Journal of Mathematical Research with Applications
Jan., 2020, Vol. 40, No. 1, pp. 87–100
DOI:10.3770/j.issn:2095-2651.2020.01.008
Http://jmre.dlut.edu.cn

Curve Reconstruction Algorithm Based on Discrete Data Points and Normal Vectors

Mingyang GUO, Chongjun LI*
Department of Mathematics, Dalian University of Technology, Liaoning 116024*, P. R. China*

Abstract This paper presents a curve reconstruction algorithm based on discrete data points and normal vectors using B-splines. The proposed algorithm has been improved in three steps: parameterization of the discrete data points with tangent vectors, the B-spline knot vector determination by the selected dominant points based on normal vectors, and the determination of the weight to balancing the two errors of the data points and normal vectors in fitting model. Therefore, we transform the B-spline fitting problem into three sub-problems, and can obtain the B-spline curve adaptively. Compared with the usual fitting method which is based on dominant points selected only by data points, the B-spline curves reconstructed by our approach can retain better geometric shape of the original curves when the given data set contains high strength noises.

Keywords curve reconstruction; curve fitting; normal vector; B-spline; dominant point

MR(2010) Subject Classification 65D07; 65D10

1. Introduction

The B-spline curve reconstruction is one of the classic problem in the field of computer-aided geometric design and it is an indispensable link in many applications [1, 2]. For example, large amounts of data generated in engineering such as reverse engineering, terrain modeling, and aerospace design must be approximated by smooth B-spline curves [3–5].

In previous studies, raw data usually only contains position coordinates of the discrete points [6–8]. Nowadays, in many applications, the measurement data includes not only the position coordinates, but also geometric features such as normal vectors and curvature [9–16]. In many practical problems, curves need to be constructed satisfying certain flow field of dynamic constraints. One key problem can be transformed into the curve reconstruction based on discrete data points and normal vectors. The main difficulty of this kind of problem is that the fitting error of discrete data points and the error of normal vector constraints should be taken into account at the same time. Moreover, the problem is an ill-posed problem, which means that the result of curve reconstruction is greatly affected by data noises.

The rest of the paper is organized as follows. The second section will briefly introduce the B-spline curve reconstruction algorithm using dominant points. The third section will introduce the

Received June 17, 2019; Accepted October 26, 2019
Supported by the National Natural Science Foundation of China (Nos. 11871137; 11572081) and the Program for Liaoning Innovation Talents in University (No. LCR2018001).
* Corresponding author
E-mail address: chongjun@dlut.edu.cn (Chongjun LI)

curve reconstruction algorithm based on discrete data points and normal vectors. The numerical results in the fourth section will show the practicability of the proposed method.

2. The B-spline curve fitting approach using dominant points

First, we review the basic definition of the B-spline curve. A p-th degree ($p+1$ order) non-uniform B-spline curve can be expressed as [17]:

$$S(u) = \sum_{i=0}^{n} Q_i B_{i,p}(u), \tag{2.1}$$

with control points $\{Q_i : i = 0, \dots, n\}$ and the p-th degee non-uniform B-spline basis functions $\{B_{i,p}(u) : i = 0, \dots, n\}$, which are determined by the corresponding knot vector $U = [u_0, u_1, \dots, u_{n+p}]$.

In previous studies, curve reconstruction algorithms are usually composed of three steps: discrete data points parameterization, knot vector determination and curve fitting approach. For the discrete data points parameterizaion, three methods have been widely used: uniform parameterization, accumulated chord length parameterization and centripetal parametrization [18]. For the knot vector determination, we briefly introduce the Dominant Points Method (DOM) [19], which solves the problem well and provides some ideas for our method. DOM is an iterative method and mainly divided into two steps: dominant points selection and knot vector placement.

Suppose the dominant points $\{P_{d_j} : j = 0, \dots, n\}$ have already been chosen, then the internal knots can be determined by the following formula:

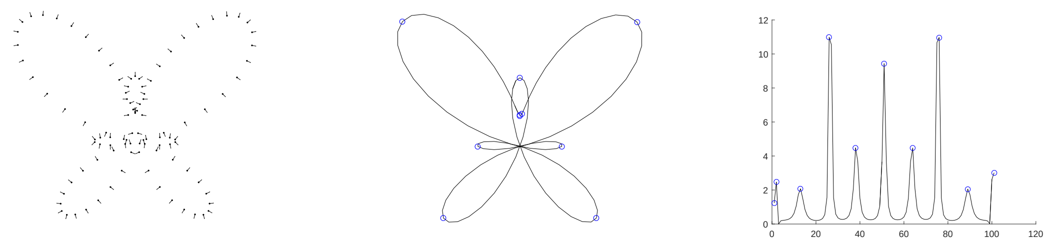
$$u_{p+i} = \frac{1}{p-1} \sum_{j=i}^{i+p-1} t_{d_j}, \quad i = 1, \dots, n-p, \tag{2.2}$$

where t_i is the parameter value of the discrete point P_i, $d_j = i$ is the original index of the point P_i if it is the j-th dominant point P_{d_j}.

Note that, according to the Schoenberg-Whitney theorem, this approach can give a non-singular stable system matrix [20]. More importantly, this method only averages the selected dominant points so that no bad results will be obtained even if the number of control points is close to the given data points.

Subsequently, we introduce the selection method of the dominant points. DOM first uses curvature to select seed points from the point set, then add new dominant point iteratively. For the seed points, DOM chooses two boundary points and other points with local curvature maximum (LCM). There are two approaches for calculating curvature. When the noise intensity is small, the local method is used to calculate the corresponding curvature value [21, 22]. When the noise is big, the curve fitting method is used [23]. Although the curve fitting method is able to overcome the influence by the noise to a certain extent, it also spends a lot of time.

Here is an example for selecting seed points using DOM. Figure 1 (a) shows the 101 points which have been sampled from an original butterfly curve, Figure 1(b) shows the seed points have been chosen and Figure 1 (c) shows the curvature variation of the whole curve. Obviously, the points corresponding to the peak of the curvature variation are chosen to be the seed points.



(a) (b) (c)

Figure 1 Seed points and their corresponding curvature

When the seed points have been selected, a B-spline curve $S(u)$ can be generated by knot vector determination and the least-square minimization. Then the point P_c can be found, which contains the largest deviation: $\|S(t_i) - P_i\|$. DOM then divide the point set into several segments according to the adjacent dominant points and using the shape index to find the new dominant point P_w in segment $S_{s,e}$ that contains P_c, the formula is expressed as follows [19]:

$$\min_{w} |\lambda_{s,w} - \lambda_{w,e}|, \quad w \in (s, e),$$

where

$$\lambda_{s,e} = r\frac{K_{s,e}}{K_{0,m}} + (1-r)\frac{L_{s,e}}{L_{0,m}}, \quad r \in [0, 1], \tag{2.3}$$

$K_{s,e}$ and $L_{s,e}$ denote the total curvature and the chord length of the segment $S_{s,e}$, defined as:

$$K_{s,e} = \sum_{i=s}^{e-1} (|k_i| + |k_{i+1}|)(t_{i+1} - t_i)/2,$$

$$L_{s,e} = \sum_{i=s}^{e-1} \|P_{i+1} - P_i\|. \tag{2.4}$$

Note that, if segment $S_{s,e}$ contains less than three points, then remove $S_{s,e}$ and find P_c in rest segments. After the new dominant point has been selected, the iterative process continues as described above.

3. Curve reconstruction algorithm based on discrete data points and normal vectors

As for curve fitting method, the least squares method is the most widely used method. Combining the normal vectors and discrete data points, we now propose a new method for B-spline curve reconstruction, which is denoted as NDOM (nromal based DOM) in this paper.

3.1. Discrete point parameterization using normal vectors

It is easy to see that the accumulated chord length parameterization is an approximation of the arc length parameterization. In some cases, there are still big errors between the approximate parameters and the original parameters. In this paper, we can further reduce the errors by using the given normal vectors $\{N_i : i = 0, \dots, m\}$ and the discrete points $\{P_i : i = 0, \dots, m\}$.

Since we can easily obtain the unit tangents $\{D_i : i = 0, \dots, m\}$ of the curve, which is perpendicular to the normal vector at each point. Naturally, it reminds us of the Taylor's formula:

$$S(t_{i+1}) = S(t_i + \Delta t_i) = S(t_i) + \Delta t_i \cdot S'(t_i) + \frac{\Delta t_i^2}{2} \cdot S''(t_i) + \frac{\Delta t_i^3}{6} \cdot S'''(\xi_i).$$

Referring to the definition of derivative, we can use the following divided difference to approximate the second-order derivative in Taylor's formula:

$$S''(t_i) \simeq \frac{D_{i+1} - D_i}{\Delta t_i}.$$

Then we can modify the Taylor's formula to the following form:

$$S(t_{i+1}) \simeq S(t_i) + \Delta t_i \cdot D_i + \frac{\Delta t_i^2}{2} \cdot \frac{D_{i+1} - D_i}{\Delta t_i}.$$

So, by the above formula, we can define the improved parameterization as follows.

Definition 3.1 *For given discrete points $\{P_i : i = 0, \dots, m\}$ and unit tangents $\{D_i : i = 0, \dots, m\}$, we compute the parameter for each point by:*

$$\begin{aligned} \Delta t_i &= \frac{2 \cdot \|P_{i+1} - P_i\|}{\|D_i + D_{i+1}\|}, \quad i = 0, 1, \dots, m-1. \\ t_0 &= 0, t_i = t_{i-1} + \Delta t_{i-1}, \quad i = 1, 2, \dots, m. \end{aligned} \tag{3.1}$$

If it is needed, we can normalize the parameters into $[0, 1]$ by dividing the length $t_m - t_0$. For example, for the discrete data points and normal vectors shown in Figure 1 with 5% noise intensity, the results obtained by new parameterization are better than the results by the accumulated chord length parameterization, as shown in the following table.

Original parameters	0.1	0.3	0.5	0.7	0.9	1
Results by chord method	0.083	0.189	0.434	0.518	0.831	1
Results by our method	0.097	0.304	0.518	0.697	0.9	1

Table 1 Comparison of parameterizations for the data in Figure 1

3.2. Dominant points selection based on normal vectors

As described above, the original dominant point selection method (DOM) [19] is based on the discrete curvature value, which is susceptible to noises and costs a lot of time when the noise intensity is large.

In this section, we introduce how to select dominant points by using given normal vectors. Similar to the DOM, we select the seed points at first. But differently, we use normal vectors instead of the curvatures in the condition of seed points selection.

Definition 3.2 *The normal deviation value δ_i is defined as follows:*

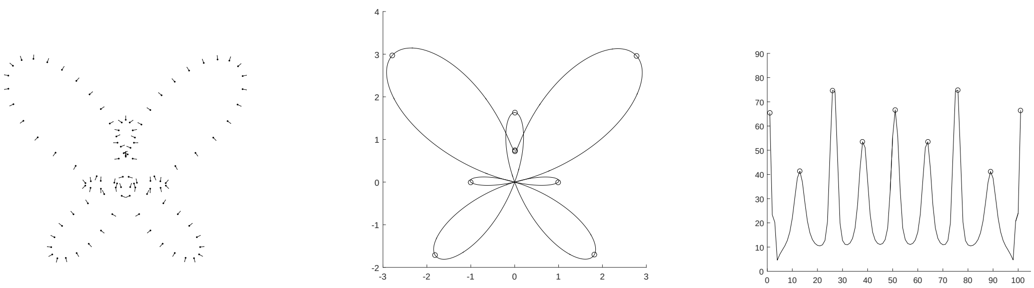
$$\delta_1 = \frac{1}{3}\sum_{j=0}^{3} d(N_1, N_j), \quad \delta_{m-1} = \frac{1}{3}\sum_{j=m-3}^{m} d(N_{m-1}, N_j),$$

$$\delta_i = \frac{1}{4}\sum_{j=i-2}^{i+2} d(N_i, N_j), \quad i = 2, \ldots, m-2, \tag{3.2}$$

where $d(N_i, N_j)$ is the angle between the two normal vectors.

Note that, the angle difference between the corresponding normal vectors is related to the complexity of the curve. Thus, we use δ_i as the criterion to determine the seed points. Let δ_{avg} be the average of the normal deviations of all data points. In order to reduce the effect of noises, besides two boundary points, we select those points with local maximal δ_i and satisfying $\delta_i > \delta_{avg}/4$ as seed points.

Using the same example in Figure 1, we test our approach for choosing the seed points. There are 101 points (without noise) which have been sampled from the original curve and we obtain the same 9 seed points as DOM.



(a) (b) (c)

Figure 2 Seed points by our approach and their corresponding normal deviations

We then can obtain a B-spline curve. Using selected dominant points, we divide the given points into a set of segments:

$$\{P_i : i = 0, \ldots, m\} = \{S_{d_0,d_1}, S_{d_1,d_2}, \ldots, S_{d_{n-1},d_n}\},$$

where $S_{d_j,d_{j+1}}$ is the segment containing all discrete points between two adjacent dominant points, i.e., $\{S_{d_j,d_{j+1}} = P_{d_j}, P_{(d_j)+1}, \ldots, P_{d_{j+1}}\}$.

Then discard the segment containing too few points from it ($|d_{j+1} - d_j| \leqslant 1$). Among the rest segments, we find the segment $S_{s,e}$ which contains the largest approximation error of the points $\|S(t_i) - P_i\|$.

Definition 3.3 *The overall normal deviation of segment $S_{s,e}$ can be defined as:*

$$\Delta_{s,e} = \sum_{i=s}^{e-1} (|\delta_i| + |\delta_{i+1}|)(t_{i+1} - t_i)/2.$$

Definition 3.4 *The proportion of segment $S_{s,e}$ can be defined as:*

$$\Delta t(s,e) = t(e) - t(s).$$

With $\Delta_{s,e}$ and $\Delta t(s,e)$, we use the following formula to find a new dominant point P_w in segment $S_{s,e}$:

$$\eta_{s,e} = r\frac{\Delta_{s,e}}{\Delta_{0,m}} + (1-r)\Delta t(s,e), \quad r \in [0,1]. \tag{3.3}$$

It is easy to see that Eq. (3.3) is changed from Eq. (2.3). We replace the curvature term by $\Delta_{s,e}$, and the chord length term by $\Delta t(s,e)$. For the parameter r in Eq. (3.3), when the noises for given data are big, the choice of r is reduced accordingly in our numerical experiments.

For the segment $S_{s,e}$ and a point $P_w \in S_{s,e}$, if it minimizes the difference $|\eta_{s,w} - \eta_{w,e}|$, then we add P_w as a new dominant point. That is, we find the point P_w satisfying:

$$\min_{w} |\eta_{s,w} - \eta_{w,e}|, \quad w \in (s,e). \tag{3.4}$$

After the new dominant point is obtained, the new curve can be reconstructed iteratively. Finally, we use the following stop condition.

Definition 3.5 *The algorithm stop condition is defined by*

$$\Delta_{s,e} < \mu \cdot \Delta_{avg}, \mu \in (0,1), \tag{3.5}$$

where $\Delta_{\rm avg}$ is the average value of all segments.

The parameter μ is selected artificially and can be modified by the noise intensity of data points. In our experiments, we use bigger parameter μ according to larger noises.

3.3. Curve fitting algorithm based on discrete data points and normal vectors

Although the least-square method is the most widely used approach, in some cases, the condition number of the linear equation system to be solved may be large and cause the problem to be ill-posed. This means that the measurement error carried by the given input data can seriously affect the result. In order to reduce the impact of data noises on the fitting results, the common practice is to change the original problem into the following variational form [24]:

$$\Phi(S) = \frac{1}{m}\sum_{i_p=0}^{m} \|S(t_{i_p}) - P_{i_p}\|^2 + \alpha\int_0^1 S^{''}(u)^2 \mathrm{d}u = \min,$$

where α is the regularization parameter. Its role is to balance the approximation error and the overall smoothing of the curve.

Different from the above method, we use another new term defined by discrete normal vectors to replace the integral term. Note that the tangent vector $S_u(t_{i_p})$ and the normal vector N_{i_p} at each point should satisfies

$$S_u(t_{i_p}) \cdot N_{i_p} = 0, \quad i_p = 0, \ldots, m. \tag{3.6}$$

We combine the normal vectors constraint with discrete data points approximation error, then the curve fitting approach is based on the following optimization problem:

$$E = E_P + \alpha E_N = \min, \tag{3.7}$$

with

$$E_P = \sum_{i_p=0}^{m} \|S(t_{i_p}) - P_{i_p}\|^2,$$

and combining Eq. (2.1) with Eq. (3.6), we have

$$E_N = \sum_{i_p=0}^{m} [S_u(t_{i_p}) \cdot N_{i_p}]^2 = \sum_{i_p=0}^{m} \Big[\sum_{i_q=1}^{n} a'_{i_q,i_p} (Q_{i_q} \cdot N_{i_p}) \Big]^2,$$

where

$$a'_{i,j} = B'_{i,p}(t_j) = p[\frac{B_{i,p-1}(t_j)}{t_{i+p} - t_i} - \frac{B_{i+1,p-1}(t_j)}{t_{i+p+1} - t_{i+1}}].$$

We can turn the above minimization problem into the following linear equation system:

$$\frac{\partial E}{\partial Q_i} = 0, \quad i = 0, \dots, n.$$

Then, we get the expanded form of the algorithm:

$$\sum_{i_q=0}^{n} (\sum_{i_p=0}^{m} a_{i,i_p} a_{i_q,i_p} I + \alpha \sum_{i_p=0}^{m} [a'_{i,i_p} a'_{i_q,i_p} N_{i_p} N_{i_p}^T]) \cdot Q_{iq} = \sum_{i_p=0}^{m} a_{i,i_p} P_{i_p}, \quad i = 0, \dots, n, \tag{3.8}$$

where I is the identity matrix of dimension 2×2 for the curve $S \in \mathbb{R}^2$ and 3×3 for $S \in \mathbb{R}^3$, and

$$a_{i,i_p} = B_{i,p}(t_{i_p}).$$

Denote two matrices by

$$A = \Big(\sum_{i_p=0}^{m} a_{i,i_p} a_{i_q,i_p} I \Big)_{i,i_q=0,\dots,n}, \quad B = \Big(\sum_{i_p=0}^{m} a'_{i,i_p} a'_{i_q,i_p} N_{i_p} N_{i_p}^T \Big)_{i,i_q=0,\dots,n},$$

the column vector composed of all unknown control points by $Q = (Q_0^T, \dots, Q_n^T)^T$, and the right column vector in Eq. (3.8) by

$$C = \Big(\sum_{i_p=0}^{m} a_{0,i_p} P_{i_p}^T, \dots, \sum_{i_p=0}^{m} a_{n,i_p} P_{i_p}^T \Big)^T.$$

Then the linear system (3.8) is the matrix form $(A + \alpha B)Q = C$.

Here, the role of the parameter α is to balance the approximation errors of the discrete points and normal vectors. In this paper, the de Boor's method is used to select the parameter α, which is to balance the trace of the two matrices A and B:

$$\text{trace}(A) = \alpha \cdot \text{trace}(B).$$

This method is also known as the trace balance method [25].

In summary, we conclude the main steps of the above algorithm, denoted by NDOM.

Algorithm 1 (NDOM) Curve reconstruction based on discrete data points and normal vectors

Input: The given discrete points $\{P_i,\ i=0,\dots,m\}$ and corresponding normal vectors $\{N_i,\ i=0,\dots,m\}$.

Output: The reconstructed curve S.

Step 1. Obtain the parameter values $\{t_i,\ i=0,\dots,m\}$ of the discrete points using the proposed parameterization method in Eq. (3.1).

Step 2. Calculate the normal deviation value $\{\delta_i,\ i=1,\dots,m-1\}$ using Eq. (3.2), and choose the seed points as initial dominant points.

Step 3. Compute the knot vector U by dominant points and Eq. (2.2).

Step 4. Obtain the reconstructed curve S using the fitting algorithm based on discrete points and normal vectors, as shown in Eq. (3.8).

Step 5. Add a new dominant point P_w satisfying Eq. (3.4), then obtain a new reconstructed curve S by repeating Steps 3 and 4.

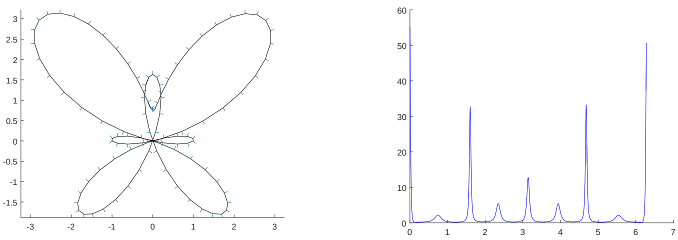
Step 6. Repeat Step 5 until the stopping condition Eq. (3.5) is satisfied.

4. Numerical experiments

In this section, compared with the algorithm DOM in [19], we test the performance of the proposed algorithm NDOM to reconstruct cubic B-spline curves from the given sample data with different noise intensities.

Example 4.1 Figure 3 shows a butterfly curve which is defined as:

$$\begin{cases} x=\sin t\cdot(e^{\cos t}-2\cos 4t+(\sin\frac{1}{12}t)^5), \\ y=\cos t\cdot(e^{\cos t}-2\cos 4t+(\sin\frac{1}{12}t)^5), \end{cases} \quad t\in[0,2\pi].$$



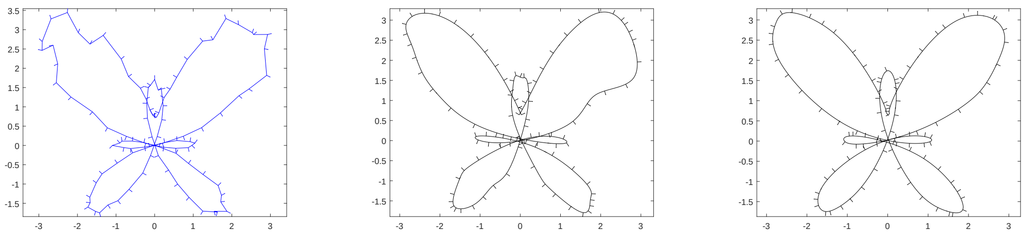
(a) Curve and normal vectors (b) Curvature plot

Figure 3 The sample curve in Example 4.1

We take 101 sample points $\{P_i=(x_i,y_i),\ i=0,\dots,100\}$ and normal vectors $\{N_i=(Nx_i,Ny_i),\ i=0,\dots,100\}$ from the curve, as shown in Figure 3 (a). Figure 3(b) shows the variation of curvature of the whole curve. Since this paper focuses on the data points containing noises, we use the following formula to simulate the noises of the discrete points and normal vectors:

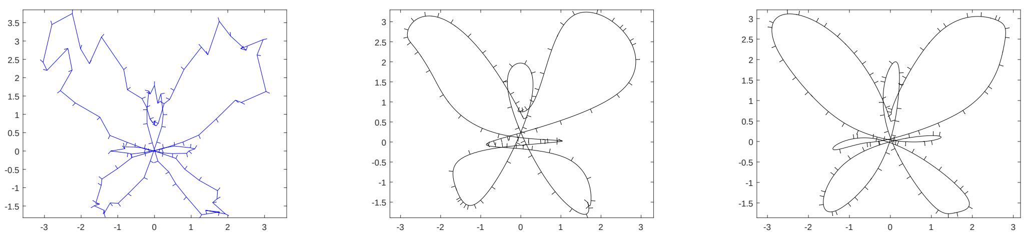
$$\begin{cases} \tilde{x}_i = x_i \cdot (1 + \psi r), \\ \tilde{y}_i = y_i \cdot (1 + \psi r), \\ \tilde{N}x_i = Nx_i \cdot (1 + \psi r), \\ \tilde{N}y_i = Ny_i \cdot (1 + \psi r), \end{cases}$$

where r is a random number with a range of $[-1, 1]$, and ψ is the simulated noise intensity. In this example, we show two cases: $\psi = 10\%$ and $\psi = 20\%$, as shown in Figure 4 (a) and Figure 5 (a).



(a) Input data with 10% noise intensity (b) Result obtained by DOM (c) Result obtained by NDOM

Figure 4 Results with 10% noise intensity in Example 4.1

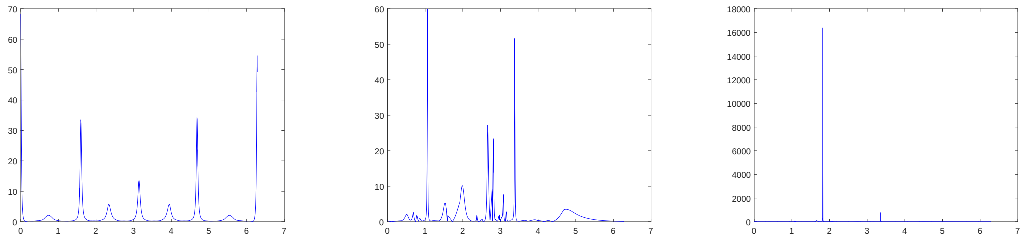


(a) Input data with 20% noise intensity (b) Result obtained by DOM (c) Result obtained by NDOM

Figure 5 Results with 20% noise intensity in Example 4.1

Figure 4 (b) and Figure 5 (b) are correspondingly reconstructed curves by DOM. Figure 4 (c) and Figure 5 (c) are results by NDOM. It is obvious that NDOM can retain better geometric shape of the curve than DOM for the two cases.

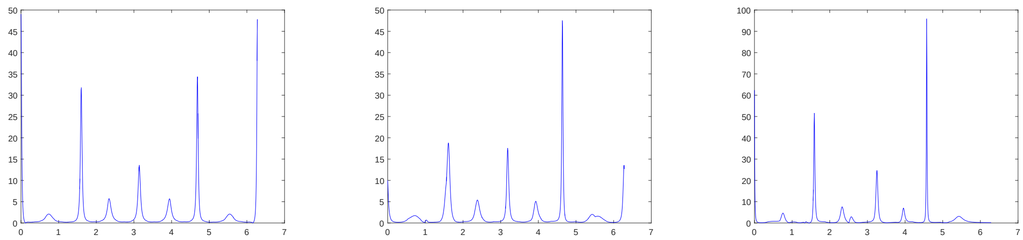
In addition, we compare curvature variation of the reconstructed curves by two algorithms DOM and NDOM, as shown in Figure 6. It also shows the results by NDOM are better than those by DOM for two cases.



(a) Dom's result wih noise free data

(b) DOM's resultwith 10% noise intensity

(c) DOM's result with 20% noise intensity



(d) NDom's result wih noise free data

(b) NDOM's resultwith 10% noise intensity

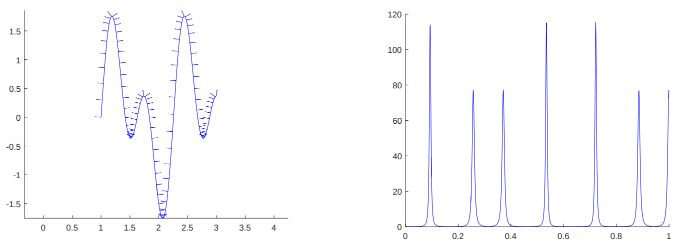
(c) NDOM's result with 20% noise intensity

Figure 6 Curvature variations of results in Example 4.1

Example 4.2 Figure 7 shows another plane curve which is defined as:

$$\begin{cases} x = 2t + 1, \\ y = \sin(10t) + \sin(20t), \end{cases} \quad t \in [0, 1].$$

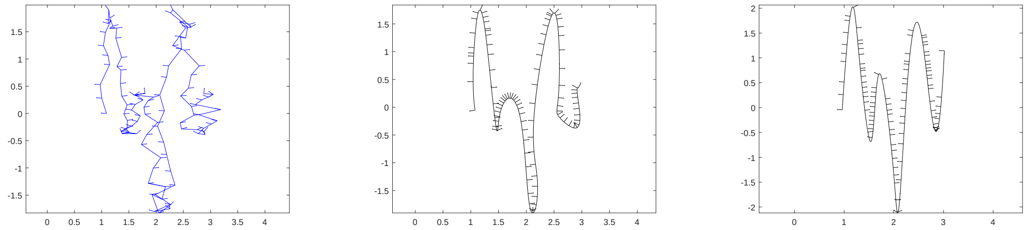
The reconstructed results are shown in Figures 8, 9, 10 by DOM and NDOM for two cases of noises, respectively. Similarly to Example 4.1, NDOM can obtain better results on geometric shape and curvature variation than DOM.



(a) Curve and normal vectors

(b) Curvature plot

Figure 7 The sample curve in Example 4.2

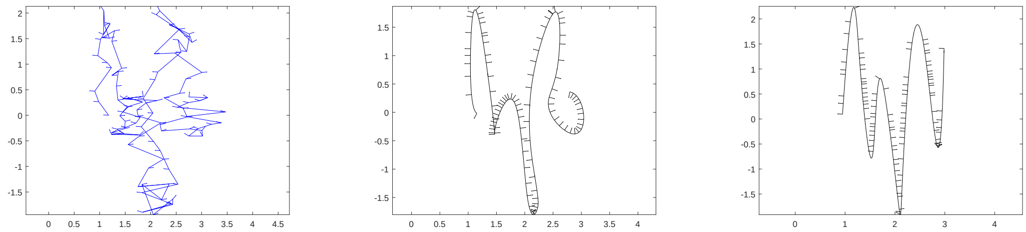


(a) Input data with 10% noise intensity

(b) Result obtained by DOM

(c) Result obtained by NDOM

Figure 8 Results with 10% noise intensity in Example 4.2

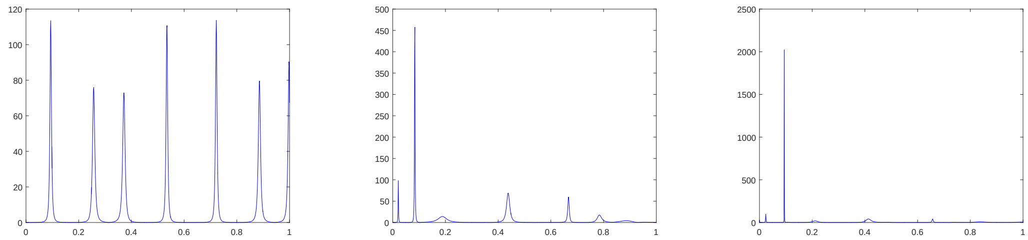


(a) Input data with 20% noise intensity

(b) Result obtained by DOM

(c) Result obtained by NDOM

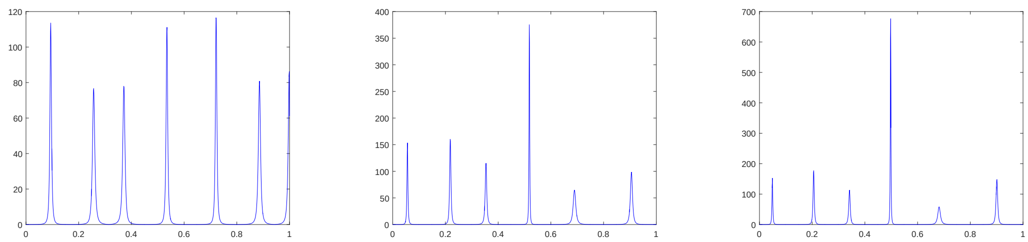
Figure 9 Results with 20% noise intensity in Example 4.2



(a) Dom's result wih noise free data

(b) DOM's resultwith 10% noise intensity

(c) DOM's result with 20% noise intensity



(d) NDom's result wih noise free data

(b) NDOM's resultwith 10% noise intensity

(c) NDOM's result with 20% noise intensity

Figure 10 Curvature variations of results in Example 4.2

Furthermore, in order to demonstrate that the algorithm NDOM can also be applied well on spatial curves, we show the following two examples.

Example 4.3 Figure 11 shows the spatial curve and reconstructed results. The curve is defined as:

$$\begin{cases} x = t\cos(t), \\ y = t\sin(t), \quad t \in [0, 16]. \\ z = t, \end{cases}$$

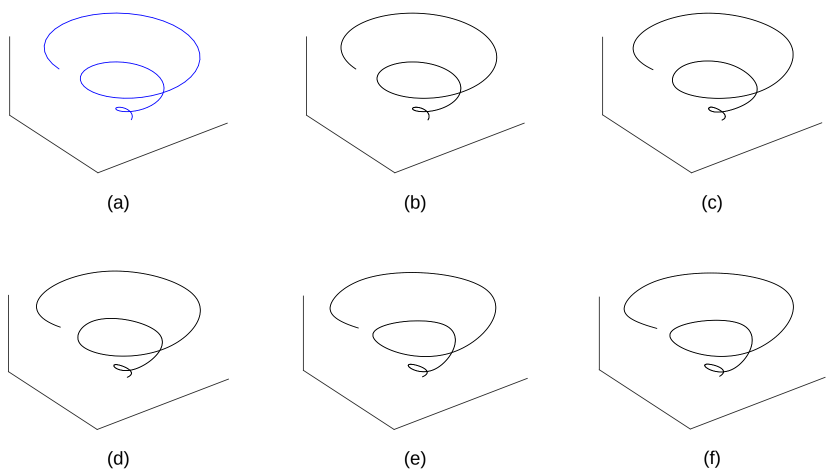


Figure 11 (a) the original curve, (b) result obtained with noise free data, (c)-(f) results obtained with noise intensity of 5%-20% in Example 4.3

Figure 11 (a) shows the original curve and Figure 11 (b)–(f) show the results obtained by NDOM for different noise intensity of 0%, 5%, 10%, 15% and 20%, respectively.

Example 4.4 Figure 12 shows the Viviani's curve and reconstructed results by NDOM. The curve is defined as:

$$\begin{cases} x = \frac{1}{2}(1 + \cos t), \\ y = \frac{1}{2}\sin t, \qquad t \in [0, 4\pi]. \\ z = \sin(\frac{t}{2}), \end{cases}$$

From the above two examples, NDOM can reconstruct good results even for the sample spatial data with big noises.

5. Concluding remarks

In this paper, we present a curve reconstruction algorithm based on discrete data points and normal vectors. This approach is different from the existing methods in the following aspects: First, it uses tangent vectors to improve the parameterization of the discrete data points, which is better than accumulated chord length parameterization. Secondly, we modify the dominant point selection method with the help of normal vectors. Thirdly, we improve the curve fitting approximation model by combining the normal vector fitting error with approximation error to discrete data points. As a result, this approach can retain good geometric shape and curvature variations, even for the data with big noises. Besides, we also show that this approach can work well on the data points in 3D space.

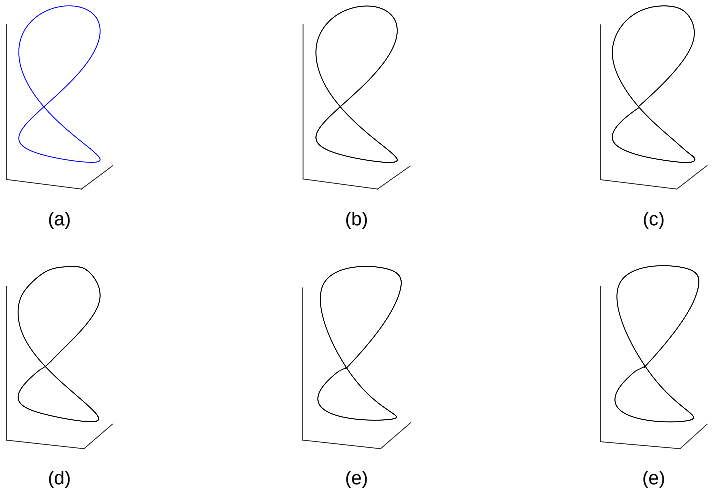


Figure 12 (a) the original curve, (b) result obtained with noise free data, (c)–(f) results obtained with noise intensity of 5%–20% in Example 4.4

References

[1] J. HOSCHEK, D. LASSER, L. L. SCHUMAKER. *Fundamentals of Computer Aided Geometric Design*. A. K. Peters, London, 1993.

[2] L. PIEGL, W. TILLER. *the NURBS Book*. Springer-Verlag, New York, 1996.

[3] G. E. FARIN, G. FARIN. *Curves and Surfaces for CAGD: a Practical Guide*. Margan Kaufmann, San Francisco, 2001.

[4] L. BARDIS, N. M. PATRIKALAKIS. *Surface approximation with rational B-splines*. Engineering with Computers., 1990, **6**(4): 223–235.

[5] S. T. TUOHY, L. BARDIS. *Low-degree approximation of high-degree B-spline surfaces*. Engineering with Computers., 1993, **9**(4): 198–209.

[6] D. F. ROGERS, N. R. FOG. *Constrained B-spline curve and surface fitting*. Computer-Aided Design., 1989, **21**(10): 641–648.

[7] B. SSRKAR, C. H. MENQ. *Parameter optimization in approximating curves and surfaces to measurement data*. Computer Aided Geometric Design., 1991, **8**(4): 267–290.

[8] H. PARK. *An error-bounded approximate method for representing planar curves in B-splines*. Computer Aided Geometric Design., 2004, **21**(5): 479–497.

[9] X YE, T. R. JACKSON, N. M. PATRIKALAKIS. *Geometric design of functional surfaces*. Computer-Aided Design., 1996, **28**(9): 741–752.

[10] F. PEREZ-ARRIBS, R. PEREZ-FERNANDEZ. *A B-spline design model for propeller blades*. Advances in Engineering Software., 2018, **118**: 35–44.

[11] Xiaodiao CHEN, Weiyin MA. *Geometric point interpolation method in $\mathbb{R}^3$ space with tangent directional constraint*. Computer-Aided Design., 2012, **44**(12): 1217–1228.

[12] Xunnian YANG. *Fitting and fairing Hermite-type data by matrix weighted NURBS curves*. Computer-Aided Design., 2018, **102**: 22–32.

[13] A. ABBAS, A, NASRI, T. MAEKAWA. *Generating B-spline curves with points, normals and curvature constraints: a constructive approach*. The Visual Computer., 2010, **26**(6–8): 823–829.

[14] Y. YAMAURA, T. NANYA, H. IMOTO. *Shape reconstruction from a normal map in terms of uniform bi-quadratic B-spline surfaces*. Computer-Aided Design., 2015, **63**: 129–140.

[15] S. GOFUKU, S. TAMURA, T. MAEKAWA. *Point-tangent/point-normal B-spline curve interpolation by geometric algorithms*. Computer-Aided Design., 2009, **41**(6): 412–422.

[16] A. SEILER, D. GROSSMANN, B. JÜTTLER. *Spline Surface Fitting using Normal Data and Norm-like Functions*. Computer Aided Geometric Design., 2018, **64**: 37–49.

[17] Renhong WANG, Chongjun LI, Chungang ZHU. *Computational Geometry Tutorial*. Science Press, Beijing, 2008. (in Chinese)

[18] E. T. Y. LEE. *Choosing nodes in parametric curve interpolation*. Computer-Aided Design., 1989, **21**(6): 363–370.

[19] H. PARK, J. H. LEE. *B-spline curve fitting based on adaptive curve refinement using dominant points*. Computer-Aided Design., 2007, **39**(6): 439–451.

[20] P. LAURENT-GENGOUX, M. MEKHILEF. *Optimization of a NURBS representation*. Computer Aided Design., 1993, **25**(11): 699–710.

[21] B. HAMANN, Jiannliang CHEN. *Data point selection for piecewise linear curve approximation*. Computer Aided Geometric Design., 1994, **11**(3): 289–301.

[22] G. H. LIU, Y. S. WONG, Y. F. ZHANG, et al. *Adaptive fairing of digitized point data with discrete curvature*. Computer Aided Design., 2002, **34**(4): 309–320.

[23] H. PARK, K, KIM, S. C. LEE. *A method for approximate NURBS curve compatibility based on multiple curve refitting*. Computer Aided Geometric Design., 2000, **32**(4): 237–252.

[24] D. N. HAO, D. LESNIC. *Heuristic regularization methods for numerical differentiation*. Computers & Mathematics with Applications., 2012, **63**(4): 816–826.

[25] C. D. BOOR. *Spline Toolbox for Use with MATLAB*. The MathWorks, Natick, 2003.