A Bag-of-Features-Based Framework for Human Activity Representation and Recognition

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

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

ABSTRACT
Human activity recognition using wearable sensors is an important topic in ubiquitous computing. 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. Experimental results validate the effectiveness of this framework for the task of human activity recognition. In addition, we have demonstrated that our statistical BoF framework can achieve a much better performance compared to the non-statistical string-matching-based approach.

Author Keywords
Activity Representation and Recognition, Motion Primitives, Bag-of-Features, Machine Learning, Wearable Sensors

ACM Classification Keywords
I.5.4 Pattern Recognition: Applications; J.3 Computer Applications: Life and Medical Sciences

General Terms
Algorithms, Design, Experimentation, Performance

INTRODUCTION
Human activity recognition is regarded as one of the most important problems in ubiquitous computing since it has a wide range of potential applications, including long term physical fitness monitoring, sleep quality monitoring, and intelligent assistance for people with cognitive disorders [1]. With the advancement of semiconductor and MEMS technologies, inertial sensors such as accelerometers and gyroscopes are miniaturized such that these sensors could be attached or worn on the human body in an unobtrusive way. As a result, it is possible to use these wearable sensors to build systems and recognize human activities.

SAGAware'11, September 18, 2011, Beijing, China.

Conventional 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. This information is then transformed into a feature vector which is used as input to the classifier for classification. Although this "whole-motion" model has been proved to be very effective in existing studies, the performance is highly dependent on the window length [2]. As a possible solution to this problem, motion primitive-based models were proposed and have recently attracted numerous research attention.

The idea of motion primitives stems from human speech due to the similarity between human motion and speech signals [3]. In speech recognition, sentences are first divided into isolated words. Each word is further divided into a sequence of phonemes. In English, there are about 50 phonemes shared by all the English words. Models are first built for each of these phonemes. These phoneme models then act as the building blocks to build words and sentences [4]. Following the same idea, in motion primitive-based activity 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 each activity class, motion primitives capture the invariance aspects of the local features.

The keys to the success of 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 according 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) with five motion primitives. For both activities, the first line shows the original sensor signal and the second line shows the primitive mapping of the original signal. Below these are five lines showing the locations of the five motion primitives (labeled A, B, C, D, E in different colors). Below this is a sample of the symbol string. To build activity models based on these extracted primitives, one common strategy is to adopt a string-matching-based approach.

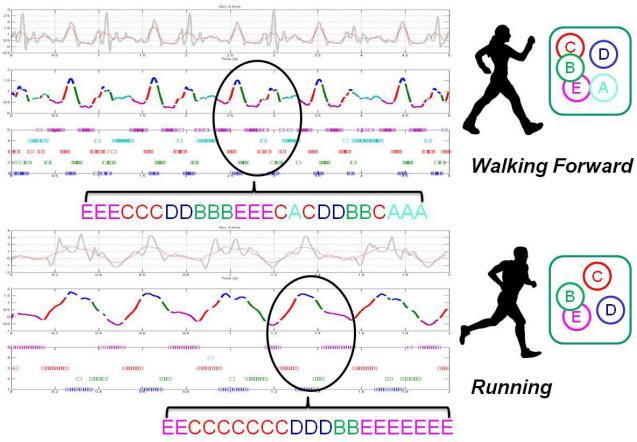


Figure 1. An example of activity representation using motion primitives. For illustration purpose, a total of 5 motion primitives are used (A, B, C, D, E). In this example, *walking forward* contains 5 types of motion primitives (A, B, C, D, E) while *running* contains 4 (B, C, D, E).

Specifically, at 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 of 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) [5]. At the recognition stage, the testing instance is first transformed into the primitive string, and then classified to the activity class whose template matches the testing instance the best. Although this string-matching-based strategy shows competitive performance in both vision-based and wearable sensor-based activity recognition tasks [6] [7] [8] [9], the main drawback is its high sensitivity to noise and its poor performance in the presence of high intra-class variation [10]. Under such conditions, it is extremely difficult to extract a meaningful template. 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 widely applied in text document classification, texture and object recognition tasks and demonstrated impressive performance [11]. The goals of this work are 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.

The rest of this paper is organized as follows. We first give a brief survey of some existing work on human activity recognition. Then we introduce the sensing platform and dataset we used for this study. Next, we describe the basic idea of BoF and outlines the key components of the BoF framework for human activity representation and recognition. Finally, we present the evaluation results and conclude the paper.

RELATED WORK

Based on the granularity level the activities are modeled, existing activity recognition methods can be broadly classified into two categories: "whole-motion"-based methods and motion primitive-based methods. For the "whole-motion" model, different combinations of features and classifiers have been extensively studied on different sets of activities. In [12], 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 [13] used similar features as in [12]. 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 methods have received numerous research attention due to their capability of capturing local characteristics of activities. In [7], 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. Ghasemzadeh et al. in [9] followed the same idea but used clustering technique to group data points with consistent feature values to construct motion primitives. As a further extension, Fihl et al. in [6] replaced the standard deterministic string-matching algorithm with a probabilistic-based string-matching strategy by using probabilistic edit distance instead of the standard edit distance.

In this work, we follow the basic principles of the motion primitive-based model. Different from the existing approaches, instead of formulating activity recognition as a string-matching problem, we leverage the power of statistical learning machines, with the hope that the statistical approach could remedy the drawbacks of string-matching-based approach and thus make the recognition system more robust.

SENSING PLATFORM AND DATASET

For this work, data is recorded using an off-the-shelf multimodal sensing platform called MotionNode [14]. MotionNode is a 6-DOF inertial measurement unit 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.

To collect data, six subjects with different gender, age, height, and weight are selected to perform nine types of activities: walk forward, walk left, walk right, go upstairs, go downstairs, jump up, run, stand, and sit. 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 ob-

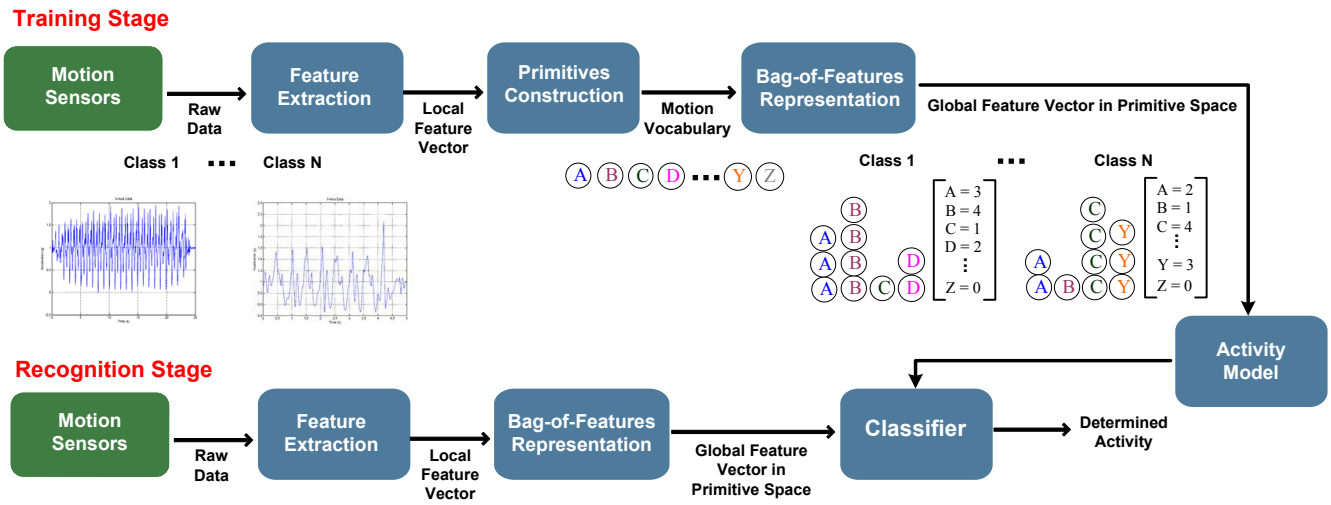
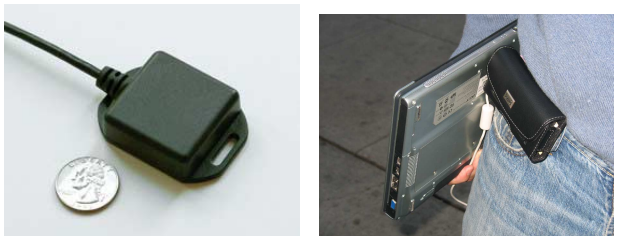


Figure 2. Flowchart of Bag-of-Features (BoF)-based framework for human activity representation and recognition

trusiveness of the sensing device, a single MotionNode is packed into a mobile phone pouch and attached to the subject's right front hip (Figure 3). Each subject performs five trials for each activity on different days at various indoor and outdoor locations without supervision. On average, we have about 20 minutes of data for each type of activity.



(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 3. MotionNode sensor and its placement during data collection

THE BAG-OF-FEATURES FRAMEWORK

Figure 2 gives a graphical illustration of the BoF-based framework for human activity representation and recognition. At 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. Each generated cluster is treated as a motion primitive in the vocabulary. By mapping the window cells to the motion primitives in the vocabulary, the activity signal is 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. At 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 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 give more details on the key components of our BoF framework described above.

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 [2]. 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. 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 extremely good performance at and beyond such scales in many existing studies.

Features

In activity recognition, a variety of features both in time and frequency domains have been investigated within the framework of the "whole-motion" model. Popular examples are mean, variance, entropy, correlation, FFT coefficients etc. However, at primitive level, since the total number of samples within each window cell is much smaller, complex features such as entropy and FFT coefficients may not be reliably calculated. Therefore, we only consider features that

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 1. Features calculated at primitive level

can be reliably calculated at primitive level. Table 1 lists the features we include in this work. These features are extracted from each axis of both accelerometer and gyroscope.

Primitive Construction and Vocabulary Size

Primitive construction forms the basis of BoF and thus plays an important role in our framework. In this work, K-means clustering is used to group local feature vectors to construct motion primitives. Each generated cluster is treated as a unique motion primitive in the vocabulary. Thus, the vocabulary size is equal to the total number of clusters. The best vocabulary size is determined empirically, similar to our determination of the best window cell size.

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, term weighting is applied. 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. 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 out of m primitives in the vocabulary. The term weighting feature mapping φ_{term} is defined as

$$\begin{aligned} \varphi_{term}(\boldsymbol{x}) &= [\varphi_1, \ldots, \varphi_m]^T, \\ \text{where } \varphi_j &= \sum_{i \in \boldsymbol{x}} \varphi_j^i, \\ \text{and } \varphi_j^i &= \delta(\boldsymbol{x}_i \in P_j). \end{aligned} \tag{1}$$

Classifier

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 the multi-class Support Vector Machines (SVMs) with a linear kernel 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 [15].

EVALUATION

To evaluate the effectiveness of our BoF framework, we adopt the leave-one-trial-out cross validation strategy. Specifically, since each subject performs five trials for each activity, we use four trials from all six subjects as training examples to build activity models. Data from the left-out trial is used for testing. This process iterates for every trial. The final result is the average value across all five trials.

Impact of Window Cell Sizes

Our first experiment aims to evaluate the effect of different window cell sizes on the classification performance. 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

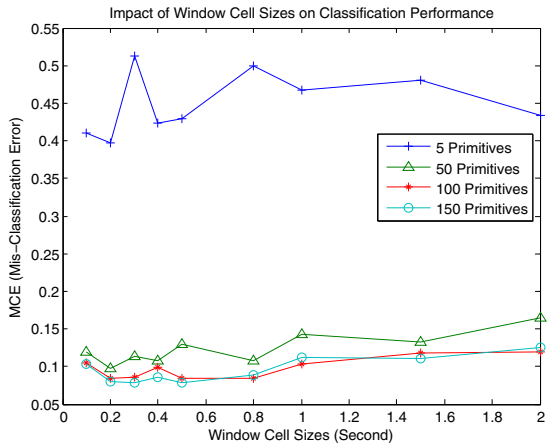


Figure 4. Impact of Window Cell Sizes on Classification Performance

shown in the figure, vocabulary size 5 has the worst performance across all window sizes. This indicates that using only 5 motion primitives are not sufficient to differentiate nine activities. In comparison, for other three vocabulary sizes, the performances are 33% better on average, with the misclassification rates ranging from 7.8% to 16.5% across all window sizes. If we look at each case individually, vocabulary size 50 reaches its minimum misclassification rate at 0.2 second window size, and the rate starts rising as window 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 size is less than 0.8 second. The performances start degrading when the window size is beyond 1 second. Based on these observations, the appropriate window size is around 0.2 second.

Impact of Vocabulary Sizes

In this experiment, we study the impact of different vocabulary sizes on our BoF framework. Based on the results in the last experiment, we fix the window size to 0.2 second. 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 show the standard deviation across five trials in the cross validation testing process. As illustrated, as

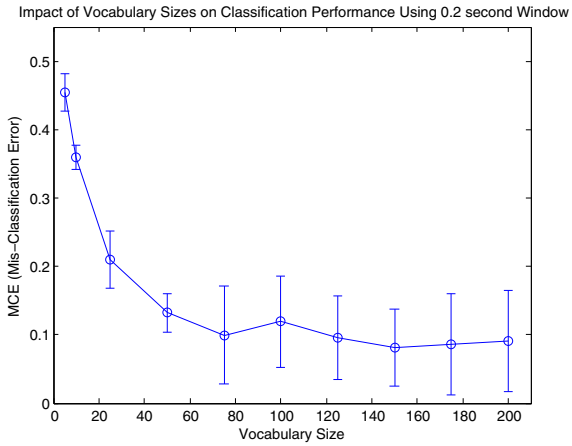


Figure 5. Impact of Vocabulary Sizes on Classification Performance

a general trend, the misclassification rate drops as the size of vocabulary increases, with the minimum of 8.1% (91.9% accuracy) when 150 primitives are used. When the number of primitives is beyond 150, the misclassification rate increases slightly. This indicates that a vocabulary of 150 primitives is sufficient for our activity set.

Performance Comparison with String-Matching

In our last experiment, we conduct a comparative evaluation with the string-matching-based approach. We implement the string-matching method described in [9]. We select it since the authors in [9] also use a clustering algorithm to construct motion primitives. To make a fair comparison, we also use a 0.2 second window cell for string-matching. The results are shown in Table 2. As expected, the string-matching approach performs worse than BoF across all vocabulary sizes. This is because extracting meaningful templates is difficult when the activity data has a high intra-class variation.

Vocabulary Size	5	25	50	75	100	125	150	175
Error Rate	48.6%	42.1%	44.7%	37.1%	49.1%	39.6%	50.3%	53.4%

Table 2. Classification performance of string-matching

CONCLUSION AND FUTURE WORK

In this paper, we have studied the feasibility of applying a Bag-of-Features (BoF)-based framework for human activity representation and recognition. Our experimental results validate the effectiveness of this approach. As a conclusion, our BoF-based framework achieves a 91.9% accuracy with a 0.2 second window cell and a vocabulary of 150 primitives. This result is 42% higher than the corresponding string-matching-based approach. Since the baseline BoF is totally based on primitive distribution, for our future work, we will explore whether using the temporal order of motion primitives in addition to BoF is beneficial.

REFERENCES

1. T. Choudhury et al. The Mobile Sensing Platform: An Embedded Activity Recognition System. *Pervasive Computing*, 7(2):32–41, April 2008.
2. T. Huỳnh et al. Analyzing features for activity recognition. In *sOc-EUSAI*, New York, USA, 2005.
3. H. Ghasemzadeh et al. A phonological expression for physical movement monitoring in body sensor networks. In *MASS*, pages 58–68, Atlanta, Georgia, USA, 2008.
4. X. Huang et al. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, NJ, USA, 1st edition, 2001.
5. R. Duda et al. *Pattern Classification*. Wiley-Interscience, 2nd edition, November 2001.
6. P. Fihl et al. Action recognition using motion primitives and probabilistic edit distance. In *AMDO*, pages 375–384, Andratx, Mallorca, Spain, 2006.
7. T. Stiefmeier et al. Gestures are strings: efficient online gesture spotting and classification using string matching. In *BodyNets*, pages 16:1–16:8, Florence, Italy, 2007.
8. T. Stiefmeier et al. Fusion of string-matched templates for continuous activity recognition. In *ISWC*, pages 1–4, Washington, DC, USA, 2007.
9. H. Ghasemzadeh et al. Collaborative signal processing for action recognition in body sensor networks: a distributed classification algorithm using motion transcripts. In *IPSN*, pages 244–255, New York, NY, USA, 2010.
10. A. Jain et al. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
11. J. Zhang et al. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vision*, 73:213–238, June 2007.
12. L. Bao et al. Activity recognition from user-annotated acceleration data. In *Pervasive Computing*, pages 1–17, Linz/Vienna, Austria, 2004.
13. N. Ravi et al. Activity recognition from accelerometer data. In *IAAI*, pages 1541–1546, Pittsburgh, Pennsylvania, USA, 2005.
14. http://www.motionnode.com.
15. C. Hsu et al. A practical guide to support vector classification. *Bioinformatics*, 1(1):1–16, 2010.