Proc. 41st Annual Conf. of the IEEE Industrial Electronics Society (IECON-2015), pp. 1507-1512
Yokohama, Japan, Nov. 2015.

Human Activity Modeling and Prediction for Assisting Appliance Operations

Yuichiro Koiwa†, Jun Miura†, and Koki Nakagawa‡
† Department of Computer Science and Engineering, Toyohashi University of Technology
‡ LG Electronics Japan Lab

***Abstract*—Recent increase of advanced appliances at home requires lots of user's operations for setting them in a desired state. Although many appliances have been developed which can adapt to the state of a room and a person, they are mainly based on simple state values such as the room temperature. More intelligent assistance will be needed for controlling electronic appliances such as a TV or an audio system. This paper describes a method of modeling human activities and predicting them for controlling appliances. We have developed an experimental system in a room with furniture and appliances. The system observes human motions and appliance operations and compiles them into *actions*, and then constructs a model of possible action sequences, which is further used for human action prediction. We test two prediction methods: a probabilistic action graph-based and an SVM (support vector machine)-based. We evaluate the methods using actual observation data for over fifty days in total. We also implement an on-line appliance control system as a proof-of-concept.**

***Index Terms*—Human activity modeling, Human activity prediction, Advanced Appliances**

I. Introduction

Recent increase of advanced appliances and electronic devices requires lots of user's operations for setting them in a desired state. Although many "smart" appliances can adapt to the state of a room and people inside, most of them change their operations based on simple state values such as room temperature and the amount of sunlight through windows. Learning how users operate appliances and controlling themselves proactively could contribute to realizing comfortable home environments. Such an assistance could also be appropriate for supporting elderly living alone.

Proactive control requires prediction of human activity. Since, in everyday life, human tend to behave routinely to some extent, we could observe human activities to find a pattern of actions to be executed with a high probability, and then use those patters for prediction.

There are many works on human motion learning and prediction. Statistical models such as Hidden Markov models and Gaussian Process have often been applied to human movement prediction [1], [2], [3] and body motion prediction [4]. These works are concerned mainly with activity classification. Various action recognition approaches [5] can also be applied to the classification and prediction.

Action recognition has been actively investigated in daily life actions such as cooking and preparing tables. Due to a large variety of possible action sequences, probabilistic approaches are often proposed. Tenorth et al. [6] model human everyday activities with probabilistic partial order relations of actions use Bayesian Logic Networks (BLNs). Koppula et al. [7] use Markov Random Fields (MRFs) to jointly model human activities, object affordances, and their temporal relationships. These models are for trying to make a model for recognizing complex and various actions in daily life.

Prediction of appliance operations is that of some specific events, and human activities need to be modeled as a sequence of events. Mori et al. [8] proposed a method of segmenting a sequence of position data into that of stay points, with which events are associated, and of learning association rules [9] for event prediction.

In this paper, we focus on predicting a human's appliance-operating action (event) from preceding action sequences before it. An example preceding sequence is: a person takes a coffee and goes to a sofa to sit. This sequence could imply that the person will turn on a TV in a near future. Such a prediction can be performed by pattern matching approaches. In this paper, we test two methods. One is association rule-based, that is, the system searches an event sequence database for a sequence which is the same as the current one, and picks up an event in the subsequent sequence as predicted if it has occurred significantly frequently compared to the others. The other method is classifier-based. Hoai and De la Torre [10] used a variant of SVM to *early event detection*, which detects an event before it finishes from a partial sequence of the event. We take this approach using SVM as a classifier.

We have developed an experimental room, where human actions are automatically recognized and recorded as event sequences. We evaluate the methods using the recorded data for over fifty days in total. We have also implemented an on-line appliance control system based on the prediction as a proof-of-concept.

II. Experimental Room

Assisting people to operate appliances is the goal of the developed system. For this purpose, we setup an experimental room, where several appliances (TV, audio system, LED light on the desk, and a window fan) are set to be computer-controlled. We put four Kinects for observing human activities. Fig. 1 illustrates the current setting.

A. Human action observation

We obtain the following three kinds of information: (1) position, (2) state, and (3) gestures.

Human position estimation is done as follows. Each Kinect tracks a person independently using the skeleton tracking

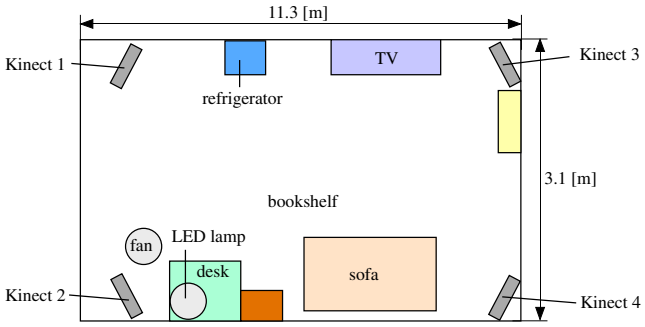


Fig. 1. Setting of the experimental room.

function, and the mean of the positions obtained by the Kinects are used as the person position. We choose one Kinect (*main Kinect*) and calibrate the others with it in advance; every position information is transformed to the one in the coordinate frame of the main Kinect. Since all Kinects cannot see one calibration pattern (i.e., chess board) at once, we take a sequence of raised hand position data for calibration.

There are two types of human actions; one is about the movement and the other is about appliance operation. We consider three actions for the former: *stand*, *seat*, and *walk*. The height and the velocity of the person are used for recognizing these actions, with thresholds determined experimentally. We do this classification every frame and perform a winner-takes-all selection for over 30 frames. Actions for the latter is detected by recognizing gestures for appliance control, explained in the next subsection.

B. Appliances control

TV, audio, and fan are controlled by a device which can generate the corresponding infrared signal patterns. For controlling an LED light on the desk, we use Philips Hue system. We record all operations on these appliances. For a person to operate them, we also developed a gesture recognition method which recognizes five hand gestures: pointing, moving up, moving down, moving left, and moving right. Pointing gestures are assigned to turn-on and turn-off operations. The others are assigned to other operations depending on the appliances. In the case of TV, for example, up-down operations are for controlling the volume and left-right ones for choosing programs.

When controlling an appliance, human usually directs his/her face it. To verify this condition, we obtain the face orientation using KinectFace API and calculate the difference between the face orientation and the direction of an appliance for identifying the appliance to be controlled.

C. Object recognition

The system recognizes a set of things the person might hold in the room. The recognition is done by the extraction of hand regions and application of a classifier.

Hand region extraction uses the skeleton tracking data. We set a specific volume around the hand position and, using the depth data, extract the image inside that volume. Fig. 2 shows an example of hand region extraction.

(a) input image. (b) Extracted hand region.

Fig. 2. Hand region extraction using depth.

(a) drink cartons.
(b) books.
(c) empty hands and others.

Fig. 3. Positive and negative images for training.

We use DeepBeliefSDK [11], which is an implementation of a convolutional neural network architecture [12] running on Linux, for object recognition. In this research, we consider two object classes: drink carton and book. We use five objects for each class and collected 200 hand region-extracted images for each object (i.e., 1000 images for one class). We also captured 1000 negative images of empty hands and miss-extracted hand regions. Fig. 3 shows examples of positive and negative images.

D. Recording human activities

The observation system continuously observes a human in the room and outputs the latest *state* when his/her action changes. The state is described by the following elements: name, appliance to operate with control parameters, movement, thing held, time, position, and the state of all appliances. The detailed explanation of each element is given in Table I. Fig. 4 shows a part of an actual recorded action sequence.

We recorded three sets of data for two subjects (P1 and P2). Table II summarizes the data sets. Fig. 5 shows the durations of using the appliances for *LongStay* data.

III. Human Activity Modeling and Prediction

Human activity prediction is a computation to predict a future human motion from past experiences. We thus model the recorded action sequences in a representation and use them

```
koide NULL NULL 0 NULL Nothing 10 17 1 16 26 52 0 320 114 0.172873 -0.0254232 1.87349 0 1 0 0 0 5 11 0 12 0 0 0 0
koide NULL NULL 0 Stand Drink 10 17 1 16 27 15 0 320 114 0.403484 0.0250023 2.66984 0 1 0 0 0 5 11 0 12 0 0 0 0
koide NULL NULL 0 Walk Drink10 17 1 16 27 17 0 320 114 0.538193 0.138809 3.68171 0 1 0 0 0 5 11 0 12 0 0 0 0
koide NULL NULL 0 Stand Drink 10 17 1 16 27 27 0 320 114 0.495717 -0.136097 3.85499 0 1 0 0 0 5 11 0 12 0 0 0 0
koide NULL NULL 0 Seat Drink 10 17 1 16 27 50 0 320 114 0.601284 -0.0722131 3.96633 0 1 0 0 0 5 11 0 12 0 0 0 0
koide TV Power 1 Seat Drink 10 17 1 16 27 50 0 320 114 0.601284 -0.0722131 3.96633 0 1 0 0 1 5 11 0 12 0 0 0 0
```

Fig. 4. Part of an actual recorded action sequence.

TABLE I
STATE DESCRIPTIONS.

element	description
person ID	ID of the person.
appliance ID	ID of the appliance to be operated. Null for no operations.
parameters of operation	parameters to control the appliance (e.g., turn-on or turn-off, TV channel, volume level, etc.)
movement type	stand, seat, or walk
thing held	name of thing the person is holding (drink, book, or NULL)
time	month, day, week of the day, hour, minute, and second.
position	3D position of the head.
appliance states	all parameters of all appliances in the room.

TABLE II
SUMMARY OF REAL ACTIVITY DATA.

label	person	# of days	# of actions	comments comments
LunchStay1	P1	24	710	take lunch and rest†.
LunchStay2	P2	13	290	take lunch and rest†.
LongStay	P1	21	1219	work, rest, and eat.

† The time to take lunch is not necessarily around noon.

for predicting, with consulting the current human state. In this paper, we test two modeling and prediction methods. To apply these methods, we first convert the observed action data to *action nodes*, which represent actions with discrete time and position data.

A. Conversion to action node

An action node holds operation, thing held, position, time, and states of appliances. The time is divided into three durations: *morning* between 6:00-12:00, *afternoon* between 12:00-18:00, and *evening* between 18:00-24:00. Position data are also discretized using a set of representative positions, which are the mean positions of a Gaussian mixture fitted using the EM algorithm to all of recorded position data. The states of appliances here are limited to its power state (PowerOn or PowerOff).

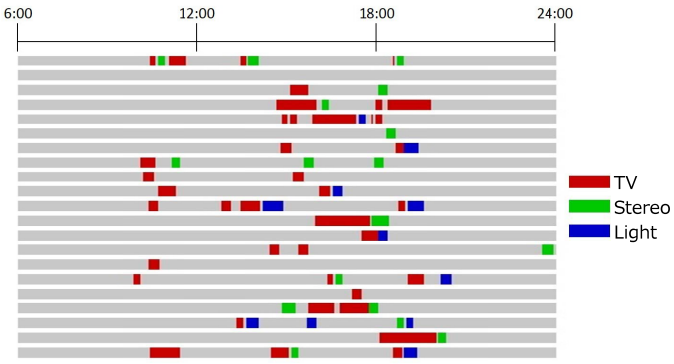


Fig. 5. *LongStay* recorded sequences. Each line corresponds to the record of a single day and colored (red, green, or blue) bars indicate time durations of using appliances.

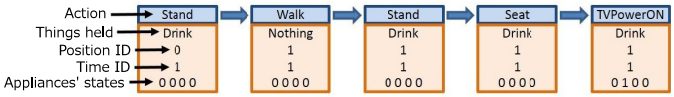


Fig. 6. An example sequence of action nodes. Appliances' states field shows ON(=1) or OFF(=0) states of Fan, TV, Stereo, and Light.

Two action nodes which happened in order within five-minute separation are considered to be consecutive. Fig. 6 shows an example sequence of action nodes.

B. Probabilistic action graph

Probabilistic action graph (PAG) is a directed graph representing all possible events and transitions obtained in the training data. Since a single action is not enough for predicting future, each node of the graph is composed of N consecutive action nodes. We use $N = 3$ in this paper. We compose each node by sliding time window of size N one by one; from an action node sequence of length m, therefore, we can get $m-2$ nodes for PAG. From the five-node action sequence shown in Fig. 6, for example, we have three PAG nodes: (Stand, Walk, Stand), (Walk, Stand, Seat), and (Stand, Seat, TVPowerON).

The edges of the PAG indicate possible transition and each edge has its transition probability. This probability is simply calculated based on the frequency assessment. The probability $P(T_{n_a \to n_b}|n_a)$ of transition $T_{n_a \to n_b}$ from n_a to n_b is given by:

$$P(T_{n_a \to n_b}|n_a) = \frac{|T_{n_a \to n_b}|}{|T_{n_a \to *}|}, \quad (1)$$

where $T_{n_a \to *}$ indicates any transition from n_a.

We would like to predict if some appliances will be operated in a near future. We thus calculate the probability for node n that a specific operation $O(a)$ of appliance a is performed within a certain time. To calculate this probability, we searches forward the subgraph starting at current node n_{curr} for node with a specific appliance operation $O(a)$ and calculate the summation of probabilities using the following expression (see Fig. 7):

$$P(O(a)) = P(O(a)|n_{curr}), \quad (2)$$

$$P(O(a)|n) = \begin{cases} 1 & (O(a) \text{ is performed at } n) \\ \sum_{n'} P(T_{n \to n'}) P(O(a)|n') & \\ & (O(a) \text{ is not performed at } n) \\ 0 & (\Delta_t(n_{curr}, n) > t_{max}) \end{cases}, \quad (3)$$

where $\Delta_t(n_1, n_2)$ is the time difference between two nodes and t_{max} is the time boundary (currently, $t_{max} = 5\,min$). This calculation is done in advance for every node and for every possible appliance operations.

Prediction and proactive operation of appliances using PAG is performed as follows:

1) Make the current PAG node from the latest three action nodes.
2) Search for the PAG node which is identical to the current PAG node.
3) If such an identical node does not exist, do nothing.

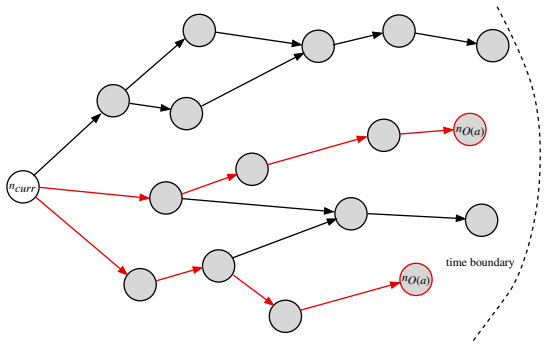
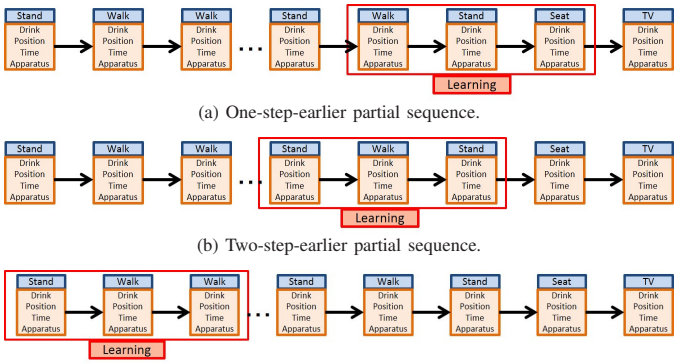


Fig. 7. Calculate the probability of a specific appliance operation. n_{curr} is the current node and $n_{O(a)}$ is node for performing operation $O(a)$.



(a) One-step-earlier partial sequence.

(b) Two-step-earlier partial sequence.

(c) n-step-earlier partial sequence.

Fig. 8. Positive training data.

If the node exists, and if the maximum value among the prediction probabilities exceeds a threshold, execute the corresponding appliance operation action automatically.

C. SVM-based classifier

The other method uses support vector machine [13] for identifying the case where appliance operations are desirable. Prediction using PAG is only effective for known (exactly experienced) situations. Using SVM, we expect to be able to cover similar but unknown situations.

The input to the classifier is a partial action sequence and the output is the likelihood of operating an appliance in a near future. The length of the partial sequence is currently set to three. The feature vector is thus the aggregation of three consecutive action node. The number of nodes for which each object is held is also added to the feature vector in order to cope with occasional object recognition failures.

Training data are collected from the recorded action data as follows. For positive data, we first locate an action node which does some appliance operation, and then trace back the action sequences to extract partial action sequences to be used as inputs. How long to trace back depends on how early the prediction is performed (see Fig. 8). In this paper, we consider $n = 1$ and $n = 2$ cases. For negative data, we pick up three-action partial sequences within a certain time from which no appliance operations happened.

We use the multi-class classification with probability output functionality of libSVM [14] for implementation. Prediction and proactive operation of appliance using SVM is performed as follows:

1) Extract latest three action nodes.
2) Generate an input vector to the SVM from the nodes.
3) If the highest probability output is above a threshold, execute the corresponding appliance operation action automatically.

IV. Experiments

A. Predicting turn-on operations

Performance of prediction and proactive execution of appliance operations is evaluated. We here deal with only turn-on operations of TV, Stereo, and Light. We do not deal with turn-off and other operations (e.g., change the program, adjust the volume) because these operations happen due to various reasons which mainly depend on the person's mental state (e.g., feels that the sound is a little bit large) or the end of some work or program (some TV program has ended and the person has next to do), and predicting such operations only from the visible actions/objects are still difficult.

B. Off-line experiments

We compare the two prediction methods using recorded or generated data in terms of prediction performance. For the SVM method, we further examine two cases, $n = 1$ and $n = 2$, separately (called SVM1 and SVM2, respectively).

We test the methods for all datasets, that is, $LunchStay1$, $LunchStay2$, and $LongStay$ using cross-validation. The evaluation factors are precision, recall, and error rate, defined as follows:

$$\begin{aligned} Precision &= \frac{N_{TP}}{N_{TP}+N_{FP}}, \\ Recall &= \frac{N_{TP}}{N_{TP}+N_{TN}}, \\ ErrorRate &= \frac{N_{FP}}{N_{PAS}}, \end{aligned} \tag{4}$$

where N_{TP}, N_{TN}, and N_{FP} are the number of true positives, that of true negatives, and that of false positives, respectively. N_{PAS} is the number of occurrences of partial action sequences which are used for predicting false positives. An error rate indicates the probability that the system erroneously operates an appliance when it is not desired.

In the evaluation, we change the thresholds for accepting the predictions and calculate the above factors to plot precision-recall and recall-error rate curves. These curves can clearly illustrate the performance comparison results.

Figs. 9 shows the recall-precision curve for *LongStay* data (see Fig. 5). We performed 21-fold cross-validation (i.e., data for one day are used for testing and the rest is for training). PAG can predict the event with about 60% recall and 70% precision in total. Both SVM1 and SVM2 can predict with about 80% recall with similar precisions, which is much better than PAG. Both TV and Compo Stereo are operated from the sofa and the subject watches TV more often than Stereo in this data set; this seems to be the reason of a high prediction rate for TV and a low for Stereo. Prediction of LED Light

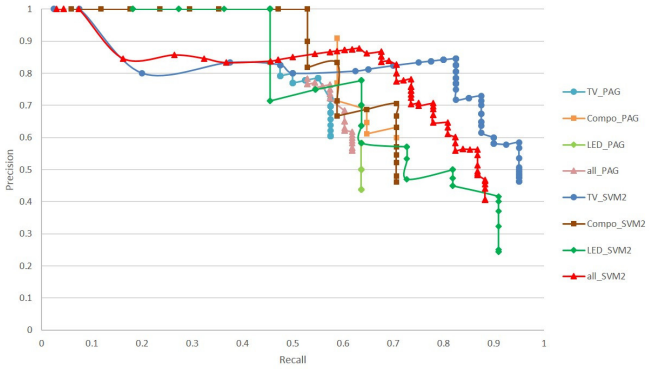


Fig. 9. Recall precision curve for PAG and SVM2 for *LongStay* data. The meanings of symbols are as follows:
TV_PAG: Predict turn-on TV operation using PAG.
Compo_PAG: Predict turn-on compo Stereo operation using PAG.
LED_PAG: Predict turn-on LED Light operation using PAG.
all_PAG: Predict turn-on operation of all three appliances using PAG.
TV_SVM2: Predict turn-on TV operation using SVM2.
Compo_SVM2: Predict turn-on compo Stereo operation using SVM2.
LED_SVM2: Predict turn-on LED Light operation using SVM2.
all_SVM2: Predict turn-on operation of all three appliances using SVM2.

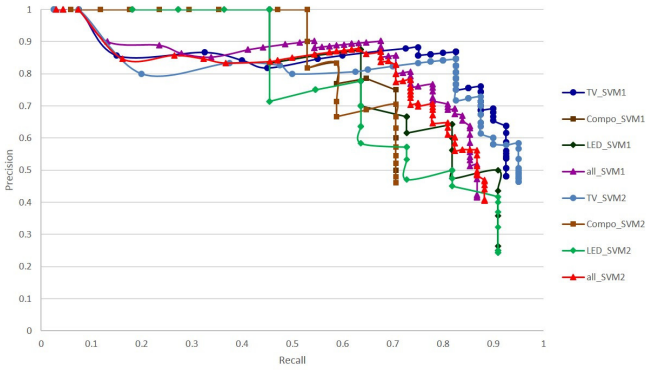


Fig. 10. Recall precision curve for SVM1 and SVM2 for *LongStay* data. Meanings of the symbols are the same as Fig. 9.

is not very good; this seems due to a small number of LED Lgiht operations in the training data.

Fig. 11 shows the earliness distribution of predictions. SVM2 and PAG can predict events more earlier than SVM1.

Fig. 12 shows the recall-error rate curve for PAG and SVM2. SVM2 can keep about 0.01 error rate with about 70% recall. Since the average number of operations at a single rest is about 20, this error rate means one erroneous appliance operation in five stays; this is not low enough but could be affordable.

C. On-line experiments

We implemented an on-line appliance prediction and control system as a proof-of-concept. We used SVM1 and SVM2 prediction methods which showed better performances than PAG in off-line experiments. Fig. 13 shows the flow of the on-line system. It recognizes actions and objects, predicts future appliance operations, and controls appliances when predicted with a high probability.

Figs. 14 and 15 show example operations of the system using SVM1 and SVM2 prediction methods, respectively. The bars on the left show the probability that each appliance will be operated in a near future. If this probability rises

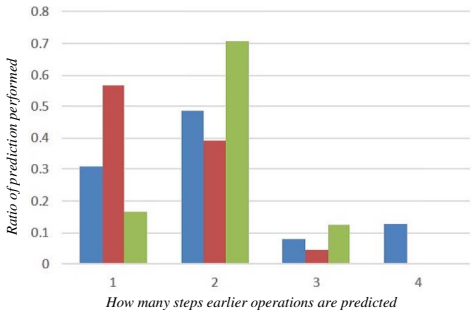


Fig. 11. Histogram of earliness of prediction for *LongStay* data.

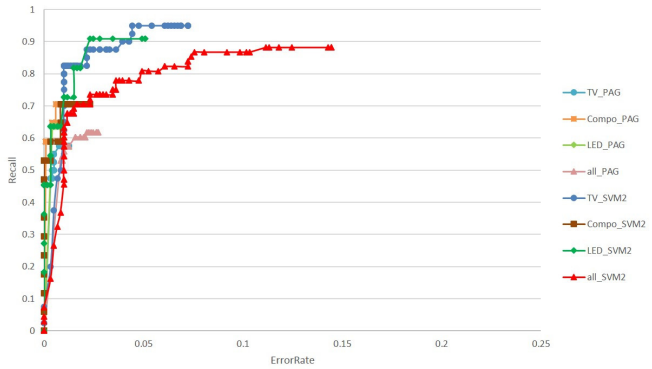


Fig. 12. Recall error-rate curve for PAG and SVM2 for *LongStay* data. Meaning of the symbols are the same as Fig. 9.

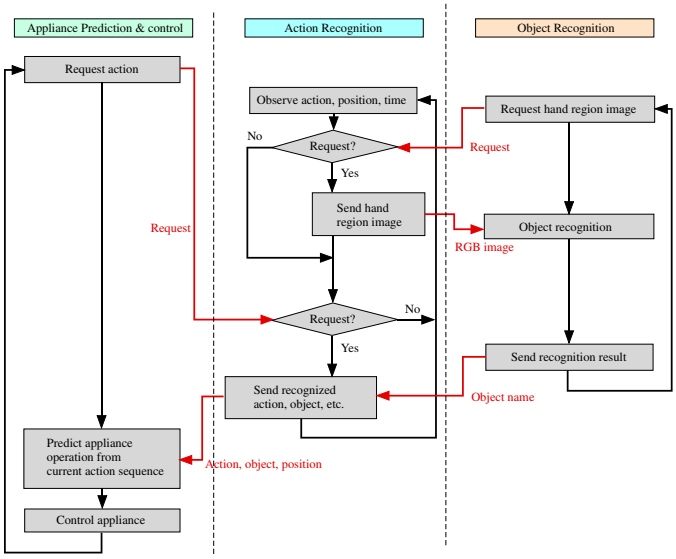
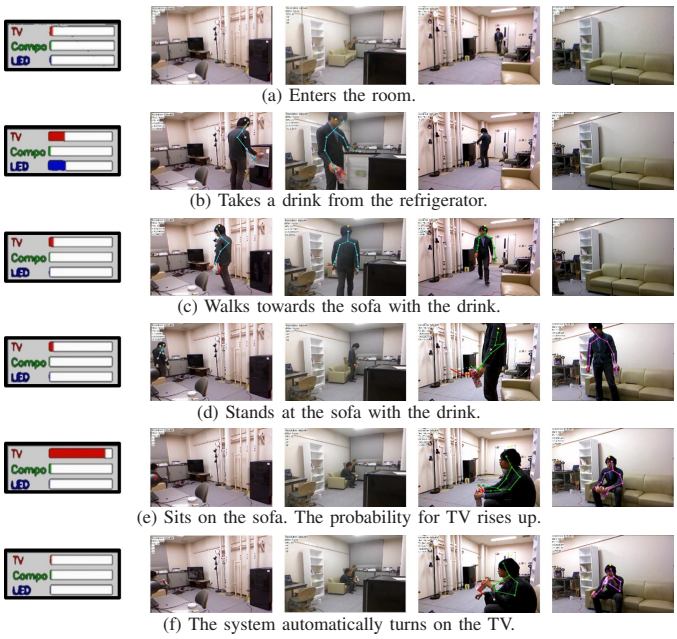


Fig. 13. Flow of the appliance control system.

up above the threshold, the corresponding appliance is turned on. Comparing SVM1 and SVM2, the latter can predict the operation earlier, thereby being able to turn on the TV before the person actually sits on the sofa.

V. Conclusions and Future Work

This paper has described a method of modeling human activities and predicting them for controlling appliances. We first developed an action recognition system using Kinects in an



(a) Enters the room.

(b) Takes a drink from the refrigerator.

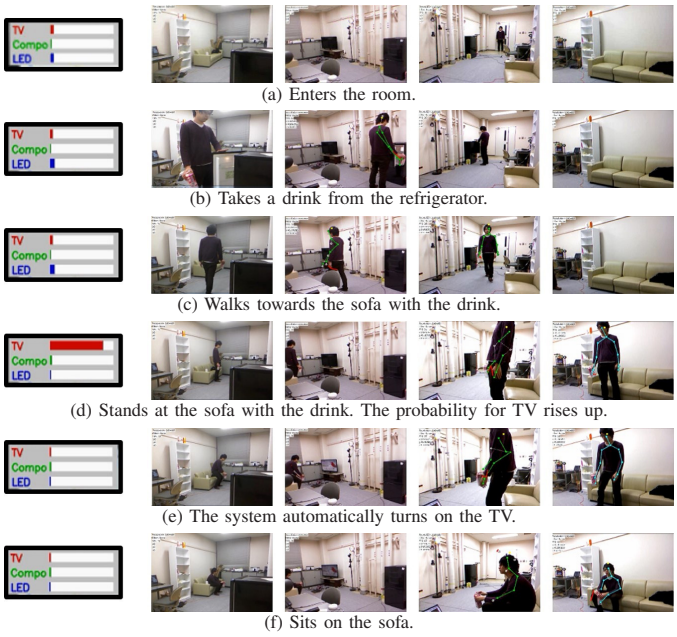
(c) Walks towards the sofa with the drink.

(d) Stands at the sofa with the drink.

(e) Sits on the sofa. The probability for TV rises up.

(f) The system automatically turns on the TV.

Fig. 14. On-line experiment with SVM1. The bars on the left shows the probability that each appliance will be operated in a near future.



(a) Enters the room.

(b) Takes a drink from the refrigerator.

(c) Walks towards the sofa with the drink.

(d) Stands at the sofa with the drink. The probability for TV rises up.

(e) The system automatically turns on the TV.

(f) Sits on the sofa.

Fig. 15. On-line experiment with SVM2. The system turns on the TV one-step earlier compared to SVM1, just before the person's sitting the sofa.

experimental room. Recorded actions sequences are compiled into two prediction models: a probabilistic action graph-based and an SVM-based. We evaluate the methods using actual observation data for over fifty days in total. Focusing not on a general action modeling but on prediction of appliance operations in a near future, the proposed methods, especially the SVM-based one, exhibit a high prediction performance. We have also implemented an on-line appliance control system as a proof-of-concept.

The current prediction methods rely on preceding actions to appliance operations, such as sitting with a drink. It is, therefore, difficult to predict operations without such preceding actions. For example, people usually turn off TV when they decide (in mind) to do so, and this is hard to predict only from their visible actions. Instead, most turning-off operations could well be handled by recognizing the human state/will not to continue to use appliances, such as leaving a sofa or getting into sleep.

We have tested the method for four appliances in one room. In actual home situations, however, a more variety of appliances and residential environments must be dealt with. Extending a set of objects to be recognized and improving action recognition performances are important future work. Experimental evaluations in real situation for a long period will also be mandatory for future deployment of such systems.

REFERENCES

[1] M. Bennewitz, W. Burgard, and S. Thrun. Learning Motion Patterns of People for Compiant Robot Motion. *Int. J. of Robotics Research*, Vol. 24, No. 1, pp. 38–48, 2005.

[2] D. Vasquez, T. Fraichard, and C. Laugier. Incremental Learning of Statistical Motion Patterns with Growing Hidden Markov Models. *IEEE Trans. on Intelligent Transportation Systems*, Vol. 10, No. 3, pp. 403–416, 2009.

[3] K. Kim, D. Lee, and I. Essa. Gaussian Process Regression Flow for Analysis of Motion Trajectories. In *Proceedings of 13th Int. Conf. on Computer Vision*, pp. 1164–1171, 2011.

[4] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models for Human Motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 2, pp. 283–298, 2008.

[5] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, Vol. 28, No. 6, pp. 976–990, 2010.

[6] M. Tenorth, F. de la Torre, and M. Beetz. Learning Probability Distributions over Partially-Ordered Human Everyday Activities. In *Proceedings of 2013 IEEE Int. Conf. on Robotics and Automation*, pp. 4539–4544, 2013.

[7] H.S. Koppula, R. Gupta, and A. Saxena. Learning Human Activities and Object Affordances from RGB-D Videos. *Int. J. of Robotics Research*, Vol. 32, No. 8, pp. 951–970, 2013.

[8] T. Mori, S. Tominaga, H. Noguchi, M. Shimoasaka, R. Fukui, and T. Sato. Behavior Prediction from Trajectories in a House by Estimating Transition Model Using Stay Points. In *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3419–3425, 2011.

[9] C. Rudin, B. Letham, A. Saleb-Aouissi, E. Kogan, and D. Madigan. Sequential Event Prediction with Association Rules. In *Proceedings of 24th Annual Conf. on Learning Theory*, 2011.

[10] H. Hoai and F. De la Torre. Max-Margin Early Event Detector. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2863–2870, 2012.

[11] DeepBeliefSDK. https://github.com/jetpacapp/DeepBeliefSDK/.

[12] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.

[13] V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

[14] C.-C. Chang and C.-J. Lin. LIVSVM: a library for support vector machines. *ACM Trans. on Intelligent Systems and Technology*, Vol. 2, No. 3, 2011. Artcile 27.