Journal of Computers Vol. 28, No. 6, 2017, pp. 184-194
doi:10.3966/199115992017122806016

Fusion Algorithm Based On Point Cloud for Non-uniform Rational B-spline Surface Reconstruction

Wei Fu[1,2], Lu-shen Wu[1*]

[1] School of Mechanical and Electrical Engineering, Nanchang University,
Nanchang, 330031, China, fuwei75@163.com

[2] School of Aeronautic Manufacturing Engineering, Nanchang Hangkong University,
Nanchang, 330069, China, wulushen@163.com

Received 30 September 2016; Revised 03 March 2017; Accepted 30 March 2017

Abstract. Surface reconstruction is a very challenging problem arising in a wide variety of applications such as CAD design, data visualization, virtual reality, reverse engineering and so on. In this paper, surface reconstruction consists of five main tasks (1) the parameterization of the point cloud data, (2) the surface fitting based on IGA adaptive node, (3) the surface control points calculated, (4) point cloud data segmentation and (5) smoothly stitch surface. Combination of immune genetic algorithm and particle swarm optimization (PSO) is proposed in surface fitting. A new immune genetic algorithm (IGA) for point cloud fitting that fits a 3D point cloud using a B-spline surface with approximate G1 continuity is presented. The B-spline patches are stitched together with approximate G1 continuity with a numerical method and the particle swarm optimization algorithm. Six examples including B-spline, shell, torus, vase, trim-Star, and fandisk surfaces, illustrate the good performance of our approach. Our experiments show that our proposed method outperforms all previous approaches in terms of accuracy and flexibility.

Keywords: immune genetic, particle swarm optimization, point cloud, surface reconstruction

1 Introduction

Surface reconstruction has been a hot topic of research in the past 20 years or so. For the unknown surface, We usually gets the 3D point cloud data by 3D scanning equipment. Surface reconstruction includes 2 main parts: surface parameterization and surface fitting. For the point cloud obtaining surface fitting appropriate, these are very crucial.

The point set is used to fit the target surface, and make these points from the target surface with the minimum deviation, this process is called surface fitting. The surface fitting is widely used in CAD design, data visualization, virtual reality, many fields of reverse engineering at home and abroad. surface fitting has rectangular surface fitting algorithm, triangle surface fitting, subdivision surface fitting, and genetic algorithm surface fitting etc.

Generally, the reconstruction of a combined parametric surface from a point cloud can be subdivided into 4steps: point cloud data parameterization; caculating adaptive knot; point cloud segmentation and 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 [1]. Earlier studies using the Bezier patch to generate a smooth surface include Eck, Tsaia, Wijk etc. [2-4]. Recently, Lin, Chen and Bao [1] proposed an adaptive mesh-fitting algorithm that fits a triangular model with G1 smoothly stitched Bezier patches.

For B spline fitting algorithm, we can refer to literature. Gofukoa, Tamura and Maekawa [5] proposed an interpolation algorithm based on the plane three spline curves which is based on the iteration geometry, and the tangent vector at the point set and the point set. Abbas, Nasri and Maekawa [6] presented a

* Corresponding Author

method of producing a uniform spline curve based on the unit normal and curve constraint. But this method can not give the vice normal vector of the space curve.

2 Related Work

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 [7]. One 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, Haradab and Yoshimotoc [8] and Tsaia, Hob, Liuc and Chouc [9] have proposed a method to solve the converter problem by modeling the candidates of the locations of knots as antibodies. Zhao, Zhang, Yang and Li [10] proposed an improved adaptive knot placement method based on GMM algorithm. Gálvez and Iglesias, Gálvez, Iglesias and Puig-Pey [11-12] reconstructed surfaces from points cloud using GA and PSO approaches respectively. However, the improved algorithm has not been applied to the different stitching patches.

The genetic algorithm that was first proposed by Hooland in 1975, originated from the study of natural and artificial adaptive systems [13-14]. However, genetic algorithm is the lack of a local searching ability and premature convergence. Thus, to find the golbal optimum is difficult.

Several methods have been proposed to improve the genetic algorithm. One of the most important method is the immune genetic algorithm [15-16].

Jiao et al proposed the immune genetic algorithm (IGA) [35]. The idea is based on the biological immune mechanism proposed for improving genetic algorithm, it will solve the problem of the objective function corresponds to the invading antigens of life, the problems of solutions corresponding to the immune system produces antibodies. When solving practical problems, the objective function and constraint condition as antigen input, then generating initial antibody group, through a series of genetic operation and antibody affinity calculation, and antibody concentration to maintain the diversity of the antibodies to find the biggest affinity antibody and antigen population.

Particle swarm optimization algorithm (PSO) was proposed by Kenndey and Eberhar [17-19] in the United States in 1995. This algorithm is an evolutionary computation technique based on swarm intelligence, which is inspired by the behavior of birds, schools of fish and human society. According to the research and practice in recent years, PSO has the advantages of fast convergence, high quality and good robustness, so it is especially suitable for engineering applications. But PSO algorithm has drawbacks, the most major problem is easy to produce premature convergence, poor local search optimization ability. According to this, there have been some improvements of PSO algorithm in recent years, such as fuzzy PSO algorithm, for example, proposed by literature [29-31], intelligent PSO algorithm, hybrid PSO algorithm etc. For the various PSO algorithm and the hybrid algorithm, most basically using a strategy to improve the algorithm, or combined with other algorithms. At the same time using the mixture of two strategies method is less.

PSO algorithm research is still in the initial stage [20-23]. It is a relatively new computing technology evolution based on swarm intelligence theory, and its basic research is still relatively poor, such as the convergence problem, how to improve the search efficiency problem of the algorithm in the solution space, and study the algorithm model in theory. In addition, parameters choice of the PSO algorithm depends on the specific problems, in general, to design a suitable parameter to repeatedly experience. How to choose and design parameters, making it less depending on the specific problems, it will greatly promote the development and application of the PSO algorithm.

In general, reconstruction of the three dimensional CAD model is the core of the reverse engineering and the main purpose, it is basis of the subsequent performance analysis, processing and manufacturing, rapid prototyping and product innovation design, and is the key and the most complicated part of reverse engineering.

In this paper, we address the above-mentioned problems by using a fusion method with IGA and PSO. In particular, given a set of 3D data points assumed to lie on an unknown NURBS surface of a certain order, the algorithm is applied to determine all relevant surface data (i.e. parametric values of data points, knot vectors, control points and their weights) thus actually returning the reconstructed NURBS surface.

3 NURBS Surfaces

Let $k \times l$ order NURBS surface expression is

$$S(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} w_{i,j} d_{i,j} N_{i,k}(u) N_{j,l}(v)}{\sum_{i=0}^{m}\sum_{j=0}^{n} w_{i,j} N_{i,k}(u) N_{j,l}(v)} . \quad \textbf{(1)}$$

where $w_{i,j}(i = 0,1...,m; 0,1,...n)$ is weight.

$\{d_{i,j}\}_{i=0,...,m; j=0,...n}$ is surface control points. $N_{i,k}(u)$ and $N_{j,k}(v)$ is B Spline base function.

4 Our Method

4.1 Points Cloud Data Parameterization

Genetic algorithm is used to the point cloud data parameter. There are two main ways of gene encoding: binary encoding and real encoding. For more variable optimization problem, we use real encoding genetic algorithm to express the genetic information. In order to improve the optimization probability, we can use high crossover probability and mutation probability.

Firstly, determining the parameter vector $T = [t_1, t_2, ..., t_m]$ and node vector U, then solving a linear optimization problem of control vertex $P_j (j = 1,2,...,l)$. As shown in the formula (2):

$$d = (Q - SP)^T (Q - SP) = \min . \quad \textbf{(2)}$$

Where $S = \begin{pmatrix} s_1(t_1) & s_2(t_1)\cdots & s_l(t_1) \\ s_1(t_2) & s_2(t_2)\cdots & s_l(t_2) \\ \vdots & \vdots & \vdots \\ s_1(t_m) & s_2(t_m)\cdots & s_l(t_m) \end{pmatrix}, Q = \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{bmatrix}, P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}$.

According to the formula (2), selecting the parameters t_i $(i = 1,2,\cdots,m)$ and the node $u_i (i = 1,\cdots,n + k + 1)$ and indepnedent variables $n - k + m$, therefore each individual has $n - k + m$ gene. $g_i (i = 1,\cdots n - k + m)$ is the the random initialization number in the interval [0,1].

Seclection opterator of the roulette is adopted in the paper. If individual adaptive value id D_i $(i = 1,2\cdots,n)$, the provability of being chosen is

$$P_i = \frac{D_i}{\sum_{i=1}^{n} D_i} . \quad \textbf{(3)}$$

Each gene in a population of a group of individuals has a mutation of the prior set of mutation probability p_m. Specific processes are as follows: each gene of the group obtained after crossing, A random number a is generated in (0,1) uniform. If $a < p_m$, that is in (0,1), to take a value at random g', assignment the gene g.

4.2 Adaptive Knot Based on IGA

The distribution and number of knots need to consider the approximation accuracy of the B-spline surface. Immune genetic algorithm (IGA) can be obtained by optimizing the B-spline surface of bidirectional nodes to get more accurate fitting surface. The algorithm consists of the following six steps:

(1) The random initialization of population, input scattered points cloud $P_i (i = 1,2,\cdots,p)$ and other initial parameters (m,n,s,t,L), m and n is the number of control points along the dirction vector of u and

v respectively. s and t is the number of spline along the dirction vector of u and v respectively

(2) Calculate the similarity of antibody and antigen

Firstly, the basic Coons surface is constructed by four B spline curves. Then, the internal points are mapped to the basic surface and their parameters are formed $q_r(u_k, v_k)$, B spline surface can be calculated by the least square method:

$$F = \sum_{r=0}^{nm} \left| S(u_r, v_r) - q_r \right|, r = 0,1,...,nm . \tag{4}$$

Where, q_r is the measurement of the data point set.

(3) Promote and restrain of antibody

The concentration of the antibody p_c is the proportion of similar antibodies in the population. p_{cv} is the proportion of the average antibodies in the population. The probability that an individual will be selected is p_f is the antibody's average probability of being selected. p is The probability distribution of all individuals to be selected.

$$\begin{cases} p = p_f(1+\alpha(p_{cv}-p_c)/(p_{cv}+p_c), p_f > p_{fv} \\ p = p_f, p_f \le p_{fv} \end{cases} . \tag{5}$$

where a > 0, The Eq. (5) shows that in the current population concentration. To decide whether to increase or reduce the probability of individual choice, if the concentration is low, to increase the probability of individual choice, that is promotion, and vice versa, that is, inhibition.

(4) The production of antibodies, the implementation of crossover and mutation in population.

(5) Perform vaccination operation, generate the new generation of population.

Perform vaccination for CK to obtain the population of DK.

(6) Termination condition

If the whole fitness value is smaller than ε (a pre-defined fitness threshold), or the maximum number of iterations has been reached, then stop; otherwise go to step 2

4.3 Curve Control Points Calculation

The control points of B spline surfaces are determined by the least square method. Given m-1 data points. $q_{0,}q_1, \cdots q_{m-1} (m > n)$, a B-spline surface with k orders of n+1 control points is computed as follows:

$$p(u) = \sum_{j=0}^{n-2} d_j N_{j,k}(u), \ u \in [0,1] . \tag{6}$$

Where $d_j (j = 0,1,2,\cdots,n-2)$: control points.

$N_{j,k}(u)(j = 0,1,\cdots,n-1)$: B spline basis function.

The least square method is used to approximate the k times B-spline curve.

A minimum value of the objective function f on the N-1 control points d_j can be defined as follows:

$$f = \sum_{i=1}^{m-1} [q_i - p(\tilde{u}_i)]^2 . \tag{7}$$

Where $\tilde{u}_i (i = 1,2,\cdots,m-1)$: the parameter values for i data point parameter values.

To make the objective function f be minimum, it should make the derivative of the N-1 control point $d_j (j = 1,2,\cdots n-1)$ be zero. Function f on the derivative of the L control point d_l can be defined as follows:

$$\frac{\partial f}{\partial d_l} = \sum_{i=1}^{m-1} [-2r_i N_{l,k}(\tilde{u}_i) + 2N_{l,k}(\tilde{u}_i) \times \sum_{j=1}^{n} d_j N_{j,k}(\tilde{u}_i)] . \tag{8}$$

So,

$$\sum_{j=1}^{n-1}\sum_{i=1}^{m-1} d_j N_{j,k}(\tilde{u}_i) N_{l,k}(\tilde{u}_i) = \sum_{i=1}^{m-1} r_i N_{l,k}(\tilde{u}_i) . \quad \textbf{(9)}$$

Here, a linear equation is derived to control point D for the unknown quantity.
Let $l = 1, 2, \cdots, n-1$ be a set of equations containing N-1 of the unknown quantity:

$$(N^T N) D = R . \quad \textbf{(10)}$$

$$N = \begin{pmatrix} N_{1,k}(\tilde{u}_1) & \dots & N_{n-1,k}(\tilde{u}_1) \\ \vdots & \ddots & \vdots \\ N_{1,k}(\tilde{u}_{m-1}) & \cdots & N_{n-1,k}(\tilde{u}_{m-1}) \end{pmatrix} . \quad \textbf{(11)}$$

$$R = \begin{bmatrix} N_{1,k}(\tilde{u}_1) q_1 + & \dots & + N_{1,k}(\tilde{u}_{m-1}) q_{m-1} \\ \vdots & \vdots & \vdots \\ N_{n-1,k}(\tilde{u}_1) q_1 + & \cdots & N_{n-1,k}(\tilde{u}_{m-1}) q_{m-1} \end{bmatrix} . \quad \textbf{(12)}$$

$$D = \begin{bmatrix} d_1 \\ \vdots \\ d_{n-1} \end{bmatrix} . \quad \textbf{(13)}$$

Where N: $(m-1)\times(n-1)$ order scalar matrix;
N^T : N transposed matrix; R and D: a vector containing n-1 elements array.

4.4 Point Cloud Data Segmentation

In this paper, the whole point cloud data is divided into several regions, each of which corresponds to a sub region of the solid object, using the region growing method to obtain the sub region, the point cloud data segmentation algorithm is as follows:
(1) Caculating the principal curvature and principal direction of the scattered point cloud.
(2) Identifying all boundary points: the state of the boundary points is labeled as S=-1, the state of the non-boundary points is labeled as S=0.
(3) Querying state S = 0 of any point as the seed point, to begin a new sub region, to establish the initial cohort, the seed points are enqueued and set its state to s = 1.
(4) the regional growth method to identify the sub area is as follows: Out team a point and query the status of all the adjacent points S=0; While (the presence of the neighboring points of the state S=0) All the adjacent points {enqueue state of S=0, the team after the state point set to s=1; Out of the team a point and the status of the s=0 query all the adjacent points;}.
(5) Querying state S=0 of any point as the next seed point, beginning the next sub region recognition, followed by repeated. If there is no seed point to exist, it indicates that all the sub area identification is completed.

4.5 Smoothly Splicing B-spline Surface Patches with PSO

Two B-spline surface patches with G0 continuity are stitched in four steps:
Step 1. Initialize the particles with random positions and velocity vectors. Initially, the PSO is composed of a swarm of pn possible particles. generated in the search space. A particle at its t-th iteration consists of a position vector and a velocity vector defined as follows:

$$X_{ij}(t) = \left\{ x_{ij}, i, j = 0, ..., 3(n-1) \right\}$$
$$V_{ij}(t) = \left\{ v_{ij}, i, j = 0, ..., 3(n-1) \right\} \quad \textbf{(14)}$$

Step 2. Calculate the best fitness. Each particle in the swarm has a personal fitness value. Because the curves which were defined on $[u_i\ ,\ ...,u_{i+k+1}]$ and $[v_j\ ,\ ...,v_{j+k+1}]$ are affected heavily by the corner control point d_c^l, Update the fitness of each particle pbest and the global best fitness gbest, as follows:

$$p_{ij}(t)=\min\{f_{p,t},t=0,1\ldots gen\}\ ,\ p_{gj}(t)=\min\{p_{ij}(t),t=0,1,\ldots n\}\ . \qquad \textbf{(15)}$$

Step 3. Update the velocity vector and position for every particle. Every particle will modify its velocity and position using the following formula:

The location of the ith partical is denoted as $x_i(t)$, the speed velocity is denoted as $v_i(t)$, The best position (the current local optimal solution) is denoted as $p_i(t)$, The best position (the current global optimal solution) of the whole particle swarm is denoted as $p_g(t)$.

$$\begin{array}{l} v_{ij}(t+1)=\omega\cdot v_{ij}(t)+c_1\cdot r_{1j}(t)(p_{ij}(t)-x_{ij}(t))+ \\ c_2 r_{2j}(t)(p_{gj}(t)-x_{ij}(t)) \end{array}\ ,\ x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t)\ . \qquad \textbf{(16)}$$

Where ω is the positive inertial weight, $\omega\in(0,1)$. c_1,c_2 is acceleration constant. $c_1,c_2\in[0,2]$. $r_{1j}(t),r_{2j}(t)\sim U(0,1)$. In the study, the values of is 0.5.

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

5 Experimental Results

In this section we report our experimental results. Some implementation issues are also briefly outlined. Finally, a comparison with some previous approaches and the issues of computation times and choice of GA parameters are discussed.

5.1 Implementation Issues

All computations in this paper have been performed on a 1.7 GHz. Intel Core 2 Duo processor with 1.88GB of RAM. The source code has been implemented by the authors in the native programming language of the popular scientific program Visual C++ 6.0.

5.2 Illustrative Examples

In this section, several examples from different families of functions. is tested in our algorithm. We consider six of them: a B-spline surface, shell surface, torus, vase, trim-Star, fandisk.

Example 1: A B-spline surface. We consider a set of 8321 data points from a (4,4) order B-spline surface with 8×6 control points. Mean errors for data points coordinates are 2.68×10^{-6}, 8.39×10^{-6} and 6.52×10^{-6}, respectively, showing that the reconstructed surface matches original surface very well. While maximum errors in data points are 2.25×10^{-5}, 5.39×10^{-5} and 4.57×10^{-5}

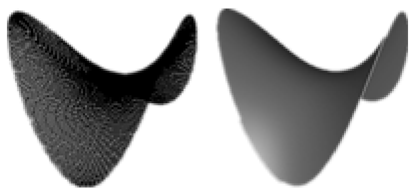

Fig. 1. B-spline point cloud (left) and fitting surface (right)

Example 2: shell surface. Next example is a parametric surface known as shell and given by:

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

We consider a set of 7500 data points with a B-spline surface of order varying from 2 to 9. Fig. 2 shows the fitting surface obtained by our method for a (8,8) order B-spline surface with a net of 19×19 control points. Mean errors in this example for data points coordinates are 7.964×10^{-7}, 7.19×10^{-7} and 1.47×10^{-8}, respectively.

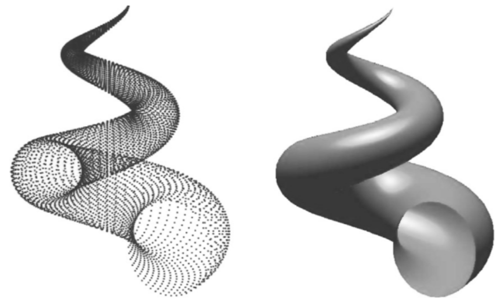

Fig. 2. Shell point cloud (left) and fitting surface (right)

Example 3: torus. Similarly to previous example, this surface is closed in one parametric direction and open in the other one, and also has a singularity point. Given a set of 4800 data points, the best fitting surface we got is a (3,4) order B-spline surface with 4×16 control points and a population of 100 individuals. Mean error for data points coordinates are respectively, 3.24×10^{-6}, 3.04×10^{-6} and 7.84×10^{-7}, while maximum errors in data points are 1.78×10^{-5}, 2.16×10^{-5} and 9.63×10^{-6}

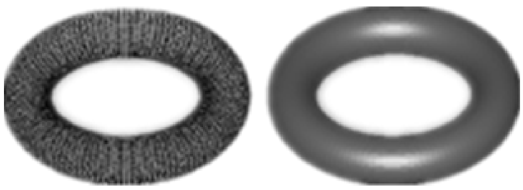

Fig. 3. Torus point cloud (left) and fitting surface (right)

Example 4: vase. This surface is expressed parametrically by:

$$\begin{cases} x = \frac{1}{2}\left[1-\cos(u)\right]\sin(u)\cos(v) \\ y = \frac{1}{2}\left[1-\cos(u)\right]\sin(u)\sin(v) \\ z = \cos(u) \end{cases}$$

For a set of 6000 data points, the best fitting surface in this case is a (5,5) order B-spline surface with 18×18control points for a population of 100 individuals, with mean errors 2.53×10^{-7}, 2.371×10^{-7} and 3.91×10^{-8} for x, y and z, respectively.

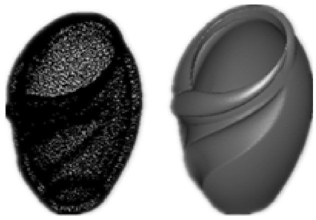

Fig. 4. Vase point cloud (left) and fitting surface (right)

Example 5: trim-star. We applied our method to a set of 5192 data points from this surface and obtained extremely accurate results. Best fitting surface for this example was a (10,10) order B-spline surface with a net of 20×20 control points. Our trials show that decreasing B-spline order or number of control points yields worse fitting errors. In surface of Fig. 5 mean errors for the data points coordinates are 3.10×10^{-15}, 3.87×10^{-15} and 5.29×10^{-15}, respectively. This example shows that the method performs extremely well even for non-zero genus surfaces, which are traditionally troublesome for many surface reconstruction methods.

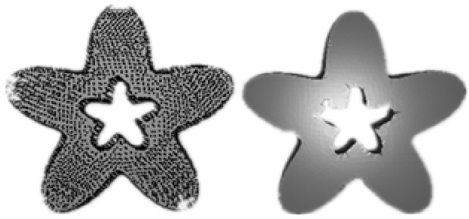

Fig. 5. Trim-star point cloud (left) and fitting surface (right)

Example 6: fandisk. In this case, the best fitting surface is a set of 16728 data points Mean error for data points coordinates is 1.45×10^{-10}, 1.57×10^{-10} and 3.66×10^{-11}, respectively. We remark the excellent results obtained by the method in this example and previous one; to the best of our knowledge, they improve by far those found in the literature on the subject.

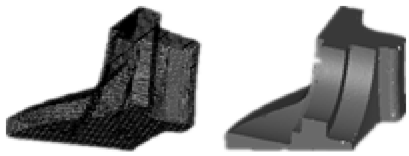

Fig. 6. Fandisk point cloud (left) and fitting surface (right)

In all, we could reconstruct the surface very well although, depending on our choice of surface parameters(order and number of control points). We also tested the robustness of our approach against noise, by perturbing some sampled data points by a real uniform random variable of mean 0 and variance $0.05\times\delta$, where δ is a parameter accounting for surface size. Our trials showed that the method is robust against low-intensity noise.

5.3 Comparison with Other Approaches

As shown earlier, our fusion method performs very well for the problems analyzed above and others not included here to keep the paper at reasonable length. Compared with other surface reconstruction approaches found in the literature, our method outperforms them in terms of accuracy and flexibility.

Table 1 summarizes our main results. Compared methods are arranged in rows and sorted so that evolutionary-based methods for polygonal meshes [29-30] and piecewise surfaces [31] follow; finally, our method is reported. For each method, the following issues are discussed: technique employed in the method, number of data points in examples reported in the corresponding entries, population size (when applicable), output of the method and number of parameters involved, runtime, reported error, behavior of the method with respect to our examples 1-6.

Table 1. Comparison of the proposed method with othe alternative surface reconstruction methods.

Authours	Method	Data points	Population	Output	Runtime	Error	Examples 1-6
Evolutionary approach Goinski [24]	Evolutionary algorithms	Not reported	30 (sphere) 30 (fractal) 50 (head)	Polygonal mesh (triangulation)	Tens of minutes to hours	Not reported	ex.1-4 works properly EX.5,6 fail
Wagner et al.[26]	Multi-objective evolutionary and genetic algorithm	823-17307	20	Polynomial B-spline	Tens of hours to days	Data points: 10^{-2}	Only ex.1 works properly ex.2-6 fail
Weinert et al. [27]	Evolutionary search and Genetic programming	192	500	PolynomialB-spline	24 h	Data points: 10^{-2}	Only ex.1 works properly ex.2-6 fail
Sarfraz [25]	Simulated evolution algorithm	441-1024	Not applicable here	Polynomial B-spline	Minutes	Data points: 10^{-1}	Only ex.1 works properly ex.2-6 fail
Our method	Fusion algorithm	5000-19728	50-500	Polynomial B-spline	Tens of seconds To minutes	Data points: 10^{-8} -10^{-15}	All work properly

Lecture [24] is hard to be compared, since its output is a polygonal mesh instead of a real mathematical surface and neither the number of data points nor the errors are reported.

A very recent approach is given in [26], where a multi-objective evolutionary algorithm approach is applied to reconstruct a siimple smooth surface with different sets of data points.

The method by Weinert, Shamsuddin and Samian [27] combines NURBS with constructive solid geometry in a hybrid evolutionary algorithm/genetic programming approach. In practice, however, the method is much simpler and less powerful than ours since no parameterization of data points is actually computed.

In [32] an evolutionary heuristic technique known as simulated evolution is applied to curve and surface fitting problems using NURBS. This method is very limited in nature, since instead of computing the knot vectors and the parameterization of data points, their values are assumed a priori. Similarly, it is assumed that the number of control points is equal to the order of the NURBS, then control points are calculated by least-squares method directly. Moreover, this method fails for non-height-map surfaces so examples 2-6 of this paper cannot be reconstructed.

To summarize, no other surface reconstruction approach described in the literature reported fitting errors as low as those achieved in this work. Besides, the examples anlayzed here are, by far, much more complex and challenging that those found in previous papers. Our method exhibits a remarkable flexibility, being able to adapt to a wide range of situations.

6 Conclusions and Future work

In this paper, the surface fitting of complex surfaces is studied, and a new algorithm based on immune genetic algorithm and Particle Swarm Optimization is proposed, The genetic immune algorithm is introduced into the Particle Swarm Optimization algorithm, and the initial population is formed by the solution group, Taking the current global optimal solution as a vaccine, the initial population selection, crossover, adaptive vaccine and so on. Through multiple iterations, the optimal solution and the current global optimal solution are compared. Experimental results show that the method has good convergence speed, and the surface fitting is less time-consuming. Through this algorithm, the fitting accuracy is higher. In all examples our fitting errors are astonishingly small when compared to any other previous approach.

In spite of these excellent results, there is still room for further improvement. Future work includes the consideration of NURBS surfaces as fitting surfaces of our method. It is also an open question how to optimize the process in order to reduce the computational load of this approach. On the other hand, our scheme is very general and can therefore be applied to a wide range of practical problems. We expect to apply it to some real-world settings described in the literature. This task will give us a very valuable feedback towards the potential improvement of our current approach.

Acknowledgements

This paper is supported by National Nature Science Foundation of China (51065021).

References

[1] H. Lin, W. Chen, H. Bao, Adaptive patch-based mesh fitting for reverse engineering, Computer-Aided Design 39(2007) 1134-1142.

[2] M. Eck, H. Hoppe, Automatic reconstruction of B-spline surfaces of arbitrary topological type, in: Proc. ACM SIGGRAPH 1996, 1996.

[3] 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.

[4] J.J.V. Wijk, Bicubic patches for approximating non-rectangular control-point meshes, Computer Aided Geometric Design 3 (1986) 1-13.

[5] S. Gofuku, S. Tamura, T. Maekawa, 2009 point-tangent / point-normal b-spline curve interpolation by geometric algorithms, Computer-Aided Design 41(2009) 412-22

[6] A. Abbas, A. Nasri, T. Maekawa, Generatiog B-spline curves with points, normals and curvature constraints: a constructive appoach, The Visual Computer 26(6-8)(2010) 823-829.

[7] H. Parka, J.H. Leeb, B-spline curve fitting based on adaptive curve refinement using dominant points, Computer-Aided Design 39(2007) 439-451.

[8] F. Yoshimotoa, T. Haradab, Y. Yoshimotoc, Data fitting with a spline using a real-coded genetic algorithm, Computer-Aided Design 35(2003) 751-760.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Longman, Boston, MA, 1989.

[14] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Machine Learning 3(1988) 95-99.

[15] Y.C. Hu, Analytic network process for pattern classification problems using genetic algorithms, Information Sciences 180(13)(2010) 2528-2539.

[16] E. Ulker, A. Arslan, Automatic knot adjustment using an artificial immune system for B-spline curve approximation, Information Sciences 179(2009)1483-1494.

[17] 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.

[18] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proc. the Sixth International Symposium on Micro Machine and Human Science, 1995.

[19] J. Kennedy, Particle swarm optimization, in: Proc. IEEE International Conference on Neural Networks, 1995.

[20] 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.

[21] H. Wang, H. Sun, S.C. Li, Rahnamayan, J. Pan, Diversity enhanced particle swarm optimization with neighborhood search, Information Sciences 223(2013) 119-135.

[22] 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.

[23] Y. Shi, H. Liu, L. Gao, G. Zhang, Cellular particle swarm optimization, Information Sciences 181(2011) 4460-4493.

[24] A. Goinski, Evolutionary surface reconstruction, in: Proc. IEEE Conference on Human System Interactions, 2008.

[25] M. Sarfraz, Computer-aided reverse engineering using simulated evolution on NURBS, Virtual and Physical Prototyping 1(4)(2006) 243-257.

[26] T. Wagner, T. Michelitsch, A. Sacharow, On the design of optimizers for surface reconstruction, in: Proc. the 2007 Genetic and Evolutionary Computation Conference-GECCO2007, 2007.

[27] K. Weinert, S.M.H. Shamsuddin, Y. Samian, Evolutionary surface reconstruction using CSG-NURBS-hybrids, in: Proc. the 2001 Genetic and Evolutionary Computation Conference-GECCO2001, 2001.

[28] L. Jiao, L. Wang, A novel genetic algorithm based on immunity, IEEE Transactions on System, Man and Cybernetics 30(5)(2000) 552-561

[29] R. Brits, A.P. Engelbrecht, F. van den Bergh, A Niching Particle Swarm Optimizer, in: Proc. 4th Asia-Pacific Conference on Simulated Evolution and Learning, 2002.

[30] G. Ciuprina, D. Inan, I. Munteanu, Use of intelligent-particle swarm optimization in electromagnetic, IEEE Trans on Magnetics 38(2)(2002) 1037-1040.

[31] Y. Shi, R.C. Eberhart, Fuzzy adaptive particle swarm optimization, in: Proc. Congress on Evolutionary Computation, 2001.