

Paper:

Sequential Human Behavior Recognition for Cooking-Support Robots

Tsukasa Fukuda*, Yasushi Nakauchi*, Katsunori Noguchi**, and Takashi Matsubara**

*College of Eng. Syst., Grad. School of Syst. and Info. Eng., University of Tsukuba

Tsukuba, Ibaraki 305-8573, Japan

E-mail: fukuda@hri.iit.tsukuba.ac.jp, nakauchi@iit.tsukuba.ac.jp

**Dept. of Computer Science, National Defense Academy

Yokosuka 239-8686, Japan

E-mail: nogupanda@yahoo.co.jp, matsubara@nda.ac.jp

[Received January 26, 2005; accepted May 17, 2005]

Recent advances in information technology are making electric household appliances computerized and networked. If our environments could intuit our activities, e.g., by sensors, novel services taking anticipated actions into account would become possible. We propose activity recognition that infers a subject's next action based on previously observed behaviors. We developed a cooking-support robot that suggests by voice and gesture what the subject may want to do next. Experimental results confirmed feasibility of the inference and the quality of support.

Keywords: intelligent environment, data mining, mobile robot, human-robot interaction

1. Introduction

Recent advances in information technology are computerizing and networking mundane objects such as electric household appliances. The introduction of "intuitive" environments, anticipating our activities, e.g., by sensors, could enable novel services, anticipating and supporting activities, unobtrusively. In this direction, Weiser proposed *ubiquitous computing* [18], which has since emerged as the Aware Home [5], Intelligent Space [9], Robotic Rooms I and II [12, 17], Easy Living [3, 8], Smart Rooms [15, 16], etc.

A key factor in such systems is the use of ubiquitous sensors to recognize human behavior. Intelligent Space detects the location of a subject using multiple ceiling cameras and having a mobile robot follow the subject [9]. Easy Living detects a subject's location and turns on nearby lights [3, 8]. These systems provide services by anticipating human intention in movement. Robotic Room I anticipates human intention more explicitly, e.g., in visual recognition of a finger pointed by a bedridden subject Robotic Room I extends a robotic manipulator to hand the object pointed at to the subject [17].

To recognize implicit intentions, Asaki et al. have proposed recognizing human behavior, e.g., changing clothes

and preparing meals, using a state transition model [1]. Moore et al. have proposed Bayesian classification enabling the recognition of behavior via learning [10, 11]. We have proposed intention recognition using an ID4-based learning algorithm and succeeded in the recognition of the intent to study, eat, rest, etc. [13].

While such research enables the recognition of certain human activities, support remains rather limited. Assume, for example, that a subject is making a cup of instant coffee, the ubiquitous system could helpfully suggest where the cream is, but to do so, it would have to know the time series of procedures and infer the subject's next action based on previously observed behavior.

We propose activity recognition that infers subsequent action taking past actions into account. With one-dimensional barcodes on merchandise in such environments as supermarkets expected to eventually be replaced by IC tags providing the manufacturer's name, the type of merchandise, the place of production, the expiration date, etc., it is only a short step to anticipate that all objects in the home will carry such labels. We assume that food have cooking utensils, tableware, and cutlery in a kitchen have IC tags and their movements is observable by antennas on shelves and kitchen counters.

We developed kitchen support using mobile robot using voice and gestures to suggest the next action a subject may want take. Our experiments confirmed the feasibility and quality of support.

2. System Design

2.1. Inference from Series of Actions

We start by discussing the types of behavior we must recognize, defining an observed action by sensors as *action* a_i and a set of actions as $A = \{a_1, a_2, \dots, a_n\}$, e.g., an action sequence for coffee making consisting of a_1 : "take a cup from the cupboard," a_2 : "take instant coffee from the cupboard," a_3 : "take a spoon from the drawer," and a_4 : "take a thermos jug of hot water."

We define a set of time series actions of arbitrary

length as *action pattern* p_i and a set of action patterns as $P = \{p_1, p_2, \dots, p_m\}$. We observed action pattern $p_o = \{\mathbf{a}_3, \mathbf{a}_2, \mathbf{a}_4\}$ in a subject and recorded action pattern $p_i = \{\dots, \mathbf{a}_3, \mathbf{a}_2, \mathbf{a}_4, a_6, \dots\}$ in database P , a collection of action patterns observed thus far.

We find the same time series action pattern in p_i and infer that the next action to be executed is action a_6 .

Subjects may behave redundantly or concurrently, e.g., p_o may include a_n as $\{\mathbf{a}_3, \mathbf{a}_2, a_n, \mathbf{a}_4\}$ or p_i may include a_n as $\{\dots, \mathbf{a}_3, a_n, \mathbf{a}_2, \mathbf{a}_4, a_6, \dots\}$. These actions are considered noise when original time series actions consist of making coffee, cooking hamburgers, etc., so we need a type of inference that is robust against such noise.

Added to is, time series actions may branch, e.g., a subject making coffee may add sugar and another subject may add cream after making black coffee. This means that $p_j = \{\dots, \mathbf{a}_3, \mathbf{a}_2, \mathbf{a}_4, a_7, \dots\}$ may exist in addition to p_i above. If inference uses both time serial information and the frequency of action patterns, it can predict a more appropriate next action e.g., when a subject who always drink black coffee adds cream, this is recognized immediately as a behavioral change.

2.2. Time-Sequence Data Mining

Data mining is categorized roughly into four types correlation analysis, time-sequence analysis, clustering, and learning [6]. Time-sequence analysis best suits our purpose. In this section, we briefly explain typical algorithms that extract temporal sequences in time-series patterns.

The a priori algorithm proposed by Agrawal is used in data for temporal sequential data [2], as we show through examples. Assume four time-series data sets $p_1 = \{\mathbf{a}_3, a_2, \mathbf{a}_4, a_6\}$, $p_2 = \{\mathbf{a}_3, a_2, \mathbf{a}_4, a_6\}$, $p_3 = \{\mathbf{a}_3, a_2, a_5, a_6\}$, $p_4 = \{\mathbf{a}_3, a_1, \mathbf{a}_4, a_6\}$ in a database. The a priori algorithm extracts partial sequences from data sets by taking into account the number of occurrences and certainty given by a user. It finds, for example, partial sequences $\{\mathbf{a}_3, \mathbf{a}_4\}$, which means “ \mathbf{a}_4 occurs after \mathbf{a}_3 .” Certainty is the occurrence ratio, i.e. \mathbf{a}_4 happens after \mathbf{a}_3 75% of the time. Experiments show that the computational cost increases exponentially as the number of data sets increases with the a priori algorithm.

Pei proposed the PrefixSpan algorithm, which extracts multiple-frequency patterns efficiently which minimizing computational cost [14]. Assume four time-series data sets $p_1 = \{\mathbf{a}_3, \mathbf{a}_2, \mathbf{a}_4, \mathbf{a}_6\}$, $p_2 = \{\mathbf{a}_3, \mathbf{a}_2, \mathbf{a}_4, \mathbf{a}_6\}$, $p_3 = \{\mathbf{a}_3, \mathbf{a}_2, a_5, \mathbf{a}_6\}$, $p_4 = \{\mathbf{a}_3, a_1, \mathbf{a}_4, \mathbf{a}_6\}$ as in the above example. PrefixSpan extracts partial sequences with the number of occurrences shown in **Fig.1**. $\{a_3/4, a_2/3, a_4/2, a_6/2\}$ shows time-series data $\{\mathbf{a}_3, \mathbf{a}_2, \mathbf{a}_4, \mathbf{a}_6\}$ with frequency indicated by suffixes, i.e., the occurrence of \mathbf{a}_3 alone is 4 and the occurrence of $\{\mathbf{a}_3, \mathbf{a}_2, \mathbf{a}_4, \mathbf{a}_6\}$ as time-series data is 2.

2.3. Behavior Inference Algorithm

With exact matching enabled by such time-sequence data mining, we propose a behavior inference algorithm

$a_1/1$	$a_4/1$	$a_6/1$	
$a_1/1$	$a_6/1$		
$a_2/3$	$a_4/2$	$a_6/2$	
$a_2/3$	$a_5/1$	$a_6/1$	
$a_2/3$	$a_6/3$		
$a_3/4$	$a_1/1$	$a_4/1$	$a_6/1$
$a_3/4$	$a_1/1$	$a_6/1$	
$a_3/4$	$a_2/3$	$a_4/2$	$a_6/2$
$a_3/4$	$a_2/3$	$a_5/1$	$a_6/1$
$a_3/4$	$a_2/3$	$a_6/3$	
$a_3/4$	$a_4/3$	$a_6/3$	
$a_3/4$	$a_5/1$	$a_6/1$	
$a_3/4$	$a_6/4$		
$a_4/3$	$a_6/3$		
$a_5/1$	$a_6/1$		
$a_6/4$			

Fig. 1. Time-series data generated by PrefixSpan.

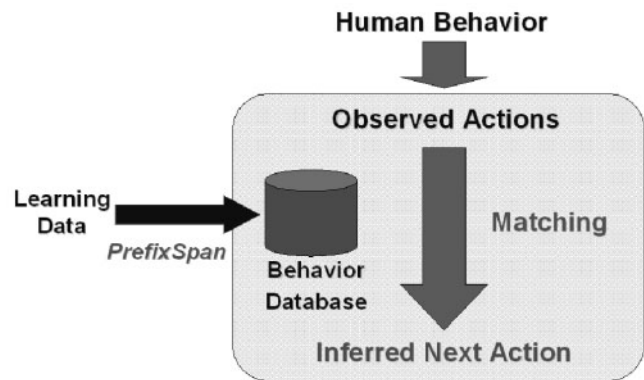


Fig. 2. Overview of proposed inference engine.

that takes into account noise in both time-sequence data in the database and in observation data.

Inference for predict a subject’s next behavior involves finding the same time-sequence data as window data from the behavior database, which consists of an enormous amount of time-sequence data observed in the past. We used PrefixSpan to generate partial time-sequence data from the behavior database since because PrefixSpan minimizes computational cost and is easy to use.

We developed window-based matching to make our proposed inference engine (Fig.2) robust against noise. We used certainty to indicate the confidence of inferred results.

We start by defining terms used in the algorithm. Each behavior observed by sensors is defined as *input data* w_i . Assume that the latest input data is w_i and the number of W input data is $\{w_{i-W+1}, \dots, w_i\}$ observed recently. W time-series data is defined as *window data* of width W .

Matching between the input data and behavior database stet window size. If we start out to find the time-series input data of window size 5 and cannot locate the exactly same sequence, for example, we reduce the window width to 4, 3, or 2, so even if input data contains some noise, we

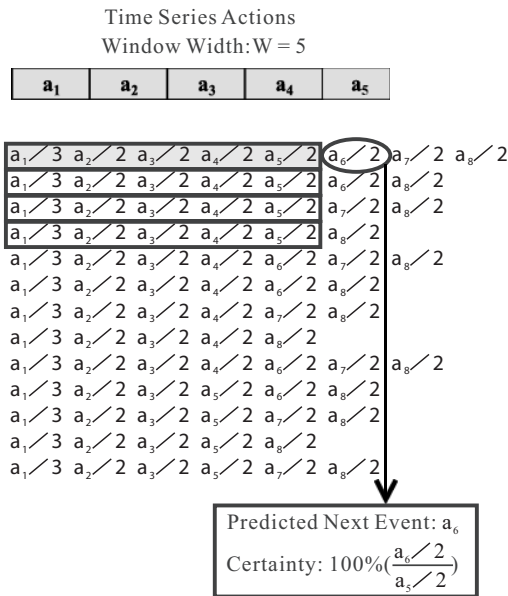


Fig. 3. Matching algorithm.

find the exact same data in the database. The maximum window size used at the beginning of a search is defined as W_{max} and the minimum is W_{min} . To infer a subject's next action, we must know several cases of time-sequence data as a hint, so W_{min} is used for terminating the search.

An *inferred event* is the action that occurs following the matching of the time-sequence data with window size W at the highest certainty. *Certainty* is calculated as follows:

$$certainty = \frac{O_{ia}}{O_{pa}}, \quad (0 \leq certainty \leq 1) \quad \dots (1)$$

where O_{ia} is the occurrence of inferred action and O_{pa} is the occurrence of previous action to the inferred action calculated by PrefixSpan (Fig.3).

The inference engine outputs one of the following: end of sequence (EOS), inferred event with certainty, or null. EOS is output when observed time-series data matches in the database but the most recent observed event is EOS in the database match, meaning that the inference engine could not infer the next event. When the inference engine finds matched time-series data and a subsequent event in the database, it outputs the succeeding event as the inferred event with certainty calculated by formula (1). Null is output when the inference engine cannot find matching data in the database even though it has reduced time-series actions to window width W_{min} .

The overall algorithm of our proposed inference engine is as follows:

- 1) A window with size $W = 5$ is created and all contents are initialized as Null.
- 2) Events observed at most $W = 5$ are input to the window so the most recent event becomes w_W .
- 3) The exact same time-series data as in the window in the database is found.

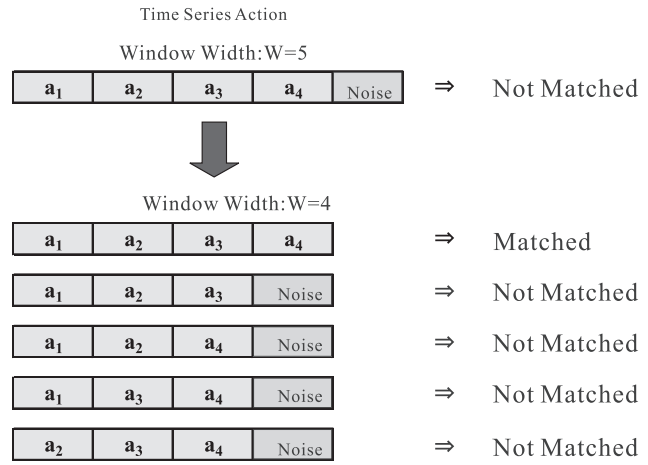


Fig. 4. Reduction of window size.

- 4) If matched data matches only one, the inferred ($W + 1$)th event in the database is output with certainty calculated by formula (1). If no ($W + 1$)th event exists, EOS is output.
- 5) If the number of matched data is multiple, the highest certainty ones are selected, then the longest sequenced ones¹. If multiple candidates still remain after select on, arbitrary candidates are selected as matched data. Stet inferred event or EOS is output as in procedure 4.
- 6) If no data matches from procedure 3, the window size is reduced to $W = W - 1$ as shown in Fig.4. Matched data is then found with these multiple window as in procedure 3. When the window size becomes $W < W_{min}$, it null is output because no is found matched data in the database. The above procedures are summarized in Table 1.

2.4. Activity Support

While have a system suggest the next action a subject may do, it is important that it do so unobtrusively and deferentially. With this in mind, we developed a mobile robot that recommends the next action by voice and gesture.

To minimize the possibility of unsuitable recommendations, we give certainty a threshold based on formula (1), having the robot make a suggestion only if the certainty of an inferred event exceeds the threshold.

We conducted experiments with 10 subjects and collected comments on whether recommendations (inferred events) were suitable. Results confirmed that most certainty subjects felt was unsuitable was below 0.55. We thus set the threshold to 0.55 and had the robot desist from making recommendations whose certainty was below them.

1. This is because the longer the time series data in the database, the more detailed procedures become.

Table 1. Matching algorithm.

No.	Procedures
1	Set $W = 5$ and create window $\{w_1, w_2, w_3, w_4, w_5\}$ with all values as null.
2	Input observed (at most) $W = 5$ events to window to make most recent event become w_W .
3	Finds exactly same time series events as for window in database. Matched data is singular: Go to 4. Matched data is plural: Go to 5. Matched data is zero: Go to 6.
4	Output inferred event ($(W + 1)th$ event in database) with certainty calculated by formula (1). Output EOS if $(W + 1)th$ event not found.
5	Select highest certainty, then select longest sequences.
6	Reduce window size to $W = W - 1$: Go to 3. If window size becomes $W < W_{min}$, it outputs null.



Fig. 5. Items with IC tags.

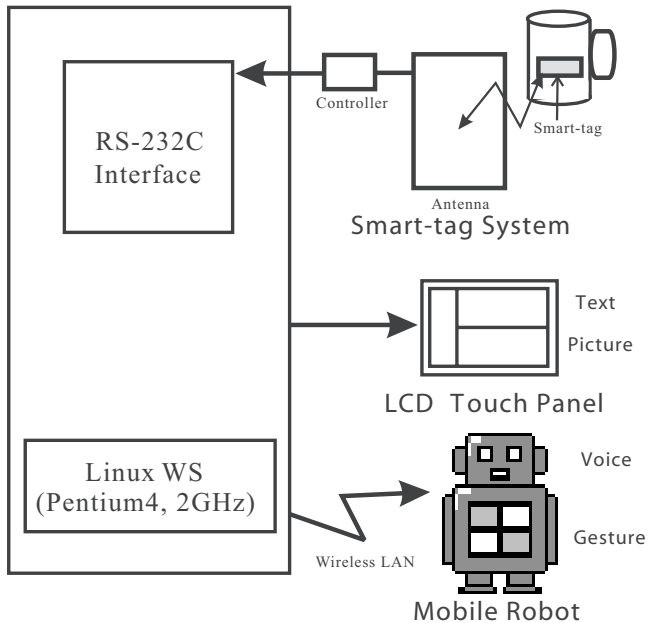


Fig. 6. Cooking support robot configuration.

3. Implementation

3.1. IC Tags

Based on our assumption that barcoded merchandise in supermarkets and department stores will eventually have IC tags, and that most items in the home and office will have IC tags, we assumed that we would trace their location and movement using antennas.

We used IC tags (Feig Electronics Co. Ltd.) with labels $2\text{cm} \times 5.5\text{cm}$. The $30\text{cm} \times 40\text{cm}$ antenna reads/writes IC tag information within 15cm.

We attached IC tags to ordinary home and kitchen items, e.g., cups, glasses, pots, instant coffee, tea bags, cream, sugar, potatoes, carrots, spoons, forks, knives, medicine chest, disinfectant, cotton and adhesive bandage (Fig.5).

Tag information was acquired by PC via an RS-232C serial link. The inference engine is implemented on the PC and inferred events are transferred to the mobile robot via a wireless LAN (Fig.6).

3.2. Inference of Next Action

To acquire learning instances for PrefixSpan, we asked ten subjects to conduct five tasks – 1) making a cup of coffee, 2) making a cup of tea, 3) treating a cut finger, 4) taking cold medicine, and 5) making rice curry (Fig.7).

To predict behavior precisely, we used IC tag information and information of tag location – a: cupboard, b: cabinet, c: medicine chest – and human action – 0: removed, 1: stored. Event Spoon-a0, for example, shows that a spoon is removed from a cupboard.

In current implementation, we use only one IC tag antenna, so we have subjects scan items. The system knows that the tagged IC object is removed when the IC tag is observed by the antenna for the first time, then that it was stored when the same IC tag is observed a second time. Storage places for items are predefined depending on the item and hard-coded in a program. In the future, by installing IC tag antennas on each shelf and kitchen counter, we can have the system know item locations in real time so subjects need not scan items.

```

Cup-a0 Pot-a0 Pot-a1 TeaBag-a0 TeaBag-a1 Sugar-a0 Sugar-a1 Cream-a0 Cream-a1 Spoon-a0
Cup-a0 Pot-a0 Pot-a1 TeaBag-a0 TeaBag-a1
Cup-a0 Pot-a0 Pot-a1 TeaBag-a0 TeaBag-a1
Cup-a0 Pot-a0 Pot-a1 TeaBag-a0 TeaBag-a1
Cup-a0 Pot-a0 Pot-a1 TeaBag-a0 TeaBag-a1
Cup-a0 Pot-a0 Pot-a1 TeaBag-a0 TeaBag-a1
Cup-a0 Pot-a0 Pot-a1 TeaBag-a0 TeaBag-a1 Lemon-a0 Lemon-a1 Spoon-a0
Cup-a0 TeaBag-a0 Pot-a0 Pot-a1 TeaBag-a1 Sugar-a0 Sugar-a1 Spoon-a0
Cup-a0 TeaBag-a0 Pot-a0 Pot-a1 TeaBag-a1
Cup-a0 TeaBag-a0 Pot-a0 Pot-a1 TeaBag-a1 Lemon-a0 Lemon-a1
    
```

Fig. 7. Example of learning data.

```

Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Cream-a0/1 Cream-a1/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Cream-a0/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Cream-a1/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Lemon-a0/2 Lemon-a1/2 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Lemon-a0/2 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Lemon-a1/2 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Spoon-a0/3
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a0/2 Cream-a0/1 Cream-a1/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a0/2 Cream-a0/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a0/2 Cream-a1/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a0/2 Spoon-a0/2
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a0/2 Sugar-a1/2 Cream-a0/1 Cream-a1/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a0/2 Sugar-a1/2 Cream-a0/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a0/2 Sugar-a1/2 Cream-a1/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a0/2 Sugar-a1/2 Spoon-a0/2
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a1/2 Cream-a0/1 Cream-a1/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a1/2 Cream-a0/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a1/2 Cream-a1/1 Spoon-a0/1
Cup-a0/20 Pot-a1/10 TeaBag-a1/10 Sugar-a1/2 Spoon-a0/2
    
```

Fig. 8. Example of time-series data generated by PrefixSpan.

The time-sequence database generated by PrefixSpan from the learning data (Fig.7) is shown in Fig.8. Data {Cup-a0/20, Pot-a1/10, TeaBag-a1/10, Spoon-a0/3}, for example, shows that the event Cup-a0 alone was observed 20 times in learning data but that the {Cup-a0, Pot-a1, TeaBag-a1, Spoon-a0} sequence – the cup is removed from the cupboard, the pot is stored in the cupboard, the tea bag is stored in the cupboard, and the spoon is removed from the cupboard – was observed 3 times.

3.3. Cooking-Support Robot

The mobile robot Robovie (ATR) [7] served as cooking support (Fig.9). The robot recommended anticipated action by synthesized voice and gestures.

Overall, we confirmed that the following support was realized: When a subject removed a cup and instant coffee from the cupboard, the robot anticipates the next action by saying “sugar is in the cupboard” and turning toward the cupboard and pointing to the shelf where the sugar is. When a subject took cold medicine and stored it in the medicine chest, the robot suggested that “the medicine chest should be stored on the shelf” and pointed to the shelf. These recommendations are automatically generated from inferred events such as Sugar-a0 and MedicineBox-b1 etc.



Fig. 9. Cooking-support robot recommending a presumed next action by voice and gesture.

4. Experimental Results

To evaluate the feasibility and quality of support, we conducted experiments with 10 subjects other than those used for collecting learning instances. We instructed the

Table 2. Evaluation phrases.

Score	Phrase
+1	“Thank you.”
0	(silent/ignore)
-1	“No thanks.”

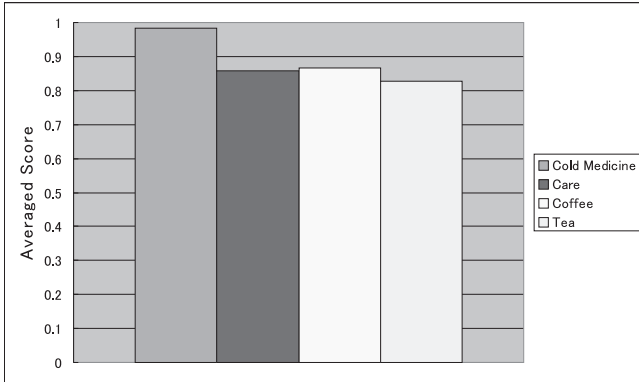


Fig. 10. Evaluation of task adequacy.

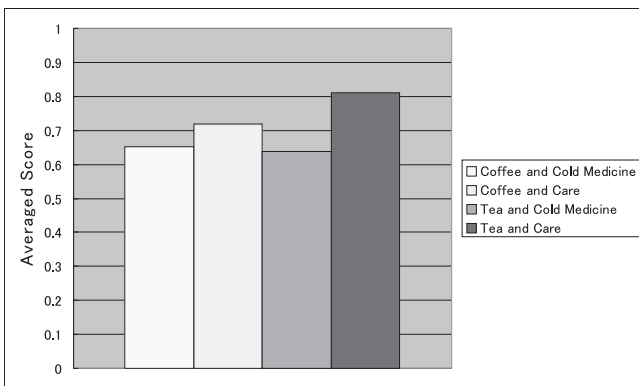


Fig. 11. Evaluation of adequacy when two tasks were interleaved.

new subjects to speak short phrases based on how they felt, each time they heard suggestions from the robot. These phrases, shown in **Table 2**, were graded -1 to +1. We videotaped experiments and counted scores each time the robot made suggestions.

To confirm the adequacy of suggestions in each task (task 1 to 4 explained in Section 3.2), we asked subjects to perform each task. Averaged scores are shown in **Fig.10** and all exceeded 0.8. Results thus confirmed that the system inferred actions appropriately and suggestions by the robot were accepted by subjects.

To evaluate robustness against noises in observed time-series data, we instructed two subjects to perform different tasks, e.g., having one to make a cup of coffee and the other take cold medicine. Their actions (item use) will be interleaved and actions of one subject become noise in the other. Averaged scores are shown in **Fig.11**.

Even though the two tasks were interleaved, averaged scores were still high, about 0.7. Results confirmed that the reduction of window size (explained in Section 2.3) effectively made the system robust against noise in observed action sequences.

5. Conclusions

We have proposed behavior recognition in which subsequent action is anticipated by taking into account previously observed behavior. We developed a cooking-support robot that recommends subsequent action by voice and gestures. Experimental results confirmed the feasibility of the proposed inference and the quality of support.

Our proposed recognition system is as follows.

- 1) It is robust against noise in both the time-sequence data in the database and in observation data. Such noise is inevitable in systems that accept free activity in intelligent space. Robustness was confirmed by experimental results.
- 2) Certainty is calculated with inferred action, the application to support subjects based on confidence.
- 3) Data mining enabled the system to adapt to many applications by expanding data. If we import recipes conducted by professional cooks, it will be attractive to both new and experienced cooks.

We are planning to have the system suggest recipes by taking account food available in the kitchen, and to extend the system so that it detects more precise and detailed activities using a variety of sensors such as vision and laser etc.

References:

- [1] K. Asaki, Y. Kishimoto, T. Sato, and T. Mori, “One-Room-Type Sensing System for Recognition and Accumulation of Human Behavior –Proposal of Behavior Recognition Techniques–,” Proc. of JSME ROBOMECH’00, 2P1-76-119, 2000.
- [2] R. Agrawal, and R. Strikant, “Fast algorithms for mining association rules,” Proc. of the 20th International Conference on Very Large Databases, pp. 487-499, 1994.
- [3] B. Brumitt et al., “Easy Living: Technologies for Intelligent Environments,” Proc. of International Symposium on Handheld and Ubiquitous Computing, 2000.
- [4] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, “EasyLiving: Technologies for Intelligent Environments,” Proc. of International Symposium on Handheld and Ubiquitous Computing, pp. 12-29, 2000.
- [5] I. A. Essa, “Ubiquitous sensing for smart and aware environments: technologies towards the building on an aware home,” Position Paper for the DARPA/NFS/NIST workshop on Smart Environment, 1999.
- [6] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, “Advances in Knowledge Discovery and Data Mining,” MIT Press, 1996.
- [7] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, “Robovie: A robot generates episode chains in our daily life,” Proc. of Int. Symposium on Robotics, pp. 1356-1361, 2001.
- [8] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, “Multi-Camera Multi-Person Tracking for Easy Living,” Proc. of 3rd IEEE International Workshop on Visual Surveillance, pp. 3-10, 2000.

- [9] J. Lee, N. Ando, and H. Hashimoto, "Design Policy for Intelligent Space," Proc. of IEEE International Conference on System, Man and Cybernetics (SMC'99), pp. 12-15, 1999.
- [10] D. J. Moore, I. A. Essa, and M. H. Hayes III, "ObjectSpaces: Context Management for Human Activity Recognition," Georgia Institute of Technology, Graphics, Visualization and Usability Center, Technical Report, #GIT-GVU-98-26, 1998.
- [11] D. J. Moore, I. A. Essa, and M. H. Hayes III, "Exploiting Human Actions and Object Context for Recognition Tasks," Proc. of The 7th IEEE International Conference on Computer Vision, pp. 80-86, 1999.
- [12] T. Mori, T. Sato et al., "One-Room-Type Sensing System for Recognition and Accumulation of Human Behavior," Proc. of IROS2000, pp. 345-350, 2000.
- [13] Y. Nakauchi et al., "Vivid Room: Human Intention Detection and Activity Support Environment for Ubiquitous Autonomy," Proc. of IROS2003, pp. 773-778, 2003.
- [14] J. Pei et al., "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth," Proc. of International Conference of Data Engineering, pp. 215-224, 2001.
- [15] A. Pentland, "Smart Rooms," Scientific American, pp. 54-62, 1996.
- [16] A. Pentland, R. Picard, and P. Maes, "Smart Rooms, Desks, and Clothes: Toward Seamlessly Networked Living," British Telecommunications Engineering, Vol.15, pp. 168-172, July, 1996.
- [17] T. Sato, Y. Nishida, and H. Mizoguchi, "Robotic Room: Symbiosis with human through behavior media," Robotics and Autonomous Systems 18 International Workshop on Biorobotics: Human-Robot Symbiosis, Elsevier, pp. 185-194, 1996.
- [18] M. Weiser, "The Computing for the Twenty-First Century," Scientific American, pp. 94-104, September, 1991.



Name:
Yasushi Nakauchi

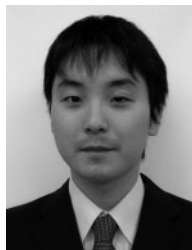
Affiliation:
Associate Professor, Department of Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba

Address:
Tsukuba, Ibaraki 305-8573, Japan

Brief Biographical History:
1993- Research Associate at National Defense Academy
1994- Assistant Professor at National Defense Academy
1998-1999 Visiting Scientist at Carnegie Mellon University
1998-2003 Associate Professor at National Defense Academy
2003- Associate Professor at University of Tsukuba

Main Works:
• "A Social Robot that Stands in Line," Autonomous Robots, Vol.12, pp. 313-324, 2002.

Membership in Learned Societies:
• The Japan Society of Mechanical Engineers (JSME)
• The Robotics Society of Japan (RSJ)
• The Institute of Electrical and Electronics Engineers (IEEE)



Name:
Tsukasa Fukuda

Affiliation:
Graduate School of Systems and Information Engineering, University of Tsukuba

Address:
Tsukuba, Ibaraki 305-8573, Japan

Main Works:
• "Time Series Action Support by Mobile Robot in Intelligent Environment," Proc. of the 2005 IEEE International Conference on Robotics and Automation (ICRA2005), pp. 2908-2913, 2005.

Membership in Learned Societies:
• The Robotics Society of Japan (RSJ)



Name:
Katsunori Noguchi

Affiliation:
Master Course Student, School of Computer Science, National Defense Academy

Address:
1-10-20 Hashirimizu, Yokosuka 239-8686, Japan

Brief Biographical History:
1998-1999 Maritime officer candidate school, Japan Maritime Self Defense Force (JMSDF)
1999-2001 Ensign at JMSDF
2001- LTJG at JMSDF
2002- Master Course Student at School of Comp. Sci., National Defense Academy

Main Works:
• "Vivid Room: Human Intention Detection and Activity Support System for Ubiquitous Autonomy," Proc. of 2003 IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS2003), pp. 773-778, 2003.

Membership in Learned Societies:
• The Robotics Society of Japan (RSJ)



Name:

Takashi Matsubara

Affiliation:

Assistant Professor, Department of Computer Science, National Defense Academy

Address:

1-10-20 Hashirimizu, Yokosuka 239-8686, Japan

Brief Biographical History:

1991- Reseach Associate at National Defense Academy

2002- Assistant Professor at National Defense Academy

Main Works:

• "On M-Sequence Generators with Fault Detecting Checkers and Their Characteristics," IEICE Trans. D-I, Vol.J81-D-I, No.3, pp. 283-291, 1998.

Membership in Learned Societies:

- The Institute of Electronics, Information and Communication Engineers (IEICE)
 - The Institute of Electrical and Electronics Engineers (IEEE)
-