Behavior Detection Based on Touched Objects with Dynamic Threshold Determination Model

Hiroyuki Yamahara, Hideyuki Takada, and Hiromitsu Shimakawa

Ritsumeikan University,
1-1-1 Noji-Higashi, Kusatsu, 525-8577 Shiga, Japan
yama@de.is.ritsumei.ac.jp

Abstract. We are developing a context-aware application for use in homes, which detects high-level user behavior, such as “leaving the home” and “going to bed”, and provides services according to the behavior proactively. To detect user behavior, a behavioral pattern is created by extracting frequent characteristics from the user’s behavior logs acquired from sensors online, using an extraction threshold based on the criterion of frequency. Most context-aware applications need to determine such a threshold. A conventional model determines a fixed common threshold value for all users. However, the common value is improper for some users because proper values vary among users. This paper proposes a detection method of high-level behavior with a model for determining the threshold value dynamically according to individual behavioral pattern.

Keywords: Threshold, context, behavior, ambient, proactive.

1 Introduction

We aim to develop a context-aware system which provides services, such as in the following example. Imagine a situation where a user leaves the home. Usually, the user keeps windows open while the user is in the home, and the user closes the windows before leaving the home. One day, the user has left the home and has carelessly left the windows open. In order to prevent such a danger in advance, our system informs the user that the windows are open before the user leaves the home. Such a service is valuable for the user because the service not only improves user amenity but brings relief and safety. In the above example, the timing to provide a service to the user is important. If the user is informed after the user leaves the home, the user must go back into house for closing the windows. The user should be informed before going outside the house. As another example, suppose the system provides a service in a situation of coming home, and an attempted delivery notice had arrived into a home server when a user comes home. In such a case, the system recommends the user to go to pick up a package before the user sits on a sofa and gets relaxed. We refer to such services, which should be provided proactively according to user behavior, as *proactive services*. In order to provide proactive services, the system must correctly detect characteristic behavior of the user in situations of leaving the home and coming home.

G. Kortuem et al. (Eds.): EuroSSC 2007, LNCS 4793, pp. 142–158, 2007.

Some existing studies propose methods for detecting user motion, such as "walking" and "standing up", and simple actions, such as "making tea" and "brushing teeth" [1,2,3,4]. However, not these low-level behaviors but high-level behaviors, such as "leaving the home" and "coming home", need to be detected for providing proactive services. A high-level behavior is a complex behavior in which some actions are interleaved. It is difficult to provide proactive services only by detecting low-level behaviors. We are developing a system for detecting high-level behaviors [5].

Context-aware applications, including our developing system, are built based on a model that collects online sensor data, which is acquired according to user behavior, as behavior logs and matches the logs with behavioral patterns for recognition. First, such systems collect a specific amount of sample behavior logs, which show characteristics of user behavior. Next, a behavioral pattern is created with the logs in every situation to be detected. User behavior is detected by matching behavior logs, which are acquired online from current user behavior, with each behavioral pattern. These systems need a specific amount of personal behavior logs as sample behavior logs to create a behavioral pattern for recognition. Therefore, services of the systems do not get available until enough sample behavior logs have been collected. If it takes a long period to collect sample behavior logs from the user activity, the user is dissatisfied with waiting a long time. In order not to dissatisfy the user, a behavioral pattern must be created with a small number of sample behavior logs which can be collected in a short duration. Most of existing methods create a behavioral pattern based on a stochastic method such as Hidden Markov Model (HMM) [6,7]. These methods need many sample behavior logs to create a behavioral pattern. Consider the problem to create a behavioral pattern of the situation of leaving the home. Only about 30 sample behavior logs can be collected even in a month. That means these methods cannot create a behavioral pattern in a short duration. These methods are not adequate to be put into practical use. Compared with these existing methods, a system we developed previously detects user behavior, using a behavioral pattern created with only 5 sample behavior logs which can be collected within a week [5].

Our system must set threshold values, which are used for creating a behavioral pattern and for matching online sensor data with the pattern. The first threshold is an extraction threshold. A behavioral pattern is created by extracting characteristics which frequently occur in sample behavior logs. The extraction threshold is a threshold of the occurrence frequency. If an improper value is set to the extraction threshold, behavior recognition accuracy is low because the characteristics of the user are not extracted adequately. The second threshold is a detection threshold. When a user's online sensor data is matched with a behavioral pattern, if the degree of conformity is more than the detection threshold then our system detects user behavior and provides services. Naturally, an improper detection threshold makes behavior recognition accuracy low. Not only our system but also most context-aware applications require thresholds to be set for creating a behavioral pattern and for matching the pattern. To make behavior recognition accuracy high, proper threshold settings are necessary. After many sample behavior logs are collected, initial values of the thresholds can be changed into more proper values by learning with the logs. The issue of the learning is not discussed in this paper. This paper discusses, as an issue to be solved, how to set an initial threshold value that achieves high recognition accuracy under a constraint of a small number of sample behavior logs.

There are several approaches to set proper threshold values in a variety of fields. In image processing, a setting method of a threshold used for extracting a specific area from a target image has been proposed [8]. This method can be used only if both parts to be extracted and parts not to be extracted exist together in a recognition target. Our issue does not meet such a condition, because behavior recognition in this paper considers whether a current behavior log conforms to a behavioral pattern or not. This approach in image processing cannot be applied to our issue. In other approaches, Support Vector Machines and boosting has been used for text categorization [9,10], and HMM is used for speech recognition [11]. These approaches can set a proper threshold value under the assumption that they can collect and analyze many samples of recognition target or many samples of others which have similar characteristics to samples of the recognition target instead. However, there is the constraint of a small number of sample behavior logs for creating a behavioral pattern in our issue. In addition, because characteristics of high-level behavior in homes are different among individual users, behavior logs of other people other than a user cannot be used for sample behavior logs. Although these methods can be used for learning a proper threshold value after many personal behavior logs have been collected, these methods cannot be used for setting a proper initial threshold value.

It is important to set a proper threshold value initially in order not to dissatisfy a user. In the conventional model for setting a threshold value, a developer of a context-aware application or an expert of the application domain sets the initial threshold value before introducing the system to a user's actual environment. Having the system used by some test users on a trial basis, the expert analyzes relativity between change of recognition accuracy and changes in a threshold value. The threshold value is determined such that the recognition rate averaged for all test users is the highest, with receiver operating characteristic curve, precision-recall curve, and so on. The value determined is used as an initial threshold value common to all users after introduction to actual user environment. However, it is difficult to achieve high recognition accuracy with the common threshold value for all users. Proper threshold values vary with individual behavioral pattern.

This paper aims to create a behavioral pattern which can bring out higher recognition accuracy by setting more proper threshold value than the conventional model, particularly for users whose behavior is not recognized well with the conventional model. Because it is difficult to determine the threshold value with only a small number of personal sample behavior logs, we consider to utilize data from test users as in the conventional model. However, unlike the conventional model, we cannot determine threshold values directly and also cannot create a behavioral pattern with many data from test users in advance, because characteristics of high-level behavior vary with individual user, as mentioned above. This paper proposes a method for determining an extraction threshold dynamically, based on a model which derives not a threshold value itself but a rule for determining the value by analyzing test user data. When acquiring knowledge by analyzing test user data, if the knowledge is not about an attribute which has high commonality among many users, then the knowledge is not meaningful. The conventional model determines the threshold value itself by analysis. The value obtained represents knowledge acquired without separating attributes, which have low commonality, from attributes which have high commonality. By analysis focused on attributes which have high commonality, more meaningful knowledge can be acquired. As such an attribute, the proposed method focuses on the number of characteristics composing a behavioral pattern. There is a famous number known as "the magical number seven, plus or minus two [12]" in cognitive science. This hypothesis proposes that the number of items of information which a human can instantaneously handle is about seven items. This is a common number for all people. This means that humans select about seven characteristic information items by screening a lot of information in order to instantaneously grasp the situation. From another point of view, the person can evaluate a situation properly by discarding excess information and selecting only information which is minimally necessary. Consider the number of characteristics composing a behavioral pattern. If there are too many numbers, the pattern will include excess elements which were not normally characteristics. If there are too few numbers, the pattern will miss useful elements as characteristics. This property of the number of characteristics is similar to the property of the number of items for human cognition. Considering such a property, this paper assumes that there is a universally ideal number of characteristics composing a behavioral pattern, which does not depend on individuals, as in the case of human cognition system. The proposed method derives a determination rule of an extraction threshold by analyzing test user data with a focus on the number of characteristics composing a behavioral pattern. A value of the extraction threshold is dynamically determined based on the rule when creating a behavioral pattern after introducing a context-aware application to the actual user environment. The proposed method has the following advantages.

- Focusing on an attribute which has high commonality, the method acquires meaningful knowledge from test user data, from which the conventional model cannot acquire meaningful knowledge, to detect high-level behaviors.
- The method dynamically determines a threshold value for individual behavioral patterns created with a small number of sample behavior logs, using a threshold determination rule derived from test user data.
- With a proper threshold for individual behavioral pattern, the method improves the recognition accuracy for users whose recognition accuracy is low with the common threshold value.

The result of an experiment shows that the proposed method improves behavior recognition accuracy, which is less than 80% with the conventional model, of some experimental subjects more than 10%.

The remaining part of this paper is as follows. Chapter 2 describes our behavior detection system. Chapter 3 explains a model for deriving a threshold determination rule and application of the model into our detection system. Chapter 4 shows evaluation by experiment. Finally, Chapter 5 concludes this paper.

2 Behavior Detection for Proactive Services

2.1 Detection of High-Level Behavior

We consider situations of leaving the home, coming home, getting up and going to bed, as situations in which proactive services can be provided effectively. For example, suppose when getting up, our system provides a reminder service, which reminds a user of one-day-schedule and of things to be completed by the time the user leaves the home. By providing this reminder service before the user starts preparing for leaving or for having a meal just after a series of actions when the user got up, the service can support the decision of next action of the user. When going to bed, our system provides services which brings relief and safety. For example, our system informs of that the windows are not closed. We consider proactive services are valuable services which can proactively prevent repentance and danger, which the user faces when the services are not provided.

Proactive services should not be provided mistakenly when "the user gets out of bed just for going to the toilet in the middle of sleep", or when "the user goes outside house just for picking up a newspaper". High-level behaviors, such as "leaving the home" and "going to bed", cannot be correctly detected only by recognizing simple actions as in the existing methods [3,4]. We consider that a high-level behavior is a long behavior of around ten minutes. Some actions are interleaved in the high-level behavior. In addition, characteristics of the high-level behavior vary with individual user. Therefore, a behavioral pattern for detecting the high-level behavior must be created with personal behavior logs of individual user. In order not to dissatisfy the user due to long waiting for collecting personal behavior logs, we consider services must get available within a week at the latest only with a small number of personal behavior logs.

2.2 Individual Habit in Touched Objects

In order to detect high-level behaviors, we must collect data which remarkably shows characteristics of individual user behavior as behavior logs. We focus on the aspect that most people often have habitual actions in a habitual order, for not making omission of things to do, in situations such as leaving the home and going to bed. Each user has his own characteristic behavior in such specific situations. For example, in a situation of leaving the home, there can be habitual actions such as going to the toilet and taking a wallet. That means the user habitually touches the same objects every time in the same situation. The kind

Fig. 1. Objects embedded by RFID tags

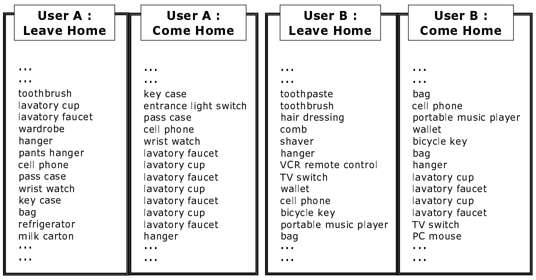

Fig. 2. Examples of behavior log

of objects the user touches and their order depend on the individual user. The number of objects a user touches in situations, such as leaving the home and going to bed, is more than the number of those in other situations. In situations such as watching TV, having a meal, and reading a book, the user touches few objects, or he touches only limited kind of objects. Compared to these situations, it is obvious the user touches more objects in situations such as leaving the home. Objects the user touches indicate his intention and his behavior remarkably. The logs of touched objects are adequate for use as individual sample behavior logs.

We record histories of touched objects as behavior logs, using 13.56MHz RFID tags. As shown in Fig. 1, the tags are embedded in various objects of a living space, such as a doorknob, a wallet, or a refrigerator. Every object can be identified by unique tag-IDs stored in the tags. In contrast, a user wears a finger-ring-type RFID reader. When the user touches objects, his RFID reader reads tag-IDs of tags embedded in objects. According to the user behavior, a time series of $< tag\text{-}ID,\ timestamp >$ is recorded in a database as the behavior log of the user. Fig. 2 shows actual behavior logs recorded by our system. The table shows behavior logs of two users in situations of leaving the home and coming home. For example, in the situation of leaving the home, the habitual actions of user A are different from those of user B. Looking at the log, it is inferred that user A brushed his teeth, changed his clothes, picked up some portable commodities, and brought out a milk carton from the refrigerator. It is inferred that user B brushed his teeth, set his hair, operated a VCR and then picked up some portable commodities. These behavior logs show that kind of touched objects and their order are different among individual users even in a same situation. Similarly, comparing each user's situation of leaving the home to that of coming home, it is found that a user touches different kinds of objects or touches the same objects in a different order in different situations.

2.3 Behavior Detection with Ordered Pairs

In order to detect high-level behavior, we create a behavioral pattern represented by a set of *ordered pairs*, which show the order relation among touched objects, with histories of touched objects as sample behavior logs.

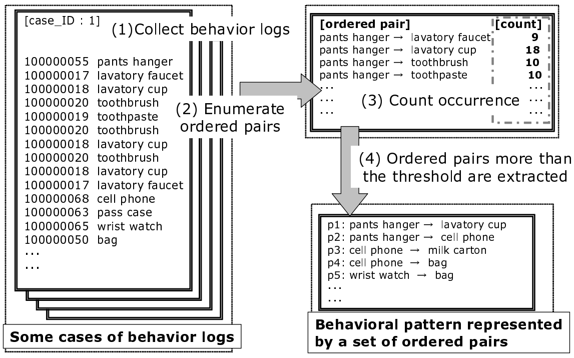

Fig. 3. How to create a behavioral pattern

The flow to create a behavioral pattern is shown in Fig. 3, with an example of a behavioral pattern in the situation of leaving the home. Generally, existing methods based on probabilistic models, such as HMM, create a behavioral pattern with high recognition accuracy using both behavior logs of the situation of leaving the home and logs of situations other than the situation of leaving the home as sample behavior logs. Consider our problem that a behavioral pattern must be created with a small number of sample behavior logs. Even behavior logs of leaving the home cannot be collected frequently. We can not expect to collect behavior logs of other situations which are adequate to make recognition accuracy high. Therefore, a behavioral pattern must be created only with behavior logs of leaving the home.

First, behavior logs of w cases are collected as sample behavior logs. The number of sample behavior logs w for creating a behavioral pattern is referred to as *window size*. The time length t_l of a sample behavior log is fixed. If m objects are sequentially touched in a behavior log l, then l is represented as a conjunction $\{o_1, o_2, \ldots, o_i, \ldots, o_m\}$, where, $o_{i-1} \neq o_i (1 < i \leq m)$. Second, all ordered pairs between two objects are enumerated from all collected sample behavior logs. If object o_j is touched after object o_i is touched, then the ordered pair p is represented as $\{o_i \rightarrow o_j\}$, which includes the case of $o_i = o_j$. For example, ordered pairs enumerated from a behavior log $\{o_1, o_2, o_3\}$ are $p_1 : \{o_1 \rightarrow o_2\} p_2 : \{o_1 \rightarrow o_3\} p_3 : \{o_2 \rightarrow o_3\}$. Next, the occurrence of all ordered pairs is counted up as occurrence count. The occurrence count means not the number of times that each ordered pair occurred in a sample behavior log, but the number of sample behavior logs including each ordered pair. For example, if an ordered pair occurs in all sample behavior logs, the occurrence count of the ordered pair is w. Finally, the ordered pairs where the ratio of the occurrence count to w is more than an extraction threshold $e\%$ are extracted as a behavioral pattern π.

The behavioral pattern π, which is created in advance, is matched with the current behavior log of time length t_l, which is acquired online from current user behavior. At the time when more than a detection threshold d % of ordered pairs, which compose the behavioral pattern π, exist in the behavior log, user behavior of leaving the home is detected.

For example, ordered pairs, such as $\{toothpaste \rightarrow toothbrush\}$, indicate the user's habitual actions, such as "brushing teeth". Ordered pairs, such as $\{toothpaste \rightarrow pants\ hanger\}$, indicates habitual order of the user actions, such as "the user wears pants after brushing his teeth". The behavioral pattern of a set of ordered pairs can represent the user's habitual actions and their order. Some existing studies create a behavioral pattern of a Baysian Network (BN) for recognizing simple actions [3,4]. To create a BN which has no cyclic path and can finish probabilistic reasoning in real-time, its network must be determined by hand. However, it is difficult to determine a network by hand so that the network can represent high-level behaviors in which some actions are intricately interleaved. Compared to the method using a BN, our detection method can extract characteristics of user behavior from such a complex behavior because our method uses an ordered pair, which is the smallest unit of order, to represent a behavioral pattern. There is an existing method which detects behavior in one situation with many behavioral patterns, which is created using time series association rule [13]. However, we need to observe user behavior for a long period in order to detect high-level behaviors. As a result, too many behavioral patterns are created by this existing method. Although this method needs to select only good behavioral patterns from many patterns, it is difficult to select good patterns under the constraint of a small number of sample behavior logs. Compared to this method, our detection method avoids comparing the quality among too many behavioral patterns, by potentially representing a variety of behavioral patterns with a set of ordered pairs.

2.4 Difficulty of Setting Threshold Values on Behavior Detection

We previously conducted an experiment in which we detected user behavior in situations of leaving the home, coming home, getting up, and going to bed, using our detection method. The recognition accuracy is evaluated both with *true-positive rate* (TPR) and with *true-negative rate* (TNR). TPR shows the rate at which behavior logs in a specific situation, which logs are referred to as *true cases*, are correctly detected with a behavioral pattern of the situation. TNR shows the rate at which behavior logs in situations other than the specific situation, which logs are referred to as $false\ cases$, are correctly neglected with the behavioral pattern of the situation. It is preferable that both TPR and TNR are high. As a result, the recognition rates of some subjects were more than 90%. Meanwhile, the recognition rates of a few users were low rates of less than 80%.

The main cause of these differences is that the extraction threshold and the detection threshold are pre-determined values common to all users. By calculating $half\ total\ true\ rate$ ($HTTR$), which is an average between TPR and TNR, these threshold values were determined such that HTTR averaged for all users is maximum. After many sample behavior logs are collected, the recognition accuracy can be improved by learning of a behavioral pattern with the logs. However, we should solve the problem that there are differences of recognition

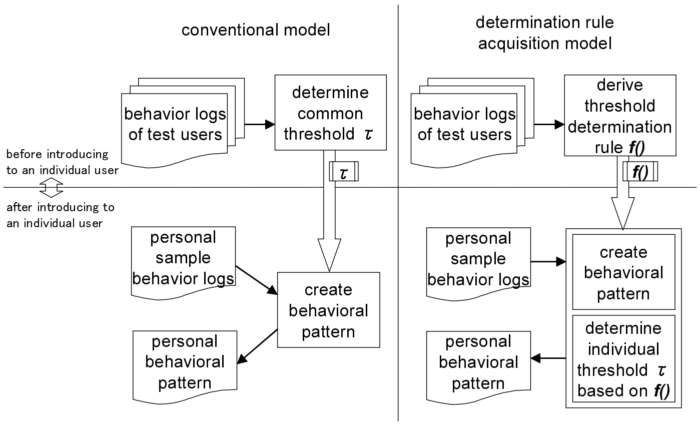

Fig. 4. Determination rule acquisition model from behavior logs of test users

rate among users depending on initial threshold values. It is necessary to improve the recognition accuracy of users, whose recognition rates are low with the common threshold values, by setting proper threshold values for individuals.

3 Dynamic Threshold Determination

3.1 Threshold Determination Rule Acquisition Model

We consider determining a threshold value dynamically for individual behavioral pattern in order to set a proper value to the threshold. For that purpose, unlike the conventional model which uses a fixed common threshold value, this paper proposes a model which acquires a rule to individually determine the threshold value for each behavioral pattern from the data of test users. The conventional model is illustrated on the left side of Fig. 4 and the threshold determination rule acquisition model, which we propose, is illustrated on the right side of Fig. 4. The horizontal center line shows a partition of the two phases for introducing a context-aware application to actual user environment. The upper portion is the development phase, before introducing the system to the actual environments of individual users. The lower side is the operation phase, after introducing the system. As shown in Fig. 4, the conventional model determines a common threshold value at the developement phase. First, the model collects behavior logs of test users. Next, for every test user, the model repeatedly creates a behavioral pattern with the logs, while matching the logs with the pattern. Analyzing the result of recognition accuracy, the model determines the threshold value with which recognition rate averaged for all test users is the highest. At the operation phase, the model creates an individual behavioral pattern with personal behavior logs. The threshold value is common irrespective of users. However, because a proper value for a threshold varies with the individual behavioral pattern of each user, behavior recognition accuracy of some users may be low with the common value.

To dynamically determine a proper threshold value for individuals, it is preferrable to acquire knowledge from personal behavior logs of individual user. However, it is difficult to determine a proper threshold value only with a small number of personal behavior logs. Therefore, the proposed model dynamically determines a threshold value by using both knowledge acquired by analysis of test user data and knowledge acquired from personal behavior logs. First, our model collects sample behavior logs of test users. Second, our model repeatedly creates a behavioral pattern with the logs and matches the logs with the pattern, for every test user. Next, our model analyzes the correlation between a threshold value and the recognition accuracy. If the threshold value is directly determined by analysis, the same problem occurs as in the conventional model. Our model derives not a threshold value itself but a rule f for determining the value by analysis. The threshold value is not determined at the developement phase. At the operation phase, the threshold value τ is determined for individual behavioral pattern by combining the rule f and knowledge acquired from a small number of personal behavior logs. If an analyst derives a rule f by focusing on an attribute, which does not depend on individual users, and personal behavior logs are used to consider attributes, which depend on individual users, then our model can determine a proper threshold value.

3.2 Effectivity of Dynamic Determination of Extraction Threshold

We apply the proposed model to our behavior detection system. The system has the extraction threshold and the detection threshold, which are described in Chapter 2.3. Primarily, it is important to set a proper value to the extraction threshold. In this paper, we consider a method for determining the value of extraction threshold dynamically.

The number of ordered pairs composing a behavioral pattern changes according to change of the extraction threshold, and affects the quality of the extracted behavioral patterns. It is preferable that a behavioral pattern includes many ordered pairs which are characteristics of user behavior in true cases. At the same time, the pattern should include few ordered pairs which can be characteristics of user behavior in false cases. If a behavioral pattern is composed of too few ordered pairs due to setting the extraction threshold high, then the behavioral pattern may not include some ordered pairs which should be normally included as user characteristics. On the other hand, if a behavioral pattern is composed of too many ordered pairs due to setting the extraction threshold low, then the behavioral pattern may include excessive ordered pairs which are not normally user characteristics. In particular, such fluctuation is a sensitive problem under the constraint of a small number of sample behavior logs. Suppose an improper value is set to the extraction threshold. It is impossible to extract ordered pairs adequately without excesses and shortages. Accordingly, recognition accuracy is low because differences between true cases and false cases are small when matching those cases with the behavioral pattern created with such ordered pairs. A proper extraction threshold sharpens differences between true cases and false cases. Consequently, recognition accuracy becomes high.

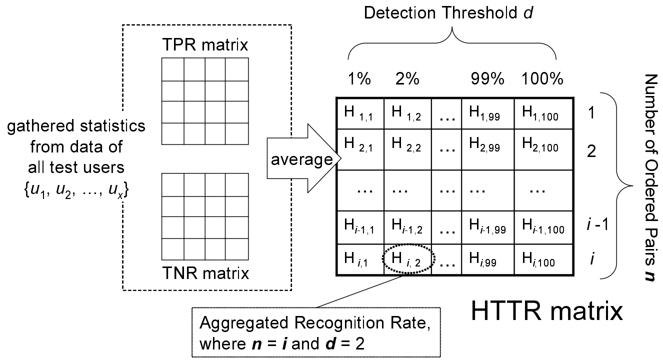

Fig. 5. TPR matrix and TNR matrix on statistics of test user data

3.3 Rating of Extraction Threshold by Statistics

Based on the threshold determination rule acquisition model, we derive a determination rule for setting the extraction threshold from data of test users. As mentioned above, the number of ordered pairs n affects the quality of behavioral patterns. The property of "the number of characteristics used for recognition", such as the number of ordered pairs, is similar to a cognitive property of human. "The magical number seven, plus or minus two [12]" in cognitive science proposes the hypothesis which indicates that humans select about seven characteristic information items by screening a lot of information in order to instantaneously grasp the situation. Consider the number of ordered pairs n. In both the case of excess ordered pairs and the case of insufficient ordered pairs, recognition accuracy is low. This property of the number of ordered pairs is similar to the property of the number of items for human cognition. Therefore, this paper assumes that there is a universally ideal number of ordered pairs, which does not depend on individuals, as in the human cognition system. In the issue of behavior detection, attributes such as kind of objects and their order have little commonality among users. It is difficult to derive a meaningful rule directly from these attributes. We attempt to derive a determination rule for the extraction threshold by evaluating the threshold value with a focus on the number, which has high commonality, of ordered pairs.

With an example of a behavioral pattern of a user v in the situation of leaving the home, we describe the proposed method which determines the threshold value dynamically. Before creating a behavioral pattern of user v, the threshold determination rule f is derived from behavior logs of x test users at the development phase. First, the proposed method executes the following procedure for every test user. The window size w is a given value which is common to all users.

1. Collect behavior logs in the situation of leaving the home as true cases, and also collect behavior logs in situations other than that as false cases.
2. Select w true cases as sample behavior logs.

3. Create w behavioral patterns with each setting of the extraction threshold value $e = 100 \times 1/w,\ 100 \times 2/w,\ ...,\ 100 \times w/w$, using the w true cases.
4. With all settings of the detection threshold d from 1% to 100%, match all true cases and all false cases with the w behavioral patterns.
5. Repeat step 2 to step 4 k times.

Second, TPR and TNR are calculated by gathering statistics on all results of the matching. As shown in Fig. 5, matrixes for the statistics of the rates are formed. The matrixes show the recognition rate with each number n of ordered pairs and each setting of the detection threshold. When a maximum number of ordered pairs is i in all created behavioral patterns, each matrix forms $i \times 100$ matrix. Finally, an HTTR matrix is formed. Each element H in the HTTR matrix is calculated by averaging each element in the TPR matrix and in the TNR matrix. Results of each row are respectively calculated with different numbers of statistical data. In the process of statistics, the method records the number of statistical data leading to results of each row of the HTTR matrix. Because there are w settings of the extraction threshold per behavioral pattern, the total number of statistical data is $w \times k \times x$. Each row of the HTTR matrix is rated with a rating score. The rating score s_i of the ith row is calculated as follows.

$$s_i = \ln(p(i)) \times \max_j(H_{i,j})$$

$\max_j(H_{i,j})$ means the maximum value in 100 elements of the ith row. $p(i)$ is the proportion of the number of statistical data used for the ith row to the total number of statistical data $w \times k \times x$. $\ln(p(i))$ is a coefficient for adding the reliability of statistics to the rating score. This method gives a higher rating score to rows using more statistical data. Next, these rows are equally divided into c clusters, such as cluster 1:{row 1, row 2, row 3}, cluster 2:{row 4, row 5, row 6}, ... The rating score of a cluster is calculated by averaging rating scores of all rows in the cluster. The value of c is empirically set to a proper number by an analyst. We assume that there is an ideal number of ordered pairs. However, because the number of ordered pairs composing a behavioral pattern depends on the number of ordered pairs occurring in sample behavior logs of individual user, one ideal number is not always identified using statistics of test user data. Therefore, this method attempts to find, not one ideal number, but "how much number is good roughly", by calculating rating scores of clusters. These rating scores correspond to the threshold determination rule. That is, when a behavioral pattern is created after introducing the behavior detection system to actual environment of user v, the extraction threshold is determined such that the behavioral pattern is composed of the number, which corresponds to as high rated cluster as possible, of ordered pairs.

4 Evaluation

4.1 Experiment

This paper describes an experiment to verify the efficacy of the proposed method. The experiment sets the time length t_l of a behavior log to 10 minutes. Before the experiment, we conducted a questionnaire survey for 2 weeks. In the questionnaire, subjects recorded the complete details about kind of objects the subjects touched and their order in 4 situations of leaving the home, coming home, getting up, and going to bed every day. With the questionnaire results, we could confirm that many people respectively touch different objects or touch objects in different orders, in different situations. After that, we experimentally embedded the RFID system described in Chapter 3 into the living space. RFID tags are embedded in many household goods such as kitchen gas stove, kitchen sink, and electric appliances, in every spaces such as living, kitchen, entrance, and so on. In such a space, we collected behavior logs of actual objects which subjects touched in the 4 situations respectively. The logs acquired online from subjects' behavior are stored in a database. We collected 70 behavior logs per subject.

To compare the proposed method, which dynamically determines the extraction threshold, with the method using the conventional model, which sets a fixed common value to the threshold, the experiment calculates TPR and TNR for behaviors of individual subjects in the 4 situations by repeatedly creating a behavioral pattern and matching behavior logs with the pattern, using behavior logs in the database. Here, true case means behavior logs of each situation, where a behavioral pattern is created, and false case means behavior logs of situations other than the situation of true case. The window size w is set to 5 in the experiment.

First, a threshold determination rule for the proposed method was derived by the calculations described in Chapter 3.3 with behavior logs of 8 subjects. In the experiment, rows in an HTTR matrix are divided into 100 clusters. Basically, each cluster includes three rows. But there are a few exceptions. Rows from the first row to the fifth row are included in a cluster which is rated as the second place from bottom, because they are empirically too small number as sample behavior logs. In addition, all of rows following the 300th row are included in the cluster same as the 300th row, whose cluster is rated as last place. Next, the following procedure was executed in order to calculate individual behavior recognition accuracy with 8 subjects. In this experiment, user behavior in each situation must be correctly detected in ten minutes, the time length t_l.

1. Select 5 sample behavior logs from true cases and create a behavioral pattern with the logs, based on the extraction threshold.
2. Select other 1 behavior log from true cases, and match the log with the behavioral pattern.
3. Match all behavior logs of false cases with the behavioral pattern, with all settings of the detection threshold d from 1% to 100%.
4. Repeat 100 times from step 1 to step 3, with a new behavioral pattern which is created by selecting new combination of 5 true cases every time.

Here, TPR is calculated based on cross validation. However, we limit the number of sample behavior logs used for creating a behavioral pattern to 5, which can be collected within a week. TNR is calculated by matching all false cases with all created behavioral patterns. The extraction threshold is determined when creating a behavioral pattern in step 1 using the threshold determination rule described

Table 1. Result of "Leaving the Home"

subject	TPR(%)	TNR(%)*
A	99	91.94
B	95	88.36
C (#1)	89 (+18)	92.84
D (#2)	94 (- 6)	98
E	99	99.68
F	100	95.04
G	99	96.6
H (#2)	88 (-10)	91.14

*TNR is rounded off in the 3rd decimal place

Table 2. Result of "Coming Home"

subject	TPR(%)	TNR(%)*
A	91	95.25
B	99	99.38
C (#1,#2)	90 (+14)	84.88 (-9.13)
D (#1)	98 (+13)	98.8
E	98	99.5
F	100	100
G	100	99.78
H	100	100

*TNR is rounded off in the 3rd decimal place

above. In this way, TPR, TNR and HTTR of all subjects are calculated for the case in which the extraction threshold is dynamically determined. After that, these rates in the case of using a fixed common value as the extraction threshold are calculated by similar steps. In that case, the extraction threshold is fixed to 80% in step 1 such that recognition accuracy is the highest. Although TPR, TNR and HTTR are calculated with all settings of the detection threshold from 1% to 100%, the results of the two methods are compared using TPR and TNR on a detection threshold with which HTTR of each method is the highest per subject.

A user touches less number and less kinds of objects, in situations other than the 4 situations to be detected in this experiment. Therefore the proposed method, which focus on kind of objects the user touches and the order of the objects, can distinguish among the 4 situations and other situations easily. Previously, we conducted an experiment in which we recognized behavior logs including behavior logs of situations other than the 4 situations. Only up to 7% of ordered pairs, which compose individual behavioral pattern, occurred in situations other than the 4 situations. This result showed that user behavior in situations other than the 4 situations has no chance to be mistakenly detected by the proposed method. With this result in mind, we evaluate the recognition accuracy only with the 4 situations in the experiment of this paper. This means we evaluate our behavior detection method under more difficult conditions.

4.2 Discussion

Based on the result of the t-test, the experiment results are evaluated with the idea that difference of more than 5% is a statistically-significant difference

Table 3. Result of "Getting Up"

subject	TPR(%)	TNR(%)*
A	96	96.2
B	84	82.48
(#2)	(- 6)	(-14.3)
C	75	96.23
(#1)	(+11)	(+12.52)
D	100	89.91
(#2)		(-9.98)
E	97	59.38
(#3)	(+31)	(-27.13)
F	96	91.45
(#2)		(-8.23)
G	100	99.98
H	59	93.6
(#3)	(-22)	(+30.22)

*TNR is rounded off
in the 3rd decimal place

Table 4. Result of "Going to Bed"

subject	TPR(%)	TNR(%)*
A	76	74.44
B	93	70.88
C	95	99.98
D	91	95.94
(#1)	(+15)	
E	47	85.68
(#1)	(+12)	
F	99	97.92
G	100	98.84
H	97	93.92
(#1)	(+15)	

*TNR is rounded off
in the 3rd decimal place

between the proposed method and the method using the conventional model. As a result of the experiment, recognition rates in the proposed method are shown from Table 1 to Table 4. The tables respectively show the results of leaving the home, coming home, getting up, and going to bed. Each table shows the TPR and the TNR by the proposed method. In addition, the difference between the proposed method and the method using the conventional model is shown in parenthesis under each value. If the value is a positive value, then the proposed method has increased the rate. The differences which are less than a statistically-significant difference are not shown.

Looking at TPR and TNR in the tables, notable results are grouped into 3 groups from #1 to #3. Group number is written under subject name in each table. In group #1, TPR or TNR have increased with the proposed method. In each situation, there is at least 1 subject whose TPR or TNR have increased with the proposed method. Particularly, subject C of Table 1, subject C of Table 3, subject D and E of Table 4 have significantly increased. Their rates have increased more than 10% with the proposed method from low rates which are less than 80%. In group #2, TPR or TNR have decreased with the proposed method. However, even after decreasing, the rates can keep more than 80% for all subjects in group #2. Considering that our detection method must be introduced into a variety of user environments, the detection method must achieve high recognition accuracy stably for behaviors of as many users as possible. The detection method should not be effective on only a portion of users. In the experiment, the proposed method has decreased the rates of some subjects whose recognition rates are very high with the method using the conventional model. This decrease is not ideal result. However, the proposed method has increased significantly the rates of some subjects whose recognition rates are low with the method using the conventional model. This result shows the proposed method can achieve stabler behavior detection than the method using the conventional model. Overall, the result of the experiment means the recognition accuracy can be improved by determining a better value of the extraction threshold with the proposed method. The result has proved the proposed method is effective. Exceptionally, the proposed method is not effective on subjects of group #3. About their TPR and TNR, one rate has increased and the other has decreased, based on just a basic relation of trade-off.

5 Conclusion

This paper proposed a detection system of high-level behavior, such as “leaving the home”, and proposed also a method for dynamically determining threshold values, which should be set in order to introduce the system to a variety of user environments. An experiment has proved our method is effective. Our method improved the recognition rate of subjects, whose rates were low with the common threshold value, more than 10%. The present recognition rate is not enough practical. In the future, we will attempt to achieve higher recognition rate by combining the present method with other informations such as position of users. In addition, we will evaluate our method by introducing more user environments.

References

1. Barbič, J., et al.: Segmenting motion capture data into distinct behaviors. In: Proc. Graphics Interface 2004, pp. 185–194 (2004)
2. Moore, D.J., et al.: Exploiting human actions and object context for recognition tasks. In: Proc. ICCV 1999, pp. 80–86 (1999)
3. Patterson, D.J., et al.: Fine-grained activity recognition by aggregating abstract object usage. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 44–51. Springer, Heidelberg (2005)
4. Wang, S., et al.: Common sense based joint training of human activity recognizers. In: Proc. IJCAI 2007, pp. 2237–2242 (2007)
5. Yamahara, H., et al.: An individual behavioral pattern to provide ubiquitous service in intelligent space. WSEAS Transactions on Systems 6(3), 562–569 (2007)
6. Aoki, S., et al.: Learning and recognizing behavioral patterns using position and posture of human body and its application to detection of irregular state. The Journal of IEICE J87-D-II(5), 1083–1093 (2004)
7. Kidd, C.D., et al.: The aware home: A living laboratory for ubiquitous computing research. In: Streitz, N.A., Hartkopf, V. (eds.) CoBuild 1999. LNCS, vol. 1670, pp. 191–198. Springer, Heidelberg (1999)
8. Kimura, Y., et al.: New threshold setting method for the extraction of facial areas and the recognition of facial expressions. In: Proc. CCECE 2006, pp. 1984–1987 (2006)

9. Shanahan, J.G., et al.: Boosting support vector machines for text classi cation through parameter-free threshold relaxation. In: Proc. CIKM 2003, pp. 247–254 (2003)
10. Cai, L., et al.: Text categorization by boosting automatically extracted concepts. In: Proc. SIGIR 2003, pp. 182–189 (2003)
11. Asami, T., et al.: A stream-weight and threshold estimation method using adaboost for multi-stream speaker veri cation. In: Proc. ICASSP 2006. vol. 5, pp. 1081–1084 (2006)
12. Miller, G.A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. The Psychological Review 63(3), 81–97 (1956)
13. Mori, T., et al.: Behavior prediction based on daily-life record database in distributed sensing space. In: Proc. IROS 2005, pp. 1833–1839 (2005)