Proc. 2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS-2018)
Madrid, Spain, Oct. 2018

Generating Adaptive Attending Behaviors using User State Classification and Deep Reinforcement Learning

Yoshiki Kohari, Jun Miura, and Shuji Oishi
Department of Computer Science and Engineering
Toyohashi University of Technology

Abstract— **This paper describes a method of generating attending behaviors adaptively to the user state. The method classifies the user state based on user information such as the relative position and the orientation. For each classified state, the method executes the corresponding policy for behavior generation, which has been trained using a deep reinforcement learning, namely DDPG (deep deterministic policy gradient). We use as a state space of DDPG a distance-transformed local map with person information, and define reward functions suitable for respective user states. We conducted attending experiments both in a simulated and a real environment to show the effectiveness of the proposed method.**

I. Introduction

Lifestyle support is one of the applications areas to which mobile robot technologies are applied. As many countries are facing aging/aged society, service robots are needed that support people for self-reliance. One promising example of services is attending. Going outside is a good practice for the elderly, but there exist many dangerous/inconvenient situations such as getting tired or sick, carrying heavy items, and losing the way, and an attendant robot has to provide services adaptively to each situation. Fig. 1 illustrates some example of robot's adaptive behaviors.

Person following is one of the fundamental functions of such robots, which is realized by combination of person detection and tracking [1], [2], [3] and dynamic path planning [4], [5]. In addition to this fundamental function, an attendant robot has to provide various behaviors depending on the state of the user to be attended. There are several works on adaptive behavior selection of robots that interact with people [6], [7], [8]. For such an adaptive behavior, both classifying the user state and generating an appropriate behavior are needed.

We have developed an attendant robot which adaptively switches its behaviors according to the classified user state [8]. The state classification is guided by a finite state machine-based state transition model and a sensor-based transition detection. This work dealt with a simple two-state case: walking and sitting. The subjective evaluation has shown that the adaptive behavior generation is favorable to users.

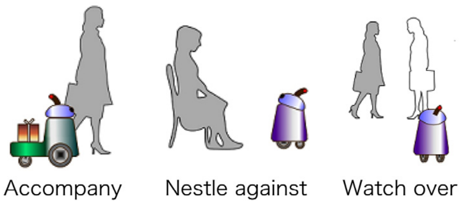


Fig. 1: Adaptive attending behaviors.

To cope with more realistic situations, we extend this approach in the following two points: (1) behavior generation becomes more general by using a deep reinforcement learning; (2) the number of user states is increased.

The rest of the paper is organized as follows. Section II describes related work. Section III explains the user state classification and its evaluation. Section IV explains a method of generating behaviors using a deep reinforcement learning. Section V describes the results of experimental evaluation. Section VI concludes the paper and discusses future work.

II. Related Work

A. Attendant robot

Many research efforts for realizing attendant robots focus on reliable person following. Various sensors are used in person detection and identification such as LIDARs (laser imaging detection and ranging) [1], [9], images [10], depth cameras [11], [2], or their combinations [12], [3].

Path planning is also needed for realizing safe following behaviors. Real-time efficiency and avoidance of dynamic obstacles (usually other people) are two important points and sampling-based methods are suitable for this purpose [4], [5].

As mentioned above, an attending task is not a simple following but consists of various behaviors. Ardiyanto and Miura [13] proposed a unique method of generating a robot motion which does not necessarily come close to a target person as long as the robot does not miss the target. This approach can reduce the person's feeling annoying as well as the energy for robot movement. Oishi et al. [8] developed a robot that switches the behavior depending on the classified person state (walking and sitting, in this case). For realizing versatile attending behaviors, a robot has to reliably recognize the person's state and generate appropriate behaviors.

B. Human action recognition

Human behavior/action recognition has widely been studied in various contexts. Image-based methods use feature sequences such as optical flow for classifying actions/behaviors [14]. Recently CNN-based approaches are greatly increasing

[15], [16], [17], [18]. Datasets for evaluation have also been developed (e.g., KTH dataset [19] and UCF sports dataset [20]). Since the relationship between a person and a mobile robot may continuously change during providing services, recognition using a single camera could sometimes be difficult.

Depth images have also been used for action recognition, usually extracting joint position sequences [21], [22]; datasets for such approaches have also been developed [23], [24]. Although they show good results, applications are limited to indoor scenarios to get reliable depth images.

C. Attending behavior generation

Designing attending behavior is challenging because various factors have to be taken into account such as relationships between the robot and the target and other persons, geometrical configuration of the surrounding environment, and the target person's feelings. As mentioned above, many path planning and positioning methods have been proposed, but the resultant behaviors are basically reactive, that is, they are planned based on the current state of the environment.

For pursuing long-term optimality, reinforcement learning (RL) approaches are effective and many efforts have been made [25], [26]. Recently combination of RL and deep learning, that is, deep reinforcement learning has been very popular. For example, Mnih et al. [27] proposed Deep Q-Network which learns the Q-values using a deep neural network (DNN). Lillicrap et al. [28] apply DNN to Actor-Critic method [29] to propose Deep Deterministic Policy Gradient (Deep DPG) method, which can learn the policy for a continuous-valued control. To effectively utilize Deep RL approaches, selection of the state space and reward functions are crucial.

III. User State Classification

A. User states

We consider the case where a robot attends a user to go out. While going out, the user faces various situations, each of such situations is called *user state*. In this paper, we deal with the following four user states and develop a method of classifying them and generate corresponding robot behaviors.

walking
The user is walking freely. The robot follows the user with an appropriate relative positioning.

standing
The user is temporarily stopping during walking. The robot stands by the user with a similar positioning to the walking case.

sitting
The user is sitting on a chair (or something like that). The robot stands by with an appropriate relative positioning to the user.

talking
The user is standing while talking with other person(s). The robot also stands by with an appropriate relative positioning which does not bother the persons in talking.

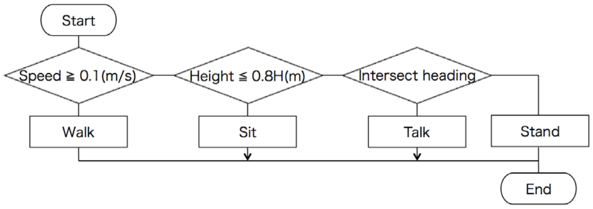
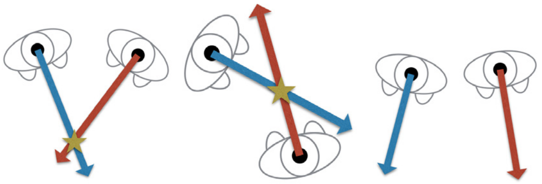


Fig. 2: Flow of user state classification.



(a) Talking. (b) Talking. (c) Not talking.

Fig. 3: Talk state classification examples.

B. User state classification

We use our person detection and identification method [3] and orientation estimation method [30] to obtain the position, the velocity, the height, and the orientation of each person. These values are analyzed to determine the user state.

Fig. 2 shows the flow of classification by a cascade of tests. The detailed classification steps are as follows:

- If the velocity of a person is larger than 0.1 [m/s] then the state is classified as *walking*.
- If the height is less than $0.55H$, then the state is classified as *sitting*. We currently use 1.71 [m] as H which is from a national statistics of Japan of men at twenties.
- Proxemics by Hall [31] suggests that persons who do not know with each other talk at the distance range of $1.2 \sim 2.0$ [m]. We thus search for other persons within the range of 2.0 [m] and, if exists, examine if the relative orientation vectors intersect or not (see Fig. 3). If yes, the state is classified as *talking*.
- If any of the above conditions do not hold, the state is classified as *standing*.

C. Evaluation of state classification

We constructed a dataset in our laboratory (Active Intelligent Systems Lab at Toyohashi Tech.). We used a web camera and a 2D LIDAR (Hokuyo UST-20LX) and took data for eight lab members, and extracted the position, the velocity, the height, and the orientation of each person at each frame. Person regions for orientation estimation are extracted by an image-based object detector [32].

The dataset contains 48,087 data: 9,618, 7,409, 12,198, and 18,862 for standing, sitting, walking, and talking, respectively. Fig. 4 shows example extracted person regions for the four user states. Table I summarizes the classification rate. The average rate is 86.1%.

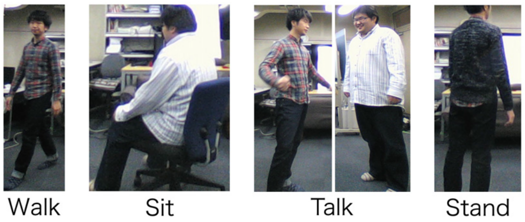


Fig. 4: Example scenes in the dataset.

TABLE I: State classification accuracy by the proposed method.

State	# of data	# of correct classifications	Accuracy[%]
Stand	9618	8317	86.5
Sit	7409	7165	96.7
Walk	12198	11553	94.7
Talk	18862	14364	76.2
Total	48087	41399	86.1

IV. Behavior Generation using Deep Reinforcement Learning

We use Deep Deterministic Policy Gradient (DDPG) algorithm [28] for generating attending behaviors. DDPG uses neural networks for representing Actor and Critic. Lillicrap et al. applied DDPG to a vehicle control task using the TORCS simulator [33]. The input is the frontal image from the vehicle and the outputs are the acceleration, the steering, and the braking controls, and DDPG was able to learn the control policy for driving.

In the case of our attendant robot, using only images as inputs does not suffice because geometrical relationships among the robot, the target person, other persons, and obstacles are very important and must be considered. Therefore, we propose to use a local map with person information as inputs (i.e., state space). Defining appropriate reward functions is also important in training. We define a different set of functions for training for each user state.

A. State space

We suppose that the robot has omnidirectional 2D LIDARs which can cover a 360 [deg] field of view. The data from the LIDARs come in as scan of range data. Since that type of data is not suitable for extracting features in a 2D coordinate system, we make a 2D local map by placing each range measurement in the 2D space. The size of local map is set to 2.5 [m] × 2.5 [m].

We also include the person information in the state space. Using the position and the velocity of each detected person, we calculate a region where that person occupies for a certain time period from the current time and draw it as a virtual obstacle both in the target person map and in the other persons map. Figs. 5 and 6 show an example situation and the corresponding maps, respectively.

These maps are binary and may not be appropriate as inputs to convolutional filters. We thus apply a distance transformation (see, for example, [34]) to the maps and the distance to obstacle cells normalized by the width of the image is recorded at each pixel. We call this map a *local distance map*. Fig. 7 shows the three local distance maps generated from the local maps in Fig. 6. The effectiveness of this representation will be validated in Sec. V-A in comparison with others.

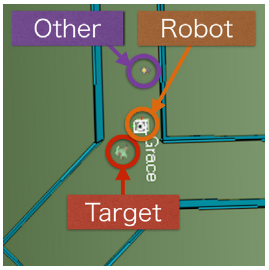
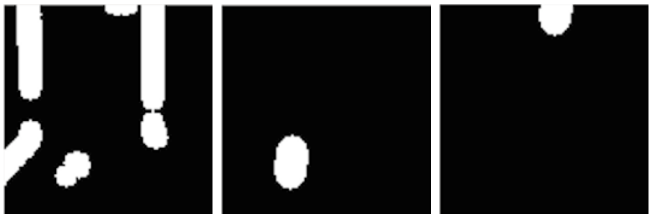
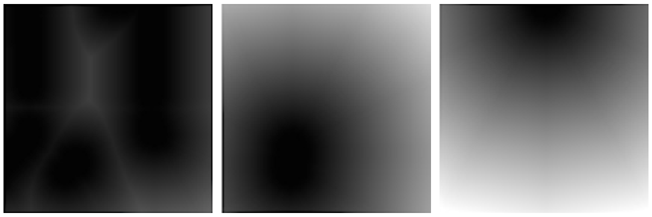


Fig. 5: Example situation.



(a) Obstacles. (b) Target person. (c) Other persons.

Fig. 6: Local maps for the situation shown in Fig. 5.



(a) Obstacles. (b) Target person. (c) Other persons.

Fig. 7: Local distance maps corresponding to the ones shown in Fig. 6.

B. Network structure

Fig. 8 shows the network structure used in DDPG. The Actor network receives the local distance maps as inputs and outputs the translational velocity v [m/s] and the rotational velocity ω [rad/s]. To introduce limitations on the velocities, we use a hyperbolic tangent function (tanh) as an activation function at the output layer. We use a linear function as that of the Critic network since no such limitation exists for Q-values (i.e., output of the Critic network).

C. Dataset construction using a simulator

We use a realistic robot simulator V-REP [35] for generating dataset for training and testing. Fig. 9 shows the model of the attendant robot used for simulation. It is equipped with a camera and two omnidirectional 2D LRFs.

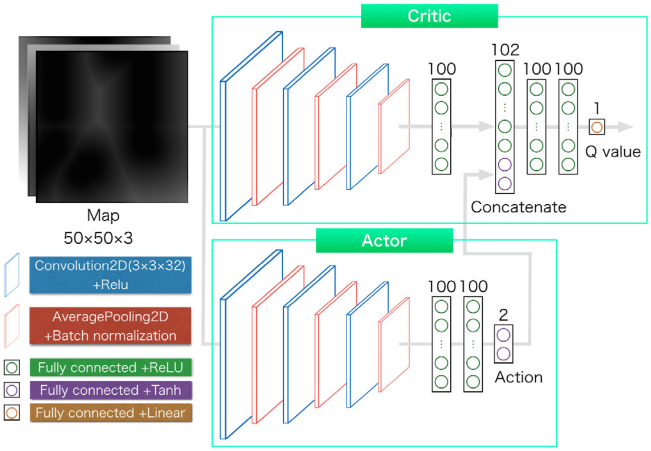


Fig. 8: Network structure.

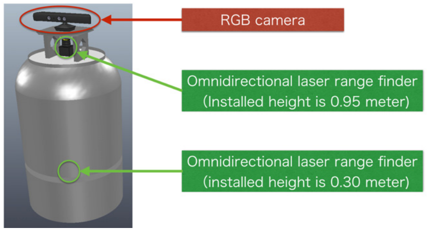

RGB camera
Omnidirectional laser range finder (Installed height is 0.95 meter)
Omnidirectional laser range finder (installed height is 0.30 meter)

Fig. 9: Robot model.

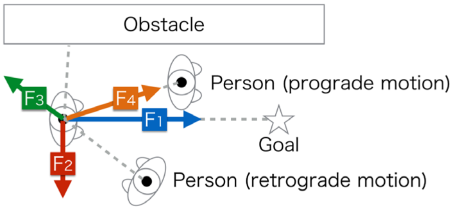


Fig. 10: Social force model.

To automatically generate a variety of situations, we simulate people movement using a social force model (SFM) [36]. SFM controls each person using an attractive force from the destination and repulsive forces from obstacles and other persons, as shown in Fig. 10.

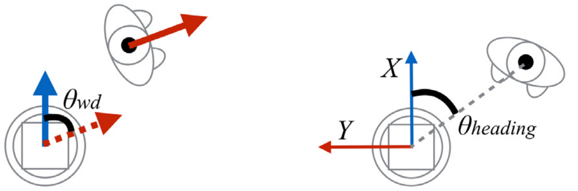
D. Reward functions

We consider four factors in defining reward functions for attending behaviors: relative orientation to the target person, translational acceleration, rotational acceleration, and relative positioning to the target person. We also define a reward function for the end of each episode. The summation of all reward functions is used for training.

1) Reward for accelerations: Abrupt changes of the speed and/or the moving direction are dangerous and increase the possibility of collisions and falls. We thus give a negative reward for accelerations above a certain threshold, defined by eqs. (1) and (2):

$$R_t^{acc}(Acc_t) = \begin{cases} 0.30 - Acc_t[\mathrm{m/s^2}] & (Acc_t > 0.3) \\ 0 & (\text{otherwise}) \end{cases} \quad (1)$$

$$R_r^{acc}(Acc_r) = \begin{cases} \pi/6 - Acc_r[\mathrm{rad/s^2}] & (Acc_r > \pi/6) \\ 0 & (\text{otherwise}) \end{cases} \quad (2)$$



(a) for walking and standing.

(b) for sitting and talking.

Fig. 11: Definitions of relative orientation.

2) Reward for relative orientation: We would like to train the robot to take an appropriate heading with respect to the position or the heading of the target. The reward function for each state is defined as follows.

a) Walking and standing state: The camera of the robot is directed forward and the robot needs to face the similar direction with that of the target, in order to watch his/her front region. We thus give a negative reward when the orientational difference θ_{wd} is large as follows (see Fig. 11(a)):

$$R_{walk,stand}^{ori}(\theta_{wd}) = \begin{cases} -1 & (\theta_{wd} > \pi/2[\mathrm{rad}]) \\ 0 & (\text{otherwise}) \end{cases} \quad (3)$$

b) Sitting and talking state: In these states, the robot needs to face to the target for watching him/her. We thus give a negative reward when the angle $\theta_{heading}$ between the robot heading and the direction to the target is large as follows (see Fig. 11(b)):

$$\begin{aligned} &R_{sit,talk}^{ori}(\theta_{heading}) \\ &\quad = \begin{cases} \pi/4 - \theta_{heading} & (\theta_{heading} > \pi/4[\mathrm{rad}]) \\ 0 & (\text{otherwise}) \end{cases} \end{aligned} \quad (4)$$

3) Reward for relative positioning: We would like to train the robot for each state to take an appropriate position with respect to that of the target.

a) Walking and standing state: A human caregiver who is attending another person watches the surroundings of that person and navigate him/her so that any dangers and accidents are avoided. To this end, the caregiver should be at either side of that person to observe him/her front region. To design a reward function for such a behavior, we analyze a dataset [37] which recorded the motions of caregivers with respect to the attended elderly. Fig. 12 shows the distribution of the caregivers' relative position to the target (indicated by an orange triangle), where red points indicate high frequencies and blue ones low frequencies. We normalize this distribution by the maximum frequency to use

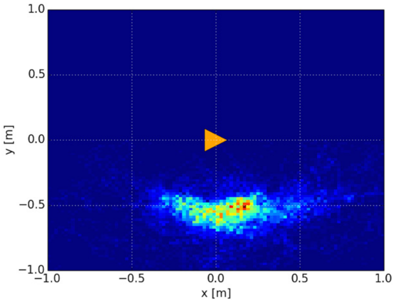
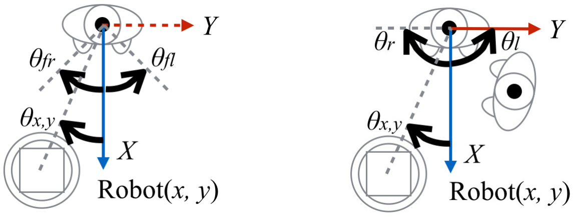


Fig. 12: Distribution of relative position of the caregiver with respect to the attended elderly.



(a) for sitting.

(b) for talking.

Fig. 13: Definitions of relative position.

as a part of reward function. We also give a negative reward when the robot is too far (more than 1 [m]) from the target. The combined reward function is then defined as:

$$R^{pos}_{walk,stand}(x,y) = \begin{cases} 1.0 - d_{x,y} & (d_{x,y} > 1.0[\mathrm{m}]) \\ distrib(x,y) & (\mathrm{otherwise}) \end{cases} \quad (5)$$

where $distrib(x,y)$ is the normalized distribution.

b) Sitting state: When attending a sitting person, the robot has to stand by by considering not only the distance to obstacles but also the comfort of the target person. Based on our previous result [8], users prefer that a robot stand by at their front-left or front-right positions. It is also necessary to keep a certain distance d_{sit} (1.3 [m] in this case) to the target. The reward function is then defined as:

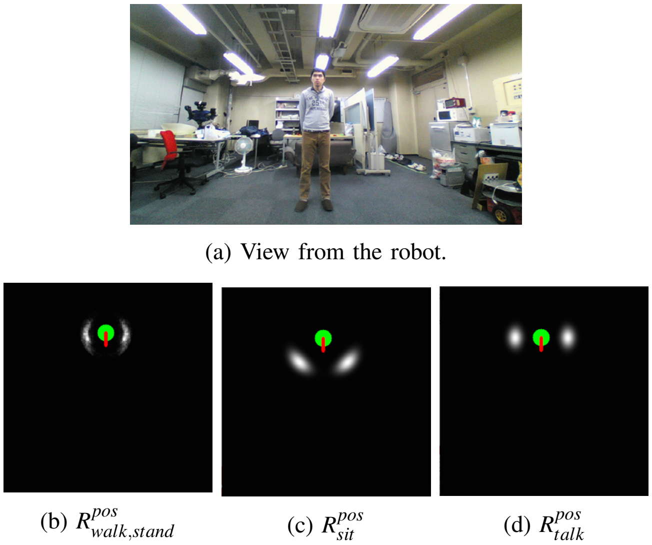
$$\begin{aligned} R^{pos}_{sit}(x,y) \;=\; & \exp\left\{-s_d(d_{x,y}-d_{sit})^2\right\} \\ & \cdot \max\left(\exp\left\{-s_{th}(\theta_{x,y}-\theta_{fl})^2\right\},\right. \quad (6) \\ & \left.\exp\left\{-s_{th}(\theta_{x,y}-\theta_{fr})^2\right\}\right), \end{aligned}$$

where the angles in the equation are indicated in Fig. 13(a). s_d and s_{th} are experimentally set to 16.0 and 8.0, respectively.

c) Talking state: When attending a person talking with another, the robot has to stand so that it does not bother them but in the view of him/her. We therefore give higher rewards at his/her left and right position and at nearer to the target distance d_{talk} (currently, 1 [m]). The reward function is then defined as:

$$\begin{aligned} R^{pos}_{talk}(x,y) \;=\; & \exp\left\{-s_d(d_{x,y}-d_{talk})^2\right\} \\ & \cdot \max\left(\exp\left\{-s_{th}(\theta_{x,y}-\theta_{l})^2\right\},\right. \quad (7) \\ & \left.\exp\left\{-s_{th}(\theta_{x,y}-\theta_{r})^2\right\}\right), \end{aligned}$$

where the angles in the equation are indicated in Fig. 13(b).



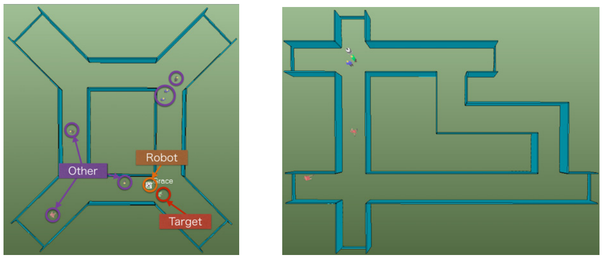
(a) View from the robot.

(b) $R^{pos}_{walk,stand}$

(c) R^{pos}_{sit}

(d) R^{pos}_{talk}

Fig. 14: Reward distribution examples.



(a) Environment 1

(b) Environment 2

Fig. 15: Simulated environments for evaluation.

d) Reward examples for relative positioning: Fig. 14 shows examples of reward functions for relative positioning. For the scene shown in Fig. 14(a), we calculated the functions using eqs. (5)(6)(7). The results are shown in Fig. 14(b)(c)(d). In the distributions, green circles and the red lines indicate the position and the orientation of the target person, respectively.

4) Reward for the end of episode: An episode ends when the target person reaches a designated goal, when the robot collides with an obstacle or a person, or when the robot loses the target person. To train the network to avoid the second and the third cases, we give −10 for collision and target lost cases.

V. Experimental Evaluation

We first evaluate the proposed state space representation in comparison with others. We then examine the effect of the local map resolution to the performance. Based on these results, we conducted experiments of robotic attending in simulated and real environments. We use environment 1 in Fig. 15 for training, and environment 2 for testing in simulation.

TABLE II: Training parameters

Batch size	32
Discount rate of reward	0.99
Target network hyperparameter	0.001
Random process	Ornstein-Uhlenbeck process
Optimizer	Adam
Actor network's learning late	0.0001
Critic network's learning late	0.001
Graphic card	GeForce Titan X Pascal

A. Selection of state space

We compare the proposed local distance maps (LDM) with the following three representations: a concatenation of LIDAR scan data and person position and velocity data (LID), the local maps (LM), and an omnidirectional image (OI). For evaluating LID and OI, we modify the feature extraction part of the network to deal with the respective state representation. In the simulation, the target person follows a designated trajectory while others appear at multiple locations and take actions of walking, standing, and talking.

Table II summarizes the parameters used for training. We train the network for 10,000 epochs and compare the performances of respective representation in terms of the averaged duration of successful attending, the number of successful episodes (i.e., the robot can attend the whole target travel), and the averaged rewards.

Table III shows the evaluation results for all state space representations. The local distance map representation is far better than the others. The rewards for accelerations decreased for this representation, however, because larger accelerations were used for successfully avoiding collisions. Note that "Collision" and "Lost" cases sometimes happen simultaneously and the sum of the counts for each state space is thus larger than 500, the total number of trials.

B. Selection of local map resolution

We compare several local map resolutions for the same environment. Table IV summarizes the results. Too low resolution misses necessary shape features, while too high resolution takes much time for training. We choose 50×50 which gives the best among the tested resolutions.

C. Example attending behavior

We performed attending experiments in the simulator, using a pre-planned scenario of the target person. Fig. 16 shows a sequence of behaviors for a single run, in which the robot adaptively changes its behavior based on the user state classification results. In each figure, the left two images show the scene from two different viewpoints and the right one indicates the map. Blue, green, red regions in the map indicate obstacles, the target person region, and other persons' regions, respectively.

In this scenario, the robot first follows the target person (Fig. 16(a)) and stops when he stops (Fig. 16(b)). Then he re-starts walking and sits on a chair. The robot follows (Figs. 16(c) and 16(d)) and stops at a stand-by position (Fig. 16(e)). The robot starts following again (Fig. 16(f)) and stands by when he is talking with others. Then the robot start following him again after he finishes talking (Fig. 16(h)).

D. Experiments using a real robot

We implemented the proposed method on a real robot and tested in various situations. Figs. 17, 18, and 19 show snapshots of robot behaviors for a walking, sitting, and talking person, respectively. Appropriate robot behaviors are generated according to the state of the target person.

VI. Conclusions and Future Work

This paper has described a method of generating attendant robot behaviors adaptively to the user state. User state classification is performed in a rule-based manner, using the position, the velocity, the height, and the orientation of the user obtained from images and LIDAR data. We have shown that a high classification performance is achieved on a newly constructed dataset. Behavior generation is done by using Deep DPG, with a new state space representation, *local distance map*, and with reward functions carefully designed by considering requirements on robot behavior for each user state. We have shown that our representation is far better than the others. We have also shown that the proposed method can cope with state changes of the user in experiments in simulated and real environments.

User state classification is a key to comfortable and safe attending. Since the current approach uses only the latest user information, there may be a delay between the change of the user state and that of robot behavior. Developing a method of early recognition of user intention is future work for a better attending behavior.

Designing reward functions is another issue. Although our local distance map representation exhibits a much better performance than the others, the ratio of reaching the designated goal is still not high enough. Reducing the reward for a narrow space, for example, could increase the ratio. It is also desirable to adjust reward functions to each user because preference to the robot behaviors such as the comfortable relative distance to the robot may be different for respective users. Adjusting such a preference through attending experiences could increase the satisfaction of the users. It is also needed to evaluate and improve the methods through a variety of real situations.

References

[1] K. Arras, O. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Proceedings of the 2007 IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 3402–3407.

[2] M. Munaro and E. Menegatti, "Fast rgb-d people tracking for service robots," *Autonomous Robots*, vol. 37, no. 3, 2014.

[3] K. Koide and J. Miura, "Identification of a specific person using color, height, and gait features for a person following robot," *Robotics and Autonomous Systems*, vol. 84, no. 10, pp. 76–87, 2016.

[4] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite rrts for rapid replanning in dynamic environments," in *Proceedings of 2007 IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 1603–1609.

[5] I. Ardiyanto and J. Miura, "Real-time navigation using randomized kinodynamic planning with arrival time field," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1579–1591, 2012.

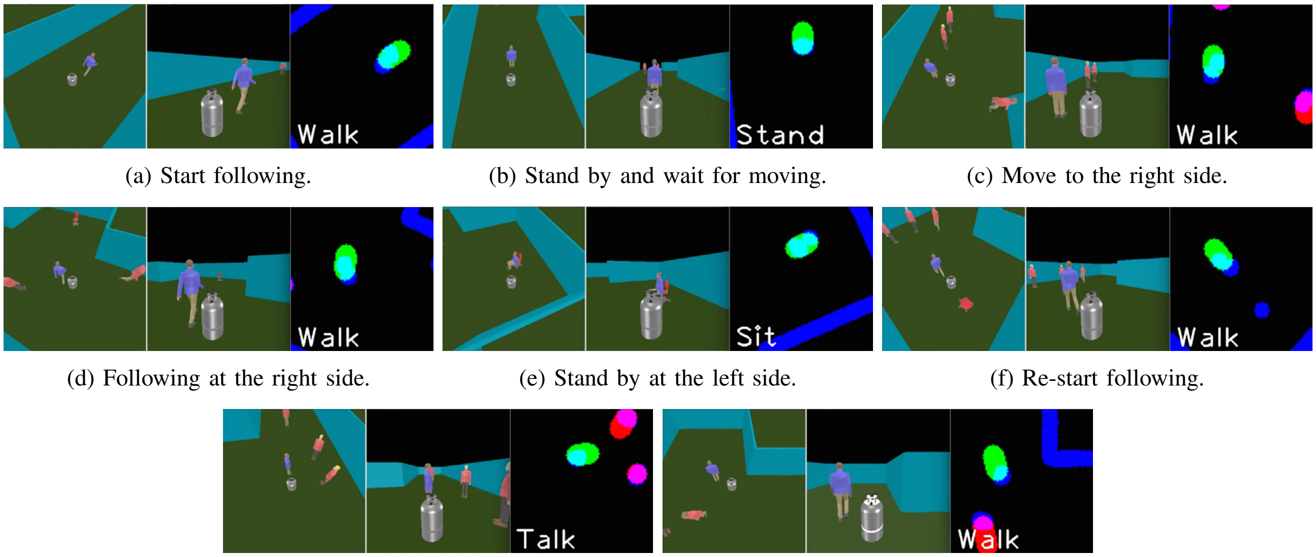
[6] M. Fiore, H. Khambhaita, G. Milliez, and R. Alami, "An adaptive and proactive human-aware robot guide," in *International Conference on Social Robotics*, 2015, pp. 194–203.

TABLE III: Performance of each state space representation. The resolution of the local maps is 50 × 50.

State Space	Averaged duration [s]	Count				Averaged reward					
		Trial	Goal	Collision	Lost	Total	Relative pos.	Trans. acc.	Rot. acc.	Heading	Episode end
LID	3.7	500	0	497	20	-21.5	-2.3	-3.1	-0.3	-5.5	-10.3
LM	2.7	500	0	500	14	-21.0	-2.3	-2.6	-0.5	-5.3	-10.3
LDM	**49.4**	500	**104**	408	4	**44.3**	67.6	-4.6	-1.3	-9.2	-8.2
OI	3.5	500	0	477	274	-21.7	-2.5	-0.6	-2.2	-1.4	-15.0

TABLE IV: Performance of each resolution of local maps. The state space representation is local distance maps (LDM).

State Space	Averaged duration [s]	Count				Averaged reward					
		Trial	Goal	Collision	Lost	Total	Relative pos	Trans. acc.	Rot. acc.	Heading	Episode end
25	2.7	500	0	495	14	-17.8	-1.7	-0.9	-0.2	-4.8	-10.2
38	47.7	500	97	393	19	23.6	47.4	-5.1	-2.9	-7.5	-8.2
50	**49.4**	500	**104**	408	4	**44.3**	67.6	-4.6	-1.3	-9.2	-8.2
75	2.8	500	1	486	33	-25.1	-6.6	-1.0	-0.8	-6.4	-10.4
100	2.2	500	1	490	65	-24.6	-6.1	-0.6	-2.6	-4.2	-11.1



(a) Start following.

(b) Stand by and wait for moving.

(c) Move to the right side.

(d) Following at the right side.

(e) Stand by at the left side.

(f) Re-start following.

(g) Stand by at the right side.

(h) Re-start following.

Fig. 16: Example attending behaviors.

[7] P. Leica, J. M. Toibero, F. Roberti, and R. Carelli, "Switched control to robot-human bilateral interaction for guiding people," *J. Intell. Robotics Syst.*, vol. 77, no. 1, pp. 73–93, Jan. 2015. [Online]. Available: http://dx.doi.org/10.1007/s10846-014-0098-6

[8] S. Oishi, Y. Kohari, and J. Miura, "Toward a robotic attendant adaptively behaving according to human state," in *Proceedings of 2016 IEEE Int. Symp. on Robot and Human Interactive Communication*, 2016, pp. 1038–1043.

[9] Z. Zainudin, S. Kodagoda, and G. Dissanayake, "Torso detection and tracking using a 2d laser range finder," in *Proceedings of Australasian Conf. on Robotics and Automation 2010*, 2010.

[10] J. Satake and J. Miura, "Robust stereo-based person detection and tracking for a person following robot," in *Proceedings of ICRA-2009 Workshop on Person Detection and Tracking*, 2009.

[11] L. Spinello and K. Arras, "People detection in rgb-d data," in *Proceedings of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 3838–3843.

[12] N. Bellotto and H. Hu, "Multisensor-based human detection and tracking for mobile service robots," *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 1, pp. 167–181, 2009.

[13] I. Ardiyanto and J. Miura, "Visibility-based viewpoint planning for guard robot using skeletonization and geodesic motion model," in *Proceedings of the 2013 IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 652–658.

[14] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3169–3176.

[15] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

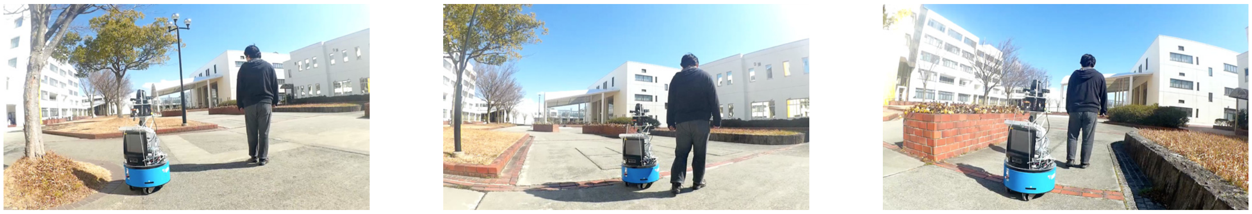
[16] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4305–4314.

[17] C. Feichtenhofer, A. Pinz, and R. Wildes, "Spatiotemporal residual networks for video action recognition," in *Advances in Neural Information Processing Systems*, 2016, pp. 3468–3476.

[18] I. Cosmin Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-temporal vector of locally max pooled features for action recognition in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3097–3106.

[19] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3. IEEE, 2004, pp. 32–36.

[20] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-

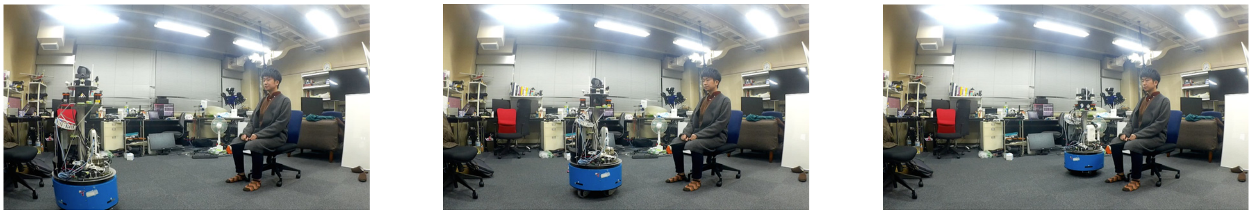


(a) Following the person.

(b) Move to the left side of the person.

(c) Move to the back of the person to avoid collision.

Fig. 17: Attendant behavior for walking.

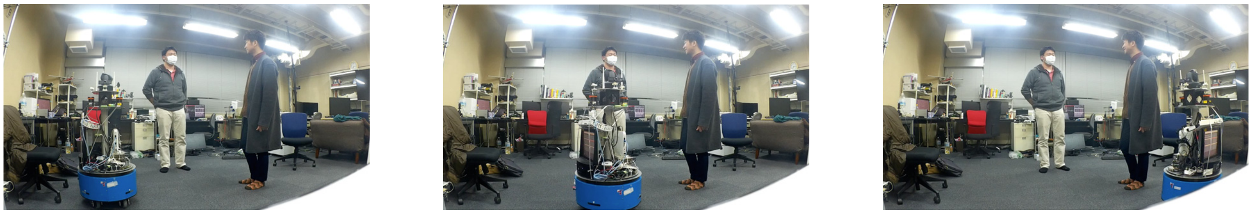


(a) Detect and localize a sitting person.

(b) Move to the stand-by position.

(c) Wait at the right side of the person.

Fig. 18: Attendant behavior for sitting.



(a) Detect and localize a person talking with another.

(b) Move to the stand-by position.

(c) Wait at the left side of the person.

Fig. 19: Attendant behavior for talking.

temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[21] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 20–27.

[22] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110–1118.

[23] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 2010, pp. 9–14.

[24] K. Cho and X. Chen, "Classifying and visualizing motion capture sequences using deep neural networks," in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 2. IEEE, 2014, pp. 122–130.

[25] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda, "Purposive behavior acquisition for a real robot by vision-based reinforcement learning," *Machine learning*, vol. 23, no. 2, pp. 279–303, 1996.

[26] W. D. Smart and L. P. Kaelbling, "Effective reinforcement learning for mobile robots," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4. IEEE, 2002, pp. 3404–3410.

[27] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[28] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[29] A. Barto, R. Sutton, and C. Anderson, "Neuronlike elements that can solve difficult learning control problems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 13, pp. 835–846, 1983.

[30] Y. Kohari, J. Miura, and S. Oishi, "CNN-based human body orientation estimation for robotic attendant," in *IAS-15 Workshop on Robot Perception of Humans*, 2018.

[31] E. T. Hall, *The hidden dimension: man's use of space in public and private*. Bodley Head, 1969.

[32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[33] B. Wymann, C. Dimitrakakis, A. Sumner, E. Espié, C. Guionneau, and R. Coulom, "TORCS, the open racing car simulator," http://www.torcs.org.

[34] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.

[35] E. Rohemr, S. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *Proceedings of 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 1321–1326.

[36] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[37] K. Koide, J. Miura, and E. Menegatti, "Aisl attendant behavior dataset," http://www.aisl.cs.tut.ac.jp/database_fukushimura.html.