Labeling Method for Acceleration Data using an Execution Sequence of Activities

Kazuya Murao
Kobe University
1-1 Rokkodaicho, Nada, Kobe,
Hyogo, 657-8501 Japan
murao@eedept.kobe-u.ac.jp

Tsutomu Terada
Kobe University/Japan Science
and Technology Agency
1-1 Rokkodaicho, Nada, Kobe,
Hyogo, 657-8501 Japan
tsutomu@eedept.kobe-u.ac.jp

http://dx.doi.org/10.1145/2494091.2495982

Abstract
In the area of activity recognition, many systems using accelerometers have been proposed. Common method for activity recognition requires raw data labeled with ground truth to learn the model. To obtain ground truth, a wearer records his/her activities during data logging through video camera or handwritten memo. However, referring a video takes long time and taking a memo interrupts natural activity. We propose a labeling method for activity recognition using an execution sequence of activities. The execution sequence includes activities in performed order, does not include time stamps, and is made based on his/her memory. Our proposed method partitions and classifies unlabeled data into segments and clusters, and assigns a cluster to each segment, then assign labels according to the best-matching assignment of clusters with the user-recorded activities. The proposed method gave a precision of 0.812 for data including seven kinds of activities. We also confirmed that recognition accuracy with training data labeled with our proposal gave a recall of 0.871, which is equivalent to that with ground truth.

Author Keywords
Activity recognition, Accelerometer, Labeling

ACM Classification Keywords
H.4 [Information Systems Applications]: Miscellaneous

Introduction

Along with the progress in wearable computing, many contextaware systems with various kinds of sensors have recently been introduced, such as systems with accelerometers, gyroscopes, electromyographs[14], electrocardiograms[10], GSR (Galvanic Skin Reflexes)[8], and manually configured devices[5]. Contextaware systems are applied to many services i.e., health care[8], recognition of workers' routine activities[7], and support during assembly and maintenance tasks[11]. Camera, GPS, gyroscope, and geomagnetic sensor are also known as devices that obtain location and motions, however these sensors have low wearability and accuracy, and cannot obtain both motion and static direction simultaneously. An accelerometer can obtain motion and static direction by sensing earth's gravity, and has high accuracy and high resolution, and is enough small to be attached on the body. For this reason, an accelerometer is the best device for activity recognition.

An activity recognition system flows as shown in Figure 1. The system obtains sensor data from sensors attached to user's body or installed in a phone that the user has. The obtained data is transmitted to a small computer or smart-phone, and converted to feature values, such as mean, variance, and Fourier's coefficient, which indicate body's direction, intensity of movement, and frequency of movement, respectively. The system has to learn the models of classifier, such as SVM or HMM, with the feature values and their correct activity labels (ground truth). Then the system recognizes unknown data by converting it to feature value and comparing it with the models.

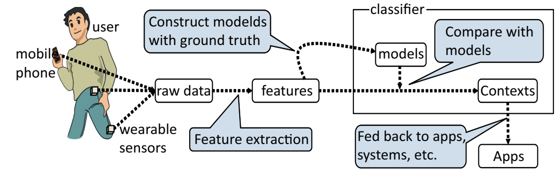


Figure 1: Activity recognition system.

There are some methods for labeling: acceleration waveform is collated with a video of activities recorded during data collection, and with a memo of activities with timestamps taken every time activity changes. Although a tool for the labeling method using video-recording has been developed[1], it is a manual work and takes longer time of the original data. In the method using a memo, noise is included in the data since memo is taken during activities, moreover it is bothersome to record activities whenever they change. It is possible that a person goes with the user to record him/her activities, however it is not good for an experiment since the subject is kept under surveillance every time and collected data is not natural. Recently, sensors with Bluetooth connectivity and sensors in a smartphone make it easy to collect data, but labeling task still requires much amount of time and labor, which is a barrier to construct activity recognition system with ease. Some works propose methods to label unlabeled data with small amount of labeled data. However, their evaluations are conducted in the environment where the amount of label is evenly decreased over the time, which means that frequency of labeling lessen but labeling is still needed. It is not a fundamental solution.

This paper proposes a labeling method for activity recognition. The proposed method uses raw data and an execution sequence of activities that the user performed. The sequence of activities is supposed to be made during a break in data collection at cafeteria or after the data collection. The user records activities that have been done until that time as far as he/she remembers. The sequence of activities includes activities in an execution order and does not include timestamps. We assume that the sequence of activities allows lack of activities since all the activities cannot be memorized. Our proposed method partitions and classifies unlabeled data into segments and clusters, and assigns a cluster to each segment, then assign labels according to the best-matching assignment of clusters and activities with the execution sequence of activities.

Related work

Most of activity recognition systems that have been proposed thus far use fully labeled data to train models. A way of learning fully labeled data is called (full-)supervised learning. On the other hand, a way of learning partially labeled data is called semi-supervised learning. Both supervised and semi-supervised learning use a classifier to obtain resultant activities, such as SVM or HMM. These classifiers require fully labeled data, and unlabeled data is discarded. Semi-supervised learning spread labels of labeled data to unlabeled data, then learning process is carried out as well as supervised learning. The spread labels are inference and may not be correct.

This section introduces several studies on semi-supervised learning. In [12], several semi-supervised learning methods are compared and Self-training algorithm[2] is introduced as one of the simple methods. Suppose few amount of labeled data and large amount of unlabeled data, self-training algorithm constructs models from labeled data, then classify unlabeled data with the models. The classification results are fed back to the unlabeled data as labels and all the data got labeled, then recognition models are re-constructed with all the data.

Maja et al.[13] proposed a method to spread labels of labeled data to unlabeled data by focusing on the fact that labels of similar data in feature domain and time domain are probably same. This method constructs graphs by assuming data is node and similarity is vertex, and calculates similarity among data from the distance of feature values and time difference. Labels of labeled data are spread to unlabeled data whose similarity is high. The amount of labels for each activity is controlled so as to follow the prior distribution of activities. In the evaluation, labeling accuracy is measured for labeling intervals from 10 to 180 minutes, obtaining 90% accuracy for 10 minutes interval and 55% accuracy for 180 minutes interval. Labeling in 10 minutes equals 2.5% of all data and 180 minutes interval equals 0.1% of all data.

A study using eigenspace has been reported[4]. One eigenspace is obtained with principal component analysis (PCA) that is mainly used for dimension reduction of multidimensional data, however multiple number of eigenspace can be found with multiple eigenspace algorithm, applying it to acceleration data, each sample belongs to one of the eigenspaces. This study focuses on the fact that there is a relation between eigenspace and activity, and uses indices of eigenspace as labels to train SVM, then consolidates the eigenspaces on error ratio. Giving small amount of labeled data to the eigenspace, indices of eigenspace and activities are associated. In the evaluation, for eight kinds of activities, 88.3% of recognition accuracy with 80%-labeled data and 80.3% with 20%-labeled data are obtained. it is confirmed that these results are higher than the result for the method using labeled data only.

As stated above, many studies on semi-supervised learning have been done, showing results of accuracy when 20% of data is labeled or when interval of labeling is 60 minutes. However, 10,800 samples are collected from sampling for three hours at 10 Hz, 0.1% of which is as many as 10 samples. This means users have to record activity 10 times evenly in three hours. Long interval of sampling also forces the users to record their activities in the fixed time, which is not considering users' state. Though it is possible to record activities after a sequence of activities finishes, it is hard to remember correct timestamps for each activity.

This paper proposes a labeling method with an execution sequence of activities that the user recorded. The execution sequence we assume in this paper is made by users when they have a rest, does not include time stamps, and allows a lack of activities. Information on activities extracted from web such as twitter and facebook is also available since our proposed method does not use time information.

Proposed method

The proposed method consists of three phases: segmentation, clustering, and labeling, as shown in Figure 2. The segmentation phase partitions the data sequence into multiple segments according to change in acceleration value in the time domain. The clustering phase classifies the data samples into clusters by analyzing a distribution of acceleration values in the feature domain. Finally, the labeling phase merges the segments and clusters, and obtains a sequence of clusters, then finds the best matching assignment of activities and clusters by comparing the sequence of clusters and an execution sequence of activities. The data is labeled based on the best matching pattern. This section explains the three phases in detail. In this paper, we assume that user attaches three accelerometers on their left wrist, hip, and right ankle. Activities to be recognized are seven kinds: sit, stand, lie, walk, run, ascend the stairs, and descend the stairs. Input data includes all the activities and may include undefined activities since users freely move. The sampling frequency is 100 Hz.

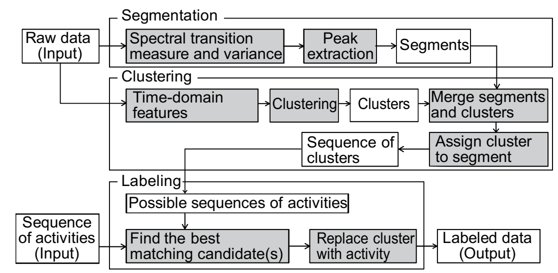


Figure 2: Flow of the proposed method.

Segmentation

The aim of the segmentation phase is to find changing points of activity and partition the acceleration data into segments. This paper adopts a phonetic segmentations with spectrum transition measure, which is a fundamental method for phonetic segmentation in the field of speech recognition[3]. In detail, spectrum transition measure (STM) $G_t(t = 1, \cdots, T)$ of acceleration data $X = \{x_1, x_2, \cdots, x_T\}$ is calculated in the following equation, where x_t is an acceleration value at time t. For the sake of simplicity, we assume there is data for one axis. T is the number of samples of acceleration data.

$$G_t = \frac{\sum_{i=1}^{p} a_{i,t}^2}{p}, \quad (1)$$

where $a_{i,t}$ is the rate of change of the spectral feature

$$a_{i,t} = \frac{\sum_{n=-M}^{M} C_{i,t+n} \cdot n}{\sum_{n=-M}^{M} n^2}, \quad (2)$$

p and M are the dimension of the spectral feature vector and window size for the change in the spectral feature, respectively, and we set $p = 10$ and $M = 1$. $C_{i,t}$ is the ith coefficient of the feature value in the frequency domain $(1 \leq i \leq p)$ extracted from the acceleration data $X'_t = \{x_{t-w}, \cdots, x_t\}$ over the time $[t - w, t]$. We adopt Cepstrum as a feature value. Cepstrum involves information on the degree of spectral change, and is derived as follows: original wave → FFT → power → log → phase unwrapping → inverse FFT → real part. The window size w should cover one cycle of a motion, but large window causes delay of recognition. We set $w = 256$, which is equivalent to 2.56 seconds, and slid in steps of one sample.

STM of acceleration waveform G_t shows small values while same activity is lasting and shows a high peak at changes in activities. Our method partitions the waveform at the peaks as follows; all peaks p_t that meets $(G_t - G_{t-1} > 0)$ and $(G_t - G_{t+1} > 0)$ are extracted, and the waveform is partitioned at the peaks meeting $p_t > \alpha$. The interval of segments becomes large for large α since the number of peaks is small, which may miss out change in activity. On the other hand, the interval of segments becomes small for small α since many peaks are found, which may partition at the points where activity does not change. The proposed system will merge similar segments upon the result of clustering in the next phase, thus α is determined so that the acceleration data are partitioned at all the changing points of activities in this phase. We set $\alpha = 5$ from the preliminary experiment.

STM can find changing points of activities while the waveform is fluctuating, however STM may not find change in activities between static activities, since high peaks do not appear at these points. For this reason, we adopts running variance to find the changes between static activities. Variance σ_t^2 are extracted from acceleration data X'_t over the window w at time t. σ_t^2 takes almost zero while static activity, and takes large value while moving, therefore our system partitions the data at time t when $((\sigma_{t-1}^2 < 1000) \cap (\sigma_t^2 \geq 1000))$ or $((\sigma_{t-1}^2 \geq 1000) \cap (\sigma_t^2 < 1000))$. Since it is difficult to find change in activities while moving only with variance, all the changes in activities can be found by combining STM and variance.

We have explained our method for 1-axis acceleration values. A sensor is actually consists of three axes, and couple of sensors are often used. For multiple dimensional data, the above procedure is applied to each axis and logical disjunction of the changing points are meant as changing points overall. Partitioning acceleration data into several segments in the time domain, segment number is assigned to each segment and a sequence of segment number *Segment* $= \{s_1, \cdots, s_T\}$ is obtained. In this paper, data on the partition belongs to the adjacent segment of younger number. At this stage, input data is divided into several sub-sequences, but no information on activities performed in each segment and similarity between segments is acquired.

Clustering

Clustering phase classifies acceleration data into clusters, then assigns one cluster number to each segment based on the composition of cluster numbers in the segment, finally obtains a sequence of cluster numbers.

Clustering acceleration data. In this work, we employ EM clustering algorithm[6]. Acceleration data is converted to mean and variance over a sliding window as feature values commonly used for activity recognition, then cluster numbers are given to each data with the EM clustering in the feature space. In detail, acceleration data $X = \{x_1, x_2, \cdots, x_T\}$ is converted to feature values $F = \{f_1, f_2, \cdots, f_T\}$, where x_t is a d-dimensional acceleration vector at time t, f_t is a feature vector at time t whose dimension is $2d$ since mean and variance are extracted from each axis. Cluster numbers $C = \{c_1, c_2, \cdots, c_T\}$ are obtained with clustering for the feature values F. EM clustering algorithm estimates mean, variance, and mixture weights of a Gaussian Mixture Models (GMM) on the assumption that the input data are generated from K models, then classifies data by judging which model the data came from. Assuming a case of collecting data in real environment, unexpected activities may be included, therefore the number of activities included in the captured data is unknown. Though the simple K-means algorithm requires the number of clusters K beforehand, EM algorithm finds proper K by calculating likelihood of the resultant models with 10-fold cross-validation from $K = 1$ and incrementing K till the likelihood does not increase any more. In addition, EM clustering performed better than the other algorithms such as K-means, X-means and hierarchical clustering even if proper K is given.

Clustering segments. Since clustering in feature domain does not consider time-series, concavo and convex parts of acceleration waveform may belong to different clusters even while a same activity is lasting, as shown in Figure 3. In the second step of clustering, our method merges clusters which would be derived from same activity by analyzing the constitution of cluster numbers in the segments, then assigns one cluster to one segment.

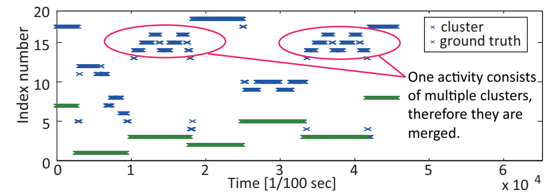


Figure 3: Example of clusters.

When multiple cluster numbers are in a segment, data of these clusters are meant to be derived from same activity and the clusters are merged to one. For example in Figure 3, cluster numbers 14, 15, and 16 are assigned to the activity performed in the third (activity index is 3), and the same activity performed in the fifth has same constitution of clusters. Table 1 shows index number of activity used in this paper. Our method reconstructs clusters by assigning cluster 15 and 16 to 14 with the following index. This index leverages COSINE[9]: a measure of relevance ratio of words from their cooccurrence.

$$I(x, y) = \frac{\sum_{S=1}^{N_S} \sqrt{f_{S_x} \cdot f_{S_y}}}{\sqrt{f_x \cdot f_y}} \tag{3}$$

,where $I(x, y)$ is the degree of cooccurrence of cluster x and y, f_{S_x} is the number of samples with cluster x in segment S, f_x is the number of samples with cluster x overall, and N_S is the number of segments. The degree of cooccurrence of two clusters which belong to same segment reaches one, and the degree of cooccurrence of two clusters which do not belong to same segment reaches zero. Clusters whose degree of cooccurrence is more than β are merged. We set $\beta = 0.4$ from our pilot study. Heuristically, the number of clusters should be less than double of the number of activities. After the integration, the cluster number with the highest proportion in a

Index number	Activity
1	Sit
2	Stand
3	Lie
4	Walk
5	Run
6	Upstairs
7	Downstairs

Table 1: Activity number and activity.

segment is assigned to the segment, finally a sequence of cluster $SC = \{SC_1, \cdots, SC_{N_S}\}$ is obtained.

At this stage, a sequence of cluster numbers in the time domain has been obtained. In the segments with same cluster numbers, the user seemed to have performed same activity, however what activity had been performed is unknown. In the next section, activity labels and cluster numbers are associated.

Labeling

Labeling phase gives activity labels for the input data by associating cluster numbers and activities that the user recorded. Suppose the number of clusters after integration is m and the number of activities to be recognized is n, possible patterns of association becomes as follows:

$$ {}_nC_m \quad (m \leq n) \tag{4} $$

$$ \frac{m!}{(m-n)!} \quad (m > n) \tag{5} $$

The reason that the number of clusters becomes more than that of activities that one activity may be assigned to multiple clusters, and unexpected activities may be included in the input data. Since it is hard to judge which reason would be, our method take an approach not to force labels, but to give up labeling as "unknown".

From each pattern of assignment, predicted activity sequence $PS = \{PS_1, \cdots, PS_i, \cdots, PS_{N_S}\}$ is obtained by calculating distance to the user-recorded activity sequence $AS = \{AS_1, \cdots, AS_j, \cdots, AS_{N_A}\}$, where N_A is the length of the user-recorded activity sequence. We employ dynamic programming (DP) matching algorithm which is one the string comparison algorithms, to calculate distance of two sequences. The algorithm works as follows. $N_S \times N_A$ matrix d_{ij} is a penalty of mismatching: $d_{ij} = 0$ when $PS_i = AS_j$ and $d_{ij} = 1$ when $PS_i \neq AS_j$.

1. **Initialization**:
   $Cost(0,0) = 0$
   $Cost(i,0) = \infty$ for $i = 1, \cdots, N_S$
   $Cost(0,j) = \infty$ for $j = 1, \cdots, N_A$

2. **Cost calculation**:
   Do for $i = 1, 2, \cdots, N_S$
   Do for $j = 1, 2, \cdots, N_A$

$$ Cost(i,j) = min \begin{cases} Cost(i-1, j-1) + d_{ij} \\ Cost(i-1, j) + d_{ij} * 2 \\ Cost(i, j-1) + d_{ij}/2 \end{cases} \tag{6} $$

3. **Output**:
   Return $Cost(N_S, N_A)$

Here, we describe the reason that the coefficient of the penalty d_{ij} is different in Equation 6. Let the real number of changes in activity be N, the relational expression $N_S > N$ is approved since the proposed method partitions acceleration data at no less than all the points where activities really change. On the other hand, the relational expression $N_A < N$ is also approved since the user-recorded activity sequence includes real activity sequence at most and may lack the activities due to forgetfulness and unconsciousness. Therefore, basically N_S is larger than N_A. The penalty for $Cost(i-1, j)$ in Equation 6 is halved since i grows many times as a necessity, and the penalty for $Cost(i, j-1)$ is doubled since growth of j neglects the user-recorded activity sequence.

The obtained $Cost(N_S, N_A)$ is the distance between PS and AS, meaning that the smaller it is, the predicted sequence is close to the user-recorded sequence. The distance for all the possible predicted sequences are calculated, and the cluster numbers are replaced with the activity index on the basis of the predicted sequence with the smallest distance. When that multiple candidates mark the smallest distance, this method selects the candidate that the amount of unknown label is the least since the amount of unexpected activity is basically small.

Lastly we describe the case that the user-recorded activity sequence includes incorrect activities. Supposing that the user performed "A→B", but recorded as "A→C" or "A→C→B", incorrect labels are apt to be given. However, it rarely occur that certain activity is incorrectly recorded many times. Increasing the amount of data collected, the influence of the misrecording would deteriorate and correct predicted sequence can be found.

Evaluation

Evaluation environment
Data were captured from three male subjects aged 22 to 29 years, who attached three 3-axis accelerometers (Wireless Technologies, Inc., WAA-010) on their left wrist, hip, and right ankle, a tablet computer (Mouse Computer Co.,Ltd., LuvPad WN701) on their back to receive data from the sensors, and wearable camera (Sony Corporation, HDR-AS15) on the head to record ground truth activities. They acted seven activities: sitting, standing, lying, walking, running, ascending stairs, and descending stairs. The sampling frequency of the accelerometers was 100 Hz.

The data collection is conducted in the following procedure. The experimenter told the subjects that the objective of the experiment is to collect data for the seven activities. The order of activities and duration of each activity are free. The subjects are allowed to perform certain activities many times, but forced to perform them at least once. Termination of the experiment is decided by the subjects themselves when they think seven kinds of activity have been collected. The experimenter does not go together with the subjects, and checks them through video captured with the wearable camera mounted to them. After the experiment, the experimenter collects acceleration data and video of wearable camera, then manually annotates the data by watching the video. Since the subjects freely acted while the experiment, unexpected activities are included in the data, which are labeled with "other" whose index number is -1.

The sequence of activities are recorded with an Android application developed by the authors. Buttons of seven kinds activity, [Delete] button, [OK] button, and [Finish] button are displayed and the subject put the buttons of activity in the performed order. If the user missed the input, it can be deleted by pushing [Delete] button. After inputing activities, the activity sequence is stored in the application by pushing [OK] button. The subjects are told to do this task at their timing. After finishing the experiment and inputing the activity sequences, all the stored activity sequences are connected and mailed to the experimenter by pushing [Finish] button.

Table 2 shows the collected data. Ground truth is a correct sequence of activities obtained from the video (not user-recorded activities). # activity is the total number of activities. Duration is time from start of sensing to time to push [Finish] button of the application. The subjects start the experiment from our laboratory located on the seventh floor, then move around the laboratory, pass through a hallway, walk up and down stairs, go out of the building, and move around the building.

Table 3 shows user-recorded activities. Time in the table shows the time to push [OK] button of the application. Sequence of activities is a order of activities that the subjects recorded. For example, in the first line for subject

Subject	# activity	Duration [sec]	Ground truth
A	19	875	Stand→Walk→Sit→Stand→Walk→Undefined (put on shoes)→Walk→Stand→Up→Stand→Down →Up→Stand→Run→Stand→Lie→Sit→Walk→Sit
B	58	1695	Stand→Walk→Stand→Walk Undefined (put on shoes)→Walk→Stand→Walk→Up→Stand→Down →Run→Stand→Down→Sit→Down→Walk→Run→Stand→Walk→Up→Walk→Stand→Walk→Up →Down→Sit→Up→Walk→Stand→Up→Walk→Run→Up→Walk→Stand→Walk→Up→Down→Sit →Walk→Run→Down→Undefined→Walk→Up→Sit→Up→Walk→Up→Walk→Undefined→Sit→Lie →Undefined→Stand→Walk→Sit
C	16	810	Stand→Walk→Down→Walk→Up→Stand→Walk→Run→Undefined→Run→Walk→Up→Sit→Walk →Lie→Sit

Table 2: Data for evaluation.

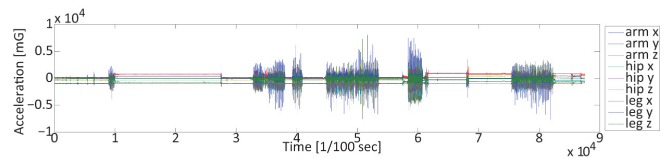
Sub-ject	Time [sec]	Sequence of activities
A	379	Sit→Stand→Walk
	562	Up→Stand→Down →Up→Stand
	875	Run→Lie→Sit→Walk →Sit
B	126	Walk→Stand→Walk →Stand
	180	Walk→Up→Stand
	279	Down→Run→Stand
	368	Down→Sit
	481	Down→Walk→Run
	662	Walk→Up→Walk→Stand
	763	Walk→Up→Down→Sit
	917	Up→Walk→Stand
	1037	Up→Run→Up→Walk →Stand
	1197	Walk→Up→Down→Sit
	1308	Walk→Run→Down →Sit→Lie
	1591	Walk→Up→Sit→Up →Walk→Up→Walk→Sit
	1695	Lie→Stand→Walk→Sit
C	259	Stand→Walk→Down
	494	Walk→Run
	696	Walk→Up→Sit
	810	Walk→Lie

Table 3: Sequence of activity that subjects recorded.

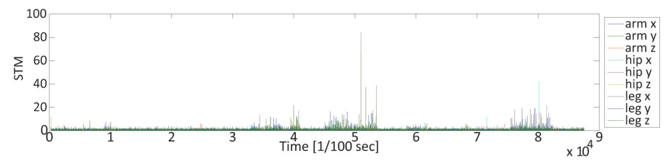
A indicates that subject A had performed sitting, standing, and then walking when 379 seconds passed after start sensing. Comparing ground truth in Table 2 with the recorded activities in Table 3, some of activities are missing in the recorded activities. This is because frequent activities such as walking, and short activities are not realized by the subject, therefore less likely to be recorded.

Results and consideration

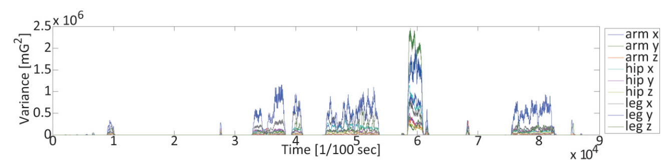
Clustering and segmentation performance. Waveform of acceleration, spectrum transition measure, running variance, results of clusters and segments, clusters and segments after integration, and ground truth for the data of subject A are shown in Figure 4. From Figure 4(b), peaks appears at the changing points of activity in the STM obtained from raw data (Figure 4(a)). Result of segmentation based on the STM and running variance (Figure 4(c)) and result of clustering in the feature domain are shown in Figure 4(d). At this stage, data are partitioned into minute segments, and multiple indices of clusters belong to one segment. However, only one cluster index is assigned to one segment by integrating segments based on the cooccurrence of clusters, as shown in Figure 4(e). Comparing the clusters after the integration with ground truth, timings of changes in cluster index and activities are matching.



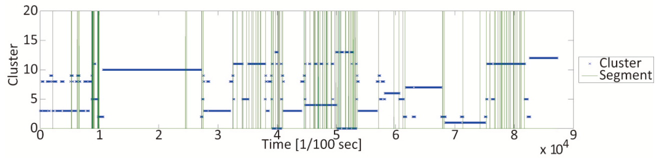
(a)Waveform of acceleration.



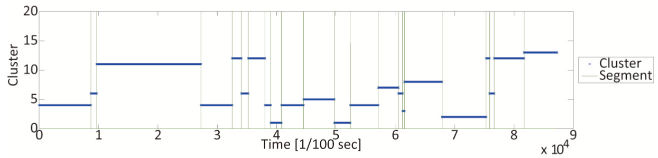
(b)Spectrum transition measurement.



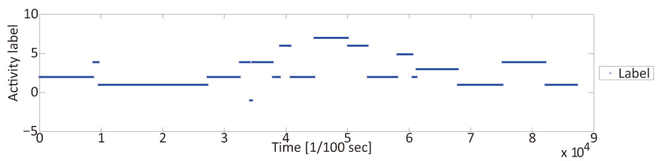
(c)Running variance.



(d)Clusters and segments.



(e)Clusters and segments after integration.



(f)Ground truth.

Figure 4: Waveforms obtained through the proposed method.

Table 4 shows the detailed results of segmentation and clustering. Data for each subject are partitioned into 51, 173, and 71 segments, which are more than the actual total number of activities. This is because acceleration data is partitioned at the points even when the user moved his/her leg while standing, or the user tripped while walking. Moreover, data for each subject are divided into 14, 18, and 26 clusters. This is because undefined activity holds a cluster, or one activity holds multiple clusters. Even if one consequent activity is divided to two or more clusters, the clusters are merged based on the cooccurrence of clusters, resulting in 22, 73, and 16 segments, which are close to the total number of activity in Table 2. The number of clusters are 11, 12, and 10, which are close to the kinds of activity.

Subject	# segment	# cluster	# segment after integration	# cluster after integration
A	51	14	22	11
B	173	18	73	12
C	71	26	16	10

Table 4: Results for segmentation and clustering.

Labeling accuracy. Table 5 shows a labeling precision and recall for the proposed method. The number of best-matching patterns with the user-recorded activities for subject A, B, and C were three, one, and three, respectively. The results shown in the table are of the pattern with the highest percentage of labeled data since the amount of undefined activities is less than that of defined activities. Precision indicates how accurate the labels are, and recall indicates how accurate the input data are labeled.

	Subject A (96.3% labeled)		Subject B (94.1% labeled)		Subject C (86.3% labeled)	
	Precision	Recall	Precision	Recall	Precision	Recall
Sit	0.923	0.171	0.588	0.735	0.477	0.481
Stand	0.963	0.921	0.95	0.784	0.994	0.883
Lie	0.211	0.995	0.851	0.785	0.0696	0.0193
Walk	0.982	0.763	0.892	0.572	0.953	0.854
Run	0.723	0.997	0.77	0.736	0.935	0.777
Up	0.913	0.828	0.64	0.901	1.00	0.900
Down	0.987	0.941	0.994	0.912	0.928	0.939
Ave.	0.815	0.802	0.812	0.775	0.765	0.693

Table 5: Labeling accuracy.

From the results, an average precision for subject A and C are 0.815 and 0.765, respectively, however *lie* of subject A and *lie* and *sit* of subject B are low. This is because length of the data for subject A and C are relatively short, and one activity is conducted approximately two times. Since our proposed method is not perfect, misses may occur in segmentation and clustering phases, such as two kinds of activities belong to one segment, one activity is divided into two segments, and two intervals of same activity belong to different clusters. When the number of activity performed is small, influence of such misses relatively increases, therefore incorrect answer is likely selected.

An average precision for subject B is 0.812 and evenly high degree of precision is obtained over all the activities although the result of *sit* and *up* activities are bit low. This is because length of the data for subject B is approximately double of subject A and C, the influence of misses in the segmentation and clustering phases decreases. In other words, for long input data, it rarely occur that all the misses happen to specific activity and are compatible to the user-recorded activity sequence.

From these results, labeling performance get stable as the input data is lengthened. Collecting data that the user freely acted for 30 minutes and an execution sequence of activity, 94% of input data are labeled and 81% of the labels are correct.

Recognition accuracy. Lastly, we measured recognition accuracy when a classifier is trained with data labeled with a comparison method and proposed method. The comparison method is a kind of baseline that ground truth is given as labels. Table 6 and Table 7 show recognition accuracy for the comparison method and the proposed method, respectively. From the results, high degree of precision and recall are obtained for the comparison method since the labels of the comparison method are ground truth. On the other hand, the proposed method showed that recall of *Sit* for A and C are quite low, which accord with the result of labeling accuracy in the previous section. From the results for B, an average recall 0.871 is obtained, which is high degree of performance compared with that of the comparison method.

	Subject A		Subject B		Subject C	
	Precision	Recall	Precision	Recall	Precision	Recall
Sit	0.991	0.994	0.967	0.991	0.988	0.997
Stand	0.973	0.987	0.929	0.985	0.965	0.947
Lie	0.993	0.989	1.00	0.963	0.993	1.00
Walk	0.89	0.964	0.873	0.913	0.887	0.936
Run	0.994	0.976	0.986	0.925	0.984	0.982
Up	1.00	0.832	0.919	0.875	0.993	0.872
Down	0.969	0.861	0.931	0.851	0.916	0.917
Ave.	0.973	0.943	0.944	0.929	0.961	0.950

Table 6: Recognition accuracy for training data labeled with ground truth.

	Subject A		Subject B		Subject C	
	Precision	Recall	Precision	Recall	Precision	Recall
Sit	0.826	0.174	0.821	0.999	0.474	0.481
Stand	0.956	0.937	0.935	0.951	0.985	0.883
Lie	0.210	0.990	1.00	0.807	0.008	0.0100
Walk	0.906	0.859	0.977	0.607	0.907	0.938
Run	0.624	1.00	0.953	0.922	0.979	0.983
Up	0.977	0.83	0.636	0.930	1.00	0.860
Down	0.983	0.828	0.926	0.880	0.920	0.948
Ave.	0.783	0.803	0.893	0.871	0.753	0.729

Table 7: Recognition accuracy for training data labeled with the proposed method.

Conclusion

We proposed a labeling method for acceleration data using an execution sequence of activities. The sequence of activity includes order of activity that the user performed, and does not include timestamps. From the evaluation, we have confirmed that 81.2% of the labels are correct for the data including seven kids of activities that user freely performed. Moreover, recognition accuracy when SVM is trained with the data labeled with the proposed method was 0.871, which is equivalent to 0.929: accuracy with training data labeled with ground truth. The activity sequence is easy to be collected since timestamps is not included. We will propose a method using activity information extracted from Web, such as Twitter and Facebook.

Acknowledgements

This research was supported in part by a Grant-in-aid for PRESTO from Japan Science and Technology Agency.

References

[1] ATR-Promotions. Syncplay. http://www.atr-p.com/SyncPlay.html.
[2] Chapelle, O., Schölkopf, B., and Zien, A. Semi-supervised learning. In *Adaptive computation and machine learning. MIT Press* (2006).
[3] Dusan, S. and Rabiner, L.R. On the relation between maximum spectral transition positions and phone boundaries. In *International Conference on Spoken Language Processing (INTERSPEECH 2006 – ICSLP)* (2006), 645–648.
[4] Huỳnh, T. and Schiele, B. Towards less supervision in activity recognition from wearable sensors. In *International Symposium on Wearable Computers (ISWC 2006)* (2006), 3–10.
[5] Laerhoven, K.V. and Gellersen, H.W. Spine versus porcupine: a study in distributed wearable activity recognition. In *International Symposium on Wearable Computers (ISWC 2004)* (2004), 142–149.
[6] McLachlan, G.J. and Krishnan T. The em algorithm and extensions. In *John Wiley and Sons* (1997).
[7] Naya, F., Ohmura, R., Takayanagi, F., Noma, H., and Kogure, K. Workers' routine activity recognition using body movement and location information. In *International Symposium on Wearable Computers (ISWC 2006)* (2006), 105–108.
[8] Ouchi, K., Suzuki, T., and Doi, M. Lifeminder: A wearable healthcare support system using user's context. In *International Workshop on Smart Appliances and Wearable Computing (IWSAWC 2002)* (2002), 791–792.
[9] Peat, H.J. and Willett, P. The limitations of term co-occurrence data for query expansion in document retrieval systems. In *Journal of the American Society for information science*, vol. 42 (1991), 378–383.
[10] Shen, C.L., Kao, T., Huang, C.T., and Lee, J.H. Wearable band using a fabric-based sensor for exercise ecg monitoring. In *International Symposium on Wearable Computers (ISWC 2006)* (2006), 143–144.
[11] Stiefmeier, T., Ogris, G., Junker, H., Lukowics, P., and Tröster, G. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *International Symposium on Wearable Computers (ISWC 2006)* (2006), 97–104.
[12] Stikic, M., Laerhoven, K.V., and Schiele, B. Exploring semi-supervised and active learning for activity recognition. In *International Symposium on Wearable Computers (ISWC 2008)* (2008), 81–88.
[13] Stikic, M., Larlus, D., and Schiele, B. Multi-graph based semi-supervised learning for activity recognition. In *International Symposium on Wearable Computers (ISWC 2009)* (2009), 85–92.
[14] Toda, M., Akita, J., Sakurazawa, S., Yanagihara, K., Kunita, M., and Iwata, K. Wearable biomedical monitoring system using textilenet. In *International Symposium on Wearable Computers (ISWC 2006)* (2006), 119–120.