The Sixteenth IEEE International Conference on Embedded and Real-Time Computing Systems and Applications

Real-time Activity Recognition in Wireless Body Sensor Networks: From Simple Gestures to Complex Activities

Liang Wang*, Tao Gu†, Hanhua Chen‡, Xianping Tao* and Jian Lu*
*State Key Laboratory for Novel Software Technology, Nanjing University, China
Email: wangliang@ics.nju.edu.cn, {txp, lj}@nju.edu.cn
†Department of Mathematics and Computer Science, University of Southern Denmark
Email: gu@imada.sdu.dk
‡Services Computing Technology and System Lab,
Cluster and Grid Computing Lab, Huazhong University of Science and Technology
Email: chenhanhua@hust.edu.cn

***Abstract*—Real-time activity recognition using body sensor networks is an important and challenging task and it has many potential applications. In this paper, we propose a real-time, hierarchical model to recognize both simple gestures and complex activities using a wireless body sensor network. In this model, we first use a fast, lightweight *template matching* algorithm to detect gestures at the sensor node level, and then use a *discriminative pattern* based real-time algorithm to recognize high-level activities at the portable device level. We evaluate our algorithms over a real-world dataset. The results show that the proposed system not only achieves good performance (an average precision of 94.9%, an average recall of 82.5%, and an average real-time delay of 5.7 seconds), but also significantly reduces the network communication cost by 60.2%.**

***Keywords*-Real-time activity recognition; gestures and high-level activities; wireless body sensor networks.**

I. Introduction

Sensor-based human activity recognition has recently attracted much attention in pervasive computing. In this paradigm, various sensors are typically attached to a human body or embedded in the environment. Sensor observations are collected in the form of continuous sensor data stream, and the data stream is then interpreted by a recognition system. The computation usually involves two phases: 1) sensor observations are used to train an appropriate activity model; 2) the trained model will then be used to predict activities for new observations.

Sensor-based activity recognition has many potential applications, including health care [1], [2], assisted living [3], sports coaching [4], and interactive games [5], [6]. In the past few years, many efforts have been devoted to this task in various domains by researchers and industrial participants. However, we have not seen many real applications being deployed in our daily lives. A number of important and challenging issues still remain unsolved. First, a practical recognition system should be able to recognize activities in a real-time manner. The real-time requirement demands for one-pass algorithms over sensor data streams with a short real-time delay. Multiple passes are usually not possible due to a large volumn of sensor data stream arrive continuously at a processing server. Second, most of the wireless body sensor networks typically use a star topology in which data generated from each sensor node are transmitted to a centralized server for further processing. The network communication can be very costly due to high sampling rates of motion sensors such as 3-axis accelerometers. Third, processing sensor data stream at a fix server may not be practical since humans often move from one place (e.g., home) to another (e.g., office) in their daily lives. In this scenario, mobile and portable devices are more suitable for the task, and hence lightweight and portable recognition solutions are highly desired.

To address the above challenges, in this paper we propose a hierarchical model to recognize human activity recognition in real time, which we first identify simple gestures at a sensor node level, and then recognizes high-level activities from these gestures at the portable node level. Our motivation is that a high-level activity typically includes a sequence of physical gestures and ambulation in the execution. For example, household cleaning can be better derived from a sequence of hand gestures (i.e., wiping and mopping patterns), body gestures (i.e., up and down patterns), and ambulation. In addition, the hierarchical model enables us to distribute the computation from a centralized server to individual sensor nodes so that the network communication cost can be significantly reduced. In this work, we first design a wireless body sensor network consisting of a number of wireless sensor nodes attached to a subject for collecting sensor observations. We then design our real-time recognition algorithms which operate in two stages. First, acceleration data are processed immediately at each sensor node by a fast, lightweight gesture recognition algorithm to detect the gestures of a subject. This is done by discovering a template for each simple gesture using an unsupervised method, and then matching the acceleration data stream with an appropriate template based on the minimum distance

1533-2306/10 $26.00 © 2010 IEEE
DOI 10.1109/RTCSA.2010.19

which is computed using Dynamic Time Warping (DTW) [7]. This algorithm outputs the gestures of both hand such as moving hand up and down [8] and the body such as walking and running [9]. Second, recognized gestures and other sensor readings (i.e., RFID tagged object and location) will be transmitted over the wireless network to a centralized device for further processing. We propose a real-time, discriminative pattern based approach to recognize high-level, complex activities. We adapt an off-line, Emerging Pattern based algorithm [10] which is capable of recognizing both simple activities (e.g., cooking and cleaning [11], [12]) and complex activities (i.e., interleaved [13] and concurrent activities [14])—to meet the real-time requirement in this work. We use a real-world dataset and develop a real-time simulator to evaluate our proposed algorithms. Our experimental studies show that the proposed system is promising in recognizing both gestures and activities in real time for mobile devices.

In summary, the paper makes the following contributions:

- To the best of our knowledge, this paper presents the first formal study of a real-time, hierarchical recognition model to recognize both physical, simple gestures and high-level, complex activities using a wireless body sensor network.
- The proposed algorithms are properly designed with respect to not only the real-time constraint, but also the lightweight constraint so that they can be deployed at sensor nodes and mobile devices.
- We conduct comprehensive experiments, and the results show that our algorithms achieve not only good performance in terms of recognition accuracy and real-time delay, but also better communication efficiency.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we present our body sensor network and provide an overview of our proposed system. We present our algorithm for gesture recognition in Section 4, followed by the algorithm for high-level activity recognition in Section 5. Section 6 reports our empirical studies, and finally Section 7 concludes the paper.

II. Related Work

Researchers are recently interested in recognizing activities using wireless body sensor networks. In such a sensor network, various sensors are used to directly measure user's movement (e.g., 3-axis accelerometer), the living environment (e.g., temperature, humidity and light sensors), object use (e.g., wrist worn RFID sensor) and user location (e.g., indoor location sensor).

Most of the existing work [8]–[17] in gesture or activity recognition are done in an off-line manner. There are some recent work focusing on real-time activity recognition. Tapia et al. [18] proposed a real-time algorithm based on decision tree for recognition of physical activities (i.e., gestures). The sensor readings from 3-axis accelerometer sensors are transmitted wirelessly to a laptop computer for processing. A C4.5 classifier is first trained, and then used to recognize gymnasium activities in real time. Krishnan et al. [19] proposed an AdaBoost algorithm based on decision stumps for real-time classification of gestures (i.e., walking, sitting and running) using 3-axis accelerometer sensors. He et al. [20] presented a Hidden Markov Model approach for real-time activity classification using acceleration data collected from a wearable wireless sensor network. The model is used to classify a number of gestures such as standing, sitting, and falling.

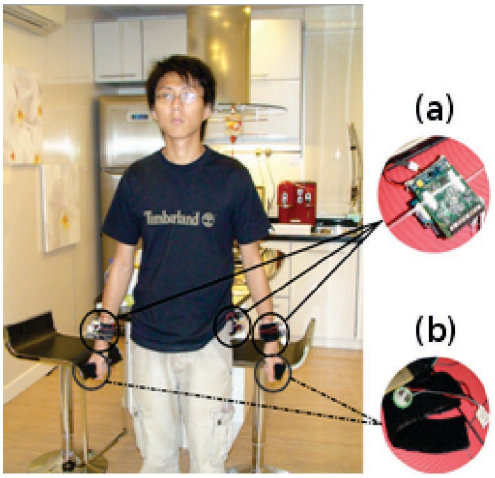

Figure 1. Our body sensor network, (a) an IMOTE2 mote, (b) an RFID reader mote.

Some recent work has been done to recognize gestures or activities in real time on resource-constraint devices. Györbíró [21] presented a real-time mobile activity recognition system consisting of wireless body sensors, a smartphone, and a desktop workstation. A sensor node has an accelerometer, a magnetometer, and a gyroscope. They proposed a recognition model based on feed-forward back-propagation neural networks which are first trained at a desktop workstation, and then run at the smartphone to recognize six different gestures. Liu et al. [22] proposed an efficient gesture recognition method based on a single accelerometer using DTW. They first define a vocabulary of known gestures based on training, then use these pre-defined templates for recognizing hand gestures.

Different from the above work which use a single layer model (i.e., a single point for data stream processing) for activity recognition, we propose a distributed approach in which the computation is divided to gesture recognition at sensor nodes and high-level activity recognition at a mobile device. Similar to the DTW-based hand gesture recognition algorithm proposed in [22], we use a similar approach to compute the distance of a test instance and a gesture template. However, their pre-defined templates are obtained by a training process in a supervised manner whereas we

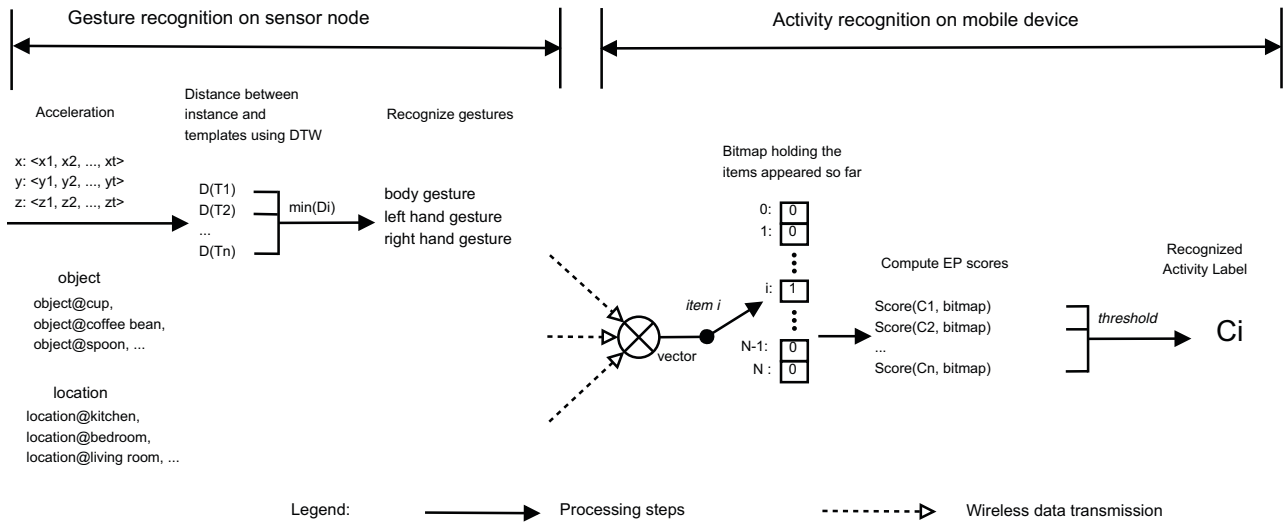

Figure 2. Overview of our real-time, hierarchical recognition model

obtain various hand and body gesture templates using an unsupervised method. The evaluation of existing activity recognition systems mainly focused on accuracy and real-time performance. In addition to these measurements, we evaluate the communication efficiency and the portability of our system which are important for real-life deployment.

III. Body Sensor Network Design and System Overview

We design a wireless body sensor network as shown in Fig. 1. It consists of five sensor nodes—three IMOTE2 motes and two RFID reader motes. An IMOTE2 mote is located on each wrist and the body of a subject to capture hand and body movement; it consists of an IPR2400 processor/radio board and an ITS400 sensor board with a tri-axis accelerometer, as shown in Fig. 1(a). 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. 1(b). An RFID reader mote is able to detect the presence of a tagged object within a few centimeters. In addition, detecting user 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. The sensor data captured by these motes can be transferred to sink nodes and logged in servers.

Figure 2 gives an overview of our real-time activity recognition system. The system operates in two stages—gesture recognition at sensor nodes and activity recognition at a mobile device. First, each IMOTE2 mote processes its acceleration data to recognize hand or body gestures by a fast and lightweight *template matching* algorithm. This is done by first obtaining a specific template for each gesture pattern using an unsupervised method, then matching a test instance obtained by applying a sliding window over the data stream with each possible template. A match is found when the distance between the test instance and a template is minimum, and then the test instance will be assigned with the corresponding template label. We compute the distance using Dynamic Time Warping which is an efficient, lightweight algorithm to match two time series samples. Next, recognized gestures, tagged objects and user locations from each node will be transmitted over the wireless network to a centralized device. These data will be synchronized and processed to generate a discrete vector stream with a fix time interval. We then apply a discriminative pattern based approach to recognize complex, high-level activities in real time. A bitmap is used to temporarily hold the data before they can be recognized. When a new vector comes, we map items in the vector into the bitmap and compute the score between the input data and discriminative patterns mined for each class C_i. If the score of one class exceeds a predefined *threshold*, we output that class as the recognized activity and clear the bitmap.

IV. Gesture Recognition

This section describes how we process acceleration data at the sensor node level to recognize the hand and body gestures of a subject.

A. Sensor Data Collection

In our body sensor network, we use three IMOTE2 motes with 3-axis accelerometers—one on each wrist to capture the hand motion patterns; one on the body to capture the body motion patterns. An acceleration data stream is generated at each node with a constant sampling rate, the data format is shown as follows:

$$[time\_stamp] < sensor\_id >< x >< y >< z >$$

where $time\_stamp$ denotes time stamp, $sensor\_id$ denotes sensor node ID, x, y and z are acceleration readings on the three different directions and they can be decoded in a 12-bits resolution ranging from -2g to +2g. These data can be transformed to a three dimensional stream of integers containing readings of the three axes using a simple parser. A sample taken from the data stream we collected is shown as follows:

$$[12/08/2008 \quad 13:22:24:765] \quad 163 \quad -664 \quad -306 \quad 612$$

In this example, 163 represents the sensor ID on the subject's right hand.

B. Gesture Templates

To recognize various gestures over acceleration data, we first need to define a set of gesture templates for left hand, right hand and body, respectively. A common way to obtain these templates is based on supervised learning, e.g., the work done in [22]. Using this method, the training data for different hand and body gestures is collected and assigned with proper labels, and will then be used to define a template for each gesture. However, in real deployment, labeling training data for hand and body gestures can be very time consuming since an annotator has to analyze the video record for each gesture, and sometime it is not possible if hand motions are blocked from the video camera. In addition, the accuracy of labeling various gestures remains uncertain because there is no common vocabulary for all the gestures performed in real life.

In this work, we propose an unsupervised method to discover gesture patterns. We use a K-Medoids clustering method to discover these template gestures. This method finds the k representative instances which best represent the clusters. The number of clusters is set to five for body gestures and ten for each hand in our study. The intuition behind this setting is that we observe ten typical patterns for hand movements—moving forward, backward, left, right, left and up, left and down, right and up, right and down. Similarly for the body gestures, there exist typically five patterns—moving up, sitting down (contain both moving down and backward), moving left, right and forward.

C. Identifying Gestures

We apply a *template matching* algorithm to identify the gestures from the acceleration data stream based on the templates we obtained. To get test instances, we use a sliding window with the fix length of 1 second to segment the data stream. For each instance obtained, we match the instance with the pre-defined templates using DTW. DTW is a classic dynamic programming based algorithm to match two time series with temporal dynamics which had shown its effectiveness in recognizing hand gestures using a predefined vocabulary [22]. We use the Euclidean distance as the function of calculating the distance between two time samples. A match is found when the distance between the test instance and a gesture template is minimum, and then the test instance will be assigned with the corresponding template label. The process of our *template matching* algorithm is illustrated in Fig. 3.

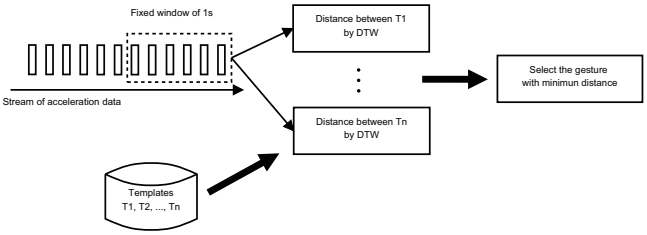

Figure 3. Gesture recognition using DTW.

D. Time and Space Complexity analysis

We analyze the time and space complexity for our *template matching* algorithm. Let $S[1...M]$ and $T[1...N]$ denote two time series, n denotes the number of templates. The time and space complexity of matching S and T are both $O(M \cdot N)$. The total time complexity of recognition is thus $O(n \cdot M \cdot N)$, the space complexity is $O(M \cdot N)$.

V. Complex and High-level Activity Recognition

The recognized gestures, tagged objects and user locations from each each node will be transmitted over the wireless network to a centralized device for recognizing complex, high-level activities. We adapt an of ine, Emerging Pattern based algorithm [10] to real-time requirements in this work. In this section, we first give the background of Emerging Pattern, and then describe how to use Emerging Pattern to recognize complex, high-level activities in real time.

A. Background of Emerging Pattern

Emerging Pattern (EP) describes significant differences between different classes of data [23]. An EP is a set of items, and it occurs frequently in one class and rarely in all the other classes. The class in which an EP occurs the most frequently is called the class of the EP. An EP can be viewed as a representative pattern of its class. If an instance contains an EP, then it is very likely that the instance belongs to the class of the EP. EPs have been successfully applied in various domains for gene classification [24] and off-line activity recognition [10], [25].

Formally, an EP is defined as follows. Let $D = \{t_1, t_2, \cdots, t_n\}$ be a dataset containing a set of instances, and each instance is a set of items. In our case, an item can be a user gesture, an object touched by users or the location of the user. Each instance has a class label which indicates the activity of the user. Let $\mathcal{C} = \{C_1, C_2, \cdots, C_k\}$ be the set of class labels. A pattern X is an itemset, and its support

in D is defined as the proportion of instances in D that contain it, denoted as $supp_D(X) = |\{t|X \subseteq t, t \in D\}|/|D|$. The discriminative power of an itemset X is measured by the ratio of the support of X in the dataset of target class c to the support of X in all the other classes, denoted as $GrowthRate_D(X) =$

$$\begin{cases} 0, & if \quad supp_{D_c}(X) = 0 \\ \infty, & if \quad supp_{D_c}(X) > 0 \ and \\ & supp_D(X) = supp_{D_c}(X) \\ \frac{supp_{D_c}(X)}{supp_D(X) - supp_{D_c}(X)}, & otherwise \end{cases}$$

where c is the class of X, and D_c is the set of instances belonging to class c.

Definition 1 [Emerging Pattern] Given a labeled dataset D, if $supp_D(X) \geq min\_sup$ and $GrowthRate_D(X) \geq \rho$, then X is called a $\rho - EmergingPattern$, where min_sup is a predefined minimum support threshold and ρ is a predefined minimum growth rate threshold.

B. Mining Emerging Patterns

To use EPs for activity recognition, we first obtain a set of EPs for each activity class. This is done by mining EPs from a training dataset containing labeled activity instances. For each activity C_i, we mine a set of EPs that occur frequently in C_i, but rarely in other classes. We denote this set of EPs as EP_{C_i}. We discover the EPs by an efficient algorithm described in [26]. An example of EP for the *brushing hair* activity is shown as follows.

$$\{object@comb, \quad gesture@body\_forward, \\ gesture@right\_forward\_upward, \\ gesture@left\_forward, \quad location@bathroom, \\ object@detangling\_spray\}$$

There are usually many EPs with different growth rates being discovered for each activity. To reduce the computation cost, we only select the EPs with the growth rate of $+\infty$ (i.e., the maximum discriminative power) for our recognition algorithm described in the next section.

C. EP-based Real-time Activity Recognition

The of ine recognition algorithm [10] use EPs to recognize complex, high-level activities. Such activities can be performed in a sequential (i.e., one activity after another), interleaved (switching between the steps of two or more activities), or concurrent (i.e., performing two or more activities simultaneously) manners. Although it is effective, it works off-line and there are at least two scans over the data stream. In real-time activity recognition, multiple scans are not possible. Thus, this algorithm cannot be directly applied in this case. We extend this algorithm and design a fast EP-based algorithm for real-time activity recognition as follows.

First, gesture, object and location data will be synchronized and processed to generate a discrete vector stream with a one second interval. A vector has the following form:

$$< body\_gesture, left\_gesture, right\_gesture, \\ left\_object, right\_object, location >$$

We then map every item in a vector to an integer. A bitmap is used to hold the items that have appeared so far. The ith bit in the bitmap is 1 if item i has appeared, otherwise, it is 0. Initially, all the bits in bitmap are set to 0. When a new vector is generated, all the bits corresponding to the items in the vector are set to 1.

Next, given the bitmap contains an EP, say X, which belongs to activity class C_i, we define a score function to measure the contribution of X as follows.

$$Score(C_i, bitmap) = \\ \sum_{X \subseteq bitmap, class(X) = C_i} \frac{GrowthRate(X)}{GrowthRate(X) + 1} \quad (1)$$

where $class(X)$ is the class of X. This score provides an indication of the conditional probability that the activity class is C_i, given it contains X [27]. If there exists an activity C such that $Score(C, bitmap)$ is higher than a predefined threshold, then class C will be the output as the recognized activity and the bitmap is cleared by setting all the bits to 0. If the scores for all the possible activities are below the threshold, then it outputs nothing and waits for a new vector. The computation is done recursively until the end of the sensor data stream. The entire process is described in Algorithm 1.

Algorithm 1 EP-based Real-time Algorithm

Input: a feature vector sequence $V = \{v_1, v_2, ..., v_T\}$ with a length of T;
activities $\{C_1, C_2, \cdots, C_m\}$.
Output: recognized activity sequence.

```
1:  Bitmap bitmap;
2:  for t = 1 to T
3:     for each item in v_t
4:        bitmap[key(item)] = 1;
5:     end for
6:     for i = 1 to m
7:        if Score(C_i, bitmap) > threshold then
8:           Recognize the current activity as C_i;
9:           Set all elements in the bitmap to 0;
10:       end if
11:    end for
12: end for
```

Table I
ACTIVITIES PERFORMED

0	making coffee	13	ironing
1	making tea	14	eating meal
2	making oatmeal	15	drinking
3	frying eggs	16	taking medication
4	making a drink	17	cleaning a dining table
5	applying makeup	18	vacuuming
6	brushing hair	19	taking out trash
7	shaving	20	using phone
8	toileting	21	watching TV
9	brushing teeth	22	watching DVD/movies
10	washing hands	23	using computer
11	washing face	24	reading book/magazine
12	washing clothes	25	listening music/radio

D. Time and space complexity analysis

Let $V[1...n]$ be the whole input vector sequence generated in the previous step, k be the number of activities in our system. Every time an input vector comes, we compute the score for each of the k activities. Let m be the number of EPs mined and l be the average number of items contained in EPs. The time complexity of matching EPs with items stored in the bitmap is $O(m \cdot l)$. After all the EPs have been checked, we check which class has a score no less than the threshold. The cost of this step, is $O(k)$. Since we only make one pass through the input vector sequence, the time complexity of recognizing the whole sequence is then $O((m \cdot l + k) \cdot n)$.

Let N be the total number of items in our system. The space cost by holding the bitmap is $\Theta(N)$. The space cost for holding the mined EPs is $\Theta(m \cdot l)$.

VI. EMPIRICAL STUDIES

We evaluate our proposed system in this section. We are interested in several aspects of performance evaluation. The fundamental question is *how accurately we can recognize activities*. Since the proposed hierarchical recognition system aims for real-time recognition at sensor nodes and mobile devices, it is critical to know *how fast we recognize both simple gestures and high-level activities*; and *how much resources are required to run the system*. Finally, reducing the network communication cost is one of the goals in our design, hence we will find out *how the entire network traffic can be reduced using our model*.

A. Real-world Dataset

We use the activity dataset collected in our previous work [10]. The data collection was done by four volunteers in a smart home over a period of two weeks. Each day, one of the volunteers wore a set of wireless sensors (shown in Fig. 1) and performed a list of 26 activities, as summarized in Table I. There was only one subject performing activities at any given time. The dataset contains both simple and complex activity cases across a variety of activities in a real-world situation. Out of 26 sequential activities, there are 15 interleaved activities (e.g., *using computer* and *using phone* can be performed in an interleaved manner), and 16 concurrent activities (e.g., *brushing teeth* while *listening music/radio* can be performed concurrently). There is a total number of 532 activity instances, and only sequential activity instances will be used for training our EP-based activity model.

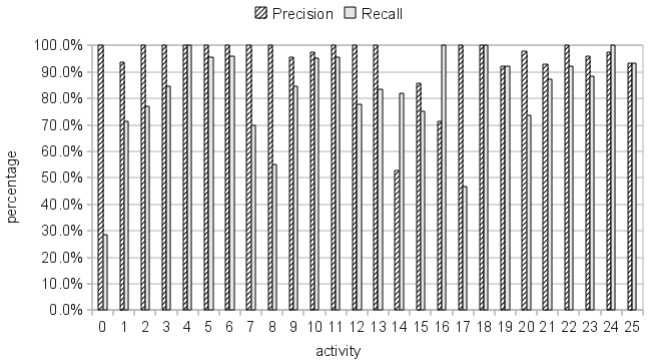

Figure 7. Precision and recall (The numbers in the X-axis are identical to the numbers in Table I)

B. Real-time Simulator

We build a real-time simulator to simulate the behavior of each sensor node, e.g., generation of continuous sensor data stream. There are a total number of six sensor nodes—three accelerometers, two RFID wristband readers and one location sensor. Each sensor node is able to generate sensor readings at an adjustable sampling rate. We implemented the gesture recognition algorithm at each accelerometer sensor node using the simulator. The recognized gestures together with objects and locations will be transferred continuously to a simulated mobile device in which the EP-based real-time recognition algorithm runs to recognize complex, high-level activities.

C. Accuracy Performance

To evaluate the gesture recognition algorithm, Table II shows the gesture templates we discovered from the acceleration data stream of a subject's left hand using our clustering method. To visualize the gesture templates obtained in Table II, we show the traces of the left-hand movements in a 3-D space in Fig. 4, assuming the initial position of the hand is at the origin of the coordinate system. Through the 3-D visualization, it will be easily to figure out what each template represents in a physical world. Basically,

Table II
TEMPLATES OF LEFT HAND GESTURES

$x :< -216, -42, -36 >$ $y :< 249, -48, 0 >$ $z :< 815, 985, 988 >$
$x :< 66, -95 >$ $y :< 1008, 1001 >$ $z :< 145, -82 >$
$x :< 605, 455, 442, 389 >$ $y :< 710, 555, 442, 442 >$ $z :< 566, 782, 796, 719 >$
$x :< 241, 92, 658 >$ $y :< -269, -395, -70 >$ $z :< 862, 749, 717 >$
$x :< -141, -169 >$ $y :< 736, 828 >$ $z :< -668, -562 >$
$x :< -922, -972 >$ $y :< 283, 38 >$ $z :< 284, 220 >$
$x :< 80, -40, -87 >$ $y :< 491, 665, 905 >$ $z :< 852, 812, 580 >$
$x :< 901, 879, 935 >$ $y :< 146, -5, 44 >$ $z :< 458, 316, 372 >$
$x :< 871, 905, 880, 853, 802 >$ $y :< 469, 489, 495, 565, 572 >$ $z :< -260, -260, -183, -255, -202 >$
$x :< -4, -21, -18, 82, -175, 7 >$ $y :< -896, -1103, -1054, -1168, -963, -1114 >$ $z :< 96, 138, 133, 93, 52, 82 >$

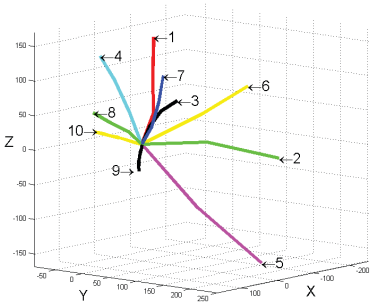

Figure 4. Traces of templates for left hand gestures.

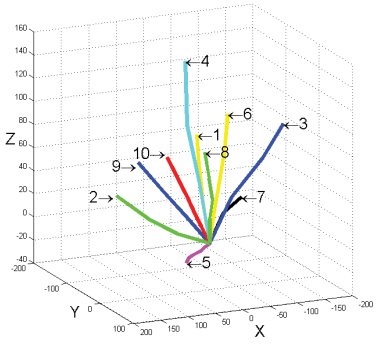

Figure 5. Traces of templates for right hand gestures.

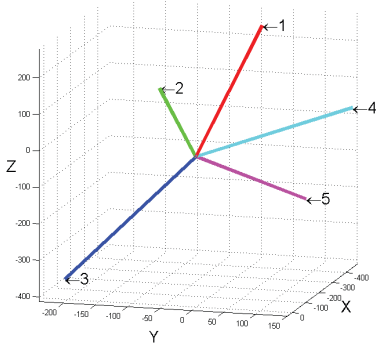

Figure 6. Traces of templates for body gestures.

these templates represent the different directions of left-hand movement in a physical space. Gestures 1, 3, 4 and 7 basically represent that the hand moves upward. By taking a closer look at the figure, gesture 1 represents *moving straight up*, gesture 7 represents *moving up and right*, gesture 4 represents *moving up and left* (i.e., opposite to gesture 7), and gesture 3 represents *moving up and forward*. Gestures 8 and 9 basically represent that the hand moves forward. While gesture 8 represents *moving forward and left*, gesture 9 represents *moving forward*. The rest of gestures are quite obvious, gesture 2 represents *moving right*, gesture 10 represents *moving left* and gesture 6 represents *moving back*. Gesture 5 represents *putting down* in which the hand movement patterns involve both *moving down* and *moving back*. It matches the natural pattern well since our arm is actually turning around the shoulder rather than going straight down when we put down our hands. Similarly for the right hand, as shown in Fig. 5, gesture 1 and 8 basically represent *moving up and left*. Gesture 7 represents *moving forward and left*. Gesture 9 and 10 represent *moving up and back*. Gesture 2 represents *moving back*. Gesture 5 represents *putting down*. Gesture 4 represents the hand movement of *moving up*. Gesture 3 represents *moving up and right*. Finally, Gesture 6 represents *moving up and forward*.

We obtain similar results for the templates of body gestures, as visualized in Fig. 6. Obviously, gestures 1 to 5 represent body *moving up*, *moving left*, *sitting down*, *moving forward* and *moving right*, respectively. It is interesting to analyze gesture 3 which involves two directions of the body movement, i.e., both backward and downward. Such movement pattern is likely to happen when we sit down. In the case, our body not only goes downward, but also goes backward when we lean our knees towards a chair.

We then evaluate the accuracy performance of recognizing complex, high-level activities. We use two common metrics for evaluating real-time activity recognition systems—*precision* and *recall*. Precision is the probability that a given inference about that activity is correct, i.e., $\frac{TP}{TP+FP}$. Recall is the probability that an activity recognition system correctly infers a given true activity, i.e., $\frac{TP}{TP+FN}$. We use ten-fold cross-validation [28] for our evaluation. Figure 7 shows the precisions and recalls for all the 26 activities. On average, our system achieves a precision of 94.9% and a recall of 82.5%.

To analyze the result in detail, we present the confusion matrix as shown in Table III. The columns show the predicted activities, and the rows show the ground-truth

Table III
CONFUSION MATRIX.

		Ground Truth Activities																									
		0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*
Predicted Activities	*0*	4																									
	1	1	15																								
	2			10																							
	3				11																						
	4					17																					
	5						21																				
	6							24																			
	7								14																		
	8									11																	
	9										22		1														
	10						1					39															
	11												22														
	12													14													
	13														20												
	14															18			16								
	15																18					1			2		
	16		3						5									20									
	17																		15								
	18																			15							
	19										1										12						
	20																				1	45					
	21									1													27	1			
	22																							12			
	23											1													23		
	24								1																	41	
	25									1		1							1				1				55
	miss	10	6	5	3		2	1	1	9	6	3		5	5	7	7			1	1	18	5		2	1	8

activities. The last row shows the number of instances that is miss detected for each activity. From the confusion matrix, we observe two main cases as shown as follows.

Miss detection: This is the case which an activity instance performed by a subject is not detected by the system. For example, for *making coffee* (i.e., activity 0 in the table), only four of fifteen instances is correctly recognized while ten instances are missed. The missing rate is 66.7%. This leads to a lower recall for this activity. By analyzing the ground truth, we found that *making coffee* is often performed with another activity such as *making tea* or *using phone* in an interleaved manner. For example, a case in the dataset shows the phone rang in the middle of making coffee, and the subject then paused *making coffee* and went to pick up the phone. When the system recognizes the *using phone* activity, it clears the bitmap resulting in a loss of the initial data for *making coffee*. Hence, when the subject came back for making coffee again, the system may no long recognize it since some data are lost. This example shows that our real-time EP-based algorithm has certain limitation in dealing with more complex cases such as interleaved activities.

False detection: This is the case which an activity is recognized as another activity. For example, 50% of *cleaning a dining table* is recognized as *eating meal*, resulting in a lower precision for *eating meal* and a lower recall for *cleaning a dinning table*. It probably can be explained as follows. These two activities share many common features, i.e., they are performed using similar objects, with similar gestures, and in the same location. With the existing sensor features, it is difficult to discriminate these two activities. One possible solution is to make use of the sequence information of hand and body gestures and objects which we leave for our future work.

D. Real-time Recognition Delay and Storage Cost

In this experiment, we evaluate the real-time recognition delay and storage cost of our system. The real-time delay (i.e., runtime) measures, for a particular activity, the time from the generation of the first sensor reading to the recognition of an activity label. The result show that our light-weight gesture recognition algorithm has a real-time delay of 25.5 μs on the sensor node. The storage cost of the gesture recognition algorithm consists of 2.4 KB for storing gesture templates and 64 B for storing the algorithm. The result implies that it is feasible to deploy gesture recognition at the sensor node level.

The real-time recognition delay of the EP-based real-time activity recognition algorithm is 5.7 s and the storage cost is less than 10 MB. This result shows the potential to deploy this algorithm in a mobile device such as a PDA or a smartphone.

E. Communication Cost Analysis

In this experiment, we analyze how the network communication cost can be reduced using our hierarchical activity recognition model. We compare the recognition model with and without a hierarchical design. In a single layer recognition model, all the sensor readings generated at each accelerometer node will be transferred over wireless links. The total amount of data transmitted on the network in one second can be computed as follows.

$$D = \sum_{i=1}^{n} \lceil \frac{f_i \cdot p_i}{m_i} \rceil \cdot (m_i + o_i) \quad (2)$$

where n is the number of sensor nodes, f_i is the sampling rate of the ith sensor node, m_i is the designed payload size for each packet, p_i is the size of each reading of the ith sensor and o_i is the overhead of sensor node i sending a packet. It is clear that $f_i \cdot p_i$ computes the total size of sensor readings of ith sensor node that is to be transmitted for each second. By taking the ceiling of $\frac{f_i \cdot p_i}{m_i}$, we get the number of packets that is sent by the ith node in one second. Finally, by multiplying the number of the packets and the size of each packet, which can be easily computed by $m_i + o_i$, we get the total number of bits transmitted for sending the ith sensor node's data in one second. Finally, the data transmitted in the entire network in one second is the sum of the data transmitted for all sensor nodes.

We have three accelerometers, two RFID sensors and one location sensor in our system. ZigBee radio is used by the sensors for wireless data transmission. The packet header size for ZigBee / IEEE 802.15.4 protocol is 120 bits. Each accelerometer sensor node has an average sampling rate of 8 Hz. Each reading has a size of three 16-bit integers (i.e., readings on the three axes) which is 48 bits in total. The packet payload size is set to be 10 readings which is 480 bits. Each RFID sensor or location sensor has a sampling rate of 1 Hz. Each reading has a size of 64 bits, which is the size of one tag ID. The packet payload size is set to 64 bits. Thus, the total amount of data transmitted in the entire system in one second is 2352 bits according to Equation 2.

In our hierarchical recognition model, we only need to transfer 1 byte of data containing a gesture label in every one second over wireless links since the acceleration data stream is processed immediately by the gesture recognition algorithm. The total amount of data transmitted for each accelerometer in one second is 128 bits (8 bits for the gesture label and 120 bits for the packet header). While the RFID and the location sensors remaining the same, the total amount of bits transmitted in the system in one second is reduced to 936 bits. Hence, we reduce the total communication cost by 60.2%. Through the above analysis, we demonstrate that a hierarchical recognition model is more appropriate for real-time activity recognition using a wireless sensor network which typically has a limited network bandwidth.

VII. Conclusion

In conclusion, this paper proposes a real-time, hierarchical model based on a wireless body sensor network to recognize human activities from physical, simple gestures to complex, high-level activities. At the sensor node level, acceleration data are processed immediately by a fast and lightweight gesture recognition algorithm for recognizing both hand and body gestures. The recognized gestures, object and location information will be transferred to a centralized device, and then processed by an EP-based real-time algorithm to recognize complex high-level activities. Our experimental studies show the proposed system achieves good performance in accuracy and real-time recognition delay, and better communication efficiency.

While the real-time, hierarchical model presented in this paper is promising, the entire system is still premature and far from real-life deployment. One limitation of our system is that although it shows a low average delay in recognition, it is not guaranteed that the system can always respect the real-time constraints given by the users. Another limitation is that we assume a perfect link qulaity which involves no packet loss and all nodes are working under a globle clock. In real life, the link quality may vary, the clock may drift among sensor nodes. These constraints are important factors that affect the system's delay and accuracy.

In our future work, we plan to extend this work in several directions. First, we will further develop our proposed algorithms to investigate the upper bound of the system's runtime which guarantees the real-time performance. Second, we plan to deploy our algorithms in sensor nodes and mobile devices for real-life trials, and conduct more evaluations to study its real-time behaviors and investigate its limitations. Finally, 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 such as physiological sensors, pressure sensor, and possibly integrate RFID tags or readers with motes as suggested in [29] to improve robustness.

References

[1] W. Chen, D. Wei, X. Zhu, M. Uchida, S. Ding, and M. Cohen, "A mobile phone-based wearable vital signs monitoring system," in *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on*, 2005, pp. 950–955.

[2] J. Mäntyjärvi, P. Alahuhta, and A. Saarinen, "Wearable sensing and disease monitoring in home environment," in *Workshop on ambient intelligence technologies for wellBeing at home. Held in conjunction with 2nd European symp. on ambient intelligence. EUSAI, Eindhoven*. Citeseer, 2004.

[3] J. Nehmer, M. Becker, A. Karshmer, and R. Lamm, "Living assistance systems: an ambient intelligence approach," in *Proceedings of the 28th international conference on Software engineering*. ACM New York, NY, USA, 2006, pp. 43–50.

[4] R. Smith, N. ZANE, F. Smoll, and D. COPPEL, "Behavioral assessment in youth sports: coaching behaviors and children's attitudes." *Medicine & Science in Sports & Exercise*, vol. 15, no. 3, p. 208, 1983.

[5] W. Cai, P. Xavier, S. Turner, and B. Lee, "A scalable architecture for supporting interactive games on the internet," in *Proceedings of the sixteenth workshop on Parallel and distributed simulation*. IEEE Computer Society Washington, DC, USA, 2002, pp. 60–67.

[6] A. Gumina, "Interactive games and method of playing," May 3 2001, uS Patent App. 09/847,336.

[7] H. Sakoe and S. Chiba, "A dynamic programming approach to continuouspeech recognition," in *Proc. 7th Int. Congress Acoust*, Budapest, Hungary, Paper 20C-13, 1971.

[8] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and S. Marca, "Accelerometer-based gesture control for a design environment," *Personal and Ubiquitous Computing*, vol. 10, no. 5, pp. 285–299, 2006.

[9] L. Bao and S. Intille, "Activity Recognition from User-Annotated Acceleration Data," in *Porc. Intl Conf. Pervasive 2004*, vol. LNCS 3001, 2004, pp. 1 – 17.

[10] 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," in *Proc. of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (Percom '09)*, Galveston, Texas, March 2009.

[11] T. Huynh, U. Blanke, and B. Schiele, "Scalable recognition of daily activities with wearable sensors," *Lecture Notes in Computer Science*, vol. 4718, p. 50, 2007.

[12] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hähnel, "Inferring activities from interactions with objects," in *IEEE Pervasive Computing*, October 2004.

[13] J. Modayil, T. Bai, and H. Kautz, "Improving the recognition of interleaved activities," in *Proc. Int'l Conf. Ubicomp*, Seoul, South Korea, September 2008.

[14] T. Wu, C. Lian, and J. Hsu, "Joint recognition of multiple concurrent activities using factorial conditional random fields," in *AAAI Workshop PAIR 2007*, 2007.

[15] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille, "A long-term evaluation of sensing modalities for activity recognition," in *Proc. Int'l Conf. Ubicomp*, Innsbruck,Austria, September 2007.

[16] J. A. Ward, P. Lukowicz, G. Tröster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," in *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, October 2006.

[17] P. Palmes, H. Pung, T. Gu, W. Xue, and S. Chen, "Object relevance weight pattern mining for activity recognition and segmentation," *Pervasive and Mobile Computing*, pp. 43–57, 2010.

[18] E. M. Tapia, S. Intille, and K. Larson, "Real-Time Recognition of Physical Activities and Their Intensities Using Wireless Accelerometers and a Heart Rate Monitor," in *Proceedings of the 11th International Conference on Wearable Computers*. Boston, MA, 2007.

[19] N. C. Krishnan, D. Colbry, C. Juillard, and S. Panchanathan, "Real Time Human Activity Recognition Using Tri-Axial Accelerometers," in *Proceedings of Sensors Signals and Information Processing Workshop*. Sedona, AZ, 2008.

[20] J. He, H. Li, and J. Tan, "Real-time Daily Activity Classification with Wireless Sensor Networks using Hidden Markov Model," in *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 22-26 Aug, 2007.

[21] N. Györbíró, Ákos Fábián, and G. Hományi, "An activity recognition system for mobile phones," in *Journal of Mobile Networks and Applications*, vol. 14, no. 1, February 2009.

[22] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," in *IEEE PerCom*, 2009.

[23] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 1999, pp. 43–52.

[24] J. Li and L. Wong, "Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns," pp. 725–734, 2002.

[25] T. Gu, Z. Wu, L. Wang, X. Tao, and J. Lu, "Mining emerging patterns for recognizing activities of multiple users in pervasive computing," in *Proceedings of the 6th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '09)*, July 2009.

[26] J. Li, G. Liu, and L. Wong, "Mining statistically important equivalence classes and delta-discriminative emerging patterns," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 2007, pp. 430–439.

[27] G. Dong, X. Zhang, L. Wong, and J. Li, "CAEP: Classification by aggregating emerging patterns," *Lecture notes in computer science*, pp. 30–42, 1999.

[28] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. of IJCAI (1995)*, 1995, pp. 1137–1143.

[29] H. Liu, M. Bolic, A. Nayak, and I. Stojmenovic, "Taxonomy and challenges of the integration of rfid and wireless sensor networks," in *IEEE Network*, 2008, pp. 26–35.