Distributed Segmentation and Classification of Human Actions Using a Wearable Motion Sensor Network*

Allen Y. Yang, Sameer Iyengar, Shankar Sastry, Ruzena Bajcsy
Department of EECS
University of California, Berkeley

Philip Kuryloski
Department of ECE
Cornell University

Roozbeh Jafari
Department of EE
University of Texas, Dallas

Abstract

We propose a distributed recognition method to classify human actions using a low-bandwidth wearable motion sensor network. Given a set of pre-segmented motion sequences as training examples, the algorithm simultaneously segments and classifies human actions, and it also rejects outlying actions that are not in the training set. The classification is distributedly operated on individual sensor nodes and a base station computer. We show that the distribution of multiple action classes satisfies 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 representation. We further provide fast linear solvers to compute such representation via ℓ^1-minimization. Using up to eight body sensors, the algorithm achieves state-of-the-art 98.8% accuracy on a set of 12 action categories. We further demonstrate that the recognition precision only decreases gracefully using smaller subsets of sensors, which validates the robustness of the distributed framework.

1. Introduction

We study human action recognition using a distributed wearable motion sensor network. Action recognition has been studied to a great extent in computer vision in the past. Compared with a model-based or appearance-based vision system, the body sensor network approach has the following advantages: 1. The system does not require to instrument the environment with cameras or other sensors. 2. The system has the necessary mobility to support continuous monitoring of a subject during her daily activities. 3. With the continuing miniaturization of mobile processors and 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 (e.g., breathing and spine movements). 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 Wii game controllers and smart phones).

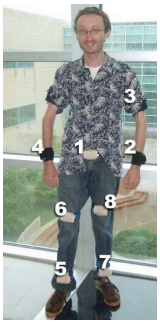

Figure 1. A wireless body sensor system.

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 such system, each sensor node will 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 the server.[1] On the server side, a global classifier receives data from the sensor nodes and further optimizes the classification. The global classifier can

*Corresponding author: yang@eecs.berkeley.edu. This work was partially supported by ARO MURI W911NF-06-1-0076, NSF TRUST Center, and the startup funding from the University of Texas and Texas Instruments.

[1] 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 [18]. Hence it is paramount in sensor networks to reduce the communication cost while preserve the recognition performance.

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.

Past studies on sensor-based action recognition were primarily focused on single accelerometers [8, 10] or other motion sensors [11, 16]. More recent systems prefer using multiple motion sensors [1, 2, 9, 12–14, 17]. Depending on the type of sensor used, an action recognition system is typically composed of two parts: a feature extraction module and a classification module.

There are three major directions for feature extraction in wearable sensor networks. The first direction uses simple statistics of a signal 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* [10, 16]. The third type is based on classical dimensionality reduction techniques such as *principal component analysis* (PCA) and *linear discriminant analysis* (LDA) [13, 14]. 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 [10, 16, 17]. Other more sophisticated techniques have also been used, such as *decision trees* [2, 3] and *hidden Markov models* [13].

For distributed pattern recognition, there exist studies on distributed speech recognition [20] and distributed expert systems [15]. One particular problem associated with most distributed sensor systems is that each local observation from the distributed sensors 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.

Design of the wearable sensor network. Our wearable sensor network consists of sensor nodes placed at various body locations, 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 8MHz microcontroller with 10K RAM and communicates using the 802.15.4 wireless protocol. Each custom-built sensor board has a triaxial accelerometer and a biaxial gyroscope, which is attached to Tmote Sky (shown in Fig 2). Each axis is reported as a 12bit value to the node, indicating values in the range of $\pm 2g$ and $\pm 500^\circ/s$ for the accelerometer and gyroscope, respectively.

To avoid packet collision in the *wireless* channel, we use a TDMA protocol that allocates each node a specific time slot during which to transmit data. This allows us to receive sensor data at 20Hz 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 mote using *nesC*, a variant of C.

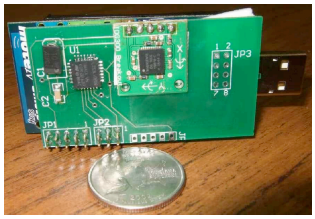

Figure 2. The sensor board with the accelerometer and gyroscope. The mother board at the back is Tmote Sky.

Problem definition. Assume a set of L wearable sensor nodes with triaxial accelerometers and biaxial gyroscopes are attached to the human body. Denote $\boldsymbol{a}_l(t) = (x_l(t), y_l(t), z_l(t), \theta_l(t), \rho_l(t))^T \in \mathbb{R}^5$ as the measurement of the five sensors on node l at time t, and $\boldsymbol{a}(t) = (\boldsymbol{a}_1^T(t), \boldsymbol{a}_2^T(t), \cdots, \boldsymbol{a}_L^T(t))^T \in \mathbb{R}^{5L}$ collects all sensor measurement. Denote $\boldsymbol{s} = (\boldsymbol{a}(1), \boldsymbol{a}(2), \cdots, \boldsymbol{a}(l)) \in \mathbb{R}^{5L \times l}$ as an action sequence of length l.

Given K different classes of human actions, a set of n_i training examples $\{\boldsymbol{s}_{i,1}, \cdots, \boldsymbol{s}_{i,n_i}\}$ are collected for each ith class. The durations of the sequences naturally may be different. Given a new test sequence $\boldsymbol{s}$ that may contain *multiple* actions and possible other *outlying* actions, we seek a distributed algorithm to simultaneously segment the sequence and classify the actions.

Solving this problem mainly involves the following challenges:

1. *Simultaneous segmentation and classification*. We seek simultaneous segmentation and recognition from a long motion sequence. Furthermore, we also assume that the test sequence may contain other unknown actions that are not from the K classes. The algorithm needs to be robust to these *outliers*.

2. *Variation of action durations*. One major difficulty in segmentation of actions is to determine the duration of a proper action. In practice, the durations of different actions vary dramatically (see Fig 3).

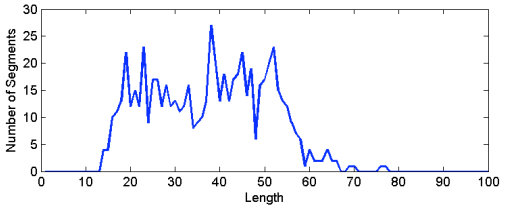

Figure 3. Population of different action durations in our data set.

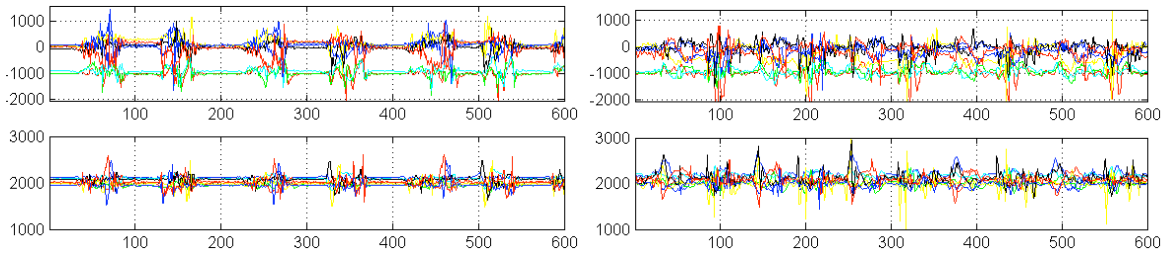

Figure 4. Readings of the x-axis accelerometers (top) and x-axis gyroscopes (bottom) from 8 distributed sensors (shown in different colors) on two repetitive "stand-kneel-stand" sequences from two subjects as the left and right columns.

3. *Identity independence*. In addition to the variation of action durations, different people act differently for the same actions (see Fig 4). For a test sequence in the experiment, we examine the identity-independent performance by excluding the training samples of the same subject.

4. *Distributed recognition*. A distributed recognition system 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 in real time using low-power processors? 3. How to design a classifier to globally optimize the recognition and be adaptive to the change of the network?

Contributions of the paper. We propose a *distributed* action recognition algorithm that simultaneously segments and classifies 12 human actions using up to 8 wearable motion sensors. The work is inspired by an emerging theory of compressed sensing and sparse representation [4, 5]. We assume each action class satisfies a low-dimensional *subspace* model. We show that a 10-D *LDA* feature space suffices to locally represent the 12 action subspaces on each node. 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. The implementation of the system consists of three integrated components: 1. Multi-resolution action feature extraction. 2. Fast distributed classifiers via ℓ^1-minimization. 3. An adaptive global classifier. The method can accurately segment and classify human actions from a continuous motion sequence. The local classifiers that reject potential outliers 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 overall classification upon available local decisions.

Finally, the research of action recognition using wearable sensors in pattern recognition has been hindered to an extent by a lack of rigorous and public database/benchmark in order to judge the performance and safeguard the reproducibility of extant algorithms. We intend to address this issue by constructing and maintaining a public benchmark system called "Wearable Action Recognition Database" (WARD). The database will contain more human subjects across multiple age groups, and it will be made available on our website.

2. Classification via Sparse Representation

We first present an efficient action classification method on each sensor node assuming action sequences are pre-segmented. Given an action segment of length l from node j, $\boldsymbol{s}_j = (\boldsymbol{a}_j(1), \boldsymbol{a}_j(2), \cdots, \boldsymbol{a}_j(l)) \in \mathbb{R}^{5\times l}$, define a new vector $\boldsymbol{s}_j^S$ as the *stacking* of the l columns of $\boldsymbol{s}_j$:

$$\boldsymbol{s}_j^S \doteq (\boldsymbol{a}_i(1)^T, \boldsymbol{a}_i(2)^T, \cdots, \boldsymbol{a}_i(l)^T)^T \in \mathbb{R}^{5l}. \qquad (1)$$

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

Since the length l varies among different subjects and actions, we need to normalize l to be the same for all the training and test samples, which can be achieved by linear interpolation or FFT interpolation. After normalization, we denote the dimension of samples $\boldsymbol{s}_j$ as $D_j = 5l$. Subsequently, we define a *full-body action vector* $\boldsymbol{v}$ that stacks the measurement from all L nodes:

$$\boldsymbol{v} = (\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, \cdots, \boldsymbol{s}_L^T)^T \in \mathbb{R}^D, \qquad (2)$$

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 $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{n_i}\}$ of the ith class sufficiently span the ith action subspace, given a test sample $\boldsymbol{y} = (\boldsymbol{y}_1^T, \cdots, \boldsymbol{y}_L^T)^T \in \mathbb{R}^D$ in the same class i, $\boldsymbol{y}$ can be linearly represented using the training examples of the same class:

$$\begin{aligned} &\boldsymbol{y} = \alpha_1\boldsymbol{v}_1 + \cdots + \alpha_{n_i}\boldsymbol{v}_{n_i} \\ \Leftrightarrow \quad &\begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_L \end{pmatrix} = \left(\begin{pmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \vdots \\ \boldsymbol{s}_L \end{pmatrix}_1 \cdots \begin{pmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \vdots \\ \boldsymbol{s}_L \end{pmatrix}_{n_i} \right) \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_i} \end{pmatrix}. \end{aligned} \qquad (3)$$

It is important to note that such linear constraint also holds on each node j: $\boldsymbol{y}_j = \alpha_1 \boldsymbol{s}_{j,1} + \cdots + \alpha_{n_i} \boldsymbol{s}_{j,n_i} \in \mathbb{R}^{D_j}$.

In theory, complex data such as human actions typically constitute complex nonlinear models. The linear models are used to *approximate* such nonlinear structures in a higher-dimensional subspace (see Fig 5). 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.

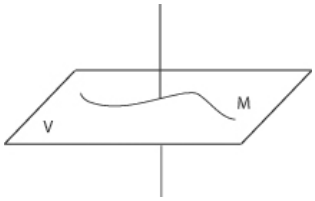

Figure 5. Modeling a 1-D manifold M using a 2-D subspace V.

To recover label($\boldsymbol{y}$), a previous study [19] proposed to reformulate the recognition using a global sparse representation: Since label($\boldsymbol{y}$) $= i$ is unknown, we can represent $\boldsymbol{y}$ using all the training samples from all K classes:

$$\boldsymbol{y} = (A_1, A_2, \cdots, A_K) \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_K \end{pmatrix} = A\boldsymbol{x}, \tag{4}$$

where $A_i = (\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \cdots, \boldsymbol{v}_{i,n_i}) \in \mathbb{R}^{D \times n_i}$ collects all the training samples of class i, $\boldsymbol{x}_i = (\alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,n_i})^T \in \mathbb{R}^{n_i}$ collects the corresponding coefficients in (3), and $A \in \mathbb{R}^{D \times n}$ where $n = n_1 + n_2 + \cdots + n_K$. Since $\boldsymbol{y}$ satisfies both (3) and (4), one solution of $\boldsymbol{x}$ in (4) should be $\boldsymbol{x}^* = (0, \cdots, 0, \boldsymbol{x}_i^T, 0, \cdots, 0)^T$. The solution is naturally *sparse*: in average only $\frac{1}{K}$ terms in $\boldsymbol{x}^*$ are nonzero.

On each sensor j, solution $\boldsymbol{x}^*$ of (4) is also a solution for the representation:

$$\boldsymbol{y}_j = (A_1^{(j)}, A_2^{(j)}, \cdots, A_K^{(j)})\boldsymbol{x} = A^{(j)}\boldsymbol{x}, \tag{5}$$

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 (4) or locally using (5), 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 (5) is the high dimensionality of the action data. In *compressed sensing* [4, 5], one reduces the dimension of a linear system by choosing a linear projection $R_j \in \mathbb{R}^{d \times D_j}$:[2]

$$\tilde{\boldsymbol{y}}_j \doteq R_j \boldsymbol{y}_j = R_j A^{(j)} \boldsymbol{x} \doteq \tilde{A}^{(j)} \boldsymbol{x} \in \mathbb{R}^d. \tag{6}$$

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 (6) is underdetermined. Numerically stable solutions exist to *uniquely* recover sparse solutions $\boldsymbol{x}^*$ via ℓ^1-minimization [6]:

$$\boldsymbol{x}^* = \arg\min \|\boldsymbol{x}\|_1 \text{ subject to } \tilde{\boldsymbol{y}}_j = \tilde{A}^{(j)}\boldsymbol{x}. \tag{7}$$

In our experiment, we have tested multiple projection operators including PCA, LDA, and random project studied in [19]. We found that 10-D feature spaces using LDA lead to best recognition in a very low-dimensional space.

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

$$\delta_i(\boldsymbol{x}) = (0, \cdots, 0, \boldsymbol{x}_i^T, 0, \cdots, 0)^T \in \mathbb{R}^n, i = 1, \cdots, K. \tag{8}$$

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

$$\text{label}(\boldsymbol{y}_j) = \arg\min_i \|\tilde{\boldsymbol{y}}_j - \tilde{A}^{(j)}\delta_i(\boldsymbol{x})\|_2. \tag{9}$$

Example 1 (Classification on Nodes) *We designed 12 action categories in the experiment: Stand-to-Sit, Sit-to-Stand, Sit-to-Lie, Lie-to-Sit, Stand-to-Kneel, Kneel-to-Stand, Rotate-Right, Rotate-Left, Bend, Jump, Upstairs, and Downstairs. The detailed experiment setup is given in Section 4.*

To implement ℓ^1-minimization on the sensor node, we look for fast sparse solvers in the literature. We have tested a variety of methods including (orthogonal) matching pursuit (MP), basis pursuit (BP), LASSO, and a quadratic log-barrier solver.[3] *We found that BP [7] gives the best trade-off between speed, noise tolerance, and recognition accuracy.*

Here we demonstrate the accuracy of the BP-based algorithm on each sensor node (see Fig 1 for their locations). The actions are manually segmented from a set of long motion sequences from three subjects. In total there are 626 samples in the data set. The 10-D feature selection is via LDA. *We require the classification to be* identity-independent. *The accuracy of the classification is shown in Table 1. Fig 6 shows an example of the estimated sparse coefficients $\boldsymbol{x}$ and its residuals. In terms of the speed, our simulation in MATLAB takes in average 0.03s to process one test sample on a typical 3G Hz PC.*

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

[3] The implementation of these routines in MATLAB is available via *SparseLab*: http://sparselab.stanford.edu

Table 1. Recognition accuracy on each node over 12 action classes.

Sen #	1	2	3	4	5	6	7	8
Acc [%]	99.9	99.4	99.9	100	95.3	99.5	93	100

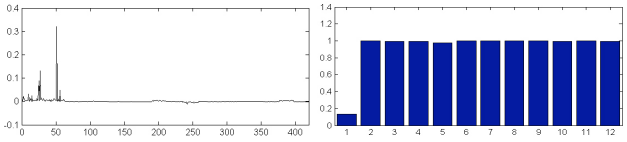

Figure 6. Left: Sparse ℓ^1 solution by BP for a Stand-to-Sit action on the waist node. Right: Corresponding residuals. The action is correctly classified as Class 1. $\text{SCI}(\boldsymbol{x}) = 0.7$ (see (10)).

Example 1 shows that if the segmentation of the actions is known and there is no other invalid samples, all sensor nodes can recognize the 12 actions individually with very high accuracy, which also verifies that the mixture subspace model is a good approximation of the action data. Nevertheless, one may question that in such low-dimensional feature spaces other classical methods (e.g., kNN and decision tree methods) should also perform well. In the next section, we will show that the major advantage of adopting the sparse representation framework is a unified solution to recognize and segment valid actions and reject invalid ones. We will also show that the method is adaptive to the change of available sensor nodes on the fly.

3. Distributed Segmentation and Recognition

We start by introducing multi-resolution action segmentation on each sensor node. From the training examples, we can estimate a range of possible lengths for all actions of interest. We then evenly divide the range into multiple length hypotheses: $(h_1, \cdots, h_s)$. At each time t in a motion sequence, the node tests a set of s possible segmentations: $\boldsymbol{y}(1) = (a(t-h_1), \cdots, a(t)), \cdots, \boldsymbol{y}(s) = (a(t-h_s), \cdots, a(t))$, as shown in Fig 7.[4] With each candidate $\boldsymbol{y}$ again normalized to length l, a sparse representation $\boldsymbol{x}$ is estimated using ℓ^1-minimization in Section 2.

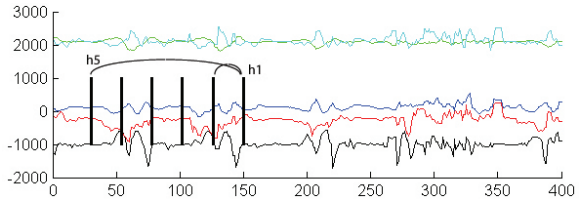

Figure 7. Multiple segmentation hypotheses on a wrist sensor at time $t = 150$ of a "go downstairs" sequence. h_1 is a good segment while others are false segments. Notice that the movement between 250 and 350 is an outlying action that the subject performed.

Based on the previous sparsity assumption, if $\boldsymbol{y}$ is not a valid segmentation w.r.t. the training examples due to either incorrect t or h, or the real action performed is not in the training classes, the dominant coefficients of its sparsest representation $\boldsymbol{x}$ should not correspond to any single class. We use a *sparsity concentration index* (SCI) [19]:

$$\text{SCI}(\boldsymbol{x}) \doteq \frac{K \cdot \max_{j=1,\cdots,K} \|\delta_j(\boldsymbol{x})\|_1 / \|\boldsymbol{x}\|_1 - 1}{K-1} \in [0,1]. \tag{10}$$

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 to all sensor nodes: If $SCI(\boldsymbol{x}) > \tau_1$, the segment is a valid local measurement, and its 10-D LDA features $\tilde{\boldsymbol{y}}$ will be sent to the base station.

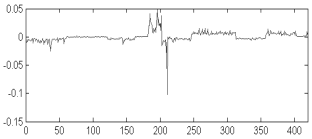

Figure 8. A invalid representation (SCI=0.13).

Next, we introduce a global classifier that adaptively optimizes the overall segmentation and classification. Suppose at time t and with a length hypothesis h, 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: $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \cdots, \tilde{\boldsymbol{y}}_{L'}$. Let $\tilde{\boldsymbol{y}}' = (\tilde{\boldsymbol{y}}_1^T, \cdots, \tilde{\boldsymbol{y}}_{L'}^T)^T \in \mathbb{R}^{10L'}$. Then the global sparse representation $\boldsymbol{x}$ of $\tilde{\boldsymbol{y}}'$ satisfies the following linear system

$$\tilde{\boldsymbol{y}}' = \begin{pmatrix} R_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & R_{L'} & \cdots & 0 \end{pmatrix} A\boldsymbol{x} = R'A\boldsymbol{x} = \tilde{A}'\boldsymbol{x}, \tag{11}$$

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 (6) then becomes a special case of (11) when $L' = 1$.

Similar to the outlier rejection criterion we previously proposed on each node, we introduce a global rejection threshold τ_2. If $\text{SCI}(\boldsymbol{x}) > \tau_2$ in (11), the most significant coefficients in $\boldsymbol{x}$ are concentrated in a single training class. Hence $\tilde{y}'$ is assigned to that class, and its length hypothesis h provides the segmentation of the action from the motion sequence.

The overall algorithm on the nodes and on the network server provides a unified solution to segment and classify action segments from a motion sequence using only two simple parameters τ_1 and τ_2. Typically τ_1 is selected to be less restricted than τ_2 in order to increase the recall rate, because

[4] Those segmentation hypotheses that overlap with previously detected actions will be ignored to avoid temporal ambiguity.

passing certain amounts of false signal to the global classifier is not necessarily disastrous as the signal would be rejected by τ_2 when the action features from multiple nodes are jointly considered. The formulation of adaptive classification (11) via a global projection matrix R' and two sparsity constraints τ_1 and τ_2 provides a simple means of rejecting outliers from a network of multiple sensors. The method compares favorably to other classical methods such as kNN and decision trees, because these methods need to train multiple thresholds and decision rules when the number L' and the set of available sensors vary in the full-body action vector $\tilde{\boldsymbol{y}}' = (\tilde{\boldsymbol{y}}_1^T, \cdots, \tilde{\boldsymbol{y}}_{L'}^T)^T$.

Finally, we consider how the change of active nodes affects ℓ^1-minimization and the classification of the actions. In compressed sensing, the efficacy of ℓ^1-minimization in solving for the sparsest solution $\boldsymbol{x}$ in (11) is characterized by the ℓ^0/ℓ^1 equivalence relation [6, 7]. 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 $\boldsymbol{x}$ is the sparsest solution in (11) for $L' = L$, $\boldsymbol{x}$ is also a solution for $L' < L$. Hence, the decrease of L' leads to possible sparser solutions of $\boldsymbol{x}$.

On the other hand, the decrease in available action features also makes $\tilde{\boldsymbol{y}}'$ less discriminant. For example, if we reduce $L' = 1$ and only activate a wrist sensor, then the ℓ^1-solution $\boldsymbol{x}$ 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 corresponding 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

We validate the performance of the system using a data set we collected from three male subjects at the age of 28, 30, and 32, respectively. Eight wearable sensors were placed at different body locations (see Fig 1). We designed a set of 12 action classes: *Stand-to-Sit* (StSi), *Sit-to-Stand* (SiSt), *Sit-to-Lie* (SiLi), *Lie-to-Sit* (LiSi), *Stand-to-Kneel* (StKn), *Kneel-to-Stand* (KnSt), *Rotate-Right* (RoR), *Rotate-Left* (RoL), *Bend*, *Jump*, *Upstairs* (Up), and *Downstairs* (Down). We are particularly interested in testing the system under various action durations. For this purpose, we have asked the subjects to perform StSi, SiSt, SiLi, and LiSi with two different speeds (slow and fast), and perform RoR and RoL with two different rotation angles (90° and 180°). All subjects were asked to perform a sequence of related actions in each recording session based on their own interpretation of the actions. In total there are 626 actions performed in the data set (see Table 3 for the numbers in individual classes).

Table 2 shows Precision versus Recall of the algorithm with different active sensor nodes. For all experiments, $\tau_1 = 0.2$ and $\tau_2 = 0.4$. When all nodes are activated, the algorithm can achieve 98.8% accuracy among the actions it extracted, and 94.2% of the true actions are detected. The performance decreases gracefully when more nodes become unavailable to the global classifier. Our results show that if we can maintain one motion sensor on the upper body (e.g., at position 2) and one on the lower body (e.g., at position 7), the algorithm can still achieve 94.4% precision and 82.5% recall. Finally, in average the 8 distributed classifiers that reject invalid local measurements reduce the node-to-station communication for above 50%.

Table 2. Precision vs. recall with different sets of activated sensors.

Sensors	2	7	2,7	1,2,7	1- 3, 7,8	1- 8
Prec [%]	89.8	94.6	94.4	92.8	94.6	98.8
Rec [%]	65	61.5	82.5	80.6	89.5	94.2

One may be curious about the relatively low recall on single sensors such as 2 and 7. This performance difference is due to the large number of potential outlying segments presented in a long motion sequence (e.g., see Fig 7). We further compare the difference using two confusion tables 3 and 4. We see that a single node 2 that is positioned on the right wrist performed poorly mainly on two action categories: Stand-Kneel and Upstairs-Downstairs, both of which involve significant movements of the lower body but not the upper one. This is the main reason for the low recall in Table 2. On the other hand, for the actions that are detected using node 2, our system can still achieve about 90% accuracy, which clearly demonstrates the robustness of the distributed recognition framework. Similar arguments also apply to node 7 and other sensor combinations.

Table 3. Confusion table using sensors 1-8.

Class (total)	1	2	3	4	5	6	7	8	9	10	11	12
1 StSi (60)	60	0	0	0	0	0	0	0	0	0	0	0
2 SiSt (60)	0	52	0	0	0	0	0	0	0	0	0	0
3 SiLi (62)	1	0	58	0	0	0	0	0	0	0	0	0
4 LiSi (62)	0	0	0	60	0	0	0	0	0	0	0	0
5 Bend (30)	1	0	0	0	29	0	0	0	0	0	0	0
6 StKn (33)	0	0	0	0	0	31	0	0	0	0	0	0
7 KnSt (30)	0	0	0	0	0	0	30	0	0	0	1	0
8 RoR (95)	0	0	0	0	0	0	0	93	0	0	0	1
9 RoL (96)	0	0	0	0	0	0	0	0	96	0	0	0
10 Jump (34)	0	0	0	0	0	0	0	0	0	31	0	0
11 Up (33)	0	0	0	0	0	0	0	0	0	0	24	0
12 Down (31)	0	0	0	0	0	0	0	0	0	0	3	26

Finally, we provide examples of the classification results on Subject 1 to demonstrate the accuracy of the proposed algorithm using all 1 - 8 sensor nodes. For clarity, each figure in Fig 9 - 21 only plots the readings from x-axis accelerometers on the 8 nodes. The segmentation results are then super-

Table 4. Confusion table using sensor 2.

Class (total)	1	2	3	4	5	6	7	8	9	10	11	12
1 StSi (60)	37	0	2	0	0	0	0	4	0	0	0	0
2 SiSt (60)	0	50	0	0	0	0	0	0	2	0	0	0
3 SiLi (62)	1	0	38	0	0	0	0	0	0	0	0	0
4 LiSi (62)	0	7	0	32	0	0	0	0	0	0	0	0
5 Bend (30)	0	1	0	0	26	0	0	0	0	0	0	0
6 StKn (33)	0	1	0	1	0	7	0	2	3	0	0	0
7 KnSt (30)	0	1	0	0	1	0	6	3	3	0	0	0
8 RoR (95)	0	0	0	0	0	0	0	92	0	0	0	0
9 RoL (96)	0	0	0	0	0	0	0	0	95	0	0	0
10 Jump (34)	0	0	0	0	0	0	0	0	1	24	0	0
11 Up (33)	0	0	0	0	0	0	0	1	8	0	0	0
12 Down (31)	0	0	0	0	0	0	1	0	3	0	0	0

imposed. The black solid boxes indicate the locations of the correctly classified action segments. The red boxes (e.g., in Fig 14) indicate the locations of false classification. One can also observe from the figures that some valid actions are not detected by the algorithm, e.g., in Fig 13.

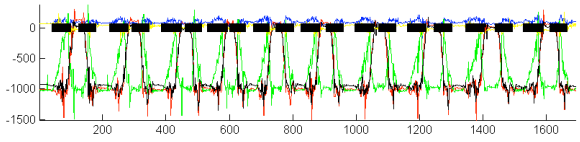

Figure 9. Segmentation of a slow Stand-Sit-Stand sequence.

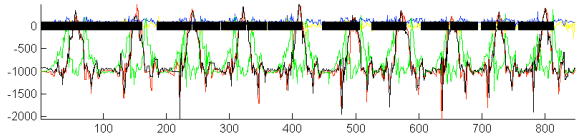

Figure 10. Segmentation of a fast Stand-Sit-Stand sequence.

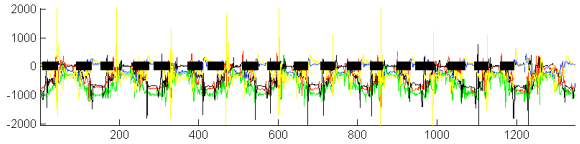

Figure 11. Segmentation of a slow Sit-Lie-Sit sequence.

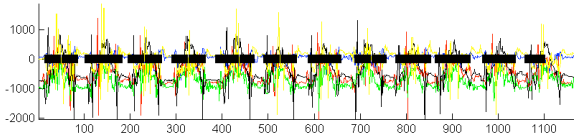

Figure 12. Segmentation of a fast Sit-Lie-Sit sequence.

5. Conclusion and Discussion

Inspired by the emerging compressed sensing theory, we have proposed a distributed algorithm to segment and classify human actions on a wearable motion sensor network.

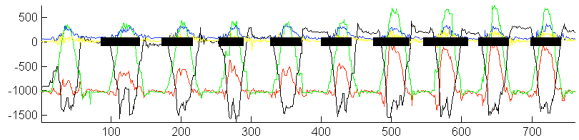

Figure 13. Segmentation of a Bend sequence.

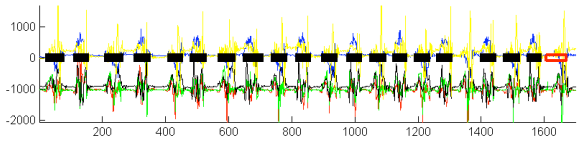

Figure 14. Segmentation of a Stand-Kneel-Stand sequence.

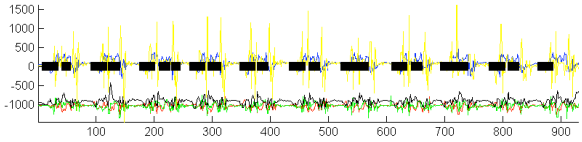

Figure 15. Segmentation of a 90° Rotate-Right-Left sequence.

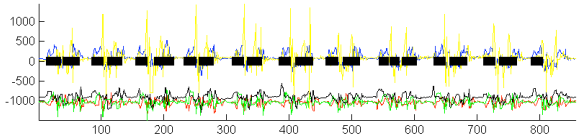

Figure 16. Segmentation of a 90° Rotate-Left-Right sequence.

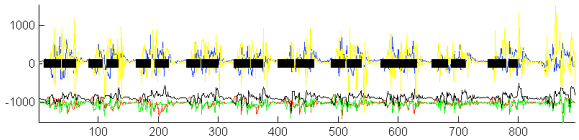

Figure 17. Segmentation of a 180° Rotate-Right sequence.

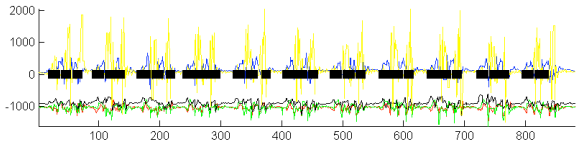

Figure 18. Segmentation of a 180° Rotate-Left sequence.

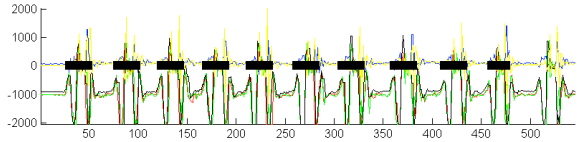

Figure 19. Segmentation of a Jump sequence.

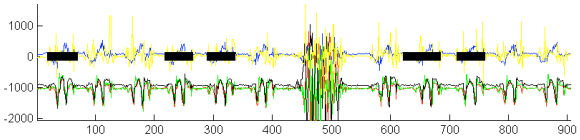

Figure 20. Segmentation of a Go-Upstairs sequence.

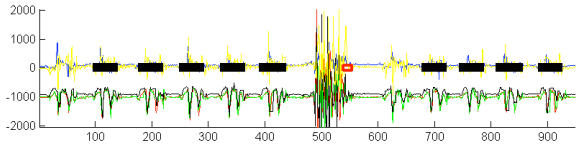

Figure 21. Segmentation of a Go-Downstairs sequence.

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 12 action classes can be accurately represented and classified using a set of 10-D LDA features measured at multiple body locations. The proposed global classifier can adaptively adjust the global optimization to boost the recognition upon available local measurements.

One 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.

References

[1] R. Aylward and J. Paradiso. A compact, high-speed, wearable sensor network for biomotion capture and interactive media. In *IPSN*, 2007.

[2] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive*, 2004.

[3] A. Benbasat and J. Paradiso. Groggy wakeup - automated generation of power-efficient detection hierarchies for wearable sensors. In *Int. Work. on Wearable and Implantable Body Sensor Networks*, 2007.

[4] E. Candès. Compressive sampling. In *Proceedings of the International Congress of Mathematicians*, 2006.

[5] E. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Information Theory*, 52(12):5406–5425, 2006.

[6] D. Donoho. Neighborly polytopes and sparse solution of underdetermined linear equations. *preprint*, 2005.

[7] D. Donoho and M. Elad. On the stability of the basis pursuit in the presence of noise. *Sig. Proc.*, 86:511–532, 2006.

[8] J. Farringdon, A. Moore, N. Tilbury, J. Church, and P. Biemond. Wearable sensor badge & sensor jacket for context awareness. In *Int. Symp. on Wear. Comp.*, 1999.

[9] E. Heinz, K. Kunze, and S. Sulistyo. Experimental evaluation of variations in primary features used for accelerometric context recognition. In *Euro. Symp. on Amb. Intel.*, 2003.

[10] T. Huynh and B. Schiele. Analyzing features for activity recognition. In *J. Conf. on Smart Objects and Ambient Intelligence*, 2005.

[11] H. Kemper and R. Verschuur. Validity and reliability of pedometers in habitual activity research. *European Journal of Applied Physiology*, 37(1):71–82, 1977.

[12] N. Kern, B. Schiele, and A. Schmidt. Multi-sensor activity context detection for wearable computing. In *European Symposium on Ambient Intelligence*, 2003.

[13] 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 *Pervasive*, 2004.

[14] J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In *Int. Conf. on Sys., Man and Cyb.*, 2001.

[15] J. Morrill. Distributed recognition of patterns in time series data. *Communications of the ACM*, 41(5):45–51, 1998.

[16] 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 Transactions on Biomedical Engineering*, 50(6):711–723, 2003.

[17] S. Pirttikangas, K. Fujinami, and T. Nakajima. Feature selection and activity recognition from wearable sensors. In *Int. Symp. on Ubi. Comp. Sys.*, 2006.

[18] C. Sadler and M. Martonosi. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *ACM Conf. on Emb. Net. Sen. Sys.*, pages 265–278, 2006.

[19] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *(in press) PAMI*, 2008.

[20] W. Zhang, L. He, Y. Chow, R. Yang, and Y. Su. The study on distributed speech recognition system. In *Int. Conf. on Acou., Speech, and Sig. Proc.*, pages 1431–1434, 2000.