A Phonological Expression for Physical Movement Monitoring in Body Sensor Networks

Hassan Ghasemzadeh, Jaime Barnes, Eric Guenterberg, Roozbeh Jafari
Embedded Systems and Signal Processing Laboratory
Department of Electrical Engineering, University of Texas at Dallas
{h.ghasemzadeh, jaime.barnes, mavpion}@student.utdallas.edu, rjafari@utdallas.edu

Abstract

Monitoring human activities using wearable wireless sensor nodes has the potential to enable many useful applications for everyday situations. The deployment of a compact and computationally efficient grammatical representation of actions reduces the complexities involved in the detection and recognition of human behaviors in a distributed system. In this paper, we introduce a road map to a linguistic framework for the symbolic representation of inertial information for physical movement monitoring. Our method for creating phonetic descriptions consists of constructing primitives across the network and assigning certain primitives to each movement. Our technique exploits the notion of a decision tree to identify atomic actions corresponding to every given movement. We pose an optimization problem for the fast identification of primitives. We then prove that this problem is NP-Complete and provide a fast greedy algorithm to approximate the solution. Finally, we demonstrate the effectiveness of our phonetic model on data collected from three subjects.

1. Introduction

Wireless sensor networks are emerging as a promising platform for a large number of application domains. Applications range from monitoring systems such as environmental and medical monitoring to detection and supervision applications such as acoustic beamforming and military surveillance. In most applications, a sensor node is expected to acquire physical measurements, perform local processing and storage, and communicate within a short distance. Employing sensor networks to understand human actions is expected to be of use in numerous aspects of everyday life. In particular, sensor networks can be effective for rehabilitation, sports medicine, geriatric care, and gait analysis.

Human motion recognition using wireless sensor networks can be done with either off-body or on-body sensor devices. The fields of computer vision and surveillance have traditionally used cameras, off-body devices, to monitor human movement [1]. Processing image sequences to recognize motions is the foundation of this approach. Cameras are used in tracking-based systems to detect actions performed by subjects inside an area. The use of video streams in wireless networks to interpret human motion has been considered for assisted living applications [2-4]. Body sensor networks (BSNs), in contrast, are built of lightweight sensor platforms, which are used to recognize the actions performed by the person wearing them [5]. The sensors can be mounted on the human body or clothing or even woven into the fabric of the clothing itself. Unlike vision-based platforms, BSNs require no environmental infrastructures. They are also less expensive. Moreover, the signal readings from on-body sensors are clearer and unbiased by environmental effects like light and the background. This makes BSNs potentially more accurate than vision frameworks for motion recognition.

We address the problem of movement representation in BSNs. Current methodologies in BSNs map all sensor readings to an identical feature space and then use traditional pattern recognition techniques to detect movement [6, 7]. A more efficient form of representation is a linguistic framework. Motivation for such an approach comes from the fact that movement and spoken language use a similar cognitive substrate in terms of grammatical hierarchy [8]. To construct a language for movement, the first step is to find the basic primitives, called phonemes, and assign appropriate symbols to them. This step, called phonology, creates the foundation for following steps. The second step, morphology, consists of combining primitives to form higher-level movements. These are called the morphemes of motion language. The last step is syntax. It addresses the construction of sentences composed of morphemes using predefined rules and enables the most meaningful level of movement recognition [9].

In general, modeling human movements as a language gives a compact representation of activities. Therefore, a linguistic model would be especially useful in BSNs, which have limited communication and computation capabilities but must collect and process data from several sensor nodes to recognize movements. The linguistic model also enables the extraction of temporal characteristics of motion. However, before one can apply a linguistic approach to movement recognition, signals must be segmented into different movements and labeled. If the signal is segmented in a less meaningful way, such as by using a fixed-size window, movements cannot be decomposed into primitives, and low-level actions cannot be combined to form higher-level actions.

In this paper, we investigate a compact and flexible representation for atomic movements using a BSN platform. We provide phonological rules for constructing primitives associated with each movement. We also provide insights into the detection of movements using primitives. In our framework, each action is described by a set of primitives in which each primitive is associated with a different node.

2. Related Work

Understanding human activity has attracted many researchers in different fields of study. In the computer vision community, both static and dynamic vision-based approaches have been developed. In static methods, individual time frames of a video sequence are used as the basic components for analysis. Recognition then involves the combination of discrete information resulting from individual frames. In dynamic methods, a fixed length of a video stream is the major unit of analysis. The Hidden Markov Model (HMM) [10], which takes into account the correlation between adjacent time instances by formulating a Markov process [11], is often used for the dynamic representation of motion due to its ability to handle uncertainty in its stochastic framework. Examples include the work presented in [12], which introduces a statistical technique for synthesizing walking patterns. The motion is expressed by a sequence of primitives extracted using a model based on HMM.

The limitation of HMM in efficiently handling several independent processes has led to the creation of grammar-based representations of human actions. In [8], Guerra-Filho et al. propose a platform for a visuo-motor language. They provide the basic kernel for the symbolic manipulation of visual and motor information in a sensory-motor system. The phonology, morphology, and syntax of obtained information are detailed. Their phonological rules are modeled by a finite automaton. For syntax, they present a model for sentence construction in which active joints, posture, human activity, and timing coordination among different joints form the basic components of language. These components correspond respectively to noun, adjective, verb and adverb in a natural language. Reng et al. [13] present an algorithm to find primitives for human gestures using electromagnetic sensors. The motion capture system measures the 3D position of body parts. The trajectory of motion is considered a gesture, and primitives are constructed based on the density of the training data set. In [14], Jin et al. propose a technique to reduce the semantic gap between raw 3D motion and actual human behavior. The process begins with spatial dimension reduction and then maps the new temporally distributed feature space into the space of motion primitives. They construct motion documentation for 3D human motion clips that can then be used for motion recognition using string matching algorithms. In [15], Abhijit et al. present a probabilistic context-free grammar (PCFG) [16] for action recognition. They create the grammar using a multi-view video sequence, and the resulting grammar is independent of the video perspective. Each action is described by a short sentence of basic key poses, and each pose is represented by a family of images observed from different viewpoints. The system has the ability to extract key poses and actions from multi-camera, multi-person training data and construct a probabilistic grammar. The grammar is used to parse a single-viewpoint video to explore the sequence of actions within the video. More content-based 3D motion detection techniques can be found in [17], [18], [19], and [20].

Recently, human motion recognition using sensor networks has been the center of much work. Lymberopoulous et al. [21] introduce a general linguistic framework for the construction of hierarchical probabilistic context-free grammars in sensor networks. The system uses a series of locations in time from sensor nodes equipped with cameras. At the lowest level, sensor readings are converted to a set of symbols that become the inputs of higher-level grammars. Each level of the system reduces the data that needs to be propagated to the higher layers. Therefore, the transition from lower levels to higher levels represents more macroscopic activities. An application of this platform is presented in [22], where the detection of cooking activity is obtained by parsing actions through a hierarchy of grammar. This hierarchical approach to human motion representation and recognition is also the foundation of work presented in [2], in which the potentials for extending the platform to an assisted-living environment are discussed. Barnes et al. report that BSNs are a promising platform for locomotion monitoring [23]. In [24], Logan et al. report the results of a study on activity recognition using different types of sensory devices, including built-in wired sensors, RFID tags, and wireless motion sensors. The analysis performed on 104 hours of data collected from more that 900 sensor inputs shows that motion sensors outperform the other sensors on many of the movements studied. A system called *CareNet* for physical activity monitoring using a wireless sensor network is presented by Jiang et al. in [25]. Commercial systems for activity recognition at home, such as [26] and [27], are available, but they work only with certain movements. In [28], Marin-Perianu et al. introduce a correlation algorithm for dynamically grouping sensor nodes equipped with motion sensors. Nodes are considered to be in the same group if their movements correlate for a certain amount of time.

Although the aforementioned linguistic methods have been successful in providing a grammatical platform for action recognition, they could not be efficiently employed in systems based on BSNs. The main reason for this incompatibility is that they use visual movements captured by cameras, whereas BSNs use inertial sensors to capture motion. The primitives constructed based on visual data are not appropriate for inertial data.

3. System Architecture and Signal Processing Flow

For our pilot application, we use a BSN consisting of several sensor units in a wireless network. Each node, which is also called a mote, has a triaxial accelerometer, a biaxial gyroscope, a microcontroller, and a radio, as shown in Figure 1. The processing unit of each node samples sensor readings at 22 Hz and transmits the data wirelessly to a base station using a TDMA protocol. The sampling rate is experimentally chosen to provide sufficient resolution of human motion data while compensating for bandwidth constraints on our sensor platform. Our motes, Tmote Sky motes, are commercially available from moteiv® and are powered by two AA batteries each. The sensor board is custom designed. The base station is another mote connected to a laptop by USB. For our experiments, we arranged 18 sensor nodes on the subjects, as shown in Figure 2. We have used 18 nodes to ensure that the system captures inertial information associated with all major parts of the body and to study which locations are most useful. Fewer nodes are sufficient to accurately recognize motion, as the results of this paper will show.

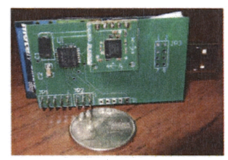

Figure 1. A node with inertial sensors

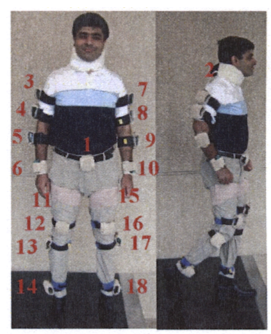

Figure 2. Experimental subject wearing sensor nodes.

The signal processing, phoneme construction, and identification process has six steps, as shown in Figure 3.

1. *Sensor Data Collection:* Data is collected from each of the five sensors on each of the 18 sensor nodes at 22 Hz.
2. *Preprocessing:* The data is filtered to enable easier processing. We use an eight-point moving average.
3. *Segmentation:* We determine the part of the signal that represents an action. Currently we perform this process manually to avoid injecting errors from automatic segmentation into our process.
4. *Feature Extraction:* Single-value features are extracted from the filtered data. Features include mean, start-to-end amplitude, standard deviation, peak-to-peak amplitude, and RMS power.

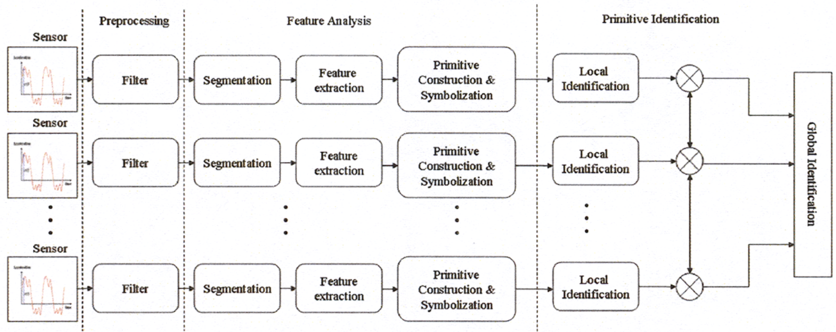

Figure 3. Signal processing flow.

5. *Primitive Construction and Symbolization*: This is the main focus of this paper. In this stage, we use a phonological approach to construct primitives for each sensor node and present human action in terms of meaningful phonemes.

6. *Per-Node Identification*: Each node uses a phonetic expression of movements to map a given action to the corresponding primitive. In this way, each node offers local knowledge of the current event in the system.

7. *Global Identification*: The global state of the system is recognized by combining local knowledge from different nodes.

Currently, we process collected data offline in MATLAB. This is convenient for rapid prototyping and algorithm development. However, the algorithms we use for signal processing and movement recognition are developed from relatively simple techniques that can be implemented and executed on the motes.

4. Preliminaries

In this section, we first investigate properties of motion recognition using our system. We then discuss the need for clustering techniques in our system. We look into the most popular clustering algorithms and the mechanisms for evaluation of clusters. Finally, we apply clustering to the problem of primitive construction.

4.1. Recognition Objectives

While detection of movements requires a global view of the whole system, each individual node in a BSN has only local knowledge of the event taking place. This makes the problem of training and recognition very challenging [21]. The amount of knowledge provided by each node determines the usefulness of that node in recognizing movements. We believe the right place to begin a discussion on phoneme construction for our system is to consider the properties of movements and the relationship between each movement and the system configuration. To achieve a generalized view of these properties, we use a per-node study; i.e., we do not make any assumptions about network topology or applications to avoid ad hoc methods. We summarize the properties of our system below.

4.1.1. Location independence: Our study is focused on using inertial sensors to measure attributes of human motions. Therefore, we do not embed any context data within the system. In particular, the training and recognition of movements is independent of the location where the action takes place. However, contextual information such as location may help to improve the accuracy of detection at a later stage. Context awareness can be implemented on top of the proposed system.

4.1.2. Distributed recognition: Each node in the system has a different perspective on the movement. The ability of a node to recognize movements varies based on the type of movement. For example, consider the two movements "stand to sit" and "bending". A node mounted on the arm might contribute to distinguishing these two movements, but a sensor on the ankle might not provide useful information. The main advantage of unsupervised classification in our system is that by clustering training sets at each node, nodes not contributing to a certain movement will be recognized. That is, we will be able to determine which nodes are useful for recognizing which movements.

4.1.3. Recognition invariance: The detection of a movement should be invariant with respect to the target population. This includes invariance to age, gender, strength, and physical capabilities. At this stage of our study, we have not investigated this property as investigating it will require data from a large and diverse set of subjects.

4.2. Clustering Implications

We employ clustering to find primitives for each movement. In this section, we explain how we benefit from clustering algorithms and strategies of finding the most effective clustering configuration.

4.2.1. Clustering Techniques. Clustering is grouping together those data points in a training set that are most similar to each other. Two major clustering techniques are hierarchical clustering [29] and *K*-means clustering [30]. In the hierarchical method, each data item is initially considered a single cluster. At each stage of the algorithm, similar clusters are grouped together to form new clusters. In the *K*-means algorithm, the idea is to define *K* centroids, one for each cluster. In this way, training data are grouped into a predefined number of clusters

Unlike hierarchical clustering, in which clusters are not improved after being created, the *K*-means algorithm visits constructed clusters at each iteration and improves them gradually. The continuously improving nature of this algorithm leads to high-quality clusters when provided appropriate data. We use this algorithm for our analysis because it is simple, straightforward, and is based on the firm foundation of analysis of variances [31].

4.2.2. Cluster Validation. Although *K*-means is a popular clustering technique, the partition attained by this algorithm is dependent on both the initial value of centroids and the number of clusters. To increase the likelihood of arriving at a good partitioning of the data, many improvements to *K*-means have been proposed in the literature. The sum of square error (SSE) is a reasonable metric used to find the global optimal solution [32]. To cope with the effects of initialization, we use uniformly distributed initial centers [33] and repeatedly search for the configuration that gives the minimum error. We calculate the SSE error function as in (1), where x_i denotes the i^{th} data item, μ_k denotes the centroid vector associated with cluster C_k, and K is the total number of clusters.

$$SSE = \sum_{k=1}^{K} \sum_{i \in C_k} (x_i - \mu_k)^2 \tag{1}$$

Another problem with *K*-means is that it requires prediction of the correct number of clusters. Usually, a cluster-validity framework provides insight into this problem. We employ the silhouette quality measure [34], which is robust and takes into account both intra-cluster and inter-cluster similarities to determine the quality of a cluster. Let C_k be a cluster constructed based on the *K*-means algorithm. The silhouette measure assigns a quality metric S_i to the i^{th} data item of C_k. This value indicates the confidence of the membership of the i^{th} item to cluster C_k. S_i is defined by (2), where a_i is the average distance between the i^{th} data item and all of the items inside cluster C_k, and b_i is the minimum of the average distances between the i^{th} item and all of the items in each cluster besides C_k. That is, the silhouette measure compares the distance between an item and the other items in its assigned cluster to the distance between that item and the items in the nearest neighboring cluster. The larger the S_i, the higher the level of confidence about the membership of the i^{th} sample in the training set to cluster C_k.

$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \tag{2}$$

While S_i, also called *silhouette width*, describes the quality of the membership of a single data item, the quality measure of a partition, called the *silhouette index,* for a given number of clusters K is calculated using (3), where N is the number of data items in the training set.

$$Sil(K) = \frac{1}{N} \sum_{i=1}^{N} S_i \tag{3}$$

To obtain the most effective configuration in terms of the number of clusters, one can choose the K that has the largest silhouette index, as shown in (4).

$$\hat{K} = \arg\max_{K} \{Sil(K)\} \tag{4}$$

4.3. Phonology

Physical movement monitoring involves mapping a series of behaviors onto a vocabulary of actions. The vocabulary represents movements performed previously and stored as training sets across nodes. This vocabulary consists of words, each composed of segments named phonemes or primitives. The initial step in a linguistic framework for action recognition is phonology, which is the process of finding basic primitives for human movement. In our networked framework, primitives are distributed among sensor nodes with varying ability to recognize movements. Our phonetic description is able to characterize each action in terms of primitives using a two-stage process that we will describe next.

4.3.1. Primitive Construction. The first step in constructing our phonetic description is to find local primitives at each node. To do this, we use *K*-means clustering to perform local clustering at each node, transforming the feature space into groups of dense data items. Each cluster corresponds to an original primitive in our model. This technique is effective since it provides insights into the usefulness of nodes in detecting each movement. Actions with similar patterns at a certain node tend to be assigned to the same cluster at that node, while they might be represented by different clusters at another node. Suppose sensor readings are mapped to m different actions $\{A_{i1}, A_{i2}, \ldots, A_{im}\}$ on some node i by segmentation. The clustering algorithm will transform these actions into a series of clusters $\{P_{i1}, P_{i2}, \ldots, P_{ic}\}$, where $c \leq m$ represents the number of clusters. We employ the validation techniques explained in Section 4.2.2 to find the most effective clustering configuration.

4.3.2. Symbolization. The second step in constructing our phonetic description is to select a final group of primitives and assign symbols to them. Some of the clusters defining our initial primitives are of low quality, meaning the primitives they define will not be good representations of our movements. We refine our clusters by calculating the silhouette quality measure for each cluster and eliminating clusters that do not meet a certain threshold. In this way, the set of primitives at node i might be reduced to $\{P_{i1}, P_{i2}, \ldots, P_{ik}\}$, where $k \leq c$ is the number of final primitives after applying the quality measure and c is the number of original primitives. After cluster refinement, each movement is represented by a set of final primitives at each node.

5. Movement Identification Problem

Physical movement monitoring by sensor networks requires the combination of local knowledge to achieve a global view of human behaviors. In this section we study the problem of action recognition using the semantic subspace generated by primitives. We pose this problem from a general detection perspective. That is, we do not make any assumptions about the communication model and signal processing techniques used for classification. Compared to a traditional classification approach, our movement recognition algorithm is faster and requires less communication and power. We use the notations in Table 1 throughout this section.

5.1. Decision Tree Representation

The problem of recognizing movements using primitives can be viewed as a decision tree problem in which internal decision nodes represent different sensor nodes, and terminal leaves correspond to movements. The input to the tree is a vector $\lambda = [\lambda_1, \ldots, \lambda_n]$, where λ_i is the primitive from node i that describes the movement to be identified, and n is the number of nodes in the system. The aim of motion recognition is to assign λ to one of m mutually exclusive actions. The ordering of nodes in the tree changes its height and thus the time needed to converge to a solution. We seek a linear ordering of the nodes that minimizes convergence time. A solution to this problem could potentially provide insights into several crucial problems, including decision tree construction, distributed classification, and scheduling.

Table 1. Notations

Symbol	Description
A	set of all actions to be detected
P	set of primitives extracted across the network
S	set of sensor nodes
M	number of actions
p	number of final primitives
n	number of sensor nodes
k	index for an action
j	index for a primitive
i	index for a node

To better illustrate the identification problem, we provide a simple example, shown in Figure 4, which depicts the mapping of actions to primitives. The system consists of three sensor nodes denoted by s_1, s_2, and s_3, and four movements denoted by A, B, C, and D. The circles depict the mapping of movements to primitives rather than the actual distribution of data. The system has seven final primitives denoted by $\{P_1, P_2, \ldots, P_7\}$. In node s_1, movement A is mapped to primitive P_1, movements B and C are mapped to primitive P_2, and movement D is mapped to primitive P_3. In other nodes, the movements are mapped to primitives as shown. This phonetic expression can effectively describe the ability of individual nodes to identify the movements. For instance, node s_1 can distinguish movement A from the rest of the movements, as it finds no ambiguity when mapping an action to P_1, but it cannot distinguish between movements B and C, as they are mapped to the same primitive. While each node has limited knowledge of the system, we require a global view in which every movement is distinguished from the rest. Furthermore, we require an ordering of sensor nodes that minimizes the total time of convergence.

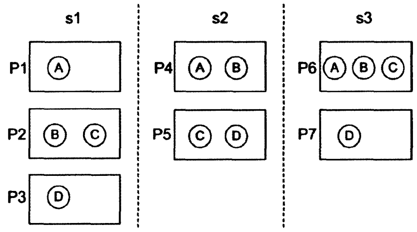

Figure 4. An example of three nodes (s_1, s_2, s_3) and four movements (A, B, C, D). The movements are mapped to seven primitives ($P_1, \ldots, P_7$). Each movement is symbolized by corresponding primitives; $A=\{P_1,P_4,P_6\}$; $B=\{P_2,P_4,P_6\}$; $C=\{P_2,P_5,P_6\}$; $D=\{P_3,P_5,P_7\}$.

Given an instance of a decision problem, one can construct different decision trees. Figure 5 illustrates a sample decision tree for the example represented in Figure 4. The problem of finding a minimal decision tree is shown to be hard to approximate [35]. Therefore, in this paper, we investigate a static design decision for constructing a decision tree for motion recognition. That is, we try to make an offline decision about the optimal ordering of the sensor nodes for recognition. Our current method of linearly ordering the nodes restricts the shape of the decision tree so that all nodes are placed on a single path from the root and the tree has a height equal to the total number of nodes required for recognition. We will investigate the construction of a full decision tree in future.

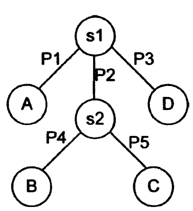

Figure 5. A sample decision tree for the example illustrated in Figure 4.

5.2. Problem Formulation

In this section, we present a formal definition of our movement identification problem.

Definition (Local Discrimination Set): Let $A=\{A_1, A_2, \ldots, A_m\}$ be a finite set of movements mapped to a set of primitives $P=\{P_1, P_2, \ldots, P_p\}$. The local discrimination set LDS_i for a node s_i is defined by:

$$LDS_i = \{(A_k, A_{k'}) \mid A_k \in P_j \ , \ A_{k'} \in P_{j'} \ , \ P_j \in s_i \ , \ P_{j'} \in s_i \ , \ j \neq j'\} \quad (5)$$

The LDS_i expresses the pairs of actions that can be distinguished by the i^{th} node. In the example shown in Figure 4, the set of movements is $\{A, B, C, D\}$, and the set of primitives is $\{P_1, P_2, \ldots, P_7\}$. Therefore, the local discrimination set for node s_1 is $LDS_1=\{(A,B), (A,C), (A,D), (B,D), (C,D)\}$. For the other nodes, we have $LDS_2=\{(A,C), (A,D), (B,C), (B,D)\}$ and $LDS_3=\{(A,D), (B,D), (C,D)\}$.

Definition (Global Discrimination Set): Let $A=\{A_1, A_2, \ldots, A_m\}$ be a finite set of actions and $P=\{P_1, P_2, \ldots, P_p\}$ a collection of primitives. The global discrimination set GDS is defined by:

$$GDS = \{(A_k, A_{k'}) \mid A_k \in A; A_{k'} \in A; \ k \neq k'\} \quad (6)$$

The global discrimination set contains all the pairs of movements that are required to be distinguished from one another. For example, the global discrimination set for the system shown in Figure 4 is $GDS=\{(A,B), (A,C), (A,D), (B,C), (B,D), (C,D)\}$. In this example, the objective is to distinguish between every pair of movements.

Definition (Complete Ordering): Let $A=\{A_1, A_2, \ldots, A_m\}$ be a finite set of actions and $P=\{P_1, P_2, \ldots, P_p\}$ a collection of primitives. An ordering $O=\{s_1, s_2, \ldots, s_n\}$ is complete if the following condition holds.

$$\bigcup_{i=1}^{n'} LDS_i = GDS \quad (7)$$

This indicates that the ordering is capable of distinguishing between all required pairs of movements. In the previous example, the ordering $O=\{s_1, s_2\}$ is complete since $\mathrm{LDS}_1 \cup \mathrm{LDS}_2 = \mathrm{GDS}$, but the ordering $O=\{s_2, s_3\}$ is not complete because this ordering cannot discriminate between movements A and B.

Definition (Ordering Cost): Let $O=\{s_1, s_2, \ldots, s_n\}$ be a complete ordering of sensor nodes and $f(\mathrm{A_k})$ a function that gives the index of the first node in which the following condition holds:

$$\{(A_k, A_{k'}) \mid k \neq k' \ , \ A_{k'} \in A\} \subset \bigcup_{i=1}^{f(A_k)} LDS_i \quad (8)$$

That is, $f(\mathrm{A_k})$ is the number of nodes necessary to distinguish movement A_k from all other movements. Then the total cost of the ordering is given by the following equation:

$$Z = \sum_{A_k \in A} f(A_k) \quad (9)$$

This formulation weights the cost of an ordering so that an ordering in which more movements require fewer nodes has a lower cost. For example, let $O=\{s_3, s_2, s_1\}$ be a complete ordering for the example shown in Figure 4. Then $f(D)=1$ because movement D can be completely identified at the first visited node (s_3). At the next node (s_2), movement C can be distinguished from the remaining movements (A and B). Thus, $f(C)=2$ because movement C is identified at the second node. Finally, movements A and B will be detected at the third visited node (s_1) meaning that $f(A)=f(B)=3$. Therefore, the total cost for this ordering is 9.

Definition (Min Cost Identification Problem): Given a finite set *GDS* and *LDS*, where $LDS=\{LDS_1, LDS_2, ..., LDS_n\}$ is a collection of subsets of *GDS* such that the union of all LDS_i forms *GDS*, Min Cost Identification (MCI) is the problem of finding a complete linear ordering such that the cost of the ordering is minimized.

In the above example, it would be easy to find the optimal solution by a brute-force technique. We can see that the cost for the optimal ordering ($\{s_1, s_2\}$) is 6.

5.3. Problem Complexity

In this section we address the complexity of Min Cost Identification. We show that this problem is NP-Complete by reduction from Min Sum Set Cover. Therefore no polynomial time algorithm exists that solves it unless P=NP.

Definition (Min Sum Set Cover): Let U be a finite set of elements and $S=\{S_1, S_2, ..., S_m\}$ a collection of subsets of U such that their union forms U. A linear ordering of S is a bijection f from S to $\{1, 2, ..., m\}$. For each element $e \in U$ and linear ordering f, we define $f(e)$ as the minimum of $f(S)$ over all $\{S_i : e \in S_i\}$. The goal is to find a linear ordering that minimizes $\sum_e f(e)$.

Theorem 1. The Min Cost Identification problem is NP-Complete.

Proof. We will prove that the Min Cost Identification problem is NP-Complete by reduction from Min Sum Set Cover (MSSC). Consider an MSSC instance (U,S) consisting of a finite set of elements U and a collection S of subsets of U. The objective is to find a minimum-cost linear ordering of subsets such that the union of the chosen subsets of U contains all elements in U. We now define a set $\tilde{U}$ by replacing elements of U with all elements $(A_k, A_{k'})$ from the *GDS*. We also define $\tilde{S}$ by replacing its subsets S_i with LDS_i. $(\tilde{U},\tilde{S})$ is an instance of the MCI problem. Therefore, MCI is NP-hard. Since solutions for the decision problem of MCI are verifiable in polynomial time, it is in NP, and consequently, the MCI decision problem is also NP-Complete.

Theorem 2. There exists no polynomial-time approximation algorithm for MCI with an approximation ratio less than 4.

Proof. The reduction from MSSC to MCI in the proof of Theorem 1 is approximation preserving; that is, it implies that any lower bound for MSSC also holds for MCI. In [36], it is shown that for every $\varepsilon > 0$, it is NP-hard to approximate MSSC within a ratio of $4-\varepsilon$. Therefore, 4 is also a lower bound for the approximation ratio of MCI.

5.4. Greedy Solution

The greedy algorithm for MCI is adapted from the greedy algorithm for MSSC and is shown in Algorithm 1. At each step, it looks for the node that can distinguish between the maximum number of remaining movements. It then adds such a node to the solution space and removes the movements it distinguishes from further consideration. The algorithm terminates when all pairs of movements are distinguished from each other. The approximation ratio is 4 as previously discussed.

Algorithm 1. Greedy solution for MCI

```
Inputs: A, P, S
Output: O
Calculate set LDS_i for every node s_i
Calculate set GDS
O = ϕ
while (O ≠ GDS)
    take node s_i s.t. LDS_i is maximum cardinality
    O = O ∪ s_i
    for each (e ∈ LDS_i)
        remove e from all LDS_j for j=1,...,n
    end for
end while
```

Algorithm 1 can be used to find the minimum number and preferred locations of sensor nodes required to recognize certain movements. This can be used for power optimization because at any time, only a subset of sensor nodes will be required to be active, based on the movements of interest at that time. Furthermore, reducing the number of required nodes helps to achieve a more wearable system, which is a critical issue in the design of BSNs.

6. Experimental Results

To validate our proposed linguistic expression and identification framework, we prepared an experiment using the system described in Section 3. We had three subjects, one male with age 32 and two females with ages 22 and 55. We placed 18 sensor nodes on each subject as shown in Figure 2 and stated in Table 3. The subjects performed the movements listed in Table 2 for ten trials each. The experiment was designed to involve a relatively wide range of movements that required motion from different parts of the human body. Although our experiments are carried out in a controlled environment in which subjects are asked to repeatedly perform specific movements, we have the suspicion that our method will be effective in less controlled environments when given a larger training set that includes a wider range of movements.

Table 2. Movements for experimental analysis.

No.	Description
1	Stand to sit (armchair)
2	Sit to stand (armchair)
3	Stand to sit (dining chair)
4	Sit to stand (dining chair)
5	Sit to lie
6	Lie to sit
7	Bend and grasp from ground with right hand
8	Bend and grasp from ground with left hand
9	Bend and grasp from coffee table with right hand
10	Bend and grasp from coffee table with left hand
11	Turn clockwise 90 degrees and return
12	Turn counterclockwise 90 degrees and return
13	Look back clockwise and return to the initial position
14	Look back counterclockwise and return to the initial position
15	Kneeling, right leg first
16	Kneeling, left leg first
17	Move forward one step, right leg
18	Move forward one step, left leg
19	Reach up to a cabinet with right hand
20	Reach up to a cabinet with left hand
21	Reach up to a cabinet with both hands
22	Grasp an object with right hand, turn clockwise and release
23	Grasp an object with two hands, turn clockwise and release
24	Turn clockwise 360 degrees
25	Turn counterclockwise 360 degrees
26	Jumping
27	Going up stairs, right leg first (one stair)
28	Going down stairs, right leg first (one stair)
29	Rising from kneeling, right leg
30	Rising from kneeling, left leg

For each of the five data streams we receive from each sensor node (x, y, z acceleration and x, y angular velocity), we extracted the five features listed in Section 3. We used 50% of the data as a training set and 50% as a test set. The training set was used for constructing primitives and finding the minimum-cost ordering of the nodes, while the test set was used for verifying the accuracy of our classification technique.

Table 3. Experiment Statistics

Node#	Description	#P	Ordering	#Actions	$\sum_{A_k \in A} f(A_k)$
1	Waist	3	-	-	
2	Neck	2	5	2	40
3	Right-arm	4	-	-	
4	Right-elbow	3	1	0	0
5	Right-forearm	3	8	3	131
6	Right-wrist	2	9	3	158
7	Left-arm	3	-	-	
8	Left-elbow	3	-	-	
9	Left-forearm	3	4	3	30
10	Left-wrist	2	-	-	
11	Right-thigh	2	-	-	
12	Right-knee	2	-	-	
13	Right-shin	4	6	3	58
14	Right-ankle	3	10	3	**188**
15	Left-thigh	2	2	0	0
16	Left-knee	2	-	-	
17	Left-shin	4	7	7	107
18	Left-ankle	2	3	6	18

6.1. Static Construction of Decision Tree

As previously stated, our method constructs a decision tree using an ordering of sensor nodes that leads to a near-optimal solution for minimum-cost action recognition. First, the feature space is mapped to the initial phonemes using our primitive-construction approach. Next, the silhouette quality measure is applied to select high-quality primitives from among the phonemes. The number of primitives for each sensor node is shown in Table 3. The number of primitives per node depends on the ability of the node to distinguish between different movements. A node with only one primitive could not distinguish between any movements, whereas a node with as many primitives as there are movements could distinguish between all movements. Our nodes have two to four phonemes each.

To verify the effectiveness of the greedy solution to the Min Cost Identification problem, we used the primitive representation of data collected from 18 nodes. Since there are 30 movements to be recognized from our experiment, the size of the *GDS* is 435, which is the total number of pairs $(A_k, A_{k'})$ to be distinguished.

The ordering obtained by the greedy algorithm is given in Table 3. The ordering implies that in the worst case, 10 sensor nodes are sufficient to achieve a global knowledge of the current event in the system. The most informative node is node 4, which has the ordering 1, meaning that it should be the first node visited for action detection. The value of the cost function associated with this node is zero because it alone cannot completely distinguish any movement from all the others. In other words, collaboration between distributed sensor nodes is required to gain a global view of the system. The remaining nodes that should be visited to recognize all movements in our system are 15, 18, 9, 2, 13, 17, 5, 6, and 14, in that order.

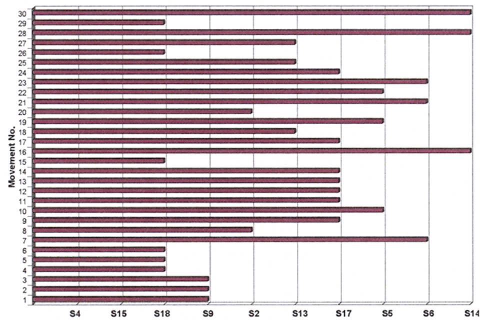

Figure 6. Identification order of movements

The fifth column in Table 3 shows how many movements are distinguished as nodes are visited. At the third visited node (node 18), six movements are identified, and at the fourth visited node (node 9), three movements are identified. The next visited node (node 2), supplies the information necessary to distinguish two additional movements from the rest. The last column in Table 3 shows the accumulated value of the cost function $f(A_k)$. The total cost of the ordering is 188. Figure 6 shows the nodes required to identify each movement using the ordering we obtained. Visited nodes are listed along the *x*-axis, and movements are listed along the *y*-axis.

6.2. Recognition Accuracy

To show the effectiveness of our motion detection algorithm using only the active nodes reported above, we used the linear ordering of nodes we obtained to classify our test set (50% of the collected data). After filtering, segmentation, and feature extraction, each test vector was mapped to its corresponding primitives on the active nodes. Using these primitives and the decision tree defined by the obtained ordering of active nodes, we achieved an accuracy of 91%.

7. Conclusions and Future Work

In this paper, we proposed a phoneme representation of human behavior for body sensor networks. This phonological approach produces a compact expression of atomic actions. We used a cluster-based technique to generate basic primitives for each sensor node. The global state of the system is determined using the local knowledge given by primitives at each sensor node and a fast recognition policy. We showed that the problem of determining the optimal ordering of nodes for movement monitoring at the primitive level in our distributed platform is NP-Complete and presented a fast greedy solution for this problem. Our experiments demonstrated the functionality of the proposed framework. We verified that our technique achieves 91% accuracy in classifying human actions. Although our linguistic framework is simplified by constructing and using only primitives, it demonstrates the potential of a linguistic approach. In the future, we plan to investigate the construction of a full decision tree for dynamic optimization. We also plan to investigate the recognition of human behavior by combining primitives to form words that describe more complex activities and combining words to form sentences that describe general behavior.

8. References

[1] H. Weiming, T. Tieniu, W. Liang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 34, pp. 334-352, 2004.

[2] T. Teixeira, D. Lymberopoulos, E. Culurciello, Y. Aloimonos, and A. Savvides, "A Lightweight Camera Sensor Network Operating on Symbolic Information," in *First Workshop on Distributed Smart Cameras 2006*, 2006.

[3] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," in *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, 2005, pp. 44-51.

[4] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hahnel, "Inferring activities from interactions with objects," *Pervasive Computing, IEEE,* vol. 3, pp. 50-57, 2004.

[5] A. Vehkaoja, M. Zakrzewski, J. Lekkala, S. Iyengar, R. Bajcsy, S. Glaser, S. Sastry, and R. Jafari, "A resource optimized physical movement monitoring scheme for environmental and on-body sensor networks," in *Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments* San Juan, Puerto Rico 2007 pp. 64-66

[6] R. Jafari, W. Li5, R. Bajcsy, S. Glaser, and S. Sastry5, "Physical Activity Monitoring for Assisted Living at Home " in *4th International Workshop on Wearable*

and Implantable Body Sensor Networks (BSN 2007), Aachen University, Germany 2007, pp. 213-219.

[7] H. Ghasemzadeh, E. Guenterberg, K. Gilani, and R. Jafari, "Action coverage formulation for power optimization in body sensor networks," in *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*, 2008, pp. 446-451.

[8] G. Guerra-Filho, C. Fermuller, and Y. Aloimonos, "Discovering a language for human activity," in *AAAI 2005 Fall Symposium on Anticipatory Cognitive Embodied Systems*, Washington,D.C., 2005.

[9] G. Guerra-Filbo and Y. Aloimonous, "A Language for Human Action," *Computer,* pp. 61-69, May 2007.

[10] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE,* vol. 77, pp. 257-286, 1989.

[11] J. K. Aggarwal and P. Sangho, "Human motion: modeling and recognition of actions and interactions," in *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, 2004, pp. 640-647.

[12] N. Naotake, Y. Junichi, and K. Takao, "Human Walking Motion Synthesis with Desired Pace and Stride Length Based on HSMM," *IEICE - Trans. Inf. Syst.,* vol. E88-D, pp. 2492-2499, 2005.

[13] L. Reng, T. Moeslund, and E. Granum, "Finding Motion Primitives in Human Body Gestures," in *Gesture in Human-Computer Interaction and Simulation*, 2006, pp. 133-144.

[14] Y. Jin and B. Prabhakaran, "Latent Semantic Mapping for 3D Human Motion Document Retrieval," in *The 31st Annual International ACM SIGIR Conference*, Singapore, 2008.

[15] A. Ogale, A. Karapurkar, and Y. Aloimonos, "View-invariant modeling and recognition of human actions using grammars," in *Workshop on Dynamical Vision at ICCV'05*, 2005.

[16] C. S. Wetherell, "Probabilistic Languages: A Review and Some Open Questions " *ACM Comput. Surv.* , vol. 12 pp. 361-379, 1980

[17] M. Muller, T. Rder, and M. Clausen, "Efficient content-based retrieval of motion capture data," *ACM Trans. Graph.,* vol. 24, pp. 677-685, 2005.

[18] L. Feng, Z. Yueting, W. Fei, and P. Yunhe, "3D motion retrieval with motion index tree," *Comput. Vis. Image Underst.,* vol. 92, pp. 265-284, 2003.

[19] J. Xiang and H. Zhu, "Motion Retrieval Based on Temporal-Spatial Features by Decision Tree," in *Digital Human Modeling*, 2007, pp. 224-233.

[20] O. C. Jenkins and M. J. Mataric, "Deriving action and behavior primitives from human motion data," in *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, 2002, pp. 2551-2556 vol.3.

[21] D. Lymberopoulos, A. S. Ogale, A. Savvides, and Y. Aloimonos, "A sensory grammar for inferring behaviors in sensor networks," in *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, 2006, pp. 251-259.

[22] D. Lymberopoulos, A. Barton Sweeney, T. Teixeira, and A. Savvides, "An easy-to-program system for parsing human activities," ENALAB 2006.

[23] J. Barnes and R. Jafari, "Locomotion Monitoring Using Body Sensor Networks," in *The 1st International Conference on PErvasive Technologies Related to Assistive Environments (PETRA)*, Athens, Greece, 2008.

[24] B. Logan, J. Healey, M. Philipose, E. Tapia, and S. Intille, "A Long-Term Evaluation of Sensing Modalities for Activity Recognition," in *UbiComp 2007: Ubiquitous Computing*, 2007, pp. 483-500.

[25] S. Jiang, Y. Xue, Y. Cao, S. Iyengar, R. Bajcsy, P. Kuryloski, S. Wicker, and R. Jafari, "CareNet: An Integrated Wireless Sensor Networking Environment for Remote Healthcare," in *Third International Conference on Body Area Networks*, Tempe, Arizona, USA 2008.

[26] "Quiet care systems, Available: www.quietcaresystems.com."

[27] "E-Neighbor system from Healthsense. Available: www.healthsense.com."

[28] R. Marin-Perianu, M. Marin-Perianu, P. Havinga, and H. Scholten, "Movement-Based Group Awareness with Wireless Sensor Networks," in *Pervasive Computing*, 2007, pp. 298-315.

[29] S. C. Johnson, "Hierarchical Clustering Schemes," *Psychometrika,* vol. 2, pp. 241-254 1967.

[30] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in *5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, 1967, pp. 281-297

[31] P. Berkhin, "Survey Of Clustering Data Mining Techniques," in *Accrue Software*, San Jose, CA, 2002.

[32] D. Steinley and M. J. Brusco, "Initializing K-means Batch Clustering: A Critical Evaluation of Several Techniques " *J. Classif.,* vol. 24, pp. 99-121 2007.

[33] S. Lamrous and M. Taileb, "Divisive Hierarchical K-Means," in *Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, 2006, pp. 18-18.

[34] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis " *J. Comput. Appl. Math.* , vol. 20, pp. 53-65 1987

[35] S. Detlef, "Minimization of decision trees is hard to approximate," *J. Comput. Syst. Sci.,* vol. 74, pp. 394-403, 2008.

[36] U. Feige and P. Tetali, "Approximating Min Sum Set Cover " *Algorithmica* vol. 40 pp. 219-234 2004.