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

Allen Yang
Roozbeh Jafari
Philip Kuryloski
Sameer Iyengar
S. Shankar Sastry
Ruzena Bajcsy

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2007-143
http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-143.html

December 6, 2007

Acknowledgement

Yang and Sastry are partially supported by ARO MURI W911NF-06-1-0076. Jafari is partially supported by the startup fund from the University of Texas and Texas Instruments. Bajcsy is partially supported by NSF IIS 0724682. Kuryloski, Iyengar, Sastry, and Bajcsy are partially supported by TRUST (Team for Research in Ubiquitous Secure Technology), which receives support from NSF CCF-0424422, AFOSR FA9550-06-1-0244, and the following organizations: Cisco, British Telecom, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia, and United Technologies.

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

Allen Y. Yang, Roozbeh Jafari, Philip J. Kuryloski, Sameer Iyengar, S. Shankar Sastry, and Ruzena Bajcsy

Abstract

We propose a distributed recognition framework to classify human actions using a wearable motion sensor network. Each sensor node consists of an integrated triaxial accelerometer and biaxial gyroscope. Given a set of pre-segmented actions as training examples, the algorithm simultaneously segments and classifies human actions from a motion sequence, and it also rejects unknown actions that are not in the training set. The classification is distributedly operated on individual sensor nodes and a base station computer. Due to rapid advances in the integration of mobile processors and heterogeneous sensors, a distributed recognition system likely outperforms traditional centralized recognition methods. In this paper, we assume 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 provide fast linear solvers to compute such representation via ℓ^1-minimization.

I. Introduction

In this paper, we consider human action recognition on a distributed wearable motion sensor network. Each sensor node is integrated with a triaxial accelerometer and biaxial gyroscope. The locations of the sensors are roughly defined to be the waist, two wrists, left arm, two knees, and two ankles, as shown in Fig 1. Action recognition has been studied to a great extent in computer vision in the past. Compared to 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 integration of mobile processors, sensors, and batteries, 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 oriented surveillance, athletic training, tele-immersion, and human-computer interaction.

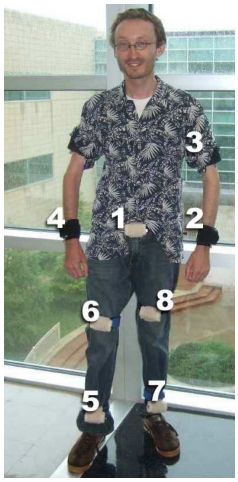


Fig. 1. A distributed wearable sensor network. The sensor on the right arm was malfunctioned during the experiment.

Yang, Iyengar, Sastry, and Bajcsy are with the Department of Electrical Engineering and Computer Science, University of California, Berkeley. Jafari is with the Department of Electrical Engineering, University of Taxes at Dallas. Kuryloski is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, and the Department of Electrical and Computer Engineering, Cornell University. Corresponding author: Allen Y. Yang, Rm 307 Cory Hall, UC Berkeley, Berkeley, CA 94720. Email: yang@eecs.berkeley.edu. Tel: 510-643-5798. Fax: 510-643-2356.

Yang and Sastry are partially supported by ARO MURI W911NF-06-1-0076. Jafari is partially supported by the startup fund from the University of Texas and Texas Instruments. Bajcsy is partially supported by NSF IIS 0724682. Kuryloski, Iyengar, Sastry, and Bajcsy are partially supported by TRUST (Team for Research in Ubiquitous Secure Technology), which receives support from NSF CCF-0424422, AFOSR FA9550-06-1-0244, and the following organizations: Cisco, British Telecom, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia, and United Technologies.

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. With recent advances in power-efficient mobile processors for sensor networks (e.g., FPGA and Intel XScale series), we are interested in studying new frameworks for *distributed pattern recognition*. In such systems, 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 becomes *active* and transmit the measurement to the server. On the server side, a global classifier receives data from the sensor nodes and further optimizes the classification. The global classifier can be more computationally involved than the distributed classifiers, but it has to adapt to the change of available active sensors due to local measurement error, sensor failure, and communication congestion.

Distributed pattern recognition on sensor networks has several advantages: 1. Good decisions about the validity of the local information can reduce the communication between the nodes and the server, and therefore reduce power consumption. Previous studies have shown the power consumption required to send one byte over a wireless network is equivalent to executing between $1e3$ and $1e6$ instructions on an onboard processor [21]. 2. The framework increases the robustness of action recognition on the network. Particularly, as we will show later, one can choose to activate some or all of the sensor nodes on the fly, and the global classifier is able to adaptively adjust the optimization process and improve the recognition upon local decisions. 3. The ability for the sensor nodes to make biased local decisions also makes the design of the global classifier more flexible. For example, a system that only monitors abnormal movements (e.g., falling or no movement) can make fairly good estimation using local decisions and discard the global optimization, and in cases that the central system fails, the network can still support limited recognition tasks using the distributed classifiers. 4. Finally, in a more general perspective beyond action recognition, the ability for individual sensor nodes to make local decisions can be used as feedback to support certain autonomous actions/reactions without relying on the intervention of a central system.

We define distributed action recognition as follows:

Problem 1 (Distributed segmentation and classification): Assume a set of L wearable sensor nodes with integrated triaxial accelerometers and biaxial gyroscopes are attached to multiple locations of 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 difficulties.

1) *Simultaneous segmentation and classification*. If the test sequence is pre-segmented, classification becomes straightforward with many classical algorithms to choose from. In this paper, 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 action recognition is to determine the duration of an action. Good classification depends on correct estimation of both the starting time and the duration of an action. But in practice, the durations of different actions may vary dramatically (see Fig 2).

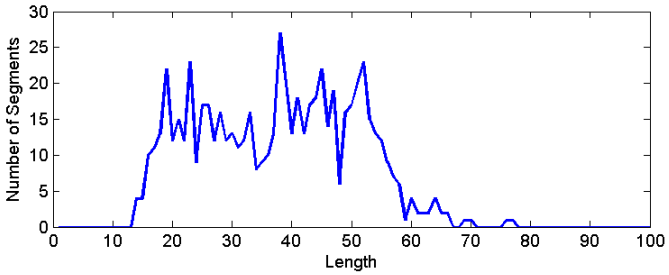


Fig. 2. Population of different action durations in our data set.

3) *Identity independence*. In addition to the variation of action durations, different people act differently for the same actions (see Fig 3). If both the training samples and the test samples are from the same subject, typically the classification could be greatly simplified. However, it is well known that collecting large numbers of training samples in human biometrics is expensive, particularly in medical-care oriented applications. Therefore it is desirable for an action recognition algorithm to be identity independent. For a test sequence in the experiment, we examine the identity-independent performance by excluding the training samples of the same subject.

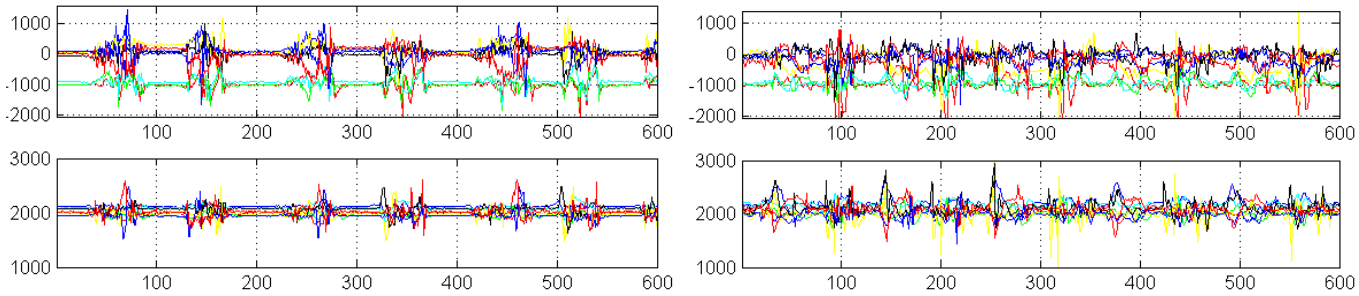


Fig. 3. Readings of the x-axis accelerometers (top) and x-axis gyroscopes (bottom) from 8 distributed sensors (shown in different colors) on two repetitive "stand-kneel-stand" sequences from two subjects as the left and right columns.

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?

a) Literature Overview.: Action (or activity) recognition using wearable motion sensors has been a prominent topic in the last five years. Initial studies were primarily focused on single accelerometers [9], [11] or other motion sensors [12], [19]. More recent systems prefer using multiple motion sensors [1], [2], [10], [13], [16], [17], [20]. 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 [2], [10], [11], [13], [20]. The second type of feature is computed using fixed filter banks such as *FFT* and *wavelets* [11], [19]. The third type is based on classical dimensionality reduction techniques such as *principal component analysis* (PCA) and *linear discriminant analysis* (LDA) [16], [17]. 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 implemented on mobile devices [11], [19], [20]. Other more sophisticated techniques have also been used, such as *decision trees* [2], [3] and *hidden Markov models* [16].

For distributed pattern recognition, there exist initial studies on distributed speech recognition [23] and distributed expert systems [18]. In [23], the authors summarized three major categories of distributed recognition:[1] 1. All data are relayed to a computer server for processing, e.g., on a closed-circuit camera system [14]. 2. All data are locally processed, e.g., [15]. One may further choose to implement a global classifier by a majority-voting scheme on local decisions. 3. A full-fledged distributed recognition system consists of both front-end processing for feature extraction and global processing for classification [6], [13], [16], [17], [20]. Our distributed action recognition system falls into the last category. One particular problem associated with this category is that each local observation from the distributed sensors is *biased* and may be *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. Consequently, one can not expect majority-voting type classifiers to perform well globally.

b) Contributions of the paper.: We propose a *distributed* action recognition algorithm that simultaneously segments and classifies 12 human actions using 1- 8 wearable motion sensor nodes. We assume the wearable sensor network is a typical one-hop wireless network and all the sensor nodes communicate with a central computer. The work is inspired by a recent study on face recognition using sparse representation and ℓ^1-minimization [22]. 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. We further study fast linear programming routines to solve for such sparse representation.

We investigate a distributed framework for simultaneous segmentation and classification of individual actions from a motion sequence. On each sensor node, a classifier searches for good segmentation on multiple temporal resolutions. We propose an effective method to reject action segments that do not correspond to any training class as outliers. Hence an inlying action segment simultaneously provides the localization of the action and its membership.

[1]In certain situations it is desirable to consider a complete distributed recognition system where there is no central system and the recognition on the nodes converge over time via node-to-node communications. In this paper, having a base station is still a practical and efficient solution.

When a sensor node detects a valid action segment, it transmits its 10-D feature to the server. The global classifier receives the distributed feature vectors, and then seeks a global sparse representation of the action features against the corresponding feature vectors of all the training samples. The global optimization is adaptive to the change of available active nodes.

The focus of this paper is about the distributed action recognition framework. The algorithm is software simulated in MATLAB. Currently our data set is mainly designed for transient actions (e.g., jumping, kneeling, and stand-to-sit), but it also contains a limited number of nontransient actions (i.e., turning, going upstairs and downstairs). We are in the process of gradually expanding the number of subjects and action classes in the database.

II. Design of the Wearable Sensor Network

The 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 4). 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. Each node is currently powered by two AA batteries.

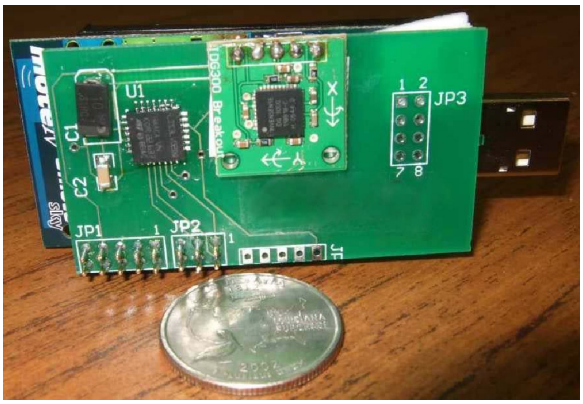


Fig. 4. The sensor board with the accelerometer and gyroscope. The mother board at the back is Tmote Sky.

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 indication 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.

To avoid packet collision in the network, 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.

III. Classification via Sparse Representation

In this section, we present an efficient action classification method to recognize pre-segmented action sequences on each sensor node via ℓ^1-minimization. We first discuss the representation of action samples in vector form. 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 $\boldsymbol{s}_j$:

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

We will interchangeably use $\boldsymbol{s}_j$ and $\boldsymbol{s}_j^S$ to denote the stacked vector 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 new vector $\boldsymbol{v}$ that stacks the measurement from all L nodes:

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

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:

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

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.

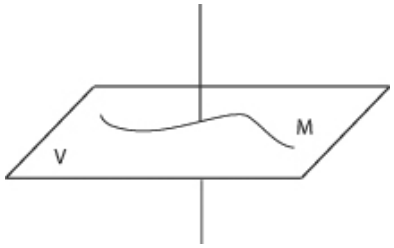


Fig. 5. Modeling a 1-D manifold M using a 2-D subspace V.

In this paper, we are interested in recovering label($\boldsymbol{y}$). A previous study [22] 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 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

$$\boldsymbol{x}^* = (0, \cdots, 0, \boldsymbol{x}_i^T, 0, \cdots, 0)^T. \tag{5}$$

The solution is naturally *sparse*: in average only $\frac{1}{K}$ terms in $\boldsymbol{x}^*$ are nonzero. Furthermore, $\boldsymbol{x}^*$ is also a solution for the representation on each node j:

$$\boldsymbol{y}_j = (A_1^{(j)}, A_2^{(j)}, \cdots, A_K^{(j)}) \cdot \boldsymbol{x} = A^{(j)}\boldsymbol{x}, \tag{6}$$

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 (6), 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 IV. In the rest of this section however, our focus will be on each node.

One major difficulty in solving (6) is the high dimensionality of the action data. For example, in this paper, we normalize $l = 64$ for all action segments (see Fig 2 for the distribution of original lengths). Then $D_j = 64 \times 5 = 320$ for $\boldsymbol{y}_j$ on each node. The high dimensionality makes it difficult to either directly solve for $\boldsymbol{x}$ on the node or transmit the action data over a band-limited wireless channel. 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]

$$\tilde{\boldsymbol{y}}_j \doteq R_j \boldsymbol{y}_j = R_j A^{(j)} \boldsymbol{x} \doteq \tilde{A}^{(j)} \boldsymbol{x} \in \mathbb{R}^d. \tag{7}$$

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

As a result, the action feature $\tilde{\boldsymbol{y}}_j$ is more efficient to transmit than $\boldsymbol{y}_j$ in the original data space D_j. On the network server, the global action vector is of the following form:

$$\tilde{\boldsymbol{y}} = \begin{pmatrix} \tilde{\boldsymbol{y}}_1 \\ \tilde{\boldsymbol{y}}_2 \\ \vdots \\ \tilde{\boldsymbol{y}}_L \end{pmatrix} = \begin{pmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ & \vdots & \vdots & \\ 0 & 0 & \cdots & R_L \end{pmatrix} \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_L \end{pmatrix} \doteq R\boldsymbol{y} \in \mathbb{R}^{dL}, \tag{8}$$

where $R \in \mathbb{R}^{dL \times D}$ is equivalent to a global projection matrix.

After the projection R_j, typically the feature dimension d is much smaller than the number n of all training samples. Therefore, the new linear system (7) is underdetermined. Numerically stable solutions exist to *uniquely* recover sparse solutions $\boldsymbol{x}^*$ via ℓ^1-minimization [7]:

$$\boldsymbol{x}^* = \arg\min \|\boldsymbol{x}\|_1 \text{ subject to } \tilde{\boldsymbol{y}}_j = \tilde{A}^{(j)}\boldsymbol{x}. \tag{9}$$

In our experiment, we have tested multiple projection operators including PCA, LDA, and random project advocated in [22]. 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

$$\delta_i(\boldsymbol{x}) = (0, \cdots, 0, \boldsymbol{x}_i^T, 0, \cdots, 0)^T \in \mathbb{R}^n,\ i = 1, \cdots, K. \tag{10}$$

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

$$\text{label}(\boldsymbol{y}_j) = \arg\min_i \|\tilde{\boldsymbol{y}}_j - \tilde{A}^{(j)}\delta_i(\boldsymbol{x})\|_2. \tag{11}$$

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 V.

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 [8] 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*. Therefore, for each test sample from a subject, we use all samples from the other two subjects to form the training set. The accuracy of the classification is shown in Table I. 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 PC.

TABLE I
RECOGNITION ON EACH NODE ON 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

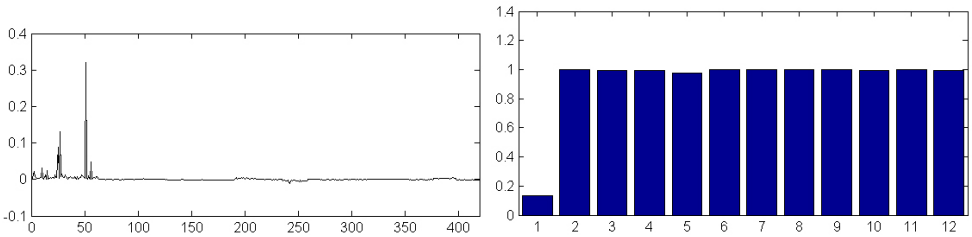


Fig. 6. A BP-based ℓ^1 solution and its corresponding residuals of a Stand-to-Sit action on the waist node. The action is correctly classified as class 1. $\text{SCI}(\boldsymbol{x}) = 0.7$ (see (13)).

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

IV. Distributed Segmentation and Recognition

There have been two major approaches in the past to provide partial solutions to simultaneous segmentation and recognition of human actions on wearable sensors. The first solution assumes different actions are separated by a "rest" state, and such states can be detected by energy thresholding or a special classifier to distinguish between *rest* and *non-rest*. The second solution assumes all sensors in the network are available at all time, and rejects invalid samples based on the sample distance between the test and training examples. These two approaches have several drawbacks: 1. For the first approach, the validity of the rest state between actions is not physically guaranteed. For example, nontransient actions such as walking and running may last for a long period. 2. The second approach is not robust when the number of active sensors changes over time. In this case, tuning a list of different distance thresholds to reject outliers when the number of sensors changes can be difficult, which still highly depends on the condition on the training samples.

We propose a novel framework to simultaneously segment and recognize human actions using the (10-D LDA) action features extracted from a network of distributed sensors. The unified outlier rejection method applies to both individual nodes and the global classifier. The outlying action segments may be caused by unknown actions performed by the subjects or by incorrect segmentation. As a result, the extracted inlying action segments simultaneously provide the segmentation of the actions and their labels. The framework is also robust w.r.t. different action durations and the change of available sensor nodes.

We first introduce multi-resolution action detection 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: [4]

$$\boldsymbol{y}(1) = (a(t-h_1), \cdots, a(t)), \cdots, \boldsymbol{y}(s) = (a(t-h_s), \cdots, a(t)), \tag{12}$$

as shown in Fig 7. With each candidate $\boldsymbol{y}$ normalized to length l, a sparse representation $\boldsymbol{x}$ is estimated using ℓ^1-minimization in Section III.

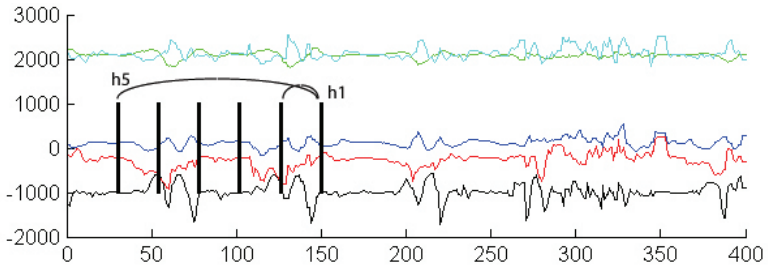


Fig. 7. Multiple segmentation hypotheses on a wrist sensor at time $t = 150$ of a "go downstairs" sequence. h_1 is a good segment while others are false segments. Notice that the movement between 250 and 350 is an outlying action the subject performed.

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 (as shown in Fig 8). We use a *sparsity concentration index* (SCI) [22]:

$$\text{SCI}(\boldsymbol{x}) \doteq \frac{K \cdot \max_{j=1,\cdots,K} \|\delta_j(\boldsymbol{x})\|_1 / \|\boldsymbol{x}\|_1 - 1}{K-1} \in [0,1]. \tag{13}$$

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.

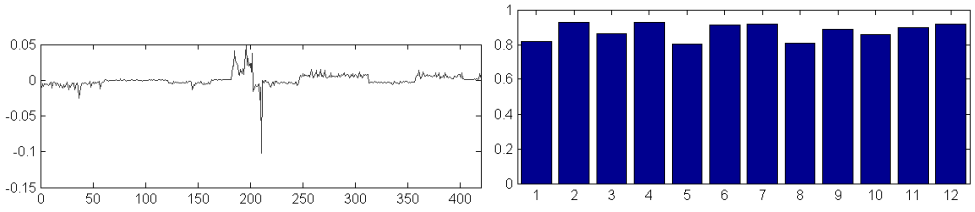


Fig. 8. The ℓ^1 solution and corresponding residuals of an outlying sample on the waist node. $\text{SCI}(\boldsymbol{x}) = 0.13$.

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

[4] A segmentation candidate should be ignored if it overlaps with a previously detected result.

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

$$\tilde{\boldsymbol{y}}' = (\tilde{\boldsymbol{y}}_1^T, \cdots, \tilde{\boldsymbol{y}}_{L'}^T)^T \in \mathbb{R}^{10L'}. \tag{14}$$

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

$$\tilde{\boldsymbol{y}}' = \begin{pmatrix} R_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & R_{L'} & \cdots & 0 \end{pmatrix} A\boldsymbol{x} = R'A\boldsymbol{x} = \tilde{A}'\boldsymbol{x}, \tag{15}$$

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 two linear systems (7) and (8) then become special cases of (15), where $L' = 1$ and L, respectively.

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 (15), 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.[5]

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 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.

Finally, we consider how the change of active nodes affects the estimation of $\boldsymbol{x}$ and the classification of the actions. In compressed sensing, the efficacy of ℓ^1-minimization in solving for the sparsest solution $\boldsymbol{x}$ in (15) is characterized by the ℓ^0/ℓ^1 equivalence relation [7], [8]. 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 (15) 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 V that indeed reducing the available motion sensors will reduce the discriminant power of the action features in a lower-dimensional space.

In summary, the formulation of adaptive global classification (15) via a global projection matrix R' compares favorably to other classical methods such as *kNN* and *decision trees* mainly for the following two reasons: 1. The framework provides a simple means to reject outliers via two sparsity constraints τ_1 and τ_2. 2. The effects of changing action features can be quantitatively studied via R' and its ℓ^0/ℓ^1 equivalence.

V. Experiment

We test 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 (e.g., Fig 3). In total there are 626 actions performed in the data set (see Table III for the numbers in individual classes).

We demonstrate the distributed recognition algorithm against three criteria: 1. What is the accuracy of the algorithm with all 8 sensors activated, and how well can the global classifier adjust when a certain number of nodes are dropped from the network. 2. Whether a set of heuristically selected parameters $\{\tau_1, \tau_2\}$ can effectively segment valid actions with different available nodes. 3. How much communication can be reduced via each node rejecting local measurement compared to simply streaming all action features to the base station.

Table II shows the accuracy of the algorithm in terms of Precision versus Recall and with different sets of sensor nodes. For all experiments, $\tau_1 = 0.2$ and $\tau_2 = 0.4$. If 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

[5] At time t, if multiple hypotheses pass the rejection threshold τ_2, one may heuristically select one based on his/her preference for longer or shorter segments, or other heuristics such as the number of active sensors.

unavailable to the global classifier. Our results show that if we can maintain one motion sensor for the upper body (e.g., at position 2) and one for 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%. Please refer to the Appendix for the rendering of the segmentation results on the motion sequences.

TABLE II
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, particularly compared to the results in Table I. 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 can further compare the difference using two confusion tables III and IV. 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 II. 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 III
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

TABLE IV
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

VI. CONCLUSION AND DISCUSSION

Inspired by the emerging compressed sensing theory, we have proposed a distributed recognition framework to segment and 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 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 is that the wearable sensors need to be firmly fastened at the designated locations. However, a more practical system/algorithm should tolerate certain degrees of offsets 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 *Proceedings of the International Conference on Information Processing in Sensor Networks*, 2007.

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

[3] 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.

[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 Transactions on Information Theory*, 52(12):5406–5425, 2006.

[6] C. Chang and H. Aghajan. Collaborative face orientation detection in wireless image sensor networks. In *Proceedings of Distributed Smart Cameras Workshop*, 2006.

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

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

[9] 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*, pages 107–113, 1999.

[10] 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*, 2003.

[11] T. Huynh and B. Schiele. Analyzing features for activity recognition. In *Proceedings of the Joint Conference on Smart Objects and Ambient Intelligence*, 2005.

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

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

[14] I. Kim, J. Shim, J. Schlessman, and W. Wolf. Remote wireless face recognition employing ZigBee. In *Proceedings of the Distributed Smart Cameras Workshop*, 2006.

[15] A. Klausner, A. Tengg, and B. Rinner. Vehicle classifcation on multi-sensor smart cameras using feature- and decision-fusion. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, 2007.

[16] 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*, 2004.

[17] 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*, 2001.

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

[19] 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.

[20] 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.

[21] 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*, pages 265–278, 2006.

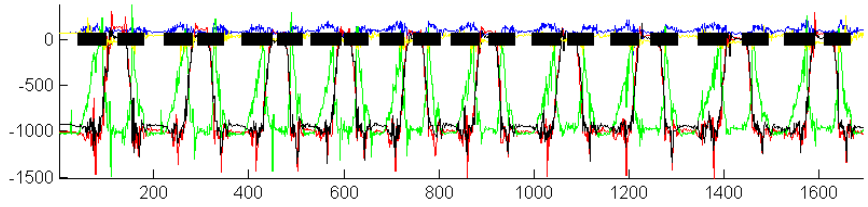
[22] A. Yang, J. Wright, Y. Ma, and S. Sastry. Feature selection in face recognition: A sparse representation perspective. Technical Report UCB/EECS-2007-99, University of California, Berkeley, 2007.

[23] 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*, pages 1431–1434, 2000.

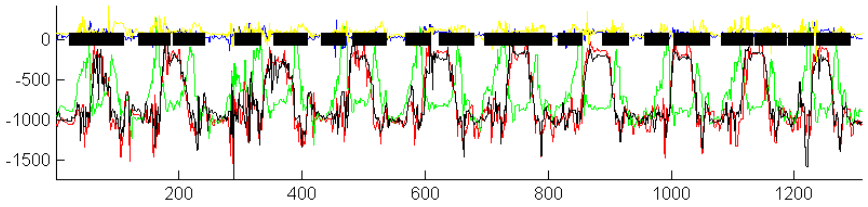
APPENDIX

In this appendix, we provide detailed classification results 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 for three motion sequences performed by the three subjects, respectively. The segmentation results are then superimposed. The black solid boxes indicate the locations of the correctly classified action segments. The red boxes (e.g., in Fig 12 and 13) indicate the locations of the false classification. One can also observe from the figures that some valid actions are not detected by the algorithm, e.g., in Fig 20.

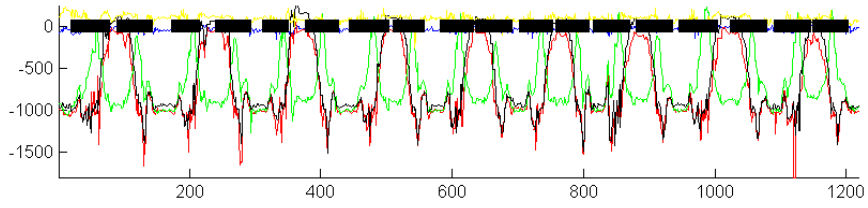
The results clearly demonstrate that the proposed algorithm can accurately segment and classify the 12 action classes with widely different durations. The overall statistics about Precision versus Recall was summarized in Table III.



(a) Subject 1

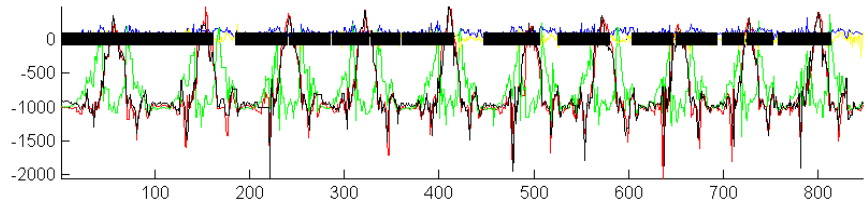


(b) Subject 2

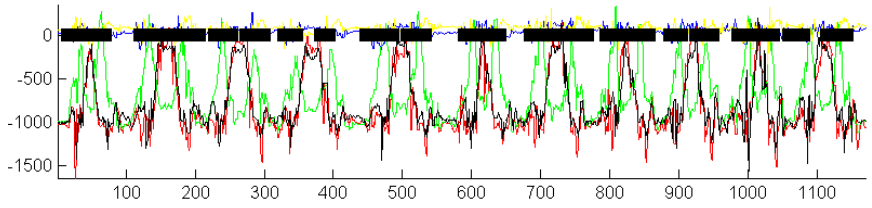


(c) Subject 3

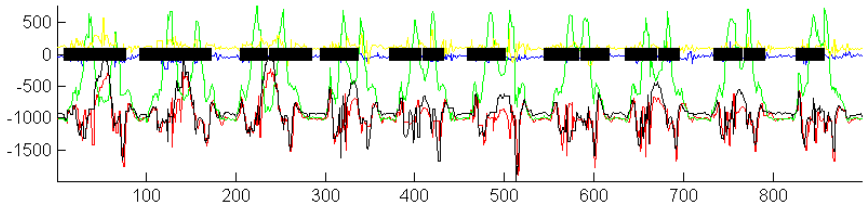
Fig. 9. Segmentation of the slow Stand-Sit-Stand sequences from the three subjects.



(a) Subject 1

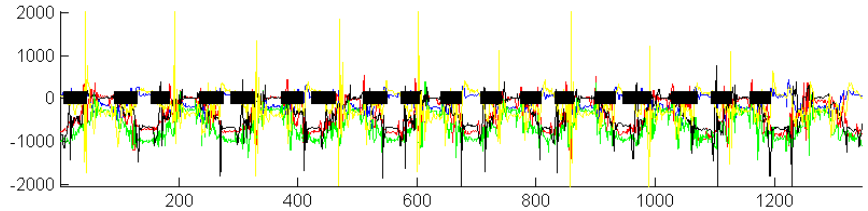


(b) Subject 2

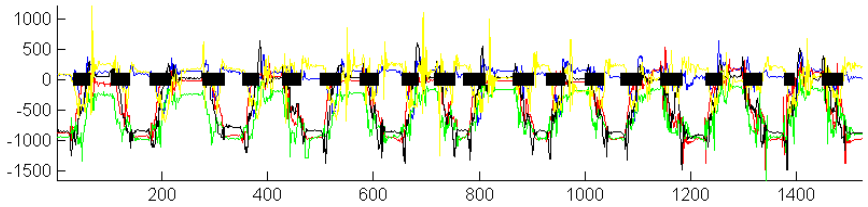


(c) Subject 3

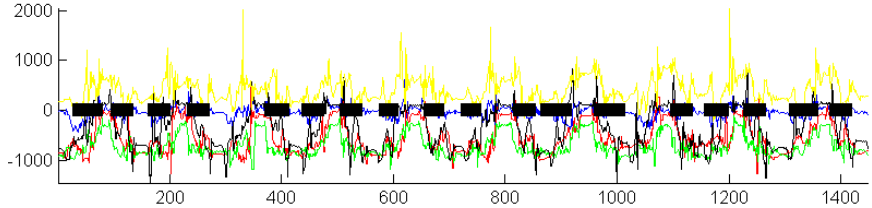
Fig. 10. Segmentation of the fast Stand-Sit-Stand sequences from the three subjects.



(a) Subject 1

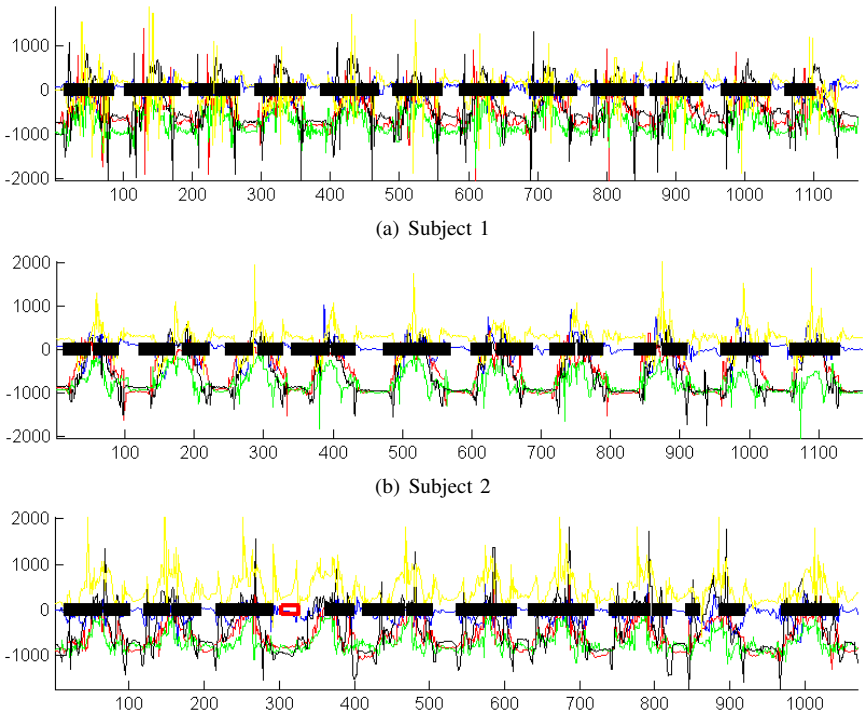


(b) Subject 2



(c) Subject 3

Fig. 11. Segmentation of the slow Sit-Lie-Sit sequences from the three subjects.

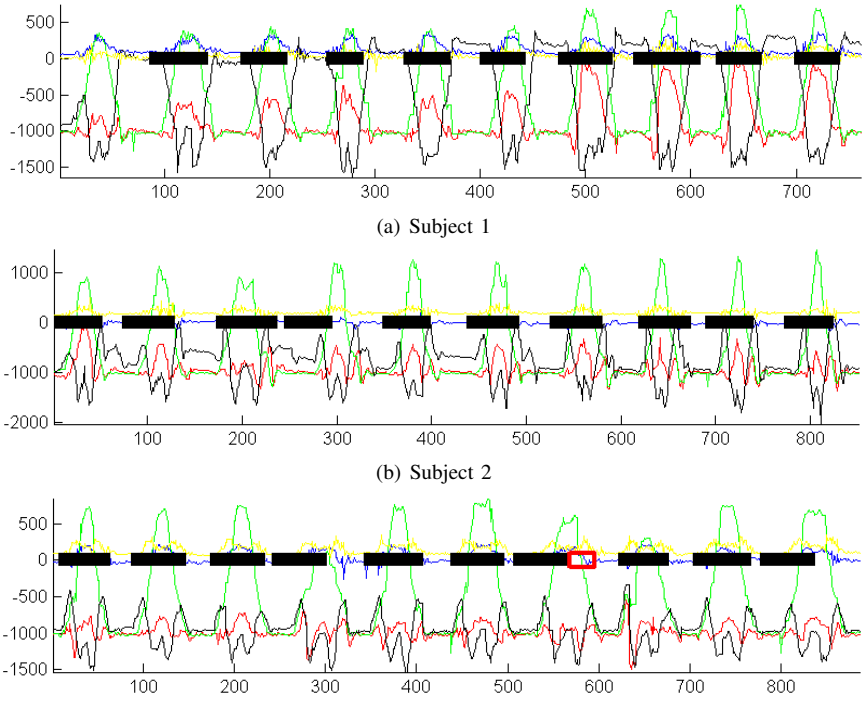


(a) Subject 1

(b) Subject 2

(c) Subject 3

Fig. 12. Segmentation of the fast Sit-Lie-Sit sequences from the three subjects.

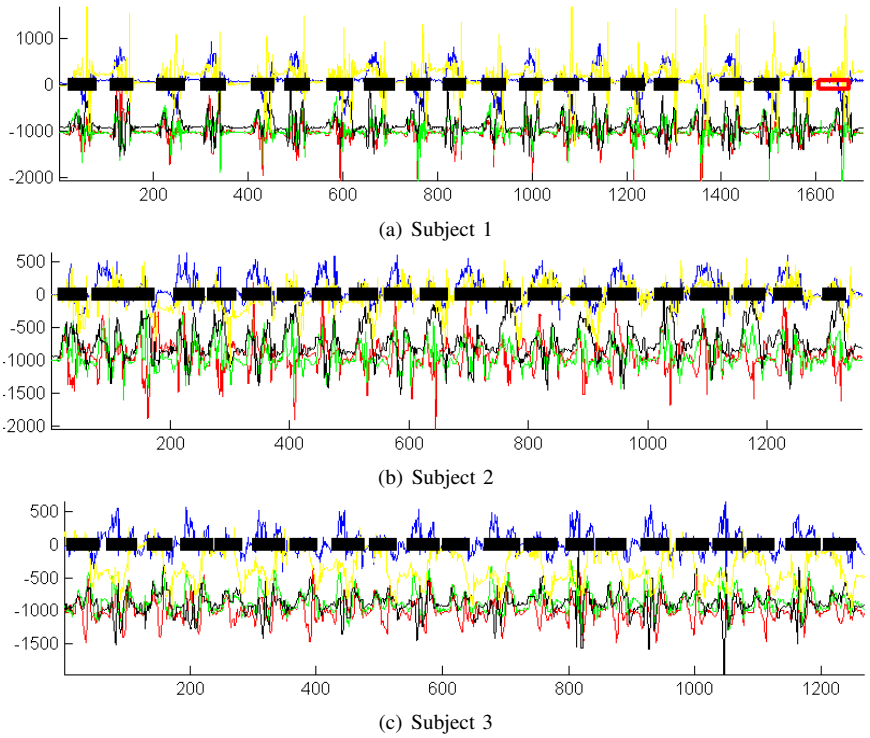


(a) Subject 1

(b) Subject 2

(c) Subject 3

Fig. 13. Segmentation of the Bend sequences from the three subjects.

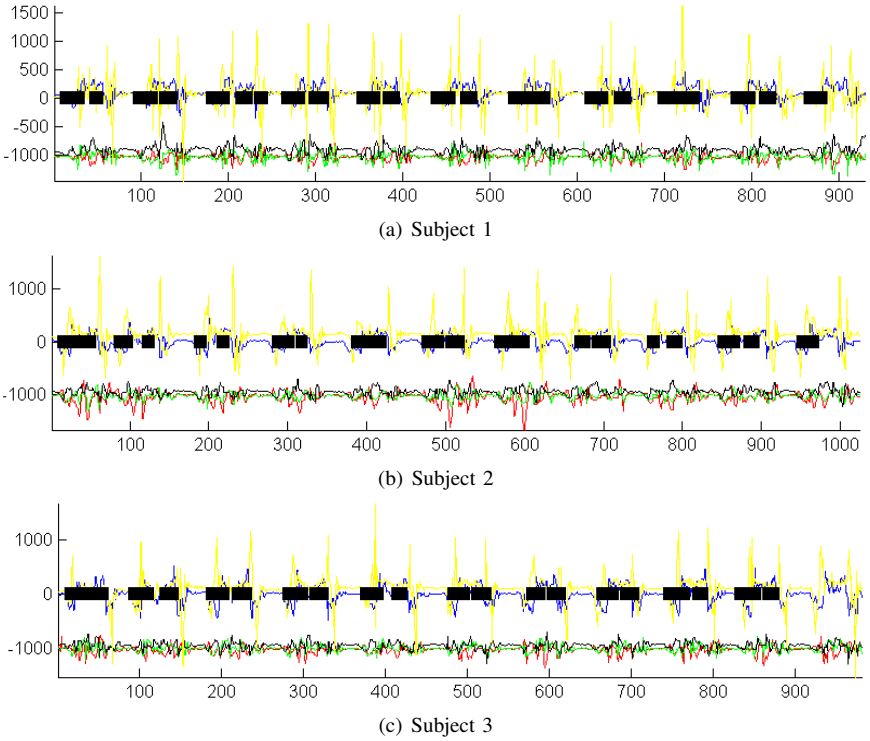


(a) Subject 1

(b) Subject 2

(c) Subject 3

Fig. 14. Segmentation of the Stand-Kneel-Stand sequences from the three subjects.

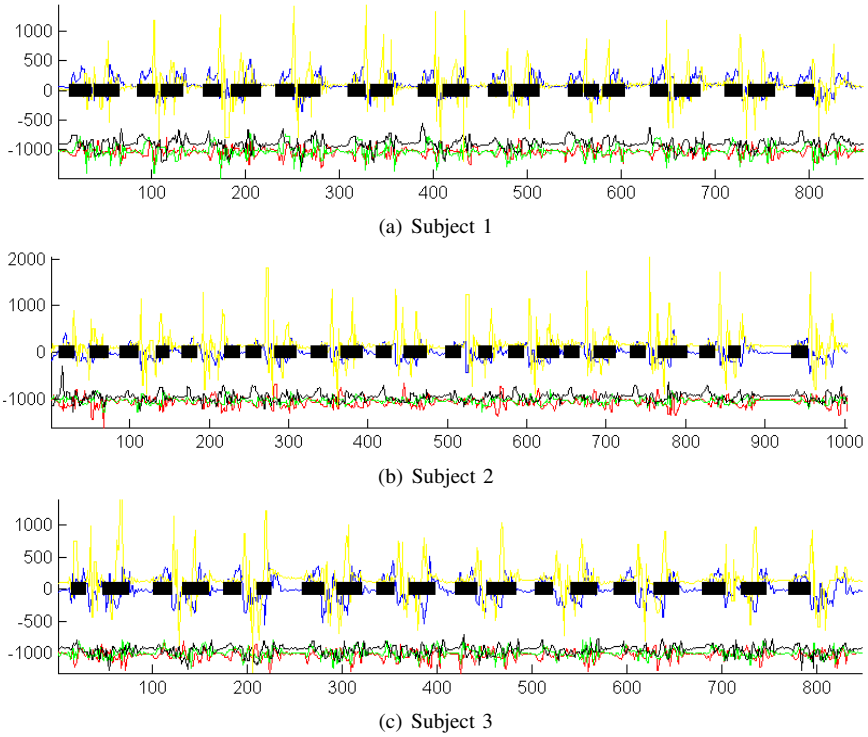


(a) Subject 1

(b) Subject 2

(c) Subject 3

Fig. 15. Segmentation of the 90° Rotate-Right-Left sequences from the three subjects.

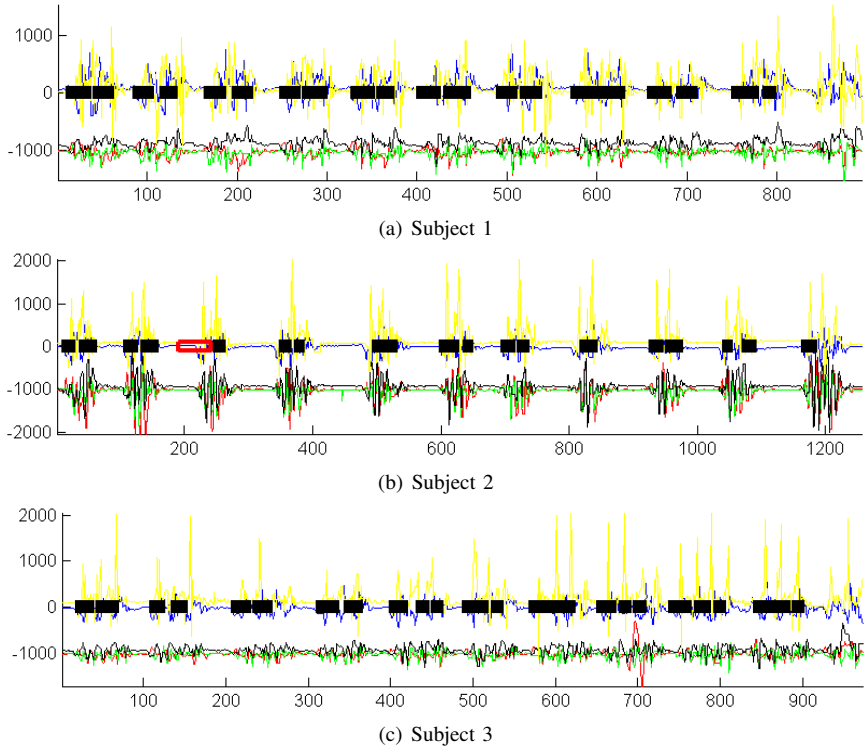


(a) Subject 1

(b) Subject 2

(c) Subject 3

Fig. 16. Segmentation of the 90° Rotate-Left-Right sequences from the three subjects.



(a) Subject 1

(b) Subject 2

(c) Subject 3

Fig. 17. Segmentation of the 180° Rotate-Right sequences from the three subjects.

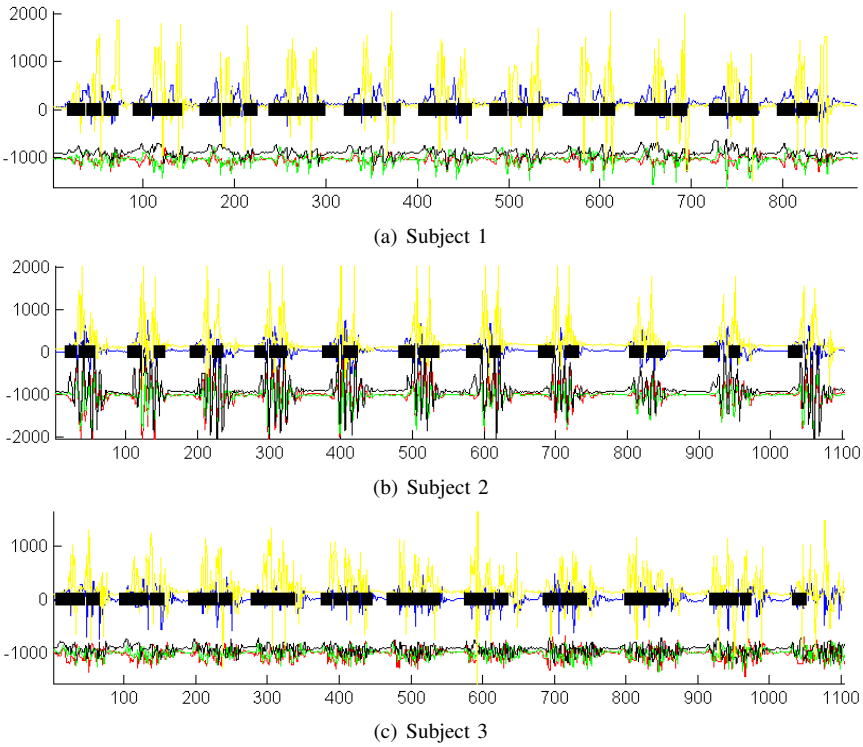


Fig. 18. Segmentation of the 180° Rotate-Left sequences from the three subjects.

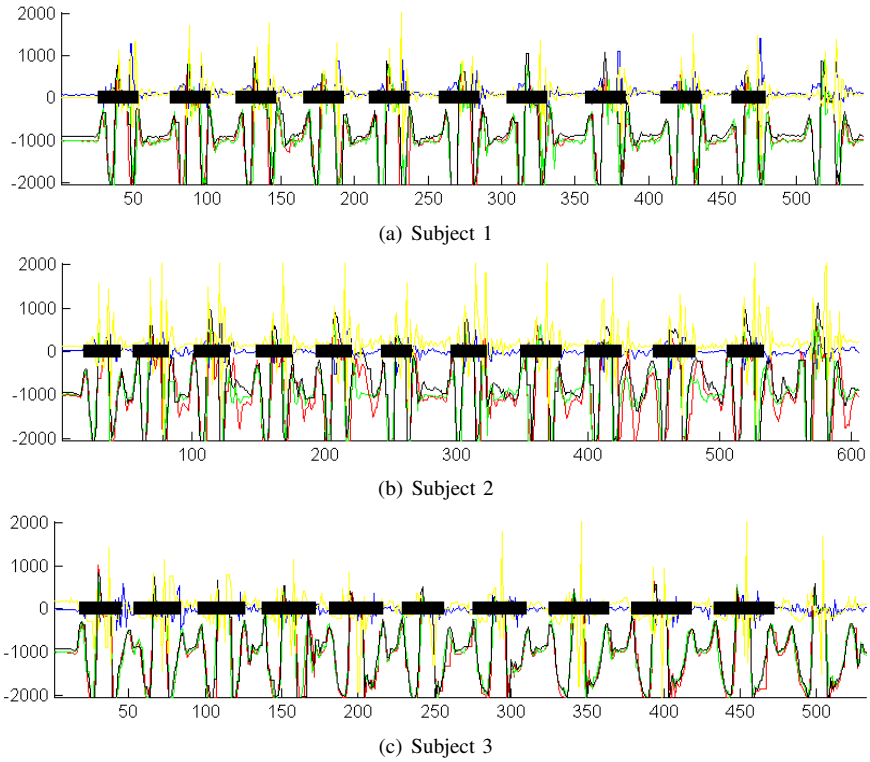


Fig. 19. Segmentation of the Jump sequences from the three subjects.

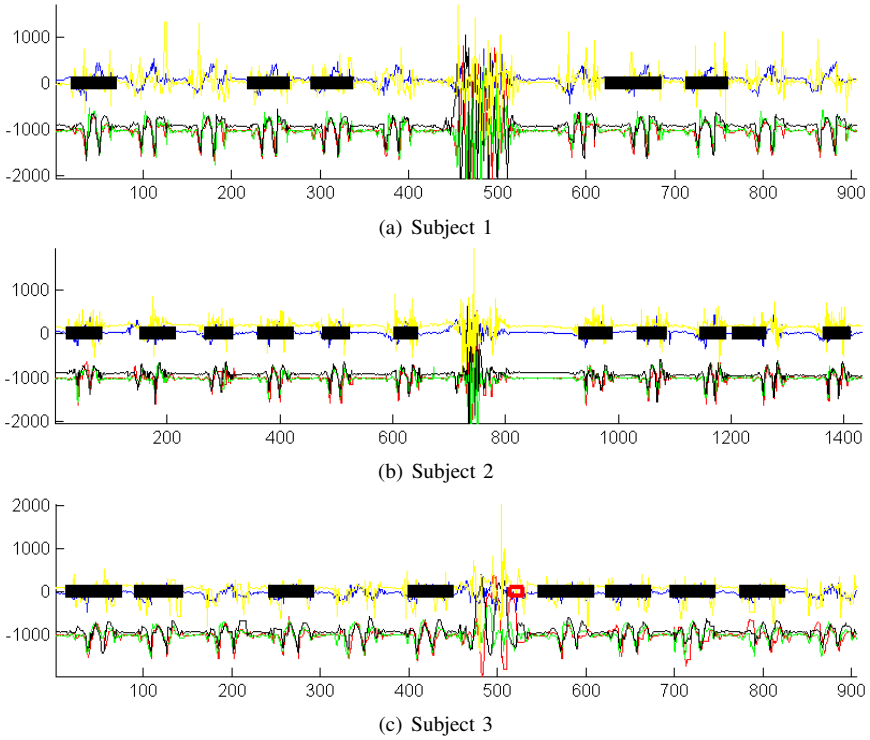


Fig. 20. Segmentation of the Go-Upstairs sequences from the three subjects.

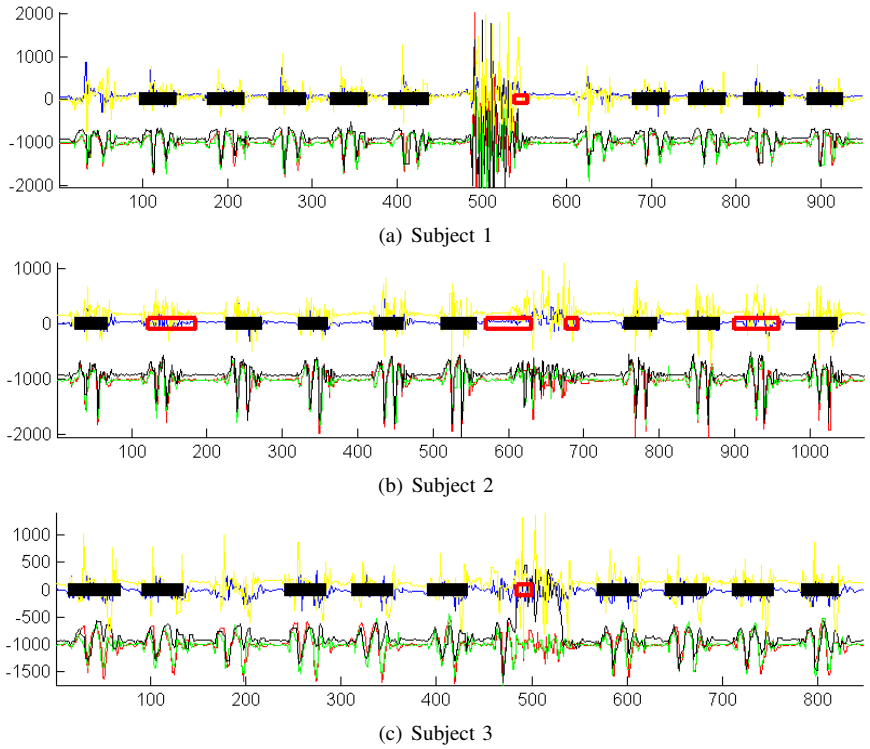


Fig. 21. Segmentation of the Go-Downstairs sequences from the three subjects.