-added data points. Because the fitting error decreases as the number of control points increases, we employ a small number of control points (using them to the greatest extent possible) on the condition that a small fitting error can be obtained in an acceptable period of time. Considering test problem 1 as an illustration, we obtain an RAE of 6.5×10^{-5} when the number of control points is 30, while a slightly larger RAE of 2.8×10^{-4} can be obtained when the NURBS curve with 20 control points is chosen. The NURBS curve with 20 control points is preferable for a fitting accuracy of 10^{-4}.

4.2. Performance of HOAAI with additional preferences for three test problems

To evaluate the performance of the HOAAI, we conduct further numerical experiments to investigate the influences of different initial point parameterizations and knot vectors on both the accuracy and runtime cost in three test problems: a spiral curve, the crescent surface and the pump surface, and the influences of different optimization algorithms in an additional two test

Table 2
Results of time, cost and accuracy(RAE) for some other preferences. NT, e are the parameters in Eqs. (8) and (9), respectively. ALG denotes the algorithms used, k and l are the orders of the chosen B-splines, and UNIFORM denotes the uniform knot vector satisfying that all intervals between two adjacent knots have the same length. LM, CG, QN, and MLM are the abbreviations of Algorithm 1, PCGAGD, quasi-Newton [20], Moré's form of Levenberg–Marquardt [27] algorithms, respectively, all the search directions are projected into the space of feasible directions as in Algorithm 1.

e	0.5		1(proposed)		1.5		2	
Spiral curve	1.3×10^{-4}	83 s	6.5×10^{-5}	83 s	8.5×10^{-5}	79 s	6.1×10^{-5}	83 s
Crescent surface	7.5×10^{-9}	5.7 m	8.6×10^{-9}	5.7 m	8.6×10^{-9}	7.2 m	4.5×10^{-8}	5.7 m
Pump surface	8.4×10^{-4}	7.3 m	8.8×10^{-4}	7.3 m	8.8×10^{-4}	10.0 m	8.8×10^{-4}	10.0 m
NT	UNIFORM		1(proposed)		2		$k/2$ or $l/2$	
Spiral curve	4.4×10^{-4}	83 s	6.5×10^{-5}	83 s	8.1×10^{-5}	83 s	8.4×10^{-5}	83 s
Crescent surface	3.5×10^{-8}	5.7 m	8.6×10^{-9}	5.7 m	9.4×10^{-9}	5.7 m	5.3×10^{-8}	5.7 m
Pump surface	5.3×10^{-3}	7.3 m	8.8×10^{-4}	7.3 m	8.8×10^{-4}	7.3 m	8.8×10^{-4}	9.2 m
ALG	CG		LM (proposed)		QN		MLM	
Spiral curve	1.3×10^{-4}	83 s	6.5×10^{-5}	83 s	6.5×10^{-5}	80 s	6.5×10^{-5}	70 s
Crescent surface	9.5×10^{-9}	5.7 m	8.6×10^{-9}	5.7 m	8.6×10^{-9}	7.1 m	5.3×10^{-9}	5.7 m
Pump surface	9.7×10^{-4}	7.3 m	8.8×10^{-4}	7.3 m	8.8×10^{-4}	6.8 m	1.4×10^{-3}	7.3 m
ALG	CG (proposed)		LM		QN		MLM	
B-spline surface	3.1×10^{-6}	7.4 m	3.1×10^{-6}	8.7 m	3.8×10^{-6}	7.4 m	3.1×10^{-6}	6.5 m
Pisot surface	1.27×10^{-6}	11.3 m	1.59×10^{-6}	11.3 m	1.27×10^{-6}	11.1 m	1.43×10^{-6}	11.3 m

problems: the B-spline surface and the pisot surface besides the above three problems.

Table 2 lists the results of the accuracy and runtime costs of the proposed HOAAI when the employed optimization algorithms or parameter values in the point parameterization or iteration formulas are modified. When one parameter value or strategy is modified for comparison, the other parameters or strategies are fixed according to the settings proposed in Table 1. The stopping criterion is when either the runtime cost or the accuracy presented in Table 1 is reached for all the test problems.

Table 2 demonstrates that the HOAAI can achieve better performance w.r.t. either the accuracy or the runtime cost when choosing different parameter values or algorithms for different test problems. It can also be seen that PCGAGD is slightly more competitive than Algorithm 1 on test problems 2 and 4; therefore, the former algorithm is also adopted in this study. The performances of the employed PCGAGD and Algorithm 1 do not differ much in the considered test problems, but it is preferable to adopt the PCGAGD algorithm when considering problems with many more variables because PCGAGD eliminates the consideration of a large matrix inverse. The slight differences between the results in Table 2 also imply that the proposed HOAAI is general for slight parameter perturbations for surface reconstruction from complicated 3D data.

4.3. Comparison with four other methods

According to the numerical results, our proposed approach, HOAAI, exhibits good performance when used for the complex problems described above. To further assess the performance of the HOAAI, four other methods in the literature [8,9,14,17] are selected for comparison, considering the CPU runtime cost and the fitting error. We implement two methods presented in the literature, Borges and Pastva [8] and Laurent-Gengoux Mekhilef [17], and directly reference two results recently reported in the literature [9,14]. A comparison of the results for test problem 2 to test problem 5 using all the considered methods is provided in Table 3.

In Table 3, we observe that the HOAAI achieves the shortest CPU runtime and an almost minimal fitting error compared with the algorithms in [8,14,17] for all the test problems. When compared with the method in [9], our proposed HOAAI also achieves a high accuracy, $10^{-6} - 10^{-9}$, for all the test problems with a reasonable runtime cost (tens of seconds to minutes). With the exception of test problem 4, in which the proposed HOAAI still achieves a high accuracy of 9.2×10^{-12} after 180 min of runtime, the HOAAI method asymptotically achieves the same performance on test problems 2 and 5. In other words, it achieves the RAE of 7.0×10^{-6} in 21 s for test problem 2 and the RAE of 6.9×10^{-11} in 18.3 min for test problem 5. For test problem 3, our method achieves a higher accuracy in a shorter time. Therefore, our proposed method provides an alternative choice for solving one class of reconstruction problems that are demanding of runtime cost without requiring high accuracy. The method in [8] was used only for curve fitting. With an accuracy of 10^{-2} to 10^{-5}, we implemented this method to solve our four test problems on surface reconstruction. The accuracy was also quite low, although the method was easy and fast to implement. The approaches in [14,17] require a substantial amount of time for the surface approximation and lose their effectiveness in some test problems. These results illustrate that the HOAAI is competitive w.r.t. both accuracy and runtime cost.

4.4. Robustness of the proposed HOAAI against noise

To investigate the robustness of the proposed HOAAI against noise, we apply this approach to sets of perturbed data points in the test problems. Two types of noise are primarily considered: one type obeys a normal distribution $\mathcal{N}(0, \frac{\sigma}{\sqrt{3}})$, and the other obeys a uniform distribution $[-\sigma, \sigma]$. In both types of noise, σ is chosen to be $F \cdot \delta$, where F is a scale factor that reflects the intensity of the noise, we consider values of 1%, 5% 10% and 20% in our trials. And δ reflects the size of the surface and is set as $\min_k(\max_{i,j} Q^k_{i,j} - \min_{i,j} Q^k_{i,j})$. The parameters required for the proposed HOAAI are set to be the same as those in Table 1. The numerical results of the averages of the overall biases and the averages of the ten largest biases w.r.t. different intensities of noise are presented in Table 4, where the biases are the differences between the points without noise and the approximate points of the perturbed points. The points perturbed by different intensities of normally distributed noise and their corresponding reconstruction curves are presented in Fig. 8. In Fig. 9, the perturbed points and the same number of points extracted from the reconstruction surface are depicted. The reconstruction surfaces from the perturbed data points, which are perturbed by 1%, 5%,10% and 20% intensity of the normally distributed noise, are presented in Fig. 10.

Table 3
A comparison between the proposed approach and four other methods on test problems (t.p.) 2–5. RT denotes the runtime on the test problem. NP is the number of data points.

Method	T.P.2(NP:14400) (RAE,RT)	T.P.3(NP:2475) (RAE,RT)	T.P.4(NP:10500) (RAE,RT)	T.P.5(NP:14384) (RAE,RT)
[8] 2002 B-spline with LSQ.	$(1.5 \times 10^{-4}$, 7.4 m)	$(4.2 \times 10^{-5}$, 8.5 m)	$(5.8 \times 10^{-2}$, 15.3 m)	$(3.6 \times 10^{-4}$, 10.7 m)
[9] 2010 Interative two-step genetic algorithm with LSQ fitting. Based on stochastic search.	$(7.0 \times 10^{-6}$, 8–15 s)	$(5.4 \times 10^{-7}$, 11–19 m)	$(3.8 \times 10^{-15}$, 120–180 m)	$(6.9 \times 10^{-11}$, 14–20 m)
[14] 2007 Multi-objective evolutionary and genetic algorithm.	$(\geq 10^{-2}$, h)	$(\geq 10^{-2}$, h)	$(\geq 10^{-2}$, h)	$(\geq 10^{-2}$, h)
[17] 1993 Penalty function with classical optimization. Infeasible iteration direction may exist,	$(1.1 \times 10^{-4}$, 81.3 m)	$(5.9 \times 10^{-4}$, 63.2 m)	$(2.3 \times 10^{-3}$, 46.7 m)	$(6.2 \times 10^{-6}$, 35.4 m)
HOAAI-projected optimization and iteration	$(3.1 \times 10^{-6}$, 7.4 m) $(7.0 \times 10^{-6}$, 21 s)	$(1.03 \times 10^{-6}$, 2.7 m)	$(1.27 \times 10^{-6}$, 11.3 m) $(9.2 \times 10^{-12}$, 180 m) $(3.8 \times 10^{-15}$, Not achieved)	$(8.6 \times 10^{-9}$, 5.7 m) $(6.9 \times 10^{-11}$, 18.3 m)

Table 4
The numerical results about data points perturbed by different types and intensities of noise. The normal and uniform noises with intensities (F) 1%, 5%,10% and 20% are considered for this experiment. Percent sign(%) after every number is omitted for simplicity. In every parenthesis, the left is the average difference between the points NP without noise and the corresponding approximation points NP of the noisy points, the right is the average of the ten largest biases between points NP and NP.

Spiral curve	Normal size (F value)				Uniform noise (F value)			
	1	5	10	20	1	5	10	20
($\sigma = 1.99$)	(0.18, 2.18)	(0.75, 11.0)	(1.6, 23.3)	(3.1, 51.7)	(0.19, 2.17)	(0.75, 11.2)	(1.7, 23.5)	(3.2, 52.1)
Pisot surface	Normal size (F value)				Uniform noise (F value)			
	1	5	10	20	1	5	10	20
($\sigma = 1.97$)	(0.45, 1.53)	(2.6, 7.68)	(5.8, 21.8)	(11.6, 45.1)	(0.47, 1.58)	(2.7, 7.85)	(5.8, 22.1)	(11.7, 45.4)

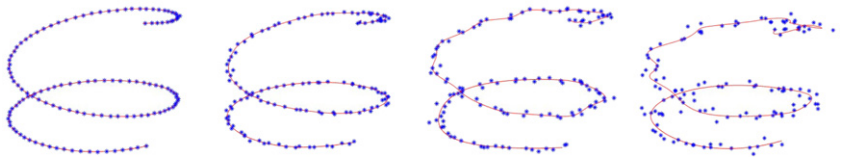

Fig. 8. Reconstructions of a spiral curve with noisy data points. From left to right, they are four sets of data points perturbed by 1%, 5%, 10% and 20% normally distributed noises from the points of test problem 1 and corresponding reconstruction curves.

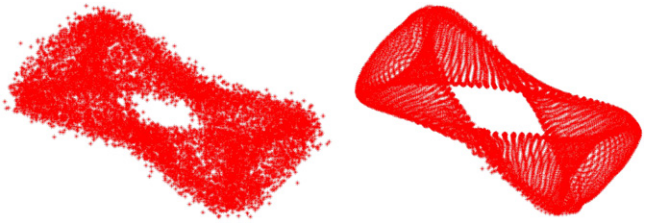

Fig. 9. Left is a set of data points perturbed by 10% normal distribution noise from the points of test problem 4. Right is a set of data points from the reconstruction surface.

Table 4 shows that the average biases between the points without noise and the approximation points of the noisy points increase according to the same orders of the added noise. However, the proposed method becomes less valid when the reconstruction curve or surface with perturbed points becomes distorted or rough in some places. It can be seen in Table 4 that the average of the ten largest biases w.r.t. 20% noise is significantly larger than twice the quantity w.r.t. 10% noise for the spiral curve, and the quantity w.r.t. 10% noise is significantly larger than twice the quantity w.r.t. 5% noise for the pisot surface. The abnormal increase of the ten largest biases implies the invalidity of the reconstruction method on the noisy points. Therefore, it can be concluded from Table 4 and Figs. 8 and 10 that the proposed method is robust against no more than 20% noise for the spiral curve and no more than 10% noise for the pisot surface.

5. Reconstruction with unorganized points

5.1. Boundary-determined parameterization

Efforts toward reconstruction with unorganized points from complicated models frequently resort to subdivision or the multi-patch of surfaces, such as subdivision methods [5,28], piecewise smooth surface reconstruction [29], or representation methods with multi-patch B-spline surfaces [4,30]. Representation with a single patch surface is also considered for slightly simpler geometry as a needless consideration of the continuity on boundaries. However, the latter class of methods requires a good initial parameterization to be made in advance.

For organized data points, Eq. (8) provides an initial parameterization for the proposed optimization method; for unorganized points, because of its complexity, there is work referring specially to this issue. A technique was developed for the parameterization of a general genus-zero surface by obtaining a counterpart on a spherical domain [31], and it was applied to a surface representation with a B-spline surface [32]. An initial base surface without intersections between the iso-parametric curves was created through PDE and SOM neural network techniques to provide an initial point parameterization [33,34]. Other parameterization methods have also been reported, such as an approximation with projected points [26], the multi-particle-based algorithm (PSO) [16], meshless parameterization [35], and dynamic base surface-based parameterization [36].

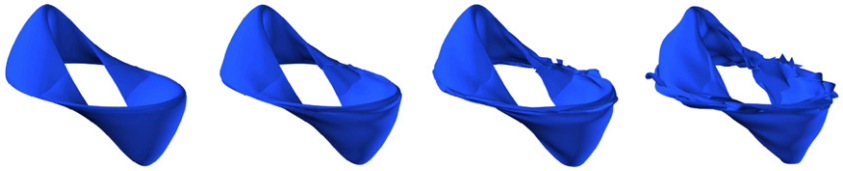

Fig. 10. Reconstructions of the pisot surface with noisy data points. From left to right, they are four reconstruction surfaces from data points perturbed by 1%, 5%, 10% and 20% normally distributed noises from the points of test problem 4.

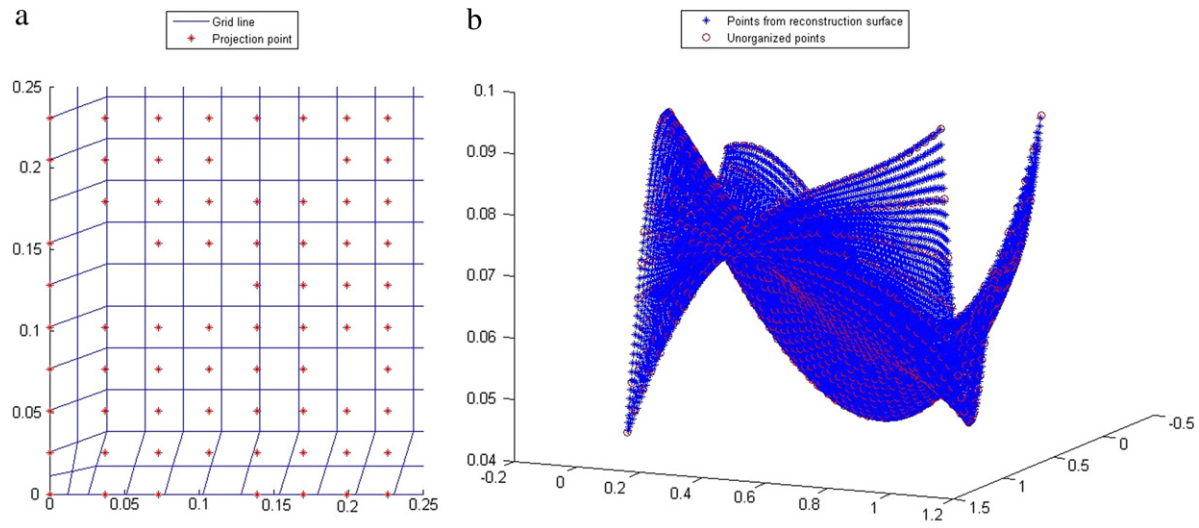

Fig. 11. (a) Depicts part of the grids and the projection points. (b) Demonstrates the provided data points for reconstruction and points from the reconstruction surface.

In this section, a method for the initial parameterization (IPM) of unorganized points from an open single-patch surface is proposed first. It is divided into the following steps:

1. Projecting the data points onto a 2D base surface and determining the boundary of the projection points.
2. Solving the Laplacian PDE equation twice with two different Dirichlet conditions, which are constructed from the boundary points w.r.t. the u and v directions, respectively. Obtaining the u-iso curves and v-iso curves by applying the Marching Cube Algorithm.
3. Intersecting the u, v iso-curves and generating $(m+1)\times(n+1)$ grids, where each set of u-direction grid lines shares the same u-parameterization $\{lu_0, \ldots, lu_{m+1}\}$, and each set of v-direction grid lines shares the same v-parameterization $\{lv_0, \ldots, lv_{n+1}\}$.
4. Allocating the projection points to the corresponding grids in which the points lie and refining the grids until each grid corresponds to a maximum of one projection point. Renewing the values of nu, nv and the parameter values of the grid lines.
5. Obtaining the approximate parameter value of each projection point with that of the corresponding grid, where the parameter value (u_i, v_j) of the (i, j) grid is computed as $u_i = \frac{lu_i + lu_{i+1} - 2\cdot lu_0}{2\cdot(lu_{m+1} - lu_0)}$, $v_j = \frac{lv_j + lv_{j+1} - 2\cdot v_0}{2\cdot(lv_{n+1} - lv_0)}$, $i = 0, \ldots, m, j = 0, \ldots, n$.
6. Approximating the initial parameterization with the parameter values of the projection points.

The preceding three steps are the same as those in [34]. To execute the approach IPM, it is worth noticing that in the fourth step, a projection point is assigned to the grid with larger u or v parameter values when the point lies on the boundary between the grids. The refining procedure starts from the grid corresponding to the most points. The points in the grid are first fitted with a linear function to determine their principal direction; the u-iso or v-iso line with the larger intersection angle with the principal direction is then chosen as the new split line. Moreover, the split line is assumed to be across the barycenter of the projection points, which can be used to approximately compute the u or v parameter value of the new grid (split) line. In the fifth step, the parameter values of the grids are calculated by averaging and normalizing each pair of consecutive parameter values of the grid lines.

After obtaining the initial parameterization, the proposed optimization method HOAAI is similarly employed with the refined grids corresponding to a set of organized points, with the difference that grids with no points make no contribution to the update of the optimization variables.

To test the performance of the HOAAI approach on these kinds of unorganized points, a example is conducted by applying the parameterization method IPM. The set of unorganized points for testing is 1400 randomly chosen points of 1600 organized (rectangular) points from a $(6, 5)$-order B-spline surface, where the four vertexes of the organized points are compulsively chosen.

Our trials show that a set of 41×50 grids separate the projection points successfully and that it takes 307 s to obtain the initial parameterization by IPM. An average fitting error RAE of 5.6×10^{-5} is then obtained after approximately 5.3 min of optimization by HOAAI, where a (4, 4)-order NURBS surface with 12 × 12 control points is adopted for the approximation. The numerical results illustrate that the proposed HOAAI is still effective for surface reconstruction with such kinds of unorganized points. Some of these projection points and grids are presented in Fig. 11(a). The provided points and points from the reconstruction surface are presented in Fig. 11(b).

5.2. Partition-based parameterization

The preceding PDE-based parameterization method is applicative to problems where the boundary curve can be found in advance; however, it loses the efficiency for semi-closed or closed

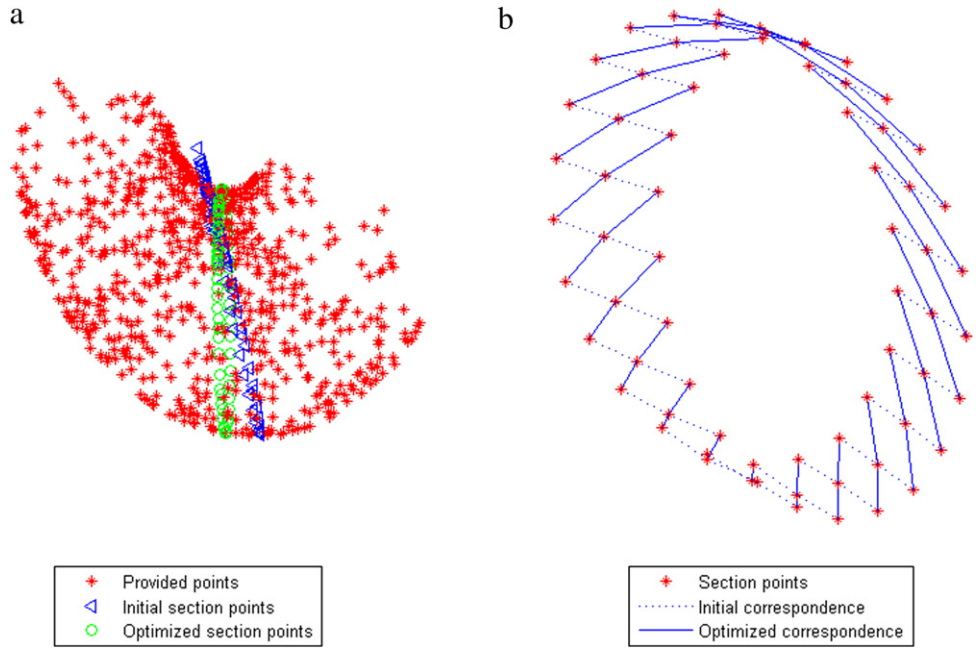

Fig. 12. (a) Presents the optimized section plane from an initial section plane. (b) Presents the optimized correspondence from an initial correspondence for distorted surface.

geometries, such as the shell surface in the test problems. In order to test the effect of the proposed HOAAI on unorganized points from the closed and distorted surface, we further propose a new method which is partitioning and parameterization (PAP) to obtain the initial parameterization.

For obtaining the initial parameterization of the points, two aspects are sequentially considered:

- Partitioning the points. Methods for segmenting the points are not uniquely determined. In the proposed PAP, the provided points are partitioned with many specially selected planes, which are assumed to be those with minimum curve lengths computed in Eq. (18).
- Parameterizing the partitioned points. After the preceding processing, the partitioned points can be viewed as from a semi-closed cylinder-like geometry. A further segmentation line on the geometry is needed to construct a surface which is topologically isomorphic to a rectangle. In other words, for the first point on the present section curve, the optimal corresponding point needs to be found on the next section curve, where the optimal correspondence means the minimum biases between the provided points and an interpolation surface.

Considering the first aspect, the section curve is the approximation of N_{pp} projection points $\{PP_i\}$ ($\{PP_i\}$ are the simplification of $\{PP_i, i = 1, \ldots, N_{pp}\}$ without confusion) with a cubic B-spline curve, where the projection points are the projections of the N_{pp} nearest points to the section plane on this plane in the normal direction. The fitting is executed after reordering the projection points in anticlockwise sequence which is presented in Appendix A, where the reordered points are still denoted as $\{PP_i\}$. Nv section points $\{IP_i\}$ with uniform parameter values are obtained from the fitting curve. By using the section points, the length of the section curve is approximated with the term as follows:

$$LengthCurve = \sum_{i=1}^{Nv-1} \|IP_{i+1} - IP_i\|_2 \tag{18}$$

and the central point of the section curve is approximated as follows:

$$CentralPoint = \frac{1}{Nv} \cdot \sum_{i=1}^{Nv} IP_i. \tag{19}$$

The procedure of finding the optimal section plane is to determine a normal direction. In this work, a probability-based algorithm-differential evolution (DE, [37]) is adopted to obtain the optimal normal direction which is presented in procedure I.

Note that the generated section points may not all be on one side of the last section plane, in which case a slight modification procedure presented in Appendix B is executed.

Considering the second aspect, we first denote the parameterization along each section curve as v-pararameterization and that along all the section curves as u-pararameterization for convenience. After performing the procedure I Nu times, Nu section planes are obtained, where Nv points $\{IP_{i,j}, j\}$ ($\{IP_{i,j}, j\}$ is the abbreviation of $\{IP_{i,j}, j = 1, \ldots, Nv\}$ without confusion) with uniform v parameter values are obtained from the i-th section curve and they are stored in anticlockwise order.

The cylinder-like geometry clamped by the i-th and $i + 1$-th section curves may be distorted, while the degree of distortion is reflected in the correspondence of the starting points with $v = 0$ (see Fig. 12(b)). In the following, the optimal correspondence is found by minimizing the biases between the clamping provided points and an interpolated surface. Without loss of generality, it is assumed that $IP_{i,1}$ corresponds to $v = 0$, and the correspondence of $i, i + 1$ section curves and that of $i + 1, i + 2$ section curves is approximately the same when the section curves are compactly obtained. The method for determining the optimal correspondence is addressed in procedure II.

In the first step of procedure II, for finding the nearest point to $IP_{i,1}$ on the $i + 1$-th section curve, the distance from Nv points on the $i + 1$-th curve to $IP_{i,1}$ are computed first, then fine searching is performed around the nearest point.

In the fourth step, rotating the points $\{IP_{i+1,j}\}t$ ($t \geq 0$) positions corresponds to rotating the initial sequence $(1, \ldots, Nv)$ of the points to the sequence $(t + 1, \ldots, Nv - 1, 1, \ldots, t, t + 1)$

Procedure I: Optimizing of the r-th normal vector.
%% Initialization
1: Randomly generating Nd normal vectors $\{NV_i\}$ around the $r-1$-th normal direction V_{r-1}.
2: Obtaining the central point Cp_r as $Cp_{r-1} + ss \cdot V_{r-1}$, where ss is the predetermined step size.
3: Computing the lengths $\{Fit_i\}$ of the section curves as in Eq. (18). $NumIter = 0$.
4: Initializing $F = 0.9$, $CR = 0.03$ for DE algorithm.
%% DE optimization
5: For each $i \in \{1, \cdots, Nd\}$,
6: Generating mutually different random numbers r_1, r_2, r_3 in $\{1, \cdots, Nd\}$ and random number jd from $\{1, 2, 3\}$.
7: For each $j \in \{1, 2, 3\}$

$$Tv_j = \begin{cases} NV_{i,j}, & \text{if } rand \geq CR \text{ or } j \neq jd, \\ NV_{i_1,j} + F \cdot (NV_{i_2,j} - NV_{i_3,j}), & \text{otherwise.} \end{cases}$$

8: End for j
9: Computing the length fit of the section curve corresponding to vector Tv.
10: If $fit \leq Fit_i$, $NV_i \leftarrow Tv$, $Fit_i \leftarrow fit$.
11: End if
12: $NumIter \leftarrow NumIter + 1$.
13: End for i
14: If $NumIter \leq MaxNumIter$, goto step 5; else, goto step 16.
15: End if
%% Output
16: Outputting the normal vector NV_{i_0} corresponds to the minimum Fit_{i_0}. Renewing the central point according to Eq. (19).

Procedure II: Determining the optimal correspondence
%% Finding initial correspondence
1: Finding the nearest point to $IP_{i,1}$ on the $i+1$-th section curve, then generating another $Nv - 1$ points with uniform v parameter values on the fitting curve. The new section points are still denoted as $\{IP_{i+1,j}, j\}$, where the first point corresponds to the nearest point.
2: Corresponding to the point $IP_{i+1,1}$, renewing the points $\{IP_{i+2,j}, j\}$ on the $i+2$-th section curve in a similar way.
%% Optimizing the correspondence
3: For each $t \in \{-MaxPos, \cdots, MaxPos\}$
4: Rotating the points $\{IP_{i+1,j}, j\}$ t positions to obtain $\{IP^{(t)}_{i+1,j}, j\}$, and the points $\{IP_{i+2,j}, j\}$ $2t$ positions to obtain $\{IP^{(2t)}_{i+2,j}, j\}$.
5: Interpolating the points $\{\{IP_{i,j}, j\}, \{IP^{(t)}_{i+1,j}, j\}, \{IP^{(2t)}_{i+2,j}\}, j\}$ with cubic B-spline surface.
6: Obtaining $5 \times Nv$ grid points with uniform parameter values from the interpolated surface.
7: For each provided point that lies in the region clamped by the i-th and $i+2$-th section curves, finding the corresponding nearest grid, then approximating the parameter values of the point as in Eq. (20).
8: Computing the biases $Bias^{(t)}$ between the provided points and the interpolated surface as in Eq. (21).
9: End for t
%% Output
10: Obtaining the optimal correspondence t_0 that corresponds to the minimum $Bias^{(t_0)}$.

because the section points are closed. The case for $t < 0$ is similarly considered.

In the seventh step, denoting the distances from the provided point $P^{(\alpha)}$ ($\alpha = 1, \ldots, NP$) to the four vertexes of the nearest grid as $\{d_k, k = 1, \ldots, 4\}$. The parameter values $(u^{(\alpha)}, v^{(\alpha)})$ of the point $P^{(\alpha)}$ are approximately computed as follows:

$$(u^{(\alpha)}, v^{(\alpha)}) = \frac{1}{\sum_{r=1}^{4} \frac{1}{d_r}} \sum_{k=1}^{4} \frac{1}{d_k} \cdot (u_k, v_k) \tag{20}$$

where $\{(u_k, v_k)\}$ are the parameter values of the four vertexes, respectively.

In the eighth step, the overall bias between the provided points and the interpolation surface is computed as follows:

$$Bias = \frac{1}{3} \max_{\alpha} B^{(\alpha)} + \frac{2}{3} \frac{1}{NP} \sum_{\alpha} B^{(\alpha)} \tag{21}$$

where $B^{(\alpha)} = \|P(u^{(\alpha)}, v^{(\alpha)}) - P^{(\alpha)}\|_2$, $P(u^{(\alpha)}, v^{(\alpha)}) = \sum_{i,j} D_{i,j} N_{i,3}(u^{(\alpha)}) N_{j,3}(v^{(\alpha)})$.

Based on the preceding partitioning and local parameterization procedures, the overall parameterization method PAP is stated as follows:

Step 1 . Finding Nu section planes by procedure I, which obtains Nu section curves, $Nu - 1$ clamping regions and the corresponding provided points lie in each region.
Step 2 . For each set of three consecutive planes, finding the optimal correspondence of the points with $v = 0$ by procedure II. Computing the approximate parameter values of each point and the index of the corresponding nearest grid. The final grid points are still denoted as $\{IP_{i,j}, i = 1, \ldots, 2 \cdot Nu - 1, j = 1, \ldots, Nv\}$, where the points corresponding to $v = 0$ have been placed at the first place of each set of section points.
Step 3 . Partitioning the grids and parameterizing the grids with the method which is stated in the fourth and fifth steps of IPM in Section 5.1 with a difference that each grid is partitioned at most once and only the most representative point is retained.
Step 4 . Approximating the parameterization of the provided points with those of the grids where the points lie.

Heuristic search: in the proposed PAP, some operations are based on the adjacent points of a considered point, while the usual handling needs to search the whole of the provided points to determine the adjacent points. In this work, to accelerate the searching procedure, the information on relative size of each coordinate of the provided points is first stored. In other words, each coordinate is partitioned into many small equidistant intervals according to the minimum and maximum values of the coordinate of all the provided points. Then the equidistant interval that the coordinate of each point lies in is stored. Based on the stored information, the adjacent points of a considered point are found in a time complexity of $O(p)$ where p is the number of points for searching, because the points lying in a local region of the considered point share the adjacent coordinate intervals.

For applying the proposed PAP to the surface reconstruction with the optimization HOAAI, two sets of scattered points (15,629 and 14,384 points) are randomly and compactly generated from shell and crescent surfaces, respectively. Numbers of 100 and 64 section planes are found for the points from shell and crescent surfaces, and 80 points are obtained on each section plane.

Numerical experiments show that the DE algorithm performs robustly on the problems. The proposed optimization method HOAAI still obtains a small fitting error 10^{-4} for unorganized points from a closed and distorted surface. It can be seen from Table 5 that most of the computational times are consumed in the procedure

Table 5
Numerical results of initial parameterization PAP and HOAAI optimization for two sets of scattered points.

Examples	Time of finding section planes (m)	Time of finding correspondence (m)	Time of optimization by PCGAGD (m)	Overall time (m)	Fitting error (RAE)
Shell surface	13.2	1.5	9.7	24.4	3.8×10^{-4}
Crescent surface	8.7	0.9	6.3	15.9	1.1×10^{-4}

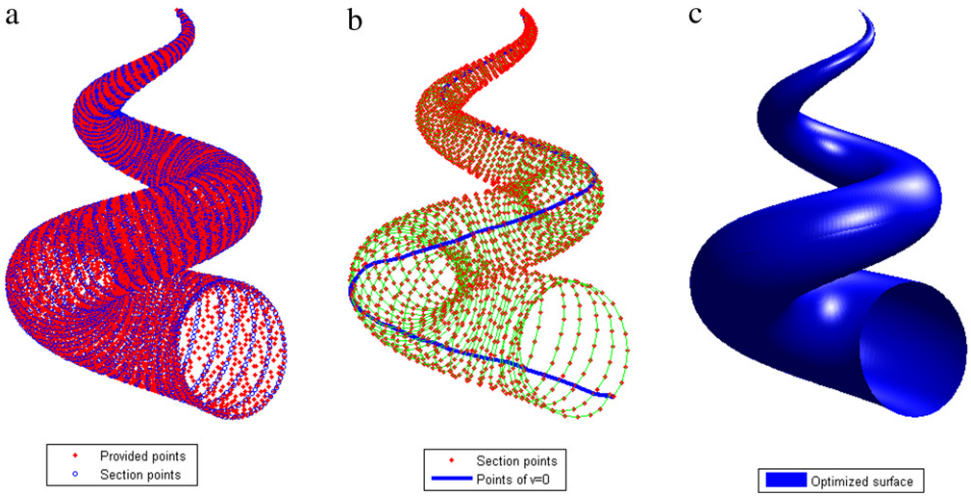

Fig. 13. Surface reconstruction of crescent surface from the initial parameterization to the final optimized surface.

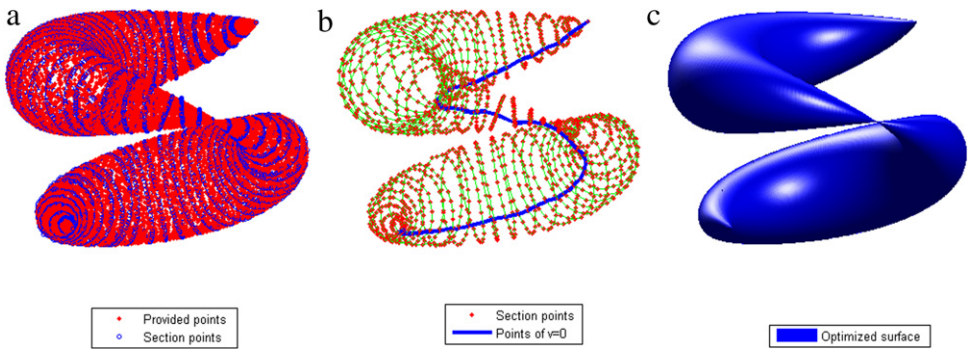

Fig. 14. Surface reconstruction of crescent surface from the initial parameterization to the final optimized surface.

for finding section planes, because the stochastic algorithm (DE) is employed.

Fig. 12(a) presents the optimization from an initial section curve to a section curve with smaller length. Fig. 12(b) addresses the optimization of the correspondence of three layers of section points. Figs. 13 and 14 present the initial parameterization and HOAAI optimization on two sets of scattered points from shell and crescent surfaces.

6. Discussions and future work

In this paper, we present a new method for the reconstruction of 3D surfaces based on non-uniform rational B-splines. Although an easily implemented optimization scheme is proposed and proven efficient for all the test problems considered here, possible modifications exist that can boost the accuracy of the approximation through the following aspects. (i) The knot vectors can be viewed as the optimization variables for some generations, where the derivatives of the fitting error w.r.t. the knot vector are calculated via a forward difference approximation rather than by differentiating both sides of Eq. (2). (ii) All the variables can be divided into several groups according to their ranges or the sizes of their absolute derivatives; these variables are then optimized group by group in several phases [9].

This study also contains some limitations. First, we primarily studied the reconstruction problems that require good smoothness, such as our real-measured data from the Pumped Storage Power Station in China. When a reconstruction problem with dense sample data does not require smoothness or requires only piecewise smoothness, the piecewise linear approximation with a subdivision surface proposed in [5,28] and other Refs. [4,29,30] can be used to expand our proposed method to solve a broader selection of problems. Second, although we introduced two approaches for applying the proposed HOAAI to the reconstruction of unorganized points in Section 5, the first approach is mainly applicable to the cases with boundary determined. The employed section planes which are a form of segmenting the point set in the second approach determine the restricted applicability of the parameterization method for general complex multi-patch surfaces. Therefore, our future work will combine the related studies in the literature on surface reconstruction for unorganized points, such as [32,33,35], to design a good initial parameterization for unorganized points. Third, the ability of the proposed method to reconstruct genus non-zero surfaces is limited, although our method was

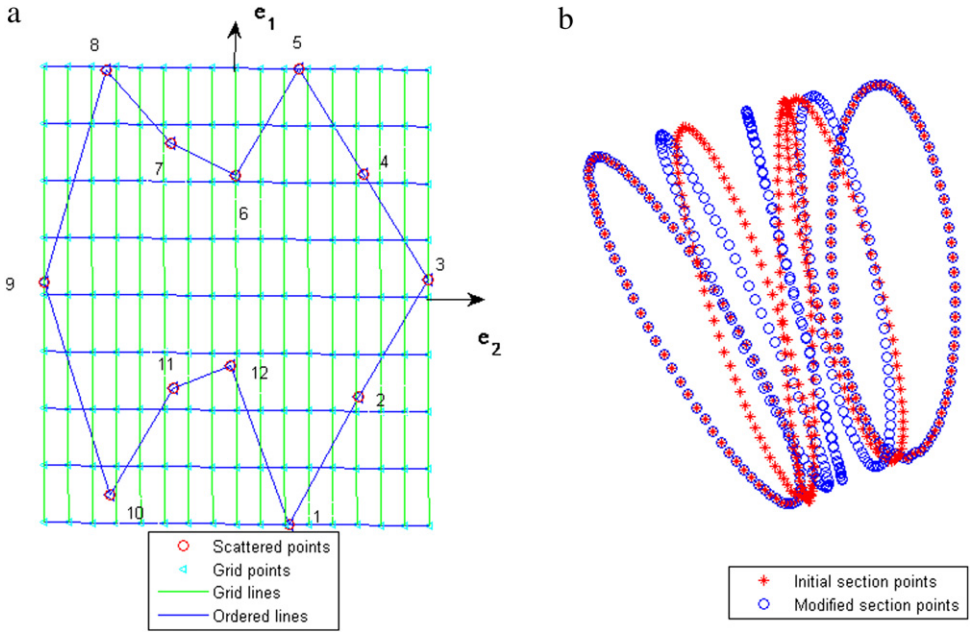

Fig. A.1. (a) Depicts the reordering procedure of scattered points. (b) Presents the modified section points for some intersecting section points.

proven effective on test problem 4, which is a genus-one surface. An effective parameterization method for these challenging problems will be worth exploring.

In addition to the possible measures for improvement and the research interests described above, a great deal of work remains to be done. On the one hand, we expect a good theoretical explanation for the effectiveness of our proposed method. On the other hand, our approach is very robust and can be applied to a variety of practical reconstructions; therefore, more real-world models are expected to be reconstructed using our method, which will provide valuable feedback toward further improvement.

7. Conclusions

In this paper, we have proposed a new method to derive all the parameters for NURBS surface approximation: the weight coefficients, the parameterization of the data points and the knot vectors. The first two sets of parameters are optimized by the projected LM and GD or CG and GD algorithms, while the last set of parameters, the knot vectors, are computed according to the calculated point parameterizations using iterations. Our approach has shown good performance in numerical experiments on six test problems, including a complicated curve, twisted and singular surfaces, real-measured data points with noise and three sets of unorganized points. Comparisons with other published methods demonstrate that our approach is competitive w.r.t. both accuracy and runtime cost.

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments and suggestions. The authors also thank Prof. Jun Zou of the Department of Mathematics of the Chinese University of Hong Kong for his useful discussions and suggestions. This work was supported by the Key Program of the National Natural Science Foundation of China under Grant 51039005 and the Chinese National Natural Science Foundation under Grants 61070007 and 61173060.

Appendix A. Reordering scattered points

The reordering method is based on a set of partitioned grids constructed by the scattered points which lie on a plane in 3D space. The procedure which is depicted in Appendix Fig. A.1(a) is divided into the following steps:

1. Finding two principal directions $\mathbf{e_1}$, $\mathbf{e_2}$ of the scattered points and the normal vector $\mathbf{e_3}$ of the fitting plane by principal component analysis (PCA) [38,39]. Determining the numbers of grids in directions $\mathbf{e_1}$, $\mathbf{e_2}$ according to the corresponding eigenvalues of the vectors.
2. Finding four boundary lines which compactly cover the provided points, and they are parallel to $\mathbf{e_1}$ or $\mathbf{e_2}$.
3. Searching the grids which are adjacent to the four boundary lines in a predetermined direction (clockwise or anticlockwise). The order of the points is approximated with the order of the grids where the points lie.

Appendix B. Modifying the section planes

The employed modifying procedure is illustrated with several sets of section points in Appendix Fig. A.1(b) and addressed as follows:

1. For the i-th section plane ($i = 2, \ldots, Nu-1$), checking whether it intersects with the $i-1$-th section plane by observing whether the i-th section points all lie on the same side of the $i-1$-th section plane.
2. If intersection is found, finding the smallest index $k \in [i+1, Nu]$ such that the k-th section plane does not intersect with the $i-2$-th section plane.
3. For the t-th section plane ($t = i-1, \ldots, k-1$), obtaining the approximated section points $\{TP_{t,j}, j = 1, \ldots, Nv\}$ as follows:

$$TP_{t,j} = IP_{i-2,j} + \frac{t-i+2}{k-i+2}(IP_{k,j} - IP_{i-2,j}).$$

4. Determining the normal direction of the section points $\{TP_{t,j}, j\}$ by PCA where the normal direction is the eigenvector with the minimal eigenvalue. Generating new section points according to the new computed normal direction. Renewing the normal direction and the section points of the t-th plane.

Appendix C. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.cad.2012.05.004.

References

[1] Floater MS. Parameterization and smooth approximation of surface triangulations. Computer Aided Geometric Design 1997;14(3):231–50.
[2] Pan RJ, Skala V. Continuous global optimization in surface reconstruction from an oriented point cloud. Computer-Aided Design 2011;43(8):896–901.
[3] Hussain MZ, Sarfraz M. Monotone piecewise rational cubic interpolation. International Journal of Computer Mathematics 2009;86(3):423–30.
[4] Eck M, Hoppe H. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques. 1996. p. 325–34.
[5] Gregorski B, Hamann B, Joy K. Reconstruction of B-spline surfaces from scattered data points. Proceedings Computer Graphics International 2000; 163–70.
[6] Sarkar B, Menq CH. Parameter optimization in approximating curves and surfaces to measurement data. Computer Aided Geometric Design 1991;8: 267–90.
[7] Yang Z, Deng J, Chen FL. Fitting unorganized point clouds with active implicit B-spline curves. The Visual Computer 2005;21(1):831–9.
[8] Borges CF, Pastva T. Total least squares fitting of Bezier and B-spline curves to ordered data. Computer Aided Geometric Design 2002;19(4):275–89.
[9] Gálvez A, Iglesias A, Puig-Pey J. Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction. Information Science 2012;182(1):56–76.
[10] Wang WP, Pottmann H, Liu Y. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. ACM Transactions on Graphics 2006;25(2):214–38.
[11] Zhao XY, Zhang CM, Yang B, Li PP. Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation. Computer Aided Design 2011;43(6):598–604.
[12] Riyazuddin M. Visualization with NURBS using simulated annealing optimization technique. Master thesis. Dept. ICS. King Fahd University of Petroleum and Minerals, Dhahran (Saudi Arabia); 2004.
[13] Sarfraz M, Raza SA, Baig MH. Computing optimized curves with NURBS using evolutionary intelligence. Lecture Notes in Computer Science 2005;3483: 806–15.
[14] Wagner T, Michelitsch T, Sacharow A. On the design of optimizers for surface reconstruction. In: Proceedings of the 2007 genetic and evolutionary computation conference-GECCO2007. London (England); 2007. p. 2195–202.
[15] Weinert K, Surmann T, Mehnen J. Evolutionary surface reconstruction using CSG-NURBS-hybrids. In: Proceedings of the 2001 genetic and evolutionary computation conference-GECCO2001. San Francisco (USA); 2001. p. 1456–463.
[16] Gálvez A, Iglesias A. Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points. Information Science 2010; http://dx.doi.org/10.1016/j.ins.2010.11.007.
[17] Laurent-Gengoux P, Mekhilef M. Optimization of a NURBS representation. Computer Aided Design 1993;25(11):699–710.
[18] De Boor C. A practical guide to splines. Springer-Verlag; 1978.
[19] Lee ET. Choosing nodes in parametric curve interpolation. Computer-Aided Design 1989;21(6):363–70.
[20] Hendrix EMT, Toth BG. Introduction to nonlinear and global optimization. Cambridge: Springer; 2010.
[21] Schwartz A, Polak E. Family of projected descent methods for optimization problems with simple bounds. Journal of Optimization Theory and Applications 1997;92(1):1–31.
[22] Lin CJ, Moré JJ. Newton's method for large bound-constrained optimization problems. SIAM Journal on Optimization 1999;9(4):1100–27.
[23] Li W, Xu S, Zhao G, Goh LP. Adaptive knot placement in B-spline curve approximation. Computer-Aided Design 2005;37(7):791–7.
[24] Ülker E, Arslan A. Automatic knot adjustment using an artificial immune system for B-spline curve approximation. Information Sciences 2009;179(10): 1483–94.
[25] Weiss V, Andor L, Renner G, Varady T. Advanced surface fitting techniques. Computer Aided Geometric Design 2002;19(1):19–42.
[26] Ma WY, Kruth JP. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. Computer-Aided Design 1995; 27(9):663–75.
[27] Moré JJ. The Levenberg–Marquardt algorithm: implementation and theory. In: Watson GA, editor. Numerical analysis. Berlin: Springer-Verlag; 1977.
[28] Krause FL, Fischer A, Gross N, Barhak J. Reconstruction of freeform objects with arbitrary topology using neural networks and subdivision techniques. CIRP Annals-Manufacturing Technology 2003;52(1):125–8.
[29] Hoppe H, DeRose T, Duchamp T, Halstead M, Jin H, McDonald J, Schweitzer J, Stuetzle W. Piecewise smooth surface reconstruction. In: Proceedings of the 21st annual conference on computer graphics and interactive techniques (SIGGRAPH'94). New York (USA); 1994. p. 295–302.
[30] Krishnamurthy V, Levoy M. Fitting smooth surfaces to dense polygon meshes. In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques (SIGGRAPH '96). New York (NY, USA): ACM; 1996. p. 313–24.
[31] Praun E, Hoppe H. Spherical parametrization and remeshing. ACM Transaction on Graphics (SIGGRAPH) 2003;22(3):340–9.
[32] Losasso F, Hoppe H, Schaefer S, Warren JD. Smooth geometry images. In: Proceedings of symposium on geometry processing. 2003. p. 138–45.
[33] Barhak J, Fischer A. Adaptive parameterization for reconstruction of 3D freeform objects from laser-scanned data based on a PDE approach. The Visual Computer 2001;17(6):351–69.
[34] Barhak J, Fischer A. Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques. IEEE Transactions on Visualization and Computer Graphics 2001;7(1):1–16.
[35] Floater MS. Meshless parameterization and B-spline surface approximation. Proceedings of IMA Conference on the Mathematics of Surfaces 2000;1–18.
[36] Azariadis PN. Parameterization of clouds of unorganized points using dynamic base surfaces. Computer Aided Design 2004;36(7):607–23.
[37] Storn R, Price K. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 1997;11(4):341–59.
[38] Jolliffe IT. Principal component analysis. 2nd edition Springer; 2002.
[39] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. SIGGRAPH Computer Graphics 1992;26(2): 71–8.