Multisignal 1–D compression by F–transform for wireless sensor networks applications

Matteo Gaeta[a], Vincenzo Loia[b], Stefania Tomasiello[b]

[a] *Department of Computer Engineering, Electrical Engineering and Applied Mathematics, University of Salerno, 84084, Fisciano, Italy*
[b] *CORISA, Department of Computer Science, University of Salerno, 84084, Fisciano, Italy*

Abstract

In wireless sensor networks a large amount of data is collected for each node. The challenge of transferring these data to a sink, because of energy constraints, requires suitable techniques such as data compression. Transform–based compression, e.g. Discrete Wavelet Transform (DWT), are very popular in this field. These methods behave well enough if there is a correlation in data. However, especially for environmental measurements, data may not be correlated. In this work, we propose two approaches based on F–transform, a recent fuzzy approximation technique. We evaluate our approaches with Discrete Wavelet Transform on publicly available real–world data sets. The comparative study shows the capabilities of our approaches, which allow a higher data compression rate with a lower distortion, even if data are not correlated.

Keywords: Data compression, Wireless Sensor Networks, F–transform, Least–squares

1. Introduction

Data compression can be regarded as one of the enabling technologies for several aspects in fields such as multimedia (for audio and video processing) and, thanks to the grown popularity of wireless sensor networks (WSNs),

Email addresses: mgaeta@unisa.it (Matteo Gaeta), loia@unisa.it (Vincenzo Loia), stomasiello@unisa.it (Stefania Tomasiello)

November 8, 2014

The published version of this manuscript is available at https://doi.org/10.1016/j.asoc.2014.11.061

measurements. The latter case is the one requiring more attention due to some constraints imposed by the structure of a WSN.

Each node of a WSN can perform three main tasks: collecting data with its sensor suite, processing data with an onboard microprocessor, and sharing data with neighboring nodes using its radio.

In all these tasks, nodes are required to be relatively inexpensive, in terms of power supply, memory capacity, communication bandwidth, and processor performance [1].

Generally, the most part of energy consumption is due to radio communication [2], so reducing the number of bits to be transmitted by means of compression techniques can have a positive effect on communication energy costs.

There are many compression techniques avilable in WSNs: a comprehensive survey can be found in [3]. Briefly, data compression techniques for WSN can be classified into distributed, which exploit spatial correlation, and local, which exploit temporal correlation. These approaches are usually used for dense and sparse networks respectively [4], even if in dense networks, spatio–temporal correlation allows the use of both the distributed and local approaches (e.g. [5]). Distributed approaches are also well suited for multivariate data [6]. As suggested by Wagner [7] one can refer to a spatio–temporal processing, i.e. to a distributed approach, when the ratio between the number of nodes and the length of the time-series stored at each node is high. A popular scheme, implemented both as a distributed and a local approach, is the transform-based compression [7],[3]. In transform–based schemes, raw data are transformed into a set of coefficients of suitable basis functions, for example wavelet functions, which are used to reconstruct the signal at the sink. Transform–based compression techniques can be viewed as either transform driven or routing driven. The last ones turn out to be more efficient for dense networks, since the transforms are computed as data are routed to the sink along efficient routing paths. In any case, the transforms can be easily integrated within existing routing protocols, as for example the SenZip compression tool [8]. Well–known transform–based algorithms such as the discrete cosine transform (DCT) and the discrete wavelet transform (DWT) perform well when adjacent data have similar values. This is a frequent case for indoor environments, which usually present strongly spatially– and temporally–correlated data, but this could not be true for outdoor environments [3]. Besides, it should be pointed out that compression techniques aim to reduce redundancy in order to increase energy efficiency, but redun-

dant deployment is necessary in the case of node or link failure, i.e. to ensure robustness, especially for in situ deployments in austere environments such as mountains, where failures are common phenomena.

Here we propose a robust local transform–based scheme in order to address all the issues above, by comparing two compression schemes based on fuzzy transform (F–transform) with a wavelet based scheme. F–transform was proposed by Perfilieva [9] as a fuzzy approximation technique. It substantially expresses a functional dependency as a linear combination of basic functions and it can be used for the solution of direct and inverse problems or least–squares approximations [10]. The major applications of the F–transform are in image processing, e.g. [11], [12], [13], [14], [15].

Herein, as a first approach, we followed the idea of compressed blocks proposed in [12]. As an alternative approach, we extended to the two–dimensional case the least–squares (LS) approximation discussed in [10].

We experimentally tested these approaches in the task of addressing data compression in two WSN applications where in some cases data lack correlation, i.e. two glacier monitoring deployments, namely, Patrouille des glacier (PDG) and Plaine Morte glacier (PM) [16]. For these deployments a small number of nodes was used and several kinds of data were collected by each node. These deployments were also considered in [17], where the authors compressed only four sensor data, by using a version of the LZW algorithm, which is a dictionary–based compression algorithm building a dictionary of repetitive patterns; by means of their approach they did not found good results for the PDG surface temperature.

We compared our results with the ones obtained by means of the wavelet–based approach, by considering accuracy (i.e. distortion), data compression rate (CR), computational complexity. It should be pointed out that for a fixed network, the energy consumption of communication when using compression scales according to $(CR - 1)$ [3], so it is desiderable having higher values of CR, but for classical transform–based techniques such as DWT the higher CR the higher distortion. By our F–transform based schemes we found a high enough value of CR with a lower distortion, if compared with DWT.

The paper is structured as follows: Section 2 provides theoretical background, explaining how the proposed approaches work, also by means of a simple example; Section 3 is devoted to numerical experiments and finally Section 4 gives some conclusions.

2. Data compression in WSNs

Let X_i be an attribute (for the sake of illustration a scalar) observed by a node in the sensor network, e.g. an environmental property being sensed by the node, such as temperature. The observed values of all the attributes $X_1, \ldots, X_n$ can be arranged in a vector, the networked data vector, which can be very large. The networked data are periodically collected at the base station, at a certain frequency. This has a cost, i.e. the total energy involved by the data collection process, to be minimized. Data compression in WSNs aims to optimize this total cost, by converting an input data stream into another one with fewer bits. It should be pointed out that the total energy is usually approximated by the communication energy, which rapresent the largest part of it. There are several decentralized compression strategies that can be utilized. One possibility is that the correlations between data at different nodes are known a priori.

Let X_i^t be a random variable that denotes the value of X_i at time t and let $H(X_i^t)$ denote the information entropy of X_i^t. In most part of the sensor network deployments, the data generated by the sensor nodes are typically highly correlated both in time and in space, i.e. $H(X_i^{t+1}|X_i^t) << H(X_i^{t+1})$ and $H(X_1^t, \ldots, X_n^t) << H(X_1^t) + \ldots + H(X_n^t)$. These correlations can be captured by means of some predictive models using either prior domain knowledge or historical data traces. Unfortunately, in many applications, prior knowledge of the precise correlations in the data is unavailable. Furthermore, in some cases there could not be any correlation between data, such as for many deployments in outdoor environments.

Data correlation is one of the typical features of compression for WSNs. The other features are:

- distortion, which occurs in the case of lossy compression, i.e. when exact reconstruction of the original data after decompression cannot be achieved, unlike lossless algorithms; Mean Square Error (MSE) is a natural distortion metric;

- data aggregation, which is involved in some applications, where only a summary of the sensor data is required, due for example to the use of statistical queries, such as MIN, AVG, MAX; in this case, the original sample values cannot be reconstructed from the summarized representation, but the communication overhead can be greatly reduced;

- symmetry, i.e. in the case of symmetric algorithms, the computational complexity of compression and decompression are similar, whereas they are

different in the asymmetric case; traditional schemes have usually higher computational complexity on the compression side, whereas in WSNs, it is desirable that compression has a lower complexity than decompression (usually performed at the sink);

- adaptivity, by modifying the compression operations and parameters in order to improve the performance for nonstationary data.

A detailed survey on all the aspects related to compression in WSNs was provided in [3].

In the next section, we introduce two data compression schemes, that allow a low distortion, with a computational cost of compression not so high with respect to decompression.

3. Data compression based on F–transform

The F–transform represents an efficient way to convert a functional problem into a respective problem of linear algebra. The approximate solution to the initial problem can be obtained via the inverse F–transform. The same ideas hold on for discrete problems by means of the concepts of discrete F–transform and inverse discrete F–transform. Since F–transform was introduced [9], several papers on the topic appeared [21]–[25]. In particular, in [24] a neural approach to the fuzzy partition construction was proposed, by using an unsupervised learning for determining the distribution of the nodes for non-uniform partitions. In [25] new types of F–transforms were presented, based on B–splines, Shepard kernels, Bernstein basis polynomials and Favard-Szasz-Mirakjan type operators.

Applications of the F–transforms in image processing seem to be prevailing, e.g. [11], [12], [13], [14], [15].

Another interesting application of F–transforms is in time series analysis [26]-[29], also by integrating the F–transform and the fuzzy tendency modeling [27] or the F–transform and fuzzy natural logic [28].

F–transforms were also used in data analysis [30],[31].

In [32] F–transforms combined with finite differences were used to numerically solve some classical partial differential equations (heat, wave, Poisson) in simple domains.

In [10] the relations between the least–squares approximation techniques and the F–transform were investigated, since least–squares approximations allow to handle some additional information on data such as the geometric information.

3.1. Preliminaries

We briefly recall some definitions. Let $I = [a, b]$ be a closed interval and $x_1, x_2, \ldots, x_n$, with $n \geq 3$, be points of I, called nodes, such that $a = x_1 < x_2 < \ldots < x_n = b$. A fuzzy partition of I is defined as a sequence $A_1, A_2, \ldots, A_n$ of fuzzy sets $A_i : I \to [0, 1]$, with $i = 1, \ldots, n$ such that

– $A_i(x) \neq 0$ if $x \epsilon (x_{i-1}, x_{i+1})$ and $A_i(x_i) = 1$;

– A_i is continuous and has its unique maximum at x_i;

– $\sum_{i=1}^{n} A_i(x) = 1, \quad \forall x \epsilon I$.

The fuzzy sets $A_1, A_2, \ldots, A_n$ are called basic functions and they form an uniform fuzzy partition if the nodes are equidistant.

The fuzzy transform (F–transform) of a function $f(x)$ continuous on I with respect to $A_1, A_2, \ldots, A_n$ is the n–tuple $[F_1, F_2, \ldots, F_n]$ whose components are

$$F_i = \frac{\int_a^b f(x) A_i(x) dx}{\int_a^b A_i(x) dx} \tag{1}$$

The function

$$f_{F,n} = \sum_{i}^{n} F_i A_i(x), \qquad x \epsilon I \tag{2}$$

is called inverse F–transform of f with respect to $A_1, A_2, \ldots, A_n$ and it approximates a given continuous function f on I with arbitrary precision, as stated by Theorem 2 in [9].

In many real cases, where the function f is known only at a given set of points $\{p_1, p_2, \ldots, p_m\}$, the discrete F–transform can be used and Eq. (1) is replaced by

$$F_i = \frac{\sum_{j=1}^{m} f(p_j) A_i(p_j)}{\sum_{j=1}^{m} A_i(p_j)}, \qquad i = 1, \ldots, n \tag{3}$$

Similarly, Eq. (2) is replaced by

$$f_{F,n}(p_j) = \sum_{i}^{n} F_i A_i(p_j), \qquad j = 1, \ldots, m \tag{4}$$

giving the discrete inverse F–transform.

The above concepts can be extended to functions in two variables, as one will see in the next subsection.

3.2. The blocks approach and the least–squares approximation

Here two approaches are examined: the first one follows the idea of compressed blocks proposed in [12], the second one extends to the two–dimensional case the least–squares (LS) approximation discussed in [10]. By following [12], the $N \times M$ data matrix $\mathbf{D}$ is subdivided in submatrices $\mathbf{D}_S$ with dimension $N(S) \times M(S)$, each one compressed to a block $\mathbf{F}_S$ of size $n(S) \times m(S)$ by means of the discrete F–transform. Obviously, the case of a single block is a particular case where no subdivision occurs.

The discrete F–transform of $\mathbf{D}_S$ with respect to $\{A_1, \ldots, A_{n(S)}\}$ and $\{B_1, \ldots, B_{m(S)}\}$ is given by the Hadamard product, i.e. in compact form

$$F^S = P^S o Q^S \tag{5}$$

being

$$\mathbf{P}_S = \mathbf{A}^T \mathbf{D}_S \mathbf{B} \tag{6}$$

and Q^S the matrix whose elements are given by $1/R^S_{kj}$, with

$$\mathbf{R}_S = \mathbf{A}^T \bar{\mathbf{I}}_S \mathbf{B} \tag{7}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the matrices with elements $A_k(i)$ and $B_l(j)$ respectively, $\bar{\mathbf{I}}_S$ is a $N(S) \times M(S)$ matrix with all unit elements.

The decompression of the block $\mathbf{F}_S$ is obtained by the discrete inverse F–transform, which is given by

$$\mathbf{D}^F = \mathbf{A}\mathbf{F}_S\mathbf{B}^T \tag{8}$$

In the LS approximation [10], the components of the discrete F–transform F^S_{kl} are replaced by the unknowns λ_{kl} and the discrete inverse F–transform is obtained by minimizing the error functional with respect to these unknowns. By writing the matrix of the errors as

$$\mathbf{E} = \mathbf{D} - \mathbf{A}\mathbf{\Lambda}\mathbf{B}^T \tag{9}$$

and by minimizing it with respect to the unknown elements of the matrix $\mathbf{\Lambda}$, one has

$$\mathbf{\Lambda} = \mathbf{K}^{-1}\mathbf{G}\mathbf{H}^{-1} \tag{10}$$

where

$$\mathbf{G} = \mathbf{A}^T\mathbf{D}\mathbf{B} \tag{11}$$

$$\mathbf{K} = \mathbf{A}^T\mathbf{A}, \qquad \mathbf{H} = \mathbf{B}^T\mathbf{B} \tag{12}$$

It should be pointed out that, since $\mathbf{A}$ and $\mathbf{B}$ are the Gram matrices associated to given sets of basis functions, they have full rank and $\mathbf{K}$ and $\mathbf{H}$ turn out to be positive definite matrices.

With regard to the blocks approach, the computational complexity for computing the components of the discrete F–transform F_{kl}^S can be estimated to be $O(n(S)M(S)(N(S)+m(S)))$. By considering one block and $N >> M$ (as usual for WSNs applications), one can write $O(nMN)$.

Since the computational complexity of the (one level) DWT is bounded by $O(NM)$ [3], for a computational convenience one has to select n as small as possible. In any case, current research is attempting to noticeably reduce the computational cost by means of multi–thread schemes.

For the LS approach, considering $N >> M$, one has a computational complexity of $O(n^3)$, which is reasonably higher than the one for the block approach.

3.3. A simple example

In order to give a clear explaination on how the proposed F–transform based techniques can be used for data compression, we present a simple example. We generated an 11×5 data matrix $\mathbf{D}$ by the function $\cos(i)sin(j)$, with $i = 1, 1.5, \ldots, 6$ and $j = 1, 1.5, \ldots, 3$. In this example, we considered the F–transform compression rate $\rho = (nm)/(NM)$.

The data matrix

$$\mathbf{D} = \begin{pmatrix} 0.454649 & 0.0595233 & -0.350175 & -0.674139 & -0.83305 \\ 0.538949 & 0.07056 & -0.415104 & -0.799137 & -0.987513 \\ 0.491295 & 0.0643212 & -0.378401 & -0.728478 & -0.900198 \\ 0.323356 & 0.0423342 & -0.249052 & -0.479462 & -0.592483 \\ 0.0762475 & 0.00998243 & -0.0587266 & -0.113057 & -0.139708 \\ -0.189529 & -0.0248134 & 0.145977 & 0.281028 & 0.347273 \\ -0.408902 & -0.0535341 & 0.314941 & 0.606307 & 0.749229 \\ -0.528162 & -0.0691477 & 0.406796 & 0.783142 & 0.967747 \\ -0.518109 & -0.0678316 & 0.399053 & 0.768236 & 0.949328 \\ -0.381205 & -0.0499079 & 0.293608 & 0.565239 & 0.69848 \\ -0.150969 & -0.0197651 & 0.116278 & 0.223852 & 0.276619 \end{pmatrix} \tag{13}$$

is first compressed to a 6×4 block, with a rate $\rho = 24/55 = 0.44$, by means of Eq. (5)

$$\mathbf{F} = \begin{pmatrix} 0.429156 & -0.0974936 & -0.588737 & -0.862984 \\ 0.410021 & -0.0931465 & -0.562486 & -0.824505 \\ 0.0636339 & -0.014456 & -0.0872961 & -0.12796 \\ -0.341258 & 0.0775253 & 0.468154 & 0.68623 \\ -0.432399 & 0.0982302 & 0.593185 & 0.869504 \\ -0.202434 & 0.0459881 & 0.277709 & 0.407072 \end{pmatrix} \tag{14}$$

The block $\mathbf{F}$ is then decompressed to the 11×5 block $\mathbf{D}^F$

$$\mathbf{D}^F = \begin{pmatrix} 0.429156 & -0.0203675 & -0.343115 & -0.6289 & -0.862984 \\ 0.419588 & -0.0199135 & -0.335466 & -0.614879 & -0.843744 \\ 0.410021 & -0.0194594 & -0.327816 & -0.600858 & -0.824505 \\ 0.236827 & -0.0112397 & -0.189346 & -0.347055 & -0.476233 \\ 0.0636339 & -0.00302003 & -0.0508761 & -0.0932512 & -0.12796 \\ -0.138812 & 0.00658794 & 0.110982 & 0.20342 & 0.279135 \\ -0.341258 & 0.0161959 & 0.27284 & 0.50009 & 0.68623 \\ -0.386828 & 0.0183587 & 0.309274 & 0.566871 & 0.777867 \\ -0.432399 & 0.0205214 & 0.345708 & 0.633651 & 0.869504 \\ -0.317416 & 0.0150644 & 0.253778 & 0.465153 & 0.638288 \\ -0.202434 & 0.00960743 & 0.161849 & 0.296654 & 0.407072 \end{pmatrix} \tag{15}$$

By using the LS approach with the same values of n and m, i.e. $\rho = 0.44$, we obtained the 6×4 matrix Λ by Eq. (10)

$$\mathbf{\Lambda} = \begin{pmatrix} 0.476013 & -0.018365 & -0.685305 & -0.868823 \\ 0.523692 & -0.0202045 & -0.753949 & -0.955849 \\ 0.0808756 & -0.00312026 & -0.116435 & -0.147615 \\ -0.434867 & 0.0167776 & 0.626069 & 0.793723 \\ -0.550365 & 0.0212336 & 0.792349 & 1.00453 \\ -0.163869 & 0.00632223 & 0.235919 & 0.299096 \end{pmatrix} \tag{16}$$

so, by substituting the matrix $\mathbf{F}$ with the matrix $\mathbf{\Lambda}$ in Eq. (8), one has

$$\mathbf{D}^F = \begin{pmatrix} 0.476013 & 0.0540349 & -0.351835 & -0.712181 & -0.868823 \\ 0.499852 & 0.0567411 & -0.369456 & -0.747849 & -0.912336 \\ 0.523692 & 0.0594473 & -0.387077 & -0.783517 & -0.955849 \\ 0.302284 & 0.034314 & -0.223427 & -0.452259 & -0.551732 \\ 0.0808756 & 0.00918066 & -0.0597776 & -0.121001 & -0.147615 \\ -0.176996 & -0.0200918 & 0.130823 & 0.26481 & 0.323054 \\ -0.434867 & -0.0493642 & 0.321423 & 0.650621 & 0.793723 \\ -0.492616 & -0.0559196 & 0.364107 & 0.737022 & 0.899127 \\ -0.550365 & -0.0624751 & 0.406791 & 0.823423 & 1.00453 \\ -0.357117 & -0.0405384 & 0.263956 & 0.534297 & 0.651814 \\ -0.163869 & -0.0186017 & 0.121121 & 0.245171 & 0.299096 \end{pmatrix} \tag{17}$$

Figure 1 shows the behaviour of the mean square error (MSE), computed over all the elements of the data matrix: by increasing the value of the rate ρ, the MSE decreases for both approaches, but the MSE obtained by the LS approach is lesser in any case.

4. Numerical results

As application examples, we considered two SensorScope deployments: Patrouille des glacier (PDG) and Plaine Morte glacier (PM) [16]. The PDG deployment had 10 locations whereas PM deployment had 13 locations. Both data sets contain data from several sensors, namely, ambient temperature (C), surface temperature (C), solar radiation (W/m^2), relative humidity (%), wind speed (m/s), wind direction (deg). For not available data we adopted zero value.

In these examples we used both the blocks approach, without subdivision of the data matrix, and the LS approach, by uniform fuzzy partitions, with sinusoidal [12] shaped basic functions.

In order to evaluate distortion, we computed the mean square errors MSE_{LS}, MSE_B, for the LS approach and the blocks approach, with two values of the data compression rate (CR). The data compression ratio can be defined as the ratio of the uncompressed data size, in bits, to the compressed size, also in bits; in local approaches CR is a node–level parameter [3].

We compared our results with the ones obtained by means of a multisignal DWT (MSE_W), by using two levels and the Haar wavelet. By using this wavelet–based compression one has $CR = 1.33$.

In order to exploit differences between the two classes of approach, we used the ratios $r_B = MSE_W/MSE_B$ and $r_L = MSE_W/MSE_{LS}$ for the two values of CR. The error values are referred to ambient temperature (AT), surface temperature (ST), solar radiation (SR), relative humidity (RH), wind speed (WS), wind direction (WD).

4.1. Example 1: PDG deployment

In this example, 10 stations collected weather–related data every 2 mins between April 16-20, 2008 and each node collected on average $3,000$ samples within the five–day period. In Table 1 the values of the ratios r_B and r_{LS} for $CR = 1.33$ (i.e. the same CR allowed by the wavelet–based compression) and $CR = 1.83$ are presented. As one can see, the MSE_W is on average higher than MSE_B and MSE_{LS} for each type of measurement; besides, especially for higher values of CR, there is no significant difference between the results by the blocks approach and the LS approach, so we prefer the blocks approach for its computational convenience. These results are detailed in Figures 2 and 3, where the behaviour of the MSE over the N nodes is presented for ambient temperature e wind speed, respectively. Similar behaviour was observed for the other kinds of measurements and the related figures are not reported for the sake of brevity. Figures 4–9 show a sample of the reconstructed data by the blocks approach with $CR = 1.83$ for the first node.

4.2. Example 2: PM deployment

This example is referred to a network, with 13 stations, which was deployed on the Plaine Morte glacier for a 5 day campaign, i.e. between March 12–16, 2007; each node collected on average $6,000$ samples about within the campaign. In Table 2 the values of r_B and r_{LS} for $CR = 1.33$ and $CR = 1.83$

Table 1: Example 1: rate r

r	CR	AT	ST	SR	RH	WS	WD
r_B	1.33	6.52	4	3.02	3.93	2.96	2.52
r_{LS}	1.33	10.92	5.86	3.97	5.55	4.05	3.26
r_B	1.83	3.25	1.87	1.95	2.41	1.82	1.61
r_{LS}	1.83	4.26	2.14	2.22	2.93	2.13	1.82

Table 2: Example 2: rate r

r	CR	AT	ST	SR	RH	WS	WD
r_B	1.33	6.51	4	3.02	3.93	2.96	2.52
r_{LS}	1.33	10.92	5.86	3.97	5.55	4.04	3.26
r_B	1.83	3.25	1.87	1.95	2.41	1.82	1.61
r_{LS}	1.83	4.26	2.14	2.22	2.93	2.13	1.82

are tabled; these values are high in any case, in reason of the low distortion of the F–transform based methods. Figures 10 and 11 show the MSE behaviour for surface temperature and relative humidity respectively (other figures are not reported for the sake of brevity). The good approximation by the F–transform based approches is confirmed, even for $CR = 1.83$. Figures 12–17 show a sample of the reconstructed data by the blocks approach with $CR = 1.83$ for the third node.

5. Conclusions

In this paper, for the first time, F–transform based compression methods have been used for wireless sensor networks applications. The proposed approaches belong to the class of transform–based methods, but unlike other schemes in this class, such as DWT, they allow a higher data compression rate with a lower distortion, even if there is no data correlation. We argue

that these are desiderable characteristics, due to the cost of the energy communication. Our conclusions are supported by a comparative study, where publicly available environmental data were used.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: A survey. Computer Networks 38(4) (2002) 393-422.

[2] G.J. Pottie, W.J. Kaiser, Wireless integrated network sensors. Communications of ACM 43 (2000) 51-58.

[3] M. A. Razzaque, C. Bleakley, S. Dobson, Compression in Wireless Sensor Networks: A Survey and Comparative Evaluation, ACM Transactions on Sensor Networks, 10(1) (2013) 5:1–43.

[4] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, K. Araki, Practical data compression in wireless sensor networks: A survey. Journal of Network and Computer Applications 35(1) (2012) 37-59.

[5] D. Chu, A. Deshpande, J.M. Hellerstein, W. Andhong, Approximate data collection in sensor networks using probabilistic models, in: 22nd International Conference on Data Engineering (ICDE06), 2006, pp. 48–52.

[6] M. A. Rassam, A. Zainal, M. A. Maarof, An adaptive and efficient dimension reduction model for multivariate wireless sensor networks applications. Applied Soft Computing 13(4) (2013) 1978-1996.

[7] R. S. Wagner, Distributed Multi-Scale Data Processing for Sensor Networks, Ph.D Thesis, 2007

[8] S. Pattem, B. Krishnamachari, R. Govindan, The impact of spatial correlation on routing with compression in wireless sensor networks, in: 3rd International Symposium on Information Processing in Sensor Networks, 2004, pp. 28-35.

[9] I. Perfilieva, Fuzzy transforms: theory and applications. Fuzzy Sets and Systems 157 (2006) 993-1023.

[10] G. Patane', Fuzzy Transform and least-squares approximation: analogies, differences, and generalizations. Fuzzy Sets and Systems 180(1) (2011) 41–54.

[11] F. Di Martino, V. Loia, S. Sessa, Fuzzy transforms method and attribute dependency in data analysis. Information Sciences 180 (2010) 493-505.

[12] F. Di Martino, V. Loia, I. Perfilieva, S. Sessa, An image coding/decoding method based on direct and inverse fuzzy transforms. International Journal of Approximate Reasoning 48 (2008) 110-131.

[13] F. Di Martino, V. Loia, I. Perfilieva, S. Sessa, Fuzzy transform for coding/decoding images: A short description of methods and techniques. Studies in Fuzziness and Soft Computing 298 (2013) 139–146.

[14] P. Vlasanek, I. Perfilieva, Influence of various types of basic functions on image reconstruction using F-transform, in: 8th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2013) - Advances in Intelligent Systems Research 32 (2013) 497–502.

[15] P. Hurtik, I. Perfilieva, Image compression methodology based on fuzzy transform using block similarity, in 8th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2013 - Advances in Intelligent Systems Research 32 (2013) 521–526.

[16] Sensorscope dataset. http://sensorscope.epfl.ch/index.php/EnvironmentalData

[17] V. Sundaram, P. Eugster, X. Zhang, Prius: Generic Hybrid Trace Compression for Wireless Sensor Networks, in: SenSys12, 2012, pp. 22–26.

[18] J. Ning, J. Wang, W. Gao and C. Liu, Wavelet–Based Data Compression Technique for Smart Grid. IEEE Transactions on Smart Grid 2(1) (2011) 212–218.

[19] Z. Wang, A. Scaglione and R. J. Thomas, Compressing Electrical Power Grids, in: First IEEE International Conference on Smart Grid Communications (SmartGridComm), 2010, pp. 13–18.

[20] R. Wael, I. Anis, M. Morcos Medhat, Novel data compression technique for power waveforms using adaptive fuzzy logic. IEEE Transactions on Power Delivery 20(3) (2005) 2136-43.

[21] I. Perfilieva, Fuzzy transforms, in: J.F. Peters, et al. (Eds.), Transactions on Rough Sets II, Lecture Notes in Computer Science, vol. 3135,2004, pp. 63-81.

[22] I. Perfilieva, M. Dankova, Towards F-transforms of a higher degree, in: IFSA-EUSFLAT Conference, 2009, pp. 585-588.

[23] M.Dankova, M.Stepnicka, Fuzzy transform as an additive normal for., Fuzzy Sets and Systems 157 (2006) 1024-1035.

[24] M. Stepnicka, O. Polakovic, A neural network approach to the fuzzy transform, Fuzzy Sets and Systems 160 (2009) 1037-1047.

[25] B. Bede, I.J. Rudas, Approximation properties of fuzzy transforms, Fuzzy Sets and Systems 180 (2011) 20-40.

[26] I. Perfilieva, V. Novak, V. Pavliska, A. Dvorak, M. Stepnicka, Analysis and prediction of time series using fuzzy transform, in: IEEE World Congress on Computational Intelligence, 2008, pp. 3875-3879.

[27] I. Perfilieva, N. Yarushkina, T. Afanasieva, A. Romanov, Time series analysis using soft computing methods, International Journal of General Systems 42(6) (2013) 687–705.

[28] V. Novk, V. Pavliska, I. Perfilieva, M. Stepnicka, F-transform and Fuzzy natural logic in time series analysis, in: 8th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2013 - Advances in Intelligent Systems Research 32 (2013) 40–47.

[29] M.L. Guerra, L. Stefanini, Expectile smoothing of time series using F-transform, in: 8th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2013 - Advances in Intelligent Systems Research 32 (2013) 559–564.

[30] F. Di Martino, V. Loia, S. Sessa, Fuzzy transforms method and attribute dependency in data analysis, Information Sciences 180 (2010) 493-505.

[31] J.K.I. Tomanova, Hidden functional dependencies found by the technique of F-transform, in: 8th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2013 - Advances in Intelligent Systems Research 32 (2013) 662–668.

[32] M. Stepnicka, R. Valasek, Numerical Solution of Partial Differential Equations with Help of Fuzzy Transform, in: The 2005 IEEE International Conference on Fuzzy Systems, 2005, pp. 1104–1109.