Recognizing Multiuser Activities Using Wireless Body Sensor Networks

Tao Gu, *Member*, *IEEE*, Liang Wang, *Student Member*, *IEEE*,
Hanhua Chen, *Member*, *IEEE*, Xianping Tao, *Member*, *IEEE*, and Jian Lu

Abstract—The advances of wireless networking and sensor technology open up an interesting opportunity to infer human activities in a smart home environment. Existing work in this paradigm focuses mainly on recognizing activities of single user. In this work, we focus on the fundamental problem of recognizing activities of multiple users using a wireless body sensor network, and propose a scalable pattern mining approach to recognize both single- and multiuser activities in a unified framework. We exploit Emerging Pattern—a discriminative knowledge pattern which describes significant changes among activity classes of data—for building activity models and design a scalable, noise-resistant, Emerging Pattern-based Multiuser Activity Recognizer (epMAR) to recognize both single- and multiuser activities. We develop a multimodal, wireless body sensor network for collecting real-world traces in a smart home environment, and conduct comprehensive empirical studies to evaluate our system. Results show that epMAR outperforms existing schemes in terms of accuracy, scalability, and robustness.

Index Terms—Wireless body sensor networks, sensor-based activity recognition, pattern mining.

1 Introduction

Wireless sensor networks have received significant attention in supporting a variety of applications such as surveillance [1], habitat monitoring [2], and infrastructure protection [3]. One of the emerging applications in recent years is understanding and recognizing human activities using body sensor networks [4], [5], [6]. Different from vision-based activity recognition, in this paradigm, a number of on-body wireless sensors are typically deployed to collect the observations of a human user and the living environment. These sensors are able to capture many useful, fine-grained observations such as human motion, human-to-object interaction, and human-to-human interaction which are not possible, if not difficult, to capture by video camera. These observations in the form of a continuous sensor data stream are used to train an activity model; the trained model can then be used to classify activities with new observations. Such system has many potential real-life applications in medical care, assistive living, entertainment, and logistics support, for example, monitoring activities for the elderly and cognitively impaired persons and providing proactive assistance [7], personal healthcare [8], and predicting transportation modes [9].

Most of the existing work on activity recognition focus on single-user activities performed by one particular user. However, there are typically multiple inhabitants in a living space, and they often perform specific tasks together. Activities that involve multiple users collaboratively or concurrently (a.k.a. multiuser activities) are common in our daily lives, especially in a home setting. A system capable of recognizing multiuser activities has a practical implication for real-world applications. Recognizing multiuser activities using on-body sensors is more challenging than recognizing single-user activities. First, user interactions often occur when two or more users perform an activity together. Such interactions can be, for example, voice conversation, handling an object together, and passing objects from one user to another. Capturing these interactions requires elaborate sensors and network design. Importantly, how to model interaction processes and perform inferences are critical to system performance. Second, processing sensor data stream generated from sensors is costly since such data stream usually contains a large volume of continuous sensor readings. Processing multiple data streams corresponding to multiple users implies a scalability issue. How to scale the activity model to the number of users is important to a practical sensor-based recognition system. The scalability problem of multiuser activity recognition remains unsolved in the literature [10]. Third, sensor data are inherently noisy which may affect system robustness seriously. The RF interference and sensor malfunction often cause errors in sensor readings. Moreover, in a multiuser

- *T. Gu is with the Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, Odense M DK-5230, Denmark. E-mail: gu@imada.sdu.dk.*
- *L. Wang is with the State Key Laboratory for Novel Software Technology, Department of Computer Science, Nanjing University, 22 Hankou Road, Nanjing 210093, China, and also with the Department of Mathematics and Computer Science, University of Southern Denmark. E-mail: wl@smail.nju.edu.cn.*
- *H. Chen is with the Services Computing Technology and System Laboratory, Cluster and Grid Computing Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the Department of Mathematics and Computer Science, University of Southern Denmark. E-mail: chenhanhua@hust.edu.cn.*
- *X. Tao and J. Lu are with the State Key Laboratory for Novel Software Technology, Department of Computer Science, Nanjing University, 22 Hankou Road, Nanjing 210093, China. E-mail: txp@ics.nju.edu.cn, lj@nju.edu.cn.*

Manuscript received 10 Apr. 2010; revised 3 Oct. 2010; accepted 2 Dec. 2010; published online 2 Mar. 2011.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2010-04-0167. Digital Object Identifier no. 10.1109/TMC.2011.43.

scenario, the data stream collected from one user may contain background noise. For example, audio information generated from other users or the environment may affect system performance seriously. The ability of noise resistance plays a more important role in multiuser activity recognition than single-user activity recognition.

Encouraged by the recent success of sensor-based activity recognition for single user, in this paper, we propose to use a body sensor network for multiuser activity recognition. To address the above mentioned challenges, we investigate this problem from two aspects—sensor platform and recognition model. We first design a wireless body sensor network to capture the observations of each user and the interactions among multiple users. Based on this platform, we propose a novel pattern-based activity model to capture unique, discriminative features from diverse activities. In addition to the single-user model, the multiuser model is able to capture user interactions. Both activity models leverage on Emerging Pattern (EP) [11]—a discriminative knowledge pattern that describes significant differences between two classes of data. We mine a set of EPs for each activity class from the trace collected from the body sensor network we build, and use the sets of EPs as powerful discriminators to differentiate activity classes. Since EPs capture significant differences, random noise can be easily eliminated. To address the scalability issue, we propose a proximity-based filtering technique to minimize the processing cost of each data stream. We conduct comprehensive experiments using the real-world activity trace we collected to evaluate this design. Results show that our design achieves an overall accuracy of 89.72 percent, outperforming existing schemes.

In summary, the main contributions of this work are three folds:

- To the best of our knowledge, this work is the first formal study of using a body sensor network for multiuser activity recognition in a smart home environment.
- We propose a novel pattern-based activity model and design epMAR—a scalable, noise-resistant, pattern-based algorithm—to recognize both single- and multiuser activities.
- We develop a prototype system, and conduct comprehensive experiments and comparison studies using the real-world activity trace we collected to evaluate the performance of our system.

The rest of the paper is organized as follows: Section 2 discusses the related work. In Section 3, we describe our body sensor network design. Section 4 gives the background on Emerging Pattern, and then describes the mining of Emerging Patterns. We present our activity models and the epMAR recognizer in Section 5. Section 6 reports our prototype system and empirical studies, and finally, Section 7 concludes the paper.

2 Related Work

In the literature review, we first briefly review previous work in single-user activity recognition. We then focus our discussion on the recent advance of multiuser activity recognition.

2.1 Single-User Activity Recognition

Much early work in human activity recognition has been done in computer vision to recognize single-user activities. They leverage on video cameras as passive sensors, and explore various spatial-temporal analysis to recognize people's actions from video sequences.

The advances of wireless networks and small form factor sensors motivate much research work in sensor-based activity recognition over the past few years. This approach is fundamentally different than vision-based systems in terms of the way to capture observations. As compared to video camera, sensor devices are portable, unobtrusive, and easy to deploy. In addition, sensors capture many useful observations about human and environment which are not possible to be done in vision-based systems. Typical sensors range from on-body sensors (e.g., accelerometer, RFID) to ambient sensors (e.g., switch sensors, infrared motion sensor). Sensor readings are collected in the form of a continuous sensor data stream, and interpreted by an appropriate activity model for classification. Models to classify sensor data are typically probabilistic-based, and they can be categorized into static classifier and temporal classifier. Typical static classifiers include naïve Bayes [5], decision tree [6], and k-nearest neighbor [12]. These classifiers are relatively fast to train, and also less computationally expensive to perform classification. However, they do not capture any temporal transition information of the modeled activities. In temporal classification, state-space models are used to model and infer hidden states (e.g., activity labels) given observations. We name a few examples here: Dynamic Belief Network (DBN) [4], Hidden Markov Model (HMM) [13], and Conditional Random Field (CRF) [14]. These approaches incorporate information about the transitions between activities via the transition matrix of the system. Other approaches include probabilistic context-free grammars proposed by Lymberopoulos et al. [15] to infer activities in sensor networks. In their framework, activities are described in high-level scripts that are directly mapped to hierarchical probabilistic grammars which are obtained through training. However, this approach has drawbacks in handling uncertainty and noise in sensor data streams. Hamid et al. [16] proposed a time series classifier in which an activity is modeled as a sequence of discrete events. Activities are recognized through discovering and matching the subsequences with similar behavior appeared frequently in time series data. Different from the time series classifier concerning the mining of regularities, in our approach we mine the abnormal growth among classes.

2.2 Multiuser Activity Recognition

Some work has been done in modeling interacting processes and recognizing multiuser activities in computer vision. Oliver et al. [17] proposed and compared HMMs and Coupled HMMs (CHMMs) for modeling interactions between people and classifying the type of interaction based on observations collected from video camera. CHMM is shown to work more efficiently than HMM. Gong and Xiang [18] developed a dynamically multilinked HMMs model to interpret group activities based on video camera. Nguyen et al. [19] employed hierarchical HMM for modeling the behavior of each person and the joint

probabilistic data association filters for data association. Park and Trivedi [20] presented a synergistic track- and body-level analysis framework for multiperson interaction and activity analysis in the context of video surveillance. An integrated visual interface for gestures and behavior was designed in [21] as a platform for investigating visually mediated interaction with video camera. To use other sensor modalities, Wyatt et al. [22] presented a privacy-sensitive DBN-based unsupervised approach to separating speakers and their turns in a multiperson conversation. They addressed the problem of recognizing sequences of human interaction patterns in meetings with a two-layer HMM using both audio and video data. These temporal models have shown good accuracy performance prior to proper training with a large data set. However, they are computationally expensive, and hence their scalability remains a serious problem. In addition, although HMM has been shown to suppress white noise in speech recognition applications [23], the ability of handling different types of sensor noise still remain questionable.

To the best of our knowledge, there is no formal study on modeling and recognizing multiuser activities using body sensor networks. Our approach is fundamentally different from the above in its use of discriminative pattern mining. Discriminative patterns mining has been studied in the data mining literature, and applied in many domains. For example, Khan et al. [24] proposed discriminative frequent pattern mining to investigate interactive bugs in sensor networks. In this work, we propose a pattern-based approach to model multiuser activities in wireless body sensor networks. In our earlier work [25], we have shown the effectiveness of a pattern-based approach to single-user activity recognition. This paper exploits a similar approach to a more challenging problem—recognizing multiuser activities. In our earlier study [26], we used machine learning algorithms to recognize multiuser activities where the scalability of the solution remains a big issue. In this work, we propose a pattern mining approach and a user proximity-based technique to improve the computation scalability significantly. The earlier version of this work appeared in [27].

3 Body Sensor Network Design

We design a wireless body sensor network, as shown in Fig. 1a. It consists of five sensor nodes—two IMOTE2 motes, two RFID reader motes, and an acoustic sensor node. An IMOTE2 mote is located on each wrist of a subject to capture hand movement, environmental temperature, humidity, and light; it consists of an IPR2400 processor/radio board and an ITS400 sensor board with a tri-axis accelerometer, as shown in Fig. 1c. An RFID reader mote is located on each hand to capture object use; it consists of a MICA2Dot mote and a coin-size short-range RFID reader, as shown in Fig. 1b. An RFID reader mote is able to detect the presence of a tagged object within a few centimeters. We use an audio recorder as an acoustic sensor to capture sound, as shown in Fig. 1d. In addition, detecting user's location at room-level granularity is done in a simple way that an UHF RFID reader is located in each room to sense the proximity of a subject wearing an UHF tag. Figs. 1f, 1g, and 1e show the sink nodes and servers where all the sensor data with timestamps are logged.

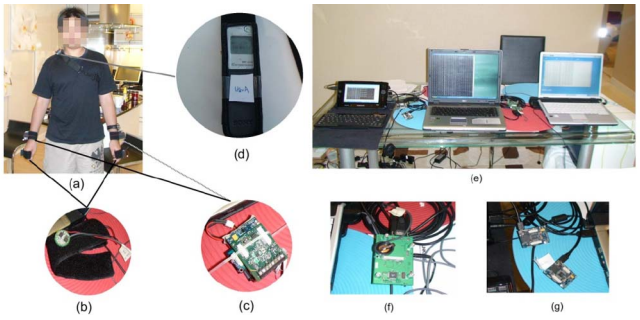


Fig. 1. (a) Our body sensor network consists of two IMOTE2 motes, two RFID reader motes, and one audio recorder, (b) RFID reader mote, (c) IMOTE2 mote, (d) audio recorder, (e) servers, (f) MICA2Dot sink node, and (g) IMOTE2 sink node.

The sensor data stream captured by our body sensor network has 12 dimensions—3-axis acceleration data for both hands, object use for both hands, temperature, humidity, light, and user location. In the following sections, we describe how to discover useful patterns from the data stream for building our activity models, and how to use these models to recognize activities.

4 Emerging Pattern for Activity Classification

Emerging Pattern is one of the discriminative patterns, and it describes significant differences between two classes of data. In this paper, we use EPs to build a novel activity model for activity classification. In this section, we first provide the background of EPs, and then describe how to discover EPs from the features extracted from raw sensor data.

4.1 Emerging Pattern

Suppose that a data set D consists of many instances. An instance contains a set of items (i.e., an item set), where an item is an attribute-value pair. The support of an item set X, $supp_D(X)$, is $count_D(X)/|D|$, where $count_D(X)$ is the number of instances in D containing X.

Definition 1 (Growth Rate). *Given two different classes of data sets D_1 and D_2, the growth rate of an item set X from D_1 to D_2 is defined as* $GrowthRate(X) =$

$$\begin{cases} 0, & if \quad supp_1(X) = 0 \quad and \quad supp_2(X) = 0, \\ \infty, & if \quad supp_1(X) = 0 \quad and \quad supp_2(X) > 0, \\ \frac{supp_2(X)}{supp_1(X)}, & otherwise. \end{cases}$$

Growth rate is used to represent frequency changes significantly from one data set (D_1) to another (D_2). EPs are those item sets with large growth rates from D_1 to D_2.

Definition 2 (Emerging Pattern). *Given a growth rate threshold $\rho > 1$, an item set X is said to be a ρ-EmergingPattern (or simply EP) from a background data set D_1 to a target data set D_2 if $GrowthRate(X) \geq \rho$.*

An EP with high support in its target class and low support in the contrasting class can be seen as a strong signal indicating the class of a test instance containing it.

Hence, it can be used as a powerful discriminator to differentiate the class membership of instances that contain the EP.

4.2 Feature Extraction

Before discovering EPs from sensor data, we first need to extract features which are potentially useful for activity classification. The raw sensor data will be first converted to a sequence of *observation vectors* by concatenating all of the raw data in a fixed time interval which is set to one second in our experiments, and then we extract features for each sensor modality which is described as follows:

For acceleration data, we use five common features—DC mean, variance, energy, frequency-domain entropy, and correlation. The DC mean is the mean acceleration value in a time interval. Variance is used to characterize the stability of a signal. Energy which captures data periodicity can be used to discriminate sedentary activities from moderate and vigorous ones; and it is computed as the sum of the squared discrete FFT component magnitudes of a signal. Frequency-domain entropy helps to discriminate activities with similar energy values, and it is computed as the normalized information entropy of the discrete FFT component magnitudes of a signal. Correlation between axes is especially useful for discriminating between activities that involve translation in just one dimension. It is computed for every two axes of each accelerometer and all pairwise axes combinations of two different accelerometers.

For audio data, we use both time-domain and frequency-domain features. The time-domain features measure the temporal variation of an audio signal, and consist of three features. The first one is the standard deviation of a reading in a time interval, normalized by the maximum reading in the interval. The second one is the dynamic range defined as $(max - min)/max$, where min and max represent the minimum and maximum readings in the interval. The third one is Zero-Crossing Rate (ZCR) which measures the frequency content of a signal, and it is defined as the number of time-domain zero crossings in a time interval. In the frequency domain, we compute two features—centroid (the midpoint of the spectral power distribution) and bandwidth (the width of the range of frequencies which a signal occupies).

For RFID reading or location information, we use object name or location name directly as features. For each RFID wristband reader, we choose the first object in a one-second interval since a user is unlikely to touch two or more objects in such a short interval. If no RFID reading is observed or in the presence of a corrupted tag ID, the value will be set to NULL.

Based on the above extraction process, we transform a 12-dimensional *observation vector* into a 47-dimensional *feature vector*. A *feature vector* consists of many *feature items*, where a *feature item* refers to a feature name-value pair in which a feature can be numeric or nominal. We denote a numeric feature as $numfeature_i$. Suppose its range is $[x, y]$ and an interval $[a, b]$ (or in other forms, $(a, b]$, $[a, b)$, or (a, b)) is contained in $[x, y]$. We call $numfeature_i@[a, b]$ a *numeric feature item*, meaning that the value of $numfeature_i$ is limited inclusively between a and b. We denote a nominal attribute as $nomfeature_j$. Suppose its range is $\{v_1, v_2, \ldots, v_n\}$, we call $nomfeature_j@v_k$ a *nominal feature item*, meaning the value of $nomfeature_j$ is v_k.

Numeric feature values are continuous, hence they need to be discretized. We follow the entropy-based discretization algorithm [28]. This algorithm partitions a range of continuous values into a number of disjoint intervals by using class information entropy. The class information entropy of candidate partitions is used to select binary boundaries, and the minimal entropy criteria are then used to find multilevel cuts for each attribute. The discretization algorithm partitions 44 numeric feature values into a total of 484 disjoint intervals. We then directly combine the feature name and its interval into a *numeric feature item*. For the nominal feature, the feature name and its value are combined as a *nominal feature item*. For the $left\_object$ and $right\_object$ features, we merge them into one feature by computing $left\_object \cup right\_object$ without losing any object due to user's handedness. In our current sensor setting, we have a total of 574 *feature items*. Finally, we obtain a sequence of *feature vectors*, and they will be used as inputs for our mining process described in the next section.

4.3 Mining Emerging Patterns from Feature Vectors

To discover EPs from *feature vectors* for each activity class, we first use 10-fold cross-validation [29] to generate training data set $\mathcal{O}$. Each training data set $\mathcal{O}$ consists of a number of instances in which each observation $o \in \mathcal{O}$ is annotated with an activity label, and we separate single-user instances from multiuser instances. We then mine the EPs in $\mathcal{O}$ for each activity class $A_i \in \mathcal{A}$, where $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$. More specifically, for User 1 and each activity A_i performed, we mine a set of EPs that occur frequently in $\mathcal{O}^1_{A_i}$ and rarely in $\mathcal{O}^1 - \mathcal{O}^1_{A_i}$, where $\mathcal{O}^1$ is the entire training data set for User 1, and $\mathcal{O}^1_{A_i}$ is the training instances for A_i. We denote $EP^1_{A_i}$ as the EPs of A_i for User 1.

We mine EPs of single-user activities and multiuser activities separately by an efficient algorithm [30]. The key idea of this algorithm is to mine a concise representation of equivalence classes of frequent item sets in a data set. An equivalence class is a set of item sets that always occur together in some transactions of the data set. The item sets within an equivalence class all share the same level of statistical significance regardless of the variety of test statistics. As an equivalence class can be uniquely determined and concisely represented by a closed pattern and a set of generators, the algorithm discovers closed patterns and generators using a simultaneous depth-first search scheme. After computation, we get m sets of EPs, one set per activity for each user. We refer $\mathcal{EP}^1$ as the entire set of EPs for User 1, where $\mathcal{EP}^1 = \{EP^1_{A_1}, EP^1_{A_2}, \ldots, EP^1_{A_m}\}$.

Table 1 presents an example of the EPs of the *brushing teeth* activity for both User 1 and 2. For readability, we translate each feature number in an EP into a feature name-value pair so that the meaning of each item is obvious. Columns 3 and 4 show the corresponding values of support and growth rate for each EP. To illustrate, the EP for User 1 contains acceleration data, objects such as toothpaste and toothbrush, location (i.e., bathroom) and audio data, and it has a support of 90 percent and a growth rate of infinity. In fact, one of the advantages of EPs is that the patterns discovered is easy to understand. These sets of EPs will be used to build our activity models described in the next section.

TABLE 1
A Partial Set of the EPs of *Brushing Teeth* for both Users

User	EPs	Support(%)	Growth rate
User 1	$\{ accel\_left\_x\_mean@(-201.9765 \sim 107.0105], accel\_correlation\_9@(-\infty \sim 77890.1185], location@bathroom, object@toothpaste, accel\_left\_x\_mean@(-831.872 \sim 612.6145], accel\_right\_z\_entropy@(-3.2115 \sim 1.511], audio\_centroid@(2750.1775 \sim 3680.728], object@water, audio\_zcr@(0.3315 \sim 0.4815], audio\_centroid@(3680.728 \sim 5090.182] \}$	90	$+\infty$
User 2	$\{object@toothbrush, audio\_centroid@(2617.5925 \sim \infty], object@water, location@bathroom, accel\_correlation\_4@(-\infty \sim 354687.569], audio\_zcr@(0.42 \sim \infty], object@toothpaste\}$	90	$+\infty$

5 EP-Based Activity Model and Recognition Algorithm

Based on the EPs we obtained above, we are now ready to build activity models and recognition algorithms. In this section, we first describe the problem statement, then present an overview of epMAR, followed by describing activity models and recognition algorithms in details.

5.1 Problem Statement

We formulate the problem of multiuser activity recognition as follows: Suppose we have n users preforming their activities, and such activity can be a single-user activity performing by each individual independently or a multiuser activity performing by several users collaboratively or concurrently. For each user, we collect an activity trace which is a sensor data stream consisting of a continuous sequence of sensor observations. Given a training data set for each user, our objective is to design an appropriate activity model and develop a recognition algorithm to assign each new observation in each data set with the correct activity label. Formally, the training data set $\mathcal{O}$ of an user consists of T observations, $\mathcal{O} = \{o_1, o_2, \ldots, o_T\}$, and each observation is associated with an activity label A_i, where $A_i \in \mathcal{A}$ and $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$, i.e., $\mathcal{A}$ is the entire set of activity classes and there are m activities in total.

5.2 Overview of epMAR

Fig. 2 gives an overview of epMAR. The inputs are multiple sensor data streams (i.e., multiple sequences of observations) where each data stream corresponds to a user. A sensor data stream will be first preprocessed into *feature vectors*. Our system operates in two phases—model training and activity recognition. In the training phase, a training data set for each of the users will be used to mine sets of EPs (described in Section 4.3) and train the activity models. The activity models consist of both a single-user model and a multiuser model. While the single-user model is used to recognize single-user activities, the multiuser model is capable of recognizing multiuser activities since it captures user interactions. Both models use EPs as the main discriminator for activity classification. In the recognition phase, given a sequence of *feature vectors* (i.e., $S_t, t = 0 \sim T$) for each user, we first obtain a test instance (i.e., $S_{t \sim t+L_{A_i}}$) for a possible activity A_i by segmenting the sequence using a sliding window with a length of L_{A_i} (L_{A_i} is the average duration of A_i and can be obtained from the training data), and then we apply our recognition algorithm to classify and label this sequence segment. The above process will be performed recursively. Since each of these sequence segments corresponds to an activity label, for each pair of consecutive sequence segments, we design an algorithm to detect and adjust the boundary. This algorithm serves as a feedback loop in our system aiming to classify sequence segments accurately and overcome the drawback of a sliding window-based segmentation method. When recognizing multiuser activities, we use a proximity-based filtering technique to reduce the model computation for user interactions to achieve better scalability.

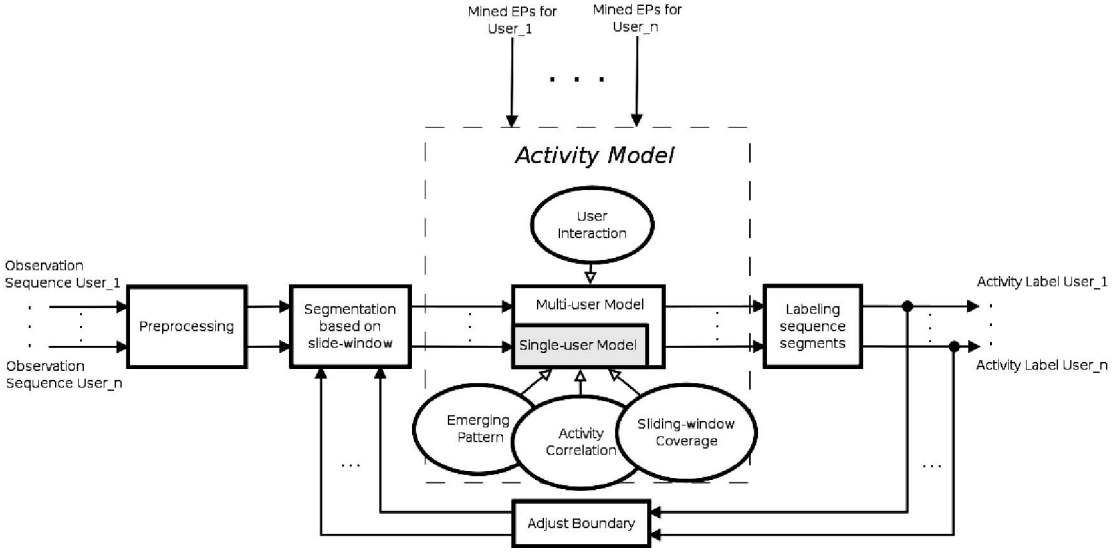


Fig. 2. Overview of epMAR.

5.3 Activity Models

An activity model measures the likelihood of each possible activity label given a test instance. We design a probabilistic score function to measure such likelihood.

5.3.1 Single-User Model

The score function for the single-user model is composed of three probability elements: *EP score*, *Sliding-Window Coverage score*, and *Activity-Correlation score*. We describe the three elements as follows:

EP Score. Given a test instance $S_{t\sim t+L_{A_i}}$ and a possible activity for A_i, this score element measures the likelihood of $S_{t\sim t+L_{A_i}}$ matching the EPs of A_i. The more fraction of EPs we detect in $S_{t\sim t+L_{A_i}}$, the higher EP score we obtain and the higher likelihood $S_{t\sim t+L_{A_i}}$ matches A_i. The measurement is done by detecting each EP_{A_i} contained in $S_{t\sim t+L_{A_i}}$. As each activity class is typically associated with more than one set of EPs, we aggregate the strength of each EP to sum up their contributions. We use a simple aggregation method described in [31], and the aggregated EP score of $S_{t\sim t+L_{A_i}}$ for A_i is defined as follows:

$$aggr\_score(A_i, S_{t\sim t+L_{A_i}}) = \sum_{X\subseteq S_{t\sim t+L_{A_i}}, X\in EP_{A_i}} \frac{growth\_rate(X)}{growth\_rate(X)+1} * supp_{A_i}(X), \quad (1)$$

where $supp_{A_i}(X)$ is the support of X in A_i, and $growth\_rate(X)$ is $supp_{A_i}(X)$ divided by the X's support in non-A_i classes. Then, the EP scores of each activity are "normalized" by dividing them using the median of the scores of the training instances of that activity. Finally, the *EP score* of $S_{t\sim t+L_{A_i}}$ for A_i is defined as follows:

$$ep\_score(A_i, S_{t\sim t+L_{A_i}}) = \frac{aggr\_score(A_i, S_{t\sim t+L_{A_i}})}{base\_score(A_i)}, \quad (2)$$

where $base\_score(A_i)$ is the median of the values of $aggr\_score(A_i, S_{t\sim t+L_{A_i}})$ obtained from training data.

Sliding-window coverage score. We use a sliding window with a length of L_{A_i} to segment S_t and obtain a test instance $S_{t\sim t+L_{A_i}}$ for each possible activity A_i. However, L_{A_i} provides only an estimation, and the actual length of $S_{t\sim t+L_{A_i}}$ for each A_i deviates from L_{A_i} in reality. To provide a measurement of how well a sliding window covers the entire $S_{t\sim t+L_{A_i}}$, we introduce *Sliding-Window Coverage score*. It measures how many relevant or irrelevant observations contained in a test instance for a particular activity. The less irrelevant observations we detect in a test instance, a higher *Sliding-Window Coverage score* we obtain, and the better the sliding window covers the test instance. We denote $coverage\_score(A_i, S_{t\sim t+L_{A_i}})$ as the *Sliding-Window Coverage score* of a test instance $S_{t\sim t+L_{A_i}}$ for an activity A_i. This score element is computed based on $relevance(A_i, f_p)$, where f_p is a feature vector contained in L_{A_i}. Recall that a feature vector is a set of feature items, $relevance(A_i, f_p)$ is computed based on $relevance(A_i, item_h)$ for each $item_h \in f_p$ which is defined as follows (note that an item $item_h \in X, X \in EP_{A_i}$ will be given more weights since it is part of EP_{A_i})

$$relevance(A_i, item_h) = P(item_h|A_i) + \sum_{item_h\in X, X\in EP_{A_i}} supp_{A_i}(X). \quad (3)$$

We then aggregate the value of $relevance(A_i, item_h)$ for all $item_h \in f_p$, and the normalized $relevance(A_i, f_p)$ can be computed as follows:

$$relevance(A_i, f_p) = \frac{unnorm\_relevance(A_i, f_p)}{base\_relevance(A_i)}. \quad (4)$$

Finally, the *coverage score* of $S_{t\sim t+L_{A_i}}$ for A_i is computed by averaging all the $relevance(A_i, f_p)$ as follows:

$$coverage\_score(A_i, S_{t\sim t+L_{A_i}}) = \frac{1}{L_{A_i}} \sum_{f_p\in L_{A_i}} relevance(A_i, f_p). \quad (5)$$

Activity-correlation score. This score element is designed to measure correlations between activities. Such correlation commonly exists in our daily lives. For example, a user usually brushes his teeth, followed by washing his face; cleans the dining table after eating a meal.

We use conditional probability to model correlations between activities. We define the *Activity-Correlation score* of A_i as $P(A_i|A_j)$, which is the conditional probability of A_i given A_j. This score can be easily obtained from the training data set. Note that the initial value of the conditional probability of A_i is set to zero, i.e., $P(A_i|NULL) = 0$.

Finally, we construct our single-user activity model using a linear combination of the above three elements, and we have the following definition.

Definition 3. *Given a time t and a labeled activity A_j which ends at t, for each possible activity A_i, a test instance $S_{t\sim t+L_{A_i}}$ is obtained from t to $t + L_{A_i}$, the score function of A_i for a single-user model is then defined as follows:*

$$\begin{aligned} single\_user\_score(A_i, A_j, S_{t\sim t+L_{A_i}}) &= c_1 * ep\_score(A_i, S_{t\sim t+L_{A_i}}) \\ &+ c_2 * coverage\_score(A_i, S_{t\sim t+L_{A_i}}) \\ &+ c_3 * P(A_i|A_j), \end{aligned}$$

where c_1, c_2, and c_3 are the coefficients, representing the importance of an individual score element.

These coefficients reveal different activity and behavior patterns of a user. For example, a higher c_1 implies that a user always performs her/his activities in a consistent manner. A higher c_2 implies that an activity is performed in a constant duration whereas a lower c_2 implies that the duration variance of the activity among all the instances can be large. If c_3 is high, it implies that there exists strong correlation between activities performed by a user, i.e., activities performed always follow a certain order. These coefficients are obtained from our experiment which will be evaluated and discussed in Section 6.

5.3.2 Multiuser Model

The multiuser model is designed to recognize multiuser activities, and it measures the likelihood of performing an activity with other users. This is achieved by modeling user interactions, in addition to modeling EPs, sliding-window

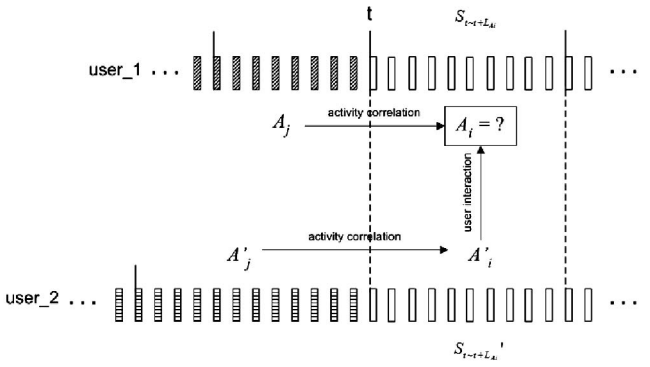


Fig. 3. An illustration of a two-user model.

coverage, and activity correlation in the single-user model. To model user interactions, we examine the test instances of other users who are possibly involving the interaction, and measure the joint probability of their possible labels. To illustrate, we take an example of a two-user scenario as shown in Fig. 3. Two sequences corresponding to User 1 and User 2, respectively, are input into the recognizer where each sequence is handled separately. The figure illustrates the case that we are now detecting a test instance $S_{t\sim t+L_{A_i}}$ of User 1 for a possible multiuser activity A_i, given the previous labels for both users (i.e., a labeled activity A_j for User 1 and another labeled activity A'_j for User 2). We introduce an additional score element, $inter\_score$, to model the interaction between these two users, which is defined as follows:

$$inter\_score(A_i, A'_j, S'_{t\sim t+L_{A_i}}) = \max_{A'_i}[P(A_i|A'_i)*P(A'_i|A_i)*confidence(A'_i, A'_j, S'_{t\sim t+L_{A_i}})], \quad (6)$$

where $S'_{t\sim t+L_{A_i}}$ is the sequence segment of User 2 during the time period from t to $t + L_{A_i}$, A'_i is the possible activity for $S'_{t\sim t+L_{A_i}}$, and $P(A_i|A'_i)*P(A'_i|A_i)$ is the joint probability of A_i and A'_i occurs while they are conditioned on each other. $confidence(A'_i, A'_j, S'_{t\sim t+L_{A_i}})$ measures the certainty of label A'_i, and it is defined as follows:

$$confidence(A'_i, A'_j, S'_{t\sim t+L_{A_i}}) = \frac{single\_user\_score(A'_i, A'_j, S'_{t\sim t+L_{A_i}})}{\sum_k single\_user\_score(A'_k, A'_j, S'_{t\sim t+L_{A_i}})}. \quad (7)$$

Finally, we construct our multiuser activity model by combining the interaction score element with the single-user model, which is formally defined as follows:

Definition 4. *Given a time t, a labeled activity A_j which ends at t for User 1 and a labeled activity A'_j which ends at t for User 2, for each activity A_i of User 1, a test instance $S_{t\sim t+L_{A_i}}$ is obtained from t to $t + L_{A_i}$, $S_{t\sim t+L'_{A_i}}$ is obtained for User 2, the score function of A_i for a multiuser model is then defined as follows:*

$$multi\_user\_score(A_i, A_j, A'_j, S_{t\sim t+L_{A_i}}, S'_{t\sim t+L_{A_i}}) = single\_user\_score(A_i, A_j, S_{t\sim t+L_{A_i}}) + c_4*inter\_score(A_i, A'_j, S'_{t\sim t+L_{A_i}}), \quad (8)$$

where c_4 is the coefficient representing the importance of $inter\_score$.

Similar to other coefficients, a higher c_4 implies there exist more interactions between multiple users when they perform a multiuser activity. This coefficient is again obtained from our experiment which will be described in Section 6. Note that while Definition 4 defines the multiuser activity model for a two-user case, it applies to n users as well.

When detecting a multiuser activity for a given user, our multiuser model captures each possible interaction between this user and each of the other users. Given n users in a sensor network space, it requires the computation of $O(n)$ for single user, and $O(n^2)$ for all the n users. To improve the scalability of our multiuser model, we propose a proximity-based filtering technique which limits the computation of user interactions based on user proximity. With this technique, the computation of the multiuser model is only necessary when users are in close physical proximity with each other, i.e., room-level proximity in a home environment. This is reasonable since multiuser activities in a home environment are typically performed at the room level. For example, if a user is currently located at the kitchen room, a multiuser activity can only occur between she/he and other users in this room. Hence, we only need to compute the interaction scores of this user and each of other users in this location, resulting in greatly reducing the computation and improving the scalability of our model. The proximity information (i.e., location in the room granularity) can be easily extracted from the sensor data stream. This technique is integrated in our recognition algorithm and will be evaluated in our experiments.

5.4 Algorithm to Detect and Adjust Boundary

As described in Section 5.2, we obtain a test instance using a sliding window, and such segmentation can be done recursively. However, since a sliding window provides only an approximation of the actual duration of the instance. Any error in a segmentation may affect subsequent segmentations. To overcome this limitation, we propose an algorithm to detect and adjust the boundary of the two label activities. This algorithm is based on the following heuristics. Intuitively, given an activity instance, its *feature vectors*, obtained from preprocessing its observations, usually have a higher relevance to this activity than all other activities. Furthermore, the relevance of its feature vectors in the same activity instance does not vary significantly as compared to the relevance of two feature vectors belonging to two different activities. Based on these heuristics, we design an algorithm, as shown in Algorithm 1, to detect the boundary and adjust the segmentation so that a sliding window can be aligned to the starting point of each test instance.

Algorithm 1. The Algorithm to Detect and Adjust Boundary-**adjustBoundary**

Input: feature vectors of a sliding-window length $L_{A_j} + L_{A_i}$: $F = \{f_{t-L_{A_j}}, \ldots, f_{t+L_{t+L_{A_i}}}\}$, where t is the existing boundary, labeled activity A_j followed by A_i.

Output: the boundary between A_j and A_i.

1: **foreach** p from $t - L_{A_j}$ to $t + L_{A_i}$ **do**
2: $RW[p] = relevance(A_j, f_p) - relevance(A_i, f_p)$;
3: **end for**

4: **foreach** p from $t - L_{A_j}$ to $t + L_{A_i}$ **do**
5: $upperSum =$ sum of all RWs from $t - L_{A_j}$ to p;
6: $lowerSum =$ sum of all RWs from p to $t + L_{A_i}$;
7: $GAIN[p] = upperSum - lowerSum$;
8: **end for**
9: boundary $= p$ such that $GAIN[p]$ is maximum;
10: **return** boundary;

5.5 The epMAR Algorithm

When recognizing activities for each user, given m possible activities, we first obtain a test instance $S_{t\sim t+L_{A_j}}$ for A_j using L_{A_j}; and we compute $score(S_{t\sim t+L_{A_j}}, A_j)$ for A_j. The one with the maximum score is assigned to $S_{t\sim t+L_{A_j}}$ with its candidate label A_j. The same process is repeated to compute another candidate label A_i. We then apply the algorithm to detect and adjust the boundary of A_j and A_i. Finally, we recompute the score for each activity label, and the label with the maximum score yields the final label for this instance. The above process then runs recursively from this boundary over the entire trace. The entire process of epMAR is described in Algorithm 2.

Algorithm 2. The epMAR Algorithm
Input: n users ($user\_1, user\_2, \ldots, user\_n$);
m activities $\{A_1, A_2, \ldots, A_m\}$;
n observation sequences with a length of T corresponding to each user:
$O^1 = \{o_1^1, o_2^1, \ldots, o_T^1\}$,
$O^2 = \{o_1^2, o_2^2, \ldots, o_T^2\}, \ldots, O^n = \{o_1^n, o_2^n, \ldots, o_T^n\}$.
Output: assign the activity label to each observation.
pre-process $O^1, O^2, \ldots, O^n$ to obtain feature vectors $F^1, F^2, \ldots, F^n$ where $F^1 = \{f_1^1, f_2^1, \ldots, f_T^1\}$, $F^2 = \{f_1^2, f_2^2, \ldots, f_T^2\}, \ldots, F^n = \{f_1^n, f_2^n, \ldots, f_T^n\}$;
$t^1 = t^2 = \cdots = t^n = 1$;
while $t^1 \le T$ or $t^2 \le T$ or ... or $t^n \le T$
  foreach pair of users $u = 1, 2$ who are located in the same room **do**
    $A^u_{previous} =$ null; $A^u_{candidate} =$ null;
    foreach activity $A_i, i = 1, 2, \ldots, m$ **do**
      get instance $S^1_{t\sim t+L_{A_i}} = \cup_{p=t}^{t+L_{A_i}} f_p^1$;
      get instance $S^2_{t\sim t+L_{A_i}} = \cup_{p=t}^{t+L_{A_i}} f_p^2$;
      compute $score\_with\_inter(A_i, A^u_{previous}, A^{u\_other}_{previous}, S^u_{t\sim t+L_{A_i}}, S^{u\_other}_{t\sim t+L_{A_i}})$;
    end for
    $A^u_{current} = A_i$ with the highest score;
    if $t^u = 1$ or $A^u_{current} = A^u_{candidate}$
      Assign label $A_{current}$ to $o_t \sim o_{t+L_{A_{current}}}$;
      $t^u = t^u + L_{A^u_{current}}$;
      $A^u_{previous} = A^u_{current}$;
      $A^u_{candidate} = null$;
    else if $A^u_{candidate} \neq A^u_{current}$
      $t^u = adjustBoundary(F^u_{t-L_{A^u_{previous}}}, t + L_{A^u_{current}}, A^u_{previous}, A^u_{current})$;
      $A^u_{candidate} = A^u_{current}$;
    end if
  end for
end while

The epMAR algorithm is flexible in a way that it works for both single- and multiusers. In the case of single-user, the recognizer takes a single-user observation sequence as input and recognizes activities based on the single-user model. In the case of multiple users, the recognizer takes multiple observation sequences (corresponding to the users in the same location) as input and detects activities based on our multiuser model.

We now analyze the complexity of epMAR. Assuming that computing a score for an activity requires time $O(S)$, and adjusting the boundary between two activities needs time $O(B)$, where both $O(S)$ and $O(B)$ are linear to the length of a slice window. Given the total number of activities m, number of users n, and assuming the average loop for confirming an activity instance is k. For each pair of users in the same room, it takes $k \cdot m \cdot O(S) + (k-1) \cdot O(B)$ time to recognize an activity. In the worst case, all users are in the same room, we need to consider all the possible pairs of the users, which is C_n^2 pairs. Assuming that the average number of activity instances contained in each user's observation sequence is l. The total complexity of epMAR can be computed as $l \cdot C_n^2 \cdot (k \cdot m \cdot O(s) + (k-1) \cdot O(B))$. Based on the measurement result in our experiments, the value of k falls in the interval [2.5, 3.0]. Therefore, the time complexity of epMAR is finally $l \cdot C_n^2 \cdot (m \cdot O(S) + O(B))$.

6 Experimental Studies

We now move to evaluate epMAR. Up to now, there are no public standard data sets available for evaluating multiuser activity recognition [31]. We build one by conducting real world trace collection. In this section, we first describe the experimental setup, then present our results.

6.1 Experimental Setup

We implement our body sensor network described in Section 3. This sensor platform measures hand movement (i.e., acceleration data), human-to-object interaction (i.e., objects touched and sound), human-to-human interaction (i.e., voice), environmental information (i.e., temperature, humidity, and light), and user location at room granularity.

To capture acceleration data, we use an IMOTE2 mote which consists of an IPR2400 processor/radio board and an ITS400 sensor board with a tri-axis accelerometer. The sensor board also measures environmental temperature, humidity, and light. To capture object use, we build a customized RFID reader mote which incorporates a MICA2Dot mote, a coin-size Skyetek M1-mini RFID reader, and a Li-Polymer rechargeable battery. The RFID reader mote is able to detect the presence of a tagged object within the range from 6 to 8 cm, thus, human-to-object interaction (i.e., objects touched by a user) can be captured. It is also able to capture human-to-human interactions through objects, i.e., handling an object together or passing an object from one user to another. To capture vocal interactions among users and environmental sound, we use an audio recorder as an acoustic sensor with a sampling rate of 16 kHz since the built-in microphone on ITS400 fails to support such high sampling rate due to the bandwidth limitation of a wireless link. In addition, detecting user's

TABLE 2
Activities Performed in Our Trace Collection

Single-user Activities	0	brushing teeth	8	vacuuming
	1	washing face	9	using phone
	2	brushing hair	10	ironing
	3	making pasta	11	toileting
	4	watching TV	12	making tea
	5	reading book/magazine	13	eating meal
	6	using computer	14	drinking
	7	making coffee		
Multi-user Activities	15	making pasta		
	16	cleaning a dining table		
	17	making coffee		
	18	toileting (with conflict)[1]		
	19	watching TV		
	20	using computer		

[1] This is a case when one person tried to use the toilet which was being occupied.

indoor location is done in a simple way that a UHF RFID reader is located in each room to sense the proximity of a user wearing a UHF tag. This simply method is able to keep track of multiple users at room-level granularity. To determine user identity, the IDs of each IMOTE2 and RFID reader are logged and bounded to a specific user. However, it is not possible to determine audio identity since audio information may be generated from other users who stay in the same room.

The sampling rate of the 3-axis accelerometer in each IMOTE2 mote is set to 128 Hz, the sampling rate of each RFID reader mote is set to 2 Hz, and the sampling rate of audio recorder is set to 16 KHz. All the sensor data with timestamps are logged.

6.2 Trace Collection

Collecting sensor data for multiple users is a difficult and time-consuming task. Aiming to have a realistic data collection, we first conduct a survey among 30 university students. In this survey, each participant was asked to report on what daily activities (both single- and multiuser activities) she/he performed at her/his home, and how each activity is performed (i.e., where each activity is performed, number of persons involved, number of times each activity is performed each day, duration of each activity, steps performed and objects used in each activity, etc.). They were asked to report not only their own experiences, but also the experiences from their family members (e.g., parents, siblings, etc.). From the survey reports, we have a number of findings as follows: In a home environment, although many daily activities can be performed by multiple users, single-user activity is still the majority. Second, there are typically less than four people involved in a multiuser activity. Third, most of the multiuser activities are performed in the same room. Fourth, interactions among users occur in a number of ways including voice conversation and objects passing.

To have a reasonable and realistic data collection, we randomly select 21 activities (shown in Table 2) from the activity list in our survey. The ratio of number of single-user activities to number of multiuser activities is close to

Fig. 4. Snapshots showing tagged objects and various activities being performed in our data collection. (a) RFID tags, (b) tagged objects, (c) making pasta, (d) brushing teeth, (e) having meal, (f) watching TV, and (g) making coffee.

the survey result. We limit the number of users to 2 for reducing annotation efforts. Our data collection was done in a smart home (i.e., a living lab environment). The smart home consists of a living room, a kitchen, two bedrooms, a study room, a bathroom, and a store room. Each room, except the bathroom, is equipped with a video camera for recording the ground truth. We tagged over 100 day-to-day objects such as tablespoons, cups, and computer mouse using HF RFID tags with three different sizes—coin, clip, and card, as shown in Fig. 4a. Fig. 4b shows a screen shot of tagged objects in the kitchen. For metal objects such as kettle, we tagged on its plastic handle. For liquid objects such as water, we tagged on the faucet with a special plastic handle to sense the use.

The data collection was deployed naturalistically. We have two male subjects and both are student volunteers from a local university. Each day, each subject wore a set of wearable sensors we developed and performed these activities at his choice in an order which is close to his daily practice. Each subject followed his own step to perform each of these activities, resembling the situation in his daily routine at his home. Fig. 4 shows some snapshots of various activities being performed in the kitchen, the bathroom, and the living room during our data collection. A set of servers was set up in the living room to log the trace. All the servers and sensors were synchronized before data collection. For each user, a trace was logged and annotated by an annotator who is also a student volunteer from a local university. The ground truth was also recorded by video cameras. Data collection was done over a period of 10 days across two weeks, and we collected a total number of 420 annotated instances for both subjects as shown in Table 3. The ratio of number of multiuser activity instances to number of single-user activity instances is higher than the result in our survey in order to have more multiuser activity instances. There

TABLE 3
Number of Instances Collected

Activity catergory	User 1	User 2
Single-user activity	150	150
Multi-user activity with collaboration	30	30
Multi-user activity with conflict	30	30

TABLE 4
Accuracy Performance

User	Single-user Activities	Multi-user Activities	Overall
User 1	86.69%	95.06%	89.72%
User 2	85.57%	95.71%	

are two cases in multiuser activities—multiuser activity with collaboration (i.e., two or more users working together to complete an activity in a cooperative manner, where each of them performs several steps of the activity) and multiuser activity with conflict (i.e., two or more users are involved in an activity in a conflicting manner, where users compete against each other for the activity).

6.3 Methodology

We use 10-fold cross-validation [29] for our evaluation. We first divide the entire trace into 10 data sets, then randomly select nine of them for training and the remaining one for testing. We evaluate the performance of our algorithm using time-slice accuracy which is a typical metric used in time series analysis. The time-slice accuracy represents the percentage of correctly labeled time slices. The length of time slice Δt is set to 15 seconds as our experiment shows different Δt doest not affect the accuracy much. This time-slice duration is short enough to provide precise measurements for activity recognition applications. The metric of the time-slice accuracy is defined as follows:

$$Accuracy = \frac{1}{N}\sum_{n=1}^{N}[predicted(n) == ground\_truth(n)], \quad (9)$$

where $N = T/\Delta t$.

6.4 Results

6.4.1 Accuracy Performance

In the first experiment, we evaluate the accuracy performance of epMAR with respect to different users and different activities. The results are summarized in Table 4, and the detailed breakdown is shown in Fig. 5. The overall accuracy of both users for both single- and multiuser activities achieves 89.72 percent, demonstrating that the epMAR recognizer is effective for recognizing activities in a multiuser scenario. Table 4 also shows that the accuracy of multiuser activity is higher than that of single-user activity

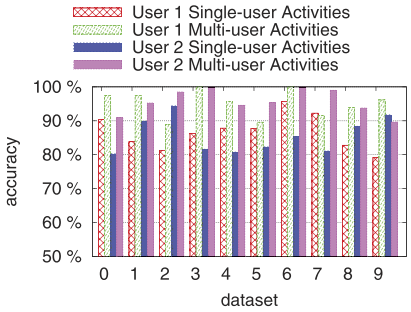


Fig. 5. Accuracy breakdown.

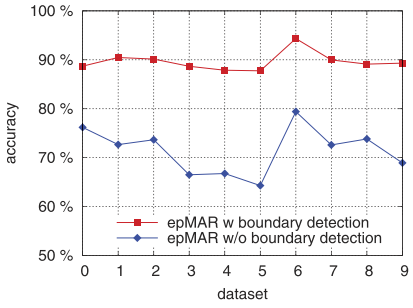


Fig. 6. Effect of boundary detection.

for both users. We analyze this phenomenon and suggest the reason as follows:

In a multiuser activity, each user usually performs only partial steps of this activity, resulting in the EPs of the multiuser activity are subsets of the EPs of the single-user activity. A test instance is likely to contain more EPs of a multiuser activity than its corresponding single-user activity, hence, the score of the multiuser activity tends to be higher. However, although the result is in favor of multiuser activities, an error in labeling for one user will not influence others as the test data are labeled separately.

6.4.2 Model Analysis

We conduct a number of experiments to analyze our system model in details. First, we evaluate the effect of boundary detection. Fig. 6 shows that epMAR with the boundary detection algorithm achieves 18.2 percent higher than that without the algorithm. This result demonstrates that the boundary detection algorithm works well to resegment and adjust the boundary of two labeled activities, and improves accuracy significantly.

Next, we evaluate the contribution of each type of sensors with respect to accuracy. We have mainly four types of sensors—accelerometer, audio, RFID tagged object, and location. In this experiment, each time we generate a new data set by removing one type of sensor data from the raw data set, and then recompute the features and reapply our epMAR algorithm to obtain the accuracy using the same methodology. This process is repeated for all the four types of sensors. As shown in Fig. 7, tagged object plays the most important role in our system, followed by location, accelerometer, and audio. It should be noted that the contribution of each type of sensors may vary from one activity to another and from one scenario to another. In a home setting, many day-to-day activities are associated with unique sets of objects, and hence extracting the EPs

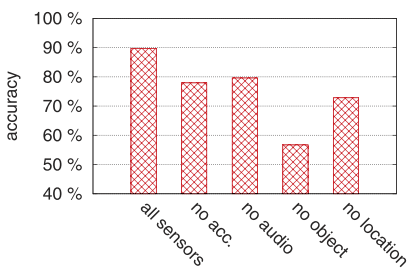


Fig. 7. Sensor selection.

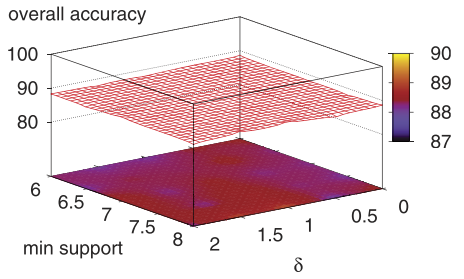


Fig. 8. Mining parameters.

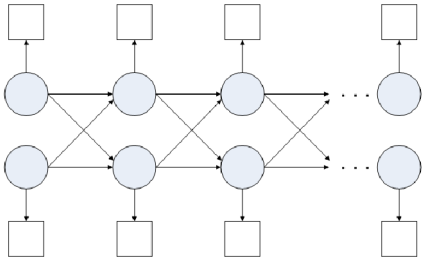


Fig. 9. Structure of CHMMs.

from those object sets plays the most important role in activity classification. Location also plays an important role since home activities are typically bound to specific locations at the room granularity, e.g., making a meal occurs merely in the kitchen. In addition, accelerometer and audio play the least important roles since many home activities share similar motion and audio patterns.

6.4.3 Parameter Analysis

There are two mining parameters in the experiment. The first one is *min support* (i.e., the minimum support threshold), and it is defined as an integer. The support sum of a frequent item set in all classes must be no less than *min support*. The second parameter is δ, and it is also defined as an integer. The support sum of an item set in all the minority classes must be no larger than δ. The EPs of an activity A_i are item sets that the support sum in all activities is no less than *min support*, and the support sum in all non-A_i activities are no larger than δ. Fig. 8 shows the effect of different parameter pairs with respect to the overall accuracy when they are set to the typical values. All the parameter pair achieve similar results with a maximum variation of 1.89 percent. We observe a smooth accuracy plane from the figure. It implies that the epMAR algorithm is not sensitive to the tuning of mining parameters.

We then evaluate the effect of coefficients (i.e., *c1*, *c2*, *c3*, and *c4*,) in our activity model. As we discussed in Section 5.3, each coefficient represents the weight of a score element (i.e., *EP*, *coverage*, *activity-correlation*, and *user interaction*). These coefficients reveal the habit of a person and the inherent patterns when performing activities. The optimal set of coefficients may vary from person to person and time to time. Investigating the effect of these coefficients with respect to different persons over different time has an important implication to personalized activity recognition and can be complex, hence we leave for our future work. In this experiment, we focus on studying their effects on a generalized activity model which intends to be applied to all the users. The experiment starts from setting a baseline value of 1 for all the four coefficients, and then varies the value of each of the coefficients, respectively. Table 5 shows the overall accuracies using different sets of coefficients. The coefficient set (1.0, 1.5, 1.0, 1.0) (i.e., the weight of the *Sliding-window Coverage* score is higher than the rest) achieves the best overall performance, and it reveals that both subjects seem to always perform their activities constantly (i.e., a constant duration for the same activity). This phenomenon matches the ground truth quite well.

6.4.4 Accuracy Comparison

As mentioned in the related work, CHMM is a typical technique to recognize multiuser activities in computer vision. CHMM extends HMM and it is designed for modeling interacting processes [32]. To capture the interaction between users over time, the coupling connects time slices with a conditional probability of transition, resulting in a structure with the crosswork of links as shown in Fig. 9, where the circle nodes represent the hidden states (i.e., activity labels) and the square nodes represent the observations (i.e., *feature vectors*).

We compare epMAR with CHMM. Both epMAR and CHMM were evaluated on the same training and testing data sets generated using 10-fold cross-validation, containing both single- and multiuser activities for both users. When CHMM is used to recognize single-user activities, the inner-chain state transition probabilities are set to zero, turning a CHMM into two basic HMMs. Fig. 10 shows the comparison result for all the 10 data sets. The overall accuracy of epMAR is 89.72 percent, outperforming CHMM. This result shows that an EP-based model is more effective than a temporal probabilistic model. It demonstrates that in

TABLE 5
Coefficients Analysis

Coefficients (c_1, c_2, c_3, c_4)	Accuracy of Single-user Activities	Accuracy of Multi-user Activities	Overall Accuracy
(1.0, 1.0, 1.0, 1.0)	84.52%	94.87%	87.75%
(1.5, 1.0, 1.0, 1.0)	83.14%	96.63%	87.43%
(1.0, 1.5, 1.0, 1.0)	86.15%	95.36%	89.72%
(1.0, 1.0, 1.5, 1.0)	84.87%	91.28%	86.07%
⋮	⋮	⋮	⋮
(1.0, 1.0, 1.0, 1.5)	85.34%	93.46%	88.76%

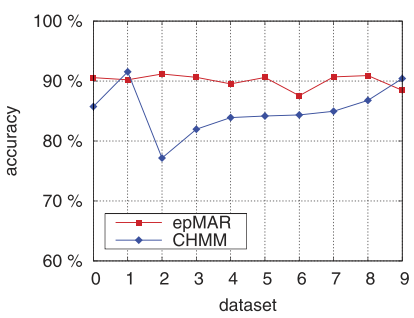


Fig. 10. Accuracy comparison.

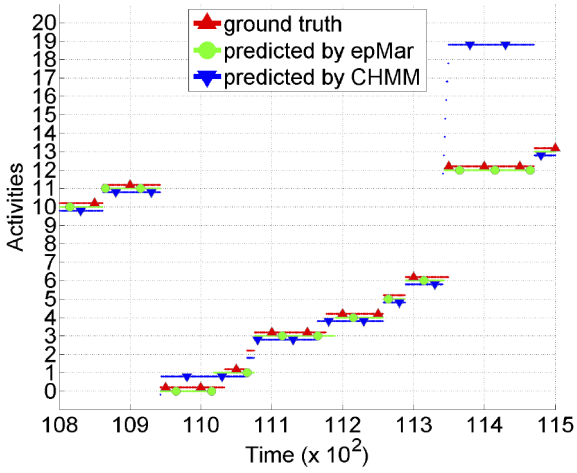


Fig. 11. A case of misdetection.

the circumstance of relatively small training dataset, mining the differences between classes is more efficient for constructing a discriminative model for multiuser activity recognition. CHMM is a directed graph-based probabilistic model, a prediction error made in one state may cause a series of errors in the state sequence. During this experiment, we notice that epMAR falls back to the correct state more quickly than CHMM when a misprediction occurs, demonstrating epMAR is more fault tolerant. We illustrate a misdetection case in the snapshot we take as shown in Fig. 11. The x-axis indicates time steps and the y-axis indicates all the 21 activities in our data set collected (their IDs are idential to the IDs in Table 2). In this snapshot, while the ground truth is changed from activities 6 to 12 at time point 11,345, epMAR turns into the correct state after nine time points while CHMM does so after 125 time points.

6.4.5 Training Data Comparison

We evaluate the influence of the amount of training data on accuracy and compare both epMAR and CHMM. We set up the experiments as follows: For the 10 sub data sets obtained, we randomly select one sub data set for testing, and select the remaining sub data sets for training. Initially, we select one training sub data set, we then increase the training data set by each time adding one more sub data set until all the remaining sub data sets are used. The result in Fig. 12 shows that more training data sets achieve better accuracy and the most training data sets achieves the highest accuracy. The figure also shows that epMAR

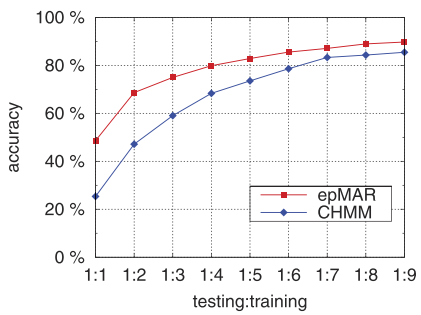


Fig. 12. Training data comparison.

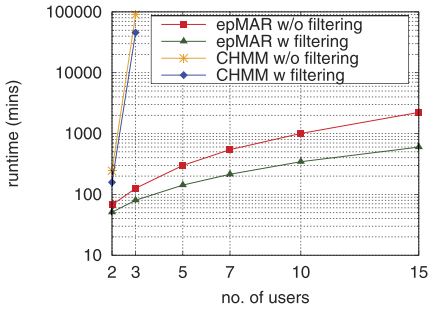


Fig. 13. Scalability comparison.

requires less training data to achieve the same accuracy as compared to CHMM. This result indicates that epMAR has better applicability than CHMM.

6.4.6 Scalability Comparison

We evaluate the scalability of epMAR and compare it with CHMM. The original data set contains only two data streams corresponding to the two users. To generate the data for more users, we randomly select the instances from these two data sets and generate the synthetic testing data sets for 3, 5, 7, 10, and 15 users, respectively. The number of instances for each user in each of these data sets is close. We then run epMAR and CHMM, and the runtime results are shown in Fig. 13. Note that while the x-axis is in a linear scale, the y-axis is in a logarithmic scale. For CHMM, we obtained the results of 2 and 3 users only. As expected, the runtime of CHMM is much higher than that of epMAR. To analyze, in a 2-chain CHMM, the time complexity is $O(TQ^4)$, where T is time and Q is the possible values of a state. For a n-chain CHMM, the time complexity is then $O[TQ^{2n}]$. As analyzed in Section 5.5, the time complexity of epMAR for n users is $l \cdot C_n^2 \cdot (m \cdot O(S) + O(B))$. Fig. 13 also shows that the proximity-based filtering technique greatly reduces the runtime of epMAR, making epMAR more practical for real deployment. Note that this technique can work with any other multiuser activity recognition systems to improve scalability.

6.4.7 Noise Resistance Comparison

We evaluate the noise resistance ability of epMAR and compare it with CHMM. Noise in sensor-based activity recognition can be errors in sensor readings caused by RF interference, corrupted tag IDs due to sensor malfunction, the tag ID of a nearby object since objects may be placed close to each other, and background readings such as audio information generated from other users or the environment. To generate noise, we randomly pick sensor readings such as tag IDs and location data, and insert these readings into the existing data set (note that the existing data set already contains some noise). We then test epMAR and CHMM on data sets with a noise ratio from 0 to 20 percent. The result is shown in Fig. 14. While the accuracies of both models decrease when the noise ratio increased, the gradient of epMAR is much better than that of CHMM. The result probably can be explained as follows: Typically, noise in sensor-based activity recognition has a random distribution. Our EP-based activity model mines the

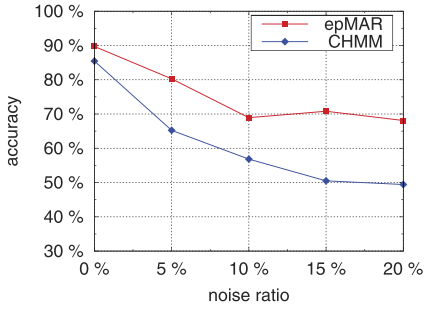


Fig. 14. Noise resistance comparison.

differences of classes, hence random noise can be easily eliminated. epMAR is more resistant to noise than CHMM, and this feature has a particular merit to sensor-based multiuser activity recognition.

7 Conclusions and Future Work

This paper presents the first formal study of using a body sensor network for multiuser activity recognition in a smart home environment. We develop a wireless body sensor network and propose a novel pattern-based approach to recognize both single- and multiuser activities. The results demonstrate that epMAR outperforms CHMM in terms of accuracy, applicability, scalability, and robustness.

As an initial exploration, we demonstrate the effective use of a body sensor network in recognizing multiuser activities. While our result is promising, the design of our sensor network is still premature and far from real-life deployment. The results we obtained are not prefect due to the limitations in wireless and sensing technologies. For future work, we have several directions. First, the types of sensor observations captured are limited. Leveraging on the fast-growing wireless and sensing technologies, we will seek to further develop our sensor nodes to integrate more sensor modalities. Second, data collection in this work was done by two users in a "mock" scenario. A more natural collection done by real users and more than two users are desired. Finally, real-life activities are often more complex than the cases studied in this paper. For example, time-correlated multiuser activities (i.e., one user performs an activity at one point of time and another user continues the activity at another point of time), interleaved activities (i.e., switching between the steps of two or more activities), and concurrent activities (i.e., performing two or more activities simultaneously). Investigating such complex cases can be very challenging while we consider both single- and multiuser activities at the same time, and hence in-depth studies are required.

Acknowledgments

This work was supported by the Danish Council for Independent Research, Natural Science under Grant 09-073281, the National 973 program of China under Grant 2009CB320702, the Natural Science Foundation of China under Grants 60736015, 60721002, and 61073031, and the Jiangsu PanDeng Program under Grant BK2008017.

References

[1] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A Wireless Sensor Network for Target Detection, Classification, and Tracking," *Computer Networks*, vol. 46, pp. 605-634, 2004.

[2] R. Szewczyk, A. Mainwaring, J. Anderson, and D. Culler, "An Analysis of a Large Scale Habit Monitoring Application," *Proc. ACM Second Int'l Conf. Embedded Networked Sensor Systems (SenSys '04)*, 2004.

[3] N. Xu, S. Rangwala, K.K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network for Structural Monitoring," *Proc. ACM Second Int'l Conf. Embedded Networked Sensor Systems (SenSys '04)*, 2004.

[4] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz, and D. Hähnel, "Inferring Activities from Interactions with Objects," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 50-57, Oct.-Dec. 2004.

[5] E.M. Tapia, S.S. Intille, and K. Larson, "Activity Recognition in the Home Setting Using Simple and Ubiquitous Sensors," *Proc. Int'l Conf. Pervasive*, pp. 158-175, 2004.

[6] B. Logan, J. Healey, M. Philipose, E. Munguia-Tapia, and S. Intille, "A Long-Term Evaluation of Sensing Modalities for Activity Recognition," *Proc. Ninth Int'l Conf. Ubiquitous Computing (Ubicomp '07)*, Sept. 2007.

[7] D.J. Patterson, L. Liao, D. Fox, and H.A. Kautz, "Inferring High-Level Behavior from Low-Level Sensors," *Proc. Int'l Conf. Ubiquitous Computing (Ubicomp '03)*, pp. 73-89, Oct. 2003.

[8] D. Jea, R. Balani, J. Hsu, D. Cho, M. Gerla, and M.B. Srivastava, "Diagnostic Quality Driven Physiological Data Collection for Personal Healthcare," *Proc. IEEE 30th Ann. Int'l Conf. Eng. in Medicine and Biology Soc. (EMBC '08)*, Aug. 2008.

[9] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M.B. Srivastava, "Using Mobile Phones to Determine Transportation Modes," *ACM Trans. Sensor Networks*, vol. 6, pp. 1-27, Feb. 2010.

[10] L. Chen and C. Nugent, "Ontology-Based Activity Recognition in Intelligent Pervasive Environments," *Int'l J. Web Information Systems*, vol. 5, pp. 410-430, 2009.

[11] G.Z. Dong and J.Y. Li, "Efficient Mining of Emerging Patterns: Discovering Trends and Differences," *Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining*, pp. 43-52, Aug. 1999.

[12] L. Bao and S.S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Proc. Int'l Conf. Pervasive*, pp. 1-17, 2004.

[13] D. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-Grained Activity Recognition by Aggregating Abstract Object Usage," *Proc. IEEE Int'l Symp. Wearable Computers*, Oct. 2005.

[14] T.Y. Wu, C.C. Lian, and J.Y. Hsu, "Joint Recognition of Multiple Concurrent Activities Using Factorial Conditional Random Fields," *Proc. AAAI Workshop Plan, Activity, and Intent Recognition*, July 2007.

[15] D. Lymberopoulos, A. Ogale, A. Savvides, and Y. Aloimonos, "A Sensory Grammar for Inferring Behaviors in Sensor Networks," *Proc. Fifth Int'l Conf. Information Processing in Sensor Networks*, 2006.

[16] R. Hamid, S. Maddi, A. Bobick, and I. Essa, "Unsupervised Analysis of Activity Sequences Using Event Motifs," *Proc. ACM Int'l Workshop Video Surveillance and Sensor*, 2006.

[17] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831-843, Aug. 2000.

[18] S. Gong and T. Xiang, "Recognition of Group Activities Using Dynamic Probabilistic Networks," *Proc. IEEE Ninth Int'l Conf. Computer Vision (ICCV '03)*, pp. 742-749, Oct. 2003.

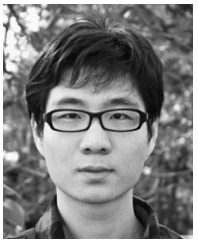
[19] N. Nguyen, H. Bui, and S. Venkatesh, "Recognising Behaviour of Multpile People with Hierarchical Probabilistic and Statistical Data Association," *Proc. 17th British Machine Vision Conf.*, 2006.

[20] S. Park and M.M. Trivedi, "Multi-Person Interaction and Activity Analysis: A Synergistic Track- and Body-Level Analysis Framework," *Machine Vision and Applications: Special Issue on Novel Concepts and Challenges for the Generation of Video Surveillance Systems*, vol. 18, no. 3, pp. 151-166, 2007.

[21] Y. Du, F. Chen, W. Xu, and Y. Li, "Recognizing Interaction Activities Using Dynamic Bayesian Network," *Proc. 18th Int'l Conf. Pattern Recognition (ICPR '06)*, Aug. 2006.

[22] D. Wyatt, T. Choudhury, J. Bilmes, and H. Kautz, "A Privacy Sensitive Approach to Modeling Multi-Person Conversations," *Proc. 20th Int'l Joint Conf. Artifical Intelligence (IJCAI '07),* Jan. 2007.
[23] D.V. Compernolle, "Noise Adaptation in a Hidden Markov Model Speech Recognition System," *Computer Speech and Language,* vol. 3, pp. 151-167, 1989.
[24] M.M.H. Khan, H.K. Le, H. Ahmadi, T.F. Abdelzaher, and J. Han, "Dustminer: Troubleshooting Interactive Complexity Bugs in Sensor Networks," *Proc. Sixth ACM Int'l Conf. Embedded Network Sensor Systems (SenSys '08),* Nov. 2008.
[25] T. Gu, Z. Wu, X. Tao, H.K. Pung, and J. Lu, "epSICAR: An Emerging Patterns Based Approach to Sequential, Interleaved and Concurrent Activity Recognition," *Proc. IEEE Seventh Ann. Int'l Conf. Pervasive Computing and Comm. (PerCom '09),* Mar. 2009.
[26] L. Wang, T. Gu, X. Tao, and J. Lu, "Sensor-Based Human Activity Recognition in a Multi-User Scenario," *Proc. European Conf. Ambient Intelligence (AmI '09),* Nov. 2009.
[27] T. Gu, Z. Wu, L. Wang, X. Tao, and J. Lu, "Mining Emerging Patterns for Recognizing Activities of Multiple Users in Pervasive Computing," *Proc. Sixth Int'l Conf. Mobile and Ubiquitous Systems: Computing, Networking and Services,* July 2009.
[28] U. Fayyad and K. Irani, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning," *Proc. Int'l Joint Conf. Artificial Intelligence,* 1993.
[29] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Proc. 14th Int'l Joint Conf. Artificial Intelligence (IJCAI '95),* pp. 1137-1143, 1995.
[30] J.Y. Li, G.M. Liu, and L. Wong, "Mining Statistically Important Equivalence Classes and Delta-Discriminative Emerging Patterns," *Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining,* pp. 430-439, 2007.
[31] L. Coyle, J. Ye, S. McKeever, S. Knox, M. Stabeler, S. Dobson, and P. Nixon, "Gathering Data Sets for Activity Identification," *Proc. CHI '09 Workshop Developing Shared Home Behavior Data Sets to Advance HCI and Ubiquitous Computing Research,* Apr. 2009.
[32] M. Brand, N. Oliver, and A. Pentland, "Coupled Hidden Markov Models for Complex Action Recognition," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 1997.

Tao Gu received the PhD degree in computer science from the National University of Singapore. He is currently an assistant professor in the Department of Mathematics and Computer Science at the University of Southern Denmark. His research interests include mobile and pervasive computing, wireless sensor networks, and distributed systems. He is a member of the IEEE and the ACM.



Liang Wang received the BSc degree in computer science from Nanjing University in 2007. He is currently a PhD student in the Department of Computer Science at Nanjing University and a visiting research assistant in the Department of Mathematics and Computer Science at the University of Southern Denmark. His research interests include pervasive computing and wireless sensor networks. He is a student member of the IEEE.

Hanhua Chen received the PhD degree in computer science and engineering from the Huazhong University of Science and Technology in 2010. He is now working as a postdoctoral researcher at the University of Southern Denmark. He worked at the Hong Kong University of Science and Technology as a postdoctoral research associate between 2009 and 2010 and as a visiting scholar between 2007 and 2009. His research interests include peer-to-peer computing and wireless sensor networks. He is a member of the IEEE.

Xianping Tao received the PhD degree in computer science from Nanjing University in 2001. He is currently a professor in the Department of Computer Science at Nanjing University. His research interests include software agents, middleware systems, Internetware methodology, and pervasive computing. He is a member of the IEEE.

Jian Lu received the PhD degree in computer science from Nanjing University in 1988. He is currently a professor in the Department of Computer Science at Nanjing University. He is also the director of the State Key Laboratory for Novel Software Technology and the vice director of the Institute of Software Technology at Nanjing University. His research interests include programming methodology, pervasive computing, software agent, and middleware. He is a member of the ACM.