This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2856119, IEEE Internet of Things Journal

Orientation Independent Activity/Gesture Recognition Using Wearable Motion Sensors

Jian Wu, *Student Member, IEEE* and Roozbeh Jafari, Senior *Member, IEEE*

***Abstract*—Activity/gesture recognition using wearable motion sensors, also known as inertial measurement units (IMUs), provides an important context for many ubiquitous sensing applications. The design of the activity/gesture recognition algorithms typically requires information about the placement and orientation of the IMUs on the body, and the signal processing is often designed to work with a *known* orientation and placement of the sensors. However, sensors could be worn or held differently. Therefore, signal processing algorithms may not perform as well as expected. In this paper, we present an orientation independent activity/gesture recognition approach by exploring a novel feature set that functions irrespective of how the sensors are oriented. A template refinement technique is proposed to determine the best representative segment of each gesture thus improving the recognition accuracy. Our approach is evaluated in the context of two applications: activity of daily living recognition and hand gesture recognition. Our results show that our approach achieves 98.2% and 95.6% average accuracies for subject dependent testing of activities of daily living and gestures, respectively.**

***Index Terms*—Activity/Gesture recognition, dynamic time warping (DTW), orientation independent, template refinement, wearable motion sensors**

I. Introduction

THE recognition of activities of daily living (ADLs) and the recognition of hand gestures have attracted much attention in recent years due to its importance in the development of various applications such as healthcare and rehabilitation[1, 2]. Vision-based recognition techniques have been widely investigated for this purpose [3, 4, 5, 6, 7]. These techniques typically require cameras mounted in the environment which inherently suffer from a limited range of vision. Further, the required infrastructure may not be available at all the desired locations or may be too expensive to implement. Issues associated with users' privacy also limit the utility of vision-based techniques. As an alternative solution, wearable sensors capable of activity/gesture recognition are gaining popularity due to their minimal cost, ubiquitous nature and ability to provide sensing opportunities at any time and place [8, 9, 10, 11].

Activity/gesture recognition algorithms based on IMUs typically assume that the configuration (*e.g.*, location and orientation) of the sensors is known and does not change throughout the deployment. However, accidental displacement of the sensors may occur due to the user's movements. Moreover, there is no guarantee that the user will place the sensors precisely at the expected orientation and location each time they put them on, as is required by most activity recognition algorithms. As internet of things (IoT) emerges, besides the form factor of wearables, low cost IMUs can be embedded in a lot of daily objects (*e.g.* pen, mouse, smartphone and so on). This enables a ubiquitous gesture interface whenever these handheld devices are available. For example, the user can hold his wireless mouse, which is equipped with IMU, to perform gestures and to interact with smart devices with his gestures recognized when wearables are not available. However, the sensor configuration of IMUs on these devices could be different due to different manufacturer and different form factor. If the algorithm is designed to work with a known sensor orientation, significant retraining and model development will be required for every new orientation and configuration requiring substantial amount of data.

To address this issue, two approaches were proposed previously. The first technique calibrates the displacement of the sensor orientation so that the sensor readings are transformed to the original orientation in which the algorithm is trained [12, 13]. This technique has two weaknesses. First, the calibration requires extra efforts. Data will need to be generated for a variety of configurations which further complicates the calibration. Second, if the calibration cannot be done at the moment the displacement occurs, the algorithm does not work unless the calibration is completed. The second approach builds classification algorithms on top of the orientation independent features [14, 15, 16, 17]. Our work belongs to the second category. Specifically, our approach applies dynamic time warping (DTW) to perform segmentation and a thresholding technique to complete the classification. It requires simple fixed-point arithmetic and less computational resources than the traditional classification algorithms (*e.g.*, support vector machine using large feature sets, Bayesian networks or hidden Markov model (HMM)). Furthermore, our proposed model operates in presence of orientation changes in IMUs. All these characteristics are suitable for low-power wearable computers.

Jian Wu is with the Department of Computer Science and Engineering of Texas A&M University, College Station, TX 77840 USA (e-mail: jian.wu@tamu.edu).

Roozbeh Jafari is with the Department of Computer Science and Engineering, the Department of Biomedical Engineering and Department of Electrical Engineering of Texas A&M University, College Station, TX 77840 USA. (email: rjafai@tamu.edu)

In addition to addressing the sensor rotational displacement, the activity/gesture recognition system could be enhanced by considering the consistent portions of movements, when movement can be performed in various ways or the movement has consistent and inconsistent segments across multiple executions. For many gestures, we observe that parts of the movement may remain consistent and parts may yield in inconsistent signals. If the system considers one gesture as a whole and performs the recognition based on the complete gesture signal, the inconsistent portion will negatively impact the recognition performance. One example is drinking coffee. This gesture typically includes three segments: 1) picking up the cup, 2) a set of arbitrary movements in middle that may or may not be present at every instance of drinking, and 3) bringing the cup to the mouth and tilting it to drink the coffee. We observe that the first and the third segments are always present with predictable/consistent signals captured by sensors while the second segment may not offer a great deal of consistency. Our system and algorithms will identify the consistent segments of the signal and use this information to enhance the gesture recognition accuracy.

The major contributions and innovation in our manuscript include:

- An orientation independent, speed independent activity/gesture recognition system is proposed. The gyroscope readings are used to obtain total angular change series of a certain movement. DTW is used to do the segmentation and thresholding technique is used to do the classification. This technique will significantly reduce the requirements on the amount of data for model training.
- A template refinement technique is applied to determine consistent segments of a movement and the inconsistent segments are eliminated using a variation of DTW called star-padding. The recognition accuracy could be enhanced by this technique.

The remainder of the paper is organized as follows: background and related works are introduced in Section II, followed by preliminaries and challenges in Section III. Our approach is explained in Section IV and the experimental setup and experimental results are discussed in Section V and Section VI, respectively. The discussion and limitations are introduced in Section VII before the paper is concluded in Section VIII.

II. Background and Related Works

It is generally known that sensor displacement affects the accuracy of the activity/gesture recognition algorithms. The impact of sensor translation and rotation on a sample activity recognition algorithm based on DTW has been discussed previously [18]. Researchers explored the impact of rotational and translational displacements on the recognition algorithms and provided recommendations on how to deal with the sensor displacement [19].

Three different approaches are proposed to address the issue when sensor rotational displacement will affect the result of the recognition algorithm. The first approach is to study the statistical distribution of the features, and adjust the features adaptively. The possibility of system self-calibration through the adjustment of the classifier decision boundaries is proposed [20]. Similarly, a method to compensate for the data distribution shift caused by sensor displacements using an expectation-maximization algorithm and covariance shift analysis is discussed [21]. These approaches adjust the feature space for small displacement ranges. However, they cannot calibrate for more substantial displacements and if a major displacement occurs, the recognition algorithm will exhibit poor accuracy.

The second approach is to recalibrate the sensor orientation and to transform the sensor readings to the original space in which the system is trained. An orientation independent approach that calculates the transformation rotation matrix with respect to a reference frame and transforms the accelerometer data vector back to this reference frame is proposed [13]. In this investigation, the researchers assume one of the sensor axes is fixed and the rotation occurs along this axis. This is not always true in reality and moreover, this technique estimates the gravity from a period of the same activity or posture (e.g., walking, cycling and standing). This technique does not work for transitional movements, like sit-to-stand or sit-to-lie. In another investigation, the use of signal average to estimate the gravity component and determine the vertical axis of the device is proposed [14]. The vertical axis is determined and the sensor readings are projected onto this axis [17]. Since the vertical axis alone cannot define a frame, the technique extracts the signal magnitude which is perpendicular to the vertical axis and is along the horizontal axis. The complete frame orientation is calculated from a period of walking [12]. The vertical axis is estimated using the method in [14], while the horizontal axis is calculated as the first component of the PCA, which corresponds to the direction in which the majority of movement occurs. The disadvantage of these approaches is that they require a period of 10 seconds or more of forward-backward movements to estimate the vertical and horizontal axes. The estimation will be incorrect if non-bipedal movements are present and moreover, the recognition algorithm will fail if the calibration is not performed in a timely manner.

The third approach explores the orientation independent features. The relative energy distribution over five different parts of the body is investigated to classify four different activities [15]. This approach performs well to distinguish dynamic activities that have different energy distribution on different body parts. However, for those activities that have similar energy distribution on different parts, (*e.g.*, walking and running), it may not exhibit acceptable performance. In another study, features in frequency domain are used to distinguish the periodic dynamic activities by analyzing the periodicity between different movements [22]. However, the frequency resolution is a concern when identifying the difference between activities that have similar periodicity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2856119, IEEE Internet of Things Journal

Moreover, these features are not suitable to detect short term transitional movements (*e.g.*, sit-to-stand, stand-to-sit).

In our work, we use the total angle change series as a time domain feature; our approach can recognize either dynamic movements or transitional movements as long as they have different angle change series. This discriminant feature is inherently present in all distinct movements. Our feature set is unique in the sense that we create a time series of total angle change in the duration of an activity at different time scales and we use it as a template for the activity.

As for location displacement of the sensors on body, the literature has considered two scenarios. The first scenario concerns displacement across different body parts; for example, the sensor should be worn in the trouser pocket whereas it is worn in the shirt pocket instead. The on-body location of the wearable sensor can be identified by classifying the accelerometer data when the user is walking [23]. In another work, the sensor location is determined by leveraging concurrent vision-based Kinect sensor skeleton information [24]. Our approach does not attempt to address this type of sensor location displacement and the existing proposed techniques could be leveraged. The second scenario of location displacement is the displacement that occurs on the same body link. The acceleration will be different if the sensor is placed in a different location of the same body link, however, the rotation angles will always be the same for the rigid body links. As a result, our approach will be robust to location displacements of this type. This analogy also holds when handheld devices are used for gesture recognition.

For template refinement and matching, several studies have been proposed to select the best representative signals to cover the variations in user gestures/activities [25, 26]. Generally, these systems will achieve better performance if a larger number of templates from training set is constructed. However, due to the computational constraints of wearable computers, only a smaller subset of templates should be considered. A template that has minimum distance to all other instances in the training set can be selected as the representative template [26]. To address the dynamically changing nature of human gestures, the templates are updated when a more recent gesture is detected or an incorrect gesture is observed by the user [25]. All these studies assume that the entire template provides a good representation for the gesture/activity. However, in reality, movements may include consistent and inconsistent segments. The system performance can be further enhanced if the inconsistent segments are identified and discarded during the template matching. To the best of our knowledge, this is the first time the notion of template refinement and discarding the inconsistent segments of the templates is considered for gesture/activity recognition.

III. Preliminaries and Challenges

A. Challenges

Activity/gesture recognition using wearable motion sensors (*e.g.*, accelerometers and gyroscopes) has several challenges in real-world applications. The first challenge is the sensor orientation displacement. When sensors are attached to human body, the sensor orientation impacts sensor readings. If the sensors are attached in different orientations, the directions of all three axes will be different. As a result, for the same movement, the signals along each axis will be different. Therefore, accuracy of the classification algorithms which use the features derived from each axis will be dramatically affected. The second challenge is the activity/gesture speed variation. In real life, people will perform the same movement with different speeds in various scenarios. This variation will cause differences in the signals generated by wearable sensors and the signal processing algorithms should be able to handle this variation. The third challenge is related to the inconsistencies present in the movements. For many gestures, parts of the movement may remain consistent and parts may yield in inconsistent signals across various executions of the movements. If the signal processing considers one gesture as a whole and performs the recognition based on the complete activity/gesture signal, the inconsistent portion will negatively impact the recognition performance. Our proposed template refinement technique determines the consistent portion of the signal and use this information to enhance the recognition accuracy.

TABLE I
SYMBOL DEFINITIONS

Symbol	Definition	Explanation
$\boldsymbol{a_i}$	$[a_{ix}, a_{iy}, a_{iz}]$	3 by 1 vector, accelerometer readings along x-axis, y-axis and z-axis at time *i*
$\boldsymbol{w_i}$	$[w_{ix}, w_{iy}, w_{iz}]$	3 by 1 vector, angular velocity along x-axis, y-axis and z-axis at time *i*
$\boldsymbol{A}$	$\boldsymbol{a_1}, \boldsymbol{a_2}, \cdots, \boldsymbol{a_i}, \cdots, \boldsymbol{a_n}$	Time series of accelerometer readings with *n* samples
$\boldsymbol{W}$	$\boldsymbol{w_1}, \boldsymbol{w_2}, \cdots, \boldsymbol{w_i}, \cdots, \boldsymbol{w_n}$	Time series of angular velocity with *n* samples
$\widetilde{I_j}$	$\{I_{ij}, i = 1,2, \cdots n\}$	A set of instances of *j-th* class constructed during the training phase; I_{ij} represents *i-th* feature vector instance of *j-th class*
A_i	$\|\boldsymbol{a_i}\|$	The magnitude of *i-th* accelerometer reading
$\boldsymbol{M}$	$A_1, A_2, \cdots, A_n$	Accelerometer magnitude series

B. Preliminaries

In this paper, a customized 6-axis wearable IMU is used for the study. The IMU has a 3-axis accelerometer and 3-axis gyroscope. The 3-axis accelerometer measures gravity and dynamic acceleration which is caused by the motion. The 3-axis gyroscope measures 3-axis angular velocity. The symbols used in this paper are defined in Table I. At each time corresponding to sample number *i*, both accelerometer and gyroscope generate a 3 by 1 vector $\boldsymbol{a_i}$ and $\boldsymbol{w_i}$. They are the raw sensor data input for our system. $\boldsymbol{A}$ and $\boldsymbol{W}$ represent time series of accelerometer and gyroscope readings, respectively. $\widetilde{I_j}$ denotes a set of feature vector instances constructed during the training phase. I_{ij} refers to *i-th* feature vector instance of *j-th* class. The construction of I_{ij} is introduced in section IV.A. A_i denotes the amplitude of acceleration $\boldsymbol{a_i}$ at time *i*. $\boldsymbol{M}$ is a time series of acceleration amplitude with *n*

samples.

IV. Proposed Approach

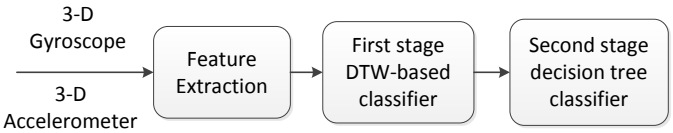

Fig. 1. System diagram.

Fig. 1 shows our proposed recognition technique which addresses all aforementioned practical challenges. A novel orientation independent time series feature set is extracted from the gyroscope data by integrating the 3-axis angular velocity during a short period covering one action, gesture or human movement. Then a first stage DTW-based classifier is used to recognize the gesture/activity. Intuitively, we consider the series of total angle change during this period. Most activities have the same series of angle change irrespective of the speed at which the activities are performed. Thus, the first stage classification is orientation independent and speed independent because the changes are independent of the sensor orientation and movement speed. The inconsistent segments are also analyzed and discarded by applying zero-padding DTW in this stage. However, our first stage classifier cannot distinguish the reversible activities (*e.g.*, sit-to-stand and stand-to-sit) because they have the same angle changes. We further propose a second stage decision tree classifier to distinguish the reversible activities by determining the magnitude of the acceleration which is also orientation independent. The second stage decision tree classifier is only deployed for the reversible activities and is not required for many activities such as hand gestures if they are not reversible.

A. Feature Extraction

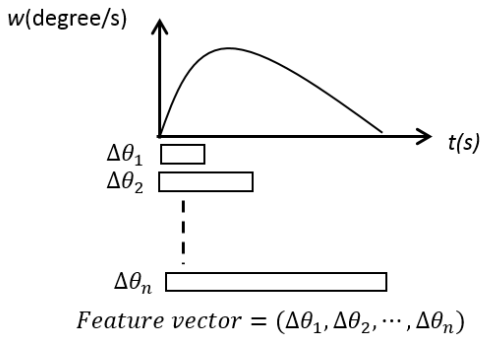

Fig. 2. Feature vector creation.

Human movements or gestures can be classified by a feature set derived from the total angle change observed on each body segment, independent of orientation and speed. The total angle change will be an integration of the total angular velocity over a short period of time (*i.e.*, covering one activity). Given an angular velocity time series $\boldsymbol{W}$ with n samples, the total angle change $\Delta\theta_n$ during the whole period can be obtained. However, only one feature value is not enough to uniquely describe the entire movement. Therefore, we construct a feature vector for a movement. This feature vector is constructed with total angle changes during different time periods. Fig. 2 provides an illustrative example for this feature set . $\Delta\theta_i$ elements in the figure are calculated as in (1).

$$\Delta\theta_i = \sqrt{(\sum_{j=1}^{i} w_{jx} * \Delta t)^2 + (\sum_{j=1}^{i} w_{jy} * \Delta t)^2 + (\sum_{j=1}^{i} w_{jz} * \Delta t)^2} \quad (1)$$

Δt is the time duration between two samples, which is the reciprocal of the sampling frequency. $\Delta\theta_1$ is the total angle change during the first Δt seconds. $\Delta\theta_2$ is the total angle change during the first $2\Delta t$ seconds and $\Delta\theta_n$ represents the total angle change during the first $n\Delta$t seconds. A feature vector of a movement is called an instance of this movement. This instance captures important rotation characteristics and details during the movement. In the feature vector, all *n* angle changes are speed and orientation independent. However, the size of the feature vector varies because the length of a movement is not likely fixed and can be shorter or longer. This challenge of varying length of the movement is addressed by the DTW.

B. First-stage DTW-based Classifier

1) DTW with auto-segmentation

DTW is a template matching algorithm for measuring similarity between two time series with different durations [27, 28]. By investigating the warping distance, the algorithm can perform the segmentation automatically which will be an inherently challenging task for classic classification algorithms. Given two time series $X = x_1, x_2, \cdots, x_i, \cdots, x_n$, and $Y = y_1, y_2, \cdots, y_i, \cdots, y_n$, the cumulative distance is calculated as in (2).

$$D(x_i, y_j) = d(x_i, y_j) + min\begin{cases} D(x_{i-1}, y_j) \\ D(x_i, y_{j-1}) \\ D(x_{i-1}, y_{j-1}) \end{cases} \quad (2)$$

$d(x_i, y_j)$ is the distance between x_i *and* y_j. Any suitable distance function can be used to estimate this, and we use the following distance function in our investigation.

$$d(x_i, y_j) = \|x_i - y_j\| \quad (3)$$

By recording the minimum cumulative distance path, *D*, the warping path between two signals is obtained.

In traditional classification paradigms, we first perform the segmentation so that the feature extraction and classification are performed for a certain segment. One popular segmentation technique is sliding window, in which a segment of signal in a fixed-size window is classified. This window moves through the whole incoming signal and time series in each window is classified as a certain target movement or non-target movement. However, due to the speed variation and the fact that different activities/gestures have different lengths, a fixed-size window cannot cover the exact duration of an activity/gesture. Thus, the accuracy of classification will be negatively impacted. In this paper, DTW is applied to complete the segmentation. A feature vector is constructed in a window with size that is slightly larger than the longest activities/gestures and the segments of potential

activities/gestures within this window will then be determined. Subsequently, the potential segments are supplied to the DTW. In this paper, we construct the feature vector in a window whose size is set to 2.5 seconds, about 1.5 times of the longest movement duration (sit-to-lie) for activity recognition. For gesture recognition, the window size is chosen as 14 seconds, which is 1.5 times of the longest gesture duration (drinking coffee). This guarantees that the time series feature vector covers all target movements even if they are performed at a slower speed. From the cumulative distance table between the template and the incoming time series feature vector, the matching part will be selected, thereby realizing segmentation automatically.

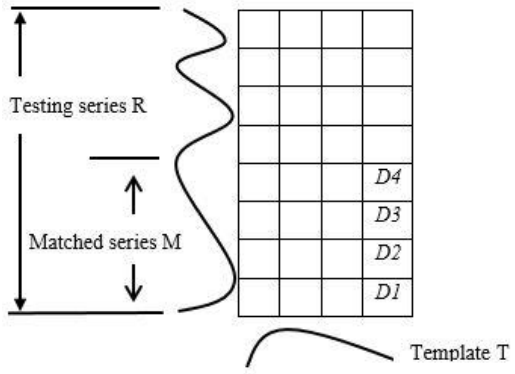

Fig. 3. Auto-segmentation by DTW.

Fig. 3 shows an example of the auto-segmentation function of DTW. Assume that the template *T* has *m* samples and that the incoming feature series *R* (incoming unknown movement) has *n* samples, where *n* > *m*. The table in Fig. 3 provides the cumulative distance table *D*, as described in (2), of the two series [29].

The minimum cumulative distance, without loss of generality as the m^{th} column in the distance table, is considered as the warping distance and the corresponding row index will be the last sample of the matched series *M* within *R*. The warping distance and index are defined in (4).

$$[D_{warp}, index] = \min(D_i), \forall i \in [1, n] \quad (4)$$

D_i is the i^{th} element in the m^{th} column in the distance table. In the example in Fig. 3, D_4 will be selected as the D_{warp}, which is the warping distance between template *T* and series *M*.

2) Template selection

Selecting multiple templates with larger variations covers more cases and may provide a better accuracy for the DTW. However, this increases the computational complexity of the signal processing. Considering the resource constrained nature of the processing units in wearable computers, we only choose one template for each activity. This assumption however does not necessarily need to be enforced and one may choose to consider multiple templates representing each movement. The template for *j*-th movement/gesture is chosen from a collection of *n* instances of the movement/gesture according to the criterion in (5).

$$T = \underset{I_{ij} \subseteq \widetilde{I_j}}{\operatorname{argmin}} \left(\sum_{k=1}^{n} D(I_{ij}, I_{kj}) \right) \quad (5)$$

Where I_{ij} represents *i-th* feature vector instance of class *j*. All these instances are constructed during the training phase with annotated activity/gesture data. *D* represents the DTW distance. The selected template is essentially closest to all the other instances of the movement and serves as the best representative template.

3) Threshold choice based on maximum-margin hyperplane

To distinguish the target movements from the non-target movements, a threshold is chosen for the DTW distance. If the threshold is too tight, certain target movements may not be detected resulting in a higher false negative rate. If the threshold is too loose, non-target movements may be classified as target leading to a large false positive rate. In this paper, we use a 5% maximum-margin hyperplane (MMH) as the threshold [30]. The MMH is the best separation margin between two classes. The 5% MMH is explained in Fig. 4. The *X-axis* represents movement instances (and their indices) and the *Y-axis* shows the DTW distance between target movement template and movement instances. The black dots and white dots represent DTW distances of non-target and target instances, respectively. The *Y-axis* value of the top red line is the mean of the smallest 5% distance samples between target movement template and non-target instances. The bottom black line is the mean of the largest 5% distance samples between the target movement template and target movement instances. The MMH is the average of these two lines represented as the dash-dotted line. We choose the mean of 5% instead of the largest or smallest sample to eliminate the effect of outliers.

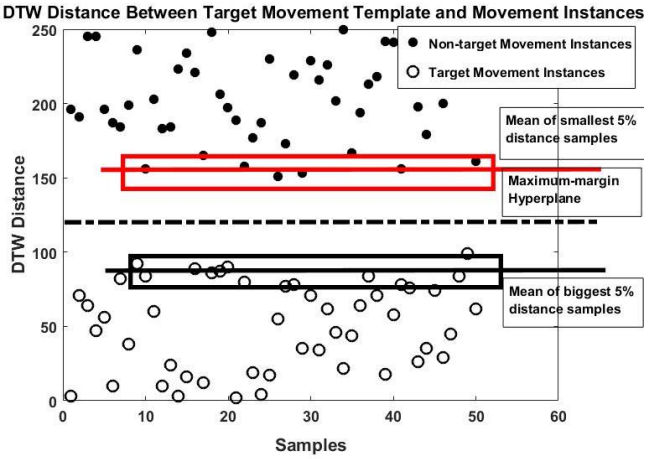

Fig. 4. Maximum-margin hyperplane definition.

C. Inconsistent Segment Analysis and Star-padding DTW

For the activities of daily living considered in this paper, there are typically no inconsistent segments during movements since the durations of activities are typically short. For the gestures, however, like “Drinking coffee”, “Picking up cell phone” or “Eating” certain inconsistent segments are present. The algorithm proposed in this section detects the inconsistent segments. Once we determine the inconsistent segment in the template, the star-padding DTW is used to complete the recognition [31]. The samples of inconsistent segment is replaced by special value (‘*’), which has zero distance with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2856119, IEEE Internet of Things Journal

ALGORITHM 1

Input: a set of movement instance $\widetilde{I_j}$;
Output: a set of consistent segment $\boldsymbol{F}$;

1: Pick an random instance from $\widetilde{I_{ij}}$ as I_{bj};
2: **for** *i* from 1 to n
3: do DTW(I_{bj}, I_{ij});
4: **end**
5: **for** all samples in I_{bj}
6: Calculate sample distance d_i according to (6);
7: **end**
8: perform hierarchal clustering for all d_i according to distance function (7);
9: the largest clusters with small intra-class distances correspond to consistent segments and are used to construct $\boldsymbol{F}$ vector;

all other samples in distance matrix. This will eliminate the effect of inconsistent segments when calculating the warping distance.

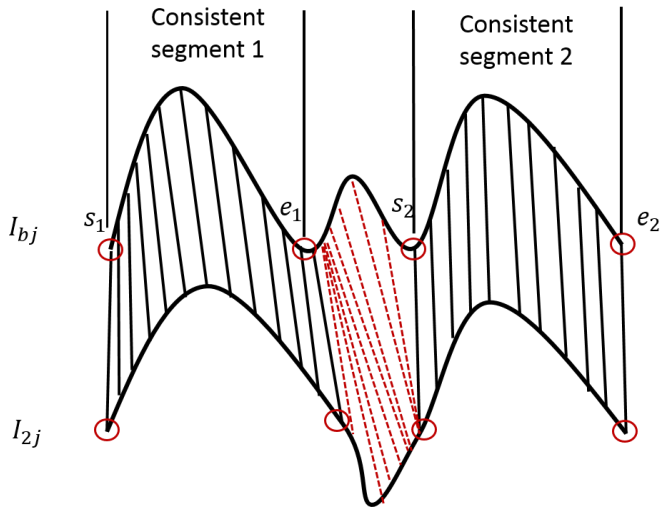

Fig. 5. Inconsistent analysis example.

The first step is to identify consistent and inconsistent segments. Fig. 5 shows an example of the inconsistent movement analysis. In this figure, there are two instances I_{bj} and I_{2j} from *j-th* class. Two consistent segments and one inconsistent segment exist among these two instances. The consistent segment 1 is determined by the starting point s_1 and the ending point e_1 and the consistent segment 2 is determined by the starting point s_2 and the ending point e_2. The solid black lines that connect samples of the two instances show the warped samples of the consistent segments after DTW is applied. The dashed red lines show the warped samples of the inconsistent segment after DTW is applied. In the consistent segments, the warped samples from the two instances are close to each other and thus, the distances between all warped samples should be small. On the other hand, the warped samples of the inconsistent segment could be much different and thus, the distances between all warped samples could be larger. Our goal is to determine all the consistent segments guided by the DTW sample-wise distance.

To formally define the objective, for a given set of movement instances $\widetilde{I_j}$, we want to determine a set of consistent segment sample indices $\boldsymbol{F}=\{(s_1,e_1),(s_2,e_2),\cdots,(s_i,e_i),\cdots((s_n,e_n))\}$ for a beacon instance I_{bj}. s_i and e_i represents starting sample index and ending sample index of *i-th* consistent segment of the instance. In our approach, a random instance is picked as the beacon instance I_{bj}. I_{bj} is the beacon instance for *j-th* class. After applying DTW between beacon instance and any other instance, the sample distance along the warping path of the consistent segment should be small and consistent while the sample distance along the path of the inconsistent segment could be large and inconsistent. Let d_{ik} denote sample distance between *i-th* sample of beacon instance and the warped sample from *k-th* instance. It is possible that *i-th* sample from beacon instance is warped to several samples of *k-th* instance. In this case, the first warped sample distance is chosen as d_{ik}. Now that we obtain the sample distances between the beacon instance and all the other instances, a unique distance for *i-th* sample of the beacon instance is calculated as in (6).

$$d_i = \frac{\sum_{k=1}^{n} d_{ik}}{n} \tag{6}$$

In order to determine $\boldsymbol{F}$, an unsupervised-learning technique hierarchical clustering is applied [32]. Since all d_i's from consistent segments should be small and consistent, they will be clustered to the same cluster while all d_i's from inconsistent segments will not be grouped immediately with consistent d_i's. In the meantime, for grouping and clustering, a spatial constraint must be enforced to avoid clustering samples that have small distance but come from different segments of the movement. For example, the *4-th* sample should be clustered with *5-th* sample before it will be clustered with *216-th* sample if all of them are from the same consistent segment. Therefore, we refine the distance function for clustering to accommodate this objective:

$$f(d_i,d_j) = (d_i - d_j)^2 + \alpha * |i-j|^2 \tag{7}$$

In (7), the first term indicates that all distances from consistent segment tend to be small and the second term adds a spatial constraint. α is a weight parameter that controls the spatial constraint for various applications. After the clustering is completed, the largest size clusters that exhibit small intra-cluster distances are used to determine the s_i and e_i for each consistent segment to construct the $\boldsymbol{F}$ vector. The samples in each selected cluster should come from the same segment and the average linkage distance for this cluster should not be large. This algorithm is summarized in Algorithm 1.

D. Second Stage Decision Tree Classifier

With the first stage DTW distance threshold based classifier, we can classify different activities except for the reversible instances. For example, sit-to-stand and stand-to-sit have similar angular change patterns, and they will exhibit small DTW distance to each other considering our proposed feature set. In order to distinguish between the reversible movements, the amplitude of the accelerometer reading is considered.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2856119, IEEE Internet of Things Journal

The accelerometer data is often noisy and therefore, before calculating the amplitude, a 5Hz low pass filter is applied to the raw acceleration samples.

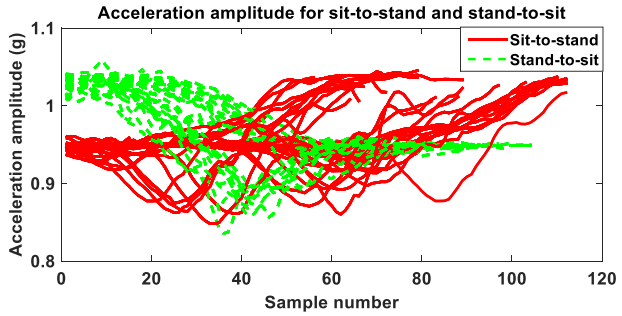

Fig. 6. Acceleration amplitude for sit-to-stand and stand-to-sit.

Fig. 6 shows the amplitude of the acceleration for sit-to-stand and stand-to-sit movements of a thigh sensor from the auto-segmentation. For the sit-to-stand, the leg has an upwards acceleration first which is against the gravity and thus leading to an acceleration whose amplitude is smaller than 1*g*. For the end of sit-to-stand, there will be an acceleration downwards which is in the same direction as gravity. This results in an acceleration whose amplitude is larger than 1*g*. To distinguish the sit-to-stand from stand-to-sit, we look for the order of the maximum peak and minimum valley. For sit-to-stand, the maximum peak happens after the minimum valley while for stand-to-sit, the maximum peak occurs first followed by the minimum valley.

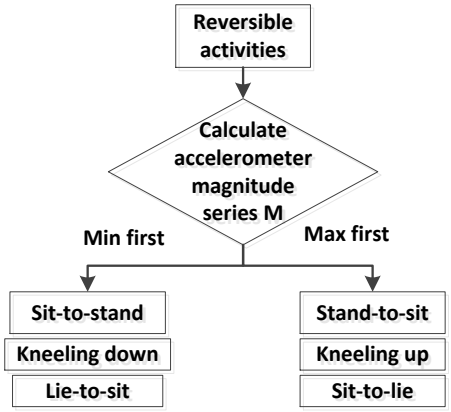

Fig. 7. Second stage decision tree classifier.

Similarly, we can distinguish between the kneeling down and kneeling up and between the sit-to-lie and lie-to-sit by looking at the thigh sensor and ankle sensor, respectively. The second stage classifier based on decision tree is shown in Fig. 7. The reason why decision tree is applied is that it is easy to interpret. For example, it is easy to understand when an accelerometer magnitude peak happens first, it is stand-to-sit; while when an accelerometer magnitude valley happens first, it is sit-to-stand. The three sets of reversible activities are classified by observing the order of occurrence for the maximum peak and the minimum valley samples. Note that for the gesture recognition task, there are no reversible movements and thus the second stage classification is not necessary.

V. Experimental Setup

We evaluate our approach using two different example applications: activity recognition of daily living and hand gesture recognition. For the activity recognition of daily living experiment, two sensors were worn by 10 participants. One was attached to the thigh and the other one was attached to the ankle. While for the hand gesture recognition application, 6 participants were enrolled with one sensor attached to the user's wrist. For both experiments, the users were asked to place the sensors in 6 different orientations (spanning 360 degrees), and perform 20 repetitions of each activity/gesture at each orientation. Each activity/gesture was performed at a slow speed for the first two orientations, at a normal speed for the middle two orientations and at a faster speed for the last two orientations. While the proposed approach is speed independent for most activities and gestures, the activities of walking and running are special cases. Due to biomechanical reasons, different speeds of walking and running have different angle rotations, so our approach would consider them as different movements. The participants performed these two movements at the normal speed for 6 orientations. The sampling frequency was set to 200Hz. The activities of daily living are listed in Table II and the hand gestures are listed in Table III. During the data collection, a camera recorded the movements along with the sensors. Both the sensor data and video data were synchronized. Then a visualization tool is used to segment and annotate the collected data [33]. The tool displays the sensor data and video data at the same time, hence segmentation can be performed and the data is annotated based on the video information. This segmentation serves as the gold standard.

In order to estimate the gravity vector and the horizontal direction vector to transform the accelerometer data to a global frame, a certain duration, such as a 10 second time window is required in certain existing approaches [12, 13, 17]. Their methods are good for the continuous movements or postures, such as sitting, lying, walking, and running, and do not perform well for the transitional movements since gravity has a major effect on some transitional movements. Thus, we

TABLE II
Activities

Activity #	Activity	Activity #	Activity
1	Sit-to-stand	6	Kneeling up
2	Stand-to-sit	7	90degree-turn
3	Sit-to-lie	8	Walking
4	Lie-to-sit	9	Running
5	Kneeling down		

TABLE III
Activities

Gesture #	Gesture	Gesture #	Gesture
1	Wave goodbye	5	Eating with spoon
2	Hand shake	6	Eating with hand
3	Picking up cell phone	7	Eating with chopstick
4	Drinking coffee		

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2856119, IEEE Internet of Things Journal

compare our approach for the task of activity recognition with a pocket cell phone based orientation independent approach proposed by Sun *et al* [16]. The authors of that investigation extract 22 features from 3-axis accelerometer data and the magnitude of the accelerometer data, which include mean, variance, correlation, FFT energy and frequency domain energy. Since our approach uses two sensors, we extract 22 features from each sensor to construct the feature vector. Sun's approach is unable to do the segmentation and it uses a windowing technique that does not necessarily capture the beginning and the end of movements at the beginning and the end of the windows. By comparison, we use the camera labeled annotations to serve as the beginning and the end of each window for their approach. This improves the performance of Sun approach. The LibSVM in Weka is used as the classification algorithm [34].

We design four experiments to evaluate our approach. In the first experiment, we analyze the performance of auto-segmentation of our approach and determine if the beginning and the end of movements are detected correctly. The second experiment evaluates the subject dependent classification performance, in which 10-fold cross validation is used to evaluate the dataset for each person separately [35]. In the third experiment, the subject-independent classification performance is evaluated with the leave-one-subject-out testing method. There are 4 classification performance metrics in our paper, which are accuracy, precision, recall and F-score [36]. In the fourth experiment, the performance of our inconsistent movement analysis algorithm and template refinement is studied.

VI. Experimental Results

A. Auto-segmentation Results

To evaluate the performance of the auto-segmentation from DTW, the video annotated segmentations serve as the gold standard and three error parameters are defined. Let t_1, t_2 be the starting and ending point of the auto-segmented section of one movement and t_1', t_2' be the starting and ending point of the corresponding gold standard section. The errors at the beginning Δt_1, at the end Δt_2 and the total Δt are defined as:

$$\Delta t_1 = |t_1 - t_1'| \quad (8)$$

$$\Delta t_2 = |t_2 - t_2'| \quad (9)$$

$$\Delta t = \Delta t_1 + \Delta t_2 \quad (10)$$

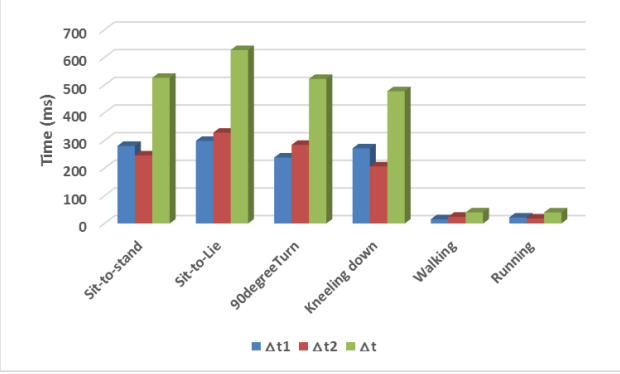

Fig. 8. Errors of auto-segmentation for different activities.

Fig. 8 shows the average error rate of the auto-segmentation for different activities for all the subjects. From the figure, we can see that the errors for walking and running are very small. The reason is that the steps are continuous and the minimum DTW distance leads to good segmentation for each step. For the other activities, we have a static period at the beginning and the end of the activities which results in a larger error for the segmentation. This static period is caused by the inaccurate annotation when the author puts the start label earlier than the actual starting of the movement and puts the end label a little later than the actual ending of the movement to ensure the complete movement is covered. However, the static periods do not affect our algorithm because the integration of the gyroscope signals approaches zero. Fig. 9 shows the auto-segmentation errors for the hand gestures. The errors for hand gestures are a little larger than those of the activities of daily living. The reason is that they also have static periods at the beginning and at the end, and the hand gestures usually take longer than activities of daily living discussed in this paper.

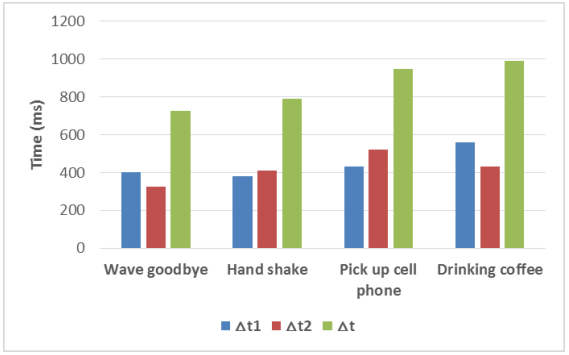

Fig. 9. Errors of auto-segmentation for hand gestures.

B. Subject Dependent Classification Results

In this experiment, 10-fold cross validation is used for each subject separately and the classification accuracy is analyzed.

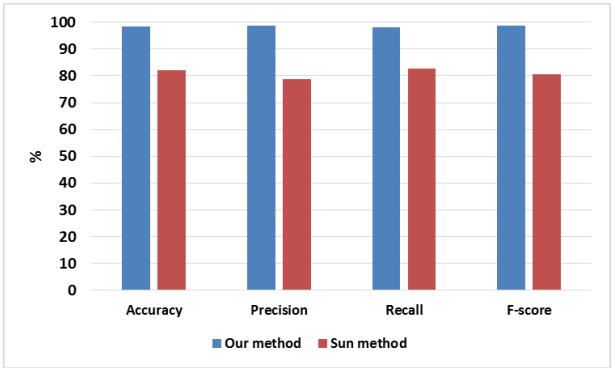

Fig. 10. Subject dependent classification results for activity recognition task.

Fig. 10 shows the classification performance of our approach compared to Sun approach. All the results are calculated as the average for all 9 activities. From the figure, we can see our approach achieves a very good performance in accuracy, precision, recall and F-score for the subject dependent testing compared to Sun's method. Our approach offers 98.52%, 98.62%, 98.21% and 98.65% for accuracy,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2856119, IEEE Internet of Things Journal

precision, recall and F-score, and the improvements of these metrics compared to Sun method are 16.41%, 18.67%, 15.48% and 18.01% respectively.

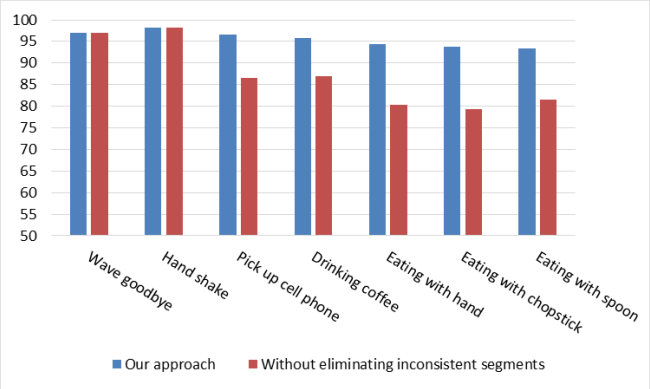

Fig. 11. Subject dependent classification results for hand gesture recognition task.

Fig. 11 shows the subject dependent classification accuracy for hand gesture recognition task when we do and we do not eliminate the inconsistent segments. From the figure, we can see for both 'Wave goodbye' and 'Hand shake', our approach achieves good performance and it is similar to the case when we don't consider the inconsistent segments and do not eliminate the effect of those segments. For 'Picking up cell phone', 'Drinking coffee' and three different styles of 'eating', our approach achieves better performance than the same approach without eliminating the inconsistent segments. Specifically, our approach shows ~10% accuracy improvement. This is because 'Picking up cell phone', 'Drinking coffee' are similar to each other and they both have inconsistent segments in middle. Similarly, the three types of 'eating' are similar to each other and they all have inconsistent segments in middle. If the inconsistent segments are considered to be part of the whole gesture, the effect will cause confusion between these gestures. Since Sun approach is proposed for activity recognition using cell phone, we do not compare to their approach for hand gesture recognition.

C. Subject Independent Classification Results

Different human subjects may perform the same activity in a slightly different way. The subject independent classification analysis tests how robust the recognition system is with respect to different subjects. In the subject independent test, the leave-one-subject-out testing method is applied.

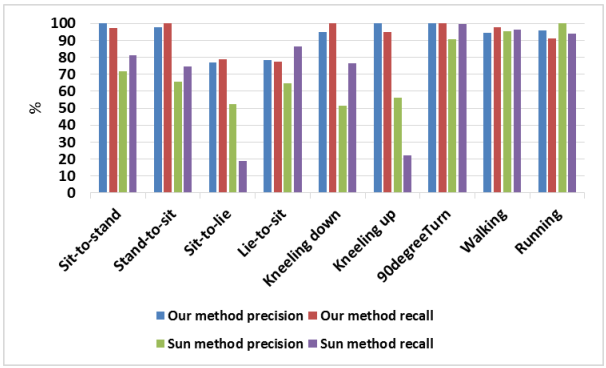

Fig. 12. Precision and recall for different activities for the subject independent test.

Fig. 12 shows the precision and recall for different activities in the subject independent test. From the figure, we observe that our approach outperforms the Sun approach for most of the activities. Only for 90degreeTurn, walking and running, Sun approach achieves similar performance to ours. One possible reason is that the frequency domain features used by Sun approach are good for discriminating the periodic movements (e.g. walking and running). We noted that our classification precisions and recalls for sit-to-lie and lie-to-sit are much lower than for the other activities. This is because one of the human subjects performed the sit-to-lie and lie-to-sit activities in a very different manner compared to the other subjects.

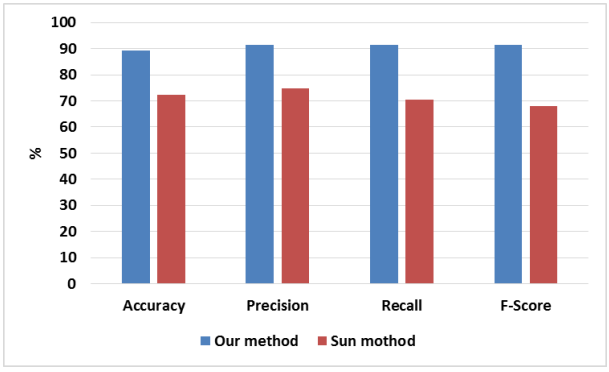

Fig. 13. Subject independent classification results for activity recognition task.

Fig. 13 shows the subject independent classification results for the activity recognition task. The figure shows our method achieves much better performance than Sun's method. It indicates that our method is more robust to subject variation. The reason may be that the activities chosen are well distinguishable from each other. The small variation caused by the subjects does not affect our algorithm too much. The improvements of our method with respect to accuracy, precision, recall and F-score are 17.14%, 16.27%, 19.86% and 23.42% respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2856119, IEEE Internet of Things Journal

TABLE IV
INCONSISTENT MOVEMENT DETECTION ERRORS

	Δt_1 (seconds)	Δt_2 (seconds)	Δt (seconds)
Picking up cell phone	0.278	0.201	0.479
Drinking coffee	0.246	0.235	0.481
Eating with hand	0.325	0.401	0.726
Eating with chopstick	0.337	0.385	0.722
Eating with spoon	0.267	0.214	0.481

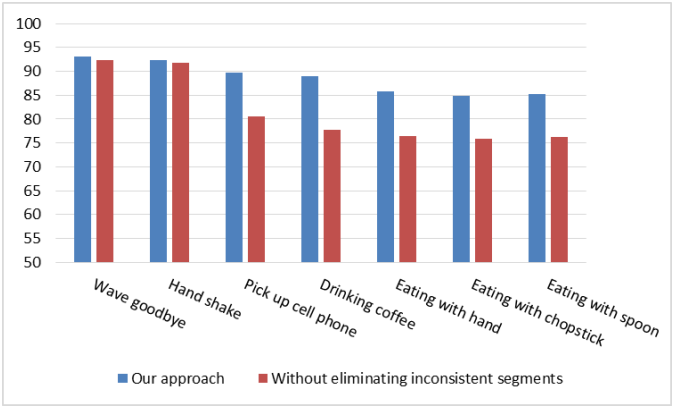

Fig. 14. Subject independent accuracy for gesture recognition task.

Fig. 14 shows the subject independent accuracy for the gesture recognition task. As shown in the figure, our approach achieves good performance for all seven hand gestures and our approach performs much better when the inconsistent parts are eliminated for both 'Picking up cell phone', 'Drinking coffee' and three different types of 'eating'.

D. *Inconsistent Movement Analysis*

Five gestures in our investigation involve inconsistent segments: 'Picking up cell phone', 'Drinking coffee' and three different types of 'eating'. All of them typically include three segments. For 'Drinking coffee', the three segments can be described as follows: 1) picking up the cup, 2) a set of arbitrary movements in middle that may or may not be present at every instance of drinking, and 3) bringing the cup to the mouth and tilting it to drink the coffee. For 'Picking up cell phone', three segments are typically observed: 1) taking cell phone out of pocket and lifting the cell phone to a certain level to read the caller information. 2) a set of different movements in middle that may or may not present at every instance (*e.g.* shift or rotate the cell phone to unlock) and 3) bringing the cell phone close to the ear to answer the phone call. For different styles of 'Eating', three segments are observed: 1) picking up food, 2) a set of arbitrary movements in middle that may or may not be present at every instance, and 3) bring the food to the mouth and feed it to the mouth. For all gestures, we observe that the first and third segments are consistent between all the instances while the second segment may vary a lot from instance to instance. To evaluate our inconsistent segment analysis technique, we use the same error metrics defined by (8) – (10). Here t_1, t_2 are the starting and ending points of the inconsistent segment of one movement and t_1', t_2' are the starting and ending point of the corresponding gold standard segment obtained from video. The average error for these five gestures are shown in Table IV. In Section VI.B and Section VI.C, the inconsistent movement analysis improves the classification performance significantly and these errors are good enough for our application.

VII. DISCUSSION AND CONCLUSION

To the best of our knowledge, the feature set and the signal processing algorithms described in this work have been proposed for the first time. Our proposed techniques address several important challenges: sensor orientation variations, movement speed variations, and the inconsistent segments present in some movements. In our approach, once an activity/gesture is detected, the time duration of the activity is calculated. The movement's speed is an interesting context for pervasive computing applications as we can infer if the person is in a hurry or is tired. Besides, our approach works for both dynamic periodic movements (*e.g.*, walking and running) and transitional movements (*e.g.*, sit-to-stand and sit-to-lie) while most orientation independent frequency-based activity recognition algorithms previously published operate solely on dynamic periodic movements.

One limitation of our approach is that we used two sensors to recognize the activities of daily living listed in this paper. If we only use one thigh sensor, our approach cannot separate sit-to-lie and sit-to-stand well. This is naturally due to the fact that the two movements have the same footprint on the thigh. As a potential alternative to using multiple sensors, finer-grained orientation independent features from accelerometers could be considered to help distinguish these two movements. In our future work, we will enhance the recognition accuracy of our algorithm to cover a larger number of movements using only one sensor. The other limitation of our approach is that different subjects have slightly different activity templates which decreases the cross subject classification performance. Selecting a larger number of templates will help enhance the cross subject classification accuracy while increasing the computational cost.

In this paper, we proposed an activity/gesture recognition approach using wearable motion sensors that can address several practical challenges and detect useful context information. An orientation independent, speed independent feature set is proposed and a two-stage signal processing algorithm is suggested to perform the activity/gesture recognition. A template refinement technique is proposed to eliminate the negative impact of the inconsistent segments of a movement. Two example applications (*i.e.*, activity recognition and gesture recognition) are utilized for the evaluation. The experimental results show good classification accuracy while retaining robustness to several practical challenges associated with wearable motion sensors in real-world scenarios.

REFERENCES

[1] S. Mitchell, J. Collin, C. De Luca, A. Burrows, and L. Lipsitz, "Open-loop and closed-loop postural control mechanisms in parkinson's disease: increased mediolateral activity during quiet standing," *Neuroscience letters*, vol. 197, no. 2, pp. 133–136, 1995.

[2] J. Pansiot, D. Stoyanov, D. McIlwraith, B. P. Lo, and G.-Z. Yang, "Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems," in *4th international workshop on wearable and implantable body sensor networks (BSN 2007)*, pp. 208–212, Springer, 2007.

[3] R. Bodor, B. Jackson, and N. Papanikolopoulos, "Vision-based human tracking and activity recognition," in *Proc. of the 11th Mediterranean Conf. on Control and Automation*, vol. 1, Citeseer, 2003.

[4] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2847–2854, IEEE, 2012.

[5] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[6] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1290–1297, IEEE, 2012.

[7] C. Zhang and Y. Tian, "Rgb-d camera-based daily living activity recognition," *Journal of Computer Vision and Image Processing*, vol. 2, no. 4, p. 12, 2012.

[8] T. Brezmes, J.-L. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," in *Distributed computing, artificial intelligence, bioinformatics, soft computing, and ambient assisted living*, pp. 796–799, Springer, 2009.

[9] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, *et al.*, "The mobile sensing platform: An embedded activity recognition system," *Pervasive Computing, IEEE*, vol. 7, no. 2, pp. 32–41, 2008.

[10] A. M. Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim, "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 5, pp. 1166–1172, 2010.

[11] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern recognition letters*, vol. 29, no. 16, pp. 2213–2220, 2008.

[12] A. Henpraserttae, S. Thiemjarus, and S. Marukatat, "Accurate activity recognition using a mobile phone regardless of device orientation and location," in *Body Sensor Networks (BSN), 2011 International Conference on*, pp. 41–46, IEEE, 2011.

[13] S. Thiemjarus, "A device-orientation independent method for activity recognition," in *Body Sensor Networks (BSN), 2010 International Conference on*, pp. 19–23, IEEE, 2010.

[14] D. Mizell, "Using gravity to estimate accelerometer orientation," in *2012 16th International Symposium on Wearable Computers*, pp. 252–252, IEEE Computer Society, 2003.

[15] N. Pham and T. Abdelzaher, "Robust dynamic human activity recognition based on relative energy allocation," in *Distributed Computing in Sensor Systems*, pp. 525–530, Springer, 2008.

[16] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *Ubiquitous intelligence and computing*, pp. 548–562, Springer, 2010.

[17] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pp. 1–10, ACM, 2009.

[18] N. Kale, J. Lee, R. Lotfian, and R. Jafari, "Impact of sensor misplacement on dynamic time warping based human activity recognition using wearable computers," in *Proceedings of the conference on Wireless Health*, p. 7, ACM, 2012.

[19] K. Kunze and P. Lukowicz, "Dealing with sensor displacement in motion-based onbody activity recognition systems," in *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 20–29, ACM, 2008.

[20] K. Forster, D. Roggen, and G. Troster, "Unsupervised classifier self-calibration through repeated context occurences: is there robustness against sensor displacement to gain?," in *Wearable Computers, 2009. ISWC'09. International Symposium on*, pp. 77–84, IEEE, 2009.

[21] R. Chavarriaga, H. Bayati, and J. D. Millán, "Unsupervised adaptation for acceleration-based activity recognition: robustness to sensor displacement and rotation," *Personal and Ubiquitous Computing*, vol. 17, no. 3, pp. 479–490, 2013.

[22] M.-M. Bidmeshki and R. Jafari, "Low power programmable architecture for periodic activity monitoring," in *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, pp. 81–88, ACM, 2013.

[23] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster, "Where am i: Recognizing on-body positions of wearable sensors," in *Location-and Context-Awareness*, pp. 264–275, Springer, 2005.

[24] G. Bahle, P. Lukowicz, K. Kunze, and K. Kise, "I see you: How to improve wearable activity recognition by leveraging information from environmental cameras," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pp. 409–412, IEEE, 2013.

[25] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[26] J.-K. Min, B. Choe, and S.-B. Cho, "A selective template matching algorithm for short and intuitive gesture ui of accelerometer-builtin mobile phones," in *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*, pp. 660–665, IEEE, 2010.

[27] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series.," in *KDD workshop*, vol. 10, pp. 359–370, Seattle, WA, 1994.

[28] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Readings in speech recognition*, vol. 159, 1990.

[29] P. Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, pp. 1–23, 2008.

[30] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, 1992.

[31] Y. Sakurai, C. Faloutsos, and M. Yamamuro, "Stream monitoring under the time warping distance," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pp. 1046–1055, IEEE, 2007.

[32] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*, pp. 321–352, Springer, 2005.

[33] T. R. Bennett, C. Savaglio, D. Lu, H. Massey, X. Wang, J. Wu, and R. Jafari, "Motionsynthesis toolset (most): a toolset for human motion data synthesis and validation," in *Proceedings of the 4th ACM MobiHoc workshop on Pervasive wireless healthcare*, pp. 25–30, ACM, 2014.

[34] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[35] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, pp. 1137–1145, 1995.

[36] D. Powers, "Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation (tech. rep.)," *Adelaide, Australia*, 2007.