# Interactive Human Intention Reading by Learning Hierarchical Behavior Knowledge Networks for Human-Robot Interaction

Ji-Hyeong Han, Seung-Hwan Choi, and Jong-Hwan Kim

**For efficient interaction between humans and robots, robots should be able to understand the meaning and intention of human behaviors as well as recognize them. This paper proposes an interactive human intention reading method in which a robot develops its own knowledge about the human intention for an object. A robot needs to understand different human behavior structures for different objects. To this end, this paper proposes a hierarchical behavior knowledge network that consists of behavior nodes and directional edges between them. In addition, a human intention reading algorithm that incorporates reinforcement learning is proposed to interactively learn the hierarchical behavior knowledge networks based on context information and human feedback through human behaviors. The effectiveness of the proposed method is demonstrated through play-based experiments between a human and a virtual teddy bear robot with two virtual objects. Experiments with multiple participants are also conducted.**

**Keywords: Human intention reading, Developmental knowledge about human intention, Hierarchical behavior knowledge network, Human–robot interaction, Reinforcement learning.**

Manuscript received Feb. 25, 2016; revised Aug. 19, 2016; accepted Aug. 25, 2016.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2014R1A2A1A10051551).

Ji-Hyeong Han (jihyeong.han@etri.re.kr) is with the SW & Content Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Seung-Hwan Choi (cyclopsh@gmail.com) is with the UX Center, Samsung Electronics, Seoul, Rep. of Korea.

Jong-Hwan Kim (corresponding author, johkim@rit.kaist.ac.kr) is with the School of Electrical Engineering, KAIST, Daejeon, Rep. of Korea.

## I. Introduction

In the near future, humans and robots will live and work together in human homes and offices because of the fast development of robot and artificial intelligence technologies. To prepare for this future, natural and rational human–robot interaction (HRI) is needed; therefore, research dealing with HRI problems has been widely carried out.

This paper focuses on how a robot infers human intentions when they work or play together with an object using context information and human behaviors. Thus, human intention in this paper is defined as the behavior a human wants a robot to do using an object. To determine the defined human intention from human behaviors, a robot should learn about the structures of these behaviors. Human behaviors are hierarchically structured as opposed to sequentially structured in order to achieve a certain goal. This is a typical feature of human behaviors [1]. Such hierarchical behavior structures differ when a human acts using different objects; therefore, a robot needs to learn the hierarchical behavior structures for different objects to read the human intention for an object.

To achieve this goal, we propose a hierarchical behavior knowledge network (HBKN) that enables a robot to learn hierarchical behavior structures for different objects. The proposed HBKN represents behavior hierarchies using directional edges between behavior nodes and adding transition probabilities to the directional edges. In addition, a human intention reading algorithm that incorporates passive reinforcement learning based on adaptive dynamic programming is proposed to learn the HBKN and infer human intention. Observed human arm behaviors are used as inputs to the proposed algorithm; therefore, we also develop a human arm behavior recognition procedure. The most popular method for human behavior recognition uses computer vision methods [2], [3], thus we also use a robot vision system to recognize human behavior. The HRI experiments, which consist of play with two virtual objects, that is, a ball and toy car, were carried out using a teddy bear robot simulator to demonstrate the effectiveness of the proposed method. In addition, experiments with multiple human participants who were not involved in developing the system were performed to determine whether the proposed method would work well with the general public.

This paper is organized as follows. Section II explains the related work for research background. Section III briefly describes the overall data flow and Section IV presents the details of the core parts, that is, HBKN and the human intention reading algorithm. In Section V, the experimental environment is presented and experimental results obtained for three different scenarios with multiple participants are discussed. Finally, the concluding remarks follow in Section VI.

## II. Related Work

There have been several studies to infer human intentions or goals. These studies can be classified according to their approaches as follows.

Simulation theory is one of the dominant human mind reading theories, which suggests that humans use their own mental mechanisms to predict the mental processes of others, like simulation does [4]. Thus, there have been several studies in which a robot infers a human intention in the same manner as a human by applying simulation theory. Gray and others presented action parsing and goal inference algorithms by considering the robot itself as a simulator [5], and Breazeal and others developed robot embodied cognition including mindreading skills [6]. Jansen and others developed a computation model to imitate an artificial agent that infers the intention of a demonstrator [7]. Hiatt and others presented an effective method to deal with human variability during HRI by employing the theory of mind and ACT-R [8].

Building probabilistic models, including Bayesian inference models, of human intention is one of the most popular ways to infer human intention. Schrempf and others developed a general model for user intention estimation using hybrid dynamic Bayesian networks [9] and proposed a method in which a robot selects a task based on the estimated user intention [10]. Kelley and others developed an intention recognition framework based on Bayesian reasoning using the context information of an object [11]. Wang and others developed an intention-driven dynamics model that is a probabilistic model for the process of intentional movements [12].

The artificial neural networks approach, which mimics the human brain, is another promising way to infer human intention. Bicho and others devised a control architecture based on a close perception-action linkage using dynamic neural fields [13] and a dynamic neural field-based model [14].

Learning a model using machine learning methodology based on data labeled by human intention is an intuitive and effective approach to human intention reading. Strabala and others presented a key discriminative feature learning method using SVM methodology for predicting the human intention to hand over an object [15]. In addition to traditional machine learning methods, there have been initial trials to apply deep learning to the human intention reading problem. Kelley and others proposed a deep learning architecture for predicting human intention with respect to objects and presented its initial results [16]. Yu and Lee developed a deep dynamic neural model for human intent recognition based on human motions [17].

## III. Overall Data Flow for Interactive Human Intention Reading

Figure 1 shows the overall data flow of the proposed method. The object information and human arm behavior information are sensed by a sensing module such as an RGB-D camera. The percepts are then passed to the attention module. A robot needs to focus its attention on a certain object and/or human in the environment in order to interact with them because there can be several objects and/or humans. Hence, the interest factor (*IF*) for each perceived object, which includes humans, is defined in the attention module as follows:

$$IF\left(O_i\right)=\begin{cases}1 & \text{when } O_i \text{ is moving,}\\ e^{-k_1(t)/\tau} & \text{when } O_i \text{ is not moving,}\end{cases} \qquad (1)$$

where $O_i$ is the $i$-th perceived object (either an object or a human), and $\tau$ is a time constant. Furthermore, $k_1(t)$ is initially zero, increments by one at each time step when $O_i$ is not moving, and returns to zero when $O_i$ moves. When the object or human moves, the corresponding *IF* immediately becomes

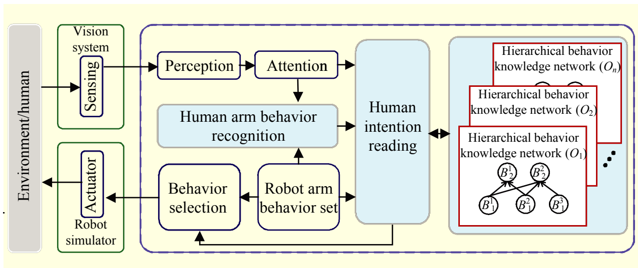


Fig. 1. Overall data flow.



ETRI Journal, Volume 38, Number 6, December 2016
http://dx.doi.org/10.4218/etrij.16.0116.0106

one. The robot gives its attention to the object or human that has the highest *IF*. When the object or human stops, *IF* is decreased by $\tau$.

The perceived human arm behavior information and information about the object that is the focus of the robot's attention are transferred to the human arm behavior recognition module and human intention reading module, respectively. The human arm behavior recognition module recognizes the observed arm behavior using the robot's arm behavior set and a dynamic time warping (DTW) algorithm. The robot behavior with the minimum final DTW distance (FDTW), that is, the most similar one, is recognized as the human behavior [18]. The robot arm behavior set module provides the possible robot arm behaviors, such as waving both hands (WB), waving one hand (WO), pointing (Po), touching (Tch), pushing (Pu), grasping (Gr), releasing (Re), and throwing (Th). The recognition result is transferred to the human intention reading module.

In the human intention reading module, a robot simultaneously develops the HBKN of the object and infers the human intention for the object using the proposed human intention reading algorithm. The details of the HBKN and human intention reading algorithm are explained in Section IV. The human intention is identified by the proposed algorithm, and then the behavior selection module selects the best behavior considering the inferred intention and it is performed using the actuator module.

## IV. HBKN and Human Intention Reading Algorithm

In this section, the HBKN and human intention reading algorithm are described in more detail. The robot develops its own knowledge by learning the HBKNs of objects through interaction with a human.

### 1. HBKN

Because different objects have different hierarchical behavior structures for the achievement of a goal and the human intention for different objects might be different, each object needs to have its own developed HBKN. A robot can then infer the human intention for an object by using an HBKN, which is defined as follows.

***Definition* 1**: An HBKN consists of behavior nodes and directional edges to represent the hierarchical relations between behavior nodes.

(i) $n(perceived\ objects) = n(HBKN)$.

(ii) An HBKN consists of several classes (*Class i*) and each class consists of behavior nodes $B_i^j$, which denotes the *j*-th behavior in the *i*-th class.

(iii) Each behavior node has utility $HBKN.U\left(B_i^j\right)$.

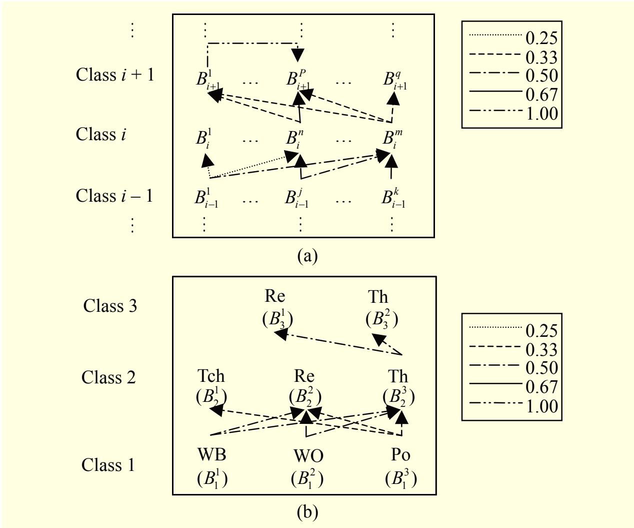


Fig. 2. (a) General description of an HBKN and (b) an example of an initial HBKN, where $B_i^j$ denotes the *j*-th behavior in *Class i* and the different dashed edges represent different transition probabilities.

(iv) Each directional edge between behavior nodes has transition frequency $HBKN.TF\left(B_i^j, B_k^l\right)$ and transition probability $HBKN.TP\left(B_i^j, B_k^l\right)$.

Figure 2 shows the general HBKN structure and an example of an initial HBKN. As shown in Fig. 2(a), HBKN consists of several classes, each class consists of several behavior nodes, and there are directional edges with transition probabilities between the behavior nodes. In Fig. 2(b), *Class* 1, *Class* 2, and *Class* 3 respectively comprise behaviors such as WB, WO, and Po that can be done without an object; behaviors such as Gr, Pu, and Tch that can be done with an object; and behaviors such as Re and Th that can be done after the robot obtains an object. After the behaviors in *Class* 3 are complete, the robot does not hold the object anymore.

### 2. Human Intention Reading Algorithm

The proposed human intention reading algorithm incorporates passive reinforcement learning with adaptive dynamic programming so that the robot infers human intentions by learning HBKNs interactively through human behaviors and feedback. To apply the reinforcement learning scheme, the behavior nodes in HBKN are treated as states and each behavior node has a utility and reward. The utilities and rewards of behavior nodes $HBKN.U\left(B_i^j\right)$ and $HBKN.R\left(B_i^j\right)$ are initially zero. In addition, the directional edges in the HBKN are treated as the transition model, and each edge has a transition probability that is calculated from the transition frequencies. Transition frequency

$HBKN.TF\left(B_i^j, B_k^l\right)$ is the number of transitions from $B_i^j$ to $B_l^k$. The initial transition frequency value is one if an edge exists between behavior nodes, and zero otherwise. Using the transition frequencies, the transition probability is calculated as $HBKN.TP\left(B_i^j, B_k^l\right) = HBKN.TF\left(B_i^j, B_k^l\right) / \sum HBKN.TF(B_i^j, \forall B)$. In Fig. 2(b), the different styles of dashed edges between behavior nodes represent the different transition probabilities.

Figure 3 shows the flow charts of the proposed human intention reading algorithm. The pseudo code of each function in the main algorithm is provided respectively in Algorithms 1–5. The main algorithm can be divided into two parts: when the robot simultaneously learns HBKN for an object and infers the human intention (Fig. 3(a)) and when the robot infers the human intention using the learned HBKNs (Fig. 3(b)).

The simultaneous HBKN learning and human intention reading algorithm (Fig. 3(a)) starts when the robot focuses its attention on a perceived object *O*. Because there is only one perceived object, the object is automatically identified as a human intention object (*HIO*). The algorithm then checks the HBKNs. If a previously learned *HBKN(HIO)* exists, it loads *HBKN(HIO)*; otherwise, it creates the initial *HBKN(HIO)*.

The algorithm runs repeatedly until the human intention (*HI*) is inferred. First, it perceives a human action (*HA*) such as a human arm behavior or feedback. If the *HA* is a recognized human behavior or positive feedback for the robot's behavior, it updates the transition probability for each edge and then updates the utilities of the behavior nodes. Otherwise, it just updates the utilities of the behavior nodes. The utilities of all the behavior nodes are updated in the same manner using the Bellman equation [19]:

$$U(B) = R(B) + \eta \sum_{\forall B'} TP\left(B, B'\right) U\left(B'\right), \tag{2}$$

where $U(B)$ is the utility of behavior node $B$, $R(B)$ is the reward for $B$, $TP(B, B')$is the transition probability from $B$ to $B'$, and $\eta$ is the discount factor. (The pseudo code for updating the transition probability and behavior node utility are provided respectively in Algorithms 1 and 2.) It then searches for a candidate behavior knowledge link (*CBKL*) and the current status (*curStatus*) in the updated *HBKN*(*HIO*). (The pseudo code for searching for the *HBKN* is provided in Algorithms 3 and 4.)

The information of *curStatus* indicates that the robot is in one of the three states: i) it infers the human intention, ii) it needs to perform trial behaviors to obtain human feedback, or iii) it is confused. In the first state, the behavior node with the maximum utility of all the behavior nodes in HBKN(HIO) has been inferred as the HI. As the robot has inferred the human intention, the algorithm is terminated. The second state indicates that there is more than one possible next behavior

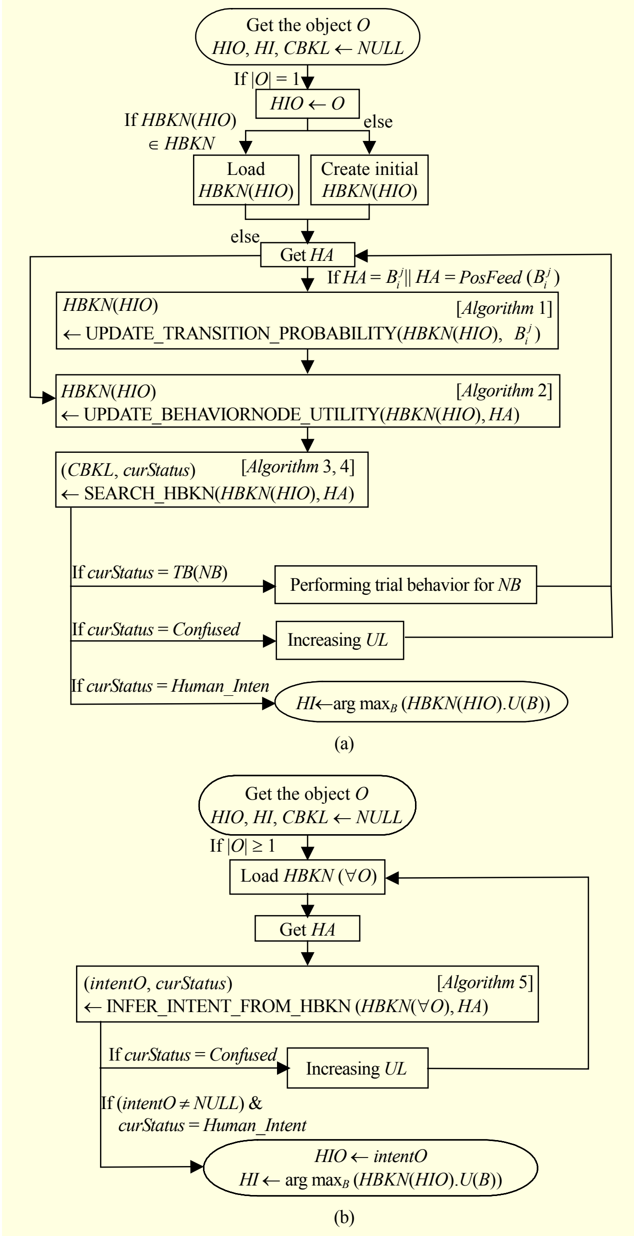


Fig. 3. Main human intention reading algorithms: (a) simultaneous HBKN learning and human intention reading algorithm and (b) human intention reading algorithm with learned HBKNs.

(NB); therefore, the robot needs to perform a trial behavior for each possible NB and must wait for positive or negative human feedback. The received human feedback is considered as an HA and the algorithm is repeated until it meets the termination condition, that is, it has inferred the human intention. In the last state, the robot is confused; therefore, it has not obtained sufficient information from the current human behavior or feedback to infer the human intention. Then, uncertainty level (*UL*) increases. The *UL* makes the robot ask for more information and is defined as follows:

$$UL = \begin{cases} c \times k_2(t)^2 & \text{when a robot is confused,} \\ 1 & \text{when } UL \geq 1, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $c$ is a positive constant and $k_2(t)$ is initially zero, increases by one at each time step when the robot is confused, and returns to zero when the confusion is resolved. Therefore, when the robot is confused, the *UL* increases proportionally to the square of the time step. When *UL* exceeds the predefined threshold value $U_0$, the robot asks for more information from the human. After receiving new information, such as a human behavior or feedback, *UL* is reset to zero and the algorithm is repeated until it meets the termination condition.

The human intention reading algorithm with learned HBKNs (Fig. 3(b)) starts when the robot focuses its attention on perceived objects ($O = \{O_1, O_2, \ldots, O_n\}$). Because the robot has already learned the HBKNs of the perceived objects, the algorithm loads these HBKNs and receives the *HA*. The algorithm runs repeatedly until *HI* is inferred with the identified *HIO*. (The pseudo code for inferring *HIO* and *HI* from the HBKN is provided in Algorithm 5.) If *HIO* is inferred, then the behavior node that has the maximum utility of all the behavior nodes in *HBKN*(*HIO*) is inferred as *HI*, and the algorithm is terminated. If the function still returns that the robot is confused, then *UL* increases and the algorithm repeats until it meets the termination condition.

• *Function UPDATE_TRANSITION_PROBABILITY*: Algorithm 1 shows the pseudo code of this function. It first updates the transition frequencies of all the edges *HBKN.TF* based on the activated behavior node $B_i^j$ and then updates the transition probabilities of all edges *HBKN.TP*. The function identifies the lower class (*Class* $i-1$) behavior nodes *maxUB*s that have the maximum utility in their class.

There are two cases based on the number of *maxUB*s. In the first case, there is only one *maxUB*, and this case has two sub-cases. If only one lower class behavior node $B_{i-1}^k$ that has *TF* to $B_i^j$ already exists, $B_{i-1}^k$ needs to be performed before $B_i^j$. Thus, the *TF*s from *maxUB* to $B_{i-1}^k$ and from $B_{i-1}^k$ to $B_i^j$ are increased by one. Otherwise, the *TF*s from *maxUB* to $B_i^j$ are increased by one. In the other case, there is more than one *maxUB*. In this case, the function identifies the behavior nodes *maxTFB*s that have the maximum *TF* to $B_i^j$ of *maxUB*s and *TF*s from all the *maxTFB*s to $B_i^j$ are increased by one.

Finally, the function updates the transition probabilities of all the edges, $HBKN.TP(B, B')$ by dividing $HBKN.TF(B, B')$ by the sum of the *TF*s from *B* to all the connected behavior nodes. After updating the transition probabilities, the function returns the updated HBKN.

**Algorithm 1.** Transition probability update function

**Input:** network (*HBKN*), activated behavior $(B_i^j)$
**Output:** updated transition probability of *HBKN* (*HBKN.TP*)
**function** UPDATE_TRANSITION_PROBABILITY($HBKN, B_i^j$)
  $maxUB \leftarrow \arg\max_{B_{i-1}^k} (HBKN.U(B_{i-1}^{\forall k}))$
  **if** $n(maxUB) = 1$ **then**
    **if** $(n(\{B_{i-1}^k \mid HBKN.TE(B_{i-1}^k, B_i^j) \neq 0\}) = 1)$ & $(B_{i-1}^k \neq maxUB)$
     **then**
      $HBKN.TF(maxUB, B_{i-1}^k)$ ++, $HBKN.TF(B_{i-1}^k, B_i^j)$ ++
    **else**
      $HBKN.TF(maxUB, B_i^j)$ ++
    **end if**
  **else if** $n(maxUB) > 1$ **then**
    $maxTFB \leftarrow \arg\max_{maxUB} (HBKN.TF(\forall maxUB, B_i^j))$
    $(HBKN.TF(\forall maxTFB, B_i^j)$ ++
  **end if**
  $\forall B, \forall B', HBKN.TP(B, B')$
     $\leftarrow HBKN.TF(B, B') / \sum_{\forall B'' \ni B'} HBKN.TF(B, B'')$
  **return** *HBKN*
**end function**

**Algorithm 2.** Behavior node utility update function

**Input:** network (*HBKN*), current human behavior or feedback (*HA*)
**Output:** updated utility of *HBKN* (*HBKN.U*)
**function** UPDATE_BEHAVIORNODE_UTILITY(*HBKN*,*HA*)
  **if** $(HA = B_i^j) \parallel (HA = PosFeed(B_i^j))$ **then**
    $HBKN.R(B_i^j) \leftarrow \alpha$
  **else if** $HA = NegFeed(B_i^j)$ **then**
    $HBKN.R(B_i^j) \leftarrow \beta$
  **else if** $HA = RightFeed(B_i^j)$ **then**
    $HBKN.R(B_i^j) \leftarrow \gamma$
  **end if**
  $\forall B, HBKN.U(B) \leftarrow HBKN.R(B)$
        $+ \eta \sum_{\forall B'} (HBKN.TP(B, B') \times HBKN.U(B'))$
  **return** *HBKN*
**end function**

• *Function UPDATE_BEHAVIORNODE_UTILITY*: Algorithm 2 shows the pseudo code of this function, which updates the utilities of *HBKN.U(B)*, which is all the behavior nodes in HBKN, based on the received human behavior or feedback *HA*. The reward for each behavior node, $HBKN.R(B_i^j)$, is determined based on *HA*. If *HA* is recognized as a behavior or positive feedback of behavior nodes $B_i^j$, then $HBKN.R(B_i^j)$ gets $\alpha$. If *HA* is a negative feedback for $B_i^j$, then $HBKN.R(B_i^j)$ gets $\beta$. If *HA* is the right intent feedback for $B_i^j$, then $HBKN.R(B_i^j)$ gets $\gamma$. Note that $\alpha$, $\beta$, and $\gamma$ have to satisfy the inequality conditions, $\gamma \gg \alpha > \beta$. The function then updates the utilities of all the behavior nodes using (2).

• *Function SEARCH_HBKN*: Algorithm 3 shows the pseudo code of this function, which searches for the candidate

**Algorithm 3.** HBKN search function

**Input:** network (*HBKN*), candidate behavior knowledge link (*CBKL*), current human behavior or feedback (*HA*)
**Output:** updated *CBKL*, current state of human intent reading process (infer the intent correctly, confused, or trial behaviors)
**function** SEARCH_HBKN (*HBKN*, *prevCBKL*, *HA*)
  $CBKL \leftarrow$ FIND_CBKL($HBKN$)
  **if** $HA = RightFeed(B_i^j)$ **then**
    **return** *CBKL*, *Human_Intent*
  **else if** $(HA = B_i^j) || (HA = PosFeed(B_i^j))$ **then**
    **if** $prevCBKL = CBKL$ **then**
      **return** *CBKL*, *Confused*
    **else**
      $pNB \leftarrow \{B | HBKN.TP(B_i^j, B) \neq 0, \forall B\}$
      $NB \leftarrow \arg\max_{pNB} (HBKN.U(pNB) \times HBKN.TP(B_i^j, pNB))$
      **if** $NB = NULL$ **then**
        **return** *CBKL*, *Confused*
      **else if** $n(NB) = 1$ **then**
        SEARCH_HBKN(*HBKN*, *CBKL*, *NB*)
      **else if** $n(NB) > 1$ **then**
        **return** *CBKL*, $TB(NB)$
      **end if**
    **end if**
  **end if**
**end function**

**Algorithm 4.** CBKL finding function

**Input:** network (*HBKN*)
**Output:** upload candidate behavior knowledge link (*CBKL*)
**function** FIND_CBKL(*HBKN*)
  $i \leftarrow 1, j \leftarrow 0, CBKL \leftarrow NULL$
  **while** $CBKL[0] = NULL$ **do**
    **if** $n(\{first\ B | \arg\max_{B_i^k} (HBKN.U(B_i^{\forall k}))\}) = 1$ **then**
      $CBKL[0] \leftarrow first\ B$
    **end if**
    $i \leftarrow i + 1$
  **end while**
  **while** $n(\{next\ B | \arg\max_B (HBKN.TP(CBKL[j], \forall B))\}) = 1$ **do**
    $j \leftarrow j + 1, CBKL[j] \leftarrow next\ B$
  **end while**
  **return** CBKL
**end function**

**Algorithm 5.** Inferring intention from HBKN function

**Input:** $HBKN(\forall O)$, current human behavior or feedback (*HA*)
**Output:** result of inferring human intended object, current state of process (infer the intent correctly or confused)
**function** INFER_INTENT_FROM_HBKN ($HBKN(\forall O)$, *HA*)
  $objectNo \leftarrow n(O), objCBKL(\forall O) \leftarrow NULL$
  **for** $i = 1 \rightarrow objectNo$ **do**
    $objCBKL(O_i) \leftarrow$ FIND_CBKL($HBKN(O_i)$)
  **end for**
  **if** $(HA \in objCBKL(O_j))$ && $(HA \notin objCBKL(O_{\forall i, i \neq j}))$ **then**
    **return** $O_j$, *Human_Intent*
  **else**
    **return** *NULL*, *Confused*
  **end if**
**end function**

behavior knowledge link *CBKL* and current status from the updated HBKN. First, it calls the *CBKL* search function and obtains the updated *CBKL* for the current updated HBKN. The details of *CBKL* search are explained in *function FIND_CBKL* along with Algorithm 4.

If *HA* is $RightFeed(B_i^j)$, that is, the right intent feedback for $B_i^j$, then the function returns the current status as *Human_Intent* because the robot has correctly inferred the human intention. In contrast, if *HA* is a recognized human behavior $B_i^j$ or $PosFeed(B_i^j)$, the positive feedback for $B_i^j$, then the function starts to search the HBKN. If the returned *CBKL* from *function FIND_CBKL* and the previous *CBKL* are the same, the candidate behavior knowledge link has not been updated. In this case, the function returns the current status as *Confused* because the robot needs more information. Otherwise, it identifies the next behavior nodes *NB* that have the maximum value of the product of utility *HBKN.U*(*possNB*) and $HBKN.TP(B_i^j, possNB)$, which is the transition probability from $B_i^j$ to the *possNB*s that have nonzero transition probabilities.

There are three cases based on the number of *NB*s. In the first case, there is no *NB*. This means that there are no more possible next behavior nodes in the HBKN and the human has not given the right intent feedback; therefore, the function returns the current status *Confused*. In the second case, there is only one *NB*; therefore, the *NB* becomes the next behavior node and the function calls itself recursively. In the last case, there is more than one *NB*. In this case, the robot needs to know which one is the human intended next behavior; therefore, the robot performs trial behavior for all the *NB*s to get human feedback.

• *Function FIND_CBKL*: Algorithm 4 shows the pseudo code of the function, which finds the *CBKL*, that is, the list of behavior nodes in order of hierarchy based on the current learned HBKN. First, the function searches the HBKN from the lowest to the highest class until there is only one behavior node that has the maximum utility in its class. If it finds a behavior node that satisfies the above condition, it stops searching and saves that node as the first element of *CBKL*. After the first element is found, the *CBKL* is sequentially filled with the behavior nodes that have the maximum transition probability from the last element of the *CBKL*. This repeats until there is either no such behavior node or more than one such behavior node, and the function returns the updated *CBKL*.

• *Function INFER_INTENT_FROM_HBKN*: Algorithm 5 shows the pseudo code of this function, which infers the *HIO* and *HI* for the object from the learned HBKNs when the robot's attention is on more than one object. The function obtains the candidate behavior hierarchies for all objects (*objCBKL*) by calling the *CBKL* search function. If *HA* is in the *objCBKL* for only one object, then the object is inferred as


http://dx.doi.org/10.4218/etrij.16.0116.0106

the *HIO,* and the function returns the *HIO* and current status as *Human_Intent*. Otherwise, if there is an *HA* for multiple objects or no objects in *objCBKLs*, the function returns no object and the current status *Confused* because the robot needs more information.

## V. Experiments

To show the effectiveness of the proposed method, experiments were carried out with a human and a virtual teddy bear robot, who played with two virtual objects, a ball and toy car. In addition, the same experiments were carried out with multiple participants who were neither involved in developing the proposed method nor given any prior knowledge about the experimental setup.

### 1. Experimental Environment

In the experiments, a vision system was employed for perceiving the human arm behaviors and feedback. The start and end of human arm behaviors and human feedback were defined by color patches to simplify the experimental setup. The parameters were set to $\tau = 5$, $c = 1/255$, $U_0 = 0.7$, $\eta = 0.9$, $\alpha = 1$, $\beta = -1$, and $\gamma = 5$, in (1), (2), (3), and the behavior node utility update function.

### 2. Human Intention Reading Results

Three experiments were performed to show the effectiveness of the proposed method. In experiments 1, the robot developmentally learned the HBKNs of a ball or toy car, respectively, through interaction with a human. In experiment 2, the robot was in a confusing situation with both objects, and solved it using the HBKNs learned in the previous experiments.

#### *A. Experiment 1 – Human and Robot Playing with a Ball or a Toy car*

In this experiment, first a human and a robot played together with a ball. Figure 4 shows key snapshots of the experimental results, and Figure 5 shows the incrementally developed HBKN of the ball along with the updated transition probabilities between behavior nodes. Figure 6 shows the updated utility values of the behavior nodes for the HBKN of the ball, and Figure 7 shows the *UL* and *IF* of the robot during the experiment.

First, a virtual ball appeared in front of the virtual teddy bear robot (Fig. 4(a)) and the *IF* of the ball increased, which caused the robot to pay attention to the ball. Because it was the first time the robot had perceived the ball, the robot created a new initial HBKN of the ball (Fig. 5(a)), and the utility values of all behavior nodes were zero (Fig. 6(a)).

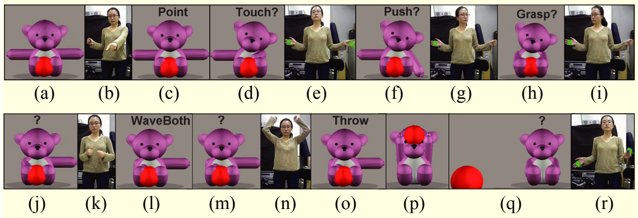


Fig. 4. Key snapshots of experiment 1.

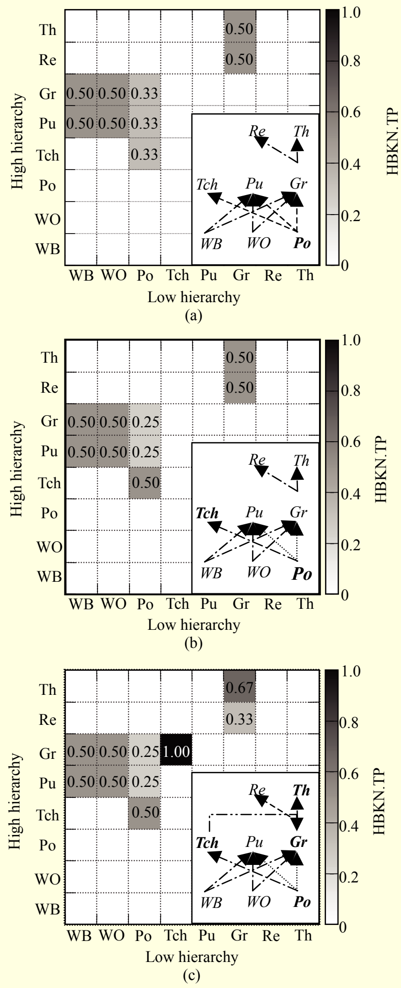


Fig. 5. *HBKN*(ball). TP and the developed HBKN structure for the ball at each step of experiment 1. The gray cells represent the transition probabilities from lower level behavior nodes to higher level behavior nodes. Transitions for (a) Figs. 4(a)–(d), (b) Figs. 4(e)–(n), and (c) Figs. 4 (o)–(r).

The human participant performed behavior *Po* to show her intention, that is, "give me the ball" (Fig. 4(b)), and the *IF* of the human increased. The *IF* of the human changed continually according to her behaviors, as shown in Fig. 7. The robot recognized the human arm behavior correctly (Fig. 4(c)) because the *FDTW* value of *Po* was the lowest. The reward of node *Po* was *HBKN.R*(*Po*) = 1, and the robot learned HBKN

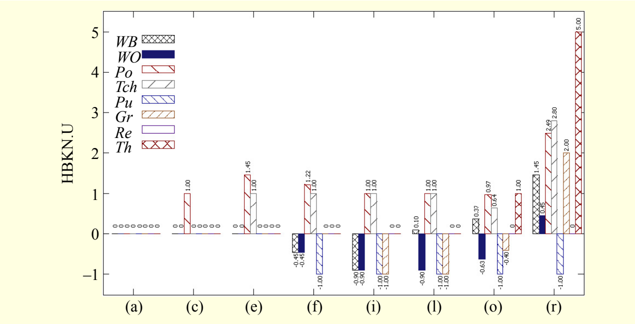

Fig. 6. Utility values of behavior nodes *HBKN*(ball). *U* at each step of experiment 1.

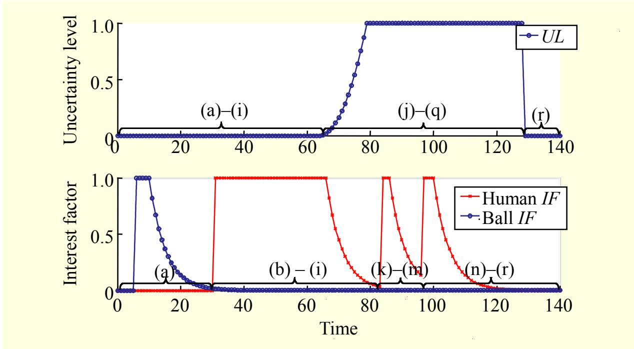

Fig. 7. Uncertainty level and interest factor of experiment 1. The letters indicate the subfigures of Fig. 4.

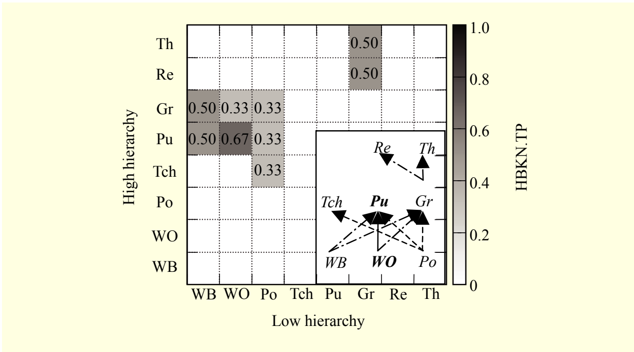

Fig. 8. *HBKN*(toycar). TP and the developed final HBKN for the toy car. The gray cells represent the transition probabilities from lower level behavior nodes to higher level behavior nodes.

as *HBKN.U*(*Po*) = 1 (Fig. 6(c)).

There were three possible next behavior nodes from node *Po*, and they had the same *HBKN.TP* and *HBKN.U*. Therefore, the robot performed trial behaviors for the *Tch*, *Pu*, and *Gr* nodes to obtain human feedback (Figs. 4(d)–(i)). Because she gave positive feedback for the trial behavior of *Tch* and negative feedback for the other trial behaviors, the reward of each behavior node was *HBKN.R*(*Tch*) = 1 and *HBKN.R*(*Pu*) = *HBKN.R*(*Gr*) = –1. The robot learned HBKN by updating the utilities and transition probabilities using the proposed algorithm (Fig. 5(b) and Figs. 6(e), (g), and (i)). Note that *HBKN.TP*(*Po*, *Tch*) was updated as 0.5 because node *Po* had the biggest utility of the behavior nodes in the class below *Tch*. Because *HBKN.TP*(*Po*, *Tch*) became 0.5, *HBKN.TP*(*Po*, *Pu*) and *HBKN.TP*(*Po*, *Gr*) were updated to 0.25.

Because there were no more possible next behavior nodes from node *Tch*, *UL* was increased (Fig. 7) and the robot asked for more information when it exceeded $U_0$ (Fig. 4(j)). The human performed behavior *WB* to show the same intention (Fig. 4(k)) and the robot recognized it correctly again (Fig. 4(l)). The reward of node *WB* was then *HBKN.R*(*WB*) = 1 and the utility values of the HBKN were updated (Fig. 6(l)). Because the *CBKL* was not updated from the previous one, *UL* still remained one and the robot asked for more information again (Fig. 4(m)).

The human performed behavior *Th* to show her intention, that is, "throw the ball to me" (Fig. 4(n)), and the robot recognized it correctly (Fig. 4(o)). The reward of node *Th* was *HBKN.R*(*Th*) = 1, and the robot learned the HBKN (Fig. 5(c) and Fig. 6(o)). Note that a new transition *HBKN.TP*(*Tch*, *Gr*) = 1 was created and *HBKN.TP*(*Gr*, *Th*) was updated to 0.67 because the current behavior node *Th* had only one previously connected lower class behavior node, that is, *Gr*, and *Tch* had the maximum utility of all the behavior nodes in the class below *Th*. Because *HBKN.TP*(*Gr*, *Th*) became 0.67, the human intention as "throwing the ball to her" and waited for her feedback for confirmation (Figs. 4(p) and (q)). She gave the right intent feedback (Fig. 4(r)). The reward for *Th* was then *HBKN.R*(*Th*) = 5, and the robot learned the final HBKN of the ball (Figs. 5(c) and 6(r)).

Note that a human and a robot also played together with a toy car in the same manner as a ball. Figure 8 shows the final HBKN of the toy car along with the updated transition probabilities between behavior nodes. As shown in Fig. 8, HBKN of the toy car had different behavior hierarchies compared to HBKN of the ball.

### B. Experiment 2 – Human and Robot Playing with a Ball and Toy Car

In this experiment, a human and a robot played together with both the ball and toy car. Figure 9 shows the key snapshots of the experiment. Figure 10 shows the *UL* and *IF* of the robot during the experiment.

First, both the ball and the toy car appeared in front of the robot (Fig. 9(a)). As the robot had already interacted with both of them, it did not need to create new HBKNs, so it loaded the previously developed HBKNs of the ball and the toy car. Because the *IF*s of both objects increased (Fig. 10), the robot was confused regarding which object it should pay attention to, and it needed more information. Therefore, *UL* increased

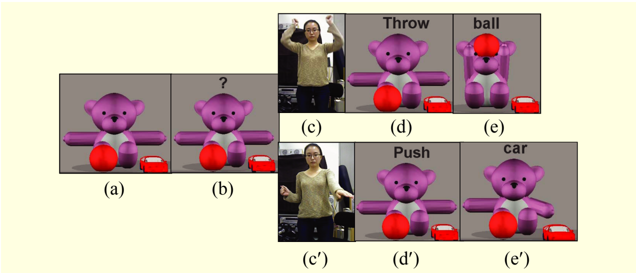


Fig. 9. Key snapshots of experiment 2.

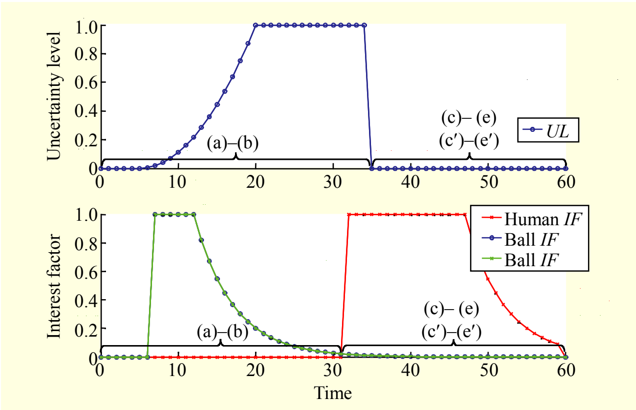


Fig. 10. Uncertainty level and interest factor of experiment 2. The letters indicate the subfigures of Fig. 9.

(Fig. 10) and the robot asked for more information when it exceeded $U_0$ (Fig. 9(b)). The robot tried to infer the *HIO* and human intention using the proposed algorithm. The *CBKL* of each object from HBKNs was {*Po*, *Tch*, *Gr*, *Th*} for the ball and {*WO*, *Pu*} for the toy car.

There were two cases according to the human feedback. In the first case, the human participant performed behavior *Th*, the robot recognized it correctly (Figs. 9(c) and (d)), and *UL* decreased because the robot obtained the new information (Fig. 10). Because behavior *Th* existed only in the ball *CBKL*, the robot inferred that the *HIO* was the ball rather than the toy car and that the *HI* was "throwing the ball" (Fig. 9(e)). In the other case, the human participant performed behavior *Pu*, the robot recognized it correctly (Fig. 9(c')(d')), and *UL* decreased because the robot obtained the new information (Fig. 10). Because behavior *Pu* existed only in the toy car *CBKL*, the robot inferred that the HIO was the toy car rather than the ball and that the human intention was "pushing the toy car" (Fig. 9(e')).

### 3. Human Intention Reading Results for Multiple Participants

Experiments with multiple human participants were conducted to determine whether the proposed algorithm would work well with people who were not involved in developing the system. Each participant played with the robot using the two objects, a ball and toy car, and we asked the participants to think of different intentions for them before the experiments were conducted.

The robot inferred each participant's intention by learning the participant's own HBKNs of the ball and the toy car using the proposed algorithm. Table 1 shows the human intention reading results for multiple participants. Tables 1(a) and (b) show the participants' intentions, their behavior sequences for interaction with the robot in the experiments, the final utilities of the behavior nodes from the learned HBKNs, and the intention reading results when playing with the ball and the toy car, respectively.

As shown in the tables, the robot inferred the participants' intentions correctly in all cases except one. The incorrect case occurred when participant 2 performed a behavior (*Pu*) that differed from his intention (to grasp the toy car). In this case, the robot had learned the participant's personalized HBKN for the toy car based on the recognized *Pu* behavior. Using the learned HBKN, the robot inferred the human intention as pushing the toy car because the participant gave it incorrect input. After the robot pushed the toy car, the object left the robot, and the experiment ended for the first trial. The participant realized why the robot did not infer his intention correctly and the robot inferred his intention correctly in the second trial.

After each participant played with the robot using both objects, the third experiment (the confusing situation with two objects) was conducted. At this point, the robot had learned each participant's HBKNs for the ball and toy car from the previous experiments. Table 1(c) shows each participant's intention, interaction behavior sequence, and intention reading result. By using the proposed algorithm and learned HBKNs, the robot inferred all of the participants' intended objects and intentions correctly.

## VI. Conclusion

This paper proposed an interactive human intention reading method based on developmental knowledge. In this system, the robot develops its own knowledge by learning an HBKN through interactions with a human. The HBKN was proposed to learn the hierarchical behavior structures for different objects and the robot can infer the human intention for an object by using learned HBKNs. The proposed human intention reading algorithm incorporates passive reinforcement learning with adaptive dynamic programming so that it learns HBKNs interactively using context information and human feedback through human behaviors. The effectiveness of the proposed method was demonstrated through play-based experiments between a human subject and virtual teddy bear robot with two

Table 1. Human intention reading results for multiple participants: (a) ball object, (b) toy car object, and (c) both objects.

(a)

| No. | Intention | Interaction behavior sequence | Final U | Result |
|---|---|---|---|---|
| P1 | Push | *Pu*, *HI*(*Pu*) | (3.00 3.00 2.25 0.00 5.00 0.00 0.00 0.00) | Right |
| P2 | Throw | *Th*, *HI*(*Th*) | (1.35 1.35 0.90 0.00 0.00 3.00 0.00 5.00) | Right |
| P3 | Push | *Tch*, *Tch*, *Pu*, *Pu*, *HI*(*Pu*) | (2.25 2.25 2.31 1.00 5.00 0.00 0.00 0.00) | Right |
| P4 | Throw | *Th*, *HI*(*Th*) | (1.35 1.35 0.90 0.00 0.00 3.00 0.00 5.00) | Right |
| P5 | Throw | *Tch*, *Gr*, *Th*, *HI*(*Th*) | (1.80 1.80 1.80 1.00 0.00 4.00 0.00 5.00) | Right |
| P6 | Release | *Tch*, *Gr*, *Re*, *HI*(*Re*) | (1.80 1.80 1.80 1.00 0.00 4.00 5.00 0.00) | Right |
| P7 | Grasp | *WO*, *Neg*(*Pu*), *Pos*(*Gr*), *Gr*, *HI*(*Gr*) | (1.80 4.15 1.20 0.00 –1.00 5.00 0.00 0.00) | Right |
| P8 | Throw | *Gr*, *Th*, *HI*(*Th*) | (2.40 2.40 1.80 0.00 0.00 4.00 0.00 5.00) | Right |
| P9 | Throw | *Th*, *HI*(*Th*) | (1.35 1.35 0.90 0.00 0.00 3.00 0.00 5.00) | Right |
| P10 | Throw | *Po*, *Neg*(*Tch*), *Neg*(*Pu*), *Neg*(*Gr*), *Po*, *WB*, *Tch*, *Th*, *HI*(*Th*) | (1.45 0.45 2.49 2.80 –1.00 2.00 0.00 5.00) | Right |

(b)

| No. | Intention | Interaction behavior sequence | Final U | Result |
|---|---|---|---|---|
| P1 | Throw | *Gr*, *Th*, *HI*(*Th*) | (2.40 2.40 1.80 0.00 0.00 4.00 0.00 5.00) | Right |
| P2 | Grasp | 1st: *Tch*, *Pu* | (0.45 0.45 0.72 1.00 1.00 0.00 0.00 0.00) | Wrong |
| | | 2nd: *Tch*, *Gr*, *HI*(*Gr*) | (2.25 2.25 2.16 1.00 0.00 5.00 0.00 0.00) | Right |
| P3 | Release | *Tch*, *Re*, *HI*(*Re*) | (1.35 1.35 2.34 3.70 0.00 3.00 5.00 0.00) | Right |
| P4 | Push | *Gr*, *Pu*, *HI*(*Pu*) | (2.70 2.70 1.58 0.00 5.00 1.00 0.00 0.00) | Right |
| P5 | Push | *Gr*, *Pu*, *HI*(*Pu*) | (2.70 2.70 1.58 0.00 5.00 1.00 0.00 0.00) | Right |
| P6 | Push | *WB*, *Neg*(*Pu*), *Neg*(*Gr*), *Tch*, *Pu*, *HI*(*Pu*) | (3.25 1.80 1.50 1.00 5.00 –1.00 0.00 0.00) | Right |
| P7 | Release | *WB*, *Neg*(*Pu*), *Pos*(*Gr*), *Re*, *HI*(*Re*) | (3.10 1.35 0.90 0.00 –1.00 4.00 5.00 0.00) | Right |
| P8 | Push | *WO*, *Neg*(*Pu*), *Pos*(*Gr*), *WB*, *Pu*, *HI*(*Pu*) | (3.70 3.70 1.80 0.00 5.00 1.00 0.00 0.00) | Right |
| P9 | Push | *Po*, *Neg*(*Tch*), *Neg*(*Pu*), *Pos*(*Gr*), *Pu*, *HI*(*Pu*) | (2.70 2.70 2.98 –1.00 5.00 1.00 0.00 0.00) | Right |
| P10 | Push | *Pu*, *HI*(*Pu*) | (3.00 3.00 2.25 0.00 5.00 0.00 0.00 0.00) | Right |

(c)

| No. | Intention | Interaction behavior sequence | Result |
|---|---|---|---|
| P1 | Push the ball | *WB*, *WB*, *Pu* | Right |
| P2 | Throw the ball | *Th* | Right |
| P3 | Push the ball | *Pu* | Right |
| P4 | Throw the ball | *Th* | Right |
| P5 | Throw the ball | *Th* | Right |
| P6 | Push the car | *Tch*, *Pu* | Right |
| P7 | Release the car | *WB*, *Po*, *Gr*, *Re* | Right |
| P8 | Throw the ball | *Th* | Right |
| P9 | Push the car | *Po*, *Pu* | Right |
| P10 | Throw the ball | *Po*, *Th* | Right |

virtual objects, a ball and a toy car. Three different scenarios demonstrated that the robot could correctly infer human intentions using the proposed method, even in confusing situations. In addition, experiments with multiple participants who were not involved in developing the proposed method were conducted, and the robot could infer all the participants' intentions successfully except in the case in which a participant gave incorrect behavior as an input.

## References

[1] M.M. Botvinick, "Hierarchical Models of Behavior and Prefrontal Function," *Trends Cognitive Sci.*, vol. 12, no. 5, May 2008, pp. 201–208.

[2] L. Zhang and D. Zhang, "Visual Understanding via Multi-feature Shared Learning with Global Consistency," *IEEE Trans. Multimedia*, vol. 18, no. 2, Feb. 2016, pp. 247–259.

[3] L. Zhang et al., "LSDT: Latent Sparse Domain Transfer Learning for Visual Adaptation," *IEEE Trans. Image Process.*, vol. 25, no. 3, Mar. 2016, pp. 1177–1191.

[4] P. Carruthers and P.K. Smith, "Theories of Theories of Mind," New York, USA: Cambridge University Press, 1996.

[5] J. Gray et al., "Action Parsing and Goal Inference Using Self as Simulator," *IEEE Int. Workshop Robot Human, Interactive Commun.*, Nashville, TN, USA, Aug. 13–15, 2005, pp. 202–209.

[6] C. Breazeal et al., "An Embodied Cognition Approach to Mindreading Skills for Socially Intelligent Robots," *Int. J. Robot. Res.*, vol. 28, no. 5, May 2009, pp. 656–680.

[7] B. Jansen and T. Belpaeme, "A Computational Model of Intention Reading in Imitation," *Robot. Auton. Syst.*, vol. 54, May 2006, pp. 394–402.

[8] L.M. Hiatt et al., "Accommodating Human Variability in Human-Robot Teams through Theory of Mind," *Proc. Int. Joint Conf. AI*, Barcelona, Spain, July 16–22, 2011, pp. 2066–2071.

[9] O.C. Schrempf et al., "A Novel Approach to Proactive Human-Robot Cooperation," *Proc. IEEE Int. Workshop Robot Human Interactive Commun.*, Nashville, TN, USA, Aug. 13–15, 2005, pp. 555–560.

[10] A.J. Schmid et al., "Proactive Robot Task Selection Given a Human Intention Estimate," *Proc. IEEE Int. Symp. Robot Human Interactive Commun.*, Jeju, Rep. of Korea, Aug. 26–29, 2007, pp. 726–731.

[11] R. Kelley et al., "Context-Based Bayesian Intent Recognition," *IEEE Trans. Auton. Mental Develop.*, vol. 4, no. 3, Sept. 2012, pp. 215–225.

[12] Z. Wang et al., "Probabilistic Movement Modeling for Intention Inference in Humanrobot Interaction," *Int. J. Robot. Res.*, vol. 32, no. 7, 2013, pp. 841–858.

[13] E. Bicho et al., "Integrating Verbal and Nonverbal Communication in a Dynamic Neural Field Architecture for Human-Robot Interaction," *Frontiers Neurorobot.*, vol. 4, no. 5, 2010.

[14] E. Bicho et al., "Neuro-Cognitive Mechanisms of Decision Making in Joint Action: A Human-Robot Interaction Study," *Human Movement Sci.*, vol. 30, no. 5, Oct. 2011, pp. 846–868.

[15] K. Strabala et al., "Learning the Communication of Intent Prior to Physical Collaboration," *Proc. IEEE RO-MAN*, Paris, France, Sept. 9–13, 2012, pp. 968–973.

[16] R. Kelley et al., "Deep Networks for Predicting Human Intent with Respect to Objects," *Proc. Annu. ACM/IEEE Int. Conf. Human-Robot Interaction*, Boston, MA, USA, Mar. 5–8, pp. 171–172.

[17] Z. Yu and M. Lee, "Human Motion Based Intent Recognition Using a Deep Dynamic Neural Model," *Robot. Auton. Syst.*, vol. 71, Sept. 2015, pp. 134–149.

[18] J.-H. Han and J.-H. Kim, "Consideration about the Application of Dynamic Time Warping to Human Hands Behavior Recognition for Human-Robot Interaction," in *Robot Intell. Technol. Appl.*, Switzerland: Springer International Publishing, 2013, pp. 269–277.

[19] R.E. Bellman, "Dynamic Programming," Princeton, NJ, USA: Princeton University Press, 1957.


**Ji-Hyeong Han** received the BS and PhD degrees in electrical engineering from Korea Advanced Institute of Science Technology, Daejeon, Rep. of Korea, in 2008 and 2015, respectively. Since 2015, she has been with ETRI, Daejeon, where she is currently a Researcher. Her research interests include human intent recognition, human–robot interaction, intelligent robotics, and smart factory.

**Seung-Hwan Choi** received the BS degree in computer science and the PhD degree in electrical engineering from KAIST in 2005 and 2013, respectively. Since 2013, he has been with UX Center, Corporate Design Center, Samsung Electronics, Seoul, Rep. of Korea, where he is currently a Senior Designer.

**Jong-Hwan Kim** received the PhD degree in electronics engineering from Seoul National University, Rep. of Korea, in 1987. Since 1988, he has been with the School of Electrical Engineering, KAIST, where he is leading the Robot Intelligence Technology Laboratory as Professor. Dr. Kim is the Director for KoYoung-KAIST AI Joint Research Center. His research interests include InT (Intelligence Technology), intelligent interactive technology, ubiquitous and genetic robots, and humanoid robots. He has authored 5 books and 5 edited books, 2 journal special issues and around 400 refereed papers in technical journals and conference proceedings. He has delivered over 200 invited talks on computational intelligence and robotics including over 50 keynote speeches at the international conferences. He was an Associate Editor of the IEEE Transactions on evolutionary computation and the IEEE computational intelligence magazine.