IPSJ Journal Vol. 0 1–10 (??? 1959)

Regular Paper

A Large Scale Gathering System for Activity Data using Mobile Devices

Yuichi Hattori[†1] and Sozo Inoue[†1]

In this paper, we show a large-scale activity gathering system with mobile sensor devices such as smart phones and accelerometers. We gathered over 35,000 activity data points from more than 200 people over approximately 13 months. We describe the design rationale of the system, and analyze the gathered data through statistics, clustering, and application of an existing activity recognition method. Our results show that there is a challenging field for activity recognition in larger-scale activity data.

1. Introduction

If human activity can be objectively measured, we can expect various applications. For example, lifestyle aspects can be quantified and used for prevention of lifestyle-related diseases.

In this research, we aim to gather open data sets for evaluating activity recognition methods which already exist or will be proposed in the future. For this, we developed a large-scale activity data gathering system named ALKAN. ALKAN is a server-client system to gather a large number of missions with mobile sensor devices such as smart phones with accelerometers. It enables a simple way of recording, and semi-automatic way of uploading activities.

In this paper, we show the design of ALKAN, and show the data obtained by large-scale experiments by over 200 users and over 35,000 activities. It reveals new challenges for activity recognition, in the sense that there are more complex situations in real activities than laboratory settings.

This paper is organized as follows. Related work is described in Section 2. The ALKAN system is described in Section 3. Gathered activity data are overviewed in Section 4. Section 5 shows the nature of gathered data by applying an existing activity recognition method. Section6 gives the summary and discussion for future work.

†1 Kyushu Institute of Technology

2. Related Work

In the literature, a lot of work has tried to recognize activities with sensor devices. Chambers et.el.[1] tried to distinguish activity and movement of arm using two sensors on the arm. Laerhoven and Cakmakci.[7] tried to distinguish activities, postures, and riding bicycle, using two sensors at waist. However, these works experiment for only one user. They don't do for varieties of people like this paper.

Lee and Mase.[4] recognizes the types and the strength of movement with eight users using several sensors at waist and thighs. Mantyjarvi et. el.[5] uses 6 sensors on the waist, and recognized activities and postures with six users. Laerhoven and Cakmakci.[7] attaches 2sensors on the back thigh, and 7 activities, postures, and bycicles are recognized with 10 users. Herren et. el.[2] uses 2 sensors on the back and on feet, and recognizes angles and walking speed with 20 users.

However, these study aquire activity data in semi-artificial environments, and the users moved by the instruction. It has not been obtained from actual daily life.

As researches which aim at activity recognition in daily life, the following work are presented. Uiterwaal et. el.[9] used two sensors at waist and one sensor at thigh, and measured movements and postures in working environments. Kern et. el.[3] used 36 sensors at each joint, and measured movements, typing, chair, handshake and writing on a blackboard. However, these two researches used only one user. Therefore, more evaluation on the user generality are required. Uiterwaal et. el.[9] used accelerometer at wrist, and recorded of the activity of six types of activies on ten users on daily life. The result showed it is possible to distinguish activities.

In contrast to these work, our work collects not the small scale and artificial data, but large scale activity data of real life. Moreover, we assume to use only one sensor for one user for usability and feasibility.

Bao and Intille[8] discusses how to learn activity recognition from annotation data of users. They explained the procedures and examples of each activities in advance to users, and eliminated the variance of annotation. Upon which, they collected the data of 5 sensors on the body, and obtained 84% of accuracy. They also addressed 2 sensors on upper and lower body each keeps accuracy well as 5 sensors.

Our work also uses parts of the same feature vectors as Bao and Intille[8], but our work use single sensor, and we focus on the system to enable gathering activity data.

Berchtold et. el.[12] propose an activity recognition service with mobile phones and achieve 97% accuracy at best for 20 subjects. While our system is similar to this work, our system focuses on gathering open data for activities with accurate labels with low stress. Moreover, our work shows the result of gathering massive data.

Kawaguchi et. el.[11] proposed a promotion to gather open activity data from multiple laboratories, and has 6,700 accelerometer data from 540 subjects in total. While their work is not a system proposal, our approach is to provide a platform system to gather activity data anytime and anywhere.

3. ALKAN System

To collect activity data efficiently, we developed a large-scale activity gathering system named ALKAN. In this section, we describe the requirement analysis and the system design.

3.1 Requirements

For collecting large scale activity data, the following requirements are addressed.

(1) (Accuracy) Labels, such as activity classes and the position of a device on the body, are accurately added.
(2) (Usability) Users can record activity data anytime and anywhere, with minimum stress.
(3) (Motivation) Users need motivation to promote recording activity data.
(4) (Flexibility) Labels must be extended if a new activity class or new position is discovered. Also, the utilization or application of the activity classes might be changed accordingly.
(5) (Scalability) The system can record the data from multiple users.

To address requirement (1), the most accurate way of labeling is *managed labeling:* to predefine labels, and designate users to repeat activities under the lablels for designated times in managed environment such as in a laboratory. However, this is too costly to collect large scale data over thousands of samples from over hundreds of people. It is not only costly for experimenter, but also costly for participant (users) to be forced to perform designated activities. Moreover, it is almost impossible to cover any kinds of activity classes in a laboratory situations, such as getting on a train.

Another option of labeling is *unmanaged labeling:* to ask users to label their activity of their real life log, once per certain period of hours/days. Although this is easier, since users can record their activities in instant ways in their life, and the experimenter can obtain 'real' activities without preparing artificial environments, the labels can be dirty in a way that they are mixed with others, polysemy, insufficient, omitted, and/or mistimed. Compromised way of managed/unmanaged labeling is required.

To compromise them, we introduce the idea of a "mission". A *mission* is a sequence of choosing an activity class, choosing the position on the body, and performimg the activity. Using this method, users can record activities anytime they want, and the label is accurately stamped within a few second. This method is not suitable for the sequence multiple activities, it can be effective for single activity recordings as first-level activity recognition.

For requirement (2), users must be able to record activities without network connectivity. In spite we live in the era of pervasive network access, there are still a lot of environments without network connection, such as in subway. Not only that, but also low latency in usage is important for usability. If a user has to wait for the response from the server before/after activity, it will be an obstacle for her/him to record. Activity recording should be independent of network connection and latency. The data collected can be uploaded to the server at a spare time after a while between recordings.

For usability, we adopted smart phones as mobile sensor devices. Standard smart phones are equipped with 3-axis accelerometers, non-volatile storage, and wireless communication. By this, an activity can be performed anytime, and stored to the storage. The data can then be uploaded to the server when it is connected to the network.

For requirement (3), since users are human, ideas not to bore them are needed. A simple way is to feedback the their own activity record by email. It will more interactive if the feedback is done within the sensor device. Feedback can be in variety of ways, such as the summary of their own log, statistic data of whole data, and ranking of each/whole activity classes. Furthermore, domain specific applications can be provided upon recorded data, such as calorie consumption estimation, training logs, practicing sports or dances. If these applications are provided, they also motivate users to record activity data.

In ALKAN, we prepared several feedback feedback services of 1) ranking of activity execution, and 2) calendar of activity history, at the first stage. Although they can be extended as mentioned below, these can be thought to help users to motivate.

To address requirement (4), we provide feedback services that are dynamically updated to users through web browser interfaces on mobile devices.

For requirement (5), smart phone client software is easy to scale up by installing client software through application deploying services.On the other hand, the server can be scaled up by existing distributed web technology.

3.2 System Architecture

The ALKAN system consists of A) mobile device clients and B) a server which gathers activity data. A user records missions useing the mobile device client. The information is uploaded to the server when it is online and accumulated in the server database. The user can view statistical information of the uploaded data, such as a calendar of activity history and rankings, by connecting to the web server through the mobile device or another web browser on a PC.

3.2.1 Client

We developed the client software both for iOS and Android OS. In this paper, we show the views on iOS, which runs on iPhones or iPodTouches by Apple, inc. in Fig .1 and Fig 2.

The client software has the following functionalities:

(1) Mission execution

(2) View and send mission history

(a) (b)

Fig. 1 Mission views in ALKAN: (a) select activity class, (b) select device position and start sensing.

(3) View statical information of the server

In (1), users first select an activity class as in Fig. 1(a)and a position as in Fig. 1(b).Then they start the activity and finish. The sensor can record GPS information and three axis accelerometer data at 20Hz.

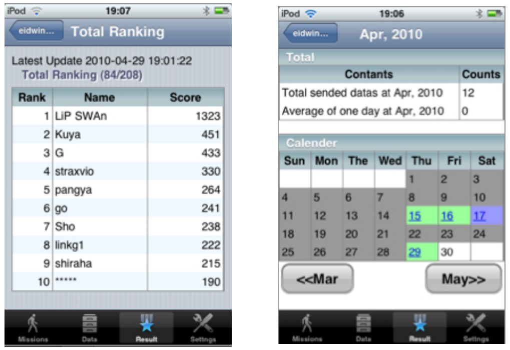
In (2), users can view the recorded mission history and add a comments to each mission as an annotation. Users can also delete missions if s/he does not wish to upload to the server. The mission data can be sent to the server as activity data either by each mission or by all at once. After the mission data are sent, they are removed from the history.

In (3), the software show web browser to access to the server, and show show statistical information such as ranking (Fig. 2(a)) and calendar history (Fig. 2(b)). This architecture of web browser interface is suitable not only when we update the statistic information, but also when we serve a new information or even when we add a service to specific a user group.

3.2.2 Server

The server gathers the activity data sent from clients, stores it to the database, and calculates and serves statistical information as a web server.

An example of current statistical information displayed is the rankings of the number of executed missions among users. The rankings are divided into the



(a) (b)

Fig. 2 Statistical information Viewed in a web browser in ALKAN: (a) ranking of the number of activities, and (b) calendar of activity history.

total ranking and those for each activity classes. By the rankings, users are expected to be motivated to perform missions. Moreover, the total ranking can be weighted by activity classes, since the activities of smaller amount should be motivated to be performed.

Other statistical information is the history of executed missions for each user. Users can view the start/end date/time, activity class, positions, and GPS information linked from a calendar format. This information is similar to lifestyle-related services, such as managing lifestyle-related diseases, in which users record their own lifestyles. Moreover, it is able to get equipped with automatic lifestyle recording when an activity recognition algorithm is implemented in the near future.

3.2.3 Data Structure

The communication between a client and the server is done over HTTP. Upon connection, the client is authenticated by a user account, and XML-formatted data or CSV-formatted data are passed between the client the server.

The passed data consist of 3 types: 1) Mission information, 2) Position information, and 3) Activity data set.

1) Mission list is a list of activity classes. 2) Position list is a list of device positions on the body. 1) and 2) are represented in XML format, and are provided by the server to each client. This makes updating the candidates of activity classes and positions dynamically in operation. To enable it, 1) and 2) also has the version information, to be able to intermingle multiple versions into the field.

Finally, 3) Activity data set is a set of mission executions, sent from each client to the server. When we call *an activity data*, we mean that it corresponds to a single mission execution, and *a activity data set*, it corresponds to one or more mission executions. An activity data includes the device product information, the user ID, the activity class ID, the position, and the sensor data with time stamps. They are mainly represented in XML format for extensibility, but the sensor data is in CSV format for efficiency of data size.

The activity class ID is one of the 1) Mission list. Moreover, the position can be either represented by position ID listed in 2) Position list, or by XYZ coordinates in the body image in Fig. 1 (b).

Sensor data is in CSV format and currently contains the data from the three axis accelerometer and GPS coordinates, but it can be easily extended by adding columns.

4. Collected Activity Data

ALKAN offers an opportunity to obtain large-scale activity data, which contributes to researches of not only activity recognition, but also context awareness, and variety of social sciences. As far as we know, such large scale of open activity data does not exist other than our work. By overviewing and analyzing the how the data gathered are, we can expect the knowledge for realizing such large-scale activity data gathering. In this section, we investigate the property of gathered activity data.

Since December 3, 2009, we have delivered 216 iPod touches as mobile sensors to university students and staff.So far, 216 devices were delivered by the January 21st, 2011.

We asked users a favor to collect activity data once a day on average. The activity data recorded by a device can be uploaded at any time when it is online to the internet. When there is no internet connection, the data is accumulated to the mobile device, and uploaded when it is connected.

4.1 Setting

The *activity classes*, which are the types of activities, are based on "Exercise Guide 2006 by the Ministry of Health Labour and Welfare, Japan, which gives a guideline to calculate *mets*: the strength of each activity. For this reason, several activities are divided to several strength, such as “walk slow”, “walk”, and “walk fast”. Additionally, to enrich the activity classes for transportations, some vehicle activities such as “train sit (sit in a train) were added. Furthermore, we added classes for recreational events, such as "bbq" and "darts", were added.

Finally, we adopted 46 activity classes and 19 device positions.

4.2 Activities

We acquired activity data continuously, and reached 35,310 activity data at the beginning of February 2011.

Fig. 3 shows the number of missions for each activity type.

From the figure, we can see that simple daily life activities were the sources of most collected data.

4.3 Positions

Fig. 4 shows the number of missions for each position. For simplicity, we only used ones which is listed in the position information, and ones represented in XYZ coordinates on the body are classified as “no labels” as well as ones with no position information. Together with no labels, “pants pocket” and “jacket pockets” are majority. Unsurprisingly, positions such as “ankle”, “arm”, and “belt” are hardly collected. “neck.strap”data are also rarely collected, but may be more frequently collected in office situations. The reason why “backpack” data are not collected often is likely because putting a device in and our of a rucksack is cumbersome, though it is more popular among students.

4.4 Clustering

To know the differences of durations among activity classes or device positions, we applied hierarchical cluster analysis as for mean durations to the groups by activity classes or device positions. In the analysis, Ward’s minimum variance method with euclidean distance was used.

Fig. 5 is the dendrogram of the clustering result for the groups by activity classes as for mean durations, after dropping the groups with under 100 missions. In the dendrogram, a branch or a leaf means a cluster, and the sum of the height of any paths denotes the distance between the clusters of the end of the path.

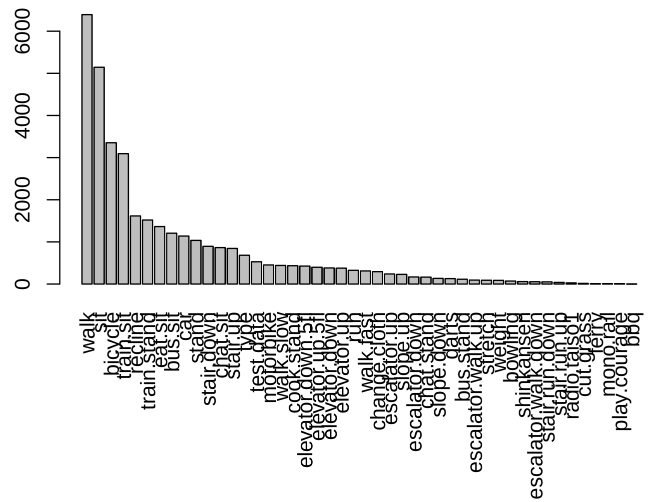


Fig. 3 Histogram of collected activity classes

From the figure, we can see that we can first divide the activity classes with a non-negligible distance into 2 clusters. From this, we can infer that there are both classes of longer duration and shorter duration.

4.5 Lessons learned

From the long-running and large-scale data collection, we can conclude the following:

(1) The ALKAN system is well designed and developed. Since the system is stable and the data has been collected constantly, we can infer that the requirements for usability, motivation, and scalability addressed in Sec.3.1 are satisfied as far as we have operated.

(2) The number of data collected is unbalanced in both activity classes and device positions. When we use the data for training the activity recognition

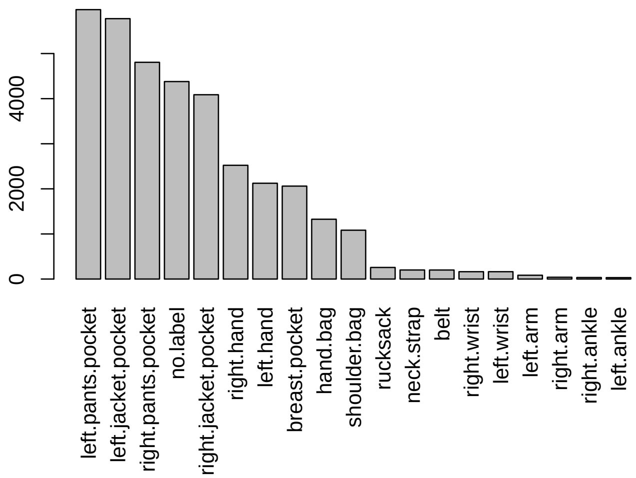


Fig. 4 Histogram of collected device positions on the body

model, there is no problem if the imbalance is caused by the probability distribution in natural life, but since this might also activity by the ease of recording data for each activity using ALKAN, we have to balance them in some way.

(3) The durations have distributions with specific variance, but it can be used for clustering both activity classes and device positions. This means this knowledge can be used for improving recognition techniques for sequences of activities. Also, device position recognition in sequences can also utilize the technique.

As for (2), we propose and use a method to balance the number of activity classes and device positions in the following section.

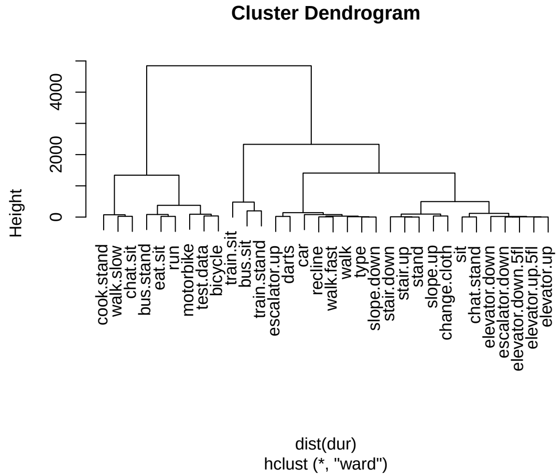


Fig. 5 Dendrogram of hierarchical clusters of activity classes by durations

5. Activity Recognition

In this section, we demonstrate activity recognition based on Bao and Intille[8]. However, the goal of this section is not to improve the accuracy of activity recognition. As shown in Bao and Intille[8], the accuracy by single mobile sensor is not well compared with multiple sensors. Moreover, better recognition are proposed in the literature. Instead, the goal of this section is to show the nature of gathered data by ALKAN, using the most standard activity recognition method.

5.1 Sampling

As addressed in Sec. 4.5, the number of data collected has large biases. Therefore, we need a way to sample activities uniformly from the gathered data. Here, we present an algorithm to pick samples from the activity data sets according to the following policy:

(1) (uniform positions) The sampled set has the same number of positions.

(2) (uniform activity classes)The sampled set has the same set of activity classes among positions.

(3) (uniform users) The sampled set has the same number of users for each

pair of position and activity, while the entities of users might be different for each pair.

(4) (uniform samples) The sampled set has the same number of samples for each combination of position and activity and user.

Algorithm 1 is the algorithm to achieve the policy. In the algorithm, we assume the Pos as the position set, Act as the set of activity classes, $User$ as the set of users, and $Sensor$ as the set of sensor data. D denotes the set of all the data, and we use the notation $Pos(D)$ $(Act(D), User(D))$ to denote the position (activity class, users, respectively) factors of D.

Moreover, we use the notation of D_{apu} to specify the subset of D with activity class $a \in Act$, position $p \in Pos$, and user $u \in User$. We also introduce wild card notation such as D_{*p*}, which means a subset of D with any activity class and any user with position p.

In the algorithm, line 1–3 are for 4) uniform samples, by omitting the combinations of (position, activity, user) with few samples and to ensure non-less than n samples for any combination. In line 11-14, n samples are sampled for each combination.

Line 4-6 are for 1) uniform users, by omitting activities with few users, for any position, and to ensure non-less than n_u users for any activity. In line 7-10, n_u users are sampled for each pair of position and activity.

We used the statistical processing software R[13] for data processing and machine learning shown in the rest of the paper.

Using the algorithm, we applied sampling upon gathered data, and investigated how many activity classes are sampled for several samples n, users n_u for single position. Fig. 6 is that of one of the positions, where $Pos = \{$`"left.pants.pocket"`$\}$.

From Fig. 6, the number of activity classes starts from 39 with 1 users, but decreases to 12 activity classes with 20 users, 6 with 60 users, and 2 with 80 users, when the number of samples is 1. If we take more samples, the values become lower, such as 8 activity classes for 20 users and 2 samples. We omit the results for other positions, but the curves are similar for other positions.

Thus, the number of activity classes and number of data is rapidly reduced when using the sampling algorithm. However, since the number of activity

Algorithm 1 Data Item Sampling

Input: $D = Pos \times Act \times User \times Sensor$, the number of users for each activity of position: n_u, and the number of samples for each user n.

Output: $D' \subseteq D$ which satisfies the policy above.

// REMOVE DATA ITEMS WITH FEW SAMPLES
1: **for all** $(p, a, u) \in Pos(D) \times Act(D) \times User(D)$ **do**
2: $D \leftarrow D - D_{pau}$ **if** $||D_{apu}|| \leq n$
3: **end for**
// REMOVE ACTIVITIES WITH FEW USERS
4: **for all** $(p, a) \in Pos(D) \times Act(D)$ **do**
5: $D \leftarrow D - D_{*a*}$ **if** $||User(D_{pa*})|| \leq n_u$
6: **end for**
// USER SAMPLING
7: $D' \leftarrow \emptyset$
8: **for all** $(p, a) \in Pos(D) \times Act(D)$ **do**
9: randomly sample n_u users from $User(D_{pa*})$, and add the corresponding data to D',
10: **end for**
// DATA SAMPLING
11: $Result \leftarrow \emptyset$
12: **for all** $(p, a, u) \in Pos(D') \times Act(D') \times User(D')$ **do**
13: randomly sample n items from D'_{pau}, and add them to $Result$
14: **end for**
15: **return** $Result$

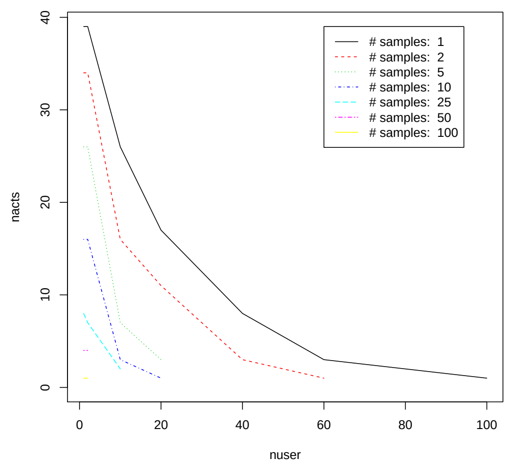


Fig. 6 Changes of # of activity classes for position: “left pants pocket”

classes, users, samples for each users and activity classes are balanced, we adopt the algorithm in the rest of the paper.

5.2 Feature Extraction

We extracted feature vectors from the 3-axis accelerometer data. At first, we removed 10 seconds of the beginning and ending, since it will include the action of touching, attaching, or operating the mobile sensors. By that, we omitted data under 20 seconds.

After that, we applied the sampling algorithm shown above, and calculated feature vectors based on Bao and Intille[8] for sampled activity sensor data.

For a sensor data item, time windows of 5-second durations are extracted at first, shifting 2.5 seconds for each extraction. For the time window, we calculated:

- Mean value of each axis.
- Frequency-domain energy: the sum of the absolute FFT values divided by the number of the FFT vlaues for each axis.
- Frequency-domain entropy: the entropy of the absolute FFT values minus the mean of the FFT values for each axis.
- Correlation among axes: the correlations between x-y, y-z, and z-x values. In Bao and Intille[8], they also used the correlations among multiple sensors on the body, but we did only used those inside single sensor device, since our setting is to use single sensor.

Thus, 12 dimensions are used for each feature vector.

5.3 Training

From the calculated feature vectors, we applied machine learning to generate a model for activity recognition. As machine learning algorithms, we used 1) a recursive partitioning tree, 2) a Naive Bayes classifier, 3) nearest neighbor classification, and 4) a support vector machine with a radial basis function kernel for the feature vectors.

To evaluate accuracy of recognition, we applied a special case of cross validation. Usual n-fold cross validation is to divide the data (feature vectors) to n groups, use $n-1$ groups for training and the rest 1 for test, and repeat it for n groups. Beside the traditional cross validation, we first picked up the subset for each activity, divided the users into n groups, and then divided the feature vectors according to the user groups. This is to maintain the balance of the numbers of each activity and user. The result is that users for training and testing are different.

Note that, since each activity data item has variety of duration, and since we extract multiple time windows from it, the number for each activities for machine learning will not be uniform, but it can be assumed to be natural, since the duration was decided by users of ALKAN.

We adopted the position of “left.pants.pocket”, picked 40 users, and sampled 1 activity data for each pair of activity and user. By this sampling, we obtained 8 activity classes. For the sampled data, we applied feature extraction, trainings, and 3-fold cross validations by each algorithm. Tab. 1 is the F-measures for each activity class and algorithm.

From the table, we get worse accuracy shown in Bao and Intille[8], although we used the same feature vectors. Aside from using single mobile sensor, the following are considered as the reasons:

- A mobile sensor is not firmly fixed to the body, but shaken in the pocket.
- The number of users is as large as 40 users. It may make the feature vectors

Table 2 Confusion matrix of activity recognition

→ trained ↓predicted	eat.sit	bicycle	car	sit	stand	train.sit	train.stand	walk
eat.sit	**1224**	170	719	1703	177	1742	165	184
bicycle	58	**1764**	1054	41	161	178	182	462
car	146	965	**3371**	918	44	1028	98	1041
sit	1193	22	538	**4156**	160	1685	139	65
stand	255	293	15	4210	**392**	1321	701	379
train.sit	803	61	1132	2625	228	**1751**	934	20
train.stand	780	172	403	341	988	272	**2684**	1072
walk	388	470	252	1278	514	50	431	**3732**

Table 1 F-measures(%) for each activity recognition algorithm

Activity Class	Rpart	NB	1-NN	SVM
eat.sit	10.74	5.19	17.58	22.40
bicycle	40.91	46.61	36.64	45.10
car	37.69	16.82	25.38	44.65
sit	33.77	55.09	30.42	35.76
stand	2.64	1.71	11.11	7.68
train.sit	36.77	8.79	23.83	22.48
train.stand	48.89	62.67	38.12	44.56
walk	54.16	50.95	42.03	53.09

non-general.

- Activity classes are similar to each other. Table. 2 is the confusion matrix of the recognition. As we can imagine, similar activity pairs such as "eat.sit"–"sit" and "sit"–"train.sit" are often mis-recognized.
- Actual activities may have varieties. Since users have performed activities in their own situations, environments could differ greatly on each trial.
- Labels are ambiguously understood by users. Since we only showed the names of labels, users may have understand each activity in varieties of ways.

Although these factors will decrease the recognition accuracy, they can produce a more challenging data set for activity recognition since these situations are more realistic than laboratory settings, which most of the existing work studied.

In this sense, these result of worse accuracy implies the data set gathered by ALKAN is a 'good' data. Especially, it will be valuable as a good testbed when we find more sophisticated feature vectors for better accuracies in the near future.

6. Conclusion

In this paper, we developed an activity data gathering system using mobile sensor devices. We also described the gathered data, introduced the sampling algorithm from unbalanced data, and presented activity recognition results to show the nature of obtained data.

ALKAN data are open and free to use. The users of ALKAN have already agreed with opening the data to public. Open data is necessary since several techniques have been proposed for activity recognition. Methodologies must be evaluated using the same data set. ALKAN could be the platform for evaluating existing or future activity recognition methodologies.

References

1) G.S. Chambers, S. Venkatesh, G.A.W. West, and H.H. Bui. Hierarchical recognition of intentional human gestures for sports video annotation. In Proceedings of the 16th International Conference on Pattern Recognition, volume 2, pages 1082–1085. IEEE Press, 2002.

2) R. Herren, A. Sparti, K. Aminian, and Y. Schutz. The prediction of speed and incline in outdoor running in humans using accelerometry. Medicine & Science in Sports & Exercise, 31(7):1053–9, 1999.

3) N. Kern, B. Schiele, and A. Schmidt. Multi-sensor activity context detection for wearable computing. In European Symposium on Ambient Intelligence (EUSAI). 2003.

4) S.W. Lee and K. Mase. Activity and location recognition using wearable sensors. IEEE Pervasive Computing, 1(3):24–32, 2002.

5) J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with

multiple acceleration sensors. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pages 747–52. IEEE Press, 2001.

6) K. Van Laerhoven and O. Cakmakci. What shall we teach our pants? In The Fourth International Symposium on Wearable Computers, pages 77–83. IEEE Press, 2000.

7) Ling Bao and Stephen S. Intille, Activity Recognition from User-Annotated Acceleration Data, International Conference on Pervasive 2004, pp. 1–17, Springer-Verlag GmbH, 2004.

8) M. Uiterwaal, E.B. Glerum, H.J. Busser, and R.C. van Lummel. Ambulatory monitoring of physical activity in working situations, a validation study. Journal of Medical Engineering & Technology., 22(4):168–72, 1998.

9) Ling Bao and Stephen S. Intille, Activity Recognition from User- Annotated Acceleration Data, International Conference on Pervasive 2004, pp. 1–17, Springer-Verlag GmbH, 2004.

10) N. Kawaguchi, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Murao, S. Inoue, Y. Kawahara, Y. Sumi, N. Nishio, "HASC Challenge: Gathering Large Scale Human Activity Corpus for the Real-World Activity Understandings", Proc. Augmented Human Conference, (to appear), March, 2011.

11) M. Berchtold, M, M. Budde, D. Gordon, H.R. Schmidtke, M. Beigl, "ActiServ: Activity Recognition Service for Mobile Phones", Proc. Int'l Symp. Wearable Computers(ISWC), pp. 1-8, 2010.

12) R Development Core Team (2010). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

13) ALKAN web site, http://alkan.jp

(Received ??? ??, ????)

(Accepted ??? ??, ????)

Yuichi Hattori was born in 1987. He is Ph.D student at Kyushu Institute of Technology, Japan. His reseach interests include human activity recognition with smart phones, healthcare application of web/pervasive/ubiquitous systems. He is a member of the Information Processing Society of Japan (IPSJ). Contact him at j350925y@tobata.isc.kyutech.ac.jp; 1-1 Sensui-cho, Tobata-ku, Kitakyushu, 804-8550, Japan.

Sozo Inoue Sozo INOUE is an associate professor in Kyushu Institute of Technology, Japan. His research interests include human activity recognition with smart phones, healthcare application of web/pervasive/ubiquitous systems. Inoue has a Ph.D of Engineering from Kyushu University. He is a member of the IEEE Computer Society, the ACM, the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE), and the Database Society of Japan (DBSJ). Contact him at sozo@acm.org; 1-1 Sensui-cho, Tobata-ku, Kitakyushu, 804-8550, Japan.