A Greedy Algorithm for Constraint Principal Curves

Shiying Yang
State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044，China
Email: 12120291@bjtu.edu.cn

Dewang Chen*
State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044，China
Corresponding author: dwchen@bjtu.edu.cn

Xiangyu Zeng
State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044，China
Email: 11120319@bjtu.edu.cn

Peter Pudney
Center for Industrial and Applied Mathematics, University of South Australia, Mawson Lakes, Australia
Email: Peter.Pudney@unisa.edu.au

***Abstract*—Principal curves can learn high-accuracy data from multiple low-accuracy data. However, the current proposed algorithms based on global optimization are too complex and have high computational complexity. To address these problems and in the inspiration of the idea of divide and conquer, this paper proposes a Greedy algorithm based on dichotomy and simple averaging, named as KPCg algorithm. After that, three simulation data sets of sinusoidal, zigzag and spiral trajectories are used to test the performance of the KPCg algorithm and we compare it with the k-segment algorithm proposed by Verbeek. The results show that the KPCg algorithm can efficiently learn high-accuracy data from multiple low-accuracy data with constraint endpoints and have advantages in accuracy, computational speed and scope of application.**

***Index Terms*—Principal curves algorithm; principal of nearest neighbor; adaptive radius; dichotomy; simple averaging**

I. Introduction

A principal curve is a smooth curve that passes through the middle of a data set. In statistics, the principal curve should be self-consistent; each point on the principal curve is the expected average of the data that projects to this point. Principal curves have been widely used in a growing number of areas, such as fingerprint skeleton extraction, hydraulic machinery, image processing [1-3]. We are particularly interested in automatic generation of railway track profiles from GPS data [4-6].

In 1904, Spearman proposed the linear principal component analysis method. This method is simple, and is now one of important tools for statistical analysis of data [7]. But not all data is linear. Hastie proposed the concept of principal curves in 1984; these smooth curves should pass through the "center" of the data distribution and satisfy a "self-consistency" property [8-9]. In 1992, Banfield and Raftery principal curves (BR principal curves) improved the previous principal curves. But BR principal curves bring the numerical instability. So BR principal curves algorithm may obtain smooth but false principal curves [10]. In 1997, Kegl proposed the concept of length constraint principal curves [11], and proved the existence and uniqueness of the K principal curves. He also proposed a polygonal-time algorithm to obtain K principal curves.

In 2000, Verbeek proposed the K-Segment principal curves (KPCv) algorithm [12]. In his paper, Verbeek uses an incremental method when finding principal curves. Line segments are fitted and connected to form polygonal lines. New segments are inserted until a performance criterion is met. However, when we add segments to improve the accuracy of a principal curve, the algorithm may fail as shown in Figure1, and the generated principal curve needs to be reprocessed in the next stage.

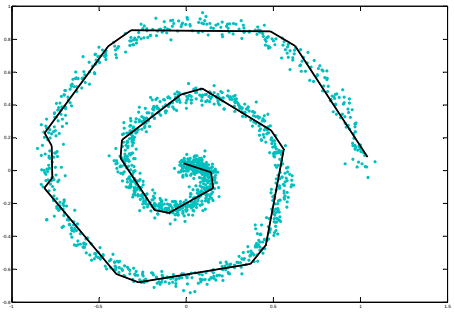
There are many principal curves algorithms based on global optimization. In 2008, Chen proposed a heuristic algorithm based on continual split and non-linear optimization to estimate the path of a railway line from multiple runs collecting GPS data [13]. Still, this algorithm has some defects including long computation times and narrow application scope. After that, Zhang and Chen proposed a principal curves algorithm [14] which resolved the generation and adaptability of principal curves with constraint points. But the algorithm is too complex to be practical. In 2011, Jia proposed the MPM (Max Point Method) optimization algorithm for constraint principal curves [15]. However, the fitness and robustness need to be improved. Also in 2011, Zhang proposed two principal curve algorithms for partitioning

doi:10.4304/jcp.9.5.1125-1130

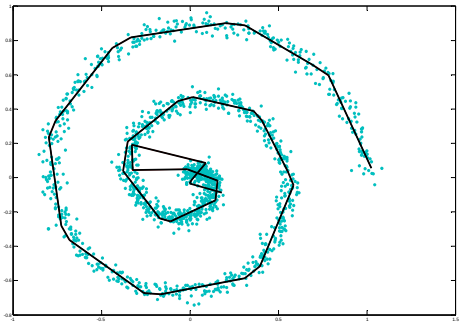
high-dimensional data spaces [16]. And both of them can show a better performance than some other competing partitioning algorithms in some ways.

To increase the computation efficiency, we propose the Greedy algorithm based on dichotomy and simple averaging, named as KPCg algorithm. As the start point and end point of the principal curve are known, we draw circles from the start point to the endpoint with adaptive radii, where the radii values are determined by dichotomy. And we search for the vertexes by simple averaging. Finally, we can get the principal curve by connecting the vertexes in order. To test the performance of the proposed principal curves algorithm, we compare it with the K-Segment algorithm proposed by Verbeek (named as KPCv algorithm in this paper). The results show that our algorithm can efficiently generate good approximations from multiple low-accuracy data and has advantages in accuracy, computational time and scope of application.

The structure of the paper is as follows: Section II defines a constraint principal curve and describes how we calculate errors; In Section III, we describe the KPCg algorithm in detail; and Section IV shows the results of the verification and analyzes the results; we end this paper with a conclusion.



(a) KPCv principal curve with 12 segments



(b) KPCv principal curve with 17 segments

Figure1. Results for KPCv algorithm

II. Constraint Principal Curves and Error Model

A. Constraint Principal Curves

In many practical applications, some points can be fixed as constraints when generating a principal curve and we need to take them into account. Therefore, a constraint principal curve is a principal curve with several fixed points (i.e. points which have been measured accurately). In Figure2, the green curve is a principal curve and the black curve with two fixed red endpoints is a constraint principal curve where V_s is the start point and the V_e is the end point.

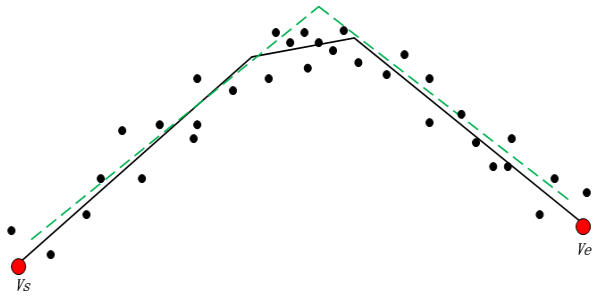


Figure2. Principal curve and constraint principal curve

When generating a principal curve, making full use of the fixed points can help us obtain a constraint principal curve of higher precision.

B. Principle of Nearest Neighbor and Error Model

Principal of nearest neighbor has been used in some areas, such as querying, filtering [17, 18]. Here, it can be used practically and extensively when establishing the error model.

Similar to K-Segment principal curves, we will represent a principal curve by a set of vertices and the line segments between adjacent vertices. By comparing the projection distance from the data points to each line segment and each vertex, the data points can be divided into different parts: one line segment or one vertex. As illustrated in Figure3, the constraint principal curve is composed of vertexes V_j (j = 1,…, n), and line segments $S_{j,j+1}$, j = 1, …, n-1. Here, n is the number of vertexes.

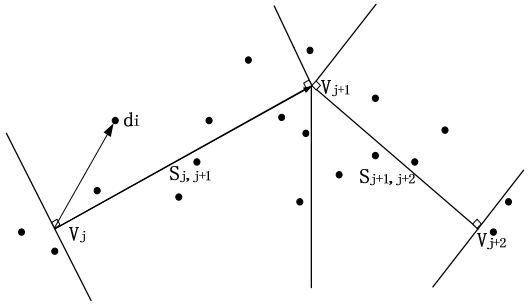


Figure3. Principle of nearest neighbor

The distances from data point d_i (i = 1, .., m, m is the number of data points), to vertex V_j can be computed by Formula (1).

$$D_{P_{i,j}} = \left|\overrightarrow{V_j d_i}\right| = \sqrt{(X_{d_i} - X_{V_j})^2 + (Y_{d_i} - Y_{V_j})^2} \quad (1)$$

The distance from data point d_i to corresponding line-segment $S_{j,j+1}$can be computed by Formula (2).

$$D_{L_{i,j}} = \left|\overrightarrow{V_j V_{j+1}} \times \overrightarrow{V_j d_i}\right| \Big/ \left|\overrightarrow{V_j V_{j+1}}\right| \quad (2)$$

The smallest one of all $D_{P_{i,j}}$ and $D_{L_{i,j}}$ (j = 1,…, n) is defined as the projection distance of each data point to the principal curve, show in the following Formula (3). So E_i is the minimum distance from point d_i to the principal curve.

$$E_i = \min\{D_{P_{i,j}}(j=1,2,\cdots,n), D_{L_{i,j}}(j=1,2,\cdots,n)\} \quad (3)$$

$\overline{E}$ is the mean of all E_i (i = 1, .., m, m is the number of data points), and E is the error of the principal curve, shown in Formula (4).

$$E = \overline{E} = \sum_{i=1}^{m} E_i \Big/ m \quad (4)$$

When we have more than two fixed points through which the principal curve must pass, we can treat each pair of adjacent fixed points as the start and end points of a separate principal curve. This reduces the problem to that of finding a principal curve where the first and last vertexes are specified. Our objective is to define vertexes V_1, …, V_n (and the corresponding line segments) so that the mean distance E is as smaller as possible.

III. The Greedy Algorithm: KPCg Algorithm

In the inspiration of the idea of divide and conquer, a Greedy algorithm based on dichotomy and simple averaging is developed. We name it the KPCg algorithm. The KPCg algorithm progresses from the start endpoint towards the end point with adaptive radius. It uses simple averaging to find new vertexes.

In the following parts, V_s is the start point, V_e is the end point. n is the number of vertexes and it will be unknown until we find all the vertexes. We set the upper and lower bounds of local error E^l_j (j=1,2,...,n-1): E_{min} and E_{max}, where E_{min}=0 usually. E is the overal error of the principal curve we obtain by the principal curves algorithm. For each E^l_j (j=1,2,...,n-1), if E^l_j <=E_{max}, e<=E_{max}. In the KPCg algorithm, simple averaging is used to calculate the new vertex V_j (j=2, 3,..., n-1, V_1 is V_s and V_n is V_e), and we use dichotomy to make the radii adaptive.

A. Dichotomy

We can continuously approximate the target by adaptive algorithm like the adaptive genetic algorithm [19]. When drawing circles, adaptive radius can be obtained according to the dichotomy to make E^l_j (j=1, 2,..., n-1) ∈ [E_{min}, E_{max}]. Specific methods are as follows:

step1. The center of a circle is V_j (j=1, 2,..., n-1), the initial interval of the radius is [R_d R_u]. R_d=0 and R_u=2* d^e_j, where d^e_j is defined as the distance from V_j to V_e;

step2. R_j= R_d + (R_u - R_d)/2, j=1,2...,n-1;

step3. Draw the circle. We can find a vertex V_{j+1} by simple averaging introduced in the next part and compute the local error E^l_j (j=1, 2,..., n-1);

step4. Judge: if E_{min}<=E^l_j<=E_{max}; the value of R_j is what we need; if E^l_j <=E_{min}, then R_d =R_j and turn to the second step; else, R_u =R_j and turn to Step2.

Then, we can get a suitable value of R_j (j=1, 2,..., n-1). By connecting V_j and V_{j+1}, we can get a sub principal curve whose local error E^l_j ∈ [E_{min}, E_{max}].

The flowchart of dichotomy is shown in Figure4.

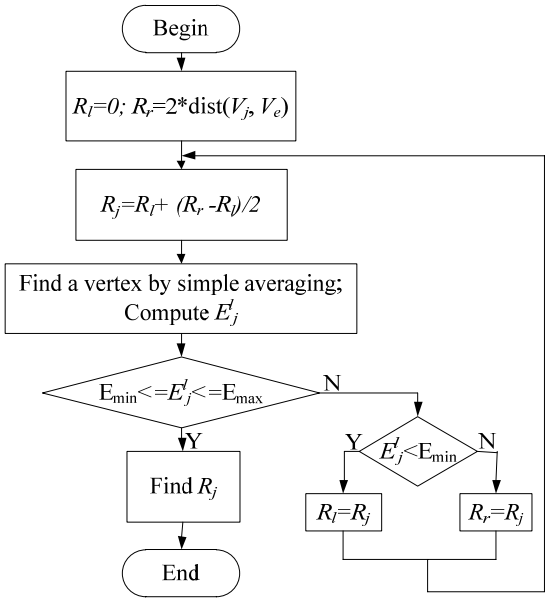


Figure4. The flowchart of dichotomy

B. Simple Averaging

When solving the problems, simple averaging can also help us make the best choice in the current view. That is to say, we can only use simple averaging to get a local optimum in some sense, but without the whole consideration. This can speed up the computation.

As we see, a principal curve is constructed by different lines that are sub principal curves. For each line, when the start point is known (that is the end point of the previous line), we need to find this line's end point, and then a line is obtained. In the KPCg algorithm, we use simple averaging to calculate the new vertexes: every vertex's coordinates are the mean value of the point coordinates within a ring. Here, the ring's inner radii R_j' (j=1,2,...,n-1) can be obtained by R_j'=R_j'*0.9 and the outer radii R_j (j=1,2,...,n-1) is obtained by dichotomy. For better displaying the idea of simple averaging, we use sectors to represent circles in Figure5. We will make a specific introduction of simple averaging by searching for V_2.

V_s is the first vertex, that is V_1, and we need to find V_2. The initial value of R_1 is the distance from V_e to V_1, that is R_1= R_u/2= d^e_1. And R_1' is obtained by R_1'=R_1*0.9. So we get the first ring, and the coordinates of V_2 are the mean value of the coordinates of the points within the ring. Assuming that V_1, V_2 and the line between them construct a sub principal curve, we compute the local error E^l_1 of the sub principal curve according to the error model introduced in Section II part B. If E_{min}<=E^l_1<=E_{max}, we get V_2. Otherwise, we change the value of R_1 by dichotomy introduced in the previous part. Finally, we can find V_2 and obtain a sub principal curve which can pass through the middle of the data set whenever possible.

Since V_2 has been obtained, we are now searching for V_3 in the same way. After that, V_3, V_4,..., V_{n-1} can also be

found in this way and V_e is the last vertex V_n (n is the number of vertexes).

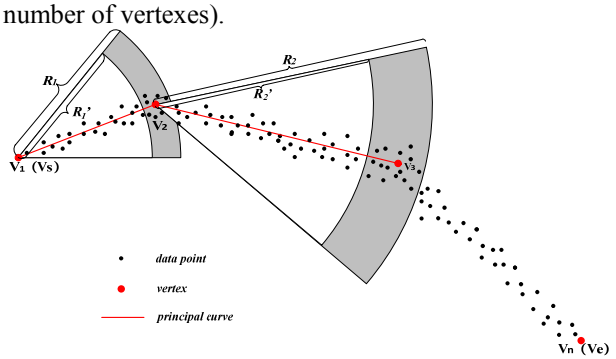


Figure5. Simple averaging

C. The KPCg Algorithm

The KPCg algorithm is a constraint principal curves algorithm based on dichotomy and simple averaging. We draw circles from the start point to the end point, where the radii' values are determined by dichotomy according to the upper and lower limits of local error, and we fit the unused data in each circle by simple averaging.

The key steps are as follow:

step1. n=2, V_s'=V_s;

step2. Connecting V_s' and V_e. Assuming that V_s', V_e and the line between them construct a sub principal curve, we compute the local error E^l_{n-1} of the sub principal curve. If E_{min}<= E^l_{n-1}<=E_{max}, turn to Step6. Otherwise, turn to step3;

step3. n=n+1;

step4. Use dichotomy to find a value of R_j and use simple averaging to find V_{n-1}, where $E^l_{n-1} \in [E_{min}, E_{max}]$;

step5. V_s'=V_{n-1}, turn to Step2;

step6. Line V_1, V_2,..., V_n. Then we get the principal curve and calculate the overall error which will be smaller than E_{max}.

The flowchart of the KPCg algorithm is shown in Figure6.

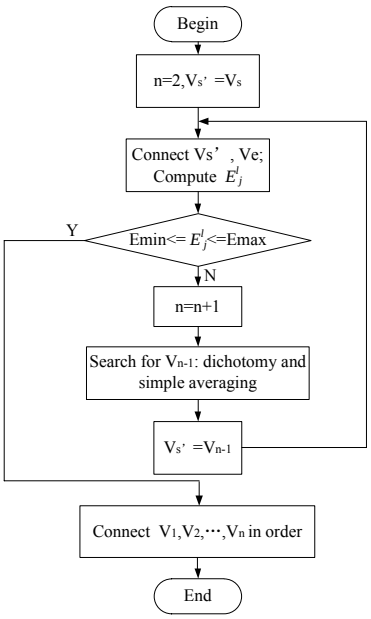


Figure6. The flowchart of the KPCg algorithm

IV. Verification and Analysis

A. Acquisition of Simulation Dataset

To test the fitness of the KPCg algorithm, we conducted experiments on three artificial datasets of sinusoidal trajectory, zigzag trajectory and spiral trajectory. And the acquisition of these simulation dataset is: Firstly, we generate an accurate simulated trajectory such as the sinusoidal trajectory. Then, some data points are randomly generated around the accurate trajectory.

In this way, we get the simulation datasets: sinusoidal dataset with 1000 data points, zigzag dataset with 2000 data points and spiral dataset with 1500 data points.

B. Evaluation Indexes

In the experiments, we do not need to compare the generated principal curves with the exact trajectory for that the data are corrupted by the noise we add.

To make the comprehensive comparision on the KPCg algorithm and the KPCv algorithm, we define the following indices: 1) N is the number of the vertexes generated by each algorithm. The smaller the N is, the less storage it consumes; 2) T represents the time each algorithm uses when generating a k-segment principal curve (KPC); 3) and E is defined in Eq.(4).

C. Verification

1. Sinusoidal data set

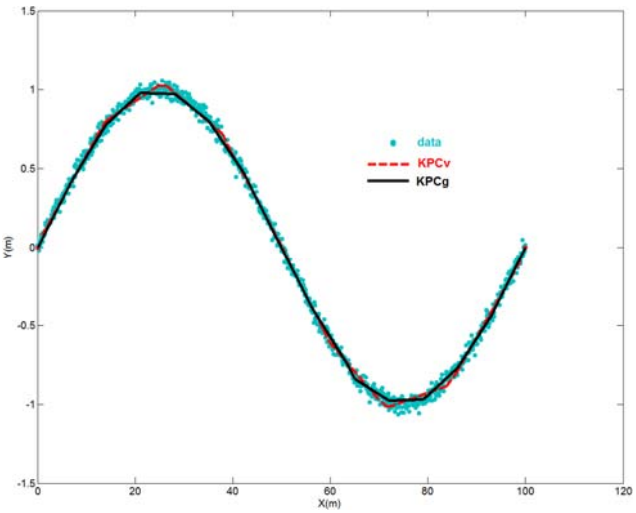


Figure7. Results on sinusoidal data set

TABLE I.
COMPARISON ON SINUSOIDAL DATA SET

Algorithm	N	T/s	E/m
KPCv	16	17.5	0.0295
KPCg	**15**	**2.8**	**0.0250**

2. Zigzag data set

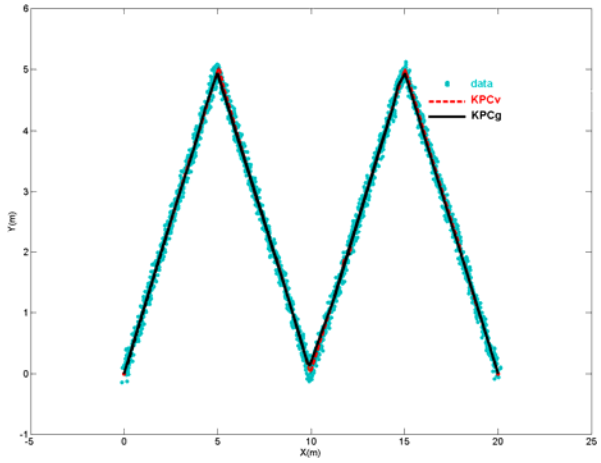


Figure8. Results on zigzag data set

TABLE II.
COMPARISON ON ZIGZAG DATA SET

Algorithm	N	T/s	E/m
KPCv	18	15.2	0.0742
KPCg	18	**7.4**	**0.0731**

3. Spiral data set

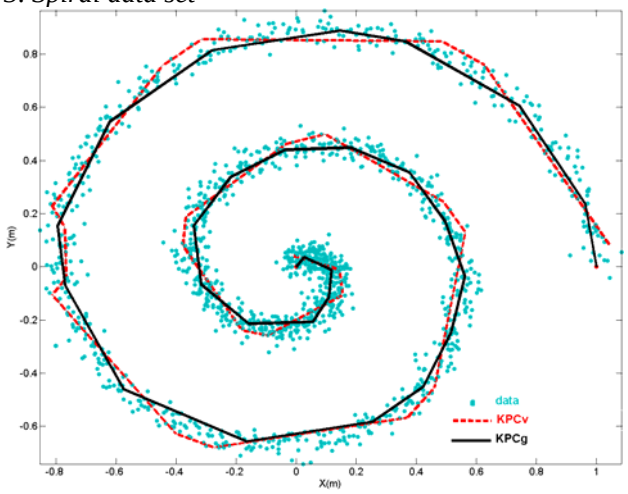


Figure9. Results on spiral data set

TABLE III.
COMPARISON ON SPIRAL DATA SET

Algorithm	N	T/s	E/m
KPCv	**24**	30.1	0.0355
KPCg	28	**9.0**	**0.0269**

D. Analysis

Figure7-9 show that the KPCg algorithm can learn high-accuracy data from multiple low-accuracy data. What's more, the KPCg algorithm also has a large scope of application.

When comparing with the KPCv algorithm in TABLEI-III, we can find that the KPCg algorithm is faster because of the simple averaging. Owing to the constraint endpoints and adaptive radius, the error of the KPCg algorithm is smaller than that of the KPCv.

V. CONCLUSIONS

In this paper, the history and defects in principal curves are introduced firstly. Then, in the inspiration of the idea of divide and conquer, we proposed the KPCg algorithm, a Greedy algorithm based on dichotomy and simple averaging. At last, we compare it with the KPCv algorithm by simulation data.

For the usage of dichotomy, simple averaging and the constraint endpoints, the results show that the KPCg algorithm has a good performance on accuracy, computational efficiency and the scope of application.

In future research, we will study the constraint principal curves algorithms deeply, and improve them so that the precision and robustness can be enhanced.

ACKNOWLEDGMENT

This work is partially supported by New Scientific Star Program of Beijing under grant 2010B015, by the Fundamental Research Funds for the Central Universities under grant 2012JBM016, by the independent research project from the State Key Laboratory of Rail Traffic Control and Safety under grant RSC2011ZT001 and by the National High Technology Research and Development Program ("863" Program) of China under grant 2012AA112800,.

REFERENCES

[1] C. Ma, H.Y. Zhang, D.Q. Miao, Improvement of principal curves algorithm and its application in fingerprint skeleton extraction, Computer Engineering and Applications, 46(16), 2010, pp. 170-173.

[2] Y.H. Wang, Z.Y. Shen, Y.Sun, Numerical simulation of characteristic curves of hydraulic turbine based on principal curves, Journal of Hydroelectric Engineering, 28(3), 2009, pp. 181-186.

[3] H. Su, F.G. Huang, An image segmentation method based on AEP and K principal curves, Journal of Harbin Engineering University, 25(6), 2004, pp. 756-760.

[4] D.W Chen, T. Tang, F. Cao, B.G. Cai, An integrated error-detecting method based on expert knowledge for GPS data points measured in Qinghai–Tibet Railway, Expert Systems with Applications, 39(2), 2012, pp. 2220-2226.

[5] D.W Chen, Y.S. Fu, B.G. Cai, Modeling and Algorithms of GPS Data Reduction for the Qinghai–Tibet Railway, IEEE Transactions on Intelligent Transport System, 11(3), 2010, pp. 753-758.

[6] G.G. Gao, B.G. Cai, Research on the Automatic Electronic Map Generation Algorithm for the Train Supervision System, Journal of the China Railway Society, 28(1), 2006, pp. 63-67.

[7] J.P. Zhang, J.Wang, Overview of principal curves, Chinese Journal of Computers, 26(2), 2003, pp.129-146.

[8] T. Hastie, Principal Curves and Surfaces: Stanford University doctoral dissertation (1984).

[9] T. Hastie, W. Stuetzle, Principal curves, Journal of the American Statistical Association, 84(406), 1988, pp. 502-516.

[10] J.D. Banfield, A.E. Raftery, Ice floe identification in satellite images using mathematical morphology and clustering about principal curves, Journal of the American Statistical Association, 87(417), 1992, pp. 7-16.

[11] B. Kegl, A. Krzyzak, A polygonal line algorithm for constructing principal curves, Proceedings of Neural Inform ation Processing Systems, 1999, pp. 501-507.
[12] J.J. Verbeek, N. Vlassis, B. Krose, A k-segments algorithm for finding Principal Curves, Pattern Recognition Letters, 23, 2002, pp. 1009-1017.
[13] D.W. Chen, B.C. Cai, T. Tang, An Information Fusion Algorithm for Multiple GPS Track Data, CA: 2008 Fourth International Conference on Natural Computation, 2008.
[14] J.P. Zhang, D.W. Chen, U. Kruger, Adaptive Constraint K-Segment Principal Curves for Intelligent Transportation Systems, IEEE Transactions on Intelligent Transportation Systems, 9(4), 2008, pp. 666-677.
[15] X.Z. Jia, D.W. Chen, Study on Information Fusion Algorithm for Multiple GPS Railway Tack Data, Journal of the China Railway Society, 33(9), 2011, PP.72-74.
[16] J.P. Zhang, X.D. Wang, U. Kruger, F.Y. Wang, Principal Curve Algorithms for Partitioning High-Dimensional Data Spaces, IEEE Transactions on Neural Networks, 22(3), 2011, pp. 367-380.
[17] C. Zhang, J.Y. Yang, D.C. Yan, S.Q. Yang, Y.T. Chen, Automated Breakpoint Generation for Debugging, Journal of Software, 8(7), 2013, pp. 603-616.
[18] P.F. Li, J.X. Huang, L.X. Ye, Y. Wang, Z.J. Li, D.W. Li, Directional Fuzzy Data Association Filter, Journal of Software, 7(10), 2012, pp. 2286-2293.
[19] C. J. Li, W. L. Jia, Y.Y. Yang, X. W, Adaptive Genetic Algorithm for Steady- State Operation Optimization in Natural Gas Networks, Journal of Software, 6(3), 2011, pp. 452-459.

Shiying Yang was born in 1989. She received the B.S. degree in Electronics and Information Engineering, Beijing Jiaotong University, China in 2012. She is working toward the M.S. degree with the Electronics and Information Engineering, Beijing Jiaotong University. Her research interests include machine learning, principle curve.

Dewang Chen was born in 1976. He received the B.S. degree in Mechanical and Electrical Engineering and the M.S. degree in Control and Automation from Harbin Engineering University in 1998 and 2000, respectively, and the Ph.D. degree in Control Theory and Control Engineering from the Institute of Automation, Chinese Academy of Sciences in 2003. He is a Professor with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University since 2011. His current research interests include intelligent control, machine learning, soft computing, optimization and their applications in intelligent transportation systems and railway systems.

Xiangyu Zeng was born in 1989. He received the B.S. in Electronics and Information Engineering, Beijing Jiaotong University, China in 2011. He is working toward the M.S. degree with the Electronics and Information Engineering, Beijing Jiaotong University. His research interests include machine learning, data mining.

Peter Pudney is a Senior Research Fellow in the Centre for Industrial and Applied Mathematics and the Institute for Sustainable Systems and Technologies at the University of South Australia. He received the B.S. degree and the M.S. degree in Applied Science in Computer Studies from South Australian Institute of Technology, and the Ph.D. degree in Mathematics from University of South Australia. He is currently working on projects related to railway train and crew scheduling, as well as helping with trials of the Freightmiser driver advice system by several railways around the world.