Lightweight Power Aware and Scalable Movement Monitoring for Wearable Computers: A Mining and Recognition Technique At the Fingertip of Sensors

Vitali Loseu, Jerry Mannil and Roozbeh Jafari
Embedded Systems and Signal Processing Lab
University of Texas at Dallas
Richardson, TX 75080-3021
{vitali.loseu, jerrymannil, rjafari}@utdallas.edu

ABSTRACT

Activity monitoring using Body Sensor Networks(BSN) has gained much attention from the scientific community due to its recreational and medical applications. Suggested techniques for activity monitoring face two major problem. First, systems have to be trained for the individual subjects due to the heterogeneity of the BSN data. While most solutions can address this problem on a small data set, they have no mechanics for automatic scaling of the solution as the data set increases. Second, the battery limitations of the BSN severely limit the maximum deployment time for the continuous monitoring. This problem is often solved by shifting some processing to the local sensor nodes to avoid a very heavy communication cost. However, little work has been done to optimize the sensing and processing cost of the action recognition. In this paper, we propose an action recognition approach based on the BSN repository. We show how the information of a large repository can be automatically used to customize the processing on sensor nodes based on a limited and automated training process. We also investigate the power cost of such a repository mining approach on the sensor nodes based on our implementation. To assess the power requirement, we define an energy model for data sensing and processing. We demonstrate the relationship between the activity recognition precision and the power consumption of the system during continuous action monitoring. We demonstrate the energy effectiveness of our approach with a classification accuracy constraint based on limited data repository.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications—*Data mining*

General Terms

Design, Algorithms, Experimentation

Wireless Health '11, October 10-13, 2011, San Diego, USA

Keywords

Body Sensor Networks, String Templates, N-grams, Patricia tree, Data Mining, Power Optimization

1. INTRODUCTION

Some phenomenons, which we could observe only approximately, now can be observed quantitatively with the help of sensors and sensor networks. Body Sensor Networks (BSNs) are a particularly interesting field in the sensor network study. They allow quantitative observations of human body, ranging from the muscle activity monitoring to reading the brain patterns. An ability to collect such readings itself is not revolutionary. However, in the past such observations have been tied to constrained lab conditions and expansive hardware, and generally were associated with the well funded industries such as medicine and military research. BSN are revolutionary due to their portability and a remarkably low cost, which makes precise sensor measurements of the human activity widely accessible not only in medicine but also sports training and games. While not providing a conceptually new functionality in medicine, BSNs make existing functionalities affordable to a larger population.

The portability requirement suggests that BSN nodes must exhibit a certain level of autonomy, as a trained operator may not be available at all times. It also implies that BSNs may be deployed in unexpected environments, so the sensor nodes should be able to evaluate the deployment scenario and adjust their operation. In the context of the inertial sensor monitoring, this requires an ability to perform action recognition and adjust the processing based on the movement the subject is performing. While a great body of literature is available on action recognition, the task is not trivial and faces serious challenges that are not well addressed. Human movements are not precise, which means that different subjects may perform the same movement very differently. This, in turn, means that a general action recognition system has to be trained on individual subjects to obtain a high performance level. Such individual attention is not practical, especially in a portable system with numerous users. Additionally, this setup is not scalable, as even if the system is individually trained for a specific subject, an extensive training process has to be repeated in case the set of target movements changes. The training process, if present, has to be automated. The nodes should collect the training data, interface with an external system, and load the configuration parameters.

The second challenge is the life time of the BSN nodes. For a continuous monitoring system to be effective, it has to be able to operate for an extended period of time. There are three sources of energy use in the BSN nodes. Energy is spent on sampling sensors, processing sampled data on the microcontroller, and forwarding the data out via a wireless link. While in general the communication accounts for the majority of the energy expenditure in BSNs, it may not be the case in a distributed action recognition system where the amount of data to be communicated over the wireless is greatly reduced. Human actions are relatively slow, for example, a *sit to stand* action normally takes about a second. Additionally, during a normal day, we do not perform the *sit to stand* action very often. In the case of distributed activity monitoring, only a label has to be communicated when the action is detected. Taking into account the speed and frequency of performing different actions, such as *sit to stand*, this will significantly reduce the amount of data to be communicated compared to forwarding the raw sensor readings. It is necessary to evaluate the energy cost of not only the wireless but all the system components.

In this paper, we propose a novel automated action recognition approach based on our previous work about the BSN repository mining[26]. It is based on the repository that has a potentially large and diverse amount of sensor readings describing different movements, different subjects, and different node placements. Our system mines the repository to find the observations most similar to the limited training data, and uses them as training. This step provides a fast, flexible and scalable training process based on the extensive data set in the BSN repository. This also means that the training data is largely out of control of our system. This suggests that the number of nodes, power consumption, and other parameters of the training data set may not be in line with the particular deployment's requirements. To address this issue, we define an energy model for the node execution that includes the cost of data collection and processing for individual sensors and sensing axis. Based on this model, we define a relationship between the power consumption and the accuracy of recognition. To evaluate our approach, we implemented the partial functionality on the sensor nodes to verify its complexity, execution time, and energy cost. We then evaluated the quality of the approach based on a limited repository we created.

2. RELATED WORKS

Activity monitoring is an important task in the field of BSN. Researchers might be interested in the type of tasks that a subject performs [37] or in the properties of a subset of the performed activities [38]. While many approaches attempt to address these tasks, they face many problems not apparent during the limited lab experiments and inherent to the practical deployment of the long term continuous activity monitoring. These problems include sensor type, sensor placement, movement, and subject variations. There is no consensus on what sensors are preferred for activity monitoring. Some researchers select accelerometer based sensors only [41]. Other select gyroscopes [25], magnetometers [23], or even RFID sensors [15]. Different deployment scenarios have different sensor demands. For example, gyroscopes excel at capturing rotations; however, they generally require an order of magnitude more power than accelerometer sensors. It is possible to approximate the information a sensor would provide from other sensors, for example some work has been done on approximating gyroscope reading using accelerometers [39]. However, these approximations are not precise and can potentially hurt the accuracy of recognition. Sensor placement is extremely important since even a slight misplacement can potentially affect the structure of the observation [16]. It is easy to make sure that the sensors are placed correctly in the lab conditions during short experiments; however it is not something that can be easily controlled during a remote continuous deployment. Even when the sensors are placed precisely, different subjects may perform the same movement differently. It heavily affects the statistical learning approaches such as Hidden Markov Model(HMM) [20], k-Nearest Neighbor(k-NN) [24], neural networks [33], Dynamic Time Warping [18], and support vector machines(SVM) [34] based approaches. As a result, these approach require to be trained on the subject they will be tested on to achieve the optimal performance. This is acceptable on a small data set, but may be impractical and, most importantly, not scalable.

In addition to the issues introduced by the heterogeneous nature of the BSN sensor readings, it is important to consider the maximum lifetime of the system. Intuitively, the maximum lifetime is the time that the system can perform its function. While in other types of systems the maximum lifetime can be increased through the introduction of redundancies, it is not an acceptable solution in the case of BSN based systems due to the wearability requirements. Wearability dictates that a BSN would contain only the minimum number of sensor nodes required for the task. This mean that the maximum life time can be defined as the time before the first sensor node failure. Sensor nodes require energy to collect the sensor data, process it, and forward it to a base station. Wireless communication is known to consume an order of magnitude more energy than processing [26], which renders centralized approaches which forward the raw sensor readings to the basestation to make the classification decision [40] unacceptable. The processing needs to be implemented locally, to allow sensor nodes to make individual classification decisions and communicate only the minimum amount of information. Many approaches in the literature address this problem [28, 42], however they do not consider the fact that after reducing the amount of data to be communicated, the wireless link may no longer accounts for the majority of the power consumption. This suggests a need for an energy model which explicitly considers the cost of sampling different sensors, and processing the collected data.

The task of optimizing the energy lifetime of the system due to sensing and processing can be implicitly addressed via optimal node selection [20] or action coverage [11] approaches. It ensures that only the nodes that are required for movement classification are selected, and little energy is spent needlessly. However, this approach does not address the fact that not all of the sensor modalities and not all of the sensing axis can contribute to an individual classification decision. For example, in a setup with a tri-axial accelerometer and a gyroscope placed on the belt, only the forward facing accelerometer can be used to recognize walking. While it is not practical to expect an individual sensor node design for each of the monitoring tasks, a variety of Microelectromechanical systems (MEMS) sensors allow individual axis power manipulations [31]. This means that a microcontroller, can potentially disable sensing axis or even entire sensors to preserve power. Each of the sensing axes can be considered to be an individual classifier. During combining individual classifiers at sensor node, it is possible explicitly considering the power and accuracy trade of. This can be a very useful tool, as different applications may have different run time and accuracy requirements. For example, an application that collects statistics about the activities a subject performs every day may accept a lower classification accuracy in exchange for a longer lifetime of the system. On the other hand, a medical application that tracks deterioration of the Parkinson disorder based on the gait parameters of walking would favor the classification accuracy over the longer lifetime.

3. SENSING ARCHITECTURE

Our sensing system consists of several XBow®TelosB sensor nodes with custom-designed sensor boards shown Figure 1. Each sensor board has a tri-axial digital accelerometer and a bi-axial analog gyroscope. Sensor nodes sample their sensors at $50Hz$, perform limited local computations, and transmit their data wirelessly to a basestation. In this experiment, the basestation is a sensor node connected to a PC via USB. During the experiment, we also use a Logitech camera to record video of the movement trials. The video frames and data samples are recorded and synchronized in MATLAB. The video of the trials is later used as a gold standard during classification verfication.

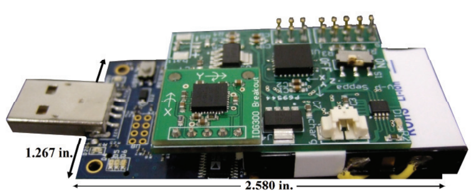

Figure 1: Mote with inertial sensors

4. BSN REPOSITORY APPROACH

The BSN repository organization and mining approach, we introduced in our previous work [26], can be utilized to address the heterogeneous nature of the BSN data. The repository does not enforce any constraints on the data itself, but rather provides a way to organize and mine heterogeneous data quickly and efficiently. This means that it can potentially hold observations of different subjects, sensor modalities and placements, providing flexible and reliable training data for a variety of deployments. Additionally, the mining approach, defined for the repository, is designed to be fast and lightweight with potential implementation on the sensor nodes in mind. In the next sections, we outline the procedure that allows utilization of the repository data for training and repository mining for the activity monitoring.

4.1 Overview of the Mining Approach

While the repository stores heterogeneous data, it uses a common approach for data representation in order to allow data mining. Every entry in the repository shares the same-floating point preprocessing. When the data is collected, the system extracts a set of common features, such as first and second derivative, from the raw sensor readings. Every point in the data is then clustered with respect to these features using a Gaussian Mixture Model(GMM) based clustering approach [9], where the cluster heads are also defined for the entire repository. A *motion transcript* [13, 12] is then generated for the new observation by replacing individual points in the raw signal with the associated cluster labels [10]. Note that an individual motion transcript is generated for each of sensing axis used to collect the data. A set of motion transcripts, representing each of the sensing axes on the nodes, is used to mine the data or make a classification decision.

Since the mining approach should be able to tolerate minor differences in the movement performance, it does not employ entire transcripts. Instead, it focuses on the *important transitions*, or transitions which can differentiate a given movement from the other movements in the repository, in motion transcripts. It is done with the help of n-grams [36], or substrings of length n. Based on the training data set, the repository selects n-grams that correspond to the important transitions for each of the movements using the idea of Information Gain(IG)[19]. It is an important step, as movements can potentially have a great number of transitions, which can slow down and obscure the mining process significantly. This process is performed for every movement in the repository. Once the repository training is concluded, every movement is represented with a unique set of n-grams at each sensing axis. The movements are grouped based on not the artificial conceptual tasks, but the similarities in the signal. Meaning, that two conceptually identical movements with signals that do not look alike are stored as two different movements. For example, a sit to stand action performed on an office chair and a sit to stand action performed on a comfortable sofa can be stored as two different movement, and represented with two different sets of n-grams.

To identify an unknown movement, the system can generate motion transcripts based on the sensor data and use them as input to the mining approach. The approach extracts all of the n-grams found to be important for any of the movements of interest from the corresponding transcripts and applies a Patricia tree [17], where the edges correspond to individual n-grams and leafs correspond to movement classes, based classifier to detect the movement.

4.2 Action Recognition Training

While the repository can contain a great deal of heterogeneous data, it is still unclear how that data can be used in a particular system deployment. Specifically, it is unclear how the repository data can be used for training an activity recognition system for a specific subject. It was mention in Section 1 that a system that requires manual training is not a practical or scalable solution. To address it, we define an automated training process based on a limited training sample.

The flow of the training process is demonstrated in Figure 2. At the beginning of training, the subject is outfitted with sensors required for the study, and asked to perform the target movements a limited number of times. The data of the trial movements, along with the node locations, is forwarded to the BSN repository. The repository combines the knowledge of the node placement with the raw sensor readings to calculate motion transcripts that represent the training data. Based on these transcripts, the system can search the repository for movements, trials, or even subjects which most resemble the training subject. Once this information is discovered, the system can use the information, found most relevant for the training subject, as the training data for the activity monitoring. This approach allows the system to take advantage of the entire data set of the repository, relevant for the subject, based on only the automatically collected limit training data. Additionally, this approach is scalable, because by identifying the set of subjects in the repository closest to the target subject, it is possible to train the system for the movements that were not directly observed during training.

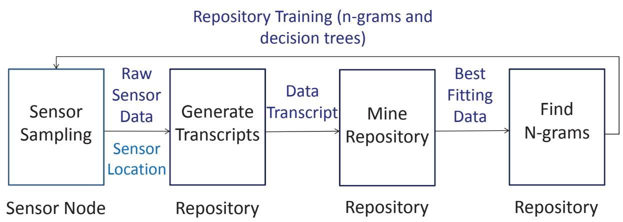

Figure 2: Training Flow for Activity Recognition

Note that the quality of the recognition is based on the assumption that the repository contains a lot of information from different subjects and movements. It follows, that it is beneficial to extend the size of the training data set. However, the original training data may become outdated as the size of the training set increases, because the current approach is based on a fitting model with respect to the available data. It is beneficial to incrementally retrain the system as the amount of data available for training increases. The details of such a procedure are outside the scope of this paper.

5. ACTION RECOGNITION EXECUTION

Once the specific subset, relevant to the training subject, is recognized in the repository, it is possible to use the n-grams that characterize training data for the activity recognition for the current target subject. This can be achieved by loading the relevant n-grams onto the node to mine the incoming data stream. We will discuss the quality of the activity recognition based on this model in Section 7.4. In the remainder of this section, we will first discuss the feasibility of implementing the mining algorithm on the node, and show its performance based on our limited implementation. We then will consider the power profile of such an implementation.

5.1 Transcript Generation Node Implementation

For the purpose of this study, we implemented the transcript generation approach on the TelosB [30] sensor nodes using TinyOS [22]. The raw sensor data is passed through a simple windowed filter that computes a z-score [29]. During the execution, features are extracted from the raw sensor reading. Utilizing the clusters defined in the repository, labels are generated on the motes based on these features. In particular, we implemented clustering based on GMM on the sensor nodes. Due to the resource constraints, clustering algorithms are not suited to run on these sensor nodes because of the amount of heavy computations they require. Additionally, while the majority of the PC-based simulations during the development utilize the floating point arithmetic, BSN devices have no efficient way to perform floating point computations. Specifically, the TelosB node utilizes a $4MHz$ TI MSP430 microcontroller based on a simple 16 bit RISC architecture. It lacks hardware multiplication and division operations, which means that they need to be simulated in the software. It does, however, support a hardware multiplier peripheral, that efficiently performs 16 bit and 8 bit multiplications. To be able to implement transcript generation on the sensor nodes, the implementation needs to be able to simplify the complexity of the algorithm without a significant loss in accuracy while observing the hardware constraints of the platform.

In the GMM based clustering, finding the label for a raw sensor reading amounts to finding the cluster that has the highest probability density value. It involves computing the individual probabilities for each of the clusters. In our implementation, the individual probability density values are computed by finding Mahalanobis distance [27] over the feature set. To simplify the processing, we moved the probability density computations to logarithmic space, which allows us to convert multiplications into additions which are easier to compute on the sensor nodes. Second, the floating point operations were converted to fixed point operations. This was done by representing the GMM coefficients as a fixed point numbers via multiplying the original coefficients by a large constant. The feature vectors from sensor data were also scaled appropriately to work with these coefficients. We then utilized the multiplication hardware peripheral available in the microcontroller to speed up the remaining multiplications. All divisions by constant numbers, occurring as a part of the coefficient calculation, were converted to multiplication by their reciprocals with appropriate scaling to maintain precision. Finally, we truncated the lower bits of the operands to ensure that no saturation of results takes place, which allowed us to to maintain the accuracy of clustering. Note that even though we have utilized a specialized hardware peripheral for classification, the log space conversion is still imperative to the calculation since addition and subtractions still take much fewer instructions than multiplications.

5.2 Transcript Generation Performance

In order to evaluate the feasibility and properties of our solution, we measured the execution time of the different code blocks of the system. This way we can evaluate how changing the data mining parameters will affect the running time. Additionally, we can verify whether the running time is sufficiently low to allow for a reasonable data collection frequency. We measured the execution time by toggling one of the GPIO pins available on the TelosB sensor node. We then connected an oscilloscope to that pin, and measure the period of the signal as the task was executed repeatedly. The results of this measurements are presented in Table 1. The first and second derivatives are calculated based on a windows of three samples centered about the point. The other features, such as mean, max, min, etc, are calculated based on a window of 25 samples, which translates to $\frac{1}{2}$ sec in our case. A couple of observations can be made about the results. First, the feature extraction time depends on the type of features and can not be predicted without more specific information. For example, the mean is implicitly extracted to calculate the other features, therefore two data sets that differ only by *max* and *mean* significantly differ in the execution time. The cluster generation takes about $1.1ms$ per feature and scales linearly. Finally, if a data collection is processed at $50Hz$, it means that in order to maintain stable processing the data has to be collected and processed within $20ms$. The table shows that even when a larger number of features is used, the overall time stays below $20ms$.

Table 1: Measured Execution Time

Processing		Running
Type	**Details**	**time**
Data collection	n/a	$4.9ms$
Pre-Processing	n/a	$1.3ms$
Feature extraction	f', f''	$98\mu s$
	f', f'', mean, std	$2.3ms$
	f', f'', min, max, std	$3.4ms$
	f', f'', min, mean, std	$3.05ms$
	f', f'', min, mean, std, rms	$3.75ms$
Cluster Generation	f', f''	$2.18ms$
	f', f'', mean, std	$4.4ms$
	f', f'', min, max, std	$5.56ms$
	f', f'', min, mean, std	$5.56ms$
	f', f'', min, mean, std, rms	$6.7ms$

After the measurements, we performed theoretical verification of our results to make sure that both software and hardware performance is consistent. Based on the time of execution and the microcontroller speed, we calculated the number of instructions performed by each of the processing blocks. We then disassembled the TinyOS code using *msp430-gcc*, and verified whether the results matched. For example, the feature extraction, using the first and second derivative, was measured to take $98\mu s$. On a $4Mhz$ microcontroller, it translates into $4MHz \times 98\mu s = 392$ instructions. From the analysis of the disassembled code, its takes about 400 instructions to complete feature extraction. Namely, the initialization takes two instructions, the loop of 79 instructions performed for each of 5 sensing axis, and 8 instructions for the bookkeeping at the end. Using this type of analyses, we verified that the execution time numbers we measured are close to the theoretical values we expected. In Section 7.2 we will discuss the quality of recognition our transcripts, generated on the node can achieve.

5.3 Power Profile of Activity Monitoring

During a data collection, where sensor nodes continuously forward all of the sensor data to the basestation, wireless plays a major role in the node's power consumption. In fact, the wireless communication consumes an order of magnitude more power than data collection or processing [32]. This is not the case when a continuous action recognition is considered. This can be easily visualized based on the amount of data required in both cases. For this setup, we consider sensor nodes equipped with a tri-axial accelerometer and gyroscope. The previous work has shown that a zigbee based low power radios, such as $cc2420$ radio [4], utilize about $80mW$ of power for transmission [21]. If we consider a node that is active for 8 hours, with a duty cycle of the movements of interest of .5%, meaning that movements of interest occure .5% of the time. Movements we consider, such as sit to stand, normally take about $.5 - 1$ seconds to perform. In this contest, the .5% duty cyle suggests that we expect about $150 - 300$ occurencies of the movement during the 8 hour period, which is a reasonable estimation of the daily activities. With this duty cycle the cost of communication over the 8 hour period can be estimated as $12J$, which in average transaltes into $.4mW$ power consumption. We also assume that the radio is only turned on when it has data to transmit. However, based on our estimation the radio will have to be turned on and off only a limited number of times. This makes the power overhead of turning the radio on and off orders of magnitude less than the rest of the communication power estimation [2]. Due to this fact, we do not include it in the model.

Table 2 demonstrates the expected power consumption for the MSP430 [1, 30], a 3-axes MEMS accelerometer [31], and a 3-axes MEMS gyroscope [7] designs. Under the same 8 hour operation requirements, the cost of running the microcontroller can be estimated as $260J$, and the cost of sensing a 3-axes accelerometer can be estimated as $155J$. The cost of radio communication is under 3% of the cost of sensing and processing the data, even without considering the cost of the gyroscope. This suggests that to optimize the power of the activity monitoring application, a power model based on the sensing and processing is required. Additionally, from Table 2 it is clear that the power consumption of the individual sensors is on par with that of the microcontroller. Suggesting that the power model should consider the classification accuracy and cost of individual sensing axes as opposed to the entire sensor node operation.

Table 2: Power Consumption of a Microcontroller and Common BSN Sensors

Device	Power Consumption
Microcontroller	$5mW$
1 Axis Accelerometer	$1mW$
1 Axis Gyroscope	$5mW$

6. POWER OPTIMIZATION

Based on the training data in the repository, the system can calculate the effectiveness of each classifier, one for each of the sensing axes, based on the cross-validation set. Effectively, for each of the sensing axes, the system can compute the contribution to the classification decision and the energy cost of making the classification decision. A common approach of combining classifiers with respect to both their quality and cost is based on the utility theory [8]. This idea is widely used in different applications such as Bayesian Information Criterion(BIC) metric[35], game theory [3], and various economical [14] and behavioral [6] models. The overall idea is to represent a complex decision that includes multiple parameters, such as accuracy and cost, into a unified function, or utility, that demonstrates how desirable the decision is. For a classification problem with multiple classifiers, it is common [5] to represent the utility of a classifier as

$$U(x_k|A_i) = accuracy(x_k|A_i) - \alpha \times cost(x_k|A_i) \qquad (1)$$

where x_k is an unknown observation, and A_i is one of the classifiers. This utility function is based on the classification accuracy of A_i that considers a penalty for the cost of computing A_i. The parameter α is introduced to allow flexibility for the weight of the cost function. Specifically, the $accuracy(x_k|A_i)$ can be defined as the probability of the correct classification of the trial x_i by the classifier A_i, and extracted from the data available in the repository.

$$accuracy(x_k|A_i) = P(x_k|A_i) \tag{2}$$

This is a simple case where only one testing sample is considered. When observing a data stream that potentially contains multiple equally likely movements, the quality of a classifier can be defined as the average probability of classifying all of the unknown movements

$$accuracy(X|A_i) = \frac{\sum_{t=1}^{N} P(x^t|A_i)}{N} \tag{3}$$

where X is the set of all the unknown observation trials, N is the number of unknown movements the classifier can observe, and x^t is an observation of a trial of movement t.

Additionally, the system may chose to use more than one classifier at a time. To address this, we can rewrite (3) as

$$accuracy(X|A_1, ...A_m) = \frac{\sum_{t=1}^{N} P(x^t|A_1, ..., A_m)}{N} \tag{4}$$

where m corresponds to the number of classifiers.

For the purposes of this formulation, we define the cost function as the power cost of sampling the data and processing all classifiers to be used on that node.

$$cost_n = P_n^{sense} + P_n^{process} \tag{5}$$

where $cost_n$ corresponds to the power consumption at $Node_n$, P_n^{sense} and $P_n^{process}$ correspond to the power cost of sensing and processing at $Node_n$ respectively. Both the sensing and processing power levels are defined in Table 2, so the equation (5) can be expanded to

$$cost_n = P_n(acc) * a + P_n(gyro) * g + P_n(proc) * p \tag{6}$$
$$\text{where p} = 1 \text{ if } (a+g) > 0$$
$$\text{p} = 0 \text{ if } (a+g) = 0$$

where $P_n(acc)$ is the power cost of sensing one axis of accelerometer, a is the number of axes of accelerometer active, $P_n(gyro)$ is the power cost of sensing one axis of gyroscope, g is the number of gyroscope axes active, and $P_n(proc)$ is the power cost of having the microcontroller on. Note that there is no reason to keep the microcontroller on if no axes are sampling data. This behavior is defined with the help of variable p. The overall cost of the system can be defined as the summation of the cost of individual nodes

$$cost(x_k|A_i) = \sum_{j=1}^{n} cost_j \tag{7}$$

Based on (3) - (7) the utility function can be defined as

$$U(X|A_0, ..., A_k) = \frac{\sum_{t=1}^{N} P(C|x^t, A_0, ..., A_k)}{N} - \alpha \times \sum_{j=1}^{n} cost_j \tag{8}$$

Since the larger utility corresponds to better quality of detection at a better cost, the objective function for a set of movements X can be defined as

$$\text{maximize } U(X|A_0, ..., A_k) \tag{9}$$

7. EXPERIMENTAL RESULTS

In this section we demonstrate different aspects of the solution. First, we verify the quality of the transcript generation on the sensor nodes. We then verify the scalability of the model by verifying the quality of recognition when the system is trained and tested on different subjects, as defined in our training model. Finally, we demonstrate the energy savings our model can generate when it tries to select sensing nodes, and sensing axes for recognition.

Table 3: Pilot Application Movements

No.	Description
1	Stand to Sit
2	Sit to Stand
3	Stand to Sit to Stand
4	Sit to Lie
5	Lie to Sit
6	Sit to Lie to Sit
7	Bend and Grasp
8	Turn Counter Clockwise 360 degrees
9	Look Back Clockwise
10	Move Forward (one step)
11	Move Backward (one step)
12	Move Left(one step)
13	Grasp Object with One Hand, Turn 90 Deg and Release
14	Grasp Object with Two Hands, Turn 90 Deg and Release
15	Jumping

7.1 Experimental Setup

For the experiment we collected data of fifteen movements from three subjects. The details of the experimental movements can be found in Table 3. Every subject repeated each movement ten times to increase the size of the data set. Each subject was equipped with nine sensor nodes positioned as demonstrated in Figure 3.

7.2 Quality of Local Transcripts

To verify the quality of the local transcript generation we compared transcripts generated on a sensing node to the transcripts generated on a computer. During the experiment, each node collected data and generated local transcripts as demonstrated in Figure 4. The figure corresponds to a sequence of movements execution. While the specific

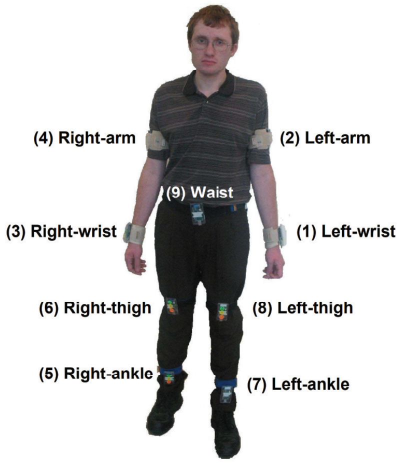

Figure 3: Subject with motes equipped

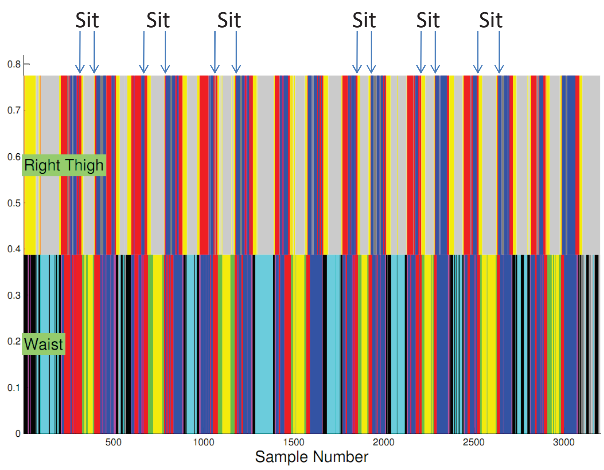

Figure 4: Sample of the Node Generated Transcripts

trials of the movements are not identical, the figure demonstrates that the structure of the generated transcripts is maintained. Furthermore, it shows that the differences in the trial length are consistent between the nodes in the system, meaning that the inconsistencies are generated by the movement variations and not node specific error. Nodes communicated the local transcripts to the server along with the raw sensor data. On the server side, the raw sensor data was converted to the transcripts. Finally, we compared the two sets of transcripts. Since both transcripts represent the same data and have the same length, the comparison is done by verifying if the points with the same index in two transcripts are the same. If the points are not the same, they are reported as mismatched.

To verify the local transcript generation quality, we performed an experiment with two sensor nodes. One node was located on the waist, and the other was located on the right thigh. The experiment consisted of three movements, namely: *sit to stand*, *sit to lie*, and *turn in bed*. Table 4 show the mismatch between node and computer generated transcripts for these movements when performed by two subjects. The first number in the cell indicates the number of mismatched points, while the second number indicates the total length of the transcript. In the worst case, the transcript generated on the nodes differ by 2% of the overall transcript length. This indicates that transcript generation can be performed on the sensing nodes without introducing a segnificant amoutn of error to the evaluation of the model performed in MATLAB.

Table 4: Transcript Mismatch

Movement	Waist Node	Right Thigh Node
Subject 1		
Sit to Stand	2/3586	84/3586
Sit to Lie	9/3196	77/3196
Turn in Bed	2/2194	12/2194
Subject 2		
Sit to Stand	1/2472	34/2472
Sit to Lie	2/1998	43/1998
Turn in Bed	1/1718	40/1718

7.3 System Scalability

The repository based system scalability will be improved over training the system for every individual subject. This setup requires only a limited movement set to be obtained from the individual subjects in order to locate the most fitting training for all of the movements of interest from the repository. This approach, however, has an assumption that such training would work without a considerable accuracy decrease. To test this hypothesis we trained the system on the data of one subject and tested on a completely different subject. It must be noted that in a large repository it is more likely to find observations, or even a subject, that well resemble the subject that the system is being deployed for.

When we trained the system with observations of 15 movements performed by the first subject and tested on the third subject, we got an average accuracy of 75%. When the system was trained on the data of the second subject and tested on the third subject, we got an average accuracy of 87%. This indicates that the step of selecting the most fitting training data is essential, as some subjects have a closer correlation in movement performance than the others. Additionally, while 87% accuracy is not perfect, we expect that in a repository with more subjects the accuracy of training on a different subjects will be improved. Finally, we considered a case where the system was trained on both first and second subject and tested on the third subject. This case produced average accuracy of 92%. Which suggests that it may not be practical to search for a subject in the repository that the system can be trained on. Instead, the system can be trained on a mixture of different subject data with superior results. However, the specific details of the system training are out of the scope of this paper and will be investigated in the future work.

7.4 Optimization for Power

We first consider the case where the individual sensing axes can not be disabled and the system can only select the best set of nodes. As a result, each node has a constant cost of sensing and processing, which equals to $18mW$. For this experiment, the system is trained on half of the trials of each subjects, and is tested on the second half of the trials. Table 5 shows average precision and power cost of the computation. It demonstrates that the most effective individual node is the waist node. It also shows that the left wrist, ankle and thigh nodes can be used to further refine the precision.

Table 5: Node Selection

Sensing Node	Average Precision	Power Cost
Waist	0.9569	18mW
Waist, Left Wrist	0.9843	36mW
Waist, Left Ankle	0.992	36mW
Waist, Left Thigh	0.9961	36mW
Waist, Left Wrist, Left Ankle	1	54mW
Waist, Left Wrist, Left Thigh	1	54mW

The actual selection of the nodes is highly dependent on the parameter α in the formulation. In other words, it is essential to define the importance of the power conservations in the context of a given application. If the selected α is greater than 0.7 than only the waist node will be selected. If α is between 0.04 and 0.7 the set of waist and left thigh will be selected. Finally, if α is below 0.04 the system will select the waist, left wrist, and left ankle nodes. Table 6 shows the effective power saving as compared to using all of the nine nodes in the repository with respect to the required error tolerance.

Table 6: Node Selection Efficiency

α	Power Saving	Error Tolerance
$[0.7, \infty)$	89%	4.4%
$[0.04, .7)$	78%	0.4%
$(-\infty, .04)$	67%	$\approx 0\%$

This result shows that at least node selection is required if a system is trained on the data from the repository. Since the training data is not controlled by the person training the system, it may contain redundant sensor nodes. Different applications require different node placement, have different tasks, and require different levels of redundancy. Additionally, this result demonstrated that only one sensor node can be used to recognize a limited set of movements reasonably well, while some nodes may have no useful information about the movements of interest and can introduce confusion to the final decision. The same might be true about the individual sensing axes on a given mote.

We next apply our optimization to the data with an assumption that individual sensing axes can be disabled. This means that we are looking for a subset of sensing nodes and a subset of sensing axes at each node to be used as training data. Intuitively, the system should select one node with a subset of sensing axes because enabling an additional node with one sensing axis enabled is relatively costly.

Once again the selection of α is very important. If α is greater than 2 then the waist node with one axis of the accelerometer will be selected. If the α is between 0.1 and 2 the system will select the left wrist node with two sensing axes enabled. Finally, if α is less than 0.1 the system would

Table 7: Node and axes Selection

Sensing		Average	Power
Node	axes	Precision	Cost
Waist	Acc_x	0.9647	5mW
Waist	Acc_x, $Gyro_x$	0.9649	10mW
Left Wrist	Acc_x, $Gyro_x$	0.9961	10mW
Left Wrist	Acc_x, $Gyro_x$, $Gyro_y$	1	15mW

suggest to enable an additional sensing axes at the left wrist node. Table 8 demonstrates the power saving the system can observe if individual sensing axis are selected as opposed to utilizing all of the nodes and all of the sensing axes.

Table 8: Node and axes Selection Efficiency

α	Power Saving	Error Tolerance
$[2, \infty)$	97%	3.6%
$[0.1, 2)$	94%	$\approx 1\%\%$
$(-\infty, .1)$	91%	$\approx 0\%$

Table 7 demonstrates that not all of the sensing axes contribute to the classification decision equally. In fact, selecting only a small subset of sensing axes can improve the overall system accuracy. This subset is not necessarily the same between different experiments, meaning that the hardware does not need to change and the selection can be made in the software. In addition to the accuracy boost, the system can reduce the power consumption almost by two orders of magnitude when the training data contains possibly redundant nodes.

8. FUTURE WORK

The major weakness of our approach is the training step. During training we assume that the data we are looking for is present in the repository, and that we know how to locate it. The second assumption can create significant issues in the context of sensing node misplacement. Our approach can tolerate some misplacement, as during our experiment we did not attempt to reproduce node placement precisely between subjects. However, we have no direct measure of misplacement and its relationship to the classification accuracy. In our future work we intend to address this problem by first quantifying the displacement, and then defining a transformation model to create a relationship for metadata of different node placements.

9. CONCLUSION

In this paper, we proposed a novel automated action recognition approach based on the idea of a BSN repository. We showed that such an approach is scalable and can potentially handle a large repository of heterogeneous data. Based on our limited implementation we showed that it is feasible to implement our approach on the resource limited sensor nodes without a significant loss of accuracy. Finally, we defined an approach to intelligently select only a subset of nodes and sensing axes based on a utility function to reduce the energy cost of the system, and thus improve the system lifetime. We evaluated the performance of our approach based on the limited repository we created.

10. REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[2] B. Bougard, F. Catthoor, D. Daly, A. Chandrakasan, and W. Dehaene. Energy efficiency of the IEEE 802.15. 4 standard in dense wireless microsensor networks: Modeling and improvement perspectives. 2005.

[3] C. Camerer and R. S. Foundation. *Behavioral game theory: Experiments in strategic interaction*, volume 9. Princeton University Press Princeton, NJ, 2003.

[4] C. CC2420. 2.4 GHz IEEE 802.15. 4/ZigBee-ready RF Transceiver. *Chipcon Products from Texas Instruments*, 2006.

[5] C. Demir and E. Alpaydin. Cost-conscious classifier ensembles. *Pattern Recognition Letters*, 26(14):2206–2214, 2005.

[6] W. Edwards. Behavioral decision theory. *Annual review of psychology*, 12(1):473–498, 1961.

[7] M. Elsayed, A. Emira, S. Sedky, and S. Habib. A single-ended CMOS sensing circuit for MEMS gyroscope with noise cancellation. In *NEWCAS Conference (NEWCAS), 2010 8th IEEE International*, pages 385–388. IEEE.

[8] P. Fishburn. Utility theory for decision making. Technical report, RESEARCH ANALYSIS CORP MCLEAN VA, 1970.

[9] H. Friedman and J. Rubin. On some invariant criteria for grouping data. *Journal of the American Statistical Association*, pages 1159–1178, 1967.

[10] H. Ghasemzadeh, J. Barnes, E. Guenterberg, and R. Jafari. A phonological expression for physical movement monitoring in body sensor networks. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pages 58–68. IEEE, 2008.

[11] H. Ghasemzadeh, E. Guenterberg, K. Gilani, and R. Jafari. Action coverage formulation for power optimization in body sensor networks. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, pages 446–451. IEEE Computer Society Press, 2008.

[12] H. Ghasemzadeh and R. Jafari. Body sensor networks for baseball swing training: Coordination analysis of human movements using motion transcripts. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 792–795. IEEE, 2010.

[13] H. Ghassemzadeh, E. Guenterberg, S. Ostadabbas, and R. Jafari. A motion sequence fusion technique based on pca for activity analysis in body sensor networks. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 3146–3149. IEEE, 2009.

[14] J. Harsanyi. Cardinal utility in welfare economics and in the theory of risk-taking. *The Journal of Political Economy*, 61(5):434, 1953.

[15] L. Ho, M. Moh, Z. Walker, T. Hamada, and C. Su. A prototype on RFID and sensor networks for elder healthcare: progress report. In *Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, pages 70–75. ACM, 2005.

[16] T. Idé, S. Papadimitriou, and M. Vlachos. Computing correlation anomaly scores using stochastic nearest neighbors. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 523–528. IEEE.

[17] S. Inenaga, H. Bannai, A. Shinohara, M. Takeda, and S. Arikawa. Discovering best variable length don't care patterns. In *Discovery Science*, pages 169–216. Springer.

[18] R. Jafari and R. Lotfian. A low power wake-up circuitry based on dynamic time warping for body sensor networks. In *Body Sensor Networks (BSN), 2011 International Conference on*, pages 83 –88, may 2011.

[19] J. Kent. Information gain and a general measure of correlation. *Biometrika*, 70(1):163, 1983.

[20] R. King, L. Atallah, A. Darzi, and G. Yang. An HMM framework for optimal sensor selection with applications to BSN sensor glove design. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 58–62. ACM, 2007.

[21] J. Lee, Y. Su, and C. Shen. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pages 46–51. IEEE, 2007.

[22] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al. Tinyos: An operating system for sensor networks. *Ambient Intelligence*, 35, 2005.

[23] K. Lim, F. Goh, W. Dong, K. Nguyen, I. Chen, S. Yeo, H. Duh, and C. Kim. A wearable, self-calibrating, wireless sensor network for body motion processing. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1017–1022. IEEE.

[24] X. Liu and H. Ferhatosmanoğlu. Efficient k-NN search on streaming data series. *Advances in Spatial and Temporal Databases*, pages 83–101, 2003.

[25] K. Lorincz, B. Kuris, S. Ayer, S. Patel, P. Bonato, and M. Welsh. Wearable wireless sensor network to assess clinical status in patients with neurological disorders. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 563–564. ACM, 2007.

[26] V. Loseu, H. Ghasemzadeh, L. Khan, and R. Jafari. A mining technique using n-grams and motion transcripts for body sensor network data repository. *Proceedings of the IEEE*.

[27] P. Mahalanobis. On the generalized distance in statistics. In *Proceedings of the National Institute of Science, Calcutta*, volume 12, page 49, 1936.

[28] M. Marin-Perianu, C. Lombriser, O. Amft, P. Havinga, and G. Tr
"oster. Distributed activity recognition with fuzzy-enabled wireless sensor networks. *Distributed Computing in Sensor Systems*, pages 296–313, 2008.

[29] G. Milligan and M. Cooper. A study of standardization of variables in cluster analysis. *Journal of Classification*, 5(2):181–204, 1988.

[30] J. Polastre, R. Szewczyk, and D. Culler. Telos:

enabling ultra-low power wireless research. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 364–369. IEEE, 2005.

[31] H. Qu, D. Fang, and H. Xie. A monolithic CMOS-MEMS 3-axis accelerometer with a low-noise, low-power dual-chopper amplifier. *Sensors Journal, IEEE*, 8(9):1511–1518, 2008.

[32] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2):40–50, 2002.

[33] B. Ripley. *Pattern recognition and neural networks*. Cambridge Univ Pr, 2008.

[34] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE.

[35] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[36] C. Shannon. Communication theory of secrecy systems. *MD Computing*, 15(1):57–64, 1998.

[37] M. Stikic and B. Schiele. Activity recognition from sparsely labeled data using multi-instance learning. *Location and Context Awareness*, pages 156–173, 2009.

[38] J. Suutala, S. Pirttikangas, and J. R
"oning. Discriminative temporal smoothing for activity recognition from wearable sensors. *Ubiquitous Computing Systems*, pages 182–195, 2010.

[39] P. Udaya Shankar, N. Raveendranathan, N. Gans, and R. Jafari. Towards power optimized kalman filter for gait assessment using wearable sensors. In *Wireless Health 2010*, pages 137–144. ACM, 2010.

[40] G. Yang. *Body sensor networks*. Springer-Verlag New York Inc, 2006.

[41] W. Yeoh, J. Wu, I. Pek, Y. Yong, X. Chen, and A. Waluyo. Real-time tracking of flexion angle by using wearable accelerometer sensors. In *Medical Devices and Biosensors, 2008. ISSS-MDBS 2008. 5th International Summer School and Symposium on*, pages 125–128. IEEE, 2008.

[42] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tr
"oster. Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. *Wireless Sensor Networks*, pages 17–33, 2008.