Behavior Prediction from Trajectories in a House by Estimating Transition Model Using Stay Points

Taketoshi Mori, Shoji Tominaga, Hiroshi Noguchi, Masamichi Shimosaka, Rui Fukui and Tomomasa Sato

Abstract— **In this paper we propose a novel method for predicting resident's behaviors in a house from one's movement trajectories. The method consists of 1) segmentation of trajectory data into staying or moving and classification of the segments and 2) prediction by time-series association rules from transition events of each segment. The method predicts the start time of target behaviors for daily life support, such as eating, taking a bath etc. The time lag between the prediction and the target behavior can be set up manually, thus the method is adaptable to a variety of supporting systems. The experimental results using real residents' trajectory data of almost two years demonstrate that prediction of behaviors by the proposed method is feasible.**

I. Introduction

It is important to recognize persons' behaviors for executing supports by robots or systems. Of course, that contains what they are doing then. However, since it takes considerable time for execution or preparation of support, to predict what they are going to do is needed. And then, the high quality support is enabled by predicting the target behavior.

Though there are many kinds of sensors for the behavior recognition, in case of supporting residents at a living space, external sensors is considered to be more suitable than the wearable ones since they do not need batteries and they can flexibly cope with continuously changing clothes in a living space. There are researches of intelligent environments in living spaces by introducing sensor networks [1], [2], [3], [4], [5]. These systems have a lot of and many kinds of sensors, and the residents' behaviors can be recorded in detail in them. However, it is currently impractical to introduce such large-scale systems into the real existing houses.

From such a background, we constructed a system [6] that calculates trajectories of the resident by measuring one's using multiple Laser Rangefinders (LRF) and estimating one's position in time series. We have accumulated trajectories from the system for approximately two years.

Trajectories are used for predicting behaviors [7], [8], [9]. These researches are often focused on the prediction of where persons are going from the position where they are then. These approaches are considered to be useful for the collision avoiding systems of mobile robots or the simple information presentation. However, since the support in a living space are focused on assistance for or substitution for behaviors such as cooking or preparation for going out and they cost more time to prepare, the further prediction will be needed.

T. Mori and H. Noguchi are members of Department of Life Support Technology (Molten), The University of Tokyo, e-mail: {tmori, noguchi}@ics.t.u-tokyo.ac.jp

S. Tominaga, M. Shimosaka, R. Fukui and T. Sato are with the Department of Mechano-Informatics, The University of Tokyo, e-mail: {tominaga, simosaka, fukui, tsato}@ics.t.u-tokyo.ac.jp

As for the meaning or information of residents' trajectories, our daily life in a house contains a lot of behaviors and they are connected like a chain. For example, there may exist washing his face, fetching a newspaper and preparing for breakfast, from getting up to eating breakfast. And, each behavior are stratified and expressed by the chain of behaviors of a deeper layer, for example fetching a newspaper are separated into going to the entrance, catching it and bringing it back. Each of such stratified and chain-like behaviors has a relationship to the location in a house, and most of them are considered to contain staying at the related locations. Table I shows general pair examples of locations in a house and behaviors of residents. In fact, the relationship between locations and behaviors have the factors that depend on the resident and the pairs in the table are not necessarily true, however, the location where the resident stays will be a clue to behavior recognition of the person. On the other hand, when the resident is not staying it can be said that one is moving, and one is staying before and after one moves. Thus, grasping the flow of behaviors is enabled by segmenting trajectories into staying or moving and mining the transition of segments.

TABLE I

Pair Examples of Locations and Behaviors

Location in a House	Corresponding Behaviors
Dining Table	Eating, Reading Books
Kitchen	Cooking
Bed	Sleeping
Bathroom	Taking a Bath
Washstand	Washing Hands or Face
Entrance	Going out, Coming Home

Therefore, in this paper, we propose a method for predicting resident's behaviors by one's trajectories. The method consists of 1) segmentation of trajectory data into staying or moving and classification of segments and 2) prediction by time-series association rules from transition events of each segment.

The rest of this paper is organized as follows. We explain the system, which calculates and accumulates trajectories in a house, in section II. Next, we estimate transition model of behaviors by segmenting the trajectory and classifying each segment in section III. And then, we extract the features existing before the target behavior starts in section IV. In section V we show the experimental result using the real trajectories of two residents for twenty one months. Finally, conclusion is discussed in section VI.

II. TRACKING SYSTEMS IN A HOUSE

A. Layout of Sensors and the Experimental House

We utilize the tracking system introduced into a real house (Fig. 1) [6]. The system is constructed with multiple LRF modules (upper left part of Fig. 1), a combination of Hokuyo URG-LX04 and Atmark Techno Armadillo-220. Specifications of the module are in Table II. The modules are arranged at hip-height and in several locations in the house (concrete locations are shown in Fig. 1), and the locations are calibrated manually by the LRF's output data. Each module is connected to a server by wired LAN and the server integrates the sensor data.

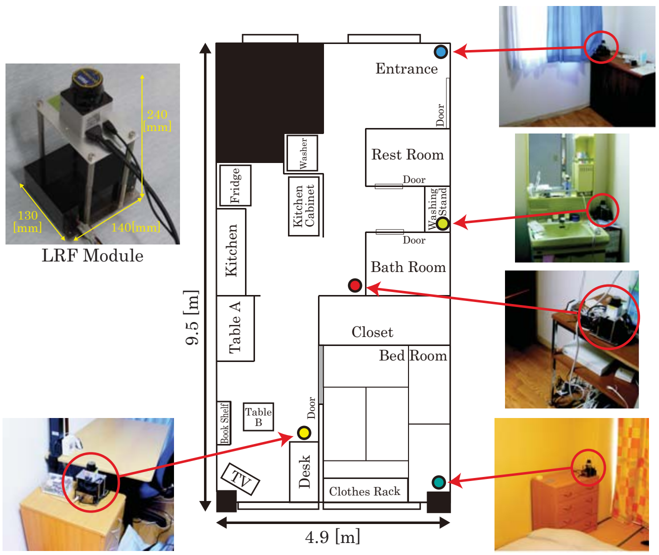


Fig. 1. Layout of Experimental House and Location of LRF Modules

TABLE II
SPECIFICATIONS OF LRF MODULE

Max Measurable Distance	5.6 [m]
Scan Range	240 [deg]
Scan Resolution	approx. 0.36 [deg]
Frequency	10 [Hz]

B. Calculation of Movement Trajectories

The method of calculating trajectories consists of preprocessing to get candidate points of the resident, detection of his location by detected points, and tracking with particle filter [10]. First, at the preprocessing step, we utilize the grid map of room layout such as Fig. 2-A. The black grids represent the location at which the residents are not to be such as walls, furniture, etc and at which there are objects at hip-height, which might be mistaken for the residents. All background-subtracted LRF data are projected on 2D coordinates and the points on black grid are removed. Next, at the detection step, we estimate the resident's location by fitting a circle to the preprocessed points with least-square method and detect the residents position as the center of the fitting circle. Last, at the tracking step, we consider the resident's 2D position $x_t = (x, y)$ as a state in the filter. Scattered particles on black grids or between foreground of scan points and LRF are removed. Supposing their movement to be uniform linear motion, the remaining particles are evaluated with distance between state (x, y) of particle and LRF points below the equation.

$$p(y_t|x_t) = \prod_{i=0}^{m} exp(\frac{-(d_i - R)^2}{\sigma^2})$$

Where m is the number of foreground points. σ is distributed variance, defined as 0.25 empirically. R is defined as 15 [cm] in our method.

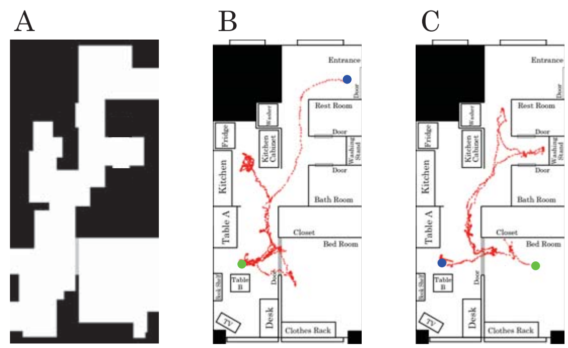


Fig. 2. Grid Map for Calculating Trajectories and Trajectory Examples (A: Grid Map B, C: Trajectory Examples)

C. Features of the Calculated Trajectories

Fig. 2-B,C are trajectory examples. In the figures, blue circles are the starting points of trajectories and the green circles are the end points. Fig. 2-B is the data from coming back to sitting down on the floor near the table B (122 seconds), and Fig. 2-C is the data from standing up near the table B to lying in bed (120 seconds). Starting or end points of trajectory data have the features below.

- The locations out of range of LRFs such as restroom, entrance, bathroom, etc.
- Near the furniture where residents lie down or sit down on the floor (as is usual with Japanese customs) below the installation height of LRFs such as table B, bed etc.

Thus, there are behaviors potentially, while the system does not track the resident.

III. ESTIMATING TRANSITION MODEL USING STAY POINTS

A. Extracting Typical Staying Locations

As we mentioned in section I, trajectories can be divided into staying and moving. In addition, each location of staying is related to the activity there. Thus, extracting typical staying locations is meaningful for understanding transition of activities. There, we call points of trajectories which considered to be staying, stay points.

Now, we explain the clustering method of stay points from the accumulated trajectories. Fig. 3 shows stay points around Aug. 2009 under the condition below a certain velocity (we defined the velocity as 0.2 [m/s]), the left part is the plot image and the other part is the two-dimensional histogram. The ellipses of the same color correspond to the same location each other. Because of the difference of the time at the locations, stay points are greatly biased and it is difficult to extract the locations considered to have a meaning with existing clustering method. Therefore, we cluster them with the method below,

1) Divides the house into meshes and count the number of stay points in each mesh
2) Binarizes the count of points with a threshold
3) Clusters the binarized mesh as a new dataset, with k-means method

The left part of Fig. 4 is the clustering result of the stay points. Thus, locations which can be seen in the left part of Fig. 3 but, because of the biased time being there, cannot be seen in the right part of the figure such as washer, entrance, etc are extracted. In the figure, red circles are the cluster centers. The right part of the figure is table of corresponding furniture of each cluster center. The cluster centers are related to furniture layout or floor plan, except position 4. The position 4 in the figure may be overlooked if these position are chosen manually by the layout, however, it may become important for detailed transition modeling of the behaviors.

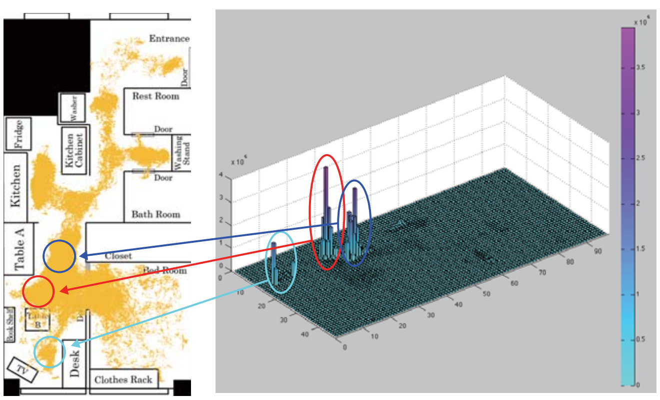
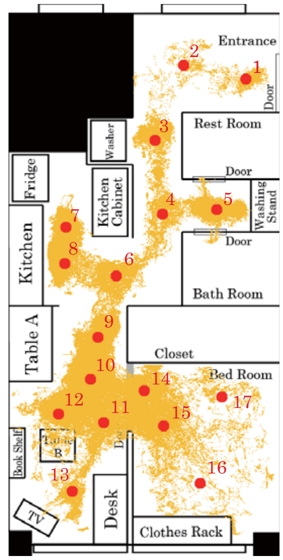


Fig. 3. Stay Points around Aug. 2009

B. Segmentation of Trajectory Data into Staying or Moving

We extract stay points from trajectories, and assign ID of the nearest cluster center. Then, we define a group of continuous points of the same ID as staying activity and a part between staying activities as movement. Each staying activity is assigned ID of the cluster center. Starting and end points of trajectories are processed as stay points. It is because activities should exist while the tracking system loses the resident such as lying in bed or taking a bath as we mentioned in section II-C.

Fig. 5-B shows an example of extracted segments from a trajectory (Fig. 5-A). The trajectory is the data from the



Pos.	Corresponding Furniture
1	Entrance
2	Window
3	Washer
4	
5	Washstand, Restroom, Bathroom
6	Cabinet
7	Kitchen, Fridge
8	Kichen
9	Table A
10	Table A, Door
11	Door
12	Table B
13	Desk
14	Door, Closet
15	Door
16	Clothes Rack
17	Bed

Fig. 4. A Clustering Result of Stay Points

bathroom to the entrance (269 seconds). Thus you can see that potential activities are extracted by the process.

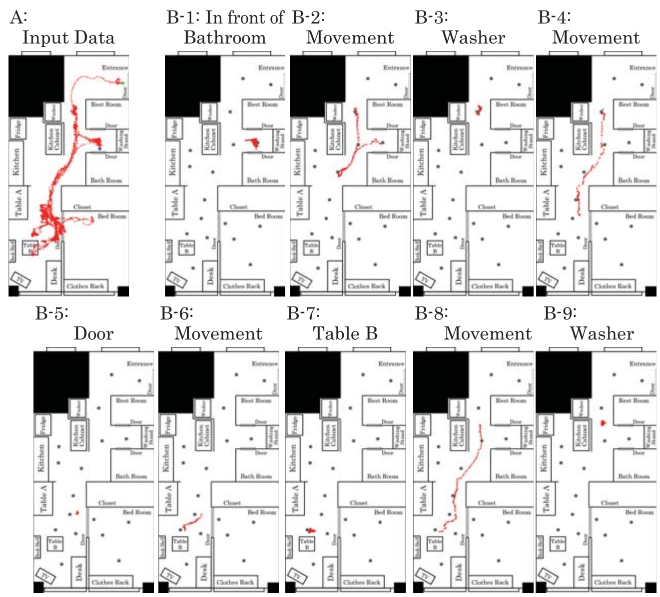


Fig. 5. The First Nine Extracted Segments from an Input Trajectory (A: Input Data B: The First Nine Extracted Segments)

IV. Prediction Algorithm with Segmented Trajectories

A. Prediction by Preceding Activities

Factors that cause resident's behaviors are considered to be,

- Elapsed time from the last occurrence of the same behavior
- Current time itself
- Preceding activities

We give examples of eating. When residents does not eat for six hours they will be hungry (the first factor), residents who usually eat at 12 possibly eat at 12 (the second factor), and there should exist a preparation of meal before eating (the third factor). Of course, each factor does not have an effect only as a stand-alone element but a combination of them. However, in case of providing real-time assistance to resident following the transitive behaviors, the third factor, preceding activities, should be the most reliable of the three. Thus in this section we construct a prediction algorithm by mining events of preceding activities.

B. Extracting Transition Event from Segmented Trajectories

We define the transition of segmented trajectories as events. Concretely the system extract events in this way below,

- Transition from a staying activity to movement is an OUT event assigned the ID of the staying activity.
- Transition from movement to a staying activity is an IN event assigned the ID of the staying activity.

Fig. 6 is an outline of transition event extraction from segmented trajectories. The system finds a combination or a sequence of events extracted this way.

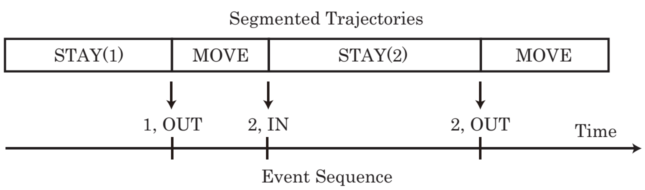


Fig. 6. Transition Event Extraction from Segmented Trajectories

C. Extracting Features for Behavior Prediction

For the following discussion, we define the terms about event mining, event sequence, window, episode, and minimal occurrence. Although there are various kinds of defined episodes [11], the system adopts the serial episode. The definition of the terms follows [12], [13].

Definition 1. Event Sequence and Window
An event sequence of E is an ordered sequence of events like $S = \langle (A_1, t_1), (A_2, t_2), ..., (A_n, t_n) \rangle$ ($A \in E$ is an event type and t is the occurrence time of the event). A window $W = (t_s, t_e)$ is a slice of an event sequence and contains those pairs (A, t) from S where $t_s \leq t \leq t_e$. The time span $t_e - t_s$ is called the width of W.

Definition 2. Episode
An episode is a shorter event sequence that occurs serially in a given longer event sequence, where some other events may occur within the range of the episode. For a window $W_i = W(t_s, t_e)$, it is said that the n-event episode α occurs in W_i if there exists a sequence of positive integers $\{\phi(i)\}$ such that $t_s \leq \phi(1) < \phi(2) < ... < \phi(n) \leq t_e$ and $\alpha(i) = S[\phi(i)]$ for any $i = 1, ..., n$.

Definition 3. Minimal Occurrence
A minimal occurrence of an episode α in an episode sequence S is the time interval (t_s, t_e) which satisfies the followings.

- α occurs in window $W = S(t_s, t_e)$
- α does not occur in any proper subwindow on W

In this research, we utilize and modify the Harms' time-series association rule [14]. The rule are described as Fig. 7. Concretely, it is like, "if A occurs in $Window$, then e (1-event Episode) occurs in $Prediction\ Time\ Range$ after $Time\ Lag$". Originally, e is a multiple-event Episode and $Time\ Lag$ is the time between the first event of A and e. However, if $Window$ is wider than $Time\ Lag$ in the original rule, a event of e itself or after e may be learned as contents of A. It is why we define $Time\ Lag$'s starting point as the last event of A. In addition, since predicted episodes are the start time of the behavior and contains only 1 event, we define the occurrence of e to have the duration like Fig. 7.

The confidence of the rule is the conditional probability of e, under the condition of occurrence of A i.e.

$$Confidence(Rule(A, e)) = p(e|A) = \frac{freq(Rule(A, e))}{freq(A)} \tag{1}$$

Where, $freq(X)$ is the frequency of X in the sequence. In this research, frequency of an episode are defined as the number of minimal occurrences.

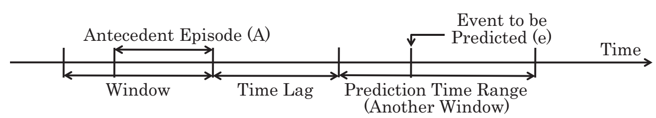


Fig. 7. Time-Series Association Rule

D. Efficient Learning of the Rules

Since it is a peculiar situation that the predicted event (the start time of the predicted behavior) are much less than the total events, there will be a lot of wasted time in case of learning with a simple method and it is difficult to utilize directly the existing method such as depth-first search algorithms [15], [16] or breadth-first search algorithms [17]. In this research, we propose a efficient method for learning rules based on the features of predicted event e below.

- Each predicted event e is already known and fixed in the learning step.
- e is a 1-event episode.

Suppose a minimal occurrence of A is (t_s, t_e), the rule holds when "e occurs in $S(t_e + time\_lag, t_e + time\_lag + predict\_time\_range)$". This situation is also described as "e occurs at t, then a minimal occurrence of A exists with the occurrence of A's last event in $S(t - time\_lag - predict\_time\_range, t - time\_lag)$". Namely, we can obtain a list of A by the candidate last events of A that are easily extracted from the list of e. In addition, since the number of A is much less than that of all episodes which occur in the event sequence, we can reduce the number of the sequence scanning significantly by extracting a list of A and counting up the held rules before calculating the frequency of each A.

Therefore we define tree-structured episodes, the node of which is an event that contains frequency of the rule and the episode, as Episode Tree (Fig. 8), and the system learns rules by buffering a Episode Tree in the step of extracting a list of A. In these steps below,

1) Extracts a candidate last event list of A from e
2) Acquires all A into an Episode Tree and count the frequency of rules by searching backward in the sequence within $Window$
3) Counts the frequency of each A in the whole sequence.
4) Extracts rules from the Episode Tree with calculating the confidence of them.

And for expanding Episode Tree and counting the number of held rules or episode occurrences, we constructed a procedure based on DFS-MO algorithm [13], which contains recursive compression of the sequence by buffering of the minimal occurrences of parent node and calculation of minimal occurrences of child nodes by one scan, called $occurrence\ deliver$.

Alg.1 is the proposed algorithm for learning the rules from the event sequence. MO means a minimal occurrence. In the function $expand\_Tree$ (row 6) the system expands the Episode Tree with counting the frequency of rules and in the function $expand\_MO$ (row 11) the system counts the frequency of episodes along the expanded Episode Tree in the whole sequence.

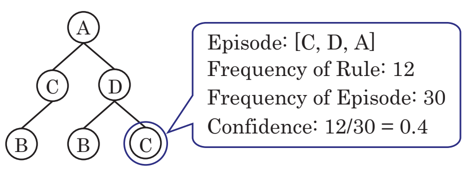


Fig. 8. Episode Tree

E. Prediction Using the Learned Rules

After learning rules of a predicted behavior, the system outputs prediction of the behavior in the way below,

1) Buffers events before in the range of $Window$
2) Extracts the list of A which occurs in the buffered events.
3) If there are detected rules the confidence of which exceeds threshold, then system outputs the prediction.

For example, where the confidence threshold 0.5 and there are rules learned for predicting eating, the contents of which are $A = [A, C, D]$ and $confidence = 0.6$, if events $[A, B, C, D]$ are buffered in this order at the current time then the system outputs prediction of eating, following the learned rules.

Alg. 1 Proposed Algorithm for Learning Rules

Input: $S_{all} \leftarrow$ seq. of all events
Input: $S_{last} \leftarrow$ seq. of candidate last events of ante. episodes

```
1  begin
2  initialize EpisodeTree
3  MOList ← MOs of each 1-event episodes in S_last
4  while  MOList ≠ null do
5        MO ← remove head item of MOList
6        expand_Tree(EpisodeTree, MO)
7  end while
8  MOList ← MOs of each 1-event episodes in S_all
9  while  MOList ≠ null do
10       MO ← remove head item of MOList
11       expand_MO(EpisodeTree, MO)
12 end while
13 get all rules from EpisodeTree
14 end
```

V. Experiment

A. Experimental Condition

In this research, we have Going out, Eating, Sleeping and Taking a bath to be the target behaviors for support, and experiment with predicting the start time of them. The used trajectory data are of two residents living alone in a existing house (the layout of which is in Fig. 1) during different periods. One is living from Apr. 2009 to Mar. 2010 (Subject A), the other from Apr. 2010 to Dec. 2010 (Subject B). The parameters of the experiment are 0.2 [m] within 1 [sec] for deciding the staying activity, 1 [cm] as the mesh size, 1 as the count threshold, 17 as the number of cluster centers (in section III-A), 60 [sec] as the width of $Window$, 30 [sec] as $Time\ Lag$, 120 [sec] as $Predict\ Time\ Range$ (for time-series association rule in section IV-C). In short, the system predicts behaviors in 30-150 [sec] from the current time.

We use all data of subject A of going out and sleeping for learning rules. We also use all data of subject B for going out, sleeping and taking a bath, three months of data of subject B for eating, when he took notes of the start time of it, for learning rules. And then, the system predicts the behaviors with learning results of the last thirty behaviors for eating and the last fifty behaviors for the other behaviors.

TABLE III
Values Used for Evaluating Prediction

		Proper Time for Prediction	
		True	False
Predictive Output	Positive	TP	FP
	Negative	TN	FN

For evaluating the system, we define accuracy, precision, and specificity. With the above-mentioned parameters, $proper\ time\ for\ prediction$ in Table III means 30-150 seconds before the start time of each behavior. Accuracy, precision, and specificity are calculated using the values in the table like:

$$
\begin{aligned}
Accuracy &= \frac{TP}{TP+TN} \\
Precision &= \frac{TP}{TP+FP} \\
Specificity &= \frac{FN}{FP+FN}
\end{aligned}
$$

Actually, accuracy is similar to recall and specificity is the complement of type I error. For calculating accuracy, TP and TN are the number of behaviors. Since we cannot evaluate FP and FN with behavior units, all values in the table are the frame time for calculating precision and specificity.

Since the system which outputs a lot of predictions in improper time is confusing and useless, specificity is to be the most important of the three. There, we calculate accuracy and precision at the confidence threshold of high specificity.

B. Experimental Results

Table IV shows accuracy and precision of each behavior where specificity is 0.99. Overall, the system has high scores on accuracy but low scores on precision. Fig. 9 is an example of succeeded prediction on going out of subject A. In the figure, the assigned numbers is the real IDs of cluster centers in this exam. In the trajectory just before going out, the system extracts the transition events from segmented trajectories and detects the rule of confidence 0.6, 45 seconds before subject A goes out. However, the accuracy of sleeping of subject A have less score than the others. It is because there are some cases of few transition events before the behaviors, for example he is sitting at a table B just before he goes to bed, and that causes no predictive output or those of not enough confidence. Thus, the system can predict the behavior which have movement as preparation of it with high accuracy.

TABLE IV

ACCURACY AND PRECISION AT SPECIFICITY 0.99

	Going out (A)	Going out (B)	Eating (B)
Accuracy	**0.93**	**0.84**	**0.63**
Precision	0.12	0.06	0.06
	Sleeping (A)	Sleeping (B)	Taking a Bath (B)
Accuracy	0.42	**0.81**	**0.97**
Precision	0.03	0.05	0.06

C. Investigation of Precision

Table V is the comparison of the proportion of predictive outputs in 30-600 [sec] before the start time of the behaviors to all predictive outputs and the proportion of predictive outputs in 30-150 [sec] before the start time of the behaviors to all predictive outputs (i.e. precision). The learning results in a short range of two minutes are detected in a little bit wider range. Thus mistaken outputs just because they are done more than 150 [sec] before the behaviors cause a little early preparation of them and they are not necessarily mistakes. In addition, Table VI is the ratio of precision of the system to random noise (a imaginary system that outputs completely

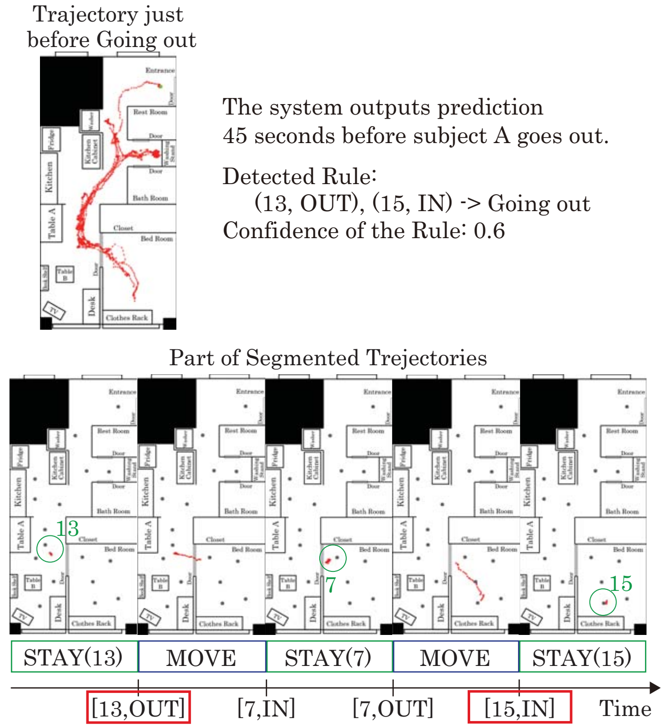


Fig. 9. Example of Succeeded Prediction

TABLE V

EXISTENCE RATE OF PREDICTIVE OUTPUT AT SPECIFICITY 0.99

	Going out (A)	Going out (B)	Eating (B)
30-600 [sec]	0.26	0.18	0.16
30-150 [sec]	0.12	0.06	0.06
	Sleeping (A)	Sleeping (B)	Taking a Bath (B)
30-600 [sec]	0.05	0.08	0.09
30-150 [sec]	0.03	0.05	0.06

TABLE VI

PRECISION RATIO OF THIS SYSTEM TO RANDOM NOISE

	Going out (A)	Going out (B)	Eating (B)
$x:1$	53	46	64
	Sleeping (A)	Sleeping (B)	Taking a Bath (B)
$x:1$	14	39	49

random predictions). You can see that the predictive outputs have considerable entropy. From the viewpoint of entropy, it can be said that the behavior prediction of residents is of use.

VI. CONCLUSION

In this research, we proposed the method for behavior prediction of residents by accumulated trajectories in a house. First, the method performs segmentation of trajectories into staying or moving and classifies each segmented trajectory. And then, the method utilizes time-series association rule mining of the transition events of segmented trajectories to find out the preceding behaviors of the target behavior. The experiment using real trajectories of almost two years demonstrated that the behaviors which have movement as preparation of them can be predicted with high accuracy and considerable precision.

In the future work, since the parameters are decided manually, we will challenge the further improvement of prediction by automated estimation of the parameters, especially of the time-series association rule based on large-scale data. In addition, we will introduce support systems such as service robots, and evaluate the overall system qualitatively.

REFERENCES

[1] C. D. Kidd, R. J. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. Mynatt, T. E. Starner, and W. Newstetter, "The Aware Home: A living laboratory for ubiquitous computing research," in *Proceedings of the 2nd International Workshop on Cooperative Buildings*, 1999, pp. 191 – 198.

[2] B. Binmitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "EasyLiving: Technologies for intelligent environments," in *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing*, 2000, pp. 12 – 29.

[3] J. H. Lee and H. Hashimoto, "Intelligent space," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2000, pp. 1358 – 1363.

[4] Y. Nakauchi, K. Noguchi, P. Somwong, T. Matsubara, and A. Namatame, "Vivid room: human intention detection and activity support environment for ubiquitous autonomy," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2003, pp. 773 – 778.

[5] T. Mori, H. Noguchi, A. Takada, and T. Sato, "Sensing Room: Distributed sensor environment for measurement of human daily behavior," in *1st International Workshop on Networked Sensing Systems*, 2004, pp. 40 – 43.

[6] H. Noguchi, R. Urushibata, T. Sato, T. Mori, and T. Sato, "System for tracking human position by multiple laser range finders deployed in existing home environment," *Aging Friendly Technology for Health and Independence*, pp. 226–229, 2010.

[7] T. Kanda, D. F. Glas, M. Shiomi, H. Ishiguro, and N. Hagita, "Who will be the customer?: A social robot that anticipates people's behavior from their trajectories," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, 2008, pp. 380 – 389.

[8] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3931 – 3936, 2009.

[9] T. Sasaki, D. Brscic, and H. Hashimoto, "Human-observation-based extraction of path patterns for mobile robot navigation," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1401 – 1410, 2010.

[10] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *Internanional Journal on Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[11] T. Katoh, H. Arimura, and K. Hirata, "Mining frequent bipartite episode from event sequences," in *Discovery Science*, ser. Lecture Notes in Computer Science. Springer, 2009, vol. 5808, pp. 136 – 151.

[12] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences," *Data Mining and Knowledge Discovery*, vol. 1, pp. 259 – 289, 1997.

[13] H. Ohtani, T. Kida, T. Uno, and H. Arimura, "Efficient serial episode mining with minimal occurrences," in *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*, 2009, pp. 457 – 464.

[14] S. K. Harms and J. S. Deogun, "Sequential association rule mining with time lags," *Journal of Intelligent Information Systems*, vol. 22, pp. 7 – 22, 2004.

[15] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining sequential patterns by prefix-projected growth," in *Proceedings of the 17th International Conference on Data Engineering*, 2001, pp. 215 – 224.

[16] T. Uno, T. Asai, Y. Uchida, and H. Arimura, "An efficient algorithm for enumerating closed patterns in transaction databases," in *Proceedings of Discovery Science*, 2004, pp. 16 – 31.

[17] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, 1996, pp. 3 – 17.