Information Sciences 245 (2013) 276–289

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

IGA-based point cloud fitting using B-spline surfaces for reverse engineering

Xiuyang Zhao [a,b], Caiming Zhang [c], Li Xu [a,b], Bo Yang [a,b,*], Zhiquan Feng [a,b]

[a] School of Information Science and Engineering, University of Jinan, Jinan 250022, PR China
[b] Shandong Provincial Key Laboratory of Network Based Intelligent Computing, Jinan 250022, PR China
[c] School of Computer Science and Technology, Shandong University, Jinan 250101, Shandong, PR China

ARTICLE INFO

Article history:
Received 13 July 2012
Received in revised form 1 April 2013
Accepted 15 April 2013
Available online 23 April 2013

Keywords:
Reverse engineering
Fitting error
Knot placement
B-spline patch approximation
IGA
PSO

ABSTRACT

Reverse engineering is a viable method to create a 3D virtual model of real physical parts. Usually, reverse engineering consists of two main steps: (1) measure the object and (2) reconstruct it as a 3D model. The measured data are usually represented as a point cloud without topological information and must therefore often be converted into a tensor product B-spline surface format, which has become an industry standard in computer graphics and in CAD systems. In this paper, a new immune genetic algorithm (IGA) for point cloud fitting that fits a noisy 3D point cloud using a B-spline surface with approximate G1 continuity is presented. The point cloud is first segmented into a set of quadrilateral patches. For every patch, a B-spline surface is reconstructed using a least-squares approximation method, and then the surface is optimized to increase the approximation precision using an IGA-based knots adjustment algorithm. Finally, the B-spline patches are stitched together with approximate G1 continuity with a numerical method and the particle swarm optimization (PSO) algorithm. A set of experimental results shows that the proposed method achieves better approximation accuracy than the Bezier-based method and the GA-based method.

1. Introduction

The manufacturing products in an information society have two types of different representations, i.e., information models and real products. The most important parts are the former, which contain the most valuable and innovative parts. Manufacturers and industrial corporations can deliver more competitive products with better quality and lower prices by secondary processing using the information model. As a bridge between the real product and the information model, reverse engineering has been extensively studied. The primary activities of reverse engineering are measuring an object and then reconstructing it as a 3D model. The real product can often be measured using 3D scanning technologies, such as laser scanners, structured light digitizers, coordinate measuring machines, or industrial CT scanners [22]. The measured data, usually represented as a point cloud, lack topological information. Therefore, the data are often reconstructed into a more usable format, such as an implicit surface model or a parametric surface model.

Given the scanned partial information about an object, the goal of surface reconstruction is to construct a fairing surface that approximates the original shape as much as possible. Implicit surfaces are iso-surfaces through some scalar field in 3D, thus, an implicit surface consists of those points in 3D space that satisfy some particular requirement, and the implicit shape representations unify surface and volume modeling. With an implicit surface representation, a given point x can easily be

[*] Corresponding author at: School of Information Science and Engineering, University of Jinan, Jinan 250022, PR China. Tel.: +86 53182765717.
E-mail address: yangbo@ujn.edu.cn (B. Yang).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.04.022

tested to see if it is on the surface. Implicit shape representations are attractive because they allow a complex shape to be described by one formula, and several complex shape editing operations are easy to perform with such models [13]. However, this approach has two serious drawbacks. First, although well-suited to ray tracing, the shapes are not easily rendered at interactive speeds, reflecting the underlying problem that it is difficult to sample them systematically. Second, the shapes of implicit surfaces have proven to be more difficult to specify and control than those of their parametric counterparts [26].

A parametric surface representation can be written as $x = F(u, v)$, where u and v are surface parameters, and x is a point on the surface. The parametric surface has become the preferred representation in geometric modeling and CAD systems. The important advantage of a parametric representation is that it is easy to enumerate points on the surface. Another advantage of a parametric surface is that a pair of parameters is associated with each surface point that can be used to index a texture map for coloring the surface.

For the complexity of the actual shape, the reconstructed model is often represented as a combined parametric surface. Generally, the reconstruction of a combined parametric surface from a point cloud can be subdivided into 3 steps: point cloud data segmentation and parameterization; parametric surface reconstruction for a patch; smoothly stitch of different surface patches. Parametric surfaces can be roughly classified into two categories according to the patch type, the Bezier patch and the B-spline patch [11]. Earlier studies using the Bezier patch to generate a smooth surface include Refs. [2,20,25]. Recently, Lin [11] proposed an adaptive mesh-fitting algorithm that fits a triangular model with G1 smoothly stitched bi-quintic Bezier patches. By recursively subdividing the underlying quadrilateral into four sub-patches, the method can reach the target, i.e., that the fitting error of every Bezier patch meets the tolerance requirement.

Because of the local support property, the B-spline surface can support more control for the patch shape than the Bezier surface does. Thus, once the patch has a complicated appearance, the B-spline surface can be subdivided less times, resulting in higher efficiency. Currently, tensor product non-uniform rational B-spline surfaces have become the industry standard in computer aided design, as well as in computer graphics [18]. Although it has been studied for decades, it is not an easy task to reconstruct high-precision and fairing B-spline surfaces from a point cloud. Krishnamurthy [10] proposed a method to fit the arbitrary topology mesh by B-spline surfaces, but did not consider the continuity of the surfaces. Wang [19] developed a local scheme of constructing G1 smooth B-spline surfaces with interior single knots over an arbitrary topology of a polygonal model. Zhang [28] extended the adaptively fitting method to B-spline surfaces instead of Bezier surfaces. Almost all the methods above are given the knots and parameters corresponding to the data points a priori, but the choice of knots has a considerable effect on the shape of a surface. Therefore, the location of the knots must be determined as precisely as possible.

In this study, the B-spline surface-fitting problem is defined to produce a B-spline surface to approximate a point cloud within a specified tolerance. A real complex object is difficult to model as a single surface patch; therefore, the point cloud is separated into quadrilateral patches, and each patch is reconstructed using a B-spline surface and then stitched together with approximate G1 continuity. During the stage of the single surface patch reconstruction, the knot placement is treated as a multivariate and multimodal non-linear optimization problem and is determined by an IGA-based optimization algorithm to reduce the fitting error of the surface. During the stage of continuous stitching, a PSO-based method is used to stitch the patches instead of the traditional method to avoid the rapidly increasing of the number of the knots.

The pipeline of the algorithm is as follows:

(1) The point cloud is segmented into a set of quadrilateral patches, which are point cloud patches with four boundaries (an example can be seen in Fig. 7b).
(2) For every quadrilateral patch, a B-spline surface is used to approximate it with the least squares method. If the fitting error of the surface patch does not meet the error tolerance requirement, IGA is used to optimize the knot vectors of the B-spline surface patch to enhance the approximation accuracy (an example can be seen in Fig. 7b, the blue-colored patches).
(3) After the placement of the adaptive knot vectors, the surface patches often have different knot vectors. The control points near the boundary are adjusted to achieve an approximate G1 continuity by using numerical methods and a PSO-based method (an example can be seen in Fig. 7c).
(4) If the fitting error still does not meet the requirement after the adaptive knot vector placement algorithm, an adaptive fitting method will be applied to the patch (described in Section 4.4).

A flowchart of the IGA-based point cloud fitting algorithm using B-spline surfaces is shown in Fig. 1.

The remainder of this paper is organized as follows. Section 2 briefly reviews related literature. Section 3 describes the IGA-based algorithm approximating a quadrilateral patch. In Section 4, the G1 continuous stitching scheme between neighboring patches is described with the adaptive fitting approach. The experimental results are reported in Section 5. Section 6 concludes the paper.

2. Related work

B-spline surface approximation based on optimization method. Generally, there are three main steps to fit a 3D point set with a B-spline curve or surface: parameterize the data points, place the knots, and apply a least-square approximation [14]. One

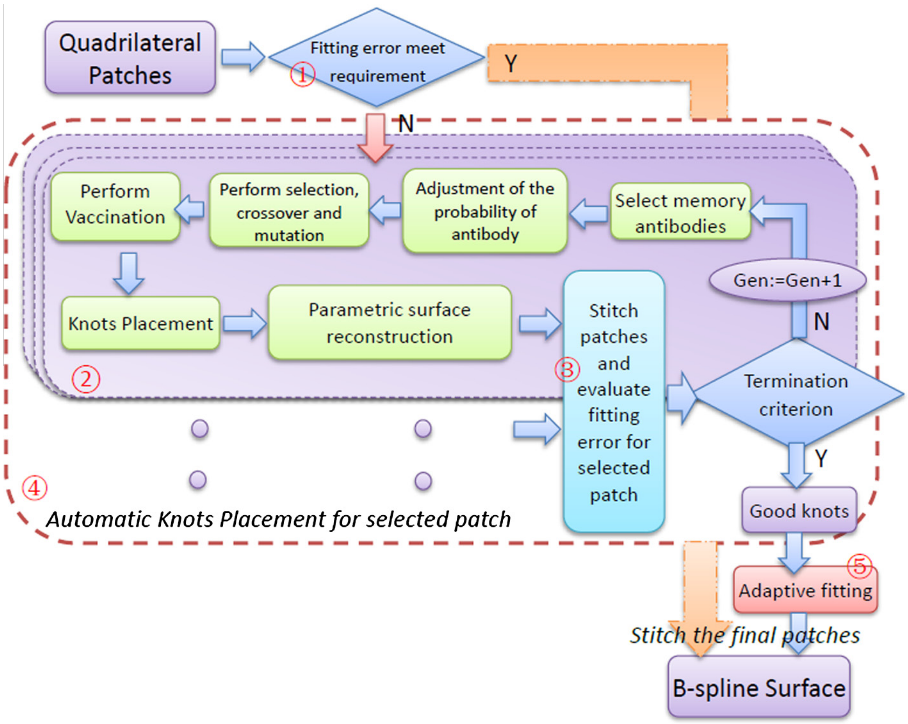
Quadrilateral Patches
Fitting error meet requirement
①
Y
N
Perform Vaccination
Perform selection, crossover and mutation
Adjustment of the probability of antibody
Select memory antibodies
Gen:=Gen+1
N
Knots Placement
Parametric surface reconstruction
Stitch patches and evaluate fitting error for selected patch
③
②
Termination criterion
Y
④
Automatic Knots Placement for selected patch
Good knots
⑤
Adaptive fitting
Stitch the final patches
B-spline Surface

Fig. 1. Flowchart of the IGA-based point cloud fitting algorithm using B-spline surfaces.

of the key steps in using splines to approximate a point cloud successfully is to determine good placements for the knots. Several methods have been used to determine knot location in the approximation of B-spline curves. Yoshimoto [27] proposed a method that converts the original problem into a discrete combinatorial optimization problem by modeling candidates of the locations of knots as genes then solving the problem using a genetic algorithm (GA). In Ref. [21], Ulker modeled a curve as a continuous, non-linear and multivariate optimization problem with many local optima by treating the knots as variables following the idea of Yoshimoto. However, he suggested a different method, i.e., the artificial immune system (AIS) to solve the converted problem by modeling the candidates of the locations of knots as antibodies. Because of the advantage over GA in the memory mechanism and production system, the AIS-based method obtained a better solution. Ref. [30] proposed an adaptive knot placement method using a Gaussian mixture model (GMM)-based continuous optimization algorithm, and, based on the good performance of the evolutionary algorithm, the method achieved a satisfactory result. The GMM-based automatic knot placement algorithm can also be used to fit a point cloud that represents a closed curve target. With respect to surface reconstruction, Refs. [3,4] reconstructed B-spline surfaces from 3D data points using GA and PSO approaches respectively. However, the data clouds are all sampled from analytic surfaces, and the algorithm has not been applied to the stitching of different patches.

Computational intelligence method. Computational intelligence is a set of nature-inspired computational methodologies and approaches to address complex real-world problems. The key idea behind computational intelligence is to understand and simulate the intelligence of biological evolution. From this viewpoint, intelligence is generated from biological genetics, variation, growth and the external environment. The main methods of computational intelligence are artificial neural networks, particle swarm optimization, genetic programs, and evolutionary processes.

The genetic algorithm (GA) that was proposed by Hooland has been widely used in machine learning, adaptive control, signal processing and artificial life [5,6]. However, GA has two main drawbacks. One is the lack of a local searching ability, and the other is premature convergence. Thus, it is difficult for the approximation method to find the global optimum. Some ideas have been proposed to improve the GA. One of the most improved GA algorithms is the immune genetic algorithm, which was proposed based on the theory of immunity in biology. The IGA mainly constructs an immune operator by vaccination and immune selection, and the immune operator can improve the searching ability and increase the converging speed greatly [25].

PSO was proposed by Eberhart [1,9] as a population-based stochastic optimization technique inspired by the social behavior of bird flocking. PSO has been successfully applied in many research and application areas [12,23,24]. In the PSO algorithm, each solution to a certain problem is a particle in the search space. All of the particles form a population. The flying direction of each particle is the dynamical interaction of individual and social flying experience. After a certain number of iterations, the position vector of the best particle will be an approximate optimum solution to the problem.

3. Approximation of a quadrilateral patch using IGA

3.1. B-spline surface approximation

A B-spline surface with order $k \times k$ can be defined as follows [15]:

$$S(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} d_{i,j}N_{i,k}(u)N_{j,k}(v), \quad u_k \leqslant u \leqslant u_{m+1}, \quad v_l \leqslant v \leqslant v_{n+1} \tag{1}$$

where $N_{i,k}(u)$ and $N_{j,k}(v)$ are the B-spline basis functions with order, $d_{i,j}(i = 0, 1, \ldots, m; j = 0, 1, \ldots, n)$ are control points, $U = [u_0, u_1, \ldots, u_{m+k+1}]$ is the u-directional knot vector and $V = [v_0, v_1, \ldots, v_{n+k+1}]$ is the v-directional knot vector. The recursive relation of the B-spline is defined as follows:

$$N_{i,k}(u) = \frac{(u - u_i)}{u_{i+k} - u_i}N_{i,k-1}(u) + \frac{(u_{i+k+1} - u)}{u_{i+k+1} - u_{i+1}}N_{i+1,k-1}(u) \tag{2}$$

where

$$N_{i,0}(u) = \begin{cases} 1, & u_i \leqslant u \leqslant u_{i+1} \\ 0, & otherwise \end{cases} \tag{3}$$

In this study, the following problem is addressed. Given a set of scattered data points $q_r(r = 0, 1, \ldots, nm)$ and four B-spline boundary curves $\Gamma_i(i = 0, 1, 2, 3)$ pre-computed from the boundary points, compute a bicubic B-spline surface to fit the points and interpolate the boundary curves. Firstly, the basic Coons surface is constructed from the four boundary curves. Then, the inner points are projected onto the basic surface and parameterized into $q_r(u_k, v_k)$ [16]. The target B-spline surface can be defined by the least squares method as follows:

$$F = \sum_{r=0}^{nm}\|S(u_r, v_r) - q_r\| \tag{4}$$

The problem has several degrees of freedom (DOF); changing the knot vectors and the parameter value will generate a different surface [7]. The problem studied here is how to exploit the DOF to optimize a certain fairing goal to reduce the approximation error of the surface.

3.2. Overview of the IGA

The IGA is an outgrowth of the theory of GA and immune concepts. As described in Ref. [25], the IGA is proposed based on the biological immune mechanism. The algorithm represents the objective function of the problem that must be resolved as an antigen invading a living being. Analogously, the solution of the problem corresponds to an antibody produced by the immune system. The affinity between the antigen and the antibody describes the degree of approximation between the feasible solution and the optimal solution. The immune system performs specific defense against agents, the antigens, which are foreign or harmful to the body. The immune system has protective memory cells that can activate and produce large amounts of antibody, making the secondary response more rapid than the initial response. To maintain the diversity of antibodies and an immune balance, the antibody with a higher concentration is restrained more heavily.

Suppose the number of antibodies in the immune system is N, the length of every gene is M, and the chromosomes are real-number encoded. The antibody population can be defined using the following format:

$$\begin{aligned}
&Anti_1(c_1^1, c_2^1, c_3^1, \ldots, c_i^1, \ldots, c_{M-1}^1, c_M^1) \\
&Anti_2(c_1^2, c_2^2, c_3^2, \ldots, c_i^2, \ldots, c_{M-1}^2, c_M^2) \\
&\vdots \\
&Anti_j\left(c_1^j, c_2^j, c_3^j, \ldots, c_i^j, \ldots, c_{M-1}^j, c_M^j\right) \\
&\vdots \\
&Anti_N(c_1^N, c_2^N, c_3^N, \ldots, c_i^N, \ldots, c_{M-1}^N, c_M^N)
\end{aligned}$$

where $c_i^j(i = 1, \ldots, M;\ j = 1, \ldots, N)$ are genes chosen randomly from the initial values in the search space. The following are some brief definitions of antigen and antibody, antibody concentration, similarity of two antibodies, immunological memory, etc.

(1) Antigen and antibody: Antigen represents the object function and the constraints. Antibody represents all possible solutions to the object function.

(2) Concentration of antibody: Given two antibodies, $Anti_j\left(c_1^j, c_2^j, c_3^j, \ldots, c_i^j, \ldots, c_{M-1}^j, c_M^j\right)$ and $Anti_k(c_1^k, c_2^k, c_3^k, \ldots, c_i^k, \ldots, c_{M-1}^k, c_M^k)$, if their Euclidean distance, recorded as $d(Anti_j, Anti_k) = \sqrt{\sum_{i=1}\left(c_i^j - c_i^k\right)^2}$, is no greater than ζ (a given smaller constant), then $Anti_j$ and $Anti_k$ are considered to be the same or similar. The concentration of antibodies refers to the proportion of same or similar antibodies in the whole population.

(3) Immunological memory and vaccination: When an antibody and an antigen have higher affinity, it means that the antibody is closer to the optimal solution. Immunological memory is a table of antibodies that have higher affinities. For each iteration, the individuals with higher affinities are added to the antibody memory table to achieve a faster convergence. Given an individual, a vaccination means modifying the genes on some bits in accordance with prior knowledge to gain higher fitness with greater probability [25].

(4) Promotion and inhibition of antibody production: During the iterative process, the concentration of the antibody that has high affinity will increase. However, high concentrations of some antibodies may lead to premature convergence of the algorithm. To overcome premature convergence and improve the performance of the algorithm, a method of regulating antibody concentration is introduced to suppress the number of identical or similar antibodies.

3.3. Adaptive knot adjustment by IGA for one patch

As described in the introduction, the number and distribution of the knots have considerable effects on the approximation precision of the B-spline surface. In this section, the IGA method used to optimize the bidirectional knots of the B-spline surface to achieve a more precise approximation is detailed. The algorithm includes the following six steps:

- Step 1. Initialize the random population.
  Treating the B-spline surface bidirectional knots as random variables, the antibody can be defined as $Anti_{ll} = \{u_0^{ll}, u_1^{ll}, \ldots, u_s^{ll}; v_0^{ll}, v_1^{ll}, \ldots, v_t^{ll}\}$, where $s = m + k + 1$ and $t = n + k + 1$ are the numbers of knots in the U and V directions, respectively. k is the order of the B-spline basis functions. Initially, for each $Anti_{ll} = \{u_0^{ll}, u_1^{ll}, \ldots, u_s^{ll}; v_0^{ll}, v_1^{ll}, \ldots, v_t^{ll}\}, u_i^{ll}$ and v_j^{ll} are chosen randomly from the initial knots $[(u_i + u_{i-1})/2, (u_i + u_{i+1})/2]$ and $[(v_j + v_{j-1})/2, (v_j + v_{j+1})/2]$, where $U = \{u_0, u_1, \ldots, u_k, \ldots, u_m, u_{m+1}, \ldots, u_{m+k+1}\}$ and $V = \{v_0, v_1, \ldots, v_k, \ldots, v_n, v_{n+1}, \ldots, v_{n+k+1}\}$ are the basic knot vectors in the U-direction and the V-direction. The basic knot vectors are pre-computed using the chord lengths method from the boundary data points. Then, the initial antibody population, $AntiPop = \{Anti_{ll};\ ll = 1, 2, \ldots, N\}$, where N is the number of antibodies in the population, is obtained.
- Step 2. Calculate the fitness between the antibodies and antigens, and select the best ones to save as memory antibodies.
  The fitness function is the approximate error of the measured data points $q_r(r = 0, 1, \ldots, nm)$ from the reconstructed surface, defined as follows:

$$f = \sum_{r=0}^{nm} |S(u_r, v_r) - q_r|, r = 0, 1, \ldots, nm \tag{5}$$

The best knots are those with the minimum fitness.

- Step 3. Promote and restrain antibody duplication based on their concentration.
  The antibody concentration, p_c, represents the proportion of same or similar antibodies in the population; p_{ca} represents the proportion of average antibodies in the population. The probability that an individual will be selected is defined as p_f; p_{fa} represents the antibody's average probability of being selected. The probability distribution for all individuals to be selected is defined as P.
  The probability that an antibody will be selected is adjusted according to the concentration and fitness probability as follows:

$$\begin{cases} p = p_f(1 + \alpha(p_{ca} - p_c)/(p_{ca} + p_c)), & \text{if } p_f > p_{fa} \\ p = p_f, & \text{if } p_f \leqslant p_{fa} \end{cases} \tag{6}$$

where $\alpha > 0$ is a constant. Eq. (6) shows that the higher the fitness of the individual, the greater the probability of being selected, and the higher the concentration of the individual, the less probability of being selected. This not only retains the value of good individuals but also ensures the diversity of the antibody population.

- Step 4. Perform selection, crossover and mutation.
  Perform the selection operation on the kth generation and use *AntiPop* to obtain *tempAntiPop* according to the probability distribution P. Then, perform crossover and mutation operations on *tempAntiPop* to obtain the new *tempAntiPop*′.
- Step 5. Perform the vaccination operation. Generate the new generation of population.
  Perform vaccination on *tempAntiPop*′ to obtain the $(k + 1)$th generation of population *AntiPop*. Vaccination means that the antibodies with smaller fitness values are replaced by some memory antibodies.

Table 1
The set of parameters for IGA.

Parameter	Man-belly	Man-head	Tooth	Shoe
Population size	50	50	50	50
String length of U	9	16	12	11
String length of V	9	16	12	11
Crossover rate	0.65	0.65	0.65	0.65
Mutation rate	0.3	0.3	0.3	0.3
Concentration regulate ratio	0.2	0.2	0.2	0.2
The maximum generation	50	120	120	120
The number of input vertices	709	43,166	7610	42,037
The number of input patches	1	39	26	34

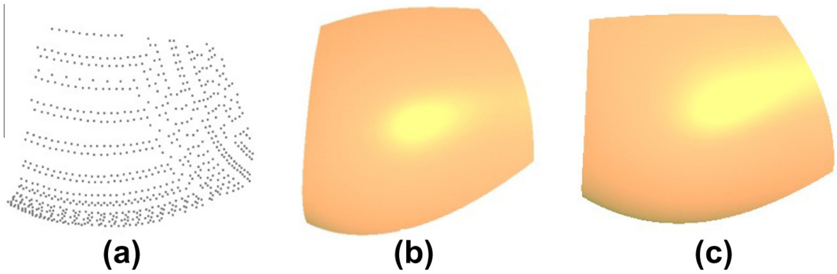

Fig. 2. (a) The input point cloud of a belly section. (b) The fitting B-spline surface without knots adjustment; the fitting error is 0.30184. (c) The fitting B-spline surface with knots adjustment by IGA; the fitting error is 0.25653.

- Step 6. Termination condition.
  If the whole fitness value is smaller than θ (a pre-defined fitness threshold), or the maximum number of iterations has been reached, then stop; otherwise go to step 2.
  In this section, an example of a point cloud with 709 vertices is used to illustrate the method. The parameters of the algorithm are shown in Table 1, and approximation results are shown in Fig. 2.

4. Smoothly stitch B-spline surface patches

Surface patches with adaptive knots adjustment often have different knot vectors. If they are stitched using the traditional knot vector compatible method, the stitching may lead to a rapid expansion of nodes [15]. To avoid this pitfall, a numerical approach is used to stitch the patches with approximate G1 continuity by adjusting the control points near the boundaries [17]. Because G1 continuity is based on G0 continuity, the B-spline surface patches with G0 continuity are stitched first (Section 4.1). Then, as described in Sections 4.2 and 4.3, the control points near the boundary are adjusted to meet the requirement of G1 continuity. In Section 4.4, PSO is used to find the best fit position of the sub-corner control points to complete the approximate G1 continuity stitch for the entire model surface. Finally, in Section 4.5, the adaptive fitting method of the algorithm is shown.

4.1. Stitch B-spline surfaces with G0 continuity

Two B-spline surface patches with G0 continuity are stitched in three steps. An example is shown in Fig. 3, where $S^l(u, 1)$ and $S^{l+1}(u, 0)$ are the two adjacent edges from different patches. First, the two adjacent edges are sampled. Then, according to the distance between the corresponding sampling points, the control points are adjusted. The process is as follows:

- Step 1: Sample curves $S^l(u, 1)$ and $S^{l+1}(u, 0)$ using the algorithm proposed in Ref. [29], obtaining the serial data points $\{Q_k, k = 0, 1, \ldots, sn\}$ and $\{Q'_k, k = 0, 1, \ldots, sn\}$. Compute the distances $\{dis_k, k = 0, 1, \ldots, sn\}$ between the corresponding points of the two data sets.
- Step 2: For every distance, if $dis_k > Tol$ (where Tol is a small constant chosen as the threshold, here it is 0.005), Q_k and Q'_k are adjusted to $Q^*_k = \frac{Q_k+Q'_k}{2}$. Otherwise, Q_k and Q'_k are considered to be approximately equal, and $Q^*_k = Q_k$.
- Step 3: Based on $Q^*_k (k = 0, 1, \ldots, sn)$, the common new control points $\{v_i, i = 0, 1, \ldots, sm\}$ of $S^l(u, 1)$ and $S^{l+1}(u, 0)$ are recalculated.

After the above adjustment, the boundary points can meet the requirement of G0 continuity.

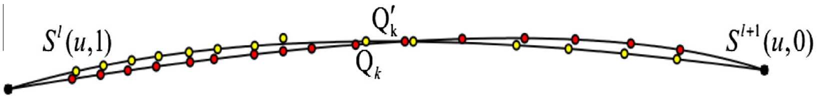

Fig. 3. Stitch two B-spline surface patches with G0 continuity, where the red dots ● represent sample points on $S^l(u, 1)$, and the yellow dots ○ represent the sample points on $S^{l+1}(u, 0)$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

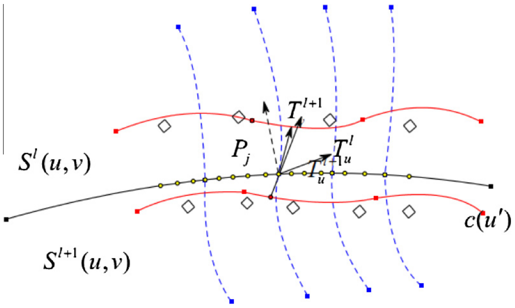

Fig. 4. G1 smooth stitch of two B-spline patches; ○ are sample points. ⇢ is the normal vector, → is the tangent vector ― is $c(u)$ and ◇ are the control points of $c(u)$ ╌ is the iso-parametric curve.

4.2. G1 smooth adjustment for the adjacent boundary curve

If the point $P_j = S^l(u', 1) = S^{l+1}(u', 0)$ on the adjacent boundary between $S^l(u, v)$ and $S^{l+1}(u, v)$ (as shown in Fig. 4) satisfies Eq. (7), $S^l(u, v)$ and $S^{l+1}(u, v)$ are G1 smooth on point P_j [16].

$$S_v^{l+1}(u',0) = aS_v^l(u',1) + bS_u^l(u',1) \tag{7}$$

where a and b are constants and $a > 0$.

The iso-parametric curve of the two surfaces (shown in Fig. 4) on the point P_j is described by the following equation:

$$\begin{cases} S^l(u',t) = \sum\limits_{j=0}^{tn^l} d_j^l N_{j,k}^l(t) = \sum\limits_{j=0}^{tn^l}\left[\sum\limits_{i=0}^{tm^l} d_{i,j}^{l+1} N_{i,k}^l(u')\right] N_{j,k}^l(t) \\ S^{l+1}(u',s) = \sum\limits_{j=0}^{tn^{l+1}} d_j^{l+1} N_{j,k}^{l+1}(s) = \sum\limits_{j=0}^{tn^{l+1}}\left[\sum\limits_{i=0}^{tm^{l+1}} d_{i,j}^{l+1} N_{i,k}^{l+1}(u')\right] N_{j,k}^{l+1}(s) \end{cases} \tag{8}$$

$P_j = d^l_{N^{l+1}} = d_0^{l+1}$ and the tangent vectors on point P_j are as follows:

$$T_u^{l+1} = T_u^l, T_v^{l+1} = \lambda^{l+1}\left(d_1^{l+1} - d_0^{l+1}\right), \quad \text{and } T_v^l = \lambda^l\left(d_{tn^l}^l - d_{tn^l-1}^l\right) \tag{9}$$

The corresponding normal vectors are as follows:

$$N_j^l = T_u^l \times T_v^l, \quad N_j^{l+1} = T_u^{l+1} \times T_v^{l+1} \tag{10}$$

If the angle arccos $\left|\left\langle N_j^l, N_j^{l+1}\right\rangle / \left\|N_j^l\right\|\left\|N_j^{l+1}\right\|\right| \geqslant \theta$ (θ is threshold), N_j^l and N_j^{l+1} should be adjusted to $N_j^{*l} = N_j^{*l+1} = \frac{N_j^l + N_j^{l+1}}{2}$. That means the control points $d^l_{tn^l-1}$ and d_1^{l+1} must be adjusted correspondingly as follows:

$$d^{*l}_{tn^l-1} = d^{*l}_{tn^l-1} - \left\langle d^{*l}_{tn^l-1}, N_j^{*l}\right\rangle \cdot N_j^{*l}, d_1^{*l+1} = d_1^{*l+1} - \left\langle d_1^{*l+1}, N_j^{*l+1}\right\rangle \cdot N_j^{*l+1} \tag{11}$$

Then $S^l(u, v)$ and $S^{l+1}(u, v)$ can meet the requirement of Eq. (7) on point P_j.

If the variable u' changes from 0 to 1, then the B-spline curve $c(u')$ determined by the surfaces $S^{l+1}(u, v)$ with control points at rank 1 are formed. The function $c(u')$ is defined as follows:

$$c(u') = \sum_{i=0}^{tm^{l+1}} d_{i,j_0}^{l+1} N_{i,k}^{l+1}(u') \tag{12}$$

Thus, the process of G1 adjustment is as follows:

- Step 1: Similar to the G0 smooth adjustment, curve $P^l(u,1)$ is sampled by a sequence of points $\{P_j, j = 0, 1, \ldots, rn\}$ first, and then, control points $d^{l}_{tn^l-1,j}$ and $d^{l+1}_{1,j}$ associated with P_j are adjusted according to Eq. (11).
- Step 2: Using $d_j^{*l+1}, j = 0, 1, \ldots, rn$ as data points, recalculate $c(u')$. Then the new control points of $c(u')$ are used to replace the old control points of $S^{l+1}(u, v)$ at rank 1. Similarly, perform the same procedure on $S^l(u, 1)$.
- Step 3: Use PSO to find the most suitable position of the sub-corner control points, as shown in Section 4.3.

4.3. G1 Smooth adjustment for sub-corner control points

PSO is used to find the optimum position of the sub-corner control points.

The procedure of the adjustment of the corner control points using PSO comprises the following four steps:

- Step 1. Initialize the particles with random positions and velocity vectors.

Assume there are n patches associated with a common corner. Then, the associated control points $\{d'_c, c = 0, 1, \ldots, n-1\}$ (shown in Fig. 5 as ⇈) form a 3n-dimensional search space. Initially, the PSO is composed of a swarm of pn possible particles (i.e., the possible positions of the control points) generated in the search space. A particle e at its $t-th$ iteration consists of a position vector and a velocity vector defined as follows:

$$\begin{aligned} x_e(t) &= \{x_e^g, g = 0, \ldots, 3(n-1)\} \\ v_e(t) &= \{v_e^g, g = 0, \ldots, 3(n-1)\}, \quad e = 1, \ldots, pn \end{aligned} \tag{13}$$

- Step 2. Calculate the best fitness.

Each particle in the swarm has a personal fitness value. Because the curves which were defined on $[u_i, \ldots, u_{i+k+1}]$ and $[v_j, \ldots, v_{j+k+1}]$ are affected heavily by the corner control point d'_c, the fitness function can be defined based on the sum of their normal vector angles as follows:

$$f_p = \min \sum_{l=0}^{2(n-1)} \sum_{r=0}^{w} \arccos \left| \left\langle N_r^l, N_r^{l+1} \right\rangle / \left\| N_r^l \right\| \left\| N_r^{l+1} \right\| \right| \tag{14}$$

where $N_r^l,\ N_r^{l+1}$ (shown in Fig. 5 ◇) are sample normal vectors defined in $[u_i, \ldots, u_{i+k+1}]$ and $[v_j, \ldots, v_{j+k+1}]$, $2(n-1)$ is the number of common boundaries in the singular point, and w is the number of sample points in $[u_i, \ldots, u_{i+k+1}]$ or $[v_j, \ldots, v_{j+k+1}]$. Update the fitness of each particle *pbest* and the global best fitness *gbest*, as follows:

$$pbest_{e,t} = \min\{f_{p,t}, t = 0, 1, \ldots, CurrentGen\} \tag{15}$$

$$gbest = \min\{pbest_{e,t}, e = 0, 1, \ldots, pn\} \tag{16}$$

- Step 3. Update the velocity vector and position for every particle.

Every particle will modify its velocity and position using the following formula:

$$v_e^g(t+1) = w \cdot v_e^g(t) + c_1 \cdot r_1 \cdot \left(pbest_e^g(t) - x_e^g(t)\right) + c_2 \cdot r_2 \cdot \left(gbest^g(t) - x_e^g(t)\right) \tag{17}$$

$$x_e^g(t+1) = x_e^g(t) + v_e^g(t+1) \tag{18}$$

where $e = 1, \ldots, pn$ is the particle's index in the swarm, $g = 0, \ldots, 3(n-1)$ represents its corresponding position in the search space, w is the positive inertial weight, c_1 and c_2 are the cognitive and social learning factors, and r_1 and r_2 are two random numbers distributed between zero and one. In this study, the values of w, c_1 and c_2 are 0.5, 1.5 and 1.5, respectively. f_p is less than or equal to 0.3.

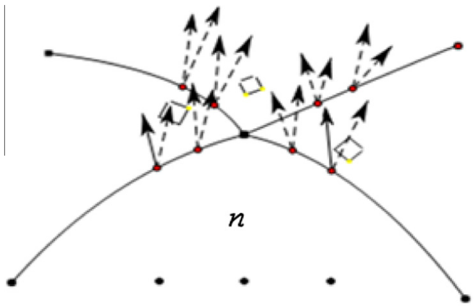

Fig. 5. The sub-corner control points ◇. ⇈ represents the normal vector of the common point on the boundary.

- Step 4. If the given stop criteria are not met, set $t = t + 1$, and go to step (2), else select the *gbest* position as the optimal solution.

After the above adjustment, boundary points can meet the requirements of $\sum_{l=0}^{2(n-1)}\sum_{r=0}^{w}\arccos\left|\left\langle N_r^l, N_r^{l+1}\right\rangle/\left\|N_r^l\right\|\left\|N_r^{l+1}\right\|\right| \leqslant \partial$ (∂ is a small constant) and can be regarded as G1 smooth at the corner point.

4.4. Adaptive fitting

In reverse engineering, the fitting error is of essential importance [11]. Thus, the fitting method must have adaptive fitting capability. Suppose that the fitting error of a B-spline patch associated with a quadrilateral patch H does not meet the predefined tolerance. The middle points of the four boundaries of the fitting surface are v_i, $i = 0, 1, 2, 3$. By connecting two pairs of points, v_0 and v_2, v_1 and v_3 with two lines, and projecting them onto the fitting surface, two intersecting curves on the surface are obtained. The intersection point is designated by v_4. Thus, the patch T can be segmented into four quadrilateral patches (see Fig. 10a). The method described in Sections 4.1, 4.2 and 4.3 is used to stitch the sub-segmented patch. However, in the T-shaped corner points, i.e., the midpoint of the four boundaries of H, the adjustment of the control points should be based on the sample points of their adjacent edges. As shown in Fig. 6, the dashed arrow vectors represent the normal vectors of the sample points on the boundary of H. These vectors can be adjusted for G1 continuity. The solid line vectors are the normal vectors associated with the adjacent quadrilateral patch. Although they all share common sample points with the normal vectors at the boundary of H, they are fixed. By adjusting the control points at the corner (shown in Fig. 6 ⋎), the angle of the normal vector pairs (i.e., the solid line vector and corresponding dashed vector share common sample data, as shown in Fig. 6) can be adjusted to below a very small value. To this end, the fitness function in Section 4.4 is modified as follows:

$$f_p = \min\sum_{l=0}^{2}\sum_{r=0}^{w}\arccos\left|\left\langle N_r^l, N_r^{l+1}\right\rangle/\left\|N_r^l\right\|\left\|N_r^{l+1}\right\|\right| \tag{19}$$

where $N_r^l,\ r = 0, 1, \ldots, w$ is a fixed normal vector obtained through the sample points on the boundary. The superscript l denotes the adjacent quadrilateral patch of H, and $l + 1$ denotes H.

Using this technique, not only the fitting error is reduced, but also the patches are approximate G1 smoothly stitched.

5. Experimental results

To demonstrate the proposed IGA-based point cloud fitting algorithm, three examples are used. All computations in this study are performed on an HP Workstation with 2.13 GHz, Intel Xeon CPU E5506 and 2 GB RAM. All algorithms are coded using the C++ programming language with the help of the Microsoft Foundation Class (MFC) Library and OpenGL. For each example, the point cloud is first segmented into quadrilateral patches, and then the B-spline surfaces are reconstructed using the least squares method. Because some patches are well approximated by the least squares method, it is not necessary to refine all the patches. The patches to be refined are selected according to the average fitness error (shown in Table 2). The values of the parameters used in the IGA are presented in Table 1 and are determined by trial-and-error. Considering the randomness of the intelligent algorithm, all verification experiments were repeated 10 times, and the results presented in this paper are the averages of the 10 tests.

The first example is a man-head model, shown in Fig. 7. The model is segmented into 39 quadrilateral patches. There are 5 patches that should be refined according to the fitting error (as shown in Fig. 7b; those patches are shown in blue color). The fitting errors against generations of IGA for the 5 patches are shown in Table 3.

As shown in Fig. 8, the second example is a tooth model segmented into 26 pieces. According to the fitting errors, 6 patches are selected for refinement. The fitting errors of these 6 patches after being optimized by IGA are shown in Table 4. In Fig. 9, the shoe model is segmented into 34 pieces, and patches 26, 28, 32 and 34 are selected to be refined, according to the fitting errors. The fitting errors of these 4 patches after being optimized by IGA are shown in Table 5.

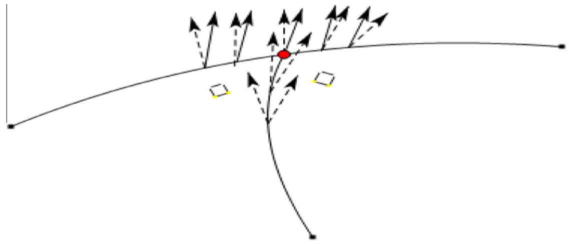

Fig. 6. T-shaped corner patch, ● represents the middle point of the boundary. ◇ represents the sub corner control points. ⇡ represents the normal vector that must be changed and ↑ represents a fixed normal vector.

Table 2
Average fitting errors of every patch of the man-head, tooth and shoe models.

Model	Average fitting error of all patches (units of error are 10^{-3})								
	P1	P2	P3	...	P15	P16	P17	P18	...
Man-head	6.551	7.495	7.108	...	7.632	6.510	6.947	5.650	...
Tooth	2.993	4.735	3.83	...	1.184	1.204	1.480	1.371	...
Shoe	3.590	3.309	5.106	...	7.414	4.333	6.852	2.477	...

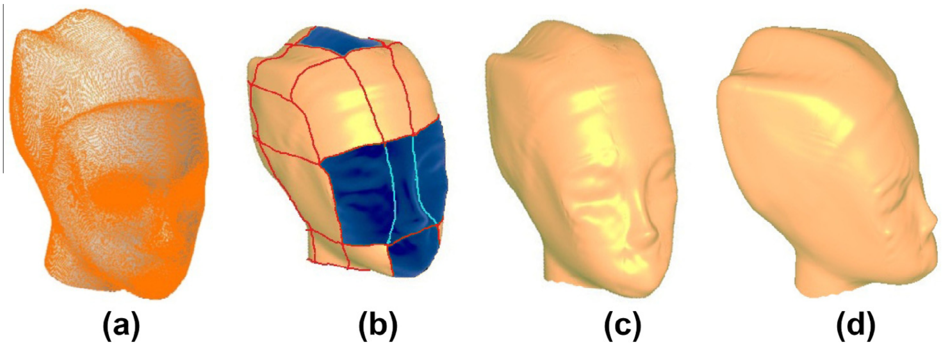

Fig. 7. (a) The input man-head point cloud. (b) The B-spline fitting surface using the least squares method, the surface patches in blue are those need to be optimized by IGA. (c) The final fitting B-spline surface from the front view and (d) from the side view. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Average fitting errors against generations of IGA for the selected patches of the man head model.

Generation	Patches of man-head (units of error are 10^{-3})				
	P4	P5	P6	P8	P15
1	8.531	8.932	8.912	7.676	7.632
5	7.818	7.356	7.637	7.201	7.142
20	7.644	6.909	7.505	7.095	7.137
⋮	⋮	⋮	⋮	⋮	⋮
80	7.398	6.850	7.154	6.972	7.000
120	7.131	6.837	7.138	6.807	6.696

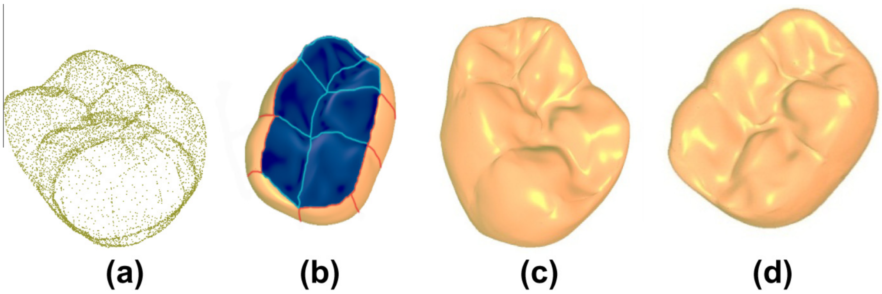

Fig. 8. (a) The input tooth point cloud. (b) The B-spline fitting surface using the least squares method, the surface patches in blue are those need to be optimized by IGA. (c) The final fitting B-spline surface from one view and (d) from another view. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5. Average fitting errors against generations of IGA for the selected patches of the shoe model.

If the fitting error of reconstruction does not meet the tolerance requirement after optimization, the adaptive fitting method is used. As shown in Fig. 10a, the eye parts in the man-head model do not meet the requirement. For those two parts, every patch is segmented into four sub-patches by using the algorithm described in Section 4.4. The results are shown in Fig. 10b and c. The fitting errors, listed in Table 6, show that the fitting errors are reduced by more than half.

Table 4
Average fitting errors against generations of IGA for the selected patches of the tooth model.

Generation	Patches of tooth (units of error are 10^{-3})					
	P1	P2	P3	P4	P5	P6
1	2.827	4.616	3.243	1.634	1.664	2.344
5	2.560	4.238	3.093	1.269	1.407	2.104
20	2.523	3.975	3.087	1.269	1.407	1.900
⋮	⋮	⋮	⋮	⋮	⋮	⋮
80	2.485	3.894	3.066	1.169	1.209	1.717
120	2.473	3.876	3.063	1.169	1.209	1.717

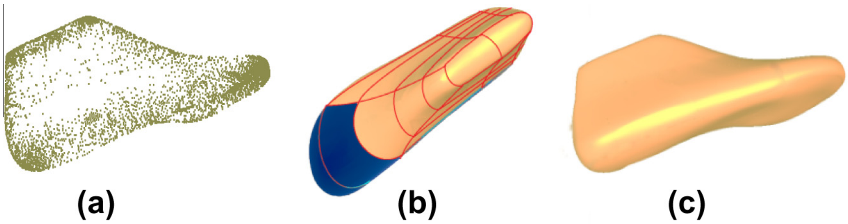

Fig. 9. (a) The input shoe point cloud. (b) The B-spline fitting surface using the least squares method, the surface patches in blue are those need to be optimized by IGA. (c) The final B-spline fitting surface. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Fitness of the Shoe four patches.

Generation	Patches of shoe (units of error are 10^{-3})			
	P26	P28	P32	P34
1	8.474	8.282	9.119	7.478
5	7.486	8.269	7.708	6.554
20	7.377	8.248	7.651	6.194
⋮	⋮	⋮	⋮	⋮
80	7.085	7.979	7.009	6.010
120	7.059	7.926	6.947	5.985

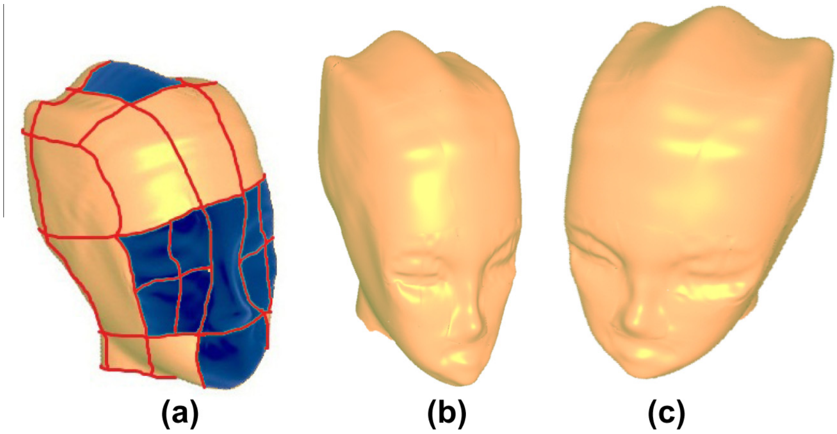

Fig. 10. (a) The eyes are subdivided. (b) The fitting patches after being subdivided, front view and (c) back view.

5.1. Comparison with other methods

The performance of this approach has been compared with the Bezier algorithm. To make an objective comparison, the vertices are parameterized by the Coons surface based method [16], as described in Section 3.1. The input patches and final

Table 6
Fitting errors of the eye parts of the man-head model after adaptive fitting.

Fitting errors before and after adaptive fitting (units of error are 10^{-3})					
Original	Sub-divided				
Patch4	Patch4-1	Patch4-2	Patch4-3	Patch4-4	Average
8.531	3.753	3.818	2.837	2.561	3.231
Patch6	Patch6-1	Patch6-2	Patch6-3	Patch6-4	Average
8.912	4.456	5.969	2.556	2.613	3.899

Table 7
Experimental results of the Bezier-based and the B-spline-based methods.

Model	Vertices	Input patches	Final patches	Average fitting error (10^{-3})	
				Proposed method	Bezier based method
Man-head	43,166	39	45	5.0410	7.6718
Tooth	7610	26	26	1.3700	1.9110
Shoe	42,037	34	34	5. 0896	6.5210

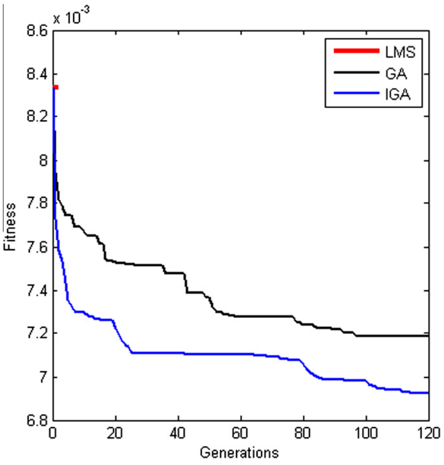

Fig. 11. Fitness against generations for the man-head.

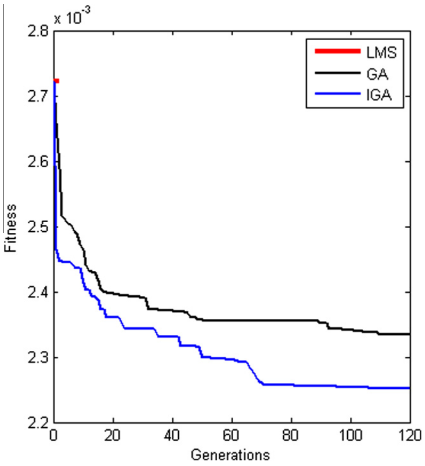

Fig. 12. Fitness against generations for the tooth.

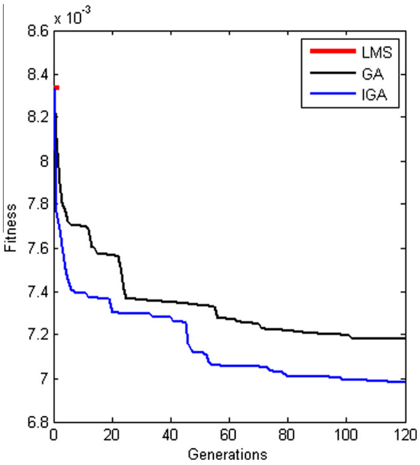

Fig. 13. Fitness against generations for the shoe.

patches are uniform, the 3D point cloud is reconstructed using the new method and a fifth-order Bezier surface method [11]. The experimental results are shown in Table 7. The results show that this new method achieves better precision because the B-spline surfaces provide more free control points for optimizing the parametric surfaces.

The proposed method is also compared with the GA-based method [4]. To make an objective comparison, all the conditions are the same except the node vector selecting process. The average fitness values vs. generations of IGA and GA for the surface reconstruction are shown in Figs. 11–13. The results show that both the GA and IGA methods give an accuracy improvement of 7%-15% over the least squares B-spline surface approximation method. The IGA-based surface reconstruction method has better accuracy than the GA-based method.

6. Conclusions

In this study, an adaptive fitting approach for a 3D point cloud, based on IGA optimization algorithm, is developed. The knot vectors of the B-spline surface patches are treated as antibodies and optimized by the IGA to determine the most reasonable location of the knots to reduce the fitting error of the B-spline surface patches. The quadrilateral patches are stitched by adjusting the position of the control points of the edge curves for every patch, and PSO is used to adjust the sub-corner control points to achieve approximate G1 continuity. The proposed algorithm achieves better fitting precision because IGA has an advantage over GA with respect to memory mechanism and the production system. The proposed method is also compared with the Bezier method. Because of the local support property, the B-spline surfaces could provide more free control points for optimizing the parametric surfaces. Thus the proposed algorithm achieves better fitting precision over the Bezier method as well. The experimental results also show that the proposed algorithm performs better than the GA-based algorithm and the Bezier-based surface reconstruction method.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China (Nos. 61020106001, 61173078, 61173079, 61103150, 61203105), the Natural Science Foundation of Shandong Province, China (Nos. ZR2010FM047, ZR2011FL016, ZR2012FQ016), Program for New Century Excellent Talents in University (No. NCET-10-0863).

References

[1] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, 1995, pp. 39–43.
[2] M. Eck, H. Hoppe, Automatic reconstruction of B-spline surfaces of arbitrary topological type, in: ACM SIGGRAPH 1996, New York, USA, 1996, pp. 325–334.
[3] A. Gálvez, A. Iglesias, Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points, Information Sciences 192 (2012) 174–192.
[4] A. Gálvez, A. Iglesias, J. Puig-Pey, Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction, Information Sciences 182 (2012) 56–76.
[5] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[6] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Machine Learning 3 (1988) 95–99.
[7] R. Goldenthal, M. Bercovier, Spline curve approximation and design by optimal control over the knots, Computing 72 (2004) 53–64.
[9] J. Kennedy, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Washington, USA, 1995, pp. 1942–1948.
[10] V. Krishnamurthy, M. Levoy, Fitting smooth surfaces to dense polygon meshes, in: ACM SIGGRAPH 1996, New York, USA, 1996, pp. 313–324.

[11] H. Lin, W. Chen, H. Bao, Adaptive patch-based mesh fitting for reverse engineering, Computer-Aided Design 39 (2007) 1134–1142.
[12] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P.N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Information Sciences 209 (2012) 16–36.
[13] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, H.P. Seidel, Multi-level partition of unity implicits, in: ACM SIGGRAPH 2003, New York, USA, 2003, pp. 467–470.
[14] H. Parka, J.H. Leeb, B-spline curve fitting based on adaptive curve refinement using dominant points, Computer-Aided Design 39 (2007) 439–451.
[15] L.A. Piegl, W. Tiller, The NURBS Book, Springer, 1997.
[16] L.A. Piegl, W. Tiller, Parametrization for surfacefitting in reverse engineering, Computer-Aided Design 33 (2001) 593–603.
[17] X. Qu, T. Ning, P. Xi, Smooth joining between adjacent B-spline surfaces, Computer-aided Design & Computer Graphics 16 (2004) 138–141.
[18] T.W. Sederberg, J. Zheng, D. Sewell, M. Sabin, Non-uniform recursive subdivision surfaces, in: ACM SIGGRAPH 1998, New York, USA, 1998, pp. 387–394.
[19] X. Shi, T. Wang, P. Wu, F. Liu, Reconstruction of convergen to G1 smooth B-spline surfaces, Computer Aided Geometric Design 21 (2004) 889–913.
[20] L.A. Shirman, C.H. Séquin, Local surface interpolation with Bézier patches, Computer Aided Geometric Design 4 (1987) 279–295.
[21] J.T. Tsaia, W.H. Hob, T.K. Liuc, J.H. Chouc, Improved immune algorithm for global numerical optimization and job-shop scheduling problems, Applied Mathematics and Computation 194 (2007) 406–424.
[22] T. Varadya, R.R. Martina, J. Coxa, Reverse engineering of geometric models-an introduction, Computer-Aided Design 29 (1997) 255–268.
[23] H. Wang, H. Sun, S.C. Li, Rahnamayan, J. Pan, Diversity enhanced particle swarm optimization with neighborhood search, Information Sciences 223 (2013) 119–135.
[24] L. Wang, B. Yang, X. Zhao, Y. Chen, J. Chang, Reverse extraction of early-age hydration kinetic equation from observed data of Portland cement, Science China Technological Sciences 53 (2010) 1540–1553.
[25] J.J.V. Wijk, Bicubic patches for approximating non-rectangular control-point meshes, Computer Aided Geometric Design 3 (1986) 1–13.
[26] A.P. Witkin, P.S. Heckbert, Using particles to sample and control implicit surfaces, in: ACM SIGGRAPH 1994, New York, USA, 1994, pp. 269–277.
[27] F. Yoshimotoa, T. Haradab, Y. Yoshimotoc, Data fitting with a spline using a real-coded genetic algorithm, Computer-Aided Design 35 (2003) 751–760.
[28] S. Zhang, Z. Li, H. Zhang, J. Yong, Multi-resolution Mesh Fitting by B-spline Surfaces for Reverse Engineering, in: 12th International Conference on Computer-Aided Design and Computer Graphics, Jinan, China, 2011, pp. 251–257.
[29] X. Zhao, Y. Yin, B. Yang, Dominant point detecting based non-uniform B-spline approximation for grain contour, Science China Technological Sciences 50 (2007) 90–96.
[30] X. Zhao, C. Zhang, B. Yang, P. Li, Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation, Computer-Aided Design 43 (2011) 598–604.

Further Reading

[8] L. Jiao, L. Wang, A novel genetic algorithm based on immunity, IEEE Systems, Man, and Cybernetics Society (2000) 552–561.