



Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation

Zhao Xiuyang^{a,b,*}, Zhang Caiming^b, Yang Bo^a, Li Pingping^a

^a School of Information Science and Engineering, University of Jinan, Jinan 250022, Shandong, PR China

^b School of Computer Science and Technology, Shandong University, Jinan 250101, Shandong, PR China

ARTICLE INFO

Article history:

Received 1 August 2010

Accepted 27 January 2011

Keywords:

B-spline curve approximation

Knot placement

GMM

EDA

Contour

ABSTRACT

One of the key problems in using B-splines successfully to approximate an object contour is to determine good knots. In this paper, the knots of a parametric B-spline curve were treated as variables, and the initial location of every knot was generated using the Monte Carlo method in its solution domain. The best km knot vectors among the initial candidates were searched according to the fitness. Based on the initial parameters estimated by an improved k -means algorithm, the Gaussian Mixture Model (GMM) for every knot was built according to the best km knot vectors. Then, the new generation of the population was generated according to the Gaussian mixture probabilistic models. An iterative procedure repeating these steps was carried out until a termination criterion was met. The GMM-based continuous optimization algorithm could determine the appropriate location of knots automatically. A set of experiments was then implemented to evaluate the performance of the new algorithm. The results show that the proposed method achieves better approximation accuracy than methods based on artificial immune system, genetic algorithm or squared distance minimization (SDM).

Crown Copyright © 2011 Published by Elsevier Ltd. All rights reserved.

The contours of images from the medical and material fields are irregularly shaped and need to be fit with smooth curves because the raw edge points of these image contours are often unorganized, non-uniformly distributed, and contain noise [1]. The snake model is a very good model for extracting object boundaries accurately from images [2,3]. In the snake model, the control points defined manually evolve in time according to internal forces determined by the shape of the curve and to image forces obtained by local intensity statistics of the image. In the case in which image forces are lacking, however, such as a point cloud acquired using a digital measuring device or a set of data points acquired by other image edge detection algorithms such as Canny operator, the snake model is not a reasonable choice.

Since the image contour shape can be complicated, it can be difficult to approximate it by a single polynomial. In this case, a B-spline curve is a good choice. B-spline forms have generally become a standard in computer-aided design, primarily because they have been widely adopted and readily available. As a result, there are many works in the literature on solving the data-fitting problem using a B-spline curve in the past twenty years [4–7]. To fit a point set with a B-spline curve, the knots and the parameters

corresponding to the data points are often given, and then least squares fitting of the data is applied. The control points of the curve are then obtained by solving a system of linear equations [8]. Thus, image contour approximation can require three steps: parameterization, knot placement, and least squares minimization.

There are several kinds of methods to define the parameters of data points, including the equally spaced, chord length and centripetal methods. The choice of knots has a considerable effect on the shape of a curve, and an unreasonable knot vector may result in an unpredictable and unacceptable shape [9,10]. Therefore, as described previously [11], the key to using B-spline successfully to approximate an image contour is to determine good knots, which requires the knots' locations to be determined as precisely as possible. In curve approximation, the number and distribution of the knots are both unknown *a priori*. Thus, knot placement can be treated as a multivariate and multimodal nonlinear optimization problem and can be solved by an optimization algorithm [10,12].

To solve the nonlinear optimization problem, many methods have been proposed [12,11,13,14]. Yoshimoto [12] proposed a method that converts the original problem into a discrete combinatorial optimization problem by taking candidates of the locations of knots as genes and solves the problem using a Genetic Algorithm (GA). Sarfraz [11] extended this work from open curves to closed curves in which the font and symbol were studied as cases. However, GA has two main drawbacks: the lack of local search ability and premature convergence. Therefore, it is difficult for this

* Corresponding author at: School of Information Science and Engineering, University of Jinan, Jinan 250022, Shandong, PR China. Tel.: +86 531 89736863.

E-mail address: zhaoxy@ujn.edu.cn (X. Zhao).

approximation method to find global optima. To avoid the drawbacks of GA, Ulker [13] suggested a difference method, the artificial immune system (AIS) [15], to solve the nonlinear optimization problem by taking the candidates of the locations of knots as antibodies. Because of the advantages in the memory education and production system, the AIS-based method obtained a better solution. Furthermore, Wang et al. [16] proposed a curvature-based squared distance minimization method for computing a planar B-spline curve to approximate a target shape defined by a point cloud. SDM is perfect for approximating unorganized, noisy data points, but the fixation of the knots of the B-spline restricted the accuracy of the approximation.

Estimation of distribution algorithms (EDAs) [17] are a class of novel stochastic optimization algorithms that have been attracted much attention in the genetic and evolutionary computation community. EDAs directly extract the global statistical information about the search space from the previous search and build a probability model of promising solutions. New solutions are then sampled from the model. EDAs have been shown to avoid the disruptive effects of genetic operators in evolutionary algorithms (EA), namely crossover and mutation, by iteratively evolving a probabilistic model to explore the search space. In this study, a new EDA using Gaussian mixture distributions by incorporating clustering techniques was applied to automatically determine the locations of knots to fit a given point set or a point cloud.

The rest of the paper is organized as follows. In Section 1, the B-spline curve-fitting problem is explained. In Section 2, EDAs and GMM are briefly introduced. Section 3 describes the proposed automatic knot placement method. The experimental results are reported in Section 4. Section 5 concludes the paper.

1. B-spline curve approximation

The equation of B-spline curve $P(t)$ of order k defined by $n + 1$ control points P_i is given by

$$P(t) = \sum_{i=0}^n P_i N_{i,k}(t), \quad t_{\min} \leq t \leq t_{\max} \quad (1)$$

where $N_{i,k}$ is the B-spline function of order k defined on a knot vector T . The knot vector T consists of non-decreasing real-valued knots. The control points form the vertices of the control polygon.

The recursive relation of the B-spline basis is defined as:

$$N_{i,k}(t) = (t - t_i) \frac{N_{i,k-1}(t)}{t_{i+k-1} - t_i} + (t_{i+k} - t) \frac{N_{i+1,k-1}(t)}{t_{i+k} - t_{i+1}} \quad (2)$$

where

$$N_{i,1} = \begin{cases} 1, & t_i \leq t \leq t_{i+1} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

In this paper, we consider the following problem: let $Q_i (i = 0, 1, \dots, m)$ be the set of points describing the image edge and then compute a B-spline curve to fit these points. It is possible to construct an optimal B-spline curve from the given points that will pass as close as possible to the given set of points. The distance is measured at the parameter values $\tilde{U} = (\tilde{u}_0, \dots, \tilde{u}_m)$. The closest spline is defined by:

$$\min_{T, \tilde{U}} \|Q(\tilde{u}_i) - Q_i\|^2. \quad (4)$$

The problem has several degrees of freedom (d.o.f.); changing the knot vector and the parameter value will generate a different curve [18]. The problem studied here is therefore how to exploit these d.o.f. to optimize a certain fairing goal to reduce the approximation error of the curve.

2. Estimation of distribution algorithms

EDAs are an outgrowth of genetic algorithms and are able to estimate the probability distribution using the selected set of

solutions itself and then employing this estimate to generate new solutions. In an EDA, a population of candidate solutions to a problem is maintained as part of the search for an optimal solution, just like in a genetic algorithm. However, the population is represented by a probability distribution that can prevent disruption of partial solutions contained in a chromosome by giving them high probability, and new candidate solutions can be obtained by sampling this distribution. This technique may have several advantages, including avoiding premature convergence and providing a more compact representation.

As described in the literature [19], EDAs replace genetic recombination and mutation operators by the following two steps:

1. A model (an estimation of the true distribution) of selected promising solutions is constructed.
2. New solutions are generated according to the constructed model.

Early EDAs for continuous optimization presumed that the selected vector was a random sample from a single Gaussian distribution [20]. However, single Gaussian distribution may become trapped in local optima when dealing with multimodal problems. To overcome the shortcomings of the single Gaussian distribution, researchers constructed EDAs using Gaussian mixture distributions by incorporating clustering techniques [20,21].

2.1. Gaussian mixture model

A Gaussian mixture model is a powerful model for multivariate density estimation. In this paper, this model is used to describe the probability distribution of all of the knots. For this approach, data $X = \{x_1, x_2, \dots, x_N\}$ were assumed to arise from a random vector with density:

$$P(x_i | \Theta) = \sum_{k=1}^K \pi_k p_k \left(x_i | \mu_k, \Sigma_k \right) \quad (5)$$

where π_k is the mixing proportion, $0 < \pi_k < 1$, and $\sum_{k=1}^K \pi_k = 1$, $\Theta(\theta_1, \theta_2, \dots, \theta_K)$ is a vector containing parameters π_i, μ_i, Σ_i . In addition, $p_k(x_i | \mu_k, \Sigma_k)$ denotes the k th component of the Gaussian distribution with mean vector μ_k and variance matrix Σ_k . The fit of the model to the data can then be measured by the total log likelihood of the data:

$$L(\Theta | x_1, \dots, x_n) = \sum_{i=1}^n \ln \left[\sum_{k=1}^K \pi_k p_k \left(x_i | \mu_k, \Sigma_k \right) \right]. \quad (6)$$

Expectation maximization (EM) algorithms are a general method for finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data is incomplete. Therefore, an EM can be used to find an estimate of the parameter π_i, μ_i, Σ_i in Eq. (6). The iterations of the EM algorithm consist of an E step and an M step. In the E step, the current conditional probabilities $\hat{p}_k(x_i)$ that x_i arises from the k th mixture component are computed. In the M step, the maximum-likelihood estimates $\hat{p}_k, \hat{\mu}_k, \hat{\Sigma}_k$ are updated using the conditional probabilities $\hat{p}_k(x_i)$ as conditional mixing weights. These two steps are repeated as necessary. Each iteration is guaranteed to increase the log likelihood, and the algorithm is guaranteed to converge to a local maximum for the likelihood function.

In this paper, the Gaussian mixture model was used to describe the probability distributions of all of the knots. It is well known that optimization algorithms converge to different local solutions depending on the initialization. Therefore, an improved cluster method was used to guarantee a reasonable initialization.

2.2. The improved k -means algorithm

We initialized the GMM using an improved k -means algorithm. In k -means clustering, each cluster is specified by its center

vector. If unreasonable initial positions are given, the classical k -means algorithm cannot guarantee correct clustering results. In this paper, to overcome the shortcomings of the classical k -means algorithm, enlightened by Ref. [22], we developed a new initial positions decision method based on data mass. The principles of data mass are as follows:

- (1) The larger the number of data points around a given point, the more mass the given point has.
- (2) The nearer a data point to a given point, the more it contributes to the mass of the given point. The mass of a given point can be computed as

$$M = \sum_{i=1}^n (R - |X_i - C|)^2 \quad (7)$$

where M is the mass of the given point, R is the radius of a circle around the given point, n is the number of points included in the circle, X_i is the coordinates of the points included in the circle and C is the coordinates of the given point. $|\cdot|$ is the Euclidean distance.

Based on the principles of data mass, we effectively improved the classical k -means algorithm. The improved clustering process can be described as follows.

- (1) Calculate the data mass for every data point according to Eq. (7), generating k centers according to the data mass.
- (2) Assign each individual to the nearest center.
- (3) Update each center to the mean vector of all the individuals assigned to it in the previous step.
- (4) Repeat the updating of the k centers until the new centers are equal to the previous ones.

3. Automatic knot adjustment by GMM-based EDA

The B-spline curve-fitting problem is to produce a B-spline curve to approximate a target curve within a prespecified tolerance. After edge tracing and sampling, the image contour is defined in a 2D plane by a sequence of ordered dense data points. The first and the last points are superposed because the contour is a closed curve, thus an additional point occurs: $q_{m+1} = q_0$. In this paper, we adopt accumulated chord length to parameterize the data points, and we used GMM-based EDA to optimize the knot vector based on a basic knot vector.

3.1. Data point parameterization

There are several methods for parameterizing data points, including arc length parameterization, accumulated chord length parameterization and centripetal chord length parameterization. Herein, we adopt the accumulated chord length to parameterize the data points. Assume that the parameter according to the point q_i is \tilde{u}_i . There are then $m + 2$ parameters for $m + 2$ data points. We can define the parameters $\{\tilde{u}_i \mid i = 0, 1, \dots, m, m + 1\}$ as follows [23]:

$$\begin{cases} \tilde{u}_0 = 0 \\ \tilde{u}_i = \tilde{u}_{i-1} + |\Delta q_{i-1}|, \quad i = 1, 2, \dots, m + 1 \end{cases}$$

where Δq_k is the forward difference and $\Delta q_k = q_{k+1} - q_k$ is the chord length. The normalization of the parameters can then be expressed as $\tilde{u}_i \leftarrow \tilde{u}_i / \tilde{u}_{m+1}$, $i = 0, 1, \dots, m + 1$.

A closed B-spline curve of order k can be defined by a linear combination of B-spline functions as follows:

$$C(u) = \sum_{i=0}^{n+k-1} N_{i,k}(u) P_{i \bmod (n+1)}, \quad 0 \leq u \leq 1, \quad (8)$$

where the $(n + 1)$ distinct control points $\{p_0, p_1, \dots, p_n\}$ build up the control polygon and the B-spline functions $P_{i \bmod (n+1)}$ of order k are defined on the knot vector $\{u_0, u_1, \dots, u_k = 0, u_{k+1}, \dots, u_n, u_{n+1}, \dots, u_{n+k+1} = 1, \dots, u_{n+2k+1}\}$.

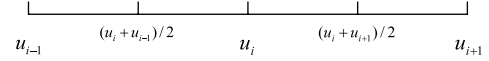


Fig. 1. The interval associated with the knot u_i .

3.2. The generation of the basic knot vector

A knot vector T should be selected according to the parameter values $\{\tilde{u}_i \mid i = 0, 1, \dots, m, m + 1\}$, the number n of control points and the number m of the given points. The knots are mostly determined to reflect the distribution of the parameter values and to guarantee that every knot span contains at least one parameter value [23]. The definition domain includes $n + 1$ knot intervals. The knots within the definition domain can be defined as [23]:

$$\begin{aligned} c &= \frac{m + 2}{n + 1}, \\ i &= \text{int}(jc), \quad \alpha = jc - i, \\ u_{k+j} &= (1 - \alpha)\tilde{u}_{i-1} + \alpha\tilde{u}_i, \quad j = 1, 2, \dots, n \end{aligned}$$

where i is the maximum integer below jc . The knots outside the definition domain can be defined as:

$$\begin{cases} u_0 = u_{n+1} - 1, u_1 = u_{n+2} - 1, \dots, u_{k-1} = u_{n+k} - 1; \\ u_{n+k+2} = 1 + u_{k+1}, u_{n+k+3} = 1 + u_{k+2}, \dots, u_{n+2k+1} = 1 + u_{2k}. \end{cases} \quad (9)$$

3.3. Initial population and fitness function

After the basic knot vector is generated, the interval $[(u_i + u_{i-1})/2, (u_i + u_{i+1})/2]$ associated with the knot u_i is regarded as the solution domain of u_i . The initial location of every interior knot can be generated using the Monte Carlo method in its solution domain, and the exterior knots can be defined by Eq. (9). For the initial population, we create kn individuals, where kn is the size of the population (Fig. 1).

For every individual, we determine the B-spline curve $C(u)$ by minimizing the least squares error defined as:

$$f = \sum_{i=0}^m |q_i - C(\tilde{u}_i)|^2 = \sum_{i=0}^m \left| q_i - \sum_{j=0}^{n+k} N_{j,k}(\tilde{u}_i) p_{j \bmod (n+1)} \right|^2. \quad (10)$$

As the fitness function, we use Eq. (11):

$$E = \sum \min |q_i - C(u)|^2 \quad 0 \leq i \leq m. \quad (11)$$

For each individual, the fitness value is calculated, and the least km individuals are selected as optimization individuals:

$$\begin{aligned} U_0 &= (u_0^3, \dots, u_0^i, \dots, u_0^{n+k+1}), \dots, U_{km} \\ &= (u_{km}^3, \dots, u_{km}^i, \dots, u_{km}^{n+k+1}). \end{aligned}$$

3.4. The EDA approach for knot optimization

We assume that the target shape is defined in a 2D plane by a sequence of ordered data points or by an extremely noisy data point set. For the ordered data curve, the feature points are sampled by a curvature estimation algorithm [24], and the points are interpolated as the initial fitting curve. For the noisy data point set, following previous work [16] we first compute a uniform spatial partition of the whole image including the data points with a proper cell size and then record those nonempty cells. Next, we use the edge trace method to trace the outer border of those nonempty cells and then sample the center of those nonempty cells as the initial points and interpolate the points as the initial fitting curve (an example can be seen in Fig. 2). The basic knot vector is computed based on the accumulated chord length method,

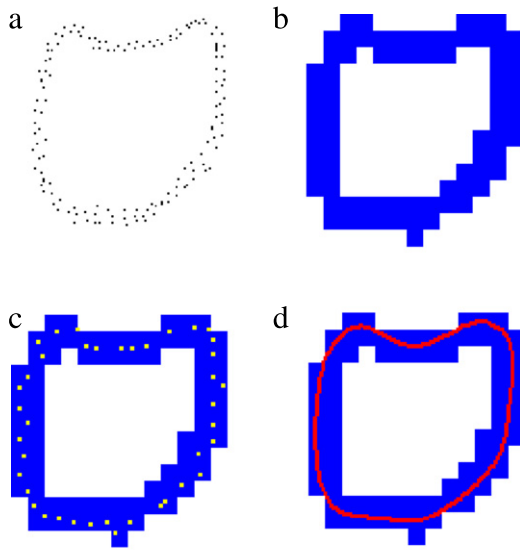


Fig. 2. The cells and the initial curves.

and individuals, that is, the different knot vectors, in the initial population can be obtained by the method described in Section 3.3. For every individual, a corresponding curve can be acquired by the least squares method, and the fitness of the curve can be calculated by Eq. (11). Based on the fitness, some good individuals can be selected and can be used to calculate the distribution probability of every knot. The new population can be generated according to the distribution probability.

The main steps of the EDA approach for knot optimization are as follows.

- (1) The first population P ($t = 0$) of kn individuals is generated. The generation of these kn individuals is achieved by a Monte Carlo method based on the basic knot vector.
- (2) A number km ($km < kn$) of individuals are selected following Eqs. (10) and (11).
- (3) For every knot, u_i , km candidate points around the initial u_i are present after selection. The data mass for all the candidate points is calculated using Eq. (7). The k best points are selected as the initial centers of the k -means algorithm according to the data mass, and then the k -means cluster method is used to cluster the km candidate points associated with the knot u_i . The Gaussian mixture model for knot u_i can be established by an expectation maximization algorithm based on the results of the k -means cluster.
- (4) According to the Gaussian mixture model of every u_i , the new generation of population consisting of new kn individuals is generated.
- (5) If the given stop criteria are not met, set $t = t + 1$ and go back to step 2.

4. Experimental results

To evaluate the proposed GMM-based automatic knot placement algorithm, four examples are used. Fig. 3 displays the initial contours of an intestine section, a cement particle, an SiC particle and a kidney section, which contain 418, 197, 323 and 294 points, respectively. The performance of our approach was then compared with that of a GA-based algorithm [12], an AIS-based algorithm proposed by Ulker [13] and the SDM algorithm [16]. For comparing the approximation error, the average error defined in Ref. [16] was converted to error as defined in Eq. (11) after every iteration.

Table 1 gives the set of parameters of the three evolutionary algorithms. In all of the following examples, we assumed that

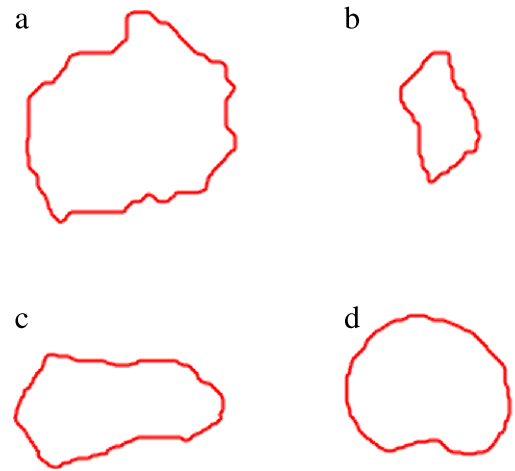


Fig. 3. The initial contours (a) of an intestine section, (b) of a cement particle, (c) of an SiC particle and (d) of a kidney section.

Table 1

The set of parameters.

Parameter	GMM	GA	AIS
Population size (kn)	300	300	300
String length	33	33	33
Crossover	None	0.6	0.6
Mutation rate	None	0.1	0.1
k	5	None	None
km	100	None	None
Memory size	None	None	30
Maximum number of iterations	200	200	200

the order of B-spline was 3, and we performed 20 runs of each algorithm, except for SDM because it was not an evolutionary algorithm. The graphs of the fitness according to generations are shown in Figs. 6, 8, 10 and 12. These graphs were generated using the performance of SDM and the average performance of the other algorithms with respect to generational distance. In these figures, the dash-dot line, the red dot line, the black dot line and the blue line are the fitness of the SDM algorithm and the average fitnesses of the GMM-based algorithm, the GA-based algorithm and the AIS-based algorithm, respectively.

The feature points were detected by an estimated curvature method [24] for the intestine section, the cement contour, the SiC particle and the kidney section contour, and the resulting 113 feature points, 54 feature points, 88 feature points and 81 feature points, respectively, are shown in Fig. 4. In Fig. 5, we show the approximated contours of the intestine section obtained using the GMM-based algorithm, the SDM algorithm, the AIS-based algorithm and the GA-based algorithm. Fig. 6 shows the fitness versus generation of the intestine section, and Fig. 7 shows the approximated contours of the cement contour obtained using the proposed method, the SDM algorithm, the AIS-based method and the GA based method. Fig. 9a and b show the approximated contours of the SiC particle based on the proposed knot placement method and the SDM algorithm, while Fig. 9c and d show the approximate contours obtained using the AIS-based algorithm and the GA-based algorithm, respectively. The average fitness for each method is also shown in Fig. 10.

The last experiment used the kidney section contour shown in Fig. 11. The initial contour contained 294 points, and the number of feature points was 81. Fig. 11a–d show the approximated contours. The average fitness versus generation of this experiment is shown in Fig. 12. The average fitness values of the four contours can be found in Table 2, where con 1, con 2, con 3 and con 4 represent the intestine section, the cement particle, the SiC particle and the

Table 2
Fitness of the four contours.

	GMM				AIS				GA			
	Con 1	Con 2	Con 3	Con 4	Con 1	Con 2	Con 3	Con 4	Con 1	Con 2	Con 3	Con 4
0	163.7341	18.1226	44.5806	27.6123	163.2452	18.4157	44.9512	27.8711	160.4832	18.5979	44.5867	28.54125
1	161.5497	17.5507	43.8591	26.9769	161.9244	17.9288	44.5939	27.6596	155.4382	17.5979	44.5768	27.98423
2	154.6729	17.3896	42.7183	26.4412	154.6821	17.0557	43.7129	26.8985	154.2406	17.1955	44.1698	27.63766
3	150.4632	16.8168	41.8176	26.2212	150.8894	16.2068	43.2977	26.5263	153.6602	17.1680	44.1698	27.63766
4	147.2566	16.6112	41.2167	26.0291	149.0742	16.8293	42.6227	26.5163	153.6602	17.1690	44.1698	27.45722
5	144.3016	16.2775	41.1327	25.6225	146.6145	16.4909	41.8005	26.5133	153.6602	17.1680	43.7252	27.29976
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
100	120.5254	13.7709	36.0616	23.2223	125.6242	14.1698	37.2331	23.7184	144.6242	16.2987	41.3774	26.2207
101	120.5254	13.7709	36.0616	23.1941	125.4609	14.1698	37.2331	23.7132	144.4609	16.2987	41.3774	26.2207
102	120.5254	13.7709	36.0616	23.1941	125.4609	14.1598	37.2331	23.7132	144.4609	16.2987	41.3774	26.2207
103	120.5254	13.7709	36.0616	23.1941	125.4609	14.1598	37.2331	23.7132	144.4609	16.2987	41.3774	26.2207
104	120.5254	13.7709	36.0616	23.1941	125.4609	14.1598	37.2331	23.7132	144.4609	16.2987	41.3774	26.0238
105	120.5254	13.7709	35.9447	23.1590	125.6242	14.1598	37.2023	23.7132	144.4609	16.2987	41.3774	26.0238
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
196	118.6533	13.6184	35.5560	22.4276	123.0125	14.0211	36.8059	23.2645	141.5611	16.1551	41.2020	25.8612
197	118.6533	13.6184	35.5560	22.4276	123.0125	14.0195	36.8059	23.2645	141.5611	16.1551	41.2020	25.8354
198	118.6533	13.6184	35.5560	22.4276	123.0125	14.0195	36.8059	23.2645	141.5611	16.1551	41.2020	25.8354
199	118.6533	13.6184	35.5560	22.3964	123.0125	14.0195	36.8059	23.2645	141.5611	16.1551	41.2020	25.8354
200	118.5977	13.6184	35.5560	22.3964	123.0125	14.0195	36.8059	23.2645	141.5611	16.1551	41.2020	25.8354

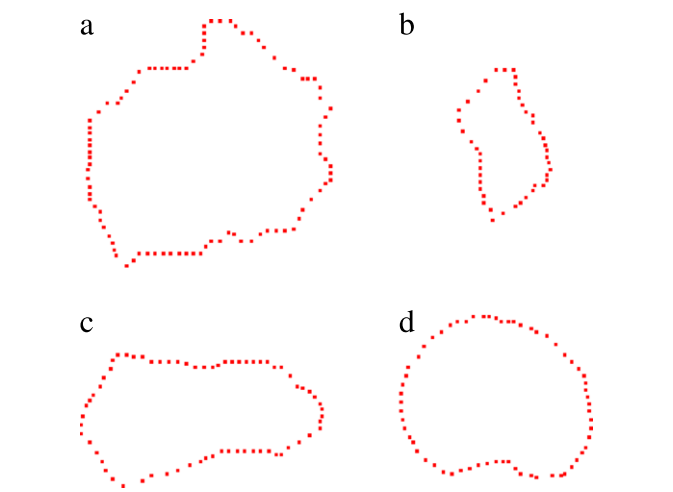


Fig. 4. The feature points (a) of the intestine section, (b) of the cement particle, (c) of the SiC particle and (d) of the kidney section.

Table 3
Fitness of the four contours of SDM.

Generation	Contours			
	Con 1	Con 2	Con 3	Con 4
1	162.5131	18.4812	44.5800	27.8455
2	157.2671	17.2612	42.1273	25.8342
3	156.5567	16.0018	40.2568	24.9365
4	153.3028	15.8044	39.9854	23.8659
5	152.4343	15.3433	37.1605	23.6658
6	133.2547	15.3609	36.465	23.5052
7	126.2143	15.2905	36.2698	23.3654
8	124.3154	14.1905	36.2698	23.1653
9	122.3154	14.0905	36.2698	22.9653
10	122.3154	14.0905	36.2698	22.9653

kidney section, respectively. The fitness values versus generation of SDM are shown in Table 3.

The proposed GMM-based automatic knot placement algorithm was also successful in fitting a point cloud that represents a closed target curve. We present below the results of applying the proposed method to fitting a cubic B-spline curve to two sets of

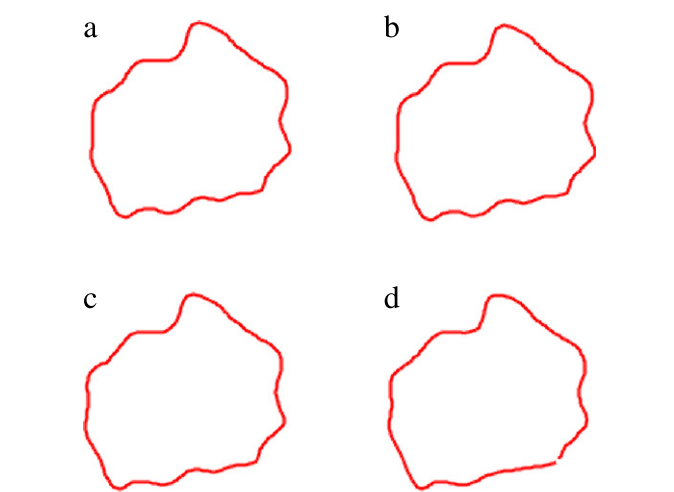


Fig. 5. The experiment for intestine section: (a) the approximated contour acquired by the proposed method, (b) the approximated contour acquired by SDM, (c) the approximated contour acquired by the AIS-based method and (d) the approximated contour acquired by the GA-based algorithm.

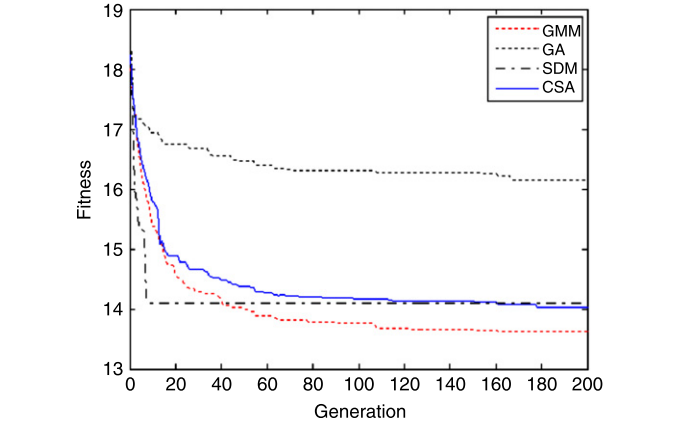


Fig. 6. Fitness versus generation for the intestine section. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

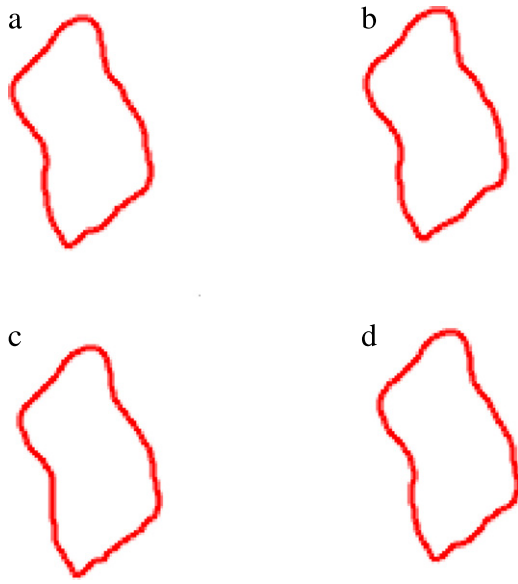


Fig. 7. The experiment for the cement particle: (a) the approximated contour obtained using the proposed method, (b) the approximated contour obtained using SDM, (c) the approximated contour obtained using the AIS-based method and (d) the approximated contour obtained using the GA-based algorithm.

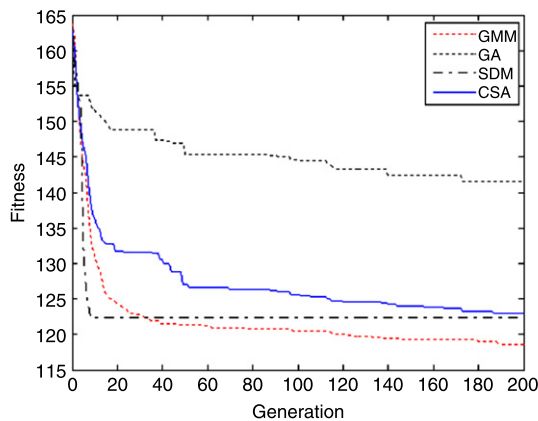


Fig. 8. Fitness versus generation for the cement particle. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

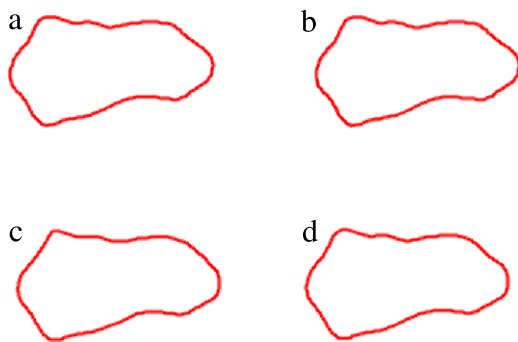


Fig. 9. The experiment for the SiC particle: (a) the approximated contour obtained using the proposed method, (b) the approximated contour obtained using SDM, (c) the approximated contour obtained using the AIS-based method and (d) the approximated contour obtained using the GA-based algorithm.

unorganized data points. Figs. 13a and 14a are the two noisy data sets that included 112 and 520 points, respectively. The initial fitting curves acquired by the method described in Section 3.4

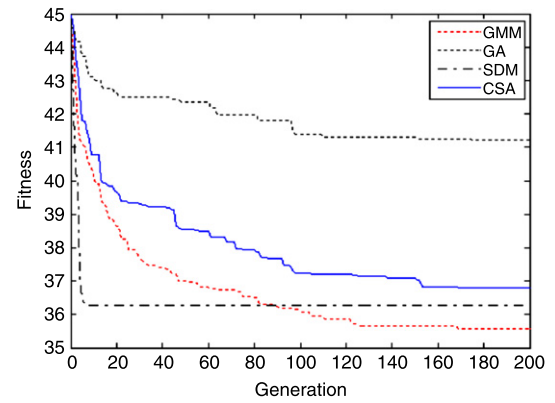


Fig. 10. Fitness versus generation for the SiC particle. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

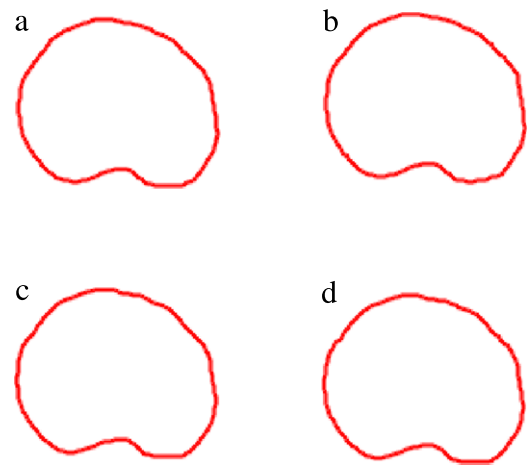


Fig. 11. The experiment for the kidney section: (a) the approximated contour acquired by the proposed method, (b) the approximated contour acquired by SDM, (c) the approximated contour acquired by the AIS-based method and (d) the approximated contour acquired by the GA-based algorithm.

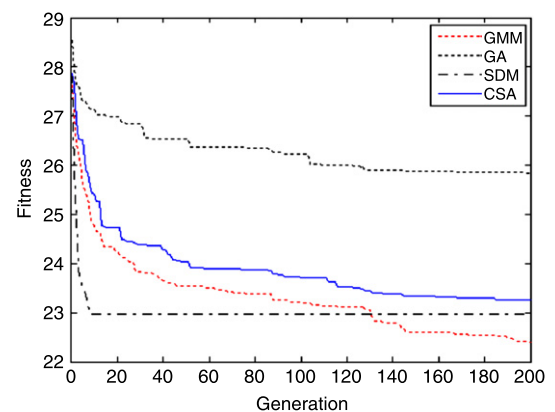


Fig. 12. Fitness versus generation for the kidney section. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

are shown in Figs. 13b and 14b, respectively. Additionally, the red curves in Figs. 13c and 14c are the fitting curves obtained using the proposed algorithm after 50 iterations, from which we could see that the algorithm converged successfully and produced acceptable results. Finally, Figs. 13d and 14d are the fitting curves and their control polygons.

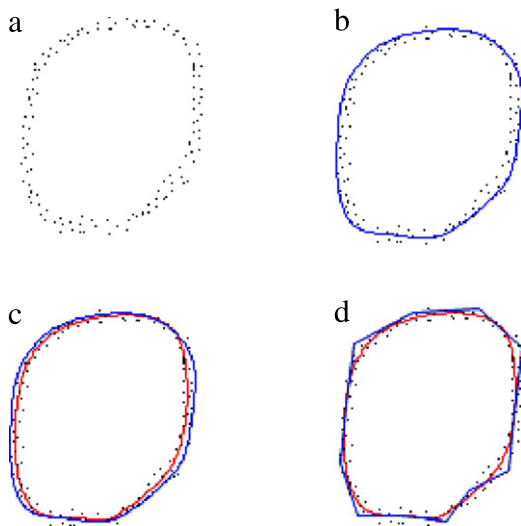


Fig. 13. Fitting a data set of 112 data points. (a) The closed target shape; (b) the initial B-spline curve; (c) the fitting curve generated by GMM and the initial curve; (d) the fitting curve and its control polygon.

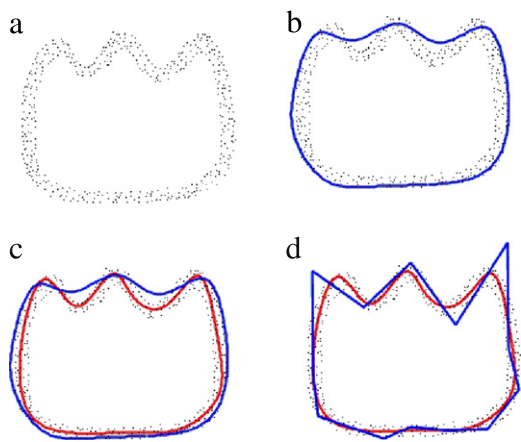


Fig. 14. Fitting a noisy data set of 520 points. (a) The closed target shape; (b) the initial B-spline curve; (c) the fitting curve generated by GMM and the initial curve; (d) the fitting curve and its control polygon.

5. Conclusions

To obtain a good B-spline curve model from a raw contour, knots are usually treated as variables. A curve is then modeled as a continuous, nonlinear and multivariate optimization problem with many local optima. Therefore, it is very difficult to reach a global optimum. To overcome this difficulty, we suggested a new method that solves this problem using a Gaussian mixture distribution based on the estimation of a distribution algorithm, and the initial parameters of the GMM were estimated by an improved k -means algorithm. The basic knot vectors were generated using a method previously reported [23], and the interval $[(u_i + u_{i-1})/2, (u_i + u_{i+1})/2]$ associated with the knot u_i was considered the solving domain of u_i . With the aid of a GMM-based EDA, good knot vectors were determined automatically.

The proposed method was also compared with a GA-based method, SDM and an AIS-based method. GA methods have global perspective and are robust, but their convergence is slower and computational cost is higher for generations much closer to the optimal solution. AIS has advantages over GA in memory education and production systems, so the AIS-based method can obtain a better solution. The proposed algorithm is an EDA whose probabilistic model is based on GMM. It acquires solutions by

statistically learning and sampling the probability distribution of the best individuals of the population at each iteration of the algorithm. In such a way, the relationships between the variables involved in the problem domain are explicitly and effectively exploited. Therefore, it is easy to find the global optimum. The experiment results also show that the proposed algorithm performs better than GA- and AIS-based algorithms.

Acknowledgements

This research was supported by the National Natural Science Foundation of China under contract (Nos. 60873089 and 60933008), the Doctor Foundation of Shandong Province under Grant (Nos. 2007BS04018 and ZR2010FM047), the Postdoctoral Innovation Fund of Shandong Province under Grant (No. 200802026) and the Technology Plan Project of the Department of Education of Shandong Province under Grant (No. J07YJ23).

References

- [1] Yang ZW, Deng JS, Chen FL. Fitting unorganized point clouds with active implicit B-spline curves. *The Visual Computer* 2005;21:831–9.
- [2] Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. *International Journal of Computer Vision* 1988;1(4):321–31.
- [3] Meegama RGN, Rajapakse JC. NURBS snakes. *Image and Vision Computing* 2003;21:551–62.
- [4] Park H. An error-bounded approximate method for representing planar curves in B-splines. *Computer Aided Geometric Design* 2004;21:479–97.
- [5] Kouibia A, Pasadas M. Approximation by smoothing variational vector splines for noisy data. *Journal of Computational and Applied Mathematics* 2008;211: 213–22.
- [6] Park H, Lee JH. B-spline curve fitting based on adaptive curve refinement using dominant points. *Computer-Aided Design* 2007;39:439–51.
- [7] Yang HP, Wang WP, Sun JG. Control point adjustment for B-spline curve approximation. *Computer-Aided Design* 2004;36:639–52.
- [8] Jing L, Sun L. Fitting B-spline curves by least squares support vector machines. In: *Proceedings of the second international conference on neural networks and brain*. Beijing (China): IEEE Press; 2005. p. 905–9.
- [9] Farin G. *Curves and surfaces for CAGD*. 5th ed. San Francisco: Morgan Kaufmann; 2002.
- [10] Li WS, Xu SH, Zhao G, et al. Adaptive knot placement in B-spline curve approximation. *Computer-Aided Design* 2005;37(7):791–7.
- [11] Sarfraz M, Raza SA. Capturing outline of fonts using genetic algorithm and splines. In: *Fifth international conference on information visualisation*. London(UK): IEEE Computer Society Press; 2001. p. 738–43.
- [12] Yoshimoto F, Moriyama M, Harada T. Automatic knot placement by a genetic algorithm for data fitting with a spline. In: *Proceedings of the international conference on shape modeling and applications*. Aizu-Wakamatsu: IEEE Computer Society Press; 1999. p. 162–9.
- [13] Ulker E, Arslan A. Automatic knot adjustment using an artificial immune system for B-spline curve approximation. *Information Sciences* 2009;179: 1483–94.
- [14] Yoshimoto F, Harada T, Yoshimoto Y. Data fitting with a spline using a real-coded genetic algorithm. *Computer-Aided Design* 2003;35:751–60.
- [15] De Castro LN, Timmis J. *Artificial immune systems: a new computational intelligence approach*. London: Springer-Verlag; 2002.
- [16] 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.
- [17] Pelikan M, Goldberg DE, Lobo FG. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications* 2002;21:5–20.
- [18] Goldenthal R, Bercovier M. Spline curve approximation and design by optimal control over the knots. *Computing* 2004;72:53–64.
- [19] Pelikan M, Goldberg DE, Lobo F. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications* 2002;21: 5–20.
- [20] Lu Q, Yao X. Clustering and learning gaussian distribution for continuous optimization. *IEEE Transaction on Systems, Man, and Cybernetics-Part C: Application and Reviews* 2005;35(2):195–204.
- [21] Bosnian PAN, Thierens D. Expanding from discrete to continuous estimation of distribution algorithms: the idea. In: *Parallel problem solving from nature-PPSN VI*. New York: Springer-Verlag; 2000. p. 767–76.
- [22] Peng L, Yang B, Chen Y, et al. Data gravitation based classification. *Information Sciences* 2009;179(5):809–19.
- [23] Pigel L, Tiller W. *The NURBS book*. 2nd ed. Berlin: Springer-Verlag; 1997.
- [24] Wu WY. An adaptive method for detecting dominant points. *Pattern Recognition* 2003;36:2231–7.