From Principal Curves to Granular Principal Curves

Hongyun Zhang, Witold Pedrycz, *Fellow, IEEE,* Duoqian Miao, and Zhihua Wei

***Abstract*—Principal curves arising as an essential construct in dimensionality reduction and data analysis have recently attracted much attention from theoretical as well as practical perspective. In many real-world situations, however, the efficiency of existing principal curves algorithms is often arguable, in particular when dealing with massive data owing to the associated high computational complexity. A certain drawback of these constructs stems from the fact that in several applications principal curves cannot fully capture some essential problem-oriented facets of the data dealing with width, aspect ratio, width change, etc. Information granulation is a powerful tool supporting processing and interpreting massive data. In this paper, invoking the underlying ideas of information granulation, we propose a granular principal curves approach, regarded as an extension of principal curves algorithms, to improve efficiency and achieve a sound accuracy–efficiency tradeoff. First, large amounts of numerical data are granulated into *C* intervals—information granules developed with the use of fuzzy C-means clustering and the two criteria of information granulation, which significantly reduce the amount of data to be processed at the later phase of the overall design. Granular principal curves are then constructed by determining the upper and the lower bounds of the interval data. Finally, we develop an objective function using the criteria of information confidence and specificity to evaluate the granular output formed by the principal curves. We also optimize the granular principal curves by adjusting the level of information granularity (the number of clusters), which is realized with the aid of the particle swarm optimization. A number of numeric studies completed for synthetic and real-world datasets provide a useful quantifiable insight into the effectiveness of the proposed algorithm.**

***Index Terms*—Fuzzy C-means (FCM) clustering, granular principal curves, interval data, multiple-criteria objective function, particle swarm optimization (PSO).**

Manuscript received August 27, 2012; revised April 2, 2013 and June 9, 2013; accepted June 11, 2013. Date of publication August 26, 2013; date of current version May 13, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61075056, Grant 61273304, Grant 61175054, and Grant 61202170, and the Fundamental Research Funds for the Central Universities. This paper was recommended by Associate Editor J. Basak.

H. Zhang, D. Miao, and Z. Wei are with the Department of Computer Science and Technology and the Key Laboratory of Embedded System and Service Computing, Ministry of Education of China, Tongji University, Shanghai 201804, China (e-mail: zhanghongyun@tongji.edu.cn; miaoduoqian@163.com; zhihua_wei@tongji.edu.cn).

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada, with the Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Kingdom of Saudi Arabia, and with the Systems Research Institute, Polish Academy of Sciences, Warsaw 00-956, Poland (e-mail: wpedrycz@ualberta.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2270294

I. Introduction

PRINCIPAL curves—a nonlinear generalization of the first principal component analysis [1]—have gained popularity as an efficient tool of data analysis, understanding, and visualizing data structures. The original version of the principal curves method proposed by Hastie and Stuetzle (hereafter HS) in 1989 [2] has undergone significant changes and resulted in a variety of modifications. The developments are numerous; they address a number of various issues, e.g., bias of the HS principal curves algorithm [3], [16] and [17], convergence of the HS algorithm [18], [19], parameter selection for principal curves [20], to name just a few interesting pursuits [10], [21]–[24]. Principal curves have been found to be an important method to summarize information residing in experimental data. Considerable work has been reported on applications of principal curves to various problems such as, e.g., shape detection [3], speech recognition [4], image skeletonization [5], [6], feature extraction, bill recognition [7]–[9], [11], intelligent transportation analysis [12], [13], high-dimensional data partitioning [14], and regression analysis [15]. However, with the rapid development of the Internet and information systems, we often need to handle massive data, and here the efficiency of existing principal curves algorithm (PC algorithm)becomes lower when dealing with massive data owing to the associated high computational complexity. Furthermore, principal curve is a single numeric construct (curve) and as such, it cannot fully capture the essence of the features hidden in data. For instance, in image description, one would be interested in capturing essential features of objects concerning width, aspect ratio, width change, etc. Granular Computing (GrC) has emerged as a new way to model ways of problem solving by concentrating on forming information granules (IGs) and realizing processing at this higher, more abstract level [25]–[41]. The theory, methodology, and algorithmic developments arising within the setting of GrC offer a variety of ways of dealing with data processing (see, e.g., [42]). Through information granulation, similar objects are gathered to form IGs. In this way, vast amounts of numeric data are transformed into far fewer IGs. Following this way, the computing time can be significantly reduced. Moreover, information granulation can hide (mask) some unnecessarily details, and improve robustness capabilities of the ensuing processing. Meanwhile, in many situations, approximate (granular) results are of interest when high efficiency and interpretability become required. The proposed granular principal curves algorithm (referred to as the GPC algorithm) forms an extension of the PC algorithm, and helps improve efficiency and achieve a sound accuracy–efficiency tradeoff. There is another useful motivating factor behind the concept of granular principal curves that is useful from the perspective of applications. In principal curves, we arrive at single (numeric) curve capturing the nature of data. In contrast, in granular principal curves, we do not form a single curve but build a granular construct, say a region, which reflects the nature of the data. In situations where we track a movement of a collection of closely linked objects (say, a flock of robots, soldiers, etc.), it is more legitimate to portray this phenomenon by a thick granular line rather than a single line.

In the design process, we first partition a large amount of numeric data into C clusters (IG) by running the fuzzy C-means (FCM) clustering algorithm, and characterize the obtained granules as C granular (interval) entities. Granular data obtained in this manner are expressed in the form of intervals. Afterward, we construct the granular principal curves by making use of the upper and the lower bounds of these interval data. Finally, we design a multiple-criteria objective function based on the criteria of confidence and specificity to evaluate the granular output of principal curves, and optimize the curves by adjusting the level of information granularity (the number of clusters), which are realized by particle swarm optimization (PSO).

The remainder of this paper is structured as follows. Section II provides a brief overview of the concept of principal curves. Next, the design process of granular principal curves is elaborated on with the four main features being stressed, namely 1) formation of the granular data; 2) construction of granular principal curves; 3) design of multiple-criteria objective function; and 4) choice of the optimal level of information granularity. Experimental studies are covered to evaluate and analyze the performance of the proposed algorithm (see Section IV). Finally, Section V presents some conclusions.

II. Principal Curves—Some Preliminaries

In this section, we review some generic concepts, main ideas, and highlight some applications of principal curves. More detailed description can be found in [2]–[24].

A. Concept

Hastie and Stuetzle generalized the self-consistent property of principal components and introduced the notion of principal curves in 1989. To highlight the properties of principal curves, we recall some basic definitions and concepts of principal curves.

Definition 2.1 [18]: A 1-D curve in R^d is a continuous function, where $f : \Lambda \longmapsto R^d$, where $\Lambda = [a, b] \subset R^1$.

The curve f can be considered as a vector of d-dimensional functions of a single variable $\lambda \in [a, b]$, $f(\lambda) = (f_1(\lambda), ..., f_d(\lambda))$, where $f(\lambda) = (f_1(\lambda), ..., f_d(\lambda))$ are called the coordinate functions. Here, λ represents the arc-length parameter along the curve in this definition.

Definition 2.2 [2]: For any $X \in R^d$, the corresponding projection index $\lambda_f(X)$ in the curve $f(\lambda)$ is defined as

$$\lambda_f(X) = \sup\{\lambda : \|X - f(\lambda)\| = \inf_{\tau} \|X - f(\tau)\|\} \qquad (1)$$

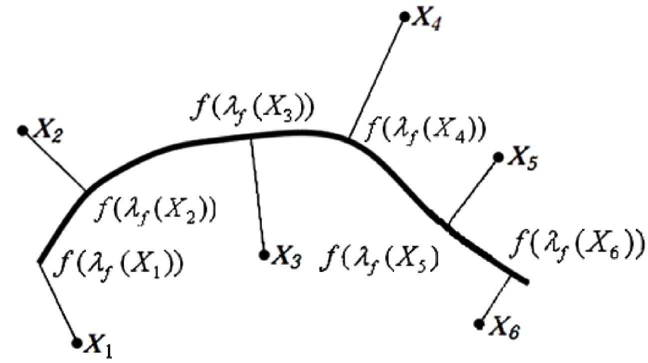

Fig. 1. Projecting points on a curve.

where $f(\lambda) = (f_1(\lambda), ..., f_d(\lambda))$ denotes a smooth curve in R^d parameterized by $\lambda \in R^1$. The projection index $\lambda_f(X)$ of X is the largest parameter value of λ for which $f(\lambda)$ is closest to X. Here, $\|.\|$ denotes the Euclidean norm in R^d. Accordingly, the projection point of X on f is $f(\lambda_f(X))$ (see Fig. 1).

Definition 2.3 [2]: Given curve $f(\lambda)$, $\lambda \in R^1$, the arc-length from λ_1 to λ_n is given by

$$l = \int_{\lambda_1}^{\lambda_n} \|f^{'}(\lambda)\| = \sum_{i=1}^{n} \|f(\lambda_{i+1}) - f(\lambda_i)\| \qquad (2)$$

where $f^{'}(\lambda)$ is tangent to the curve at λ which is sometimes called the velocity vector at λ and n is the number of data.

Definition 2.4 [2] (The HSPC): The smooth curve $f(\lambda)$ is a principal curve if and only if:

1) $f(\lambda)$ does not intersect itself;
2) $f(\lambda)$ has finite length inside any bounded subset of R^d;
3) $f(\lambda)$ is self-consistent, that is

$$f(\lambda) = E[X \left| \lambda_f(X) = \lambda\right], \forall \lambda \in R, X \in R^d. \qquad (3)$$

A principal curve is self-consistent meaning that each point on the curve is the condition expectation of those points that project to this point. Thus, passing through the middle of the distribution and providing a good 1-D nonlinear description of the data, the principal curve is a smooth self-consistent curve. Here, a 1-D curve in a d-dimensional space is a vector f of d functions indexed by a single variable λ. The parameter λ is the arc length computed along the curve. Principal curves form a nonlinear generalization of the principal component analysis. The theoretical fundamentals behind these curves relate to a low-dimensional nonlinear manifold embedded in a high-dimensional space. Fig. 2 shows a first principal component line and a principal curve. Compared with the corresponding principal component, two evident advantages of a principal curve are visible. First, a principal curve can retain more information about the data. Second, it follows the data more closely and captures the geometry of the data more accurately.

In the sequel, on the basis of the definition of the HSPC, a variety of other definitions of principal curves have been given in [16]–[18] and [21] and more recently in [10], [20], [22], and [23], which differ essentially in how the center of the distribution is determined.

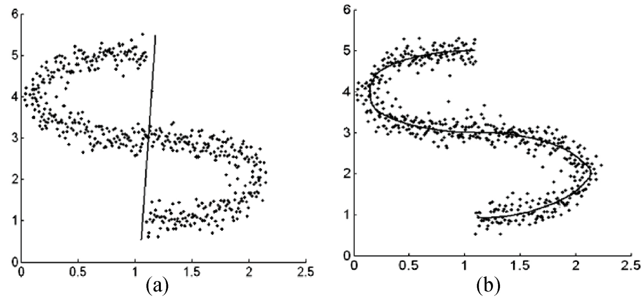

Fig. 2. Comparison between first principal component and principal curve. (a) First principal component. (b) Principal curve.

B. Applications

A great number of applications of principal curves have been reported in the literature. The first real-world application of principal curves was a part of the Stanford Linear Collider project [2]. Banfield and Raftery described an almost fully automatic method for identifying ice floes and their outlines in satellite images. The core procedure of the method uses a closed principal curve to estimate the floe outlines. Furthermore, they extended existing clustering methods by allowing groups of data points to be centered about principal curves rather than points or lines [3]. Stanford and Raftery further extended principal curve clustering and proposed an automatic method for extracting curvilinear features of simulated and real-world data [7]. Principal curves are applied to model the short time spectrum of speech signals. First, high-dimensional data points representing diphones are projected to a two-dimensional subspace. Each diphone is then modeled by a principal curve [4]. The method of principal curves is used to describe and analyze the interaction among freeway traffic-stream variables and their joint behaviors without utilizing conventional assumptions made with regard to the functional forms of interactions [12], [13]. The methods of principal curves are often used to extract image skeletons and select recognition features such as character, tubular objects, and fingerprint [5], [6], [8], and [11].

III. GPC Algorithm

Assume that a set of numeric data points $X_N = \{x_1, ..., x_N\} \subset R^d$ is given. In the GPC algorithm, we follow the strategy outlined in the following.

1) Divide the data into C clusters by running the FCM algorithm;
2) Form C interval (granular) data by running the principle of justifiable granularity. This result comes in the form of d-dimensional hyper-rectangles (boxes);
3) Select $2^{d-1} \times C$ vertices to form the upper and the lower bounds of the dataset, where a d-dimensional interval data are represented by a hyper-rectangle;
4) Extract upper principal curves f^+ following the upper dataset by using PL principal curves proposed by Kégl. The lower curves f^- are attained by the same method;
5) Check if f^+ and f^- satisfy the convergence condition. If satisfied, output f^+ and f^-, else adjust the level of information granularity (the value of C) and go to step 1).

In the following sections, we elaborate on these steps in more detail.

A. Formation of the Granular Data

Information granulation, introduced by Zadeh, offers a new perspective at problem representation and problem solving in computer science, logic, and philosophy [30]. Information granulation is the process of forming meaningful pieces of information, called IGs, that are regarded as entities that embrace collections of individual elements (e.g., numerical data) that exhibit some functional commonalities or closeness. The principle of justifiable granularity, proposed by Pedrycz [42]–[44], offers a way of constructing legitimate (justifiable) IGs.

In many situations, when describing a problem, we tend to shy away from numbers, and instead use aggregates (IGs) to look at the problem. If we gather similar objects into IGs according to the principle of justifiable granularity, then a large amount of data will become transformed into a far fewer granules. This way, we can reduce the computing time and achieve a sound accuracy–efficiency tradeoff.

1) *Construction of IGs:* For numeric data, there are two effective approaches to construct the IGs, namely information granulation based on neighborhood relations [27], [28] and clustering-based information granulation [25], [26]. Neighborhood relations are often used to generate a family of neighborhood granules from the dataset, which constitute a covering of the dataset. The size of neighborhood granules depends on a certain threshold δ. Clustering algorithms divide N numerical data into K granules, which constitute a partition of the dataset. A partition consists of disjoint granules of the dataset, and a covering consists of overlapping granules. In this paper, we choose the method of clustering-based information granulation; however, the use of the granulation method based on neighborhood relations could be a viable alternative. Fuzzy clustering, especially FCM [25], [26], as the extension of C-means, is one of the commonly encountering algorithms being used in the formation of IGs. Since the concepts and algorithmic aspects of the FCM algorithm are well reported in the literature, we only offer here a concise summary. Next, we elaborate on the process of granulation of numeric data. Original numerical data $X_N = \{x_1, ..., x_N\} \subset R^d$ are divided into C clusters $(1 < C < N)$; we construct a partition matrix $U = [u_{ik}], i = 1, 2, ..., C; k = 1, 2, ..., N$ with C being the number of clusters and N denoting the number of data. A set of prototypes of the clusters is given as $\{v_1, v_2, ..., v_C\}$. Every cluster and its prototype, in particular, $v_1(i = 1, 2, ..., C)$, represents an IG.

2) *Description of IGs:* Another issue arising when analyzing and processing these granules is about their formal description. In this paper, we utilize interval-valued data to describe IGs represented by hyper-rectangles (boxes). IG usually contains more than a single object. We use the principle of justifiable granularity [42]–[44] to form the upper boundary and the low boundary of a granule (cluster) to capture the numeric data embraced by the granule. The resulting information

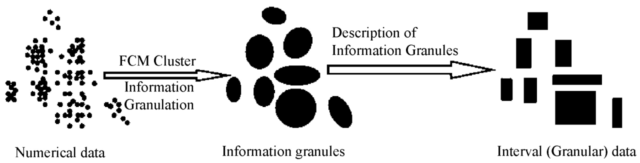

Fig. 3. Main idea of the granular data formation.

granular are described in the form of an interval (($[\underline{x}, \overline{x}]$)). The details of the method are outlined as follows.

Let the numeric center (representative) of the granule v_i be denoted by m_{ij}. The granule v_i is transformed into an interval data $X_i = ([\underline{x}_{ij}, \overline{x}_{ij}]), j = 1, ..., d$ following the principle of justifiable granularity. Following this principle, a construction of interval IG is guided by two criteria, namely, the coverage criterion and the specificity criterion. The coverage criterion is quantified by summing up the membership grades of data falling within the bounds of Ω (viz., the interval $[m_{ij}, \overline{x}_{ij}]$), namely, $\sum_{x_k \in \Omega} u_{ik}$. The specificity criterion is articulated in terms of the length of the resulting interval. In general, any continuous decreasing function f of the length of the interval can serve as a sound indicator of the specificity of the IG. Among possible design alternatives regarding the choice of functions f, we consider it in the following form $f(|m_{ij} - \overline{x}_{ij}|) = \exp^{(-\alpha|m_{ij} - \overline{x}_{ij}|)}$, where α is a positive parameter supplying some flexibility when optimizing the interval Ω. As these two criteria are in conflict, we consider a maximization of the composite multiplicative index, that is

$$E(\overline{x}_{ij}) = (\sum_{x_k \in \Omega_1} u_{ik}) \times \exp^{(-\alpha|m_{ij} - \overline{x}_{ij}|)} \tag{4}$$

$$E(\underline{x}_{ij}) = (\sum_{x_k \in \Omega_2} u_{ik}) \times \exp^{(-\alpha|m_{ij} - \underline{x}_{ij}|)} \tag{5}$$

where $\Omega_1 = \{x_k \in v_i | m_{ij} \leq x_k \leq \overline{x}_{ij}\}$ and $\Omega_2 = \{x_k \in v_i | \underline{x}_{ij} \leq x_k \leq m_{ij}\}$. We obtain the optimal upper bound $\overline{x}_{ij}$ by maximizing the value of $E(\overline{x}_{ij})$. In the same way, the optimal lower bound $\underline{x}_{ij}$ is constructed.

Fig. 3 illustrates a generic idea of the proposed methodology. First, a large collection of similar objects is arranged together (by running FCM) to form a few IGs. Next, following the principle of justifiable granularity, we form interval IGs ($[\underline{x}, \overline{x}]$) and construct hyperboxes (hyper-rectangles) to fully describe the granules. A suitable level of granularity can be determined. A more detailed discussion on the determination of the optimal level of granularity is provided in Section III-D.

B. Construction of Granular Principal Curves

In this section, we first introduce the notion of the upper and lower bounds of granular data and discuss how to construct such granular entities. Then, granular principal curves are extracted on the basis of the upper and lower bounds of these data. We also elaborate in detail on the concepts behind the construction of the granular principal curves.

1) *Generation of Upper and Lower Data:* Once the N numeric data have been granulated resulting in C interval (granular) data represented by hyper-rectangles, we obtain $2^d \times C$ vertices with d being the dimensionality of the data space. We divide these vertices into two subsets with a similar number of elements in each part to describe the interval data. These are referred to as the upper and lower data. For every hyper-rectangle, if data distributions have both maximum and minimum only in the direction of a certain coordinate axis, upper data are composed of 2^{d-1} vertices whose coordinate values are higher in the dimension; otherwise, upper data are composed of 2^{d-1} vertices which are farther from the central point of data distributions. The remaining vertices form the lower data. For instance, for the sine wave exist simultaneously maximum (convex point) and minimum (concave point) only along the direction of y-axis, so we choose two vertices from every rectangle as upper data, whose y-coordinate values are higher [see Fig. 4(a)]. For a circle, upper data are composed of $2 \times C$ vertices, which are more distant from the central point of data distributions [see Fig. 4(b)]. The lower and upper data are formed as follows. Let $M = (m_1, ..., m_d)$ be the center coordinate of N numeric data, and $V_i = (v_{i1}, ..., v_{i2^d})$ be the set of 2^d vertices of the ith interval data. Suppose that x_{ui} and x_{li} are the data points whose coordinates are the maximum and minimum observed along the ith coordinate. If the number of x_{ui} (or x_{li}) is greater than one, we choose the point whose the Euclidean distance between x_{ui} (or x_{li}) and the center point M is the shortest. θ_{ui} is the included angle between the line of Mx_{ui} and the ith coordinate axis. Likewise, θ_{li} is the included angle between the line of Mx_{li} and the ith coordinate axis. Let $\alpha_i = \theta_{ui} + \theta_{li}$. If $\theta_{uj} < \pi/18$, $\theta_{lj} < \pi/18$ and $\alpha_i > \pi/6$, $j = 1, ..., d$, $j \neq i$, then 2^{d-1} vertices whose coordinates are higher in the ith coordinate are selected as the upper data and the remaining vertices become the lower data; otherwise, for every interval data $X_i (i = 1, ..., C)$, let $d(M, v_{ij})$ be the Euclidean distance between v_{ij} and the center point $M (j = 1, ..., 2^d)$. We sort the vertices in a descending order with respect to the distance $d(M, v_{ij})$. The previous 2^{d-1} vertices are chosen as the upper data and the remaining 2^{d-1} vertices form the lower data.

2) *Extraction of Granular Principal Curve:* We extract the upper principal curves f^+ from the upper data and the lower principal curves f^- from the lower data based on the existing PC algorithms. Here, the PL principal curves algorithm proposed by Kégl [18] is selected to extract f^+ and f^-. Then, we assess whether the granular principal curves (f^+ and f^-) meet the convergence criterion meaning that a local maximum of the objective function has been achieved; if this is the case, we produce f^+ and f^-; otherwise, we adjust the level of information granularity (viz., the number of clusters) and proceed with the process as described in Section III-A.

To explain the process of granular principal curves extraction, we generate a 2-D synthetic data affected by a high level of noise ($\sigma = 0.6$) and composed of 1256 data points that are distributed along a certain generating curve [see Fig. 5(a)]. In this experiment, the data are granulated by forming 45 interval data (C= 45, $\alpha = 1$) according to the method described in Section III-A. The interval data are

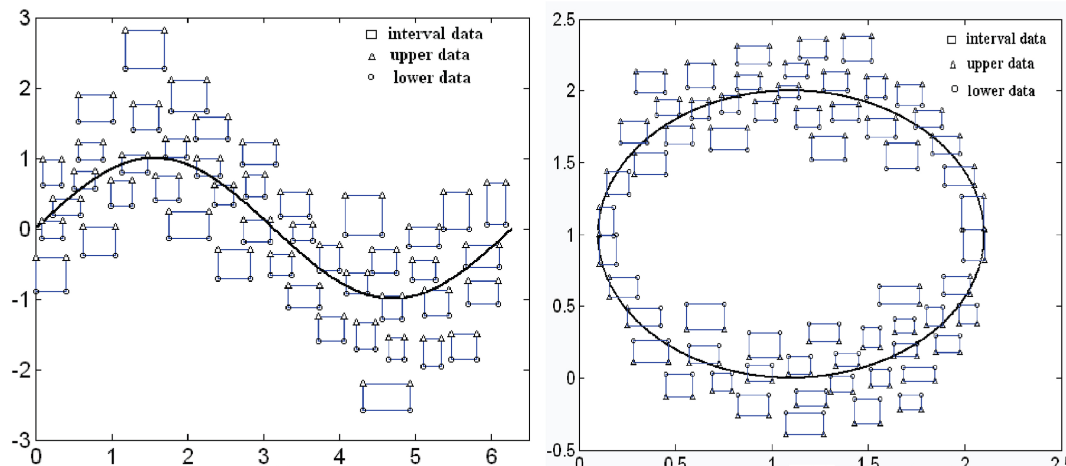

Fig. 4. Formation of upper and lower data. (a) Upper and lower data obtained from sine wave data. (b) Upper and lower data obtained for circle.

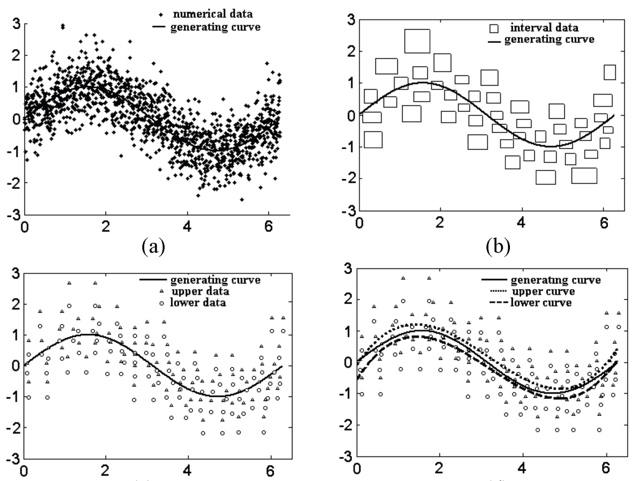

Fig. 5. Results obtained at various stages in granular principal curves extraction when applied to noisy synthetic data. (a) Numerical data. (b) Interval (Granular) data. (c) Upper and lower data. (d) Granular principal curves.

represented by a collection of rectangles [see Fig. 5(b)]. The upper and lower data are formed as described previously. The upper data are composed of triangular points and the lower data are composed of circular points [see Fig. 5(c)]. Finally, the granular principal curves are formed on the basis of the upper and lower data by following the PL principal curves algorithm [18] [see Fig. 5(d)]. The experimental results produced at various development stages of the algorithm are presented in Fig. 5.

Note that in the algorithm the granular output of the principal curves has to be evaluated in terms of an objective function. The algorithm terminates when objective function achieves its local optimum which is accomplished by adjusting the level of information granularity (the value of C that is the number of clusters). The design of the objective function will be discussed in detail in Section III-C.

C. Design of the Objective Function

As we are concerned with the granular outputs of the principal curves, which have to be evaluated with regard to the target, two criteria (performance indexes) are worth considering. The first one looks at the quantification of the concept of confidence—an extent to which the target values are captured (represented by the curve). If the deviations between the granular principal curves (f^+ and f^-) and the generating curve f^g are lower than a given threshold ε, then we gain a high confidence about the granular output formed by the GPC algorithm. Another criterion is focused on expressing a level of specificity of the granular curves produced by the GPC algorithm. This criterion is articulated in terms of the width of the interval of the granular output.

Let us assume that for the purpose of the evaluation, we consider some data F coming in the form of the pairs $(X_P, f^g(\lambda))$, $X_P = \{x_1, ..., x_n\}$, $P = 1, ..., M$. Here, M is the number of testing sets randomly generated along the curve $f^g(\lambda)$ by the commonly used additive Gaussian noise. Let $f^-_{X_P}$ and $f^+_{X_P}$ be the lower and upper principal curves, respectively, being estimated for the testing set X_P, respectively.

Furthermore, let $D_{f^-_{X_P}}$ be the deviation between $f^-_{X_P}$ and the generating curve f^g. $D_{f^+_{X_P}}$ denotes the deviation between $f^+_{X_P}$ and the generating curve f^g. If $D_{f^-_{X_P}}$ and $D_{f^+_{X_P}}$ are less than a given threshold ε, then the granular output formed by the GPC algorithm is deemed acceptable. $D_{f^-_{X_P}}$ is defined as follows:

$$D_{f^-_{X_P}} = \left| 1 - \frac{\frac{1}{n}\sum_{i=1}^{n} \triangle(x_i, f^g)}{\frac{1}{n}\sum_{i=1}^{n} \triangle(x_i, f^-_{X_P})} \right| \tag{6}$$

where $\triangle(x_i, f^g) = \min_{\lambda} \|x_i - f^g(\lambda)\|^2$ and $\triangle(x_i, f^-_{X_P}) = \min_{\lambda} \|x_i - f^-_{X_P}(\lambda)\|^2$. The smaller the value of $D_{f^-_{X_P}}$, the closer $f^-_{X_P}$ and f^g are. $D_{f^+_{X_P}}$ can be defined by the same way.

The confidence criterion is quantified by a certain index Q, expressed as follows:

$$Q = \text{card}\{(X_P, f^g(\lambda)) \in F | D_{f^-_{X_P}} < \varepsilon \wedge D_{f^+_{X_P}} < \varepsilon\}/M. \quad (7)$$

The specificity criterion can be expressed in terms of the average distance (width) between bounds of $f^-_{X_P}$ and $f^+_{X_P}$ produced by the GPC algorithm for the inputs coming from the dataset F. It is quantified by an index W expressed in the form

$$W = \sum_{P=1}^{M} \frac{1}{l(f^+_{X_P})} \int \min_t \| f^+_{X_P}(\lambda) - f^-_{X_P}(t) \| d\lambda / M \quad (8)$$

where $f^-_{X_P}$ and $f^+_{X_P}$ are parametrized by their arc length.

In the design of the stopping condition, we consider several design scenarios.

1) Use of the confidence criterion. This gives rise to a single-criterion objective function $V(C) = f_1(Q)$. Here, f_1 is a continuous increasing function of Q;
2) Use of the two criteria, viz., confidence and specificity. The higher the value of confidence (Q) is, the better granular outputs are, and the lower the value of specificity (W) is, the better granular outputs are. Therefore, these two criteria are very likely in conflict. In this case, a two-criteria objective function considers a maximization of the composite multiplicative index, that is, $V(C) = f_1(Q) * f_2(W)$. f_1 is continuous increasing function of Q, and f_2 is continuous nonincreasing function of W. Among many possible design alternatives regarding functions f_1 and f_2, we consider the following options: $f_1(Q) = Q$ and $f_2(W) = e^{-\beta W}$. Thus, the objective function is defined as $V(C) = Q * e^{-\beta W}$.

On the one hand, the higher the value of C (the number of clusters), the higher the value of $V(C)$. On the other hand, when C is increased beyond a certain threshold, the increment of $V(C)$ becomes very low. The stopping condition brings these two aspects together. The algorithm terminates when $V(C)$ is higher than some predetermined threshold ω_1, while the difference $\Delta V(C) = V_k(C) - V_{k-1}(C)$ is lower than some other threshold ω_2, $V_k(C)$ is the value obtained at the kth iteration.

The pseudocode supporting the calculations of the two-criteria objective function is outlined as follows.

Algorithm 1 Objective function calculation algorithm (**OFC** algorithm)

input: the massive dataset $X_N = \{x_1, ..., x_N\}$ and the number of cluster C.
Output: objective function value $V(C)$

Divide data X_N into C clusters by running the FCM algorithm
Let $(\mu_{i1}, ..., \mu_{id})$ be ith cluster center coordinate
For every cluster v_i **do**
  Compute the lower bound of the ith cluster $\underline{x}_{ij}$ by maximizing $E(\underline{x}_{ij})$
  Compute the upper bound of the ith cluster $\overline{x}_{ij}$ by maximizing $E(\overline{x}_{ij})$
  Form granular data $X_i = ([\underline{x}_{ij}, \overline{x}_{ij}])(i = 1, ..., C)$
end for
Compute the center coordinate $M = (m_1, ..., m_d)$ of X_N
Let vertices set of interval data X_i be $V_i = \{v_{i1}, ..., v_{i2^p}\}$
Let x_{ui} be the data points whose coordinate is maximum in the ith coordinate
Let x_{li} be the data points whose coordinate is minimum in the ith coordinate
Compute the included angle θ_{ui} between the line of Mx_{ui} and ith coordinate axis
Compute the included angle θ_{li} between the line of Mx_{li} and ith coordinate axis
Let $\alpha_i = \theta_{ui} + \theta_{li}$
if$(\alpha_i > \pi/6) \wedge (\theta_{lj} < \pi/18) \wedge (\theta_{lj} < \pi/18), (j \neq i)$ **then**
  Choose 2^{d-1} vertices whose coordinate are higher in the ith dimension as the upper data
  Choose remaining 2^{d-1} vertices as the lower data
else
  for every interval data X_i **do**
    for every vertex v_{ij} in V_i **do**
      Compute the Euclidean distance, $D(M, v_{ij}) = \|M - v_{ij}\|$
    end for
    Sort the vertices by $D(M, v_{ij})$ in descending order
    Choose previous 2^{d-1} vertices as the upper data
    Choose remaining 2^{d-1} vertices as the lower data
  end for
end if
Determine f^- from the lower dataset using PL algorithm
Determine f^+ from the upper dataset using PL algorithm
Compute coverage criterion Q by (7) and specificity criterion W by (8)
Compute $V(C) = Q * e^{-\beta W}$

D. Determination of the Optimal Level of Information Granularity Based on PSO

Two types of search strategies can be used to determine the optimal level of information granularity (number of clusters C): 1) enumeration and 2) heuristics-supported search. In the first case, the optimal level of information granularity is found by successively increasing the number of clusters and evaluating the obtained results. If the optimal number of clusters is C_{opt}, then the enumeration requires C_{opt} evaluations of the resulting construct—granular principal curve. This method is simple; however, the value of C_{opt} could be very high (often higher than 60), resulting in a substantial computing overhead.

Here, we adopt a heuristic search strategy to search the optimal level of information granularity. Among many available alternatives, we choose PSO [45]. There are several reasons behind this choice: 1) PSO, as a relatively new swarm intelligence-based heuristic global optimization technique, is easy to understand and implement. It requires a limited parameter tuning and exhibits robust global convergence; 2) in the experiments, since m particles search for the optimal solution

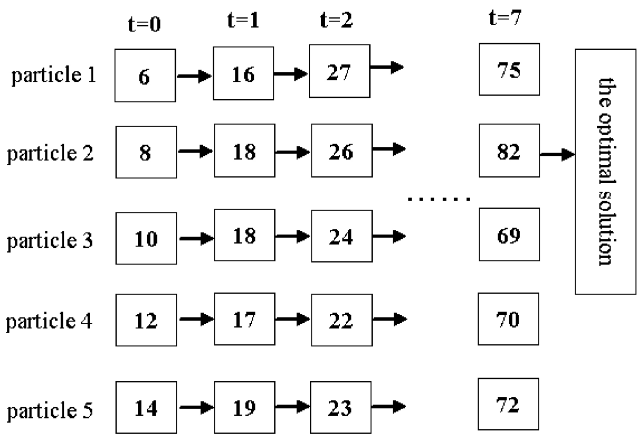

Fig. 6. Example of overall search process based on PSO.

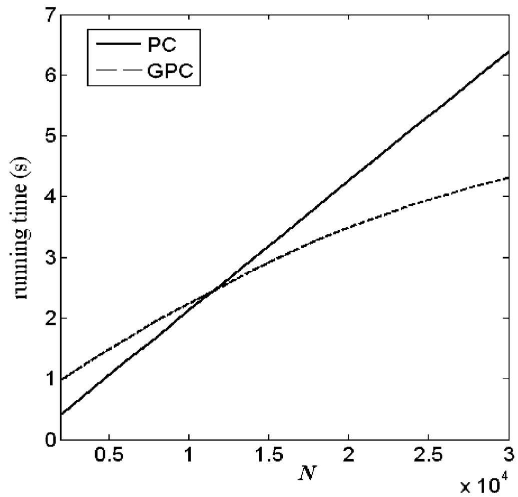

Fig. 7. Comparison of running time of PC and GPC algorithms for different data size N.

at the same time, we find that the optimal solution can be obtained after several iterations and the number of evaluation of PSO is substantially lower than C_{opt}. Furthermore, in our case, the computing overhead of the algorithm is reasonable. Let us briefly recall the PSO search process.

First, we initialize m particles corresponding with m different numbers of clusters and increments of cluster numbers as initial positions $s_i(0)$ and initial velocity $v_i(0)$, where $v_i(0)$ is a certain integer and not less than m. Next, each particle proceeds through the search space at a given velocity v that is dynamically modified on a basis of its own experience and the best local performance reported so far (p). The velocity is also affected by experience of other particles, thus resulting in the best global value, global best (p_g) [45]. Finally, if the convergence condition has been satisfied in the tth generation, then the position $s_j(t)$, namely p_g, corresponds to the optimal level of information granularity, that is, if $V(s_j(t)) > \omega_1$ and $V(s_j(t)) - V(s_j(t-1)) < \omega_2$, then we report the value of $s_j(t)$.

Note that the position s_i is an integer, meaning that the computed velocity v_i has to be rounded up in all generations. There are also some limits imposed on the velocities, say $v_{\max}$ and $v_{\min}$. Here, we set $v_{\min}$=1, $v_{\max}$=$C_{\max}$, where $C_{\max}$ is the maximal allowed number of clusters.

Let the number of particles (m) is equal to 5, and t be the number of generations. An example of the search process is illustrated in Fig. 6. In this example, the total number of evaluations of the fitness realized by the PSO is 35 ($m \times t$), while the direct enumeration results in 82 evaluations. In essence, the PSO is superior to the enumeration method.

E. Computational Complexity of the Approach

As earlier, let N be the number of numeric data points, d be the dimensionality of the data, C be the number of clusters, L_1 be the iteration number of the FCM, and L_2 be the iteration number of GPC. The GPC algorithm uses FCM to granulate the numeric data. After the numeric data are granulated, the number of data which needs be deal with by PC algorithms is reduce to 2^dC. The computational complexity of the FCM algorithm is $O(NCL_1)$ [25], and computational complexity of the state-of-the-art PC algorithm is $O(N^2d^3)$ [24]. Thus, the overall computational complexity of the GPC algorithm becomes $O(NCL_1L_2) + O((2^dC)^2L_2d^3)$.

When we are concerned with the massive data, C, L_1, and L_2 are very small relative to N, which can almost be ignored. If the dimensionality of massive data is not very high, then we have $2^dC \ll N$ and $CL_1L_2 \ll N$. We can draw a conclusion that $O(NCL_1L_2) + O((2^dC)^2L_2d^3) < O(N^2d^3)$. Therefore, the computational complexity of the GPC algorithm can be significantly decreased when dealing with large datasets. Fig. 7 shows the experimental results dealing with the running time of the PC and GPC algorithms reported for different sizes. Apparently, the GPC algorithm has lower computational complexity when data size N is higher than a certain critical value. Furthermore, with the growth of the data size, the positive effect of the GPC algorithm becomes more visible.

IV. Experimental Results and Their Analysis

In this section, we first present a series of experiments completed for several synthetic datasets to illustrate the proposed algorithm, observe the effect of C and α on experimental results, and identify the underlying relation among α, C, and W. Meanwhile, the results of principal curves and granular principal curves are compared. Then, the results produced by a set of experiments, using the real-world images, are reported to test the applicability of the proposed approach. Finally, the performance of the proposed algorithm and the impact of noise σ and numeric data sizes N on the performance of the algorithm are investigated, respectively.

A. Experimental Results—Synthetic Datasets

To compare the experimental results for the data before and after granulation, we observe the effect of the parameters α and C on the experimental results and identify the relationship among α, C, and W. Here, the method is used for several synthetic datasets. We generated datasets distributed along some curves and affecting the data by additive Gaussian noise. The noise has been independently imposed on different dimensions of the corresponding generating curves.

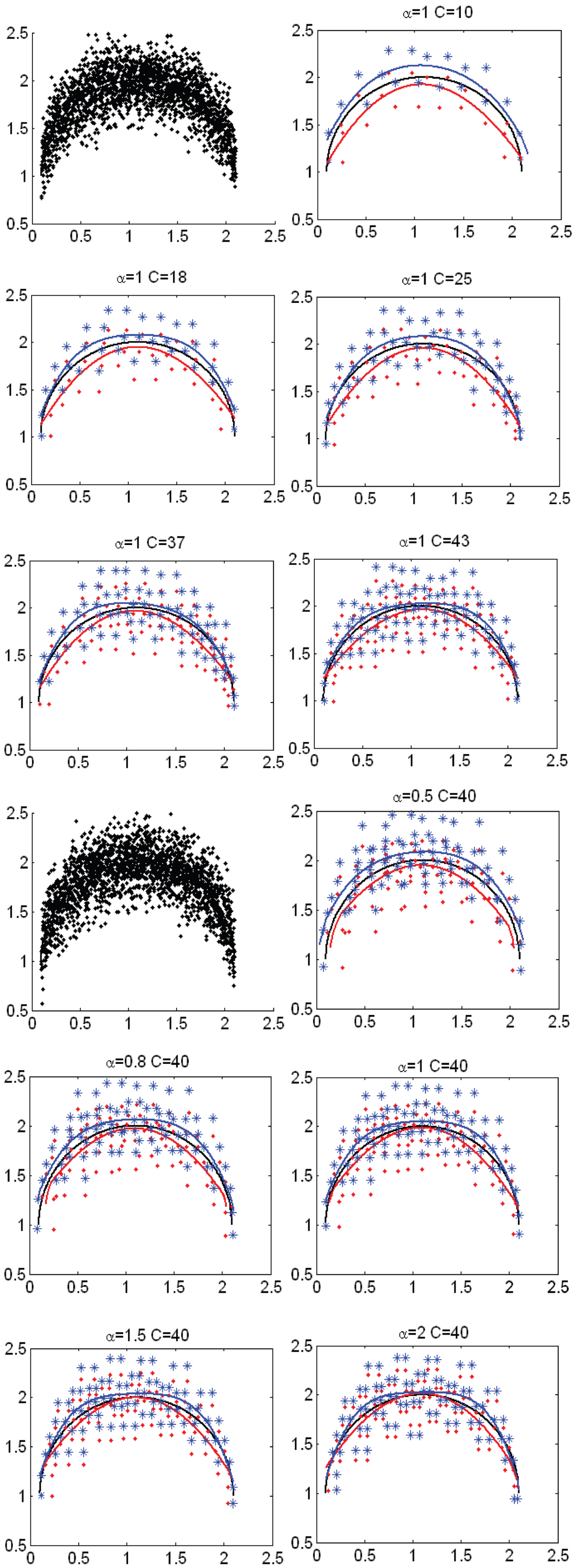

Fig. 8. Granular principal curves produced for half-circle data.

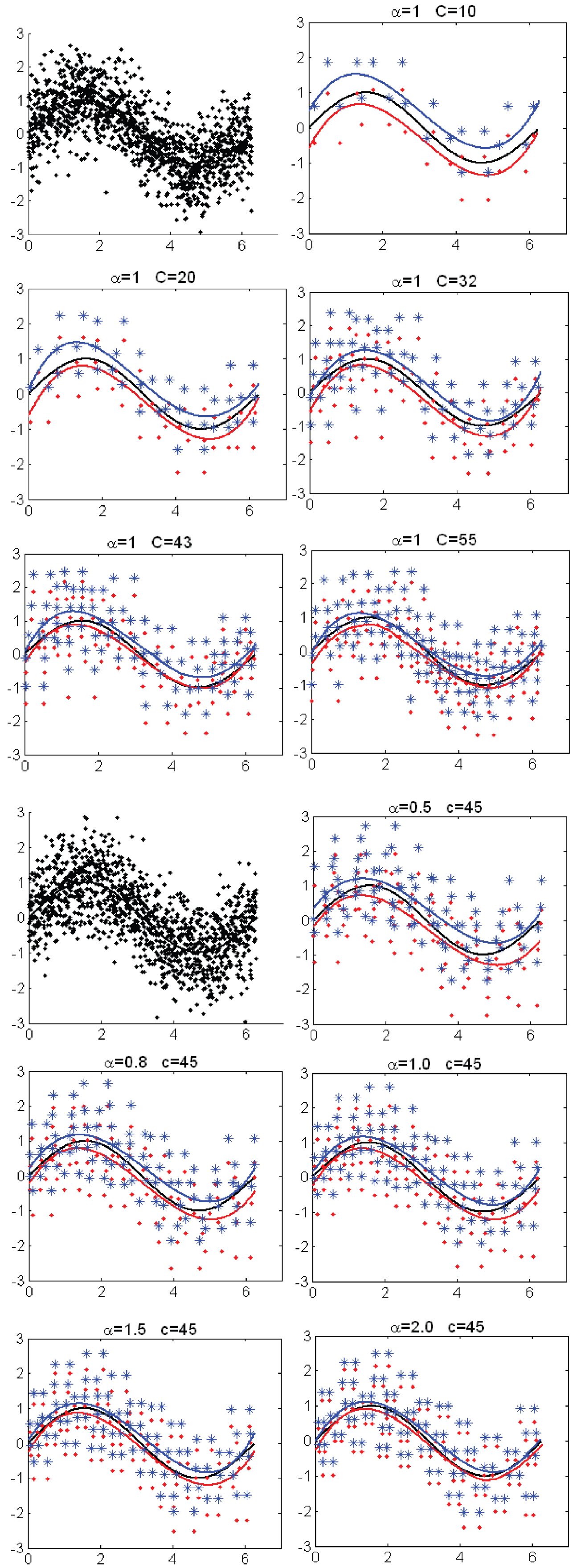

Fig. 9. Granular principal curves obtained for sine wave data.

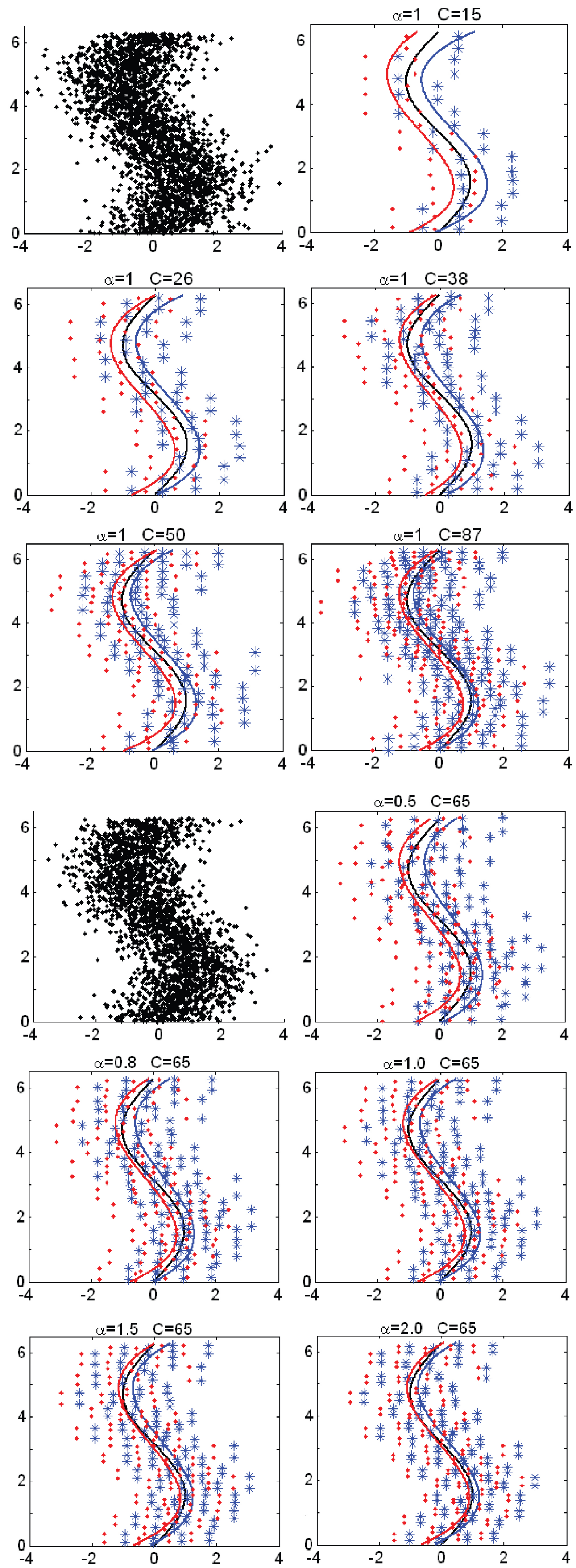

Fig. 10. Granular principal curves produced for s-shaped data.

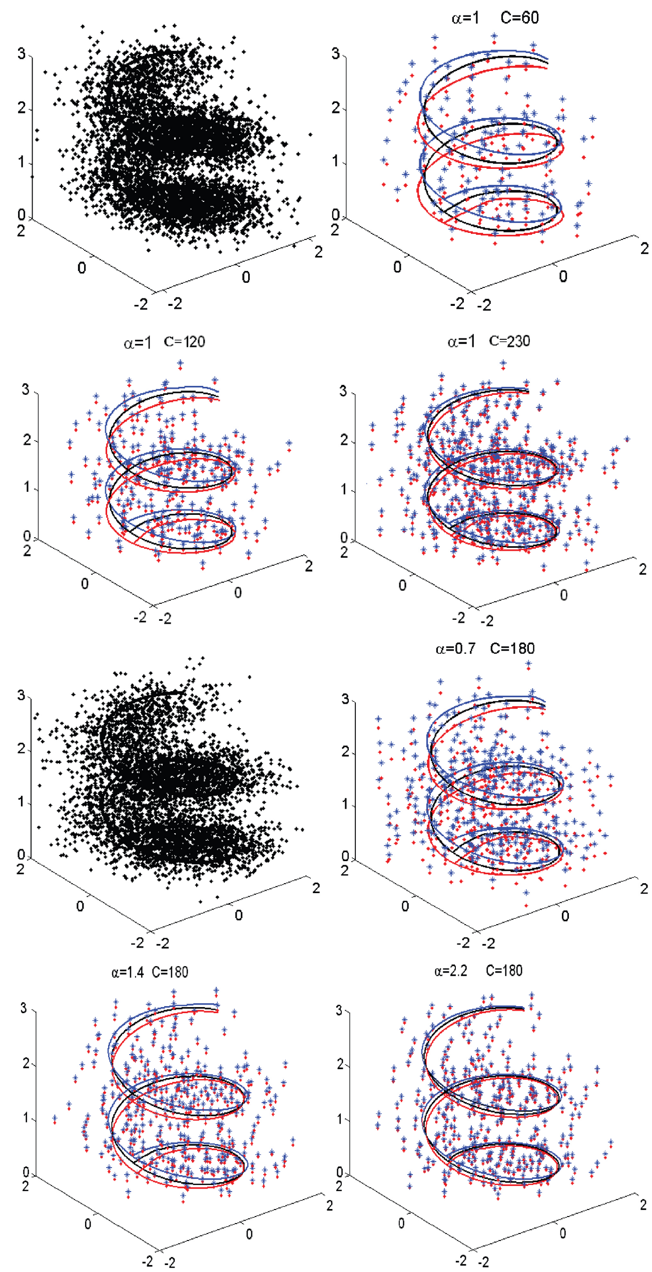

Fig. 11. Granular principal curves formed for 3-D helix-shaped data.

We constructed curves of various shapes, such as half-circle, sine wave, s-shaped, 3-D helix-shaped, etc. We consider several different scenarios: first, $n = 2000$ data points are generated by means of an underlying half-circle of radius $r = 1$, contaminated by noise of high amplitude (σ= 0.2).In the sequel, we consider a sine wave affected by significant noise (σ= 0.6) and composed of 1256 data points. Third, we investigate an s-shaped pattern again contaminated by noise of high amplitude (σ= 0.8), where 2512 data points were generated. Finally, we generate $n = 5160$ data points (σ= 0.6) distributed along a 3-D helix curve. In all experiments, we manually changed the values of the parameter α ranging from 0.2 to 2.0, and varied the cluster number C from 20 to 70. For each particular number of clusters C and the value of the parameter α, 100 random datasets were generated for

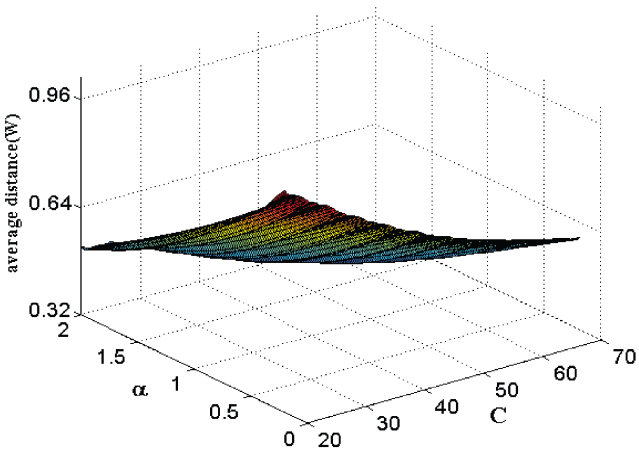

Fig. 12. Relationships among α, C, and W.

the four differently shaped data. We run the GPC algorithm on each dataset, and recorded the measurements (width) in each experiment. Finally, the results were then averaged over the experiments. The results of granular principal curves for different values of α and C are displayed in Figs. 8–11. The relationship among α, C, and the average distance (W) is plotted in Fig. 12. Results of comparative analysis produced by principal curves [18] and their granular counterparts are presented in Fig. 13.

As shown in Figs. 8–12, the higher the number of clusters C and the higher the value of α, the lower the average distance between the bounds of the produced granular (interval) principal curves and the better the result of granular (interval) principal curves. The reason is that the higher the number of cluster C is, the more accurately the geometry of the data becomes captured. On the other hand, the essential role of α is to calibrate an impact of the specificity criterion on the constructed granular data and higher values of α stress of increasing importance of this criterion.

Note that the parameters ε, β, ω_1, and ω_2 used in the stopping condition can affect the granular principal curves. Different values of parameters ε, β, ω_1, and ω_2 lead to different levels of information granularity. By adjusting the values of the parameters ε, β, ω_1, and ω_2, we can change the level of information granularity and roughness of granular outputs in any practical application. For the sake of simplicity, in all experiments, the parameters ε, β, ω_1, and ω_2 are set as constants, namely, ε= 0.186, β= 1, ω_1= 0.8, and ω_2= 0.017, respectively, These specific numeric values of these parameters result in a sound design option.

B. Experimental Results Obtained for Real-World Images

To evaluate the applicability of the proposed approach, the performance of the algorithm was tested on the two suites of real images. The first one deals with images of isolated handwritten characters captured by using a graphic's tablet. The second one is composed of the bilevel images of objects which were transformed from original images by bilevel thresholding. In this experiment, the GPL PC algorithm proposed by Kégl [19] is adopted. Some results obtained for

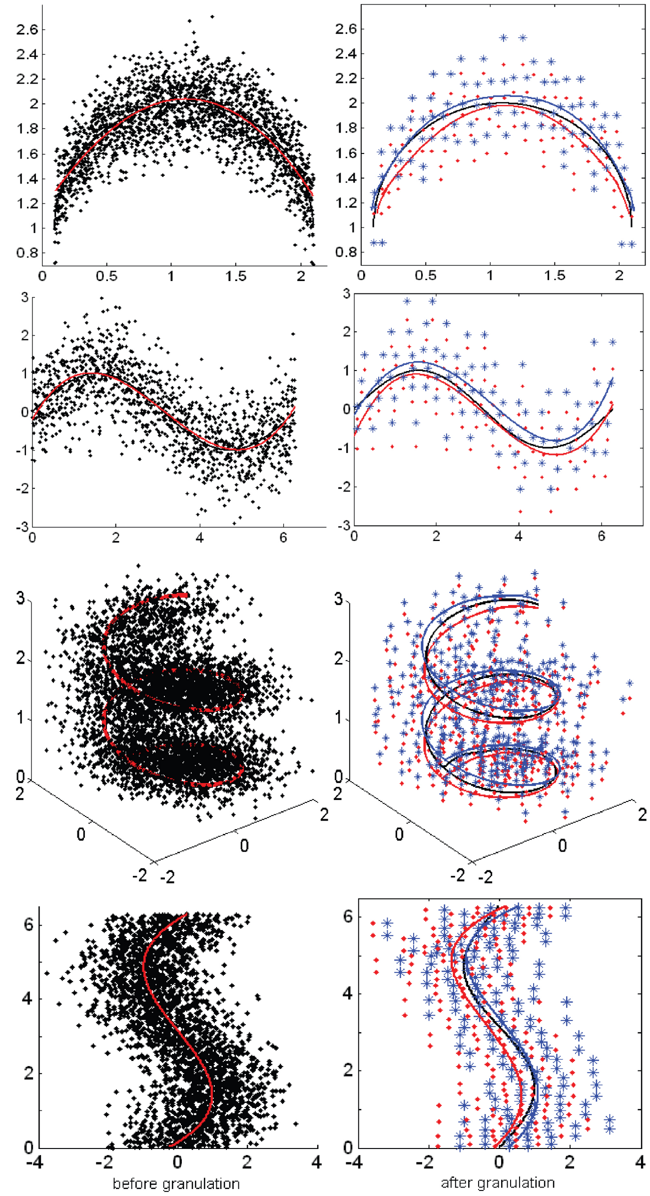

Fig. 13. Principal curves and granular principal curves constructed for four varying shape data: a comparative view.

these two sets of images are shown in Figs. 14 and 15. The experimental results demonstrate that the granular principal curves exhibit some advantages when compared with the corresponding principal curves. For example, granular principal curves are more closely positioned to the original image and in this way help capture the shape features of the original image more accurately. Furthermore, since the principal curve is a single curve, it cannot fully reveal and retain a shape information of different parts of the original image, such as width, aspect ratio, and width change. In contrast, the granular principal curves can.

C. Performance Analysis

We first compare the PC algorithm and GPC algorithm, that is, the implementation results before and after granulation, and evaluate in a quantitative manner the performance of

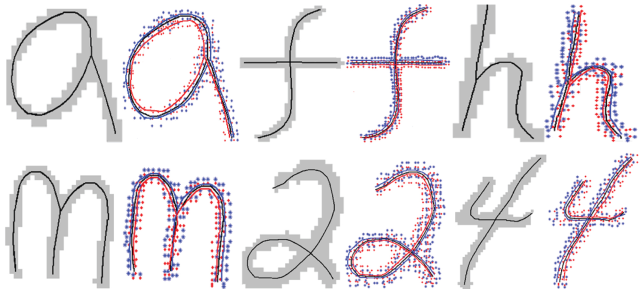

Fig. 14. Principal curves and granular principal curves produced for handwritten characters.

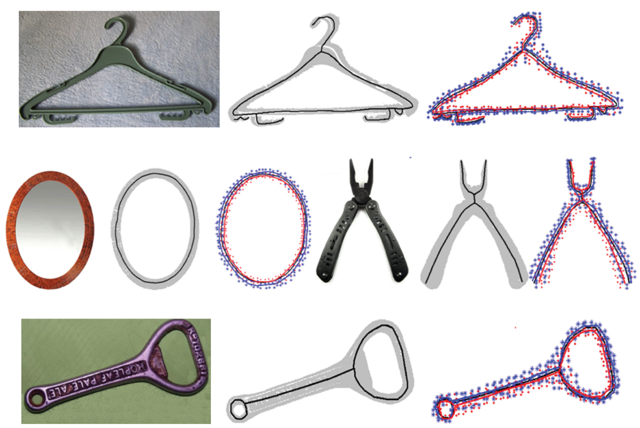

Fig. 15. Principal curves and granular principal curves obtained for a collection of objects.

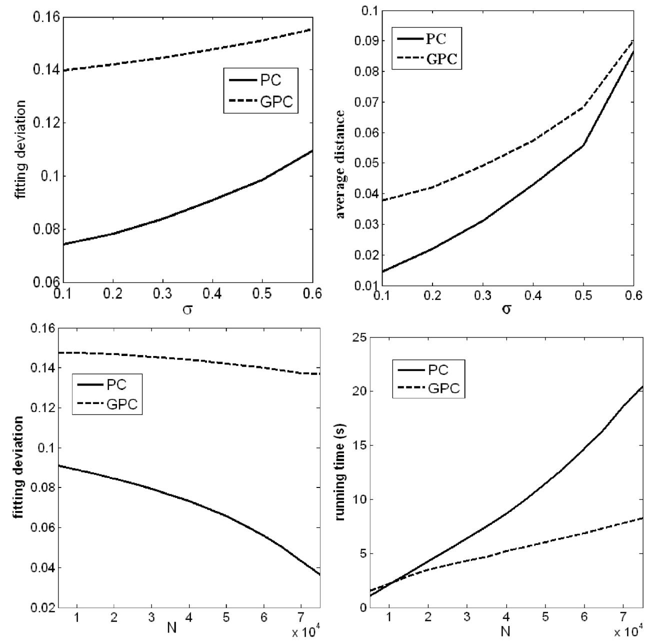

Fig. 16. Comparison of performance of PC and GPC algorithms.

GPC algorithm by using different noise intensities σ and data size N. Then, we investigate the impact of noise σ, cluster size C and numeric data size N on the performance of GPC algorithm, respectively. Finally, relevant evaluation and conclusions are summarized.

1) *Quantitative Comparison of Performance of PC and GPC Algorithms:* The PC algorithm [18] and GPC algorithm are quantitatively compared based on the three performance indexes. We consider deviation (D_f) of estimating curves and generating curves, the average distance (W_e), and running time (T) as the three indexes. In the GPC algorithm, the average distance is the distance between generating curves and the upper principal curves f^+ (or the lower principal curves f^-). In the PC algorithm [18], the average distance is the distance between generating curves and estimating curves. Concepts of fitting deviation and average distance have been introduced in detail in Section III-C. For the sake of simplicity, in all experiments, the parameter α is set to 1. We vary noise parameters and data sizes to evaluate the performance of GPC. On the one hand, using the same data sizes ($N = 5000$) and cluster numbers ($C = 80$), we change the deviation of noise σ by ranging it from 0.1 to 0.6. We investigate the effect of the noise σ on D_f and W_e in order to compare the robustness of PC and GPC. On the other hand, using the same data noise level σ= 0.4 and 15 different data sizes ranging from $N = 5000$ to $N = 75\,000$, we compare the efficiency of the PC algorithm and GPC algorithm in terms of D_f and the running time (T), and demonstrate an impact of data size N on D_f and T. Note that the parameters ε, β, ω_1, and ω_2 need be set according to the different levels of information granularity, and different ε, β, ω_1, and ω_2 may result in different values of D_f and T. Since we only consider the impact of the data size N on D_f and T, the four parameters of the GPC algorithm are set in the experiments as the constant values, say ε= 0.186, β= 1, ω_1= 0.8, and ω_2= 0.017. Hundred random datasets were generated for the four different datasets and the resulting D_f and T values were averaged over these experiments. The relationships between the performance of the algorithm, noise σ, and data size N are shown in Fig. 16. Through observations from experiments, GPC turns out to be more robust to alterations in the noise σ. The reason is that noise level can be obviously reduced during data granulation. Furthermore, we find that the data size (data density is more exact) mainly impact the running time (T). On the contrary, the data size (data density is more exact) has a slight effect on D_f of the granular (interval) principal curves. Anyway, the GPC method evidently outperforms the PC without granulation in terms of the running time (T) when dealing with large datasets.

2) *Impact of Noise, Cluster Size, and Data Size on the Performance of the GPC Algorithm Performance:* We investigate the impact of C, N, and σ on the objective function $V(C)$ and look carefully at the two aspects. On the one hand, we keep noise σ= 0.4 unchanged, observe the $V(C)$ by altering C and N. On the other hand, using the same data size $N = 2000$, we observe the behavior of $V(C)$ by altering the values of C and σ. In the first case, we consider 19 different numbers of clusters C ranging from 10 to 100, and five different data sizes ranging from $N = 1000$ to $N = 10\,000$. The second case is the one when we consider 19 different numbers of clusters C ranging from 10 to 100, and five different levels of noise ranging from σ= 0.1 to σ=0.6. For each particular values of C,

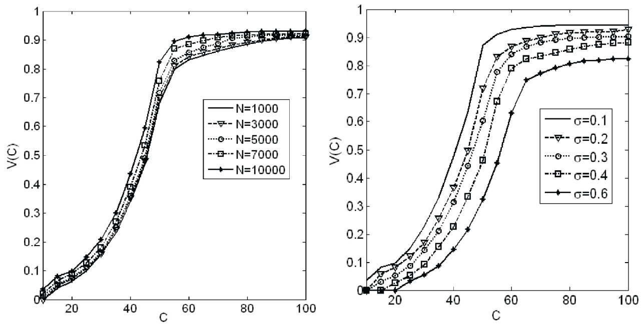

Fig. 17. Relationship between $V(C)$ and C reported for different data sizes N and noise variance σ.

N, and σ, 100 random datasets were generated for the three different datasets and the resulting values $V(C)$ were averaged over all experiments. The relationship between the objective function $V(C)$ and the numbers of clusters for different data sizes and noise levels is plotted in Fig. 17.

The values of the objective function increase as the number of clusters increases and the data size grows.The values are reduced when the variance of the noise increases. However, it is noticeable that the data size (data density to be more exact) has a slight impact on the objective function. To the contrary, the noise and cluster size can significantly change the values of the objective function. This is not surprising as granular principal curves are detected on the basis of granular data. Furthermore, the impact of data size on formation of granular data is much more limited than the one that associates with the noise component.

V. Conclusion

Traditional PC algorithms tend to lose their efficiency when dealing with large datasets. This paper extended the existing concept of principal curves to granular principal curves built on the basis of IGs constructed on the basis of large numeric data. FCM clustering and the principle of justifiable granularity formed a two-phase process supporting the formation of the hyperboxes (hyper-rectangles) IGs. We showed that in this way, the computation time could be substantially reduced. We constructed the two-criteria objective function to evaluate the granular output of the principal curves, and adjust the level of information granularity (C) to optimize the granular principal curves. The proposed approach helped achieve a sound tradeoff between efficiency and accuracy.

A number of numeric studies were completed for synthetic and real-world dataset to demonstrate the effectiveness of the proposed approach. When encountering different noise levels, the proposed method was more robust. Furthermore, the GPC method outperformed the PC counterpart without data granulation when being evaluated in terms of the running time (T).

Given the nature of real-world problems, how to detect granular principal surfaces or even granular principal manifolds of higher dimensions becomes an important issue for future research. It is worth noting that the term granular refers here to interval-valued data and subsequently interval-valued principal curves. An interesting avenue to investigate would be to engage other formalisms of IGs such as fuzzy or rough sets, which may lead to notions of fuzzy principal curves or rough principal curves.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] P. Baldi and K. Hornik, “Neural networks and principal component analysis: Learning from examples local minima,” *Neural Netw.*, vol. 2, no. 1, pp. 53–58, 1989.
[2] T. Hastie and W. Stuetzle, “Principal curves,” *J. Amer. Stat. Assoc.*, vol.84, no. 406, pp. 502–516, 1989.
[3] J. D. Banfield and A. E. Raftery, “Ice floe identification in satellite images using mathematical morphology and clustering about principal curves,” *J. Amer. Stat. Assoc.*, vol. 87, no. 417, pp. 7–16, 1992.
[4] K. Reinhard and M. Niranjan, “Parametric subspace modeling of speech transitions,” *Speech Commun.*, vol. 27, no. 1, pp. 19–42, 1999.
[5] X. Liu and Y. Jia, “A bottom-up algorithm for finding principal curves with applications to image skeletonization,” *Pattern Recognit.*, vol. 38, no. 7, pp. 1079–1085, 2005.
[6] E. Bas and D. Erdogmus, “Principal curves as skeletons of tubular objects: Locally characterizing the structures of axons,” *Neuroinformatics*, vol. 9, nos. 2–3, pp. 181–191, 2011.
[7] D. C. Stanford and A. E. Raftery, “Finding curvilinear features in spatial point patterns: Principal curve clustering with noise,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 601–609, Jun. 2000.
[8] H. Zhang, “The research of off-line handwritten character recognition based on principal curves,” Ph.D. dissertation, Dept. Comp. Sci. Technol., Tongji Univ., Shanghai, China, 2005.
[9] H. N. Wang and T. C. M. Lee, “Extraction of curvilinear features from noisy point patterns using principal curves,” *Pattern Recognit. Lett.*, vol. 29, no. 16, pp. 2078–2084, 2008.
[10] H. Zhang, W. Pedrycz, D. Miao, and C. M. Zhong, “A global structure-based algorithm for detecting the principal graph from complex data,” *Pattern Recognit.*, vol. 46, no. 6, pp. 1638–1647, 2013.
[11] H. Zhang, D. Miao, and C. M. Zhong, “Modified principal curves based fingerprint minutiae extraction and pseudo minutiae detection,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 25, no. 8, pp. 1243–1260, Dec. 2011.
[12] D. Chen, J. Zhang, S. Tang, and J. Wang, “Freeway traffic stream modeling based on principal curves analysis,” *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 246–258, Dec. 2004.
[13] J. Zhang, D. Chen, and U. Kruger, “Adaptive constraint K-segment principal curves for intelligent transportation systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 4, pp. 666–677, Dec. 2008.
[14] J. Zhang, X. Wang, U. Kruger, and F.-Y. Wang, “Principal curve algorithms for partitioning high-dimensional data spaces,” *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 367–380, Mar. 2011.
[15] E. Jochen, T. Gerhard, and E. Ludger, “Data compression and regression based on local principal curves,” in *Advances in Data Analysis, Data Handling and Business Intelligence*. Berlin, Germany: Springer, 2009.
[16] R. Tibshirani, “Principal curves revisited,” *Stat. Comput.*, vol. 2, no. 3, pp. 183–190, 1992.
[17] P. Delicado and M. Huetra, “Principal curves of oriented points: Theoretical and computational improvements,” *Comput. Stat.*, vol. 18, no. 2, pp. 293–315, 2003.
[18] B. Kégl, A. Krzyzak, and T. Linder, “Learning and design of principal curves,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 3, pp. 281–297, Mar. 2000.
[19] B. Kégl, “Principal curves: Learning, design, and applications,” Ph.D. dissertation, Dept. Comp. Sci., Concordia Univ., Montreal, QC, Canada, 1999.
[20] B. Gérard and F. Aurlie, “Parameter selection for principal curves,” *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 1534–1570, Nov. 2011.

[21] J. J. Verbeek, N. Vlassis, and B. Krose, "A k-segments algorithm for finding principal curves," *Pattern Recognit. Lett.*, vol. 23, no. 10, pp. 1009–1017, 2002.

[22] J. Einbeck, G. Tutz, and L. Evers, "Local principal curves," *Stat. Comput.*, vol. 15, no. 4, pp. 301–313, 2005.

[23] J. Zhang, U. Kruger, X. Wang, and D. Chen, "A Riemannian distance approach for constructing principal curves," *Int. J. Neural Syst.*. vol. 20, no. 3, pp. 209–218, 2010.

[24] U. Ozertem and D. Erdogmus, "Locally defined principal curves and surfaces," *J. Mach. Learn. Res.*, vol. 12, no. 4, pp. 1249–1286, 2011.

[25] W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*. Hoboken, NJ, USA: Wiley, 2005.

[26] W. Pedrycz, "A dynamic data granulation through adjustable fuzzy clustering," *Pattern Recognit. Lett.*, vol. 29, no. 16, pp. 2059–2066, 2008.

[27] Q. Hu, D. Yu, J. Liu, and C. Wu, "Neighborhood rough set based heterogeneous feature subset selection," *Inf. Sci.*, vol. 178, no. 16, pp. 3577–3594, Sep. 2008.

[28] Q. Hu, W. Pedrycz, D. Yu, and J. Lang, "Selecting discrete and continuous features based on neighborhood decision error minimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 1, pp. 137–150, Feb. 2010.

[29] W. Pedrycz and A. Bargiela, "Granular clustering: A granular signature of data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 2, pp. 212–224, Apr. 2002.

[30] L. A. Zadeh, "Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, no. 2, pp. 111–127, 1997.

[31] W. Pedrycz and A. V. Vasilakos, "Linguistic models and linguistic modeling," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 6, pp. 745–757, Dec. 1999.

[32] W. Pedrycz, B. J. Park, and S. K. Oh, "The design of granular classifiers: A study in the synergy of interval calculus and fuzzy sets in pattern recognition," *Pattern Recognit.*, vol. 41, no. 2, pp. 3720–3735, 2008.

[33] Y. Yao, "Interpreting concept learning in cognitive informatics and granular computing," *IEEE Trans. Syst., Man, Cybern. B, Cybern.* vol. 39, no. 4, pp. 855–866, Aug. 2009.

[34] W. Pedrycz, "The design of cognitive maps: A study in synergy of granular computing and evolutionary optimization," *Expert Syst. Appl.*, vol. 37, no. 10, pp. 7288–7294, 2010.

[35] J. T. Yao, A. V. Vasilakos, and W. Pedrycz, "Granular computing: Perspectives and challenges," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1–13, Mar. 2013.

[36] Y. Yao, "Information granulation and rough set approximation," *Int. J. Intell. Syst.*, vol. 16, no. 1, pp. 87–104, 2001.

[37] W. Pedrycz, "Fuzzy equalization in the construction of fuzzy sets," *Fuzzy Sets Syst.*, vol. 119, no. 2, pp. 329–335, 2001.

[38] J. Yao and Y. Yao, "Induction of classification rules by granular computing," in *Proc. 3rd Int.Conf. Rough Sets Current Trends Comput.*, vol. LNAI-2475. 2002, pp. 331–338.

[39] W. Pedrycz, *Granular Computing: Analysis and Design of Intelligent Systems*. Boca Raton, FL, USA: CRC Press, 2013.

[40] D. Miao and J. Wang, "On the relationships between information entropy and roughness of knowledge in rough set theory," *Pattern Recognit. Artif. Intell.*, vol. 11, no. 1, pp. 34–40, 1998.

[41] D. Miao and J. Wang, "An information representation of the concepts and operations in rough set theory," *J. Softw.*, vol. 10, no. 2, pp. 113–116, 1999 (in Chinese).

[42] W. Pedrycz, "The principle of justifiable granularity and an optimization of information granularity allocation as fundamentals of granular computing," *J. Inf. Process. Syst.*, vol. 7, no. 3, pp. 397–412, 2011 (Invited Paper).

[43] W. Pedrycz, "Interpretation of clusters in the framework of shadowed sets," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2439–2449, 2005.

[44] W. Pedrycz and A. Bargiela, "An optimization of allocation of information granularity in the interpretation of data structures: Toward granular fuzzy clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 582–590, Jun. 2012.

[45] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362–1381, Dec. 2009.

Hongyun Zhang received the Ph.D. degree in pattern recognition and intelligence system from Tongji University, Shanghai, China, in 2005.

She is currently an Associate Professor at Tongji University. She is the author or co-author of nearly 50 journal papers and conference proceedings in principal curves, pattern recognition, machine learning granular computing, and rough set. Her current research interests include principal curves, pattern recognition, data mining, rough set theory, and granular computing.

Witold Pedrycz (F'89) received the M.Sc., Ph.D., and D.Sci. degrees from Silesian University of Technology, Gliwice, Poland, in 1977, 1980, and 1984, respectively.

He is currently a Professor and Canada Research Chair (CRC—computational intelligence) with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia. He is also with the Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland. He also holds an appointment of special professorship in the School of Computer Science, University of Nottingham, Nottingham, U.K. In 2009, he was elected as a foreign member of the Polish Academy of Sciences. He is also an author of 15 research monographs covering various aspects of computational intelligence and software engineering. His current research interests include computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering. He has published numerous papers in this area.

Mr. Pedrycz became a fellow of the Royal Society of Canada in 2012. He has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He is intensively involved in editorial activities. He is an Editor-in-Chief of *Information Sciences* and Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and is a member of a number of editorial boards of other international journals. In 2007, he received a prestigious Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Council and in 2013 a Killam Prize. He is the recipient of the IEEE Canada Computer Engineering Medal 2008. In 2009, he received a Cajastur Prize for soft computing from the European Centre for Soft Computing for pioneering and multifaceted contributions to granular computing.

Duoqian Miao received the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1997.

He is currently a Professor with the School of Electronics and Information Engineering and the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, China. His current research interests include soft computing, rough sets, pattern recognition, data mining, machine learning, and intelligent systems. He has published more than 160 papers in this area, more than nine books and academic works, and nine national invention patents.

Zhihua Wei received the double Ph.D. degrees in pattern recognition and intelligent system from Tongji University, Shanghai, China, and Lyon University 2, Lyon, France, in 2010.

She is currently a Lecturer at Tongji University. Her current research interests include machine learning, image processing, and natural language processing.