Research
Robotics—Article

Engineering 2015, 1(1): 79–84
DOI 10.15302/J-ENG-2015024

Personalizing a Service Robot by Learning Human Habits from Behavioral Footprints

Kun Li[1] and Max Q.-H. Meng[2*]

ABSTRACT For a domestic personal robot, personalized services are as important as predesigned tasks, because the robot needs to adjust the home state based on the operator's habits. An operator's habits are composed of cues, behaviors, and rewards. This article introduces behavioral footprints to describe the operator's behaviors in a house, and applies the inverse reinforcement learning technique to extract the operator's habits, represented by a reward function. We implemented the proposed approach with a mobile robot on indoor temperature adjustment, and compared this approach with a baseline method that recorded all the cues and behaviors of the operator. The result shows that the proposed approach allows the robot to reveal the operator's habits accurately and adjust the environment state accordingly.

KEYWORDS personalized robot, habit learning, behavioral footprints

1 Introduction

Traditionally, a personal robot is designed to provide standard services in different scenarios. For example, by incorporating a door recognition and manipulation algorithm, the robot can open various kinds of doors in different houses in exactly the same way. This strategy, combined with commands from the operator, allows the robot to complete each task consistently in different environments. This feature is desirable when the robot is used in fixed and repeating scenarios, but if the operator requires personalized services, this strategy is inadequate.

The requirement for personalization is particularly evident in a smart home, where the robot needs to both monitor and adjust the home state intelligently. For example, the robot may need to open a door to different extents, as some operators like it to be fully open, while others may prefer it to be half open. This kind of state adjustment, if designed in an offline way, requires a remarkable amount of manual work. To solve the problem, the robot must be personalized by having it learn the habits of the operator, in order to adjust itself according to the habit of each operator.

To learn a habit, the robot needs to observe the environment and extract information related to the habit. A habit is determined by three factors: the cue, the behavior, and the reward [1]. After sufficient experiences with the three factors, the operator behaves involuntarily when seeing the cue, instead of acting intentionally to collect the maximum reward. For a robot to understand the operator's habit, it may collect all pairs of cues and behaviors from the observations to guide its future actions, or it may try to learn the rewards based on the observations in order to determine its future actions. The first solution is straightforward, because the robot can iterate its memory to find the best-matching behavior when it faces a cue, but it is inefficient in dealing with newly emerging cues; the second solution requires an additional learning process, but the learned reward can guide the robot's action when new cues occur. In this work, the first solution is implemented as the baseline method, and we focus on the second solution.

In this article, we propose a method to learn the habit of an operator based on observations, in the framework of inverse reinforcement learning. The behavior is described by the environment state changes due to the behaviors, namely the behavioral footprints. Meanwhile, the robot observes the cues based on the contacts between the operator and the objects, and learns the habits as a reward function based on the operator's behaviors in the house. After that, it uses the reward function to guide its future actions, in order to serve the operator autonomously. This method is implemented with a case study of autonomous indoor temperatures adjustment. Our contributions include the incorporation of behavioral footprints to represent the operator's behaviors and a proposal of robot personalization based on the operator's habits.

2 Related work

Traditional research on personal robots focuses on designing

[1]California Institute of Technology, Pasadena, CA 91125, USA; [2] The Chinese University of Hong Kong, Hong Kong, China
* Correspondence author. E-mail: max@ee.cuhk.edu.hk
Received 26 March 2015; received in revised form 29 March 2015; accepted 30 March 2015

hardware and software to make each robot generally applicable. For example, Meeussen et al. [2] develop a personal robot to open the door and charge itself. Rusu et al. [3] develop a perception system with visual sensors to guide the robot's motion in different environments. Gorostiza et al. [4] use multiple sensors to develop a framework for human-robot interaction. Wyrobek et al. [5] develop a personal robot that is both safe and useful. In a domestic environment, Falcone et al. [6] develop a personal rover that can serve both children and adults.

Many publications cover the putting of a personal robot in a house. For example, in Ref. [7], an electroencephalography signal is used to control a tele-presence robot and assist motor-disabled people. In Ref. [8], a tele-presence robot is designed to help the elderly with interpersonal communications. In Ref. [9], a tele-medicine system is designed to monitor the health and activity of the elderly. To include robot actions during home monitoring, in Ref. [10] the service robots use sensor networking and radio frequency identification to guide their actions.

With different types of sensors installed in a house, the environment state can be described using hierarchical states, and its changes can be described with a layered hidden Markov model [11], where multiple layers of hidden Markov model are stacked to describe the hierarchical state transition; and a hierarchical hidden Markov model [12], where each state of the higher layer incorporates a hidden Markov model in the lower layer.

To personalize the robot's service, the robot needs to learn the operator's habits from observation. To combine robot actions and environment state modeling, many methods have been proposed within the framework of reinforcement learning [13]. Besides, learning by demonstration technique [14] allows a robot to imitate an operator and learn different behaviors. In our applications, the robot can observe the behavior of an operator; thus it adopts inverse reinforcement learning [15] to encode the operator's habits.

In this work, we use inverse reinforcement learning to enable a robot to learn a reward function as the operator's habit. During learning, the operator's behaviors are represented with behavioral footprints, and after collecting a set of observations on these behaviors, the robot tries to learn the operator's habits.

3 Methods

3.1 Behavioral footprints

A habit is determined by three parts: the cue, the behavior, and the reward. To learn the operator's habit, the robot must observe the environment to obtain the cues, and observe the operator to get the behaviors; thus it can learn the reward function to describe the operator's habits. For this purpose, the robot needs to represent the environment accurately, and in this work, we use the objects inside a room to describe the home state:

$$E = (C_1, ..., C_n)$$

where E denotes the environment states and C_i (i = 1, ..., n) denotes the ith object in the environment. An illustration is shown in Figure 1.

To represent the operator's behaviors, A, we adopt behavioral footprints, defined as the changes of object states due to the operator's actions, because this representation can describe different types of behaviors more meaningfully, and exclude those behaviors that do not change the environment states:

$$A = (E_i, E_j)$$

where E_i and E_j denote the transition of home states due to the operator's behaviors. An illustration is shown in Figure 2.

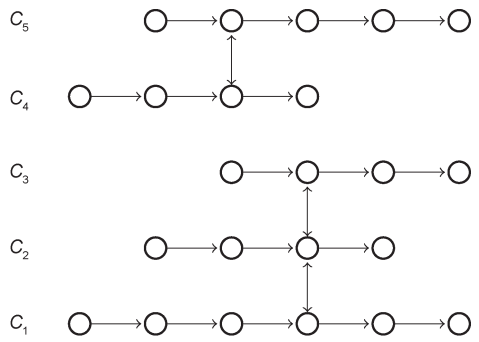

Figure 1. An environment state is composed of multiple chains of object states, and these object states may be asynchronous and uncorrelated.

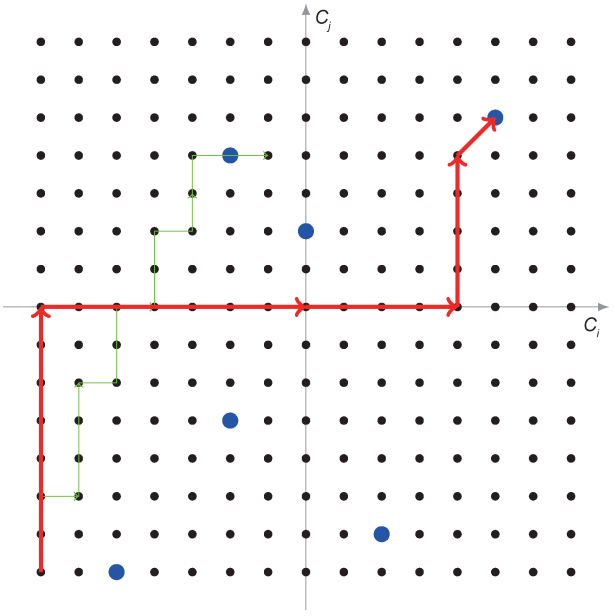

Figure 2. The home state can be represented as a point in the state space, and it evolves naturally, as shown by the thin green trajectory. However, the operator may manually change it into some desired states, represented by the thick blue points, and these actions lead to the home state change, represented by the thick red trajectory.

3.2 Cues and behaviors

With the behavioral footprints, the robot can observe the operator's behaviors, along with the cues that trigger the behaviors.

The behaviors are represented by changes in object states due to the operator's contact. However, some of the behaviors are random, and do not follow the operator's habits, and these need to be excluded. To evaluate the regularity of the operator's behaviors, we use the following measurements:

$$r = r(A)$$

where r measures the standard deviation of the cues leading to behavior A. With the measured regularity level and a

threshold value selected based on experiments, the robot only keeps the samples with regular behaviors.

Another important factor of a habit is the cue, defined as the environment state when the behaviors occur. The cue is identified by collecting data samples right before the operator's emergence.

$$S = [D_1(S_{t1}, ..., S_{tj}), ..., D_m(S_{tk}, ..., S_{tn})]$$

where D_i, $i = 1, ..., m$ denotes the moment when the operator appears, and each $(S_{tk}, ..., S_{tn})$ denotes a set of home states following the operator's emergence.

Two types of cues exist: agreeable ones, where the operator does not change the environment states, and disagreeable ones, where the operator manually changes certain object states. Based on the observations, the samples are assigned with binary indicators of agreeability:

$$R = [R_1, ..., R_n]$$

3.3 Rewards

Using the samples of the operator's regular behaviors and the binary indicators of the environment's agreeability, the robot infers the operator's habits. This problem is formulated as inverse reinforcement learning, where the robot learns a reward function by observing the operator's actions [16]:

$$\max \sum_{s \in S} \min_{\alpha \in \{A\}} \{p(E_{s' \in P_{s\alpha_1}}[V^{\pi}(s')] - E_{s' \in P_{s\alpha}}[V^{\pi}(s')])\}$$
$$\text{s.t.} |\alpha_i| \leqslant 1, \ i = 1, ..., d \tag{1}$$

where α denotes the parameter of the reward function, and

$$V^{\pi}(s_0) = E_{\pi}(\sum_{t=0}^{\infty} \gamma^t R(s_t)) \tag{2}$$

denotes the expected discounted reward under a policy. An illustration is shown in Figure 3.

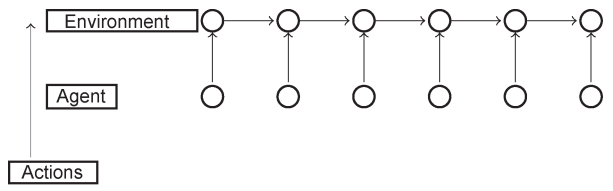

Figure 3. Inverse reinforcement learning intends to reveal the reward function based on the optimal action policy.

This optimization maximizes the differences between the operator's actions and other actions, allowing the robot to learn the operator's habits.

With only binary indicators of the environment states' agreeability, the maximization in Eq. (1) is simplified as:

$$\max \min \{E_{s' \in P_{s\alpha_1}}[V^{\pi}(s')] - E_{s' \in P_{s\alpha_2}}[V^{\pi}(s')]\}$$
$$\text{s.t.} |\alpha_i| \leqslant 1, \ i = 1, ..., d \tag{3}$$

where α_1 denotes the actions agreeable with the operator's habits, and α_2 denotes the actions disagreeable with the operator's habits. The agreeability is measured with the binary indicators.

With Eq. (3), the robot learns a reward function, a function of the environment states:

$$R_t = \phi(S_t)$$

The learning of the reward function is based on the formulation in Ref. [15], where the reward function is a linear combination of a set of predesigned basis functions:

$$R_t = \omega_1 \phi_1(S_t) + ... + \omega_n \phi_P(S_t) \tag{4}$$

and ϕ_i is a basis function.

In a personalized environment, the reward function must encode potential changes of environment states due to the appearances and disappearances of the objects inside the environment. With behavioral footprints, this problem is solved by clustering the state space dimensions into multiple abstracted dimensions, with the correlations between different dimensions as the distances:

$$(cst_1, ..., cst_n) = partition\ (S, RLT)$$

The clustering not only excludes redundant information due to object state correlations, but also reveals invisible state transitions. In addition, it avoids having the basis functions redesign when the objects' number changes, because only an object uncorrelated with all existing dimensions requires redesigned basis functions. Besides, this clustering allows the robot to use one action to change the states of all related objects.

Based on the dimension clustering, each basis function records one combination of cluster states:

$$F_i = \phi_i(cst_1, ..., cst_n)$$

Substituting the basis function into Eq. (4), the reward function is:

$$R(S_t) = \omega \cdot \phi(S_t) \tag{5}$$

where $\omega = [\omega_1, ..., \omega_n]$ and $\phi = [\phi_1, ..., \phi_p]$.

Substituting Eq. (5) into Eq. (2):

$$V_{\pi}(S_0) = \omega \cdot E_{\pi}(\sum_{t=0}^{\infty} \gamma^t \phi(S_t))$$

With Eq. (2), Eq. (3) is simplified as:

$$\max \min \omega \cdot (\mu_1 - \mu_2)$$
$$\text{s.t.} |\omega_i| \leqslant 1, \ i = 1, ..., d$$

where $\mu_i = E_{\pi_i}[\sum_{t=0}^{\infty} \gamma^t \phi(S_t)]$, describing the expected reward under the ith action policy.

Inspired by the work in Ref. [15], we transform this maximization into an optimization similar to the Support Vector Machine (SVM):

$$\max_{t, \omega} t$$
$$\text{s.t.} \ \omega \cdot \mu_1 \geqslant \omega \cdot \mu_2 + t,$$
$$\|\omega\|_2 \leqslant 1$$

This optimization is solved with an existing SVM implementation [18].

3.4 Robot actions

Using the learned reward function to indicate the operator's

habits, the robot can guide its actions as a normal reinforcement learning problem.

4 Experiments and results

4.1 Setup

We use Turtlebot as the personalized robot to observe the behaviors of people in an environment composed of four outdoor states and four indoor states. The four outdoor states include outside temperature, humidity, wind, and rain; and the four indoor states include a thermometer, a door, air conditioner switches, and the state of the operator. To observe the indoor objects accurately, a map is built with a Gmapping package [17] in the robot operating system. After collecting the states for about seven days, the robot tries to learn the habit and use the habit to guide future actions.

Our robot is not equipped with a robot hand to physically change the object states, so the robot actions are simulated.

4.2 Experiments

4.2.1 Habit observation

Four weather conditions are observed, including the temperature, humidity, rain, and wind, which are extracted from a weather website (www.weather.com). These environment states are collected for seven days for the city of Hong Kong, as shown in Figure 4.

Four indoor objects are observed, including a thermometer, a door, the air conditioner switch, and the status of the operator in a house. The states of these objects are measured by the robot based on their visual appearances, as shown in Figure 5.

4.2.2 Habit learning

Based on the observations, the robot collects the operator's behaviors and the cues leading to the behaviors. The cues are collected as the environment states when the operator has contact with the objects. For example, when the operator enters the room and turns on the air conditioner, the current environment states are collected as the cue that leads to changes to the air conditioner switches.

The behaviors are collected as the changes of environment states due to the operator's actions, such as the switching of the air conditioner, the opening of the door, and so on.

After collecting cues and behaviors for seven days, the robot uses them to learn the operator's habit and to update the result based on new observations. This habit is represented with the reward function.

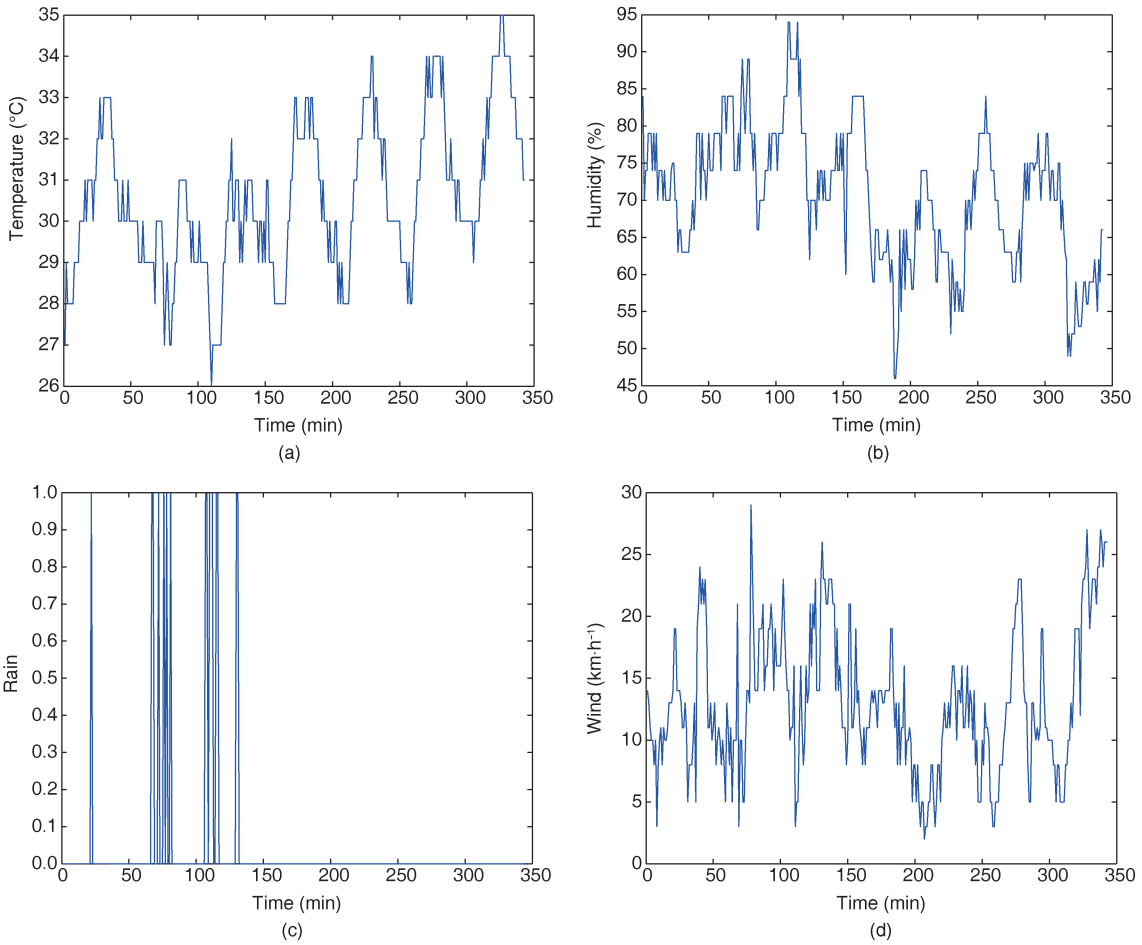

Figure 4. The weather condition has four components, including temperature, wind, humidity, and rain. The figures show the samples collected from 25 July to 31 July at an interval of 30 min. (a) The temperature outside, 1 sample every 30 min; (b) the humidity outside, 1 sample every 30 min; (c) the rain outside, 1 sample every 30 min; (d) the wind outside, 1 sample every 30 min.

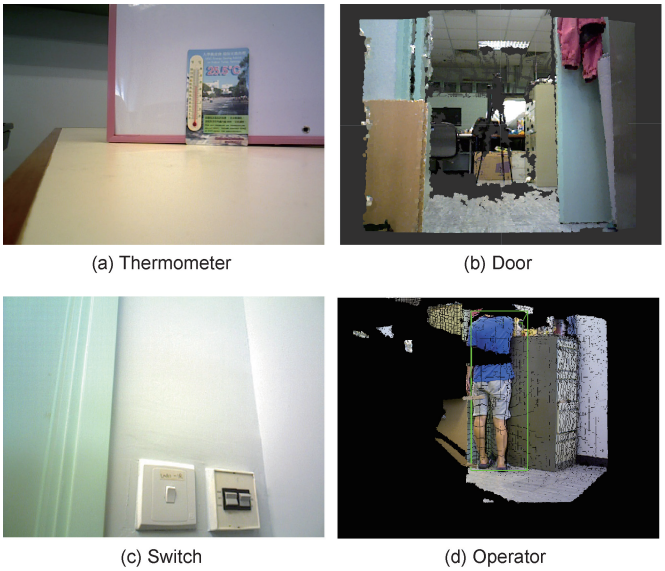

Figure 5. The four indoor object states are detected visually. The robot collects these object states periodically to monitor the home states.

4.2.3 Robot actions

With the learned reward function, the robot searches for the optimal actions to adjust the environment. In this work, the generated actions are applied manually to evaluate their effects.

4.3 Results

After collecting observations and learning the operator's habits for one week, the robot extracts a set of reward functions, corresponding to increasing samples. To evaluate these learned reward functions, two indexes are adopted, including the accuracy of reward function r_A, computed by comparing the robot's evaluation of the home states on agreeability and the true values provided by the operator, and the accuracy of robot action r_D, indicated by the ratio of disagreement on actions between the robot and the operator.

Two sets of experiments are conducted, corresponding to different numbers of objects in the environment, as shown in Figures 6 and 7. In each set of experiments, both the baseline method and the proposed method are implemented, with evaluation based on r_A and r_D. The results are shown in Figures 6 and 7.

The results show that the two methods have similar accuracy in evaluating home states, but the proposed method is much more accurate in guiding the robot's actions. The reason is that in a new state, a robot using the baseline method has to search in the records. However, if the action-cue pair is not in the record, the baseline method will not be able to find a correct strategy. By learning the reward function, the proposed method can generate different actions according to the environment states.

5 Conclusions

In this article, we propose a method to enable a robot to learn the habit of an operator based on observations, in the framework of inverse reinforcement learning. The behavior is described by the environment state changes due to the behaviors, namely the behavioral footprints. The robot learns the cue based on the contact between the operator and the objects, and learns the habits as a reward function based on the operator's behaviors in the house. After that, it uses the reward function to guide its future actions, in order to serve the operator autonomously. This work concentrates on the robot learning how to adjust indoor temperatures, and compares the proposed method with a baseline method on home state evaluation and robot action selection. The results show that the proposed method is more accurate in guiding the robot's actions in complicated scenarios.

In future work, the proposed method can be improved in multiple aspects. First, the basis function can be designed more flexibly, in order to analytically describe the change

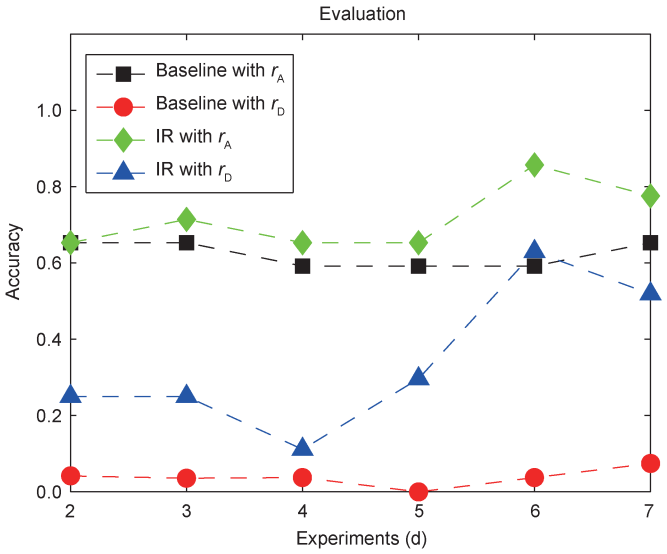

Figure 6. The robot observes the weather conditions, the air conditioner switch, and the door, and learns the operator's habits, in order to act on the switches and the door to adjust the environment states. "IR" denotes inverse reinforcement learning, r_A denotes the accuracy of reward functions, and r_D denotes the accuracy of robot actions.

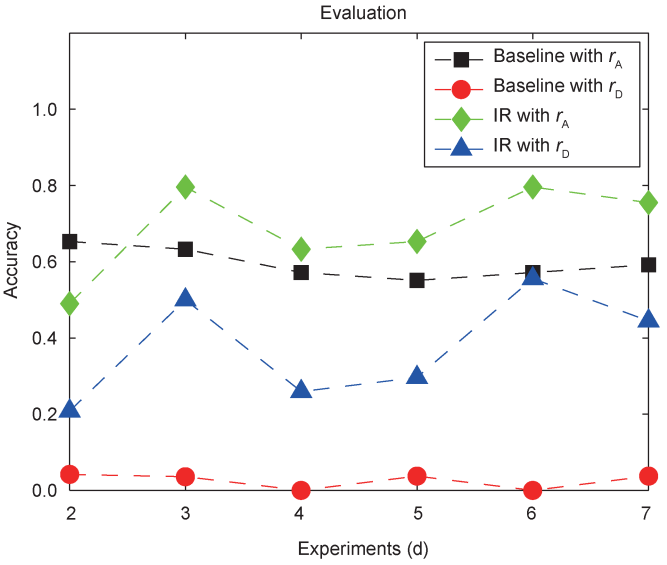

Figure 7. The robot observes the weather conditions, the air conditioner switches, the door, and the thermometer, and learns the operator's habits, in order to act on the switches and the door to adjust the environment states.

of environment states. The learning method can also be improved to cover different types of habits, in addition to the one represented by a set of basis functions.

Acknowledgements

This project was supported in part by Hong Kong RGC GRC (CUHK14205914 and CUHK415512), awarded to Max Q.-H. Meng.

Compliance with ethics guidelines

Kun Li and Max Q.-H. Meng declare that they have no conflict of interest or financial conflicts to disclose.

References

1. W. Wood, D. T. Neal. A new look at habits and the habit-goal interface. *Psychol. Rev.*, 2007, 114(4): 843–863
2. W. Meeussen, et al. Autonomous door opening and plugging in with a personal robot. In: *Proceedings of 2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010: 729–736
3. R. B. Rusu, I. A. Sucan, B. P. Gerkey, S. Chitta, M. Beetz, L. E. Kavraki. Real-time perception-guided motion planning for a personal robot. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009: 4245–4252
4. J. F. Gorostiza, et al. Multimodal human-robot interaction framework for a personal robot. In: *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication*, 2006: 39–44
5. K. A. Wyrobek, E. H. Berger, H. F. M. Van der Loos, J. K. Salisbury. Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 2008: 2165–2170
6. E. Falcone, R. Gockley, E. Porter, I. Nourbakhsh. The personal rover project: The comprehensive design of a domestic personal robot. *Robot. Auton. Syst.*, 2003, 42(3–4): 245–258
7. L. Tonin, T. Carlson, R. Leeb, J. del R. Millán. Brain-controlled telepresence robot by motor-disabled people. In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011: 4227–4230
8. T. C. Tsai, Y. L. Hsu, A. I. Ma, T. King, C. H. Wu. Developing a telepresence robot for interpersonal communication with the elderly in a home environment. *Telemed. J. E Health*, 2007, 13(4): 407–424
9. P. R. Liu, M. Q. H. Meng, P. X. Liu, F. F. L. Tong, X. J. Chen. A telemedicine system for remote health and activity monitoring for the elderly. *Telemed. J. E Health*, 2006, 12(6): 622–631
10. M. Baeg, J. H. Park, J. Koh, K. W. Park, M. H. Baeg. Building a smart home environment for service robots based on RFID and sensor networks. In: *Proceedings of International Conference on Control, Automation and Systems*, 2007: 1078–1082
11. N. Oliver, A. Garg, E. Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Comput. Vis. Image Underst.*, 2004, 96(2): 163–180
12. S. Fine, Y. Singer, N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Mach. Learn.*, 1998, 32(1): 41–62
13. R. S. Sutton, A. G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998
14. B. D. Argall, S. Chernova, M. Veloso, B. Browning. A survey of robot learning from demonstration. *Robot. Auton. Syst.*, 2009, 57(5): 469–483
15. P. Abbeel, A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In: *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004: 1
16. A. Y. Ng, S. J. Russell. Algorithms for inverse reinforcement learning. In: *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000: 663–670
17. G. Grisetti, C. Stachniss, W. Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans. Robot.*, 2007, 23(1): 34–46
18. T. Joachims. Making large-scale SVM learning practical. In: B. Schölkopf, C. J. C. Burges, A. J. Smola, eds. *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA: MIT Press, 1999: 169–184