# Motion Primitive-Based Human Activity Recognition Using a Bag-of-Features Approach

Mi Zhang
Signal and Image Processing Institute
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089 USA
mizhang@usc.edu

Alexander A. Sawchuk
Signal and Image Processing Institute
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089 USA
sawchuk@sipi.usc.edu

## ABSTRACT

Human activity modeling and recognition using wearable sensors is important in pervasive healthcare, with applications including quantitative assessment of motor function, rehabilitation, and elder care. Previous human activity recognition techniques use a "whole-motion" model in which continuous sensor streams are divided into windows with a fixed time duration whose length is chosen such that all the relevant information in each activity signal can be extracted from each window. In this paper, we present a statistical motion primitive-based framework for human activity representation and recognition. Our framework is based on Bag-of-Features (BoF), which builds activity models using histograms of primitive symbols. We experimentally validate the effectiveness the BoF-based framework for recognizing nine activity classes and evaluate six factors which impact the performance of the framework. The factors include window size, choices of features, methods to construct motion primitives, motion vocabulary size, weighting schemes of motion primitive assignments, and learning machine kernel functions. Finally, we demonstrate that our statistical BoF-based framework can achieve much better performance compared to a non-statistical string-matching-based approach.

## Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications; J.3 [**Computer Applications**]: Life and Medical Sciences

## General Terms

Algorithms, Design, Experimentation, Performance

## Keywords

Pervasive Healthcare, Wearable Sensing Technologies, Human Activity Recognition, Motion Primitives, Pattern Recognition, Bag-of-Features


*IHI'12,* January 28–30, 2012, Miami, Florida, USA.


## 1. INTRODUCTION

In pervasive healthcare, human activity analysis and recognition plays a central role because the specific activities people perform in their daily lives can be used to assess the fitness of human body and quality of life. Traditionally, activity analysis is studied as a computer vision problem where human activities are captured by cameras deployed in the infrastructure. The major drawback of the vision-based platform is its inability to track people beyond the reach of the cameras. The emergence of wearable sensor systems attempts to address this problem. These sensors are miniaturized such that they can be worn on the human body and continuously capture people's activity signals unobtrusively.

Most wearable sensor-based activity recognition techniques represent activities using a "whole-motion" model in which continuous sensor streams are divided into fixed-length windows. The window length is properly chosen such that all the information of the activity can be extracted from each window. Features are then extracted from the window which are used as input to the classifier for classification. Although this "whole-motion" model has proven very effective in existing studies, the performance is highly dependent on the window length [1]. As a possible solution to this problem, motion primitive-based models were proposed and have recently attracted numerous research attention.

The motion primitive-based models are inspired by the similarity of human speech signals and human motion [2]. In human speech recognition, sentences are first divided into isolated words, which are then divided into a sequence of phonemes. Models are first built for the approximately 50 phonemes shared by all words (in English). These phoneme models then act as the basic building blocks to build words and sentences in a hierarchical manner [3]. Following the same idea, in motion primitive-based model, each activity is represented as a sequence of motion primitives which act as the smallest units to be modeled. Different from the "whole-motion" model that examines the global features for human activities, motion primitives capture the invariance aspects of the local features and more importantly, provide insights for better understanding of human motion.

The key issues related to the motion primitive-based model are: (1) constructing meaningful motion primitives that contain salient motion information; and (2) representing activities based on the extracted primitives. Most existing approaches construct primitives either using fixed-length windows with identical temporal/spatial duration or through clustering. Each window is then mapped to a symbol ac-

cording to a specific mapping rule. As a consequence, the continuous activity signal is transformed into a string of symbols where each symbol represents a primitive. Figure 1 shows an example on two activity classes: *walking forward* (top) and *running* (bottom). For illustration purposes, a total of five motion primitives are used (labeled A, B, C, D, E in different colors). In this example, *walking forward* contains five types of motion primitives (A, B, C, D, E) while *running* contains four (B, C, D, E). For both activities, the first line shows the original sensor signal and the second line shows the primitive mapping of the original sensor signal. Below these are five lines showing the locations of the five motion primitives in the signal. The last line is a sample of the symbol string. To build activity models

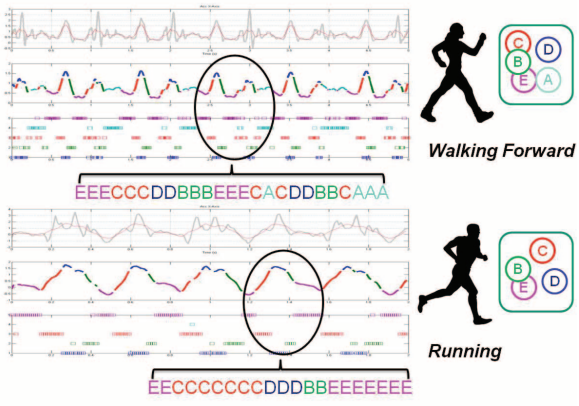


**Figure 1: An example of activity representation (*walking forward* (top) and *running* (bottom)) using five motion primitives (labeled A, B, C, D, E in different colors).**

based on these extracted primitives, one common strategy is to adopt a string-matching-based approach. Specifically, in the training stage, for each activity class, a string which minimizes the sum of intra-class distances is created and acts as a template to represent all training instances belonging to that class. Since different strings in general do not have the same length, the distances between them are normally measured by edit distance (Levenshtein distance) [4]. In the recognition stage, the test instance is first transformed into the primitive string, and then classified to the activity class whose template matches the test instance the best. Although this string-matching-based strategy shows competitive performance in both vision-based and wearable sensor-based activity recognition tasks [5] [6] [7] [8], the main drawback is its high sensitivity to noise and its poor performance in the presence of high intra-class variation [9]. Under such conditions, it is extremely difficult to extract a meaningful template for each activity class. Therefore, to overcome this problem, we use a statistical-based approach.

Our statistical motion primitive-based framework is based on the Bag-of-Features (BoF) model, which has been applied in many applications such as text document classification, texture and object recognition and demonstrated impressive performance [10]. Different from the string-matching-based strategy, our BoF-based framework takes advantage of the state-of-the-art learning machines with the aim to build statistically robust activity models. There are two goals of this work. The first goal is to explore the feasibility of applying a BoF-based framework for human activity recognition and examine whether BoF can achieve better performance compared to the string-matching-based approach. Our second goal is to perform a thorough study on several factors which could impact the performance of the framework. These factors include the size of windows, choices of features, methods to construct motion primitives, size of motion vocabulary, weighting schemes of motion primitive assignments, and kernel functions of the learning machines.

The rest of this paper is organized as follows. Section 2 gives a brief survey of some recent work on human activity recognition. Section 3 introduces the sensing platform and dataset used for this study. Section 4 describes the basic idea of BoF and outlines the key components of the BoF framework. Section 5 presents our experimental results on the evaluations of these factors and compares the performance between BoF and the traditionally used string-matching-based approach. Finally, section 6 concludes this paper and establishes directions for future work.

## 2. RELATED WORK

In keeping with our recognition technique, we broadly group existing activity recognition methods into two categories based on the granularity level human activities are modeled: "whole-motion"-based methods and motion primitive based methods. In this section, we review some recent work from each category respectively.

In the case of "whole-motion" model, different combinations of features and classifiers have been extensively studied on different sets of activities. In [11], Bao *et al.* studied statistical and frequency domain features in conjunction with four classifiers including decision trees (C4.5), decision tables, naive Bayes and nearest-neighbor. Among these classifiers, the decision tree achieved the best performance with an overall recognition accuracy of 84%. Ravi *et al.* in [12] used similar features as in [11]. They compared the performance of various base-level classifiers with meta-level classifiers including Bagging, Boosting, Plurality Voting, and Stacking. Based on the experimental results, they concluded that using meta-classifiers was in general effective. In particular, combining classifiers using Plurality Voting turned out to be the best classifier.

Recently, motion primitive-based approaches receive numerous research attention due to their capability of capturing local characteristics of activity signals. In [6], motion primitives were constructed by dividing the activity trajectory into fixed-length windows with identical spatial duration, where each window was mapped to a motion primitive based on its trajectory direction in the Cartesian space. The problem of activity recognition was then formulated as a standard string-matching problem. Fihl *et al.* in [5] took a similar idea but replaced the standard deterministic string-matching algorithm with a probabilistic-based string-matching strategy by using the probabilistic edit distance instead of the standard edit distance. Keogh *et al.* in [13] tackled the problem from a different angle. They constructed motion primitives based on the shapes of the raw streaming sensor signals where a subsequence of the raw sensor signal was identified as a motion primitive if its shape was maximally representative of an activity class. Activity recognition was performed by searching the subsequence which had the most similar shape as the known motion primitive.

In this work, we follow the basic principles of the motion primitive-based model. Different from the existing approaches which formulat activity recognition as a string/sha-pe matching problem, we take advantage of the statistical learning machines, with the hope that the statistical approach could remedy the drawbacks of the non-statistical approaches and therefore make the recognition system more robust.

# 3. SENSING PLATFORM AND DATASET

For this work, data is recorded using an off-the-shelf multimodal sensing platform called MotionNode [14] (see Figure 2). MotionNode is a 6-DOF inertial measurement unit (IMU) specifically designed for human motion sensing applications. It integrates a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. In this work, only the data sampled from the accelerometer and gyroscope is considered. The measurement range for each axis of accelerometer and gyroscope is $\pm 6g$ and $\pm 500dps$ respectively. The sampling rates for both accelerometer and gyroscope are set to 100 Hz. This setting is high enough to capture all the details of normal human activities [11].

To collect data, six participants with different gender, age, height, and weight are selected to perform nine types of activities listed in Table 1. We select these activities because they correspond to the most basic and common activities in people's daily life and are useful for both elder care and personal fitness applications. During data collection, to extract the maximal information while minimizing the obtrusiveness of the sensing device, a single MotionNode is packed into a mobile phone pouch and attached to the participant's right front hip. Each participant performs five trials for each activity on different days at various indoor and outdoor locations without supervision.

(a) MotionNode platform

(b) During data collection, MotionNode is packed firmly into a mobile phone pouch and attached to the subject's front right hip.

**Figure 2: MotionNode sensor and its placement during data collection**

# 4. THE BAG-OF-FEATURES FRAMEWORK

Figure 3 gives a graphical overview of our BoF-based framework for human activity representation and recognition. The framework consists of two stages. In the training stage, the streaming sensor data of each activity is first divided into a sequence of fixed-length window cells whose length is much smaller than the duration of the activity itself. Features are extracted from each window cell to form a local feature vector. The local feature vectors from all training activity classes are then pooled together and quantified through an unsupervised clustering algorithm to construct the motion vocabulary, where the center of each generated cluster

| ID | Activity | Description |
|---|---|---|
| 1 | Walk forward | The subject walks forward along a straight street/corridor |
| 2 | Walk left | The subject keeps walking counter-clockwise around the circle |
| 3 | Walk right | The subject keeps walking clockwise around the circle |
| 4 | Go up stairs | The subject goes up one flight |
| 5 | Go down stairs | The subject goes down one flight |
| 6 | Run forward | The subject runs forward at his/her normal speed |
| 7 | Jump up | The subject stays at the same position and keeps jumping up |
| 8 | Sit on a chair | The subject sits on a chair either working or resting. Fidgeting is also considered to belong to this class |
| 9 | Stand | The subject stands and talks to somebody |

**Table 1: Activities and their brief descriptions**

is treated as a unique motion primitive in the vocabulary. By mapping the window cells to the motion primitives in the vocabulary, the activity signal is then transformed into a string of motion primitives. Here, we assume that activity signals do not follow any grammar and thus information about the temporal order of motion primitives is discarded. Instead, we construct a histogram representing the distribution of motion primitives within the string, and map the distribution into a global feature vector. Finally, this global feature vector is used as input to the classifier to build activity models and learn the classification function. In the recognition stage, we first transform the unknown stream of sensor data into motion primitives and construct the global feature vector based on the distribution of the motion primitives. Then we classify the unknown sensor data to the activity class that has the most similar distribution in the primitive space. In the remainder of this section, we present the details of all the key components of this framework.

## 4.1 Size of Window Cells

As the first parameter of our BoF framework, the size of window cells is known to have a critical impact on recognition performance [1]. A large size may fail to capture the local properties of the activities and thus dilute the discriminative power of the motion primitive-based model. A small size, on the other hand, is highly sensitive to noise and thus is less reliable to generate meaningful results. This trade-off between discrimination and stability motivates the studies of the size of window cells. Our survey shows that a wide range of window cell sizes have been used in previous work, leading to difficulties in interpreting and comparing their results. At one extreme, Huynh *et al.* in [15] and Krause *et al.* in [16] extracted features from a 4 seconds window and a 8 seconds window respectively. At the other extreme, Stiefmeier *et al.* in [6] adopted a 0.1 second window. In this work, we experiment with window sizes ranging from 0.1 to 2 seconds. The best size is the one at which the classification accuracy reaches the maximum. We did not experiment with window size beyond 2 seconds since the "whole-motion" model has exhibited good performance at and beyond such scales in many existing studies.

## 4.2 Features

It is well understood that high quality features are essential to improve the classification accuracy of any pattern recognition system. In human activity recognition, a variety

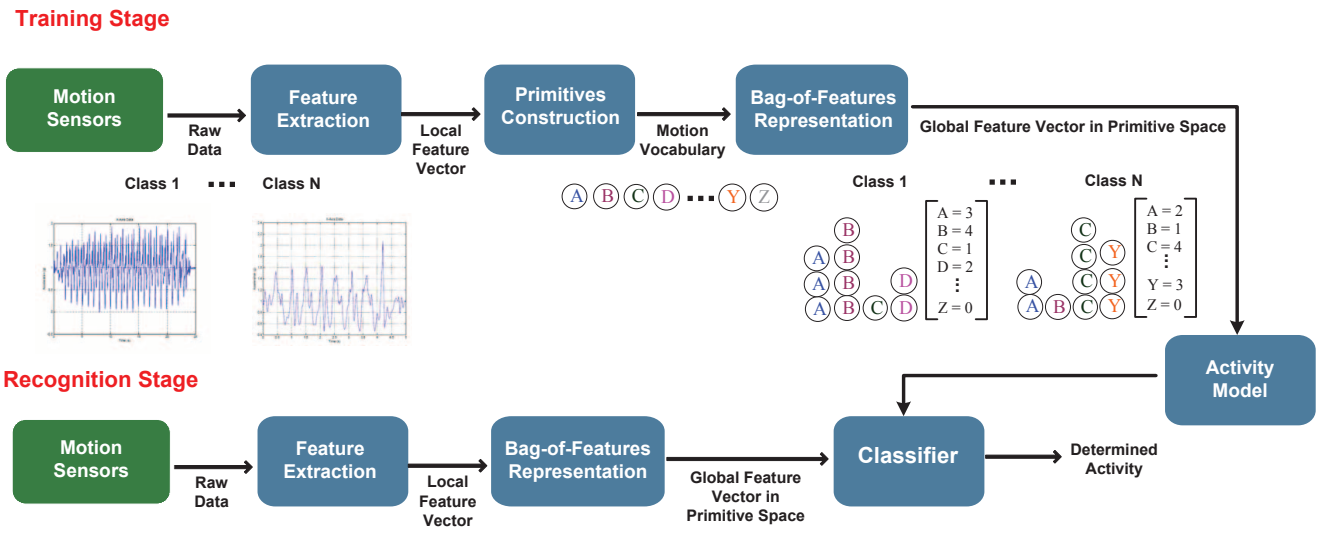


**Figure 3: Block diagram of Bag-of-Features (BoF)-based framework for human activity representation and recognition**

of features both in time and frequency domains have been investigated within the framework of the "whole-motion" model. Popular examples are mean, variance, FFT coefficients, spectral entropy, correlation etc. However, little work has been done on feature analysis at the primitive level. As a consequence, whether the features that work well at the "whole-motion" level can be extended to the primitive level remains an interesting question to be explored.

In this work, we evaluate two feature sets at the primitive level. The first feature set contains traditionally used statistical features. However, at primitive level, since the total number of samples within each window cell is much smaller, complex statistical features such as skewness, kurtosis, and spectral entropy may not be reliably calculated. Therefore, we only consider statistical features that can be reliably calculated at primitive level. Table 2 lists the statistical features we include in this work. These features are extracted from each axis of both accelerometer and gyroscope.

The second set of features are called physical features, which are derived based on the physical parameters of human motion [17]. The definitions of the physical features we study in this work are listed below.

- **Movement Intensity (MI)**: MI is defined as

$$MI(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2}, \quad (1)$$

the Euclidean norm of the total acceleration vector after removing the static gravitational acceleration, where $a_x(t)$, $a_y(t)$, and $a_z(t)$ represent the $t^{th}$ acceleration sample of the $x$, $y$, and $z$ axis in each window respectively. This feature is independent of the orientation of the sensing device, and measures the instantaneous intensity of human movements at index $t$. We do not use MI directly, but compute the mean (AI) and variance (VI) of MI over the window and use them as two features given by

$$AI = \frac{1}{T}\left(\sum_{t=1}^{T} MI(t)\right) \quad (2)$$

$$VI = \frac{1}{T}\left(\sum_{t=1}^{T} (MI(t) - AI)^2\right) \quad (3)$$

where $T$ is the window length.

- **Normalized Signal Magnitude Area (SMA)**: SMA is defined as

$$SMA = \frac{1}{T}\left(\sum_{t=1}^{T} |a_x(t)| + \sum_{t=1}^{T} |a_y(t)| + \sum_{t=1}^{T} |a_z(t)|\right), \quad (4)$$

the acceleration magnitude summed over three axes within each window normalized by the window length. This feature is used as an indirect estimation of energy expenditure [18].

- **Eigenvalues of Dominant Directions (EVA)**: Dominant directions refer to the directions along which intensive human motion occurs. They are extracted as the eigenvectors of the covariance matrix of acceleration along the $x$, $y$, and $z$ axis in each window. The eigenvalues measure the corresponding relative motion magnitude along the directions. In this work, we use the top two eigenvalues as our features, corresponding to the relative motion magnitude along the vertical direction and the heading direction respectively.

- **Correlation between Acceleration along Gravity and Heading Directions (CAGH)**: CAGH is calculated as the correlation coefficient between the acceleration along the gravity direction and the acceleration along the heading direction.

- **Averaged Velocity along Heading Direction (AVH)**: AVG is computed by averaging the instantaneous velocity along the heading direction at each time $t$ over the window. The instantaneous velocity at each time $t$ is calculated by numerical integration of the acceleration along the heading direction.

- **Averaged Velocity along Gravity Direction (AVG)**: AVG is computed by averaging the instantaneous velocity along the gravity direction at each time

| Statistical Feature | Description |
|---|---|
| Mean | The DC component (average value) of the signal over the window |
| Standard Deviation | Measure of the spreadness of the signal over the window |
| Root Mean Square | The quadratic mean value of the signal over the window |
| Averaged derivatives | The mean value of the first order derivatives of the signal over the window |
| Mean Crossing Rate | The total number of times the signal changes from below average to above average or vice versa normalized by the window length |

**Table 2: Statistical features calculated at the primitive level**

$t$ over the window. The instantaneous velocity at each time $t$ is calculated by numerical integration of the acceleration along the gravity direction.

- **Averaged Rotation Angles related to Gravity Direction (ARATG)**: ARATG captures the rotation movement of the human torso by calculating the cumulative sum of the rotation angles around the gravity direction, normalized by the window length.
- **Averaged Acceleration Energy (AAE)**: AAE is defined as the mean value of the energy over three acceleration axes, where energy is the sum of the squared discrete FFT component magnitudes of the signal from each sensor axis, normalized by the window length. The DC component of the FFT is excluded in this sum since it is already measured by the mean feature.
- **Averaged Rotation Energy (ARE)**: ARE is defined as the mean value of the energy over three gyroscope axes.

It is worth noting that the extraction of physical features is different from statistical features. For statistical features, each feature is extracted from each sensor axis individually. In comparison, most of the physical features are extracted from multiple sensor axes. In other words, sensor fusion is performed at the feature level for physical features.

## 4.3 Primitive Construction

Primitive construction forms the basis of BoF and thus plays an important role in our framework. The extracted primitives are expected to contain salient human motion information and thus could be used to interpret human motion in a more meaningful way. Existing approaches construct motion primitives either using fixed-length windows with identical temporal/spatial duration or through unsupervised clustering. Stiefmeier *et al.* in [6] first recorded the motion trajectory and divided the trajectory into fixed-length windows with identical spatial duration. Motion primitives were then constructed by quantifying all the fixed-length windows based on their trajectory directions calculated in the Cartesian space. Krause *et al.* in [16], Huynh *et al.* in [15], and Ghasemzadeh *et al.* in [8] followed the same procedure as in [6], but using clustering algorithms to group data points with consistent feature values to construct motion primitives. In [16] and [15], authors used $K$-means for clustering. In [8], Gaussian Mixture Model (GMM) was used and was argued to outperform $K$-means by the authors due to its tolerance to cluster overlap and cluster shape variation. In this work, we evaluate both $K$-means and GMM methods.

## 4.4 Vocabulary Size

The result of primitive construction is a motion vocabulary where each generated cluster is treated as a unique motion primitive in the vocabulary. As a result, the vocabulary size is equal to the total number of clusters. Vocabulary Size has a similar effect as the size of window cells mentioned in Section 4.1. Specifically, a small vocabulary may lack discriminative power since two window cells may be assigned into the same cluster even if they are not similar to each other. On the contrary, a large vocabulary is sensitive to noise and thus susceptible to overfitting.

In [15], Huynh *et al.* experimented with vocabulary sizes ranging from 10 to 800. The best vocabulary size was determined based on the classification accuracy. Ghasemzadeh *et al.* in [8] selected the vocabulary size with the best Bayesian Information Criterion (BIC) score. In [16], the best vocabulary size was determined by the guidance of Davies-Bouldin index. In our study, we experiment with vocabularies of 5 to 200 primitives. These vocabulary sizes cover most of the implementation choices in the existing work. The best vocabulary size is determined empirically, similar to our determination of the best window cell size.

## 4.5 Primitive Weighting

Given the motion vocabulary, the next step is to construct the global feature vector to represent activities based on the distribution of the motion primitives. There are many ways to describe the distribution. In this work, we evaluate three weighting schemes that map the distribution of motion primitives to the global feature vectors.

- **Term Weighting**: Term weighting originates from text information retrieval where the counts of occurrences of words in a given text are used as features for text classification tasks. In our case, the local feature vector extracted from each window cell is first mapped to its nearest motion primitive in the feature space. This quantization process generates a primitive histogram which describes the distribution of the motion primitives for each activity. Given the primitive histogram, the feature value of each dimension of the global feature vector is set to the count of the corresponding motion primitive in the histogram.

  Formally, let $\boldsymbol{x}_i$ be the local feature vector associated with the $i^{th}$ window cell of the activity signal $\boldsymbol{x}$, and let $P_j$ denote the $j^{th}$ primitive (cluster) out of $m$ primitives (clusters) in the vocabulary. The term weighting feature mapping $\varphi_{term}$ is defined as

$$\begin{aligned} \varphi_{term}(\boldsymbol{x}) &= [\varphi_1, \ldots, \varphi_m]^T, \\ \text{where } \varphi_j &= \sum_{i \in \boldsymbol{x}} \varphi_j^i, \\ \text{and } \varphi_j^i &= \delta(\boldsymbol{x}_i \in P_j). \end{aligned} \tag{5}$$

- **Binary Weighting**: Binary weighting is similar to term weighting, but with the difference that the feature value of each dimension of the global feature vector is

either 1 or 0. The value 1 indicates the presence of the corresponding motion primitive in the primitive histogram while value 0 indicates the absence. The binary weighting feature mapping $\varphi_{binary}$ is defined as

$$\begin{aligned} \varphi_{binary}(\boldsymbol{x}) &= [\varphi_1, \ldots, \varphi_m]^T, \\ \text{where } \varphi_j &= \bigvee_{i \in \boldsymbol{x}} \varphi_j^i, \\ \text{and } \varphi_j^i &= \delta(\boldsymbol{x}_i \in P_j). \end{aligned} \tag{6}$$

where $\bigvee$ is the logical OR operator.

- **Soft Weighting**: The two weighting schemes described above are directly migrated from the text information retrieval domain. For text, words are discrete and sampled naturally according to language context. For human motion signals in our case, signals are continuous and motion primitives are the outcome of clustering. Based on this difference, although the harsh quantization that associates each window cell with only its nearest cluster shows good performance in the tasks of text analysis and categorization, it may not be optimal for continuous smoothly-varying human motion signals. For example, two window cells assigned to the same motion primitive are not necessarily equally similar to that primitive since their distances to the primitive may be different. Therefore, the significance of motion primitives is weighted more accurately if these distances are taken into consideration. In this work, we propose a soft weighting scheme that takes the distances (similarity) between window cells and motion primitives into account during weight assignment.

  Formally, let $\boldsymbol{c}_j$ denote the $j^{th}$ cluster center (primitive prototype), and let $K(\cdot, \cdot)$ represent the kernel function for similarity measure. The soft weighting feature mapping $\varphi_{soft}$ is defined as

$$\begin{aligned} \varphi_{soft}(\boldsymbol{x}) &= [\varphi_1, \ldots, \varphi_m]^T, \\ \text{where } \varphi_j &= \sum_{i \in \boldsymbol{x}} \varphi_j^i; \varphi_j^i = K(\boldsymbol{x}_i, \boldsymbol{c}_j). \end{aligned} \tag{7}$$

  where $K(\boldsymbol{x}_i, \boldsymbol{c}_j)$ measures the similarity between the $i^{th}$ window cell $\boldsymbol{x}_i$ and cluster center $\boldsymbol{c}_j$. In this work, we use the Laplacian kernel

$$K(\boldsymbol{x}_i, \boldsymbol{c}_j) = \exp(-\frac{||\boldsymbol{x}_i - \boldsymbol{c}_j||}{\sigma_j}) \tag{8}$$

  where $\sigma_j$ is the standard deviation of primitive $P_j$. As a consequence, the feature value of the $j^{th}$ dimension of the global feature vector $\varphi_j$ measures the total similarity of all the window cells of the activity signal $\boldsymbol{x}$ to the primitive prototype $\boldsymbol{c}_j$.

## 4.6 Classifier and Kernels

The choice of classifier is critical to the recognition performance. Since the size of the motion vocabulary can be potentially large, in this work, we choose Support Vector Machines (SVMs) to be our learning machine. They have proved to be very effective in handling high dimensional data in a wide range of machine learning and pattern recognition applications [19].

SVM aims to maximize the margin between different classes, where margin is defined as the distance between the decision boundary and the nearest training instances. These instances, called support vectors, finally define the classification functions [20]. Mathematically, for a two-class classification scenario, given a training set of instance-label pairs $(\boldsymbol{x}_i, y_i), i = 1, \ldots, l$ where $\boldsymbol{x}_i \in R^n$ represents the $n$-dimensional feature vector and $y_i \in \{1, -1\}$ represents the class label, the support vector machines require the solution of the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} + C \sum_{i=1}^{l} \xi_i \\ &\text{subject to: } y_i \left( \boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b \right) \geq 1 - \xi_i, \\ &\xi_i \geq 0, i = 1, \ldots, l \end{aligned} \tag{9}$$

where $\phi$ is a function that maps training instance $\boldsymbol{x}_i$ into a higher (maybe infinite) dimensional space; $\xi_i$ are called slack variables, which measure the degree of misclassification; and $C > 0$ is the soft-margin constant acting as a regularization parameter to control the tradeoff between training error minimization and margin maximization.

To enable efficient computation in high-dimensional feature space, a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ is defined. The choice of the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is critical for statistical learning. Although a number of general purpose kernels have been proposed, it is unclear which one is the most effective for BoF in the context of human activity classification. In this work, we evaluate the following two kernels which are all Mercer kernels [20].

- **Linear kernel**:

$$K_{linear}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{10}$$

- **Gaussian RBF kernel**:

$$K_{Gaussian}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2), \gamma > 0 \tag{11}$$

# 5. EVALUATION

In this section, we evaluate the effectiveness of our BoF-based framework. We divide the dataset into training set and test set. Since each participant performs five trials for each activity, we use three trials from each participant as training set to build activity models. Three-fold cross validation is used to determine the corresponding parameters. The vocabulary of motion primitives is learned from half of the training set. The remaining two trials from each participant are used as test set. A confusion table is built from the test set to illustrates the performance of the framework.

## 5.1 Impact of Window Cell Sizes

Our first experiment aims to evaluate the effect of different window cell sizes on the classification performance. In this experiment, we use the statistical feature set, $K$-means for primitive construction, term weighting and linear kernel for SVM training. Figure 4 shows the average misclassification rates as a function of window cell sizes 0.1, 0.2, 0.3, 0.4, 0.5, 0.8, 1, 1.5, and 2 seconds. Each line represents one vocabulary size. As shown in the figure, vocabulary size 5 has the worst performance across all window cell sizes. This indicates that using only 5 motion primitives is not sufficient to differentiate nine activities. In comparison, for other three vocabulary sizes, the performances are 30% better on average, with the misclassification rates ranging from 12.4% to 19.8% across all window cell sizes. If we look at each case

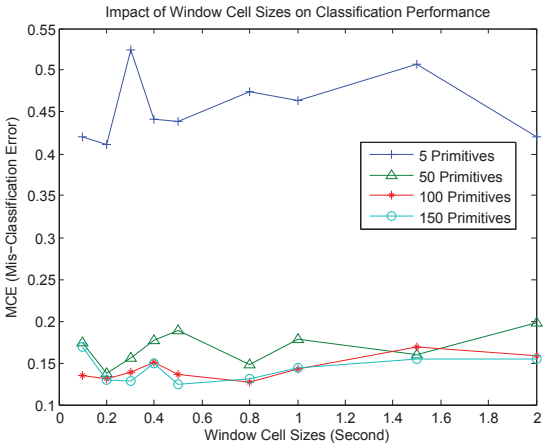


**Figure 4: Impact of Window Cell Sizes**

individually, vocabulary size 50 reaches its minimum misclassification rate at 0.2 second window cell size, and the rate starts rising as the size increases. For vocabulary size 100 and 150, the misclassification rates reach the first local minimum at 0.2 second, and only vary slightly when the window cell size is less than 0.8 second. The performances start degrading when the size is beyond 1 second. Based on these observations, we conclude that the appropriate window cell size is around 0.2 second. Therefore, we only use 0.2 second window cell in the remaining experiments.

## 5.2 Impact of Vocabulary Sizes

In this experiment, we study the impact of different vocabulary sizes on the classification performance of our BoF framework. We fix the window size to 0.2 second and keep other factors the same as in the last experiment. Figure 5 shows the average misclassification rates as a function of vocabulary sizes 5, 10, 25, 50, 75, 100, 125, 150, 175, and 200. The error bars represent the standard deviation in the cross validation. As illustrated in the figure, the misclassification rate drops significantly from vocabulary size 5 and stabilizes starting from vocabulary size 75. The misclassification rate reaches the minimum of 12.0% (88.0% accuracy) when

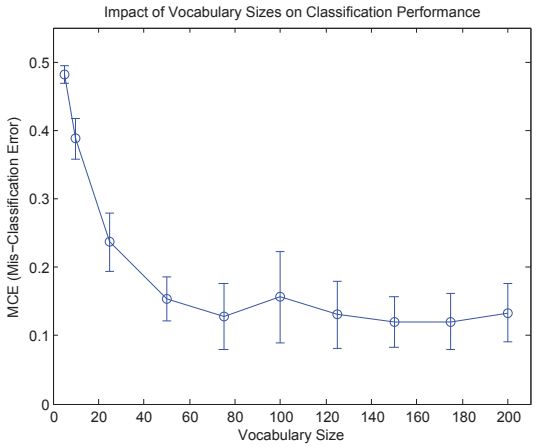


**Figure 5: Impact of Vocabulary Sizes**

150 motion primitives are used. When the number of motion primitives is bigger than 150, the misclassification rate increases slightly. This indicates that a vocabulary of 150 primitives is sufficient for our activity set. Another interesting observation when combining the results in Figure 4 and Figure 5 is that vocabulary size has a more significant impact on the performance than the size of window cell.

## 5.3 Comparison of Features

Next, we examine the effects of features. Specifically, we use the statistical feature set and physical feature set described in the previous section and keep other factors the same to construct motion primitives and build activity models respectively. The results are shown in Figure 6. Similar

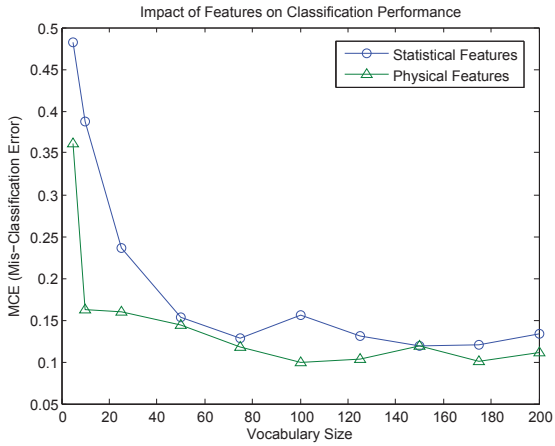


**Figure 6: Comparison of Features**

to the statistical features, the misclassification rate based on the physical features drops significantly from vocabulary size 5 and stabilizes starting from vocabulary size 75. The misclassification rate reaches its minimum of 9.9% (90.1% accuracy) when 100 motion primitives are used.

In addition, it is interesting to observe that the physical features outperform the statistical features consistently across all vocabulary sizes, with an improvement of 5.7% for the same vocabulary size on average. This indicates that primitives constructed from physical features contain more salient and meaningful motion information compared to the primitives constructed from statistical features. In order to validate this argument and have a better understanding of why the physical features perform better, we map the primitives constructed by statistical features and physical features onto the original sensor signals respectively. Figure 7 shows the primitive mappings based on the physical features (top) and the statistical features (bottom) on the same sensor signal (*running* in this example). For illustration purposes, a total of five motion primitives are used, with different colors representing different primitives. As illustrated, different feature sets lead to different primitive mappings. For statistical features, it is obvious that primitives are constructed based on the signal's statistical characteristics. For example, the primitive in red corresponds to the data points which have mid-range raw values and a positive derivative. The primitive in blue corresponds to the data points which have high raw values and a small derivative. In comparison, primitives constructed based on the

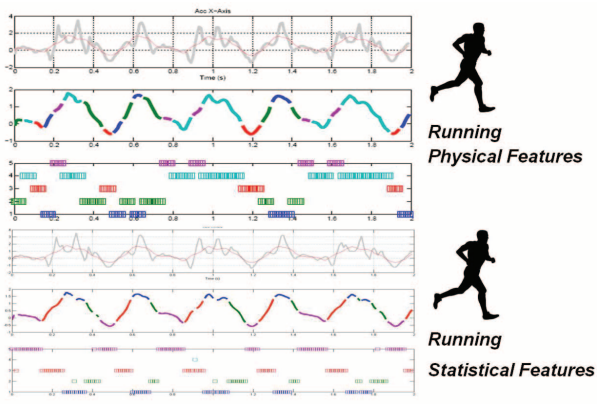


**Figure 7: The difference of primitive mapping between physical features (top) and statistical features (bottom)**

physical features contain useful physical meanings that help discriminate different activities. For example, the primitive in cyan illustrated in Figure 7 only occurs in half cycle. This primitive may be a very important primitive for describing the motion of the subject's left/right hip (the subject wears the sensing device at this location (see Figure 2)) during running. Since physical features outperform statistical features consistently across all vocabulary sizes, only physical features will be used in the remaining experiments.

## 5.4 Comparison of Primitive Construction Algorithms

This section compares the performance of the two primitive construction algorithms: $K$-means and Gaussian Mixture Model (GMM). As shown in Figure 8, for GMM, the misclassification rate drops significantly from vocabulary size 5. The misclassification rate reaches the minimum of 18.5% (81.5% accuracy) when 150 motion primitives are used. Compared to GMM, it is interesting to see that $K$-means achieves better performance consistently across all vocabulary sizes, with an improvement of 13.5% for the same vocabulary size on average. Our result contradicts the arguments

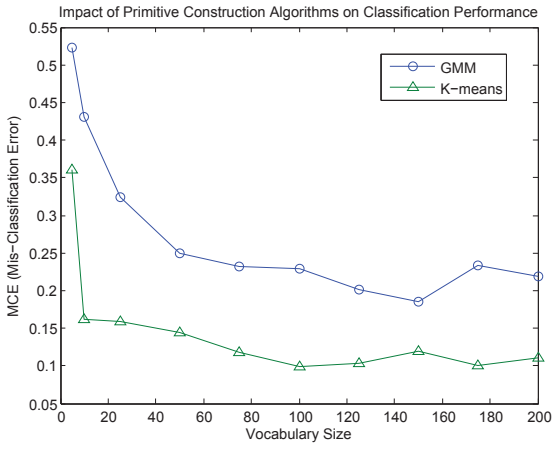


**Figure 8: Comparison of Primitive Construction Algorithms**

of the authors in [8], indicating that $K$-means is powerful to handle cluster overlap and shape variations of human motion data as long as the number of clusters is sufficient.

## 5.5 Comparison of Weighting Schemes

Figure 9 illustrates the performance differences between three primitive weighting schemes. We first examine the relationship between binary weighting and term weighting. In both cases, the misclassification rates drop and then stabilize as the size of vocabulary increases in general. The difference between these two cases is that term weighting outperforms binary weighting by a large margin when the vocabulary size is small and by a small margin when the vocabulary size becomes large. This is because that, with a larger vocabulary size, the counts of a large number of motion primitives are either 0 or 1, which makes term weighting and binary weighting similar. Next, we see that the Laplacian kernel-based soft weighting scheme outperforms both term weighting and binary weighting across all vocabulary sizes except vocabulary size 10. In particular, soft weighting achieves the minimum misclassification rate at 7% (93% accuracy) when 125 motion primitives are used. This result indicates that, different from words in text information retrieval which are discrete, motion primitives extracted from continuous human motion signals are smooth, and taking the smoothness into account is significant to the classification performance of the BoF framework.

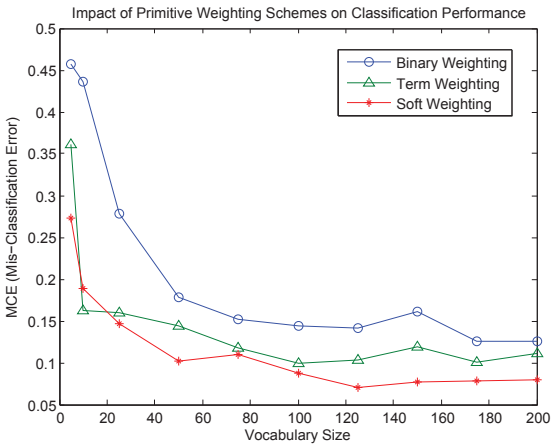


**Figure 9: Comparison of Weighting Schemes**

## 5.6 Comparison of Kernel Functions

In this experiment, we examine the performance of BoF framework when two different kernel functions are used. The results are shown in Figure 10. As illustrated, neither kernel predominates everywhere, but linear kernel is preferred when the vocabulary size is equal or larger than 100. This observation can be attributed to the fact that motion primitives are linear separable when the dimension of primitive space is high.

## 5.7 Confusion Table

The experimental results presented in the previous subsections demonstrate that all six factors are influential to the final classification performance of our BoF-based framework. Here, we investigate the best possible choices of the six fac-

| Ground Truth \ Classified Activity | Walk forward | Walk left | Walk right | Go up stairs | Go down stairs | Run forward | Jump up | Sit on a chair | Stand | Total | Recall |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 Walk forward | 126 | 6 | 7 | 2 | 1 | 0 | 0 | 0 | 0 | 142 | 88.7% |
| 2 Walk left | 7 | 159 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 166 | 95.8% |
| 3 Walk right | 9 | 0 | 190 | 2 | 0 | 0 | 0 | 1 | 0 | 202 | 94.1% |
| 4 Go up stairs | 3 | 0 | 0 | 27 | 1 | 0 | 0 | 0 | 0 | 31 | 87.1% |
| 5 Go down stairs | 2 | 1 | 0 | 0 | 26 | 1 | 0 | 0 | 0 | 30 | 86.7% |
| 6 Run forward | 0 | 0 | 0 | 0 | 0 | 93 | 0 | 0 | 0 | 93 | 100% |
| 7 Jump up | 0 | 0 | 0 | 0 | 0 | 0 | 54 | 1 | 0 | 55 | 98.2% |
| 8 Sit on a chair | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 169 | 9 | 178 | 94.9% |
| 9 Stand | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 22 | 134 | 158 | 84.8% |
| Total | 147 | 167 | 198 | 31 | 28 | 94 | 54 | 193 | 143 | | |
| Precision | 85.7% | 95.2% | 96.0% | 87.1% | 92.9% | 98.9% | 100% | 87.6% | 93.7% | | |

**Table 3: Confusion table for the best factor combination when using 0.2 second window cell, physical feature set, vocabulary size = 125, $K$-means for primitive construction, soft weighting for motion primitive assignment, and linear kernel for SVM training. The entry in the $i^{th}$ row and $j^{th}$ column is the count of activity instances from class $i$ but classified as class $j$. Overall classification accuracy is 92.7%.**

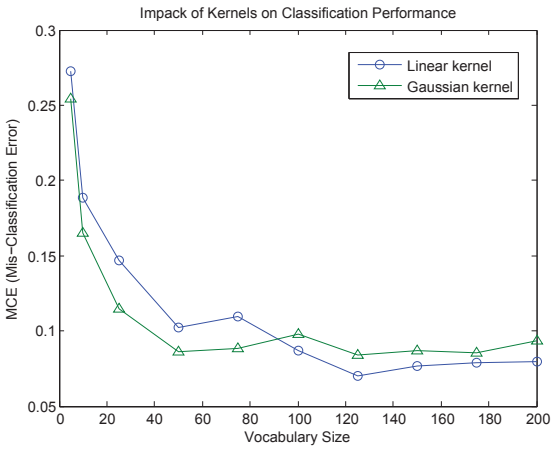


**Figure 10: Comparison of Kernel Functions**

tors with the goal of exploring the upper limit of the performance of the BoF framework for human activity recognition. Based on the results presented earlier, we determine the best combination of factors to be 0.2 second window cell, physical feature set, a vocabulary with 125 motion primitives, $K$-means for primitive construction, Laplacian kernel-based soft weighting for motion primitive assignment, and linear kernel for SVM training. To evaluate the performance of the BoF-based framework with the best combination of factors, a confusion table is built from the test set and is shown in Table 3. The overall recognition accuracy across all activities is 92.7%. If we examine the recognition performance of each activity individually, *jump up* and *run forward* are the two easiest activity classes to recognize. *go upstairs* and *go downstairs* have relatively low recall values since they can be confused with other walking-related activities. *stand* has the lowest recall value because it is often confused with *sit on a chair*. This result makes sense since both *stand* and *sit on a chair* are static activities, and we expect difficulty in differentiating different static activity classes especially when the sensing device is attached to the hip of the subjects. Finally, for *walk forward*, *walk left* and *walk right* are the two dominant activity classes which *walk forward* is misclassified into. However, *walk left* and *walk right* never get confused with each other.

## 5.8 Comparison with String-Matching

As our last experiment, we conduct a comparative evaluation with the non-statistical string-matching-based approach. We implement the string-matching method described in [8]. We select this method because the authors in [8] also use a clustering algorithm to construct motion primitives. To make a fair comparison, we use a 0.2 second window cell with statistical features and $K$-means primitive construction algorithm for both BoF and string-matching. The results are shown in Figure 11. As illustrated in the figure, the average misclassification rate of the string-matching-based approach ranges from 37% to 54% across all vocabulary sizes. In addition, there is no clear trend of the misclassification rate as the vocabulary size varies. This indicates that the string-matching-based approach is not stable such that it is extremely difficult to determine a meaningful vocabulary size. Moreover, as expected, the string-matching-based approach performs consistently worse compared to BoF by a large margin across all vocabulary sizes. As explained in the first section, this is because extracting meaningful string templates for the string-matching-based approach is difficult when the activity data is noisy and has a large intra-class variation.

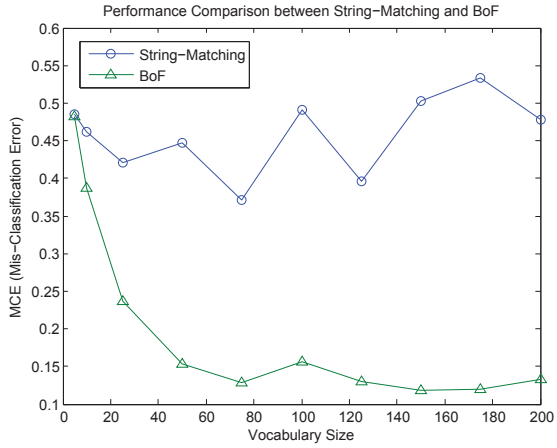


**Figure 11: Performance Comparison with String-Matching-Based Approach**

## 6. CONCLUSION AND FUTURE WORK

In this paper, we have investigated the feasibility of using a Bag-of-Features (BoF)-based framework for human activity representation and recognition. The benefit of this framework is that it is able to identify general motion primitives which act as the basic building blocks for modeling different human activities. We have studied six key factors which govern the performance of the BoF-based framework. Our experimental results validate the effectiveness of this framework and show that all the six factors are influential to the classification performance of the BoF framework. As a summary, our BoF framework achieves a 92.7% overall classification accuracy with a 0.2 second window cell and a vocabulary of 125 motion primitives constructed based on physical features using $K$-means clustering and soft weighting. This result is 32.3% higher than the corresponding non-statistical string-matching-based approach.

Since we assume that activity signals do not follow any grammar, our baseline BoF framework is totally based on the primitive distribution. Therefore, it is interesting to explore whether using the temporal order of motion primitives in addition to BoF is beneficial. Hidden Markov Model (HMM) and Conditional Random Field (CRF) are two powerful models to capture temporal correlations of the sequential signals. Although HMM and CRF models have been applied to the activity recognition problem in recent years, little work has done on building HMM/CRF on top of BoF. We will work along this direction as our future work.

## 7. REFERENCES


[1] Tâm Huynh and Bernt Schiele. Analyzing features for activity recognition. In *Joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies (sOc-EUSAI)*, pages 159–163, Grenoble, France, 2005.

[2] Hassan Ghasemzadeh, Jaime Barnes, Eric Guenterberg, and Roozbeh Jafari. A phonological expression for physical movement monitoring in body sensor networks. In *IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 58–68, Atlanta, Georgia, USA, 2008.

[3] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

[4] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, November 2001.

[5] Preben Fihl, Michael B. Holte, Thomas B. Moeslund, and Lars Reng. Action recognition using motion primitives and probabilistic edit distance. In *International Conference on Articulated Motion and Deformable Objects (AMDO)*, pages 375–384, Andratx, Mallorca, Spain, 2006.

[6] Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. Gestures are strings: efficient online gesture spotting and classification using string matching. In *International Conference on Body Area Networks (BodyNets)*, pages 16:1–16:8, Florence, Italy, 2007.

[7] Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. Fusion of string-matched templates for continuous activity recognition. In *IEEE International Symposium on Wearable Computers (ISWC)*, pages 1–4, Boston, MA, USA, 2007.

[8] Hassan Ghasemzadeh, Vitali Loseu, and Roozbeh Jafari. Collaborative signal processing for action recognition in body sensor networks: a distributed classification algorithm using motion transcripts. In *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 244–255, Stockholm, Sweden, 2010.

[9] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[10] Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73:213–238, June 2007.

[11] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*, pages 1–17, Linz/Vienna, Austria, 2004.

[12] Nishkam Ravi, Nikhil Dandekar, Prreetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 1541–1546, Pittsburgh, Pennsylvania, USA, 2005.

[13] Abdullah Mueen, Eamonn Keogh, and Neal Young. Logical-shapelets: an expressive primitive for time series classification. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1154–1162, San Diego, California, USA, 2011.

[14] http://www.motionnode.com.

[15] Tâm Huỳnh, Ulf Blanke, and Bernt Schiele. Scalable recognition of daily activities with wearable sensors. In *International conference on Location-and context-awareness (LoCA)*, pages 50–67, Oberpfaffenhofen, Germany, 2007.

[16] Andreas Krause, Daniel P. Siewiorek, Asim Smailagic, and Jonny Farringdon. Unsupervised, dynamic identification of physiological and activity context in wearable computing. In *IEEE International Symposium Wearable Computers (ISWC)*, pages 88–97, White Plains, NY, USA, 2003.

[17] Mi Zhang and Alexander A. Sawchuk. A feature selection-based framework for human activity recognition using wearable multimodal sensors. In *International Conference on Body Area Networks (BodyNets)*, Beijing, China, November 2011.

[18] Dean M. Karantonis, Michael R. Narayanan, Merryn Mathie, Nigel H. Lovell, and Branko G. Celler. Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):156–167, 2006.

[19] Chih-Chung Chang Chih-Wei Hsu and Chih-Jen Lin. A practical guide to support vector classification. *Bioinformatics*, 1(1):1–16, 2010.

[20] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.