Recursive Information Granulation: Aggregation and Interpretation Issues

Andrzej Bargiela, *Member, IEEE,* and Witold Pedrycz, *Fellow, IEEE*

***Abstract*—This paper contributes to the conceptual and algorithmic framework of information granulation. We revisit the role of information granules that are relevant to several main classes of technical pursuits involving temporal and spatial granulation. A detailed algorithm of information granulation, regarded as an optimization problem reconciling two conflicting design criteria, namely, a specificity of information granules and their experimental relevance (coverage of numeric data), is provided in the paper. The resulting information granules are formalized in the language of set theory (interval analysis). The uniform treatment of data points and data intervals (sets) allows for a recursive application of the algorithm. We assess the quality of information granules through the application of fuzzy c-means (FCM) clustering algorithm. Numerical studies deal with two-dimensional (2-D) synthetic data and experimental traffic data.**

***Index Terms*—Complex systems, data mining-oriented time-series analysis, fuzzy sets, granular clustering, information granules and granulation, interval analysis, perception, time-series, traffic data.**

I. Introductory Comments

INFORMATION granulation and information granules play a crucial role in many areas of knowledge representation and problem solving. Briefly speaking, we may claim that those information granules permeate most cognitive activities of humans and help organize knowledge about the external world (or being more precise build its perception) for the purposes of decision-making, control, system description, prediction, etc. Zadeh [31]–[34] promoted a notion of information granulation in the framework of fuzzy sets. Other formal and commonly exploited environments of information granulation deal with rough sets [23], probability, and set theory (interval analysis) [1], [9], [11], [18], [21], [23]. In a nutshell, information granules are treated as collections of entities (e.g., numeric readings) that are collected together because of their similarity, functional closeness, or any other criterion that captures a feature of indistinguishability. Information granules give rise to hierarchies of cognitive entities. Depending upon the level of details in which one is interested, we need information granules of different sizes (obviously, this term requires a formal definition). It goes without saying that information granules are *conceptual* constructs not necessarily directly implied by the needs of the physical world. In this way, information granules feature a high level of flexibility. On the other hand, they have to be anchored in the world of experimental data to reflect in some way the reality (i.e., reflect the physical world). To put it another way, the design of information granules needs to take the perception and experimental evidence into consideration. Adopting this point of view, this paper casts the problem of information granulation in a well-defined algorithmic setting. We propose a detailed algorithmic path showing how information granules can be constructed on the basis of the existing experimental evidence. It is also revealed how the derived information granules can be combined even further via fuzzy clustering.

In this paper, we are concerned with information granules and information granulation carried out in the setting of set theory and interval analysis. The rationale behind a selection of this formal framework is twofold. First, interval analysis has been around as one of the cornerstones of granular computing. As originating from set theory, the conceptual aspects of interval analysis are well developed. Second, the algorithmic layer of set (interval) calculus has been mastered for a long time and has resulted in a vast number of algorithms [11], [21], [22]. Interestingly enough, the findings reported in this paper could be translated and applied to other frameworks of granular computing, such as fuzzy sets (the conversion of the results derived here hinges on the idea of representing fuzzy sets through their α-cuts [15], [27], [36]; that is, splitting the problem into a family of set-based granulation tasks).

This paper is organized as follows. First, in Section II, we concentrate on selected areas in which information granulation and information granules play a crucial role. Next, in Section III, a detailed two-level algorithm for information granulation is discussed along with a characterization of information granules in terms of their size (cardinality), specificity (width of hyperbox projections), and an integrative measure that we call "information density." Summarization of information granules and assessment of the effectiveness of granulation is performed through fuzzy c-means (FCM) clustering of granules in the augmented pattern space. This issue is discussed in Section IV. A broader perspective on alternative granulations of time-series data is provided in Section V. These alternative granulations are also assessed using FCM clustering. An illustrative example concerning urban traffic data is included in Section VI. This is followed by concluding remarks presented in Section VII.

Manuscript received October 29, 2001. This work was supported by the Engineering and Physical Sciences Research Council (U.K.), the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Alberta Consortium of Software Engineering (ASERC). This paper was recommended by Associate Editor G. Skarmeta.

A. Bargiela is with the Department of Computing, The Nottingham Trent University, Nottingham NG1 4BU, U.K. (e-mail: andre@doc.ntu.ac.uk).

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB R3T 2N2, Canada and with the Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland (e-mail: pedrycz@ee.ualberta.ca).

Digital Object Identifier 10.1109/TSMCB.2003.808190

II. Data Granulation: Examples, Definitions, and Algorithmic Issues

There are a number of representative domains where information granules can emerge as a useful vehicle to represent a given problem and make problem solving more efficient [20], [26], [29]. The following three areas are among the most prominent applications of information granulation.

1) Granulation of Time-Series: Time-series are commonly encountered in numerous practical problems [4]. There have been various approaches to the description of time-series and their classification. They are carried out in the time domain and frequency domain. Prior to any detailed processing, time-series are compressed in order to retain the most essential information and suppress details that are deemed redundant from the standpoint of further classification and processing. The essence of granulation of time-series is to "discover" dominant components of the series. We may perceive these components as playing a role of basic conceptual blocks easily understood by humans and capturing the semantics of the underlying phenomenon. For instance, information granules may be formed as segments of consecutive samples of the signal. Then, each segment may be labeled according to the configuration of the samples, e.g., rapidly increasing signal, steady signal, slowly decreasing signal, etc. [8]. Alternatively, as we propose in this paper, one may consider granulating the time-series value with its gradient (and/or higher order derivatives) in individual time instances. Note that standard sampling techniques are very specific examples of granulation of time-series (as we attempt to capture a segment of a signal falling under a given sampling window by a single numeric value).

2) Granulation of Digital Images: Digital images are two-dimensional (2-D) relations. As far as understanding and processing of images is concerned, a crux there is to identify some higher level entities rather than being buried in a minute analysis completed at the level of individual pixels. Such tangible and semantically sound entities are information granules. They may arise at the level of basic homogeneous regions (in terms of brightness, color, and texture) one can identify in an image. These entities are inherently hierarchical; at a higher level we may think of individual objects in the image (that are composed of the granules arising at the lower level with more specific and less abstract information granules). At the technical end, the simplest and least abstract information granules are formed by defining $n \times m$ blocks of pixels [19], [24], [25]. At the higher level, we are concerned with various clustering techniques that help us construct abstractions out of the low-end (more detailed) information granules such as the already mentioned blocks of pixels.

3) Granulation of Spatial Structures: An array of current modeling pursuits occurs in the realm of distributed systems such as networks. Obvious examples of these architectures are electric networks, water networks, or telecommunication networks. In spite of their evident diversity, the networks share several profound commonalties. In particular, a hierarchical type of modeling is omnipresent there. Instead of analyzing the entire network, we split it into subnetworks (modules) that are loosely connected and proceed with a detailed analysis at this level. Obviously, this task is more tangible and manageable from the computational and interpretation standpoint. Each subnetwork is an information granule that is afterwards treated as a conceptual and algorithmic entity. For instance, when looking into a flow of traffic in a complex network, we partition the network into modules (call them telecommunications granules) and study all incoming and outgoing traffic from this perspective, refer to Fig. 1. The concept of hierarchy and information granulation is inherently associated with geographic information systems (GIS) where we anticipate various levels of detail and control the process of concentrating on specific aspects by establishing proper levels of information granularity.

III. Information Granules: Design and Characterization

In this section, we discuss the algorithmic layer of information granulation by studying a way of constructing information granules. As we have confined ourselves to a set-theoretic formalism of information granulation, we show how to construct intervals or their multidimensional versions (that is hyperboxes). In the discussed framework, granulation applies to numeric data. The granular properties of sets are straightforward: the larger the size of the interval, the lower its granularity. The cardinality of a set, card(.), serves as a suitable measure of information granularity. In general, the following holds: the bigger the cardinality of the set, the lower its granularity.

A. Building Set-Based Information Granules

In the proposed approach, information granules are designed in two stages (phases). First, in the entire data set under analysis, we define the size of a segment (window of granulation), specify the elements (data points) within each segment and, in sequel, use these elements to construct a detailed form of the information granule. More formally, the granulation process can be delineated as follows:

$$\mathbf{X} \underset{\Omega}{\Rightarrow} A \tag{1}$$

where

- $\mathbf{X}$ original data set;
- Ω set of disjoined time periods Ω_k representing windows of observation;
- A set of information granules.

The first phase is straightforward: by defining the size of the data segment ($\omega_0 = \text{card}(\Omega_k), k = 1, 2, \ldots, G$) we embrace a collection of data points that is of interest and need to be considered together when constructing a detailed model of the information granule. It should be emphasized that while ω_0 defines a maximum cardinality of initial granules it does not prevent formation of several smaller granules within any given window of observation (as will be explained later). Windows of observation can be formed in many different ways (see Fig. 2). The choice depends on the problem and reflects a way in which the semantics of the problem is addressed. For instance, the window can include a fixed number of samples. In some other cases, as shown in Fig. 2(b), the window can exhibit a different level of granularity by being formed using monotonic (increasing or decreasing) parts of the signal, which may be of varying length.

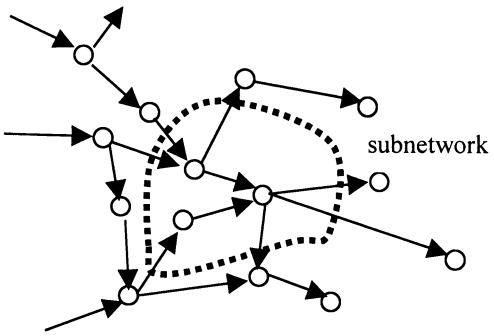

Fig. 1. Concept of network granulation; we focus on structural granules by studying a flow of traffic at the level of a selected part of the network.

Note that the formation of the window of observation is implied by the characteristics of the problem at hand.

In the sequel, the process of constructing (and subsequent recursive refinement) of information granules comprises two phases:

1) derivation of information granule(s) from the original numeric data contained in the window of observation;
2) recursive processing of the mixture of granular and numeric data.

In the detailed construct, we start with a collection (block) of data $\{x_j\colon j \in \Omega_k\}$, as shown in Fig. 3(a). The phase-1 granulation results in a mixture of granules and data points that represent local maxima of the "information density" function (defined later in this section). The result of this granulation is visualized in Fig. 3(b). Subsequent recursive granulation of the output from phase-1 produces level-2, level-3, etc., granules [as shown in Fig. 3(c)]. For simplicity of discussion, we consider here just scalar numeric data but we can easily generalize this construct to a vector case by studying each coordinate of the multidimensional data separately.

In this paper, we adopt a set theoretic framework for the development of the granulation algorithm, thus focusing on the "bottom-up" approach to understanding the nature of data, as advocated in [5], [8], and [12]. In this context, one of the contributions of this paper is the development of the optimization-based granulation algorithm. The proposed algorithm does not require specification of the number or size of information granules and it focuses fully on the character of the data itself.

Building interval-valued granules arises as a compromise between two evidently conflicting requirements.

1) The interval should "embrace" as many elements of $\{x_j\colon j \in \Omega_k\}$ as possible (to be a sound representation of the window of observation).
2) The interval should be highly specific. This translates into the requirement of a minimal length of this interval (set).

As far as the first requirement is concerned, a cardinality of the set covering elements of Ω_k is a suitable criterion, that is

$$\text{card}(I) = \sum_{x_i \in X} \chi_{[a,b]}(x_k) \tag{2}$$

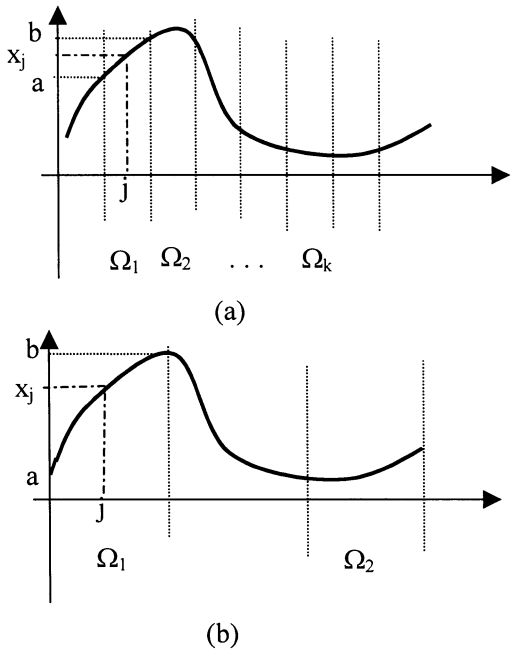

Fig. 2. Examples of segments (windows of observation) Ω_k: (a) of the same granularity and (b) variable granularity induced by the monotonicity of the signal.

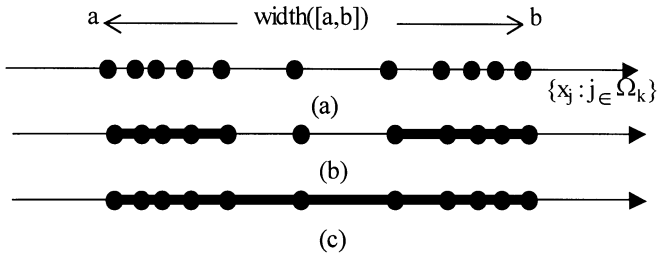

Fig. 3. Illustration of the concept of recursive granulation: (a) original data, (b) phase-1 granulated data, and (c) (phase-2, phase-3, etc.) granulated data.

where $I = [a, b]$ denotes the interval we are about to construct and $\chi_{[a,b]}$ stands for its characteristic function, that is

$$\chi_{[a,b]}(x) = \begin{cases} 1, & \text{if } x \text{ is in } [a, b] \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The specificity of the interval can be directly associated with its width

$$\text{width}(I) = \text{width}([a, b]) = b - a. \tag{4}$$

More precisely, the larger the width of the interval, the lower its specificity. While this definition is straightforward, we will be using its slightly enhanced version expressed as

$$\phi(\text{width}([a, b])) \tag{5}$$

where ϕ is monotonically increasing function of the original width and $\phi(0) = 1$ (as will become obvious soon, this boundary condition is introduced for the sake of uniformity of processing of data points and data intervals). For instance, a mapping of interest can assume the form

$$\phi(u) = \exp(u). \tag{6}$$

Bearing in mind a conflicting nature of the requirements 1) and 2) that is captured in the form

$$\text{card}(I) \rightarrow \max \qquad \phi(\text{width}([a, b])) \rightarrow \min \tag{7}$$

it is legitimate to take a ratio of these equations

$$\sigma = \frac{\text{card}(I)}{\phi(\text{width}(I))} \tag{8}$$

and determine the interval I so that it maximizes (8). In this way, we cope simultaneously with the two contributing optimization problems defined in (7). We refer to the optimization expressed by (8) as maximization of "information density" of granules. This is to distinguish it from the concept of "data density" that is typically represented as a ratio of cardinality of a given set over the volume of the pattern space containing this set. Consequently "data density" is not defined for a single numeric data (zero volume in pattern space).

The choice of function $\phi(u)$ depends on the preference for large or small information granules. Fig. 4 shows contour plots of (8) obtained with $\phi(u)$ as defined by (6). It can be seen that the decrease of the gradient of the contours with the increase of the cardinality of the granules implies inherent preference for smaller granules. This is an advantageous feature as it gives us a possibility of avoiding undue influence of inherently local optimization on the more global view of data that is obtained through recursive application of the granulation algorithm. An alternative choice of $\phi(u) = 1 + u$ results in constant-gradient contours of (8) and is thus less appropriate in the context of our algorithm. A function $\phi(u) = (1 + u)^2$ results in contour plots that are broadly similar to those obtained with $\phi(u) = \exp(u)$, but it is less convenient numerically.

The above considerations generalize easily on the case of multidimensional data. The maximization of information density, implied by (8), can be performed for multidimensional hyperboxes. We consider in this case a ratio of the cardinality of the input data set contained in such hyperboxes to a function of volume of the hyperboxes. However, such a direct approach creates dependence of the information density measure on the dimensionality of the pattern space. Given that in the interest of the uniformity of processing of data points and intervals we deliberately increase the dimensionality of the pattern space (as explained in detail in Section IV) it is advantageous to consider a dimensionality-invariant version of mapping $\phi(u)$. This can be given as follows:

$$\phi(u) = \exp\left(\max_i(u_i)\right) * \exp\left(\max_i(u_i) - \min_j(u_j)\right) \tag{9}$$

where $u = [u_1\ u_2\ \cdots\ u_n]$, $u_i = \text{width}([a_i, b_i])$, and $i, j = 1, 2, \ldots, n$ is an index of the dimension of the pattern space. The first exponent function in (9) ensures that the specificity of information granules is maximized through the reduction of the maximum width of the hypercube along all dimensions in the pattern space. The second exponent in (9) ensures that the hyperboxes are as similar to hypercubes as possible. The above function can be expressed in a more compact form

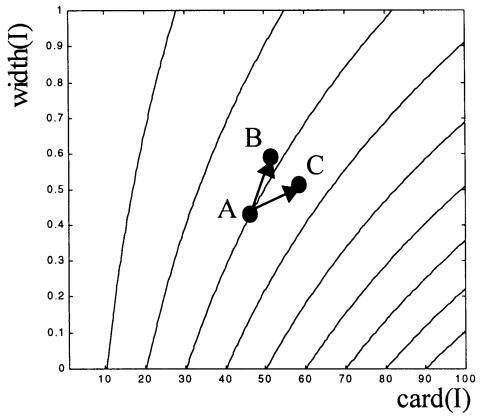

Fig. 4. Contour plot of the information density function (8); $\sigma(I) = \text{const}$. Transition from granule A to B represents a net decrease of information density and is therefore avoided. Transition from A to C represents formation of a granule with higher information density.

$$\phi(u) = \exp\left(2 * \max_i(u_i) - \min_j(u_j)\right) \tag{10}$$

where $i, j = 1, \ldots, n$. It is clear that (10) is not affected by the dimensionality of the pattern space. The maximization of the width of the hyperbox (granule), over all dimensions of the pattern space, results in a scalar value that is of the same order regardless of the space dimension. Furthermore, the function satisfies the original boundary condition $\phi(0) = 1$, since for the point-size data $\max_i(u_i) = \min_i(u_i) = 0$, $i = 1, \ldots, n$.

While, in general, the pattern space $\mathbf{X}$ can be any subset of $\mathbf{R}^n$, we restrict the operation of the optimization task (8) to a unit hypercube $[0, 1]^n$. Such a restriction does not imply any loss of generality of our approach while affording clear computational benefits [with regard to mapping $\phi(u)$].

The optimization-based granulation of data is carried out as a one-pass simulation process:

1) normalize data to a unit hypercube;
2) initialize data structures representing cardinality and the width of individual data items (1 and 0, respectively, for the point-data);
3) calculate and store the value of "information density" [as implied by (8)] of hypothetical granules formed by any two data items in the input data set. This forms an upper-diagonal matrix D of size $N \times N$, where N is the cardinality of the input data set.
4) find a maximum entry in D;
5) if the maximum corresponds to an off-diagonal element ($j \in \{1, 2, \ldots, N\}$, $i \in \{j, \ldots, N\}$):
   - merge the two information items (identified by the ith row and jth column) into a single information granule, which has width defined by the maximum and minimum values of coordinates in each dimension from the two component granules; i.e.:
     - find the $\min$ of each of the coordinates of the ith and jth granule;

- find the max of each of the coordinates of the ith and jth granule;
- modify the ith granule so that its size is defined by min- and max-values identified above;

- update the cardinality of the resulting granule to the sum of the cardinality counts of the component granules;
- update the ith row and column of D with the information pertinent to the newly formed information granule and remove the jth row and column from D; i.e.:
  - copy rows 1 to $j-1$ from matrix D to matrix $D1$;
  - copy rows $j+1$ to size(D) from matrix D as rows j to size(D) $-$ 1 in matrix $D1$;
  - copy columns 1 to $j-1$ from matrix $D1$ to matrix $D2$;
  - copy columns $j+1$ to size(D) from matrix $D1$ as columns j to size(D)-1 in matrix $D2$;
  - overwrite matrix D with matrix $D2$;
- return to d).

6) If the maximum corresponds to a diagonal element ($i = j$):

- copy the granule to an output list and remove the corresponding row and column from matrix D; i.e.:
  - copy rows 1 to $j-1$ from matrix D to matrix $D1$;
  - copy rows $j+1$ to size(D) from matrix D as rows j to size(D) $-$ 1 in matrix $D1$;
  - copy columns 1 to $j-1$ from matrix $D1$ to matrix $D2$;
  - copy columns $j+1$ to size(D) from matrix $D1$ as columns j to size(D)-1 in matrix $D2$;
  - overwrite matrix D with matrix $D2$;
- if the size of matrix D is greater than 1, return to d), otherwise terminate.

Computational complexity of this granulation algorithm is $O(N^2)$ owing to the computations of matrix D in step c). However, unlike the clustering techniques (such as FCM), the granulation process has an inherently local character and can be easily applied to a partitioned input data thus circumventing the high computational cost associated with large data sets. It is worth pointing out that the size of matrix D is being reduced by one row and column in each iteration thus the number of iterative steps equals $N-1$.

The above algorithm is somewhat similar to "subtractive clustering" proposed by Chiu [5] in that the algorithm avoids any arbitrary partition of the input space and is driven purely by the existing input data. In contrast to grid-based methods, areas of input space that do not have data are simply ignored by the two algorithms. Both our algorithm and "subtractive clustering" avoid combinatorial explosion of relationships with the increasing dimension of the input space. Since the algorithms maintain linear computational complexity with respect to the input space dimension (not to be confused with the complexity with respect to the cardinality of the data set which is $O(N^2)$), they are particularly suitable for processing multidimensional data. Another common characteristics of the two algorithms is that they maintain a localized view of data. As the granulation proceeds, the identified clusters do not exercise further influence on data points that remain after their removal.

However, there is a significant difference between the two algorithms. In our algorithm we do not make any assumptions about the maximum size of granules. Granules are allowed to grow as long as their local data density keeps increasing. Furthermore, we do not make any arbitrary decision about the separation of cluster centers. The formation of closely separated granules is largely avoided by the very nature of maximization of information density, which tends to increase the size of granule if it means adding sufficiently large number of data items (another granule) without undue increase of its volume. If, on the other hand, the increase in volume would imply the reduction of information density, the granule does not expand and remains well separated from the neighboring granules. Another distinguishing feature of our algorithm is that it allows processing both point-size and hyperbox data. This is an important characteristic that allows hierarchical granulation of data. It should be noted that hierarchical granulation enables overcoming the limitations of the "local view" of data while supporting the application of the algorithm to a partitioned input data set.

It is also instructive to point out an important difference between the hierarchical clustering and the hierarchical granulation proposed here. In hierarchical clustering, the similarity or proximity measure is evaluated for all data in the unpartitioned pattern space. This renders hierarchical clustering computationally expensive at the early clustering stages. By contrast, hierarchical granulation can operate on the partitioned pattern space (thus achieving significant computational gains because of the quadratic computational complexity of the algorithm with respect to the cardinality of the data set) and the subsequent application of the algorithm to the partially granulated patterns enables arriving at the globally optimal granulation.

The granulation algorithm introduced here is developed against a background of some recent developments in this area. In particular, it complements the fuzzy granulation approaches proposed in [2], [12], and [30]. We demonstrate that crisp granulation followed by fuzzy clustering offers a powerful framework for deriving data abstractions, but further study will be undertaken to investigate the relative merits of generalizing our algorithm to its fuzzy variant.

To illustrate the operation of the granulation algorithm, we consider here a synthetic 2-D time-series, as presented in Fig. 5.

The granulation algorithm is then applied to data from Fig. 5(b). Apart from identifying the granules themselves, we monitor the value of information density index throughout the granulation process. Of course, the information density of granules identified toward the end of the process is lower then the information density of the early granules. This is because the removal of "high information density" granules leaves effective voids in the pattern space. We can utilize this

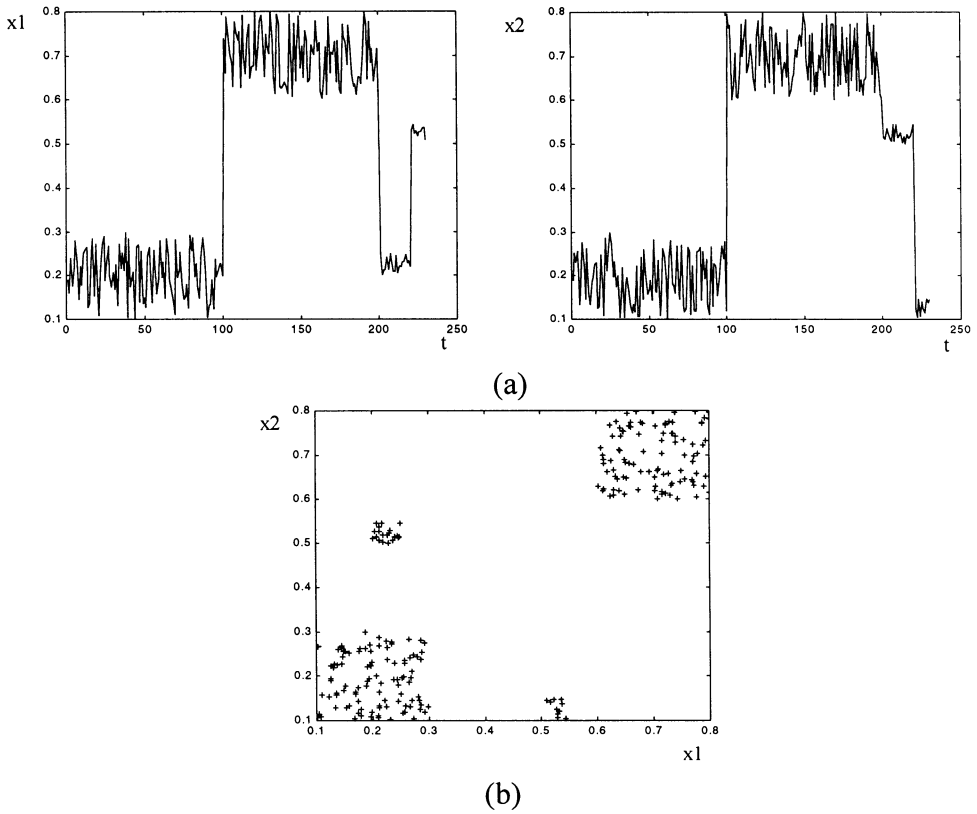

Fig. 5. Synthetic 2-D time-series: (a) time plot and (b) state plot.

indicator in two ways. We can either terminate the granulation process when the information density of granules reaches a prespecified threshold level (which effectively discards some data), or we can perform a "higher level" granulation on the identified granules (which merges granules from the previous level). Here, we adopt the latter approach.

The operation of the granulation algorithm, over three hierarchical levels of granulation, is illustrated in Fig. 6. The "level-1" granulation compresses the original set of 230 data points into 27 granules. These granules are presented as input data to "level-2" granulation, which results in nine granules. "Level-3" granulation reduces this number further to six granules. It is self-evident that the hierarchy of granules forms an abstraction that preserves the essential characteristics of the original data (that of four relationships in the pattern space). Of even greater importance, the granulation has balanced the relative count of data items in large and small data groupings, thus helping smaller data groups to be "noticed" in subsequent processing (clustering).

It is worth noting that the number of granulation levels does not need to be defined in advance. The hierarchical granulation is simply carried out until the number of granules identified at the subsequent granulation levels does not change. Of course, in any practical application, the maximum size of granules is frequently predefined so that the granules map conveniently onto some linguistic entities. In this case, the relative weighing of the two components in (8) can be adjusted so as to achieve the required granularity.

IV. Assessment and Interpretation of Information Granules Through Fuzzy Clustering

The recursive application of the granulation algorithm discussed in the previous section, condensed the data quite significantly. What is of fundamental interest, however, is whether this "condensing" sharpens the view of the essential characteristics of data. We assess here the quality of granulation by identifying a limited number of representatives of both the original numeric data and the constructed information granules. This is accomplished by clustering and identifying prototypes (representatives) of the granules [10]. In particular, a fuzzy clustering method, a well-known FCM algorithm [3], [13], is of interest here. As a result of this clustering mechanism, the method returns a partition matrix. This matrix captures all granules in the form of some generalized architecture of fuzzy sets formed over the family of the original information granules.

Note that in contrast to the "standard" clustering method, we are concerned with a collection of hypercubes—sets in $\mathbf{R}^n$ (see Fig. 7). As in any clustering pursuit, our objective is to reveal a structure in a set of these granular data. As a consequence of the granular nature of the data set, we anticipate that the prototypes returned by the FCM are also information granules.

Owing to the granular nature of the data to be clustered, they need to be represented (encoded) in such a way that their aspect of granularity can be properly captured by the FCM method. A parametric method of processing heterogeneous data is a sound solution to this problem. Within this scope, several directions could be sought (see Fig. 7).

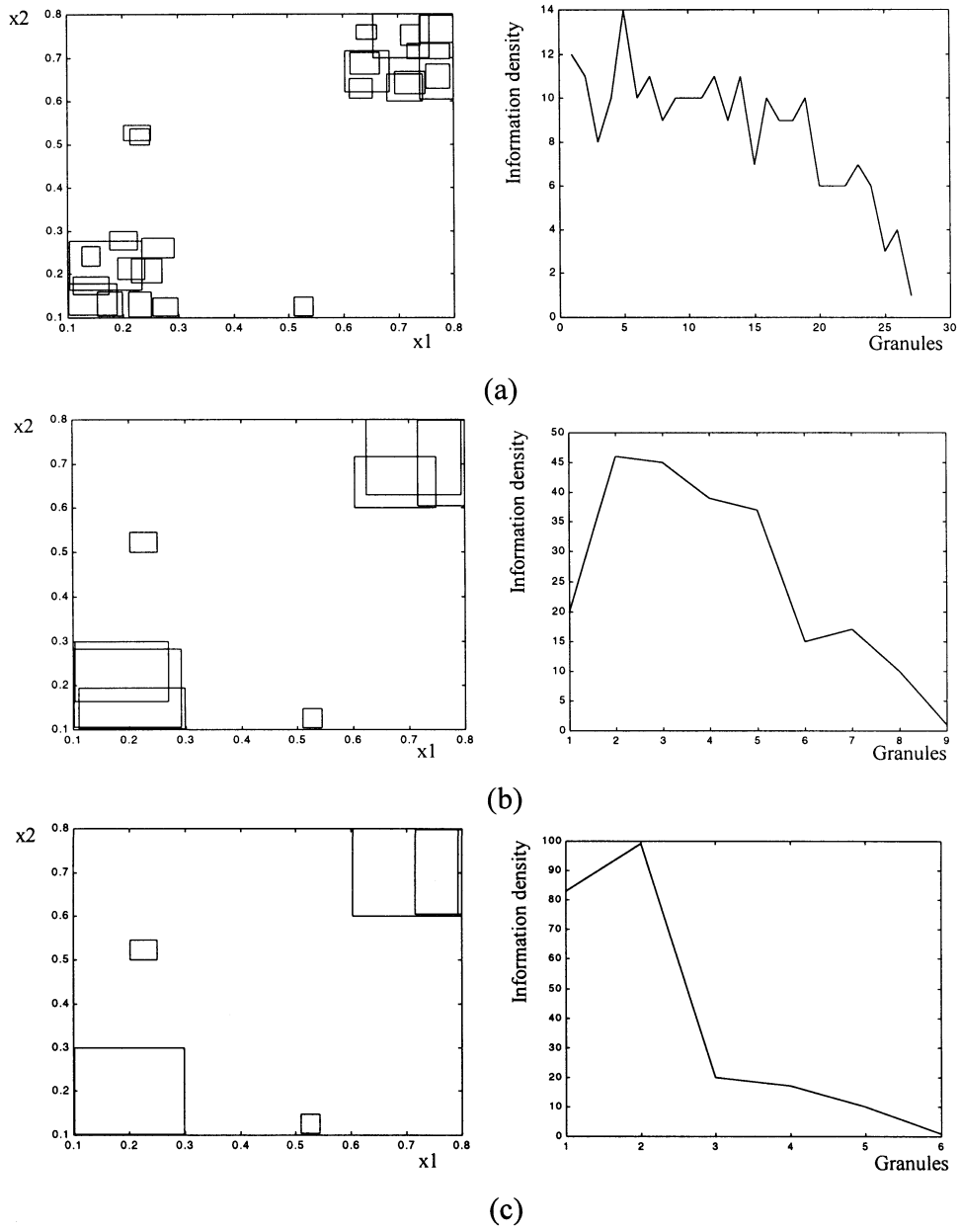

Fig. 6. Illustration of the operation of the granulation algorithm applied to synthetic 2-D time-series: (a) represents the "first-level" granulation, (b) represents the "second-level" granulation, and (c) represents the "third-level" granulation.

1) The lower and upper bound of each coordinate (feature) of the information granule (we refer to it as a *bound encoding*) can be represented. Thus, for the n-dimensional information granule, we end up with a $2n$-dimensional space of objects $\mathbf{x}^{\sim}$ to be clustered

$$\mathbf{x}^{\sim} = [x_{1-}\ x_{1+}\ \cdots\ x_{i-}\ x_{i+}\ \cdots\ x_{n-}\ x_{n+}].$$

2) Each coordinate can be represented by a center point of the granule and its width (center-width encoding). Again, this form of representation gives rise to the $2n$-dimensional space

$$\mathbf{x}^{\sim} = [m_1\ \delta_1\ \cdots\ m_i\ \delta_i\ \cdots\ m_n\ \delta_n].$$

This representation is suitable if we have an interval that is distributed symmetrically around the center. Otherwise, one has to incorporate the lower and upper width. In this case, such representation implies a higher dimensionality of the space in which the clustering takes place.

The topological implications of increasing the dimensionality of the pattern space in the above two representations can be appreciated by analyzing a single dimension of a granule. We consider here three intervals (I^A, I^B, and $I^C \subset \mathbf{R}^1$) and a point $P^D \in \mathbf{R}^1$, as shown at the top of Fig. 8.

The bounds-encoding method generates points in $\mathbf{R}^2$ that have their first coordinate (x_{i-}) representing the lower bound of the interval, and the second coordinate (x_{i+}) representing the upper bound of the interval. It is clear that all points in $\mathbf{R}^1$, map onto points along the line $x_{i-} = x_{i+}$ in $\mathbf{R}^2$ and all intervals in $\mathbf{R}^1$ map onto points in a half-space $\{x_i = (x_{i-}, x_{i+})$: $x_{i+} > x_{i-}, x_{i\in}\mathbf{R}^2\}$. What has been achieved therefore is that a heterogeneous mix of intervals and points in $\mathbf{R}^1$ has been con-

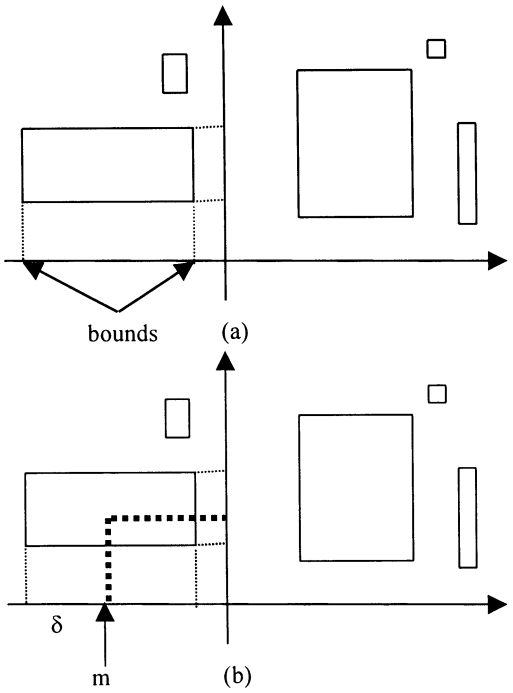

Fig. 7. Information granules to be clustered and their representation. (a) Bound representation. (b) Center-width $(m\text{-}\delta)$ representation.

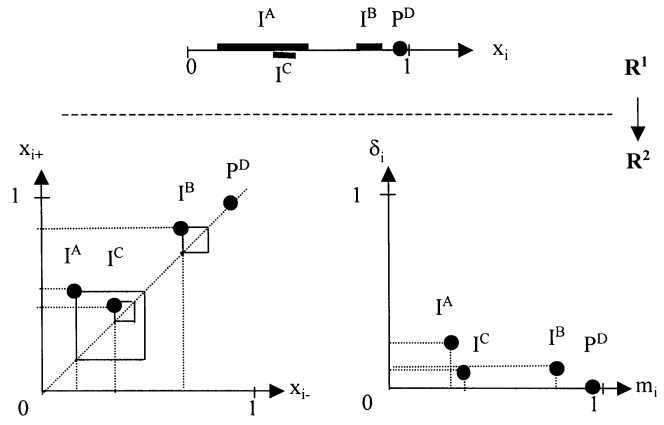

Fig. 8. Mapping of three intervals and a point in $\mathbf{R}^1$ to $\mathbf{R}^2$ space using bounds- and center-width encoding.

verted into a homogeneous set of points in $\mathbf{R}^2$; (to be precise we are only concerned with the unit interval [0, 1] and a unit box [0, 1] $\times$ [0, 1]). An interesting feature of the bounds-encoding mapping is that the occurrence of inclusion/overlap of intervals is easily detected in the mapped image in $\mathbf{R}^2$. The symmetrical reflection of the mapped intervals with respect to the diagonal line $x_{i-} = x_{i+}$ gives rise to a "box" (as illustrated in Fig. 8) for each interval. The boxes for disjoint intervals are disjoint and the boxes for overlapping intervals overlap as well.

The center-width mapping also achieves a conversion of heterogeneous mix of intervals and points in $\mathbf{R}^1$, into a homogeneous set of points in $\mathbf{R}^2$. However, the topological property of interval overlap is more difficult to identify in the mapped image. This is because the character of the two component dimensions is quite different. One dimension represents a value of data and the other dimension represents a relative variation from this value.

Bearing this in mind, we adopt in our subsequent study the bounds-encoding method so that we can maintain topological interpretability of the mapped points and intervals. It should be pointed out that, in general, we will concern ourselves with mapping from $\mathbf{R}^n$ to $\mathbf{R}^{2n}$ (or more precisely from $[0, 1]^n$ to $[0, 1]^{2n}$).

Having achieved a homogeneous representation of input data, the application of standard FCM clustering [3] returns a partition matrix and a collection of cluster prototypes. These prototypes are of the same dimensionality as the input data; therefore, they can be interpreted in the original data space as hyperboxes. In particular, the prototypes represent now fuzzy decomposable relations in the feature space [27], [34], [35], in addition to representing, through the partition matrix, the fuzzy membership of data in clusters. The combination of the two aspects delivers a more comprehensive insight into the granular nature of information being summarized by the prototypes.

To illustrate the clustering of information granules, we continue with the example given in the previous section. The FCM algorithm is deployed first on the original data (to provide a base reference) and then on the granulated data. The number of clusters is kept constant $(c = 4)$, so that the issue of size and positioning of prototypes is brought into sharper focus. The results are shown in Fig. 9 and Table I.

As expected, the granular input data gives rise to granular prototypes. The size of the prototypes affords a good appreciation of the spatial dimensions of the original data groupings. This is in contrast to the standard result [see Fig. 9(a)] where the prototypes are point-size and, in themselves, do not convey this information. Although the FCM partition matrix contains information that represents the area of influence of individual clusters its direct interpretation is quite difficult due to the complex topology of the contour plots of the partition matrix. In this sense, clustering of granular data affords a better insight into the nature of data.

Another important observation that can be made from the above results is that the information granulation helps to overcome the well-known bias of the FCM algorithm, that of underrepresenting smaller groupings of data. Since the granulation reduces the number of information items in the high data density areas, the relative count of granules in large and smaller groupings of data evens out. In other words, granulation substitutes explicit enumeration (that unduly affects FCM) with an update of the cardinality attribute associated with individual granules (that is transparent to FCM). It can be seen that the clustering of level-2 and level-3 granulated data [see Fig. 9(c) and (d)] does not have any problems associating prototypes with the two smaller data groupings. This is a significant result that illustrates how data granulation complements fuzzy clustering.

V. Granular Time Series

The concept of information granulation opens up a new avenue of signal processing both in terms of signal representation and modeling relationships between the granular entities. In this section, we look at two possible approaches to capturing the dynamics of time-series and compare it to the granulation approach described in the previous sections.

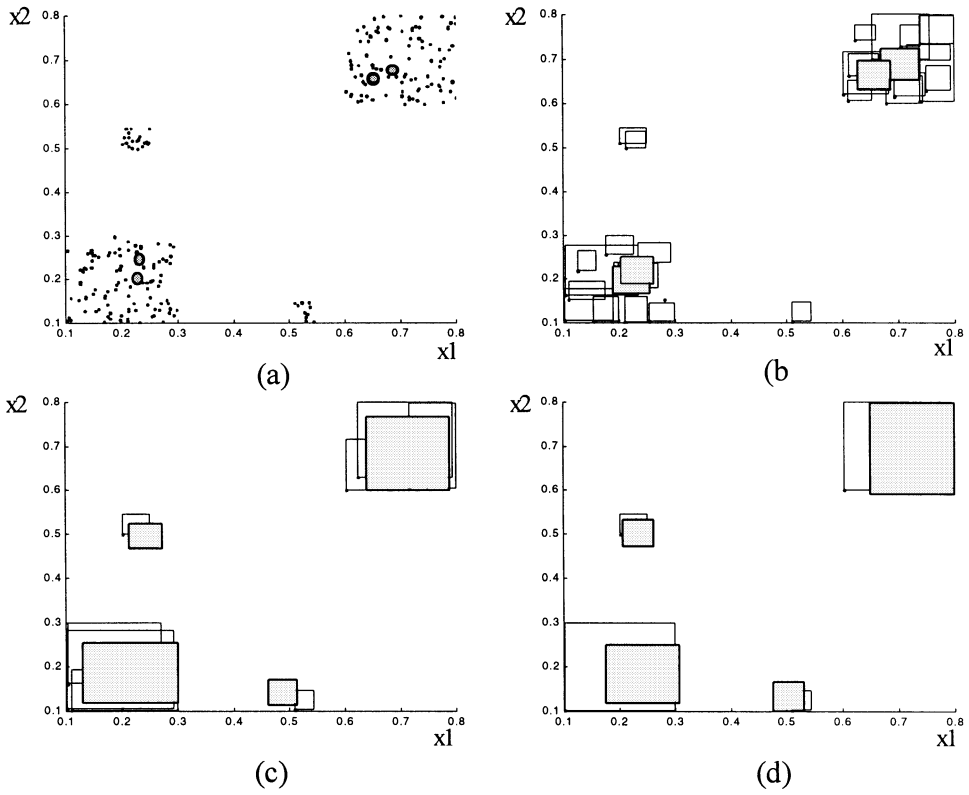

Fig. 9. Clustering of granular data performed using FCM. (a) Reference case of clustering the original 230 data points (the size of prototypes has been exaggerated for the sake of the clarity). (b) Clustering of level-1 granulated data (27 granules). (c) Clustering of level-2 granulated data (nine granules). (d) Clustering of level-3 granulated data (six granules).

TABLE I
COORDINATES OF FCM PROTOTYPES AS ILLUSTRATED IN FIG. 9

Case	Prototype	Prototype coordinates			
Reference case – original data	P1	0.6869	0.6811	0.6869	0.6811
	P2	0.6536	0.6593	0.6536	0.6593
	P3	0.2255	0.2033	0.2255	0.2033
	P4	0.2324	0.2482	0.2324	0.2482
Level-one granulated data	P1	0.1987	0.1873	0.2518	0.2381
	P2	0.6772	0.6538	0.7364	0.7226
	P3	0.2013	0.1966	0.2548	0.2470
	P4	0.6275	0.6328	0.6859	0.6982
Level-two granulated data	P1	0.2124	0.4718	0.2659	0.5214
	P2	0.4652	0.1206	0.5042	0.1656
	P3	0.6422	0.6086	0.7769	0.7674
	P4	0.1293	0.1290	0.2902	0.2501
Level-three granulated data	P1	0.1792	0.1267	0.2965	0.2446
	P2	0.4829	0.1113	0.5199	0.1572
	P3	0.6546	0.5983	0.7924	0.7935
	P4	0.2061	0.4821	0.2575	0.5312

A. Time-Domain Granulation

The state-space granulation (and subsequent clustering) described in the previous two sections is now compared to the direct approach in which the information granules are formed by predefined sets of consecutive elements of the time-series. This approach is referred to as *time-domain granulation*. The simplest strategy within this approach is to define a "window of observation" and to evaluate an appropriate granular representative within such a segment of time-series. Fig. 10 illustrates the principle of this approach.

Bearing in mind our earlier comments on the bounds-encoding of information granules, we can formalize the time-domain granulation as a mapping of the original data set $\mathbf{X} = \{x^1, x^2, \ldots, x^N\}$ onto a set of intervals $\mathbf{I} = \{I^1, I^2, \ldots, I^G\}$ where N is a number of elements in the time-series and G is the number of granules.

Individual granules I^k are described as follows:

$$I^k = \left(\min_{j \in \Omega_k} (x^j), \max_{j \in \Omega_k} (x^j) \right) \tag{11}$$

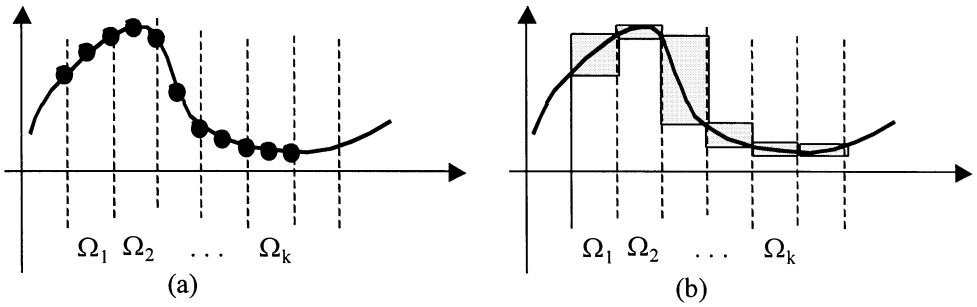

Fig. 10. Simple time-domain granulation: (a) time-series and (b) information granules formed with $\omega_o = 3$.

where

$$\Omega_k = \{i\colon \omega_o(k-1)+1 \le i \le \omega_o k\} \tag{12}$$

and $k = 1, 2, \ldots, G$, $\omega_o G \le N$, ω_o is a granulation window. Equations (11) and (12) are easily generalized to multidimensional time-series by applying the $\min$- and $\max$-operations to all coordinates of the original data, in which case we have

$$I^k = \left(\min_{j\in\Omega_k}(x_1^j),\ \max_{j\in\Omega_k}(x_1^j),\ \min_{j\in\Omega_k}(x_2^j),\right.$$
$$\left.\max_{j\in\Omega_k}(x_2^j),\ \ldots,\ \min_{j\in\Omega_k}(x_n^j),\ \max_{j\in\Omega_k}(x_n^j)\right). \tag{13}$$

It is clear that the mapping of $\mathbf{X}$ onto $\mathbf{I}$ involves the increase of dimensionality of the pattern space from $\mathbf{R}^n$ to $\mathbf{R}^{2n}$, where n is a dimension of patterns. The set of intervals $\mathbf{I}$ now represents a granulated information from the original time-series. As such, the intervals I^k can be used to extract specific knowledge about the system at the higher level of abstraction compared to the one afforded with the original time-series. In order to check the effectiveness of knowledge abstraction based on the information granules (13), we apply the time-domain granulation to the synthetic data of Fig. 5. To make a fair comparison of our algorithm with time-domain granulation, we select ω_o to be 8, 25, and 3, 8 so as to ensure that the granulation returns 27, nine, and six granules, respectively. However, we start first with granulation windows ω_o equal 2, 3, 5, and 6, which imply formation of 115, 76, 46, and 38 granules, respectively. Results of the granulation and subsequent FCM clustering (four clusters) are presented in Fig. 11.

One fact that is immediately obvious from these results is that time-domain granulation is very sensitive to the selection of ω_o. If the window of observation is aligned well with the boundaries of significant changes in the time-series, as is the case for $\omega_o = 2$ and $\omega_o = 5$, the resulting granulation gives a good abstraction of the original data and the FCM clustering identifies prototypes that represent the data well. However, in a more typical case, when the window of observation includes data that belongs to two different data groupings, the time-domain granulation generates large, unrepresentative granules that adversely affect subsequent FCM clustering. Such performance is, in fact, to be expected since the transition of the time-series from one data grouping to another represents a high-frequency signal that is not matched by the sampling frequency of the granulation window (as defined by the inverse of the window's width). Therefore, the result is an irretrievable loss of information (Shannon theorem, [24], [25]) that demonstrates itself here through large, low-specificity information granules. The FCM prototypes that are build on such granules also have low specificity and they occupy most of the pattern space [see Fig. 11(d)–(f)]. We conclude, therefore, that a simple time-domain granulation should be avoided if there is no additional information available concerning the appropriateness of width of the specific granulation window.

A refinement of the simple time-series granulation approach has been proposed by Das *et al.* [8]. The extended method considers fixed-length subsequences of the series rather than just individual data items (see Fig. 12). The subsequences are represented as data points in the augmented input space that has dimension defined by the length of the subsequences (ω_o). Subsequences that have similar "shape" are represented as nearby points in the augmented space and can be clustered using some appropriately defined distance function. However, because the property of shape similarity should be independent from the actual values of time-series, the subsequences need to be normalized to a fixed range (typically [0, 1]) before they are clustered. It is worth noting that every change of the width of the granulation window (ω_o) implies the need for a renormalization of the subsequences. The clustering process can be seen as a formation of a "vocabulary" (codebook) of information granules that are viewed as conceptual entities aimed at capturing the original numeric signal [28].

Clearly, the approach generalizes to multidimensional time-series. In this case, the subsequences q^j are formed by patterns $x^i \in \mathbf{R}^n$, where n is a dimension of individual patterns. They are therefore elements of $(\omega_1)^n$-dimensional space, where ω_1 is a number of patterns formed from ω_0 sequences in each dimension (typically $\omega_1 = 4$ for $\omega_0 = 3$). The clusters of subsequences are therefore hyperboxes in $R^{2(\omega_1)^n}$. Unfortunately, the exponential increase of the dimensionality of the "shape-space" makes this impractical.

B. Phase-Space Granulation

In this paper, we have adopted an alternative approach to capturing the nature of subsequences of time-series that avoids undue augmentation of the input space. We characterize the "shape" of subsequences by a range of gradient angles between the first and every other pattern in the subsequence (in Fig. 13). This results in an interval (hyperbox) description of "shape" that is fully compatible with the interval (hyperbox) description of the time-series values, as defined in (11)–(13). The advantage

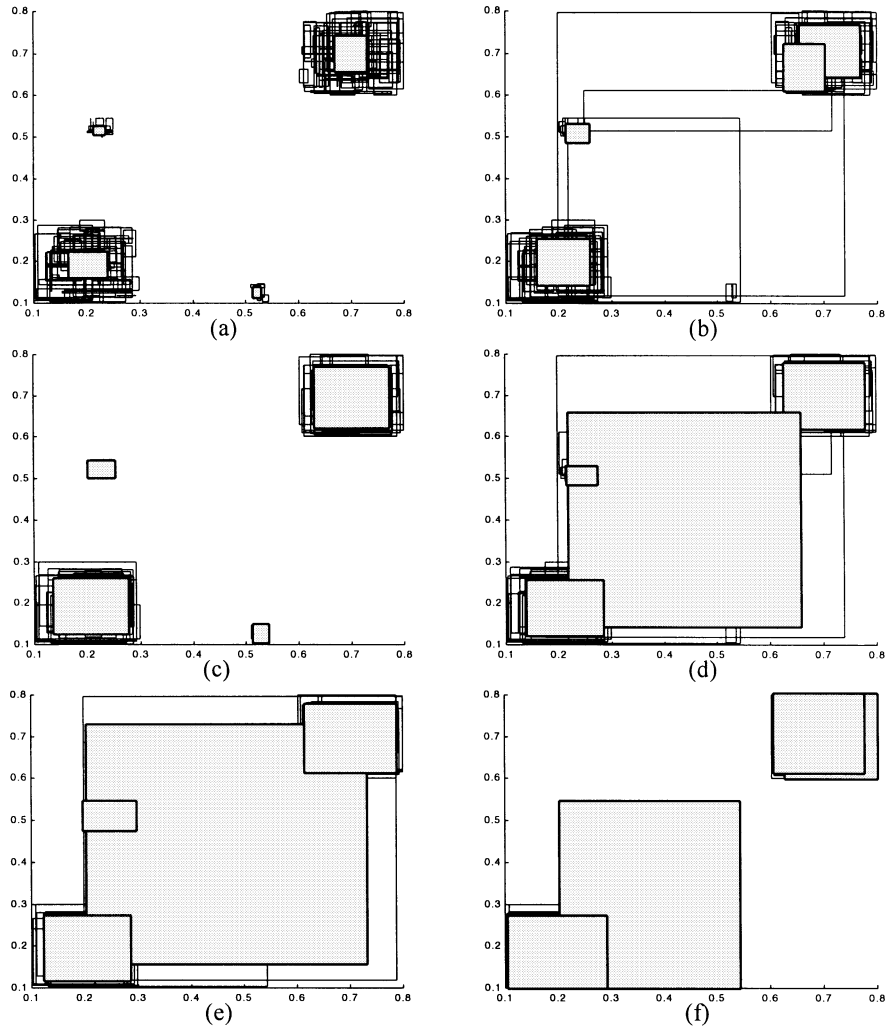

Fig. 11. Results of time-domain granulation and FCM clustering: (a) $\omega_o = 2$ (115 granules), (b) $\omega_o = 3$ (76 granules), (c) $\omega_o = 5$ (46 granules), (d) $\omega_o = 6$ (38 granules), (e) $\omega_o = 8$ (28 granules), and (f) $\omega_o = 25$ (nine granules).

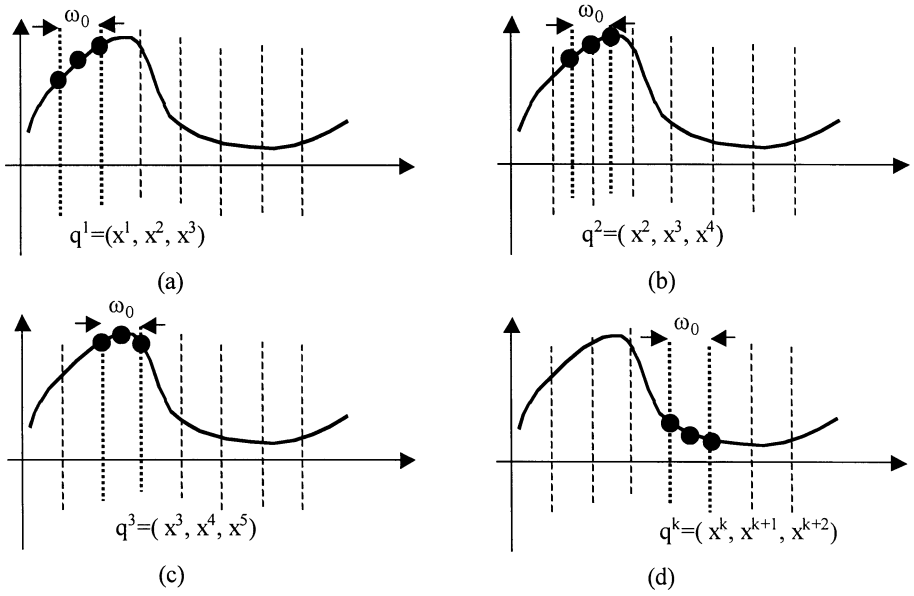

Fig. 12. Granulation of time-series $x^i \in \mathbf{R}^n$ $(i = 1, 2, \ldots, N)$ into fixed-length subsequences $q^j \in R^{(\omega_1)^n}$ $(j = 1, 2, \ldots, N - \omega_o + 1)$; $\omega_o = 3$; $\omega_1 = 4$.

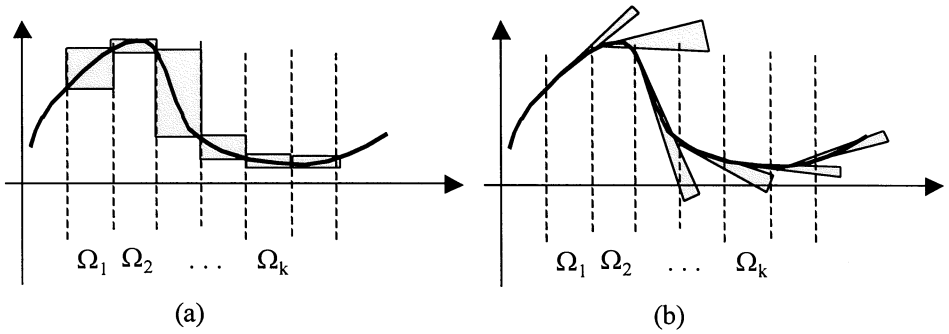

Fig. 13. Phase-space granulation: (a) intervals of time-series values and (b) intervals of gradient angles (granulation window $\omega_o = 3$; see Fig. 10).

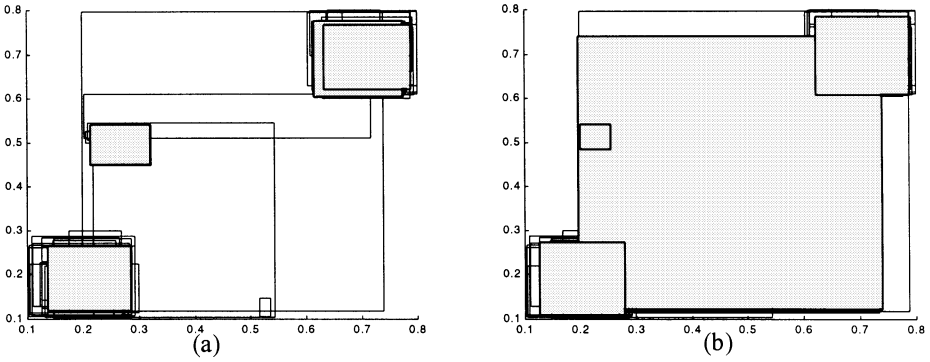

Fig. 14. Phase-space granulation and FCM clustering: (a) $\omega_o = 6$ (38 granules) and (b) $\omega_o = 8$ (28 granules).

of this granulation is that subsequent clustering does not imply any further increase of the input space dimension.

Since the intervals of time-series values in each granulation window are already contained within the [0, 1] range, only the intervals of gradient angles need to be normalized from $[-\pi/2, \pi/2]$ to [0, 1].

We can formalize the phase-space granulation as a mapping of the original data set $\mathbf{X} = \{x^1, x^2, \ldots, x^N\}$ onto a set of hyperboxes $\mathbf{H} = \{H^1, H^2, \ldots, H^G\}$ where N is a number of elements in the time-series and G is the number of granules. A hyperbox H^k is formed as a Cartesian product of two intervals; $H^k = I^k \times J^k$, where I^k is an interval of time-series values and J^k is an interval of gradient angles in the kth granulation window

$$I^k = \left(\min_{j \in \Omega_k}(x^j), \max_{j \in \Omega_k}(x^j) \right) \tag{14}$$

$$J^k = \left(\min_{j \neq l;\, j,\, l \in \Omega_k} (norm(grad(x^j, x^l))), \max_{j \neq l;\, j,\, l \in \Omega_k} (norm(grad(x^j, x^l))) \right) \tag{15}$$

where $grad(.)$ is an angle-valued gradient function and $norm(.)$ is a normalization function. The granulation window Ω_k is defined as in (12) and the generalization of the granulation to a multidimensional time-series is analogous to (13). In this general case, the dimension of the input space is $4n$ (where n is a dimension of x^k) and the subsequent clustering of hyperboxes H^k does not imply any further increase of the dimension of the pattern space ($H^k \in \mathbf{R}^{4n}$). It is worth emphasizing that increasing the width of the granulation window ω_o reduces the number of granules to N/ω_o while maintaining the dimensionality of the input space. Consequently, the computational complexity of the subsequent FCM is reduced by a factor $(\omega_o)^2$. This is in a sharp contrast to the granulation proposed in [8], where the increase of the width of the granulation window reduces the number of input patterns to N-ω_o, but it increases the dimensionality of the input space by a factor ω_o, thus increasing the computational complexity of FCM also by a factor ω_o.

Tests performed on the synthetic time-series data indicate that while the inclusion of the gradient of the time-series goes some way toward filtering out unrepresentative granules, the FCM clustering of phase-space granulated data is broadly comparable to the results obtained with simple time-domain granulation. Fig. 14 provides an illustration of granulation and FCM clustering obtained for $\omega_o = 6$ and $\omega_o = 8$. Results illustrated in Fig. 14(a) are an improvement on the results from Fig. 11(d) but there is very little (if any) improvement discernible in Fig. 14(b) compared to Fig. 11(e).

VI. Numerical Studies

In this section, we apply granular analysis to a time-series of traffic queues collected by an urban traffic control (UTC) system. The data represents traffic on a crossroad in Mansfield (Nottinghamshire, U.K.) during a morning rush hour. The topology of the selected crossroad is illustrated in Fig. 15. The junction is controlled by an adaptive system called the *split-cycle-offset optimization technique (SCOOT)*, that attempts to maximize the traffic throughput of the junction by adaptively modifying the duration of the red/green signaling

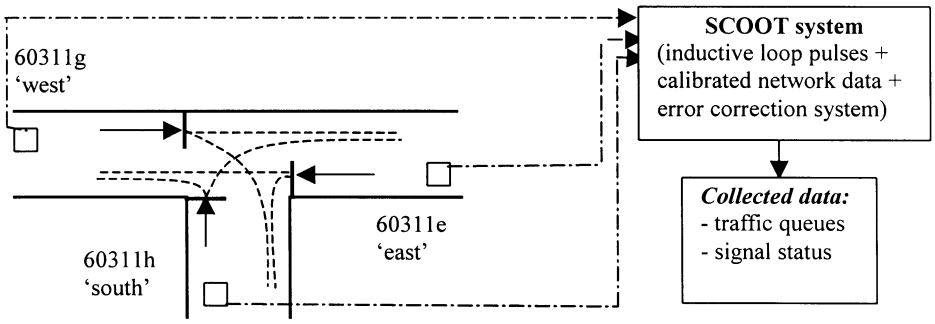

Fig. 15. Junction with three measured traffic flows (in Mansfield, U.K.).

stages. However, the details of SCOOT heuristics that implement the traffic signals optimization are not readily available since the system is a commercial product. This is unfortunate because the development of various high-level traffic management tasks such as in-car traffic information, variable message signs, and public transport information (all of which require predictions of traffic flows over extended time-scales) is critically tied to the SCOOT system itself [7], [17]. We shall show here that, by performing granular analysis of traffic data, it is possible to infer operational control rules that can provide a basis for the development of high-level traffic management tasks while, at the same time, leaving SCOOT fully in charge of detailed optimization of traffic signals.

While the full Mansfield SCOOT system involves some 40 intersections, for the sake of clarity of presentation, we limit ourselves to just one intersection, as illustrated in Fig. 15. The three inductive loops are the measuring devices that count discrete pulses generated by cars passing over them. The number of pulses generated by a car is proportional to the length of the car and inversely proportional to its speed. Therefore, a small vehicle moving slowly and a large vehicle moving quickly may generate the same number of pulses. This is actually a very advantageous property of this type of measuring devices because it enables focusing on generic rather than specific vehicles. The inductive loop measurements are combined with real-time readings of traffic signal status and also the calibrated travel times between each inductive loop and its corresponding stop-line. On this basis, SCOOT is able to estimate the number of vehicles that will arrive at the stop-line during the red signaling stage. This estimate, updated in real time, is referred to as "traffic queue measurement." Since the integration of inductive pulses is prone to systematic error, there are additional inductive loops (not shown in Fig. 15), which are used to reset this error to zero for some specific queue length. In effect, the SCOOT system has a built-in "safety net" for the traffic queue measurements. By monitoring the "discharge flows" from the stop-line during the green signaling stage, SCOOT also accounts for the queue remaining from the previous signaling stage in the derived traffic queue measurements.

In the first instance, we analyze a three-dimensional (3-D) time-series of changes of traffic queues in the links "60 311g," "60 311e," and "60 311h." We will refer to these links as "west," "east," and "south," respectively. Clearly, the expectation is that the relative changes of traffic queues in any pair of links will reflect the embedded "rules of operation" of this specific junction.

The original time-series are presented in Fig. 16. They consist of 705 discrete measurements for each inductive loop. The readings are time-aligned and form a 3-D vector of system states for 705 time instances. In order to achieve consistent representation of data points and intervals, we increase the dimensionality of the pattern space from three to six (as illustrated in Fig. 8). We apply the granulation and clustering to this six-dimensional state vector and visualize the results by three 2-D projections. It is worth mentioning that while the granulation and clustering operates on data that is normalized to a unit hypercube ($[0, 1]^3$), the results are converted back to the original data values.

Figs. 17–19 reveal some interesting properties of the system. First, the plot of the original data and FCM prototypes (see Fig. 17) is somewhat surprising in that significant relationships between traffic in various directions appear not to be fully represented by the prototypes. This is because the data grouped along the axes ("west" $= 0$, "east" $= 0$, and "south" $= 0$) exerts undue influence on the FCM algorithm. The situation changes quite dramatically when we consider granulated data (see Figs. 18 and 19). In this case, the prototypes cover a larger proportion of data and become more representative of the overall operation of the junction.

Second, the FCM prototypes reveal something that is not obvious from the plot of original data, namely, that the queue changes on the "west" and "south" link are significantly larger than on the "east" link. The examination of the physical road layout reveals that the "south" and "west" links have separate "right-turner lanes" and the corresponding inductive loops are spreading there across two rather than just one lane. While the essence of this relationship has been captured by the FCM prototypes built both on original and granulated data, the granular version of FCM appears to deliver more representative results in that the ratio of "south"/"east" and "west"/"east" is approximately 2/1 for the granular prototypes and is approximately 4/1 for the point-size prototypes.

Third, the granular FCM prototypes, unlike the standard ones, reveal that there is a significant "right-turner" traffic on the "west" link. This is represented by a prototype that assumes positive values (queue increases) on the "west" link when there are negative values (queue decreases) on the "east" link (compare the "east"–"west" plots). Notice that there is no similar effect caused by the "right-turners" on the "south" link, which means that the operation of the "south" link is mutually exclusive with "west" and "east" links.

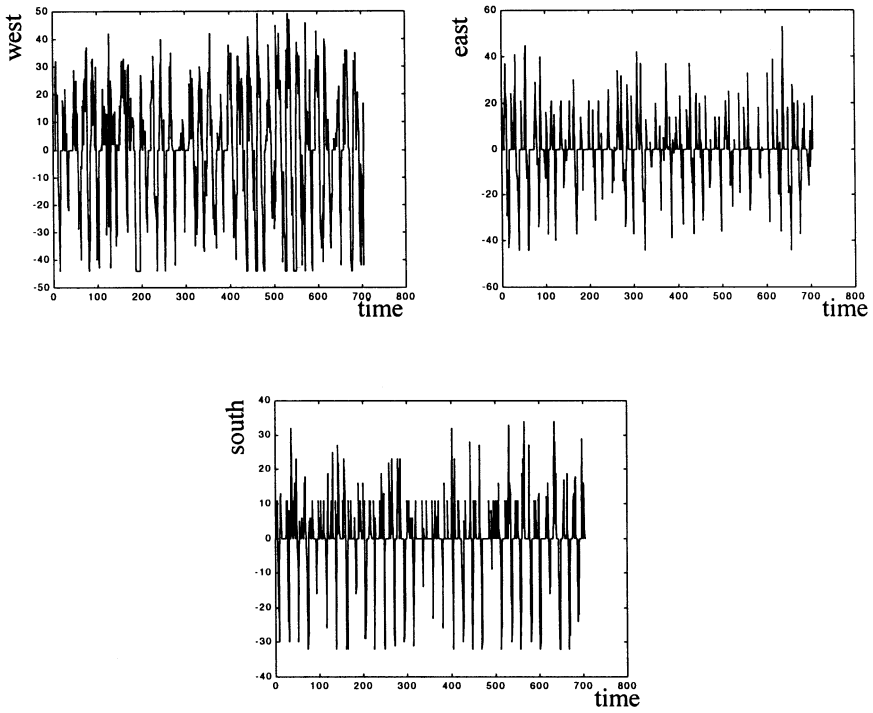

Fig. 16. Time-series of changes to traffic queues on the three stop-lines.

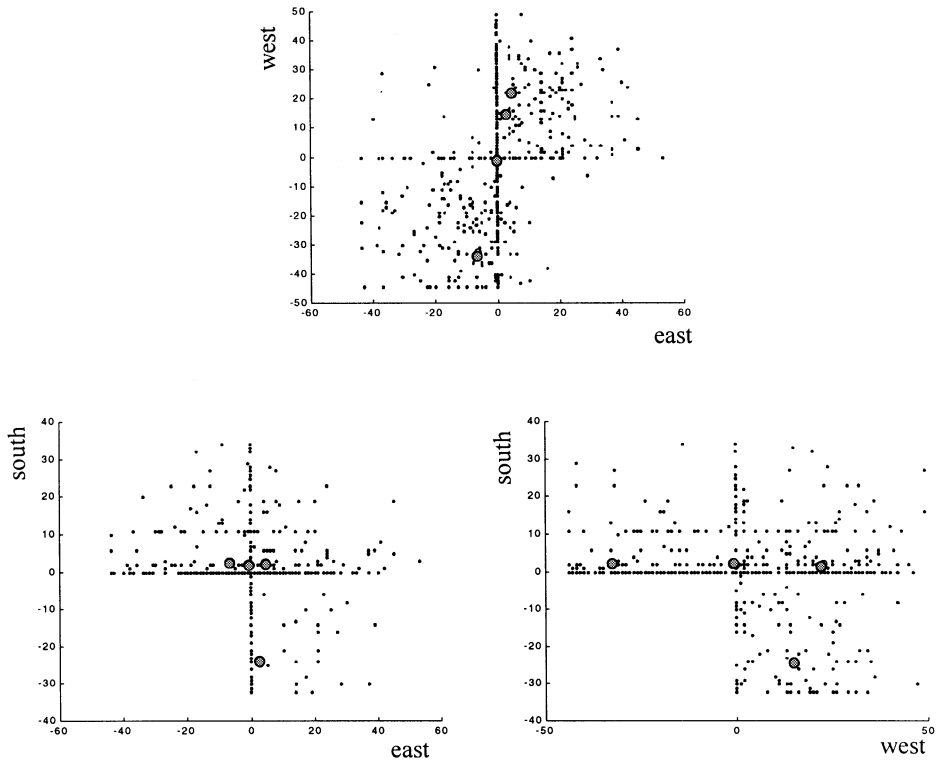

Fig. 17. FCM granular prototypes for the level-1 granulated data (705 data points).

Although we have demonstrated, in the previous section, that the time-domain granulation produces much inferior results, we enclose here, for completeness, results obtained for such granulated traffic data. In order to achieve comparability of the results we select $\omega_o = 6$, giving 117 granules, which compares to 114 granules from Fig. 18. As expected, the time-domain granulation results are poor. Fig. 20 shows the FCM prototypes build on time-domain granulated data. The specificity of prototypes is all but lost and while one can discern some similarity in the distribution of prototypes none of the earlier detailed analysis of the operation of the junction seem possible. In fact, the prototypes depicted on the "south"/"east" and "south"/"west" projections

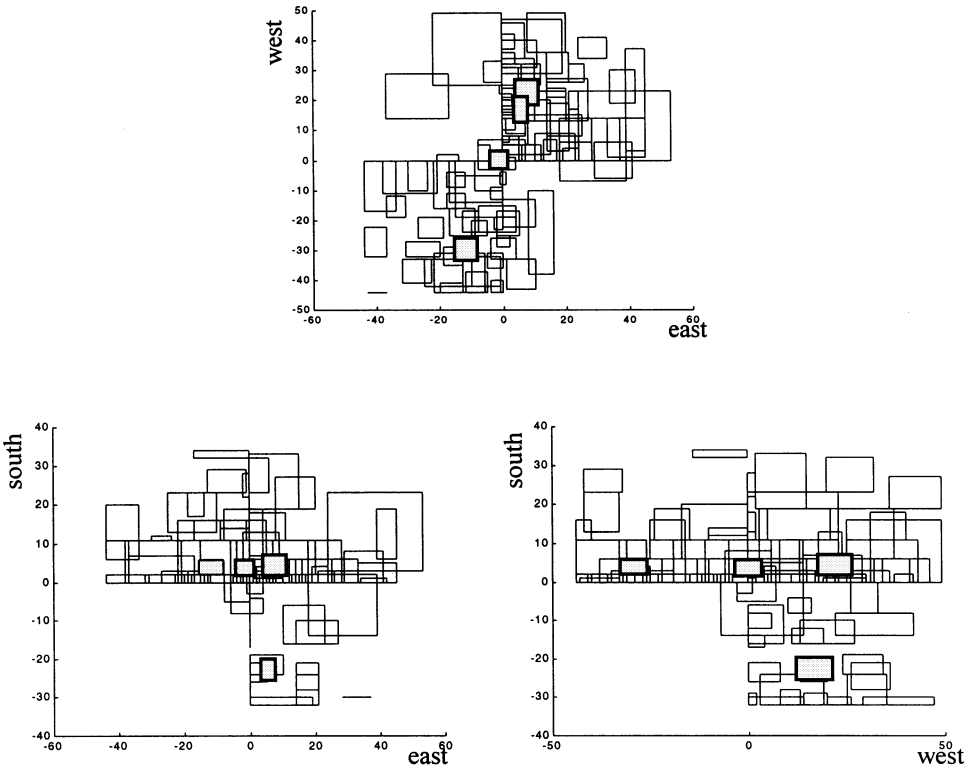

Fig. 18. FCM granular prototypes for the level-1 granulated data (114 granules).

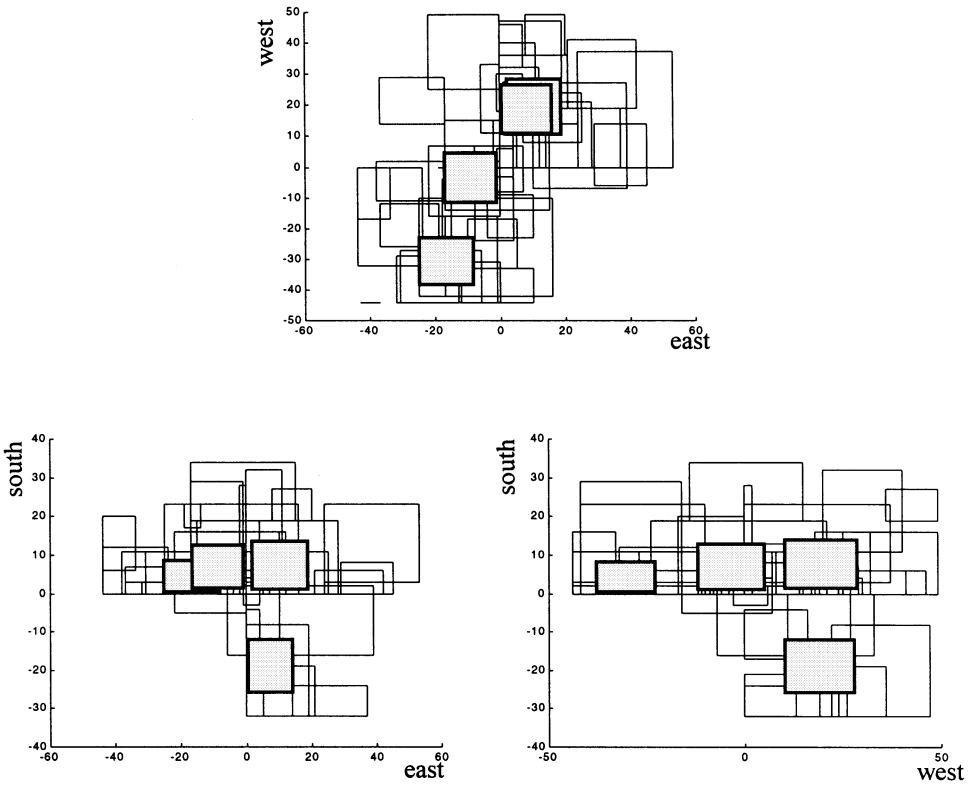

Fig. 19. FCM granular prototypes for the level-1 granulated data (46 granules).

indicate that it is possible to have simultaneous queue reduction on the corresponding "south"–"east" and "south"–"west" links. This is an erroneous indication since such operation of the junction would clearly lead to a collision and, as such, is specifically

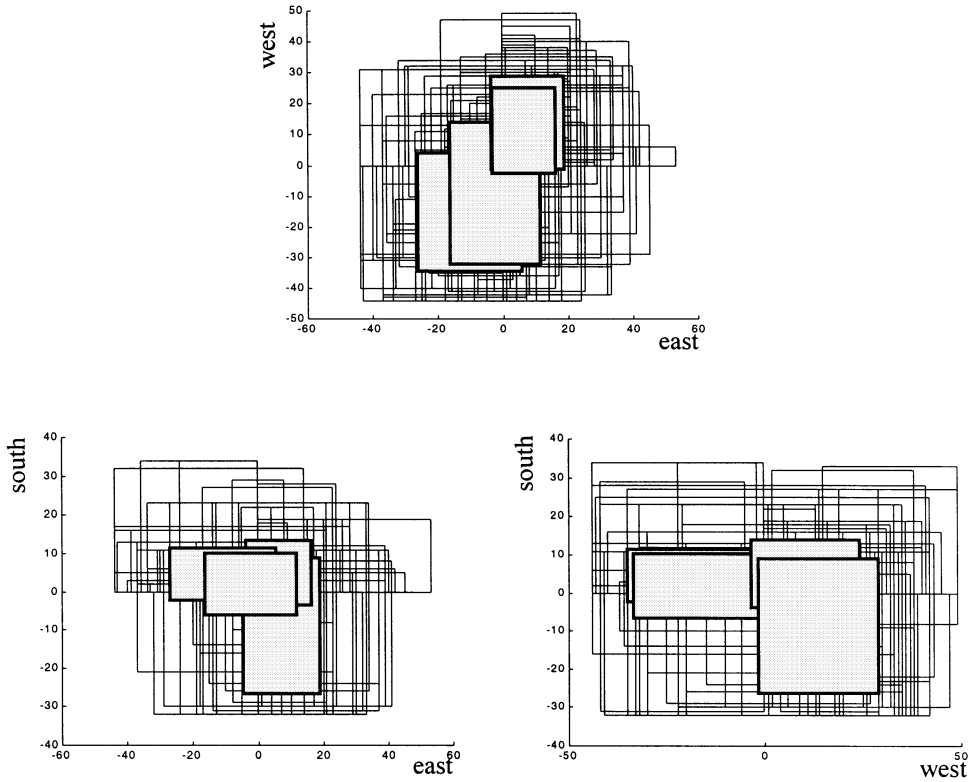

Fig. 20. FCM prototypes for the time-domain granulated traffic data ($\omega_o = 6$ giving 117 granules).

prevented by traffic signals. Of course, time-domain granulation can deliver significantly better results if narrower granulation windows are used. However, this defeats the idea of granulation and even with $\omega_o = 2$ the results are not as crisply defined as those of Fig. 18.

The application of phase-space granulation to the traffic system data produces similar results to those obtained with simple time-domain granulation (see Fig. 20). The FCM prototypes build on phase-space granules are significantly less specific than the prototypes obtained with state-space granules and, as such, are not as well suited for system modeling purposes. We conclude therefore that the state-space granulation based on maximization of information density has a potential to be of benefit in many practical applications requiring efficient data abstraction.

VII. Concluding Remarks

In this paper, we have discussed a notion of granular data, elaborated on the recursive information granulation and assessed the quality of summarization of information granules through FCM clustering. The experiments involving both synthetic data and real-world traffic data illustrate the usefulness of this approach. The clustering method applied to granular data gives rise to granular prototypes. These, in contrast to numeric prototypes, are more user-oriented, reveal and deliver a compact characterization of the main relationships existing in the data.

The granulation of time-series exhibits several essential features.

1) It helps concentrate on a certain level of detail while ignoring (on purpose) more detailed relationships that may be pertinent only to the higher level of granularity.
2) The form of data segments allow to emphasize the essence of granulation (for instance, monotonic segments of data, segments of equal width, etc.)
3) It promotes a knowledge intensive, data mining-oriented approach to time-series.
4) By considering information granules, we can easily convert the problem into a hierarchy of manageable subproblems. Large, less specific information granules form a first level of analysis that could be afterwards refined by defining more specific information granules capturing more details and geared toward some specific analysis. Formally speaking, denoting an information granule at the higher, and more abstract level, by A, the more detailed analysis relies on information granules $B_1, B_2, \ldots, B_c$, such that all of them are included in A, $B_i \subset A$ and they "cover" A in the sense that $A = \bigcup_{i=1}^{c} B_i$.

While the experimental part of the study concentrated on multidimensional time-series, the same methodology applies to other multidimensional data such as images. Future work will investigate whether the hierarchical granulation in the multidimensional space of color/brightness/texture/shape produces image abstractions that are more closely related to human image processing.

Overall, the granular description is very much intuitive and qualitative and provides the designer/user with a general insight into the very nature of the phenomenon manifesting through this time-series. In this sense, this analysis concurs with a general agenda of qualitative modeling [29] and fuzzy qualitative modeling [37].

REFERENCES

[1] A. Bargiela, "Interval and ellipsoidal uncertainty models," in *Granular Computing*, W. Pedrycz, Ed. New York: Springer-Verlag, 2001.

[2] M. Berthold and K.-P. Huber, "Constructing fuzzy graphs from examples," *Intell. Data Anal.*, vol. 3, no. 1, pp. 37–54, 1999.

[3] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[4] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, ser. CA. San Francisco: Holden-Day, 1970.

[5] S. Chiu, "Method and software for extracting fuzzy classification rules by subtractive clustering," in *Proc. NAFIPS*, 1996, pp. 461–465.

[6] K. Cios, W. Pedrycz, and R. Swiniarski, *Data Mining Techniques*. Norwell, MA: Kluwer, 1998.

[7] C. Claramunt, B. Jiang, and A. Bargiela, "A new framework for the integration, analysis and visualization of urban traffic data within geographic information systems," *Transp. Res.—Part C*, pp. 167–184, 2000.

[8] Das *et al.*, "Rule discovery from time series," in *Proc. 4th Int. Conf. Knowledge Discovery and Data Mining*, 1998, pp. 16–22.

[9] E. Davis, "Constraint propagation with interval labels," *Artif. Intell.*, vol. 24, pp. 347–410, 1987.

[10] B. S. Everitt, *Cluster Analysis*. Berlin, Germany: Heinemann, 1974.

[11] B. Gabrys and A. Bargiela, "General fuzzy min–max neural network for clustering and classification," *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 769–783, 2000.

[12] Y. Hata and M. Mukaidono, "On some classes of fuzzy information granularity and their representations," in *Proc. ISMVL*, Japan, 1999, pp. 288–293.

[13] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis*. New York: Wiley, 1999.

[14] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[15] A. Kandel, *Fuzzy Mathematical Techniques With Applications*. Reading, MA: Addison-Wesley, 1986.

[16] D. L. Kenneth and L. A. Jeffrey, *Robust Regression. Analysis and Applications*. New York: Marcel Dekker, 1990.

[17] I. Kosonen, A. Bargiela, and C. Claramunt, "A distributed information system for traffic control," in *Proc. 10th ESS*, 1998, pp. 355–361.

[18] B. J. Kuipers, *Qualitative Reasoning*. Cambridge, MA: MIT Press, 1984.

[19] V. K. Madisetti and D. B. Williams, *The Digital Signal Processing Handbook*. Boca Raton, FL: CRC, 1998.

[20] I. Mani and M. T. Maybury, Eds., *Advances in Automatic Text Summarization*. Cambridge, MA: MIT Press, 1999.

[21] R. Moore, *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

[22] R. E. Moore, Ed., *Reliability in Computing: The Role of Interval Methods*. New York: Academic, 1988.

[23] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Norwell, MA: Kluwer, 1991.

[24] A. V. Oppenheim and A. S. Wilsky, *Signals and Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[25] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[26] W. Pedrycz, *Computational Intelligence: An Introduction*. Boca Raton, FL: CRC, 1997.

[27] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets*. Cambridge, MA: MIT Press, 1998.

[28] W. Pedrycz, "Fuzzy equalization in the construction of fuzzy sets," *Fuzzy Sets Syst.*, vol. 119, no. 2, pp. 321–327, 2001.

[29] J. F. Sowa, *Knowledge Representation. Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks/Cole, 2000.

[30] H. Thiele, "On algebraic foundations of information granulation III, investigating the HATA-MUKAIDONO approach," in *Proc. ISMVL*, 2000.

[31] L. A Zadeh, "Fuzzy sets and information granularity," in *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds. Amsterdam, The Netherlands: North Holland, 1979, pp. 3–18.

[32] ——, "Fuzzy logic = computing with words," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 2, pp. 103–111, 1996.

[33] ——, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, pp. 111–117, 1997.

[34] ——, "From computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions," *IEEE Trans. Circuits Syst.*, vol. 45, pp. 105–119, 1999.

[35] L.A. Zadeh and J. Kacprzyk, *Computing With Words in Information/Intelligent Systems*, ser. Studies in Fuzziness and Soft Computing. Berlin, Germany: Physica-Verlag, 1999, vol. 33/34.

[36] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*. Norwell, MA: Kluwer, 1985.

[37] F. Herera and L. Martinez, "A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multiexpert decision making," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 227–234, Apr. 2001.

Andrzeg Bargiela (M'96) is Professor and Director of the Intelligent Simulation and Modeling Laboratory at the Nottingham Trent University, Nottingham, U.K. He is currently a Visiting Professor at the University of Alberta, Edmonton, AB, Canada. Since 1978, his main research focus has been on processing of uncertainty in the context of modeling and simulation of various physical and engineering systems. This research involves development of algorithms for processing uncertain information, investigation of computer architectures for such processing, and the study of information reduction through visualization. He is widely published and lectures on simulation and modeling at universities in Europe, the United States, and Japan.

Dr. Bargiela has been involved in the activities of the Society for Computer Simulation (SCS) European Council since 1994 and has been a Member of international program committees and/or Track Chair at many international conferences. In 1999, he was elected to serve as Chairman of the SCS European Conference Board and is an Associate Vice-President (conferences) of the SCS (United States). He is also Acting Chairman of the SCS European Council Board. He represented the Nottingham Trent University on the U.K. Engineering Professors Council between 1996 and 2002 and is a Member of the U.K. Conference of Professors and Heads of Computing.

Witold Pedrycz (M'88–SM'94–F'99) is a Professor and Chair in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also a Canada Research Chair (CRC) in computational intelligence. He is actively pursuing research in computational intelligence, fuzzy modeling, knowledge discovery and data-mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, relational computation, and software engineering, and has published numerous papers in this area. He is the author of seven research monographs covering various aspects of computational intelligence and software engineering.

Dr. Pedrycz has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He currently serves as an Associate Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS and IEEE TRANSACTIONS ON FUZZY SYSTEMS.