2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Wearable Accelerometer Based Extendable Activity Recognition System

Jie Yang, Member IEEE, Shuangquan Wang, Ningjiang Chen, Xin Chen, Pengfei Shi, Senior Member IEEE

Abstract—**Recognizing the human activities of daily living (ADL) is an important research issue in the pervasive environment. Activity recognition is treated as a classification problem and the multi-class classifier is often used. Though the multi-class classifier can obtain high classification accuracy, it can not detect the noise activities and unknown activities, and the system has no extendable recognition capability. In this paper, we proposed a recognition system which can recognize known activities and detect unknown activities simultaneously. For each known activity, one one-class classification model is built up and the combined one-class classification models are used to judge whether a test sample belongs to known activities. For the known samples, the multi-class classifier is used to recognize their types. For the continuous unknown samples, based on segmentation algorithm, training samples of new activities are extracted and added into the recognition system to extend the system's recognition capability.**

I. Introduction

Technology advances in wearable computing enable the development of various applications which intend to provide appropriate and friendly services based on the recognition of human's activities. In [1], Paul Lukowicz uses body worn sensors to automatically track the progress of maintenance or assembly tasks in a wood shop. [2] measures triaxial accelerations in freestyle swimming on Japanese top-level college swimmers to analyze and evaluate swimmers' stroke technique. In [3] and [4], several human activities of daily living (ADL), including standing, walking, climbing up/down stairs and brushing teeth, are analyzed.

Activity recognition is treated as a classification problem and the multi-class classifier is often used to classify the testing samples. Before recognition, the training samples for each activity are collected and one multi-class classification model is trained based on these training samples. During implementation, each testing sample is fed into the multi-class classification model and its output is the recognition result, i.e., the most likely activity the human should be. Though the multi-class classifier can find the optimal classification boundaries between classes and obtain high classification accuracy, it can not detect the noise activities and unknown activities. In addition, the recognition system cannot be extended. It can only recognize the known activities defined at the beginning.

In this paper, we propose an extendable activity recognition system which can 1) detect the unknown and noise activities; 2) recognize the types of the known activities and 3) extract the training samples of the new activities to extend the system's recognition capability. In this system, for each known activity, one one-class classification model is trained to judge whether a testing sample belongs to this activity or not. If a testing sample does not belong to any known activities, it will be treated as noise activity or unknown activity and preserved for further processing. Otherwise, it will be fed into a multi-class classification model whose output is the recognized activity type. For the continuous unknown samples, based on segmentation algorithm, training samples of new activities are extracted and added into the recognition system to extend the system's recognition capability.

II. Wearable Accelerometer Based Extendable Activity Recognition System

A. The Flowchart of the Proposed System

Fig.1 shows the flowchart of the extendable activity recognition system. The implementation process has three phases: training phase, testing phase and updating phase, as the dashed arrows, solid triangle arrows and dashed triangle arrows show, respectively. The training phase includes the following operation steps:

Step 1: The representative sensor data of activity i $(i = 1,2,\cdots,n)$ is collected and is segmented with the same window length, L;

Step 2: For each segment, the same features are extracted and form a feature vector. Suppose activity i has s_i feature vectors, there are totally s $(s = \sum_{i=1}^{n} s_i)$ feature vectors. These feature vectors are preserved;

Step 3: All the features are normalized. The normalization parameters and the normalized feature vectors are also preserved for online data processing;

Step 4: Principal component analysis (PCA) algorithm is then applied to optimize the normalized feature vectors and to reduce its dimension by selecting the most significant feature components. The result feature vectors are the training samples. Training samples of each activity form the

Jie Yang, Shuangquan Wang, Pengfei Shi are with the Institute of Image Processing & Pattern Recognition, Shanghai Jiao Tong University,Shanghai, China, 200240. Ningjiang Chen, Xin Chen are with the Philips Research East Asia, Shanghai, China, 200233. This research is partly supported by NSFC(No. 60775009).

training set of its one-class classifier. Training samples of all predefined activities form the training set of the multi-class classifier. Transformation model of the PCA algorithm is also preserved;

Step 5: For each activity, based on the training set obtained above, a one-class classification model is trained;

Step 6: A multi-class classification model is trained.

The training phase is executed before the testing phase and updating phase, and often offline. The online testing phase includes following four steps:

Step 1: During recognizing, the real-time sensor data with window length of L is collected and the features are extracted from it;

Step 2: Based on the normalization parameters and the transformation model of the PCA algorithm obtained in the training phase, we normalize the testing feature vector and reduce its dimension. The result vector is the testing sample;

Step 3: The testing sample is fed into the n one-class classification models and the n detection results are fused using the fusion algorithm. In our experiments, the logical OR operation is used as the fusion rule. That is, if a testing sample belongs to any known activity, it is a known sample. Otherwise, it is an unknown sample;

Step 4: If the testing sample is known, it will be fed into the multi-class classification model to recognize which activity type it should be; if it is unknown, the corresponding feature vector before normalization and dimension reduction will be preserved for further processing (updating phase).

The updating phase aims to extract training samples of new activities and extend the system's recognition capability. It has the following steps:

Step 1: Eliminating the noise and discrete activity data from the unknown testing samples, and only preserving the continuous segments whose duration is larger than t (we assume that the duration of one new activity should be no less than t);

Step 2: Analyzing the pattern of each segment. If one segment contains two or more new activities, segmenting it into subsegments so that each subsegment contains only one activity;

Step 3: Extracting the raw feature vectors for each new activity. After normalization and dimension reduction, the training samples of all known and new activities are obtained;

Step 4: Retraining the one-class classification model for each activity and the multi-class classification model. The operations in Step 3 and Step 4 are similar to that in the training phase;

Step 5: Replacing the classification models and updating the recognition system.

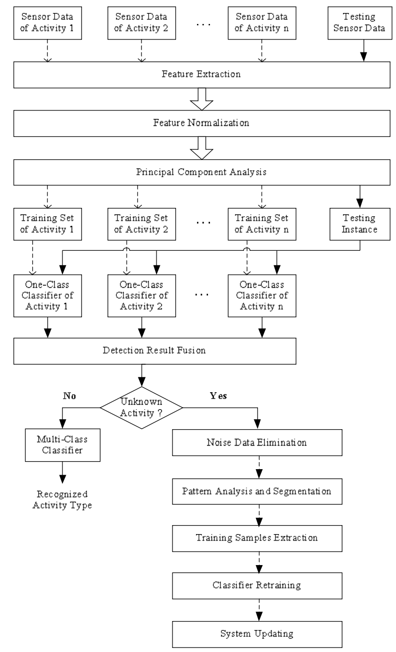

Figure 1. Flowchart of the extendable activity recognition system

B. *Feature Extraction and Normalization*

Features are extracted from the raw triaxial accelerometer data using a window size of 64 with 32 samples overlapping between consecutive windows. This window size enables easy computation of Fast Fourier Transform (FFT) for calculating some features. Feature extraction on a sliding window with 50% overlap has demonstrated to be successful in previous work [5]. At a sampling frequency of 32Hz, each window represents 2 seconds.

Eleven features are extracted from each of the three axes of the accelerometer. The features are: mean, standard deviation, energy, four amplitude statistics features and four shape statistics features of the power spectral density (PSD). In addition, correlation between each pair of axes is also included, giving a total of 36 features.

The DC feature is the mean acceleration value of the signal over the window [6]. Standard deviation is used to characterize the stability of the signal. The energy feature is calculated as the sum of the squared discrete FFT component magnitudes of the signal.

PSD is defined as the Fourier transform of the autocorrelation of the time series signal and describes the energy distribution of a signal in the frequency domain. We derive features by calculating the amplitude statistics and the shape statistics of the PSD. The amplitude statistics is defined as:

Amplitude: $$\mu_{amp} = \frac{1}{N}\sum_{i=1}^{N} C(i) \tag{1}$$

Standard deviation: $$\sigma_{amp} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(C(i) - \mu_{amp}\right)^2} \tag{2}$$

Skewness: $$\gamma_{amp} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{C(i)-\mu_{amp}}{\sigma_{amp}}\right)^3 \quad (3)$$

Kurtosis: $$\beta_{amp} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{C(i)-\mu_{amp}}{\sigma_{amp}}\right)^4 - 3 \quad (4)$$

where $C(i)$ is the PSD magnitude for the i th frequency bin, and N is the number of the frequency bins. Similarly, the shape statistics is defined as:

Mean: $$\mu_{shape} = \frac{1}{S}\sum_{i=1}^{N} iC(i) \quad (5)$$

Standard deviation: $$\sigma_{shape} = \sqrt{\frac{1}{S}\sum_{i=1}^{N}(i-\mu_{shape})^2 C(i)} \quad (6)$$

Skewness: $$\gamma_{shape} = \frac{1}{S}\sum_{i=1}^{N}\left(\frac{i-\mu_{shape}}{\sigma_{shape}}\right)^3 C(i) \quad (7)$$

Kurtosis: $$\beta_{shape} = \frac{1}{S}\sum_{i=1}^{N}\left(\frac{i-\mu_{shape}}{\sigma_{shape}}\right)^4 C(i) - 3 \quad (8)$$

where $S = \sum_{i=1}^{N} C(i)$.

Correlation between each pair of axes is calculated as the ratio of the covariance and the product of the standard deviations, i.e. $corr(x,y) = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y}$.

To eliminate the scaling effects among different features, all the features are normalized using the z-score normalization algorithm [7].

C. One-Class Classification

One-class classification is a special type of classification problem which classifies the testing samples into the target class (known class) or the outlier class (unknown class). In one-class classification, it is assumed that only information of the target class is available. The boundary between the target class and the outlier class has to be estimated from data of the target class. The task of one-class classification is to define a classification model, i.e. a boundary around the target class, such that the classification model accepts the target samples as much as possible, while minimizes the chance of accepting outlier samples.

Because only the information of the target class is available in one-class classification, in order to set the boundary around the target class tightly, a exclusion threshold δ is set so that $1-\delta$ of the target samples is in the boundary and δ of the target samples is on or out of the boundary.

In our proposed system, one one-class classification model is built up for each known activity. The feature vectors after normalization and dimension reduction are used as the training samples (target samples) of the one-class classifier. Three one-class classifiers, K Nearest Neighbor (KNN) one-class classifier, Support Vector Data Description (SVDD) one-class classifier and Gauss one-class classifier [8] are compared.

D. Multi-Class Classification

Multi-class classification is the problem of separating a set of samples into two or more classes, and giving a criterion for determining whether a particular sample is or is not in a particular class. In multi-class classification, it is assumed that the information of each class is available and the samples in each class are assigned a particular class label. Using the samples of all the classes, a classification model is built and used to classify a testing sample into the most likely class in the learned classes.

We compare the performances of three multi-class classifiers. They are Decision Tree (DT) algorithm [7], K Nearest Neighbor (KNN) algorithm [9] and Weighted Support Vector Machines (WSVM) algorithm [10].

Support vector machine (SVM) is a new learning method developed in recent years based on the foundations of statistical learning theory. It is gaining popularity due to many attractive characters and promising empirical performance in the fields of nonlinear and high dimensional pattern recognition. However, because the class size may be uneven, the classification results based on SVM are undesirably biased toward the classes with more samples. In other words, the larger the sample number of a class, the smaller the classification error is; whereas the smaller the sample number of a class, the larger the classification error is [10]. In order to compensate for the unfavorable impact caused by the uneven class size, the WSVM is adopted to deal with this problem by assigning samples of different classes different weights.

E. Pattern Analysis and Segmentation

After noise data elimination, the remained continuous segments are the data of new activities. For each segment, if it contains two or more new activities, we should segment it into subsegments so that each subsegment contains only one activity. Pattern analysis aims to decide whether one segment contains only one new activity and, if not, how many subsegments it should have. In the proposed system, the pattern analysis is performed through user interaction. That is, it is the user who decides how many subsegments one segment should have. With the input, the PCA based bottom-up segmentation algorithm [11] is used to segment it into subsegments.

Assume time series $T = \{X_k | 1 \le k \le N\}$ is a finite set of N samples, where $X_k = [x_{1,k}, x_{2,k}, \cdots, x_{n,k}]^T$. A segmentation of T is a set of consecutive time points $S(a,b) = \{X_k | a \le k \le b\}$, $X_a, X_{a+1}, \cdots X_b$. The c-segmentation of time series T is a partition of T to c non-overlapping subsegments $S_T^c = \{S_i(a_i, b_i) | 1 \le i \le c\}$, such that $a_1 = 1$, $b_c = N$, and

$a_i = b_{i-1} + 1$. The bottom-up segmentation algorithm is shown in Fig.2. Based on the PCA model, the merging cost, $\cos t(S_i)$, can be calculated as the *Hotelling* T^2 measurement. Suppose $F_i = U_i \Lambda_i U_i^T$ is the covariance matrix of S_i, then:

$$\cos t(S_i) = \frac{1}{b_i - a_i + 1} \sum_{k=a_i}^{b_i} T_{i,k}^2 = \frac{1}{b_i - a_i + 1} \sum_{k=a_i}^{b_i} y_{i,k}^T y_{i,k} \qquad (9)$$

where $y_{i,k} = W_i^{-1}(S_i) = W_i^T(S_i)$, $W_i = U_{i,p} \Lambda_{i,p}^{\frac{1}{2}}$. p is the number of dominant features.

1. Create initial fine approximation
2. Find the cost of merging for each pair of segments: $merge\cos t(i) = \cos t(S(a_i, b_{i+1}))$
3. While stopping _ criteria is False
    1) Find the cheapest pair to merge: $i = \arg\min_i (merge\cos t(i))$
    2) Merge the two segments, update the a_i, b_i, boundary indices, and recalculate the merging costs.
       $merge\cos t(i) = \cos t(S(a_i, b_{i+1}))$
       $merge\cos t(i-1) = \cos t(S(a_{i-1}, b_i))$

   End

Figure 2. Bottom-up segmentation algorithm

III. EXPERIMENT AND RESULTS

A. System Setup

The acceleration data is collected using the KXP74 accelerometers [12]. Its sensitivity is programmed from -2.0g to +2.0g. This can fully meet our demands, for previous work shows promising activity recognition results from ±2.0g acceleration data [13].

The sensor is mounted onto the sensor board and sealed hermetically. The sensor node attached with the sensor board wirelessly transmits the data via RF signal to the base station, which is connected to the serial port of a laptop through an interface board and a serial cable.

Our experiment is focused on analyzing the human's movement activities, which include standing, walking, running, climbing up stairs and climbing down stairs. These activities can be differentiated from each other through the movements of the human's leg. Thus, we attach one sensor node on the front of the testee' right leg (near the ankle) using the elastic medical bandages. For different testees, the sensor node has the same deployment in the whole experiment process. As to complex human activities involving more body parts, several sensor nodes can be attached at the related places.

B. Data Collection

The sensor data of the accelerometer has the following attributes: time, acceleration along x-axis, acceleration along y-axis and acceleration along z-axis. It is collected with a sampling frequency of 32 Hz and no noise filtering is carried out.

Each of these five activities is performed by four testees. During data collection, all the testees carry out each activity in approximate frequency and intensity.

C. Experimental Results

Using the sensor data of these four testees, totally 1606 feature vectors of these five activities are obtained. After normalization, we randomly divide these feature vectors into two parts: the training dataset and the testing dataset. The training dataset has 811 feature vectors in total: 256 for standing, 225 for walking, 207 for running, 54 for climbing down stairs and 69 for climbing up stairs. The testing dataset has 795 feature vectors in total: 240 for standing, 214 for walking, 216 for running, 62 for climbing down stairs and 63 for climbing up stairs.

Then, the PCA is executed based on the training dataset. Fig.3 shows the correlation between the preserved energy, i.e. $1-\eta$, and the number of dominant features in the transform space. If the loss of energy is set as $\eta = 0.01$, 15 dominant features is enough. For simplification, in our experiments, the number of dominant features is set as 15. Based on the transformation model of PCA, the testing dataset is also dimension reduced.

1) One-Class Classifier Selection

From the training dataset, each activity dataset is in turn taken out and four one-class classification models are built up using the other four activity datasets. Then, the remaining one activity dataset is tested by these four one-class classification models and the fused results indicate the unknown detection performance of the one-class classifier. Table I shows the unknown detection accuracy for each activity of three one-class classifiers ($\delta = 0.05$).

From table I we can see that these three one-class classifiers get high detection accuracy for standing and running. In these five activities, standing and running are distinctive to be differentiated from any other activities. In the feature space, the feature regions of standing and running can be easily separated. Comparatively, because of the similarity among walking, climbing down stairs and climbing up stairs, their feature regions may be close and even partially overlapped. In this situation, the tighter the boundaries are, the higher accuracy we can obtain. Table I shows that the Gauss one-class classifier outperforms the other two, which indicates that the Gauss one-class classifier defines the tightest boundary for each activity.

According to the experimental results and above analysis, the Gauss one-class classifier is adopted in our proposed system.

2) Multi-Class Classifier Selection

Based on the training dataset, self-consistency test and cross-validated test are conducted to evaluate the classification performance of the multi-class classifiers. In the self-consistency test, the classification model is built up and tested using the training dataset. In the cross-validated

TABLE I. THE UNKNOWN DETECTION ACCURACY FOR EACH ACTIVITY OF THESE THREE ONE-CLASS CLASSIFIERS

	Standing	Walking	Running	Climbing down stairs	Climbing up stairs	Total
KNN	252/256=98.44%	80/225=35.56%	204/207=98.55%	7/54=12.96%	48/69=69.57%	591/811=72.87%
SVDD	256/256=100%	200/225=88.89%	204/207=98.55%	38/54=70.37%	51/69=73.91%	749/811=92.36%
Gauss	256/256=100%	201/225=89.33%	207/207=100%	47/54=87.04%	65/69=94.20%	776/811=95.68%

TABLE II. THE CLASSIFICATION ACCURACY FOR THE SELF-CONSISTENCY TEST OF THREE MULTI-CLASS CLASSIFIERS

	Standing	Walking	Running	Climbing down stairs	Climbing up stairs	Total
KNN	256/256=100%	225/225=100%	197/207=95.17%	46/54=85.19%	61/69=88.41%	785/811=96.79%
SVDD	256/256=100%	225/225=100%	206/207=99.52%	51/54=94.44%	66/69=95.65%	804/811=99.14%
Gauss	256/256=100%	225/225=100%	207/207=100%	54/54=100%	69/69=100%	811/811=100%

TABLE III. THE CLASSIFICATION ACCURACY FOR THE CROSS-VALIDATED TEST (n=4) OF THREE MULTI-CLASS CLASSIFIERS

	Standing	Walking	Running	Climbing down stairs	Climbing up stairs	Total
KNN	256/256=100%	224/225=99.56%	193/207=93.24%	37/54=68.52%	51/69=73.91%	761/811=93.83%
SVDD	254/256=99.22%	212/225=94.22%	200/207=96.62%	39/54=72.22%	52/69=75.36%	757/811=93.34%
Gauss	256/256=100%	220/225=97.78%	206/207=99.52%	48/54=88.89%	61/69=88.41%	791/811=97.53%

TABLE IV. THE CONFUSION MATRIX OF THE DETECTION AND CLASSIFICATION RESULTS WHEN THE STANDING ACTIVITY IS UNKNOWN

	Unknown	Walking	Running	Climbing down stairs	Climbing up stairs	Total
Unknown	240	0	0	0	0	240/240=100%
Walking	22	190	2	0	0	190/214=88.79%
Running	21	2	193	0	0	193/216=89.35%
Climbing down stairs	22	1	1	36	2	36/62=58.06%
Climbing up stairs	14	2	0	3	44	44/63=69.84%

test, the above dataset is randomly partitioned into n mutually exclusive and exhaustive ones. During test, each in the n sets is in turn taken out and classified using the rule parameters derived from the remaining $n-1$ sets. In our experiments, $n=4$.

Table II and table III show the classification accuracy for the self-consistency test and the cross-validated test of three multi-class classifiers, respectively.

Table II and table III show that WSVM algorithm obtains the highest classification accuracy in both self-consistency test and cross-validated test. In addition, as we mentioned in subsection 2.5, the WSVM can deal with the uneven class size problem. As shown in table III, WSVM obtains relatively high classification accuracy for climbing down stairs and climbing up stairs, though they both only have several tens of training samples. Accordingly, WSVM is adopted in our proposed system.

3) Detection and Classification Performance of the Proposed System

With Gauss one-class classifier and WSVM multi-class classifier, we evaluate the performance of the proposed system on unknown detection and classification.

Suppose the standing activity is unknown at first, and the other four activities are known. We use the training datasets of these four activities to train four Gauss one-class classification models and one WSVM multi-class classification model. Then, the testing dataset is tested and the detection and classification results are shown in table IV.

Table IV shows that the unknown activity (standing) can be detected accurately. The total accuracy for the unknown activity detection is 100% and the detection and classification accuracies for walking and running are about 90%. However, the accuracies for climbing down stairs and climbing up stairs are relatively low. The reason maybe is that the training samples of these two activities in the training dataset are much less, which are 54 and 69, respectively. Gauss one-class classifier is a density estimation based algorithm [8], scarceness of training samples will decrease its detection accuracy. We can also see from the first column of the confusion matrix that, some known samples are treated as unknown (we call this kind of error false unknown error). This originates from the tight boundary defined by the Gauss one-class classifier.

In the same way, the confusion matrixes of the detection and classification results of the proposed system when the unknown activity is walking, running, climbing down stairs and climbing up stairs are shown in table V, table VI, table VII and table VIII, respectively. These four tables have the similar characters with that of table IV.

4） Extension Recognition Capability of the Proposed System

A series of continuous activities of one testee are used to evaluate the extension recognition capability of the proposed system. These continuous activities in turn are standing, walking, running, climbing down stairs and climbing up stairs. Climbing down stairs and climbing up stairs are repeated once and there is a brief stay (standing) between them.

Each of above activities lasts a little time and there are several or tens feature vectors can be extracted from the sensor data. As we discussed in subsection 3.3.3, scarceness of training samples will decrease the detection accuracy of Gauss one-class classifier. In order to deal with this problem, the following resampling method is used to increase the number of training samples:

Suppose one activity has m ($m<200$) feature vectors, the mean and standard deviation of the i th feature are μ_i and σ_i. We sample $200-m$ new feature vectors and add them into the training dataset of this activity. For each new feature vector, its i th feature is modeled as random variable $\mathrm{N}(\mu_i,\sigma_i^2)$.

TABLE V. THE CONFUSION MATRIX OF THE DETECTION AND CLASSIFICATION RESULTS WHEN THE WALKING ACTIVITY IS UNKNOWN

	Standing	Unknown	Running	Climbing down stairs	Climbing up stairs	Total
Standing	227	13	0	0	0	227/240=94.58%
Unknown	0	192	16	1	5	192/214=89.72%
Running	0	21	195	0	0	195/216=90.28%
Climbing down stairs	0	22	1	36	3	36/62=58.06%
Climbing up stairs	0	15	0	3	45	45/63=71.43%

TABLE VI. THE CONFUSION MATRIX OF THE DETECTION AND CLASSIFICATION RESULTS WHEN THE RUNNING ACTIVITY IS UNKNOWN

	Standing	Walking	Unknown	Climbing down stairs	Climbing up stairs	Total
Standing	227	0	13	0	0	227/240=94.58%
Walking	0	190	24	0	0	190/214=88.79%
Unknown	0	0	216	0	0	216/216=100%
Climbing down stairs	0	1	22	37	2	37/62=59.68%
Climbing up stairs	0	2	14	3	44	44/63=69.84%

TABLE VII. THE CONFUSION MATRIX OF THE DETECTION AND CLASSIFICATION RESULTS WHEN THE CLIMBING DOWN STARIS ACTIVITY IS UNKNOWN

	Standing	Walking	Running	Unknown	Climbing up stairs	Total
Standing	227	0	0	13	0	227/240=94.58%
Walking	0	190	2	22	0	190/214=88.79%
Running	0	2	193	21	0	193/216=89.35%
Unknown	0	3	0	48	11	48/62=77.42%
Climbing up stairs	0	2	0	15	46	46/63=73.02%

TABLE VIII. THE CONFUSION MATRIX OF THE DETECTION AND CLASSIFICATION RESULTS WHEN THE CLIMBING UP STARIS ACTIVITY IS UNKNOWN

	Standing	Walking	Running	Climbing down stairs	Unknown	Total
Standing	227	0	0	0	13	227/240=94.58%
Walking	0	190	2	0	22	190/214=88.79%
Running	0	2	193	0	21	193/216=89.35%
Climbing down stairs	0	1	1	36	24	36/62=58.06%
Unknown	0	1	0	9	53	53/63=84.13%

At first, we take standing as known activity and the other as unknown activities. Because there is only one known activity, the multi-class classifier is not necessary then. The entire activity series is also used as the testing dataset. The recognition result is shown in Fig.3.

Fig.4 shows that the recognition system can detect the unknown activities correctly. For the four continuous unknown activity segments, according to the user interaction, we can know only the first segment contain three new activities and each of the other three segments only contain one activity. The PCA based bottom-up segmentation algorithm is used to segment the first segment. The initial number of subsegments is set as 20 and the number of dominant features for segmentation is set as $p = 4$.

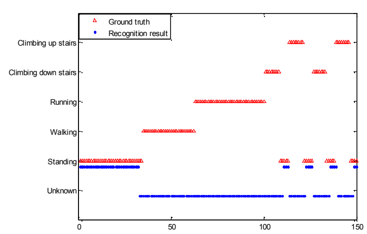

Figure 3. The recognition result (Standing is known)

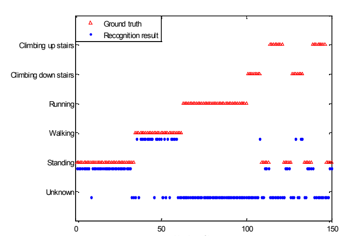

Figure 4. The recognition result (Standing and walking are known)

The first subsegment is walking. We extract its feature vectors and enlarge its number to 200 using the resampling method. After normalization and dimension reduction, the training samples for standing and walking are used to train their one-class classification models and the multi-class classification model. Then, the entire activity series is recognized again, and the result is shown in Fig.4.

Fig.5 shows some recognition errors. These errors come from three aspects: 1) the samples at the joint of two activities may be recognized as unknown samples; 2) as the one-class classifier defines the boundary through excluding δ target samples, there are some known activity samples are excluded as unknown ones; 3) because of the similarity between walking and climbing up/down stairs, some samples of climbing down stairs are wrongly recognized as walking.

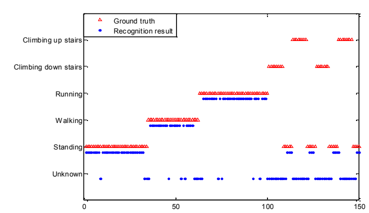

Figure 5. The recognition result (Standing, walking and running are known)

When the walking, climbing down stairs and climbing up stairs are in turn added into the recognition system, the recognition results are shown in Fig.5, Fig.6 and Fig.7, respectively. For climbing down/up stairs, we take their first segments as the raw training data.

From the recognition results we can see that, basically, the proposed extendable recognition system can recognize these activities correctly. One exception is, as shown in Fig.7, the system cannot recognize the second segment of climbing up stairs. The reason maybe is the difference of these two segments is larger than the difference of the raw feature vectors in the first segment.

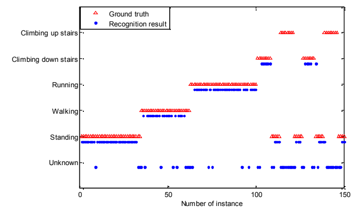

Figure 6. The recognition result (Standing, walking, running and climbing down stairs are known)

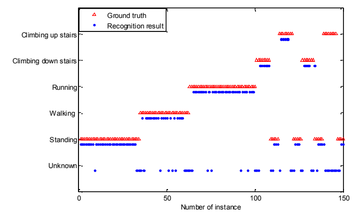

Figure 7. The recognition result (all the activities are known)

IV. DISCUSSION

Each activity has its own feature region in the feature space. If the feature regions of all the activities do not overlap, a known sample can at most be contained in one feature region. Then, the unknown sample detection and known sample classification can be finished simultaneously and no multi-class classifier is needed. However, if two activities are similar and their feature regions overlap with each other, the known samples may lie in the overlapping region. Then, the multi-class classifier is necessary to recognize their activity types. From section 3 we can see some activities are similar and their feature regions partially overlap with each other. Therefore, in the proposed recognition system, we combine the one-class classifier and multi-class classifier to detect the unknown samples and classify the known samples.

In the proposed recognition system, the bottom-up segmentation algorithm is used. This algorithm requires user to input the final number of segments, which decreases the system's intelligent degree. [11] proposed a clustering algorithm based fuzzy segmentation method. This method can automatically determine the required number of segments. However, it has two main disadvantages: 1) compared with the bottom-up segmentation algorithm, it needs much more computation cost; 2) the final number of segments is heavily affected by the segmentation parameters, such as the initial number of segments, fuzzy parameters and the merging threshold. Comparatively, it is much harder for user to determine these segmentation parameters than to determine the number of activity subsegments. Therefore, the bottom-up segmentation is adopted. During implementation, the occurrence time and the activity types before and after one segment are provided for helping the user to determine.

V. CONCLUSION

In this paper, an extendable activity recognition system is proposed. This system can detect the unknown activities, recognize the known activities, and extend the system's recognition capability.

Real experiments are conducted and the experimental results show the Gauss one-class classifier has the best detection performance and the Weighted Support Vector Machines (WSVM) multi-class classifier has the best classification performance. With the combination of Gauss one-class classifier and WSVM multi-class classifier, the system can correctly detect unknown activities and recognize known activities. In addition, based on the bottom-up segmentation algorithm, training samples of new activities are extracted and added into the recognition system to extend its recognition capability.

REFERENCES

[1] Lukowica P., Ward J. A. et al., Recognizing workshop activity using body worn microphones and accelerometers, In 2nd International Conference on Pervasive Computing, Vienna, Austria, Springer, 2004, pp.18-22.

[2] Ohgi Y., Yasumura M., Ichikawa H. and Miyaji C., Analysis of stroke technique using acceleration sensor IC in freestyle swimming, The Engineering of SPORT, Blackwell Publishing, 2000, pp.503-511.

[3] Mantyla J., Himberg J. and Seppanen T., Recognizing human motion with multiple acceleration sensors, In IEEE International Conference on Systems, Man and Cybernetics, Tucson, USA, IEEE Press, 2001, pp.747-752.

[4] Ravi N., Dandekar N., Mysore P. and Littman M., Activity recognition from accelerometer data, In proceedings of the 17th Annual Conference on Innovative Applications of Artificial Intelligence, Pittsburgh, USA, AAAI Press, 2005, pp.141-1546.

[5] Devaul R. W. and Dunn S., Real-Time Motion Classification for Wearable Computing Applications, Technical report, MIT Media Laboratory, 2001.

[6] Bao L. and Intille S. S., Activity recognition from user-annotated acceleration data, In 2nd International Conference on Pervasive Computing, Vienna, Austria, Springer, 2004, pp.1-17.

[7] Han J., Kamber M.，Data Mining: Concepts and Techniques, Morgan Kaufmann Publisher, 2000.

[8] Tax D. M. J., One-class classification, [Dissertation], Delft University of Technology, 1999.

[9] Duda R. O., Hart P. E. and Strok D. G., Pattern classification (2nd ed.), New York: Wiley, 2001.

[10] Wang M., Yang J., Liu G. P., Xu Z. J. and Chou K. C., Weighted SVM for predicting membrane protein types based on pseudo amino acid composition, Protein Engineering, Desigan & Selection, 2004, 17(4), pp.1-8.

[11] Abonyi J., Feil B., Nemeth S. and Arva P., Modified gath-geva clustering for fuzzy segmentation of multivariate time-series, Fuzzy Sets and Systems, 2005, 149(1), pp.39-56.

[12] KXP74 Series datasheet. www.kionix.com/Product%20Sheets/ KXP74%20Series.pdf.

[13] Kern N., Schiele B. and Schmidt A., Multi-sensor activity context detection for wearable computing, In European Symposium on Wearable Computers, Eindhoven, The Netherlands, IEEE Press, 2003, pp.220-234.