Distributed recognition of human actions using wearable motion sensor networks [1]

Allen Y. Yang [a,*], Roozbeh Jafari [b], S. Shankar Sastry [a] and Ruzena Bajcsy [a]

[a] *Department of EECS, University of California, Berkeley, Berkeley, CA 94720, USA*
E-mail: {yang,sastry,bajcsy}@eecs.berkeley.edu
[b] *Department of EE, University of Texas at Dallas, Richardson, TX 75083, USA, E-mail: rjafari@utdallas.edu*

Abstract. We propose a distributed recognition framework to classify continuous human actions using a low-bandwidth wearable motion sensor network, called distributed sparsity classifier (DSC). The algorithm classifies human actions using a set of training motion sequences as prior examples. It is also capable of rejecting outlying actions that are not in the training categories. The classification is operated in a distributed fashion on individual sensor nodes and a base station computer. We model the distribution of multiple action classes as a mixture subspace model, one subspace for each action class. Given a new test sample, we seek the sparsest linear representation of the sample w.r.t. all training examples. We show that the dominant coefficients in the representation only correspond to the action class of the test sample, and hence its membership is encoded in the sparse representation. Fast linear solvers are provided to compute such representation via ℓ^1-minimization. To validate the accuracy of the framework, a public wearable action recognition database is constructed, called wearable action recognition database (WARD). The database is comprised of 20 human subjects in 13 action categories. Using up to five motion sensors in the WARD database, DSC achieves state-of-the-art performance. We further show that the recognition precision only decreases gracefully using smaller subsets of active sensors. It validates the robustness of the distributed recognition framework on an unreliable wireless network. It also demonstrates the ability of DSC to conserve sensor energy for communication while preserve accurate global classification.

Keywords: Action recognition, wearable sensor network, distributed perception, sparse representation, compressive sensing

1. Introduction

Action/activities recognition has been extensively studied in the past in the literature of computer vision. Compared with either model-based or appearance-based vision systems, the body sensor networks we study in this paper have several distinct advantages: 1. Body sensor systems do not require that the environment be instrumented with cameras or other sensors. 2. Such systems also have the necessary mobility to support persistent monitoring of a subject during her daily activities in both indoor and outdoor environments (e.g., ALARM-NET [31]). 3. With the con-

tinuing miniaturization and integration of mobile processors and wireless sensors, it has become possible to manufacture wearable sensor networks that densely cover the human body to record and analyze very small movements of the human body such as breathing and spine movements with higher accuracy than most extant vision systems (e.g., the Moven motion capture system). Such sensor networks can be used in applications such as medical-care monitoring, athlete training, tele-immersion, and human-computer interaction (e.g., integration of accelerometers in the Wii game controller and smart phones).

In traditional sensor networks, the computation carried by the sensor board is fairly simple: Extract certain local information and transmit the data to a computer server over the network for processing. In this paper, we propose a new method for *distributed pattern recognition*. In this system, each sensor node will

*Corresponding author.

be able to classify local, albeit biased, information. Only when the local classification detects a possible object/event does the sensor node become *active* and transmit the measurement to a network server.[2] On the server side, a global classifier receives the data from the sensor nodes and further optimizes the classification upon local sensor decisions. The global classifier can be more computationally involved than the distributed classifiers, but it has to adapt to the change of available network sensors due to local measurement error, sensor failure, and communication congestion.

1.1. Literature overview

Past studies on sensor-based action recognition were primarily focused on single accelerometers [12,15] or other motion sensors [16,24]. More recent systems prefer using multiple motion sensors [1,2,14,17,19,20, 26]. Depending on the type of sensor used, an action recognition system is typically comprised of two parts: a feature extraction module at the sensor level and a classification module at the server level.

There are three major directions for feature extraction in wearable sensor networks. The first direction uses simple statistics in a motion sequence such as the max, mean, variance, and energy. The second type of feature is computed using fixed filter banks such as *FFT* and *wavelets* [15,24]. The third type is based on classical dimensionality-reduction techniques such as *principal component analysis* (PCA) and *linear discriminant analysis* (LDA) [19,20].

In terms of classification on the action features, a large body of previous work favored thresholding or *k-nearest-neighbor* (kNN) due to the simplicity of the algorithms for mobile devices [15,24,26]. Other more sophisticated techniques have also been used, such as *decision trees* [2,4] and *hidden Markov models* [19].

For distributed pattern recognition, there exist studies on distributed speech recognition [33] and distributed expert systems [23]. One particular problem associated with most distributed sensor systems is that each local observation from a distributed sensor is *biased* and *insufficient* to classify all classes. For example in our system, the sensors placed on the lower-body would not perform well to classify those actions that mainly involve upper body motions, and *vice versa*.

Consequently, traditional majority-voting type classifiers may not achieve the best performance globally.

Due to the unique mobility of wearable sensor networks, such systems have been applied to a variety of applications, especially in the area of human-computer interaction. One dominant application in the past has been single action detection for elderly people, such as falling [7,9,28,30] and walking [3,25]. There have been other systems that tackle more general problems of recognizing multiple different human actions that would be commonplace in people's daily lives [8,19,21,22]. The algorithm proposed in this paper falls in the latter category.

1.2. Design of the wearable sensor network

Our wearable sensor network is based on a novel sensor platform called DexterNet [18]. DexterNet is an open-source body sensor platform that adopts a three-level architecture to control heterogeneous body sensors: At the *body sensor layer*, the platform can integrate and control a variety of experimental and commercial sensor units, including MICAz, SHIMMER, and GPS. At the *personal network layer*, a mobile base station (e.g., a PDA or a smartphone) is utilized to communicate and process the sensor data from a single subject. At the *global network layer*, multiple personal network components communicate with a remote Internet server (e.g., located at a hospital clinic or a research lab) to permanently log the sensor data and support higher-lever applications.

The implementation in this paper consists of five custom-built sensor nodes placed at different body locations (see Fig. 1), which communicate with a base station attached to a computer server through a USB port. The sensor nodes and base station are built using the commercially available Tmote Sky boards. Tmote Sky runs TinyOS on an 8 MHz microcontroller with 10K RAM and communicates using the 802.15.4 wireless protocol. Each custom-built sensor board has a tri-axial accelerometer and a biaxial gyroscope, which is attached to Tmote Sky (shown in Fig. 2). Each axis is reported as a 12 bit value to the node, indicating values in the range of ± 2 g and $\pm 500°/$s for the accelerometer and gyroscope, respectively.

The current hardware design of the sensor contributes certain amounts of measurement error. The accelerometers typically require some calibration in the form of a linear correction, as sensor output under $1g$ may be shifted up to 15% in some sensors. It is also worth noting that the gyroscopes produce an indica-

---

[2]Studies have shown that the power consumption required to successfully send one byte over a wireless channel is equivalent to executing between $1e3$ and $1e6$ instructions on an onboard processor [27]. Hence it is paramount in sensor networks to reduce the communication cost while preserve the recognition performance.

Fig. 1. A subject wearing a body sensor system with the numbering of the sensors superimposed in the image. The sensor system consists of five wireless motion sensors, two on the wrists, one on the waist, and two on the ankles, respectively.
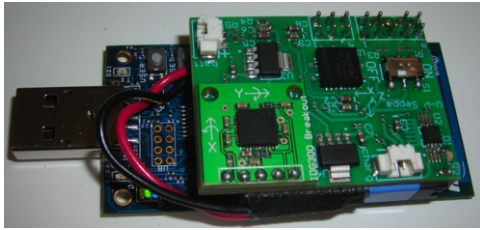

Fig. 2. Illustration of a motion sensor node. The sensor board on the top is a custom-built motion sensor with a triaxial accelerometer and a biaxial gyroscope. The middle layer contains a Li-ion battery. The sensor board on the bottom is a standard Tmote Sky network node.

tion of rotation under straight line motions. Fortunately these systematic errors appear to be consistent across experiments for a given sensor board. However, without calibration to correct them, the errors may affect the action recognition if different sets of sensors are used interchangeably in the experiment.[3]

To avoid packet collision in the wireless channel, we use a *time division multiple access* (TDMA) protocol that allocates each node a specific time slot during which to transmit data. This allows us to receive sensor data at 30Hz with minimal packet loss. To avoid drift in the network, the base station periodically broadcasts

a packet to resynchronize the nodes' individual timers. The code to interface with the sensors and transmit data is implemented directly on the motes using *nesC*, a variant of C.

1.3. Wearable Action Recognition Database

We have constructed a benchmark database for human action recognition using the above wearable motion sensor network, called *Wearable Action Recognition Database* (WARD). The purpose of WARD is to offer a public and relatively stable data set as a platform for quantitative comparison of existing and future algorithms for human action recognition using wearable motion sensors. The database has been carefully constructed under the following conditions:

1. The database contains sufficient numbers of human subjects with a large range of age differences.
2. The designed action classes are general enough to cover most typical actions that a human subject is expected to perform in her daily life.
3. The locations of the wearable sensors are selected to be practical for full-fledged commercial systems.
4. The sampled action data contain sufficient variation, measurement noise, and outliers in order for existing and future algorithms to meaningfully examine and compare their performance.

The WARD database is available for download at: http://www.eecs.berkeley.edu/~yang/ software/WAR/. The data are sampled from 7 female and 13 male human subjects (in total 20 subjects) with age ranging from 19 to 75. The current version, version 1.0, includes the following 13 action categories: 1. *Stand* (ST). 2. *Sit* (SI). 3. *Lie down* (LI). 4. *Walk forward* (WF). 5. *Walk left-circle* (WL). 6. *Walk right-circle* (WR). 7. *Turn left* (TL). 8. *Turn right* (TR). 9. *Go upstairs* (UP). 10. *Go downstairs* (DO). 11. *Jog* (JO). 12. *Jump* (JU). 13. *Push wheelchair* (PU). For more details about the data collection, please refer to the human subject protocol included in the WARD database. The sensor data have been converted and saved in the MATLAB environment. The database also includes a MATLAB program to visualize the action data from the five motion sensors.

1.4. Contribution

We propose a *distributed* action recognition algorithm using up to five wearable motion sensors. The work is inspired by an emerging theory of compressive

---

[3]More sophisticated motion sensors do exist in the industry, which can utilize heterogeneous sensor fusion techniques to self-calibrate the accelerometer and gyroscope. One example is the Microstrain Gyro Enhanced Orientation Sensor at: http://www.microstrain.com/.

sensing [5,6]. We assume each action class satisfies a low-dimensional *subspace* model. If a linear representation is sought to represent a valid test sample w.r.t. all training samples, the dominant coefficients in the *sparsest* representation correspond to the training samples from the same action class, and hence they encode the membership of the test sample.

A distributed recognition system on wireless sensor networks needs to further consider the following issues:

1. How to extract compact and accurate low-dimensional action features for local classification and transmission over a band-limited network?
2. How to classify the local measurement efficiently using low-power processors?
3. How to design a classifier to globally optimize the recognition and adapt to the change of the network?
4. Whether the accuracy of an action recognition system is identity independent? That is, a good classifier should only be sensitive to different action classes, but neutral to the subject who performs the actions.

We tackle these problems by proposing a novel recognition framework consisting of the following three integrated components: 1. Low-dimensional action feature extraction. 2. Fast distributed classifiers via ℓ^1-minimization. 3. An adaptive global classifier on the base computer. The method can accurately classify human actions from a continuous motion sequence. The local classifiers that reject potential outliers can reduce the sensor-to-server communication to about 50%. One can also choose to activate only a subset of the sensors on the fly due to sensor failure or network congestion. The global classifier is able to adaptively update the optimization process and improve the global classification upon available local decisions. Finally, in the experiment, we examine the identity-independence performance on a test sequence by excluding the training samples of the same subject.

Note that a similar algorithm was previously published in a manuscript [29]. In comparison, [29] mainly discusses simultaneous segmentation and classification of transient actions, such as from standing to sitting, from sitting to lying down, and bending. In this paper, we discuss classification of continuous actions. The preliminary results shown in [29] only contain recognition results from three human subjects with age ranging from 28 to 32. In this paper, the system utilizes the much larger WARD benchmark to validate its performance.

The rest of the paper is organized as follows. Section 2 proposes a unified classification algorithm via a novel sparse representation framework on individual motion sensors to classify human actions with local bias. Section 3 further proposes a global classification algorithm on a base computer that receives action features from active sensors in the network and adaptively boost the recognition upon individual sensor decisions. Finally, we demonstrate the performance of the overall algorithm based on the WARD benchmark in Section 4.

2. Classification via sparse representation

We first define the problem of *distributed action recognition*.

Problem 1 (Distributed Action Recognition). *Assume a set of L wearable sensor nodes with triaxial accelerometers (x, y, z) and biaxial gyroscopes (θ, ρ) are attached to the human body. Denote*

$$\boldsymbol{a}_j(t) \doteq (x_j(t), y_j(t), z_j(t), \theta_j(t), \rho_j(t))^T \in \mathbb{R}^5 \quad (1)$$

as the measurement of the five readings on node j at time t, and

$$\boldsymbol{a}(t) \doteq (\boldsymbol{a}_1^T(t), \boldsymbol{a}_2^T(t), \dots, \boldsymbol{a}_L^T(t))^T \in \mathbb{R}^{5L} \quad (2)$$

collects all L sensors at time t. Further denote

$$\boldsymbol{s} = (\boldsymbol{a}(1), \boldsymbol{a}(2), \dots, \boldsymbol{a}(l)) \in \mathbb{R}^{5L \times l} \quad (3)$$

as an action segment of length l in time.

Given K different classes of human actions, a set of n_i training examples $\{\boldsymbol{s}_{i,1}, \dots, \boldsymbol{s}_{i,n_i}\}$ are collected for each ith class, all of which have the same duration l. Given a new test sequence $\boldsymbol{s}$, we seek a distributed algorithm to classify the action into one of the K categories, or reject the action as an invalid measurement. Finally, given continuous measurements of different human actions, determine an optimal duration parameter l to extract training samples and test samples $\boldsymbol{s}$.[4]

In this section, our focus should be an action classification method on each sensor node assuming a fixed

---

[4]The continuous actions such as sitting and walking can last from a few seconds to minutes or hours. The single duration parameter l is defined w.r.t. the recognition algorithm such that different actions can be accurately classified using the shortest duration window. The reader is referred to [29] for the segmentation problem of transient actions with different action durations.

duration l. Given $\boldsymbol{s}_j = (\boldsymbol{a}_j(1), \boldsymbol{a}_j(2), \ldots, \boldsymbol{a}_j(l)) \in \mathbb{R}^{5 \times l}$ on node j, define a new vector $\boldsymbol{s}_j^S$ as the *stacking* of the l columns of $\boldsymbol{s}_j$:

$$\boldsymbol{s}_j^S \doteq (\boldsymbol{a}_j(1)^T, \boldsymbol{a}_j(2)^T, \ldots, \boldsymbol{a}_j(l)^T)^T \in \mathbb{R}^{5l}. \quad (4)$$

We will interchangeably use $\boldsymbol{s}_j$ to denote the stacked vector $\boldsymbol{s}_j^S$ without causing ambiguity.

Subsequently, we define a *full-body action vector* $\boldsymbol{v}$ that stacks the measurement from all L nodes:

$$\boldsymbol{v} \doteq (\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, \ldots, \boldsymbol{s}_L^T)^T \in \mathbb{R}^D, \quad (5)$$

where $D = D_1 + \cdots + D_L = 5lL$.

In this paper, we assume the samples $\boldsymbol{v}$ in an action class satisfy a *subspace* model, called an *action subspace*. If the training samples $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n_i}\}$ of the ith class sufficiently span the ith action subspace, given a test sample $\boldsymbol{y} = (\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_L^T)^T \in \mathbb{R}^D$ in the same class i, $\boldsymbol{y}$ can be linearly represented using the training examples of the same class:

$$\boldsymbol{y} = \alpha_1 \boldsymbol{v}_1 + \cdots + \alpha_{n_i} \boldsymbol{v}_{n_i}$$
$$\Leftrightarrow \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_L \end{pmatrix} = \left(\begin{pmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \vdots \\ \boldsymbol{s}_L \end{pmatrix}_1 \cdots \begin{pmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \vdots \\ \boldsymbol{s}_L \end{pmatrix}_{n_i} \right) \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_i} \end{pmatrix}. \quad (6)$$

It is important to note that such linear constraint also holds for each node j in (6):

$$\boldsymbol{y}_j = \alpha_1 \boldsymbol{s}_{j,1} + \cdots + \alpha_{n_i} \boldsymbol{s}_{j,n_i} \in \mathbb{R}^{D_j}. \quad (7)$$

In theory, complex data such as human actions typically constitute more complex nonlinear models. The linear models are used to *approximate* such nonlinear structures in a higher-dimensional subspace (see Fig. 3). Notice that such linear approximation may not produce good estimation of the distance/similarity metric for the samples on the manifold. However, as we will show in Example 1, given sufficient samples on the manifold as training examples, a new test sample can be accurately *represented* on the subspace, provided that any two classes do not have similar subspace models.
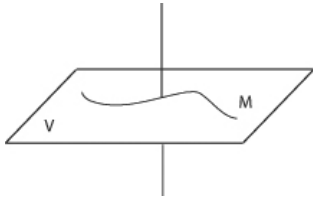

Fig. 3. Modeling a 1-D manifold M using a 2-D subspace V.

To recover label($\boldsymbol{y}$), a previous study [32] proposes to reformulate the recognition using a sparse representation: Since label($\boldsymbol{y}$) $= i$ is unknown, we can represent $\boldsymbol{y}$ using all the training samples from all K classes:

$$\boldsymbol{y} = (A_1, A_2, \ldots, A_K) \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_K \end{pmatrix} = A\boldsymbol{x}, \quad (8)$$

where

$$A_i = (\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \ldots, \boldsymbol{v}_{i,n_i}) \in \mathbb{R}^{D \times n_i} \quad (9)$$

collects all the training samples of class i,

$$\boldsymbol{x}_i = (\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n_i})^T \in \mathbb{R}^{n_i} \quad (10)$$

collects the corresponding coefficients in (6), and $A \in \mathbb{R}^{D \times n}$ where $n = n_1 + n_2 + \cdots + n_K$. Since $\boldsymbol{y}$ satisfies both (6) and (8), one solution of $\boldsymbol{x}$ in (8) should be

$$\boldsymbol{x}^* = (0, \ldots, 0, \boldsymbol{x}_i^T, 0, \ldots, 0)^T. \quad (11)$$

The solution is naturally *sparse*: in average only a fraction of $\frac{1}{K}$ terms in $\boldsymbol{x}^*$ are nonzero.

It is important to note that, on each sensor j in this section, solution $\boldsymbol{x}^*$ of (8) is also a solution for the representation:

$$\boldsymbol{y}_j = (A_1^{(j)}, A_2^{(j)}, \ldots, A_K^{(j)}) \boldsymbol{x} = A^{(j)} \boldsymbol{x}, \quad (12)$$

where $A_i^{(j)} \in \mathbb{R}^{D_j \times n_i}$ consists of row vectors in A_i that correspond to the jth node. Hence, $\boldsymbol{x}^*$ can be solved either globally using (8) or locally using (12), provided that the action data measured on each node are *sufficiently discriminant*. We will come back to the discussion about local classification versus global classification in Section 3. In the rest of this section however, our focus will be on each node.

One major difficulty in solving (12) is the high dimensionality of the action data. In *compressive sensing* [5,6], one reduces the dimension of a linear system by choosing a linear projection $R_j \in \mathbb{R}^{d \times D_j}$:[5]

$$\tilde{\boldsymbol{y}}_j \doteq R_j \boldsymbol{y}_j = R_j A^{(j)} \boldsymbol{x} \doteq \tilde{A}^{(j)} \boldsymbol{x} \in \mathbb{R}^d. \quad (13)$$

After projection R_j, typically the feature dimension d is much smaller than the number n of all training samples. Therefore, the new linear system (13) is un-

---

[5]Notice that R_j is not computed on the sensor node. These matrices are computed offline and simply stored on each sensor node.

derdetermined. Numerically stable solutions exist to *uniquely* recover sparse solutions $\boldsymbol{x}^*$ via ℓ^1-minimization [10]:

$$\boldsymbol{x}^* = \arg\min \|\boldsymbol{x}\|_1 \text{ subject to } \tilde{\boldsymbol{y}}_j = \tilde{A}^{(j)}\boldsymbol{x}. \quad (14)$$

These routines include (orthogonal) matching pursuit (MP), basis pursuit (BP), the LASSO.[6]

In our experiment, we have tested multiple projection operators including PCA, LDA, locality preserving projection (LPP) [13], and random project studied in [32]. We found that 40-D feature spaces using LPP produces the best recognition in a very low-dimensional space. Throughout this paper, we will use 40-D LPP features to represent local motions measured on sensor nodes.[7]

After the (sparsest) representation $\boldsymbol{x}$ is recovered, we project the coefficients onto each action subspaces

$$\delta_i(\boldsymbol{x}) = (0,\ldots,0,\boldsymbol{x}_i^T,0,\ldots,0)^T \in \mathbb{R}^n, i = 1,\ldots,K. \quad (15)$$

Subsequently, the membership of the test sample $\boldsymbol{y}_j$ is assigned to the class with the smallest residual

$$\text{label}(\boldsymbol{y}_j) = \arg\min_i \|\tilde{\boldsymbol{y}}_j - \tilde{A}^{(j)}\delta_i(\boldsymbol{x})\|_2. \quad (16)$$

The overall algorithm deployed on each sensor node is summarized in Algorithm 1, which is called *local sparsity classifier* (LSC).

Algorithm 1: Local Sparsity Classifier (LSC).

Input: A set of training samples $A^{(j)} = (\,\boldsymbol{s}_{j,1} \cdots \boldsymbol{s}_{j,n}\,)$, a test sample $\boldsymbol{y}_j$ on a sensor node j, and a linear projection matrix R_j.

  1: Projection: $\tilde{\boldsymbol{y}}_j = R_j\boldsymbol{y}_j$.
  2: $\boldsymbol{x}^* = \arg\min \|\boldsymbol{x}\|_1$ subject to $\tilde{\boldsymbol{y}}_j = R_j A^{(j)}\boldsymbol{x}$.
  3: $\text{label}(\boldsymbol{y}_j) = \arg\min_{i=1,\ldots,K}\|\tilde{\boldsymbol{y}}_j - R_j A^{(j)}\delta_i(\boldsymbol{x})\|_2$.

Output: label($\boldsymbol{y}_j$), action feature $\tilde{\boldsymbol{y}}_j$, and $\boldsymbol{x}^*$.

---

[6]The implementation of these routines is available in a MATLAB toolbox called *SparseLab*: http://sparselab.stanford.edu.

[7]The choice of an "optimal" low-dimensional feature space is not the emphasis of this paper. On one hand, a practitioner may easily replace LPP with other feature spaces without modification of the algorithm. On the other hand, a previous result in [32] has shown that the accuracy of sparse representation via ℓ^1-minimization converges among different linear projections, as long as the dimension of the feature space is sufficiently high. The result renders the choice of a particular feature space not very significant in solving for a sparse representation.

Example 1 (Classification on Nodes)**.** We demonstrate the recognition accuracy of LSC on individual nodes based on the WARD database. First, we look for fast sparse solvers in the literature. We found that BP [11] gives the best trade-off between speed, noise tolerance, and recognition accuracy.

We design the training set and the test set as follows. For each motion sequence in the WARD database, we randomly sample 10 segments of length l in the training set. In total, there are $20 \times 13 \times 5 \times 10 = 13000$ training examples. During the testing, LSC attempts to classify all continuous segments of length l in the WARD database. With respect to each subject, the corresponding training examples will be excluded from the training set before classification. Therefore, any test subject is not present in the training set, and the recognition is *subject independent*. In the experiment, we found that $l = 45$ is a short action duration that yields satisfactory performance, which corresponds to 1.5 seconds given the 30 Hz sampling rate.

Figure 4 illustrates an example of sparse representation $\boldsymbol{x}$ and its corresponding residuals estimated on the first node (left wrist) of a jumping sequence (Action 12).

Table 1 shows the recognition accuracy of LSC on all the 13 categories. There should be no surprise that LSC alone based on the single node measurement of human actions does not produce good classification, as many human actions engage movements at multiple body parts. For example, nodes at the two ankle positions cannot differentiate walking forward and pushing wheelchair because the feet engage similar movements in both categories. In Table 1, we also show the per-
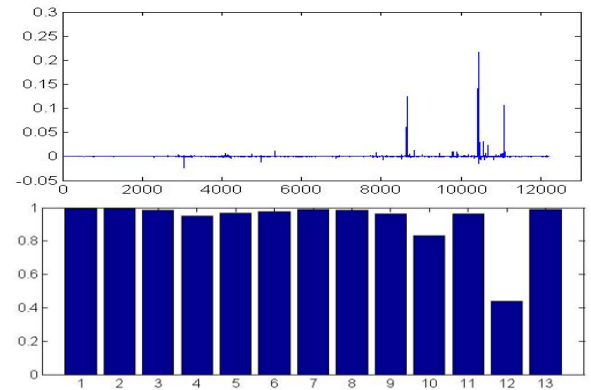

Fig. 4. Top: Sparse ℓ^1 solution by BP of a jumping motion on the left wrist node. Bottom: Reconstruction residuals with respect to the 13 action categories. The test sample is correctly classified as Class 12. $\text{SCI}(\boldsymbol{x}) = 0.335$ (see (17)).

Table 1

Recognition accuracy via LSC on WARD. The last column (1–5) shows the recognition accuracy using majority voting

Sen #	1	2	3	4	5	1–5
Acc [%]	65.08	61.26	63.9	78.56	77	90.2

Table 2

Recognition accuracy via nearest neighbor on WARD. The last column (1–5) shows the recognition accuracy using majority voting

Sen #	1	2	3	4	5	1–5
Acc [%]	64.9	59.3	67.4	80.3	76.0	90.5

formance of a simple global classifier: majority voting. If all the local decisions are collected and a majority vote is chosen as the overall classification of the test action, LSC achieves 90.2% accuracy. This will become a baseline benchmark to compare with an adaptive classifier we will introduce in the next section.

Nearest neighbor (NN) is one of the popular methods used in sensor networks for classification. Table 2 shows the recognition accuracy of NN on the WARD database. We compare Table 1 and Table 2. Because the inherent correlation between the distributed motion sensors are not considered beyond the majority-voting process, the two algorithms generate very similar global recognition accuracy. Using majority voting, nearest neighbor achieves 90.5%.

3. Adaptive global recognition

In this section, we introduce an adaptive framework to optimize a global classification based on all the available distributed sensor data. First, we discuss an outlier rejection criterion to identify invalid motion samples measured on the individual sensor nodes. The invalid samples would not be sent to the global classifier that we will introduce later. The ability to locally reject invalid measurement reduces the power consumption on the sensor nodes to communicate with the network station, as we will show in Section 4.

Based on the previous sparsity assumption, if $\boldsymbol{y}_j$ is not a valid segment on node j w.r.t. the training examples $A^{(j)}$, the dominant coefficients of its sparsest representation $\boldsymbol{x}$ should not correspond to any single class. We utilize a *sparsity concentration index* (SCI) [32]:

$$\text{SCI}(\boldsymbol{x}) \doteq \frac{K \cdot \max_{j=1,\dots,K} \|\delta_j(\boldsymbol{x})\|_1 / \|\boldsymbol{x}\|_1 - 1}{K - 1} \in [0, 1].$$

(17)

If the nonzero coefficients of $\boldsymbol{x}$ are evenly distributed among K classes, then $\text{SCI}(\boldsymbol{x}) = 0$; if all the nonzero coefficients are associated with a single class, then $\text{SCI}(\boldsymbol{x}) = 1$. Therefore, we introduce a sparsity threshold τ_1 applied on individual sensor nodes: If $SCI(\boldsymbol{x}) > \tau_1$, the motion sample is a valid local measurement, and its 40-D LPP features $\tilde{\boldsymbol{y}}$ will be sent to the base station; otherwise, the sample will be ignored.

It is important to note that a local measurement that is labeled as a valid sample w.r.t. τ_1 may not truly correspond to a valid human action when multiple sensor data are jointly considered based on the training actions defined in the WARD database. For example, WF, UP, and DO all involve similar upper body movements; on the other hand, if a subject only tries to mimic a WF motion by moving the upper body but not the lower body, the movement becomes an invalid action when both the upper body data and the lower body data are jointly considered. Therefore, a global constraint is needed to reject such invalid samples, which will be discussed next.

Suppose at time t, the base station receives L' action features from the active sensors ($L' \leq L$). Without loss of generality, assume these features are from the first L' sensors: $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \dots, \tilde{\boldsymbol{y}}_{L'}$.

Denote

$$\tilde{\boldsymbol{y}}' = (\tilde{\boldsymbol{y}}_1^T, \dots, \tilde{\boldsymbol{y}}_{L'}^T)^T \in \mathbb{R}^{dL'}.$$

(18)

Then the global sparse representation $\boldsymbol{x}$ of $\tilde{\boldsymbol{y}}'$ satisfies the following linear system

$$\tilde{\boldsymbol{y}}' = \begin{pmatrix} R_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & R_{L'} & \cdots & 0 \end{pmatrix} A\boldsymbol{x} = R'A\boldsymbol{x} = \tilde{A}'\boldsymbol{x},$$

(19)

where $R' \in \mathbb{R}^{dL' \times D}$ is a new projection matrix that only extracts the action features from the first L' nodes. Consequently, the effect of changing active sensor nodes for the global classification is formulated via the global projection matrix R'. During the transformation, the data matrix A and the sparse representation $\boldsymbol{x}$ remain *unchanged*. The linear system (13) then becomes a special case of (19) where $L' = 1$.

Similar to the outlier rejection criterion on each node, we introduce a global rejection threshold τ_2. If $\text{SCI}(\boldsymbol{x}) > \tau_2$ in (19), the most significant coefficients in $\boldsymbol{x}$ are concentrated in a single training class. Hence $\tilde{y}'$ is assigned to that class. Otherwise, the sample will be rejected as an outlier. The overall algorithm on the network station is summarized in Algorithm 2, which

is called *distributed sparsity classifier* (DSC). DSC provides a unified solution to detect and classify action segments in a network of body sensors using only two simple parameters τ_1 and τ_2.

Algorithm 2: Distributed Sparsity Classifier (DSC).

Input: A set of stacked training samples $A = \{v_1, \ldots, v_n\}$ from sensors $1, \ldots, L$, test sample y of action features measured from L active sensors, and sparsity parameters τ_1, τ_2.

  1: **for all** each sensor $1 \le j \le L$ **do**
  2:    Solve for sparse representation x^* using Algorithm 1 with parameters $A^{(j)}$ and y_j.
  3:    If $\text{SCI}(x^*) > \tau_1$, send feature vector $\tilde{y}_j$ to the network station.
  4: **end for**
  5: Collect all valid features $\tilde{y}'$, construct corresponding projection matrix R'.
  6: Solve $x^* = \arg\min \|x\|_1$ subject to $\tilde{y}' = R'Ax$.
  7: **if** $\text{SCI}(x^*) > \tau_2$ **then**
  8:    $\text{label}(y) = \arg\min_{i=1,\ldots,K} \|\tilde{y}' - R'A\delta_i(x)\|_2$.
  9: **else**
10:    $\text{label}(y) = -1$ (outlier).
11: **end if**

Output: label(y).

Example 2 (Distributed Sparsity Classifier). Consider Action 13 in the WARD database, i.e., PU (pushing a wheelchair). While the upper body motion of this action is quite distinct, the lower body motion often resembles several other actions in the database, such as WF and UP. Figure 5 illustrates the ℓ^1 solutions on the five individual sensor nodes.

First, we observe that the local sparsity classifier (LSC) returns five different labels w.r.t. to the local measurement on the five sensors. It shows that majority-voting type solutions mostly should fail to correctly classify this motion. Second, using a threshold τ_1 against the SCI values of the representations, we can reject certain number of the local motions as invalid measurements.

Assume $\tau_1 = 0.1$ is selected for all five sensors, then measurements from Sensors 1 and 2 will be rejected and DSC solves for a sparse representation using the three 40-D action features from Sensors 3, 4, and 5. Figure 6 shows the global ℓ^1 solution of (19), and the full-body motion is correctly classified as from Action 13.

Notice that at the node level, none of Sensors 3–5 correctly classifies the action based on the available lo-

cal observations, because they are also similar to other actions such as UP, TR, and WR. However, when the measurements from multiple sensors are combined in (19) to represent the full-body motion, the incorrect local decisions are rectified. Such ability is the main reason that the proposed DSC framework can outperform other majority-voting type algorithms. We will examine the performance of DSC in more detail in Section 4.

The DSC method compares favorably to other classical methods such as NN and decision trees, because these methods need to train multiple thresholds and outlier rejection rules when the number L' and the set of available sensors vary in the full-body action vector $\tilde{y}' = (\tilde{y}_1^T, \ldots, \tilde{y}_{L'}^T)^T$. Particularly, a global nearest-neighbor (GNN) algorithm can be modeled as a special case of sparse representation. Suppose in (19) there are L' active sensors and denote $A' = (v_1', v_2', \ldots, v_n')$. Then GNN solves for the following sparse representation of y':

$$x^* = (0, \ldots, 0, 1_i, 0, \ldots, 0)^T$$
$$\text{subject to } i = \arg\min_j \|y' - v_j'\|_2. \tag{20}$$

The optimal solution x^* for GNN is clearly sparse with only one nonzero coefficient corresponding to the closest neighbor of y' in the training set A'. The formulation also generates to k-nearest-neighbors (kNN) and other similar variations.

Finally, we consider how the change of active nodes affects ℓ^1-minimization and the classification of the actions. In compressive sensing, the efficacy of ℓ^1-minimization in solving for the sparsest solution x in (19) is characterized by the ℓ^0/ℓ^1 equivalence relation [10,11]. A necessary and sufficient condition for the equivalence to hold is the *k-neighborliness* of $\tilde{A}'$. As a special case, one can show that if x is the sparsest solution in (19) for $L' = L$, x is also a solution for $L' < L$. Hence, the decrease of L' leads to possible sparser solutions of x.

On the other hand, the decrease in available action features also makes $\tilde{y}'$ less discriminant. For example, if we reduce $L' = 1$ and only activate a wrist sensor, then the ℓ^1-solution x may have nonzero coefficients associated to multiple actions with similar wrist motions, albeit sparser. This is an inherent problem for any method to classify human actions using a limited number of motion sensors. In theory, if two action subspaces in a low-dimensional feature space have a small subspace distance after the projection, the corre-

(a) Sparse representation of the left wrist motion. Local classification label is 13 (PU).

(b) Sparse representation of the right wrist motion. Local classification label is 4 (WF).

(c) Sparse representation of the waist motion. Local classification label is 9 (UP).

(d) Sparse representation of the left ankle motion. Local classification label is 8 (TR).

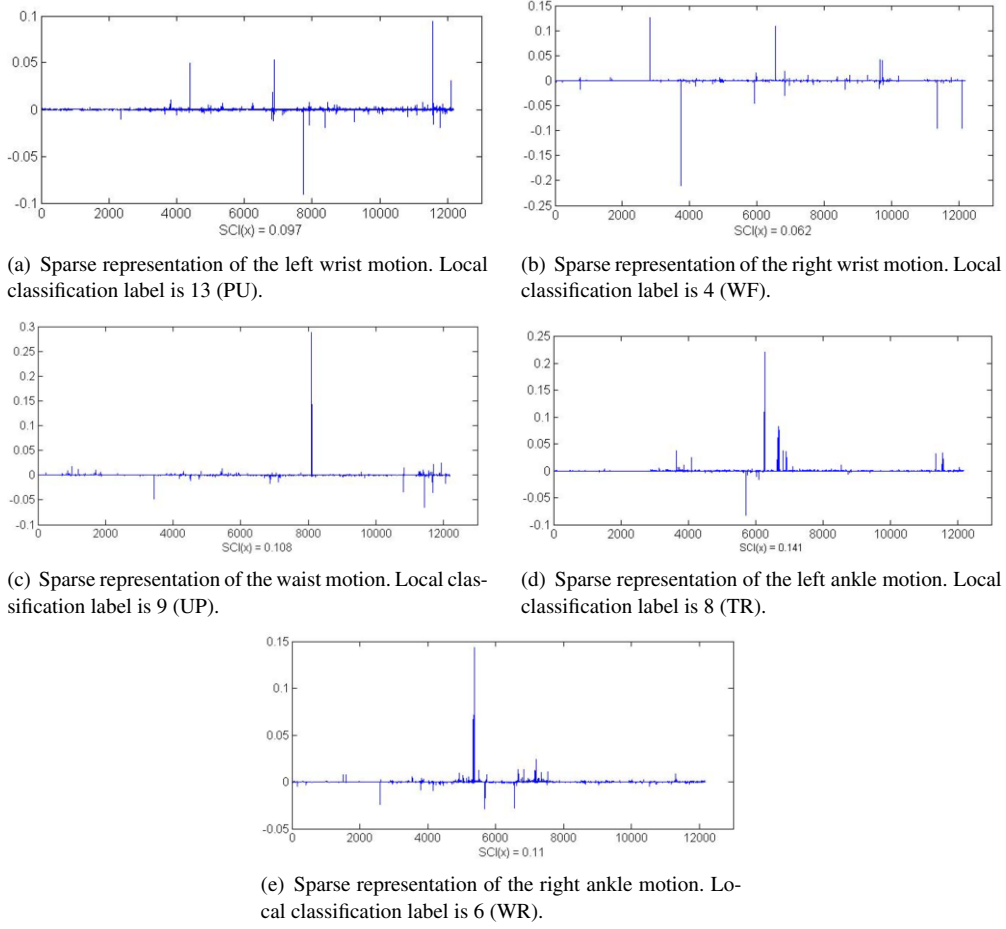(e) Sparse representation of the right ankle motion. Local classification label is 6 (WR).

Fig. 5. Illustration of a PU motion (action 13) classified on individual sensor nodes. Each LSC estimates a different action category that correlates to the true action. Compared to Fig. 4, these solutions have much lower SCI values.
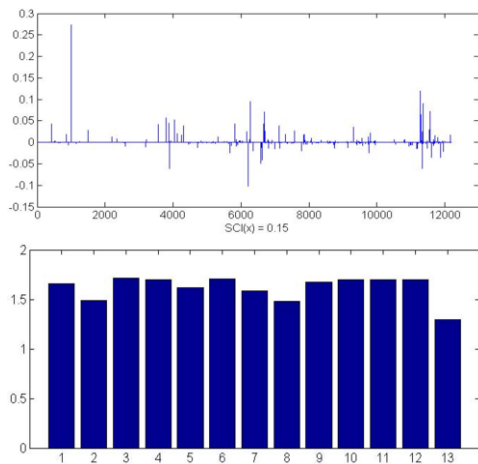

Fig. 6. Top: DSC sparse representation of a sample from action 13 in Fig. 5. Assume $\tau_1 = 0.1$ and $\tau_2 = 0.08$, and Sensors 1 and 2 are rejected. Bottom: Reconstruction residuals with respect to the 13 action categories. The test sample is correctly classified as Class 13.

sponding sparse representation cannot distinguish the test samples from the two classes. We will demonstrate in Section 4 that indeed reducing the available motion sensors will reduce the discriminant power of the sparse representation in a lower-dimensional space.

4. Experiment

In this section, we conduct extensive experiments to examine the performance of the DSC framework using the WARD database. Two different scenarios are considered. First, we calculate the classification accuracy with different subsets of motion sensors available in the network. This experiment is intended to verify that DSC is adaptive to the change of network configuration on-the-fly due to real-world conditions such as sensor failure, battery failure, and network congestion. Second, we consider the effect of the local outlier rejection threshold τ_1 to the accuracy of the global clas-

sification: Higher rejection thresholds save power consumption in communication at the expense of less local information available to the global classifier, and *vice versa*. It is important to note that to measure the performance under the identity-independence assumption, all training examples of a test subject should be excluded from the training set during the experiment. For each motion sequence in the WARD database, we randomly sample 10 segments of length $l = 45$ as training examples.

4.1. Classification with different network configurations

We first test the performance of DSC by manually eliminating certain number of available sensors in the network. Based on the total number L' of LPP feature vectors received, DSC is able to update the classification criterion (19) on-the-fly and adapts to the potentially adversary condition. Table 3 shows the performance of the algorithm, which is quantified by false positive rate (FPR), verification rate (VR), and active sensor rate (ASR).[8] For all the trials, the outlier rejection thresholds τ_1 and τ_2 are set to be 0.08, respectively. The duration l of the test action length is set to be 45, which corresponds to 1.5 seconds given the 30 Hz sampling rate. When all continuous action segments of length 45 are classified in the experiment, the total number of test samples amounts to 500828.

We compare the performance of DSC to the conventional solution of GNN (20). Since the WARD does not purposely contain outliers, we did not use any outlier rejection rule in searching for nearest neighbors, which could be difficult to tune when the available action features change on-the-fly. Table 4 shows the performance of the algorithm. Compared with Table 2, we observe that there is no improvement w.r.t. classification using all five sensors. In fact, the accuracy in Table 4 is lower than the accuracy of 90.5% in Table 2 using majority-voting. This result demonstrates the dependency of NN-type algorithms toward the (dense) distribution of training examples in a high-dimensional data space. Compared with Table 3, GNN also underperforms DSC. For example, DSC outperforms GNN

Table 3

Performance of DSC measured by false positive rate (FPR), verification rate (VR), and active sensor rate (ASR)

Sen #	1–5	1,3,4	1,4	1,3	3,4
FPR [%]	7.14	8	11.49	17.97	14.63
VR [%]	94.59	96.84	98.19	95.57	97.28
ASR [%]	91.85	54.82	37.66	35.58	36.76

Table 4

Performance of GNN measured by false positive rate (FPR), verification rate (VR), and active sensor rate (ASR)

Sen #	1–5	1,3,4	1,4	1,3	3,4
FPR [%]	10.64	14.54	13.93	26.88	18.27
ASR [%]	100	60	40	40	40

by about 6% using Sensors $(1, 3, 4)$, and about 9% using Sensors $(1, 3)$.

We further analyze the classification between different action categories. Table 5 shows a confusion table of the DSC results using all the five sensors in accuracy percentage. The confusion table clearly indicates several action categories mostly contribute to the false positive rate.

1. We observe that three action categories, i.e., ST, SI, and LI, have the highest misclassification. Particularly, it is difficult to differentiate between standing and sitting in the WARD database using both DSC and NN (whose confusion table is not shown in this paper). We argue that the problem is mainly contributed by the choice of the locations of the two lower-body sensors at the ankle locations, because human subjects do not have to move the ankles to perform both standing and sitting actions, and inherently the change of the orientation of the waist sensor is also small between standing and sitting. To improve the classification of the three action categories, one solution could be to introduce new sensor locations around the knees and the thighs.[9]

2. Between the actions WF, WL, and WR, the algorithm in fact performs better than we have expected, because the difference of the three actions is small. For example, 2.5% of the WF action is misclassified as WL, 1.6% misclassified as WR, and furthermore 2.3% misclassified as PU. These are the actions that are all similar in nature.

---

[8]FPR is the percentage of samples that are either true outliers falsely classified as inliers or true inliers assigned to the wrong classes. VR is the percentage of samples that are correctly classified as inliers. Note that the WARD database does not purposely contain outlying actions, hence FPR is equal to one minus the accuracy percentage.

[9]In a previous study [29], we have also suggested that the sensors placed at the ankle locations tend to provide less action information than the other conventional locations such as the knees and the waist.

Table 5

Confusion table of the 13 action classes for DSC using sensors 1–5 (in percentage)

	1	2	3	4	5	6	7	8	9	10	11	12	13
1 (ST)	87.2	10.2	0.7	0	0	0	0.1	1.8	0	0	0	0	0
2 (SI)	25.2	66.8	6.8	0	0	0	0.1	0.1	0	0.1	0	0.1	0.7
3 (LI)	2.6	5.1	91.8	0	0	0	0	0	0	0	0	0.1	0.3
4 (WF)	0	0	0	92	2.5	1.6	0.2	0.2	0.4	0.7	0	0.2	2.3
5 (WL)	0.1	0	0	0.2	97.3	0	0.6	0.3	0.3	0.1	0.1	0.2	1
6 (WR)	0	0	0	0.1	0.1	95.7	0.2	0.4	0.4	0.4	0.5	0.2	2
7 (TL)	0	0	0	0	0.6	0	97	2.3	0	0	0	0	0.1
8 (TR)	0	0	0	0	0	1.6	3.1	95.2	0	0	0	0	0
9 (UP)	0	0	0	0	0	0	0	0	98	0.1	1.6	0.1	0.2
10 (DO)	0	0	0	0.2	0.1	0	0	0	0.1	98.3	0	0.5	0.8
11 (JO)	0	0	0	0	0	0	0	0	0.5	0	99.3	0.1	0.1
12 (JU)	0.1	0	0	0	0	0	0	0	0.3	0.6	0.5	97.9	0.5
13 (PU)	0.3	0.1	0	0.1	0.2	0.1	0.1	0.1	0	0.2	0.2	0.1	98.6

Table 6

Recognition accuracy of DSC with different local rejection thresholds

τ_1	0.08	0.12	0.18
ASR [%]	91.85	72.19	45.58
FPR [%]	7.14	7.58	7.96
VR [%]	94.59	91.03	88.33

3. Despite the similarity of local motions between PU and several other motions, the recognition of PU is quite accurate. The last row of Table 5 shows that about 0.1% to 0.3% test samples are misclassified as 10 of the other 12 categories. Nevertheless, the true positive rate of PU is above 98%.

4.2. Classification with different rejection thresholds

In this experiment, we test the effect of different local rejection threshold τ_1 on the global classification. During the experiment, the global rejection threshold τ_2 is fixed at 0.08. Table 6 shows the performance of the DSC algorithm. First, naturally ASR decreases as τ_1 increases. Particularly, compared to $ASR = 91.85\%$ when $\tau_1 = 0.08$, the rate is reduced to 45.58% when $\tau_1 = 0.18$, which means in average less than half of the sensors transmit action features during the experiment. With more than half of the sensors inactive in the network to conserve power consumption, the experiment shows that DSC still achieves below 8% FPR globally, and VR is above 88%. The result corroborates the design principle of the DSC algorithm that the distributed classification framework via sparse representation is capable of effectively reducing the power

consumption on communication yet at the same time perserving highly accurate recognition accuracy.

5. Conclusion and discussion

Inspired by the emerging compressive sensing theory, we have proposed a distributed algorithm, i.e., distributed sparsity classifier (DSC), to classify human actions on a wearable motion sensor network. The framework provides a unified solution based on ℓ^1-minimization to classify valid action segments and reject outlying actions on the sensor nodes and the base station. We have shown through our experiment that a set of 13 action classes can be accurately represented and classified using a set of 40-D LPP features measured at multiple body locations. The proposed global classifier can adaptively adjust the global optimization to boost the recognition upon available local measurements. To corroborate the validity of the algorithm, and to safeguard the reproducibility of the system performance, we have published an open benchmark database called WARD with this paper. The high recognition accuracy on the WARD database indicates that DSC should be able to classify other action categories such as falling, bicycling, and hand motions with similar high accuracy.

One important observation w.r.t. to the choice of sensor locations on the human body is that the motion measurements from the ankle locations may not discriminate certain categories of upper-body motions and even lower-body motions. We have suggested to replace the ankle locations with other locations around the knees and thighs in order to improve the classi-

fication. Another limitation in the current system and most other body sensor systems is that the wearable sensors need to be firmly positioned at the designated locations. However, a more practical system/algorithm should tolerate certain degrees of shift without sacrificing the accuracy. In this case, the variation of the measurement for different action classes would increase substantially. One open question is what low-dimensional linear/nonlinear models one may use to model such more complex data, and whether the sparse representation framework can still apply to approximate such structures with limited numbers of training examples. A potential solution to this question will be a meaningful step forward both in theory and in practice.

Acknowledgments

References

[1] R. Aylward and J. Paradiso, "A compact, high-speed, wearable sensor network for biomotion capture and interactive media," in *Proceedings of the International Conference on Information Processing in Sensor Networks*, pp. 380–389, 2007.

[2] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Proceedings of the International Conference on Pervasive Computing*, pp. 1–17, 2004.

[3] P. Barralon, N. Vuillerme and N. Noury, "Walk detection with a kinematic sensor: Frequency and wavelet comparison," in *Proceedings of the 28th IEEE EMBS Annual International Conference*, pp. 1711–1714, 2006.

[4] A. Benbasat and J. Paradiso, "Groggy wakeup – automated generation of power-efficient detection hierarchies for wearable sensors," in *Proceedings of International Workshop on Wearable and Implantable Body Sensor Networks*, 2007.

[5] E. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, pp. 1–20, 2006.

[6] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Information Theory*, vol. 52, No. 12, pp. 5406–5425, 2006.

[7] J. Chen, K. Kwong, D. Chang, J. Luk and R. Bajcsy, "Wearable sensors for reliable fall detection," in *Proceedings of the IEEE Engineering in Medicine and Biology Conference*, pp. 3551–3554, 2005.

[8] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Borriello, B. Hemingway, P. Klasnja, K. Koscher, J. Landay, J. Lester and D. Wyatt, "The mobile sensing platform: An embedded activity recognition system," *Pervasive Computing*, pp. 32–41, 2008.

[9] T. Degen, H. Jaeckel, M. Rufer and S. Wyss, *SPEEDY: A fall detector in a wrist watch*, Proceedings of the IEEE International Symposium on Wearable Computers, pp. 184–187, 2003.

[10] D. Donoho, "Neighborly polytopes and sparse solution of underdetermined linear equations," (preprint) 2005.

[11] D. Donoho and M. Elad, "On the stability of the basis pursuit in the presence of noise," *Signal Processing*, vol. 86, pp. 511–532, 2006.

[12] J. Farringdon, A. Moore, N. Tilbury, J. Church and P. Biemond, "Wearable sensor badge & sensor jacket for context awareness," in *Proceedings of the International Symposium on Wearable Computers*, pp. 107–113, 1999.

[13] X. He, S. Yan, Y. Hu, P. Niyogi and H. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[14] E. Heinz, K. Kunze and S. Sulistyo, "Experimental evaluation of variations in primary features used for accelerometric context recognition," in *Proceedings of the European Symposium on Ambient Intelligence*, pp. 252–263, 2003.

[15] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proceedings of the Joint Conference on Smart Objects and Ambient Intelligence*, pp. 159–163, 2005.

[16] H. Kemper and R. Verschuur, "Validity and reliability of pedometers in habitual activity research," *European Journal of Applied Physiology*, vol. 37, no. 1, pp. 71–82, 1977.

[17] N. Kern, B. Schiele and A. Schmidt, "Multi-sensor activity context detection for wearable computing," in *Proceedings of the European Symposium on Ambient Intelligence*, pp. 220–232, 2003.

[18] P. Kuryloski, A. Giani, R. Giannantonio, K. Gilani, V. Seppa, E. Seto, R. Gravina, V. Shia, C. Wang, P. Yan, A. Yang, J. Hyttinen, S. Sastry, S. Wicker and R. Bajcsy, "DexterNet: An open platform for heterogeneous body sensor networks and its applications," in *Proceedings of the Body Sensor Networks Workshop*, 2009.

[19] P. Lukowicz, J. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash and T. Starner, "Recognizing workshop activity using body worn microphones and accelerometers," in *Proceedings of the International Conference on Pervasive Computing*, pp. 18–32, 2004.

[20] J. Mantyjarvi, J. Himberg and T. Seppanen, "Recognizing human motion with multiple acceleration sensors," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 747–752, 2001.

[21] T. Martin, B. Majeed, B. Lee and N. Clarke, "Fuzzy ambient intelligence for next generation telecare," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 894–901, 2006.

[22] M. Mathie, A. Coster, N. Lovell and B. Celler, "Accelerometry: Providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiological Measurement*, vol. 25, pp. R1–R20, 2004.

[23] J. Morrill, "Distributed recognition of patterns in time series data," *Communications of the ACM*, vol. 41, no. 5, pp. 45–51, 1998.

[24] B. Najafi, K. Aminian, A. Parschiv-Ionescu, F. Loew, C. Büla and P. Robert, "Ambulatory system for human motion analysis using a kinematic sensor: Monitoring of daily physical activity in the elderly," *IEEE Trans. Biomedical Engineering*, vol. 50, no. 6, pp. 711–723, 2003.

[25] I. Pappas, T. Keller, S. Mangold, M. Popovic, V. Dietz and M. Morari, "A reliable gyroscope-based gait-phase detection sensor embedded in a shoe insole," *IEEE Sensors Journal*, vol. 4, no. 2, pp. 268–274, 2004.

[26] S. Pirttikangas, K. Fujinami and T. Nakajima, "Feature selection and activity recognition from wearable sensors," in *Proceedings of the International Symposium on Ubiquitous Computing Systems*, 2006.

[27] C. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, pp. 265–278, 2006.

[28] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly," *Pervasive Computing*, pp. 42–47, 2004.

[29] A. Yang, R. Jafari, P. Kuryloski, S. Iyengar, S. Sastry and R. Bajcsy, "Distributed segmentation and classification of human actions using a wearable sensor network," in *Proceedings of the CVPR Workshop on Human Communicative Behavior Analysis*, 2008.

[30] G. Williams, K. Doughty, K. Cameron and D. Bradley, "A smart fall and activity monitor for telecare applications," in *Proceedings of the IEEE International Conference in Medicine and Biology Society*, 1998.

[31] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin and J. Stankovic, "ALARM-NET: Wireless sensor networks for assisted-living and residential monitoring," Technical report, Department of Computer Science, University of Virginia, 2006.

[32] J. Wright, A. Yang, A. Ganesh, S. Sastry and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[33] W. Zhang, L. He, Y. Chow, R. Yang and Y. Su, "The study on distributed speech recognition system," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1431–1434, 2000.