Inferring other's intention based on estimated state value of self

Yasutake Takahashi, Teruyasu Kawamata, Tom Tamura, and Minoru Asada
Dept. of Adaptive Machine Systems,
Graduate School of Engineering, Osaka University
Handai FRC
Yamadaoka 2-1, Suita, Osaka, 565-0871, Japan
{yasutake, kawamata, tamura, asada}@er.ams.eng.osaka-u.ac.jp

Abstract

Recognition of other agent intention in a multi-agent environment is a very important issue to realize social activities, for example, imitation learning, understanding intention, cooperative/competitive behavior, and so on. Conventional approaches to infer the other agent intention need a precise trajectory in Cartesian or joint space that is sometimes hard to measure from the viewpoint of an observer. It is also difficult to estimate a same intention but with different realizations because they try to match just a certain trajectory during the trial. We propose a novel method of inference of other agent's intention based on state value estimation. The method does not need a precise world model or coordination transformation system to deal with view dependency. This paper shows an observer can infer an intention of other not by precise object trajectory in Cartesian space but by estimated state value transition during the observed behavior.

1 Introduction

Inference of others' intentions what they like to do is one of the most formidable issues in multi-agent systems in which actions appropriate for the others' intentions are needed to accomplish the cooperative tasks. For example Schaal et al. [4] proposed a motor learning method through imitation of teacher's behaviors. They assume that a learner can observe all state variables and their trajectories in Cartesian coordinate system of the environment or the joint space of the others and the learner imitates manipulative tasks or gestures. Doya et al. [2] proposed to estimate intention of other agent for imitation learning and/or cooperative behavior acquisition based on multi-module learning system. Takahashi et al. [6] proposed a method that interprets instruction given by a coach and divides the given complicated task to a number of simple sub-tasks each of which can be learned with a simple behavior learning module with limited capability. Most existing approaches assume the detailed knowledge of the task, the environment, and the others (their body structure and sensor/actuator configuration) based on which they can transform the observed sensory data of the others' behaviors into the Cartesian coordinate system of the environment or the joint space of the others to infer their intentions. However, such an assumption seems unrealistic in the real world and brittle to the sensor/actuator noise(s) or any possible changes in the parameters. In other words, it is very difficult to infer others' intentions based only on these geometric parameters.

On the other hand, another approach that estimates behavior of others through observer's viewpoint without any coordination conversion has been proposed, too. Ledezma et al. [3] proposed to make a classifier to label other agent's behavior based on observation and use this classifier to label the behavior. Their method, however, needs a full teaching data of a set of labels and sequence observation in order to model the other agent actions and cannot handle the change the sequence of the other agent's actions even if it does the same task. Takahashi et al. [5] presented an approach that constructs a set of state transition models for the opponent behaviors from a viewpoint of observer and selects an appropriate behavior for observer according to a current situation in which one of the models matches. The observer can choose one model according to the other agent's behavior, however, it cannot infer the intention of the other agent.

Recently, reinforcement learning has been studied well for motor skill learning and robot behavior acquisition. It generates not only an appropriate policy (map from states to actions) to achieve a given task but also an estimated discounted sum of reward value that will be received in future while the robot is taking the optimal policy. We call this estimated discounted sum of reward "state value." This state value roughly indicates closeness to a goal state of the given task, that is, if the agent is getting closer to the goal, the state value becomes higher. This suggests that the observer may understand which goal the agent likes to achieve if the state value of

the corresponding task is going higher.

The relationship between an agent and objects such that the agent gets close to the object or the agent faces to a direction is much easier to understand from the observation, and therefore such qualitative information should be utilized to infer what the observed agent likes to do. The information might be far from precise ones, however, it keeps topological information and we can acquire good estimation of temporal difference of state value with this method.

Then, we propose a novel method to apply the above idea to infer the others' intentions supposing that the observer has already estimated the state values of all kinds of tasks the observed agent can do. The method does not need a precise world model or an accurate coordination transformation system to cope with the problem of view dependency. We apply the method to a simple RoboCup situation where the agent has kinds of tasks such as navigation, shooting a ball into a goal, passing a ball to a teammate, and so on, and the observer judges which task the agent is now achieving from the observation with estimated state values. The preliminary experiments are shown and future issues are discussed.

2 Intention Inference by State Value Estimation

In this section, a rough description of state value function and behavior inference is described. We assume that the observer has already acquired a number of behaviors based on a reinforcement learning method. Each behavior module can estimate state value at arbitrary time t to accomplish the specified task. Then, the observer watches the performer's behavior and maps the sensory information from an observer viewpoint to the agent's one with a mapping of state variables. The behavior modules estimate the state value of the observed behavior and the system selects ones that matches estimation of state value.

2.1 State Value Function

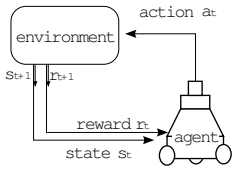


Figure 1: A basic model of agent-environment interaction

Fig.1 shows a basic model of reinforcement learning. An agent can discriminate a set S of distinct world states. The world is modeled as a Markov process, making stochastic transitions based on its current state and the action taken by the agent based on a policy π. The agent receives reward r_t at each step t. State Value V^π, discounted sum of the reward received over time under

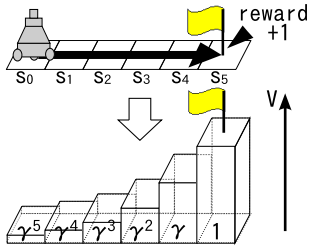


Figure 2: Sketch of state value propagation

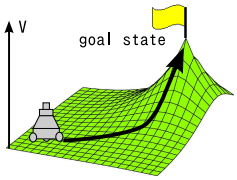


Figure 3: Sketch of a state value function

execution of policy π, will be calculated as follows:

$$V(s) = \sum_{t=0}^{\inf} \gamma^t r_t \quad . \tag{1}$$

Figs.2 and 3 show sketches of a state value function where a robot receives a positive reward when it stays at a specified goal while zero reward else. The state value will be highest at the state where the agent receives a reward and discounted value is propagated to the neighbors states (Fig.2). As a result, the state value function seems to be a mountain as shown in Fig.3. The state value becomes bigger and bigger if the agent follows the policy π.

2.2 Basic Idea of Intention Recognition

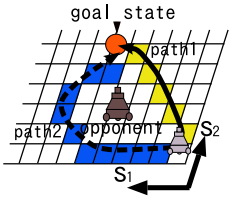


Figure 4: Sketch of different behaviors in a grid world

Fig.4 shows an example task of navigation in a grid world. There is a goal state at the top center of the world. An agent can move one of the neighbor grids every one-step. It receives a positive reward only when it stays at the goal state while zero else. There are various optimal policies for this task as shown in Fig.4. If one tries to match the action that the agent took and the one based on a certain policy in order to infer the agent's intention, you have to maintain various optimal policies and evaluate all of them in the worst case.

On the other hand, if the agent follows an optimal policy, the state value is going up even if the agent takes

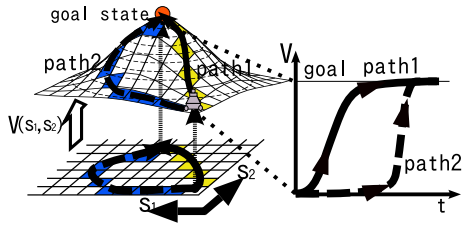


Figure 5: Inferring intention by the change of state value

an arbitrary policy from the optimal ones. Fig.5 shows that the state value becomes larger even if the agent takes different paths. We can regard that the agent takes an action based on one policy when the state value is going up even if it follows various kind of policies.

This indicates a possibility of robust intention recognition even if they would be several optimal policies for the current task. An agent tends to acquire various policies depending on the experience during learning. The observer cannot practically estimate the agent's experience beforehand, therefore, it needs a robust intention recognition method provided by the estimation of state values.

2.3 Modular Learning System

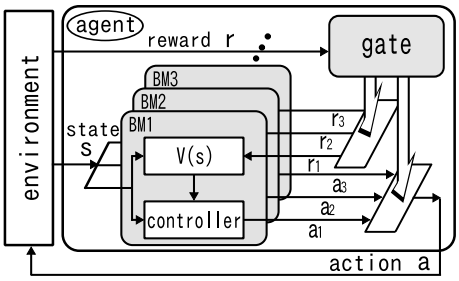


Figure 6: Modular Learning System

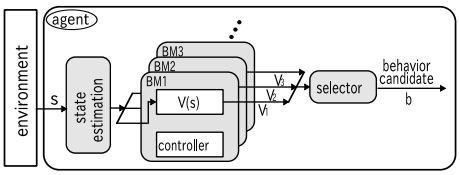


Figure 7: Behavior inference diagram

In order to evaluate a number of behaviors simultaneously, we adopt a modular learning system. Jacobs and Jordan [1] proposed a mixture of experts, in which a set of the expert modules learn and the gating system weights the output of each expert module for the final system output. Fig.6 shows a sketch of such a modular learning system. We prepare a number of behavior modules each of which acquired a state value function for one goal-oriented behavior. A learning module has a controller that calculates an optimal policy based on the state value function. Gating module selects one output from a module according to the agent's intention.

2.4 Intention Inference under Multiple Candidates

At intention inference stage, the system uses same behavior modules as shown in Fig.7. While an observer watches an behavior of a performer, the system estimates the relationship between the agent and objects such as rough direction and distance of the objects from the agent. Then, each behavior module estimates the state value based on the rough estimated state of the agent and sends it to the selector. The selector watches the sequence of the state values and selects a set of possible behavior modules of which state values are going up as the performer is taking the behavior. As mentioned in 2.1, if the state value goes up during a behavior, it means the module seems valid for explaining the executing behavior execution. The goal state/reward model of this behavior module represents the intention of the agent.

Here we define reliability g that indicates how much the intention inference would be reasonable for the observer as follow:

$$g = \begin{cases} g + \beta & \text{if } V(s_t) - V(s_{t-1}) > 0 \text{ and } g < 1 \\ g & \text{if } V(s_t) - V(s_{t-1}) = 0 \\ g - \beta & \text{if } V(s_t) - V(s_{t-1}) < 0 \text{ and } g > 0 \end{cases}$$

where β is an update parameter, which is 0.1 in this paper. This equation indicates that the reliability g will become large if the estimated state value rises up and it will become low when the estimated state value goes down. We put another condition in order to keep g value from 0 to 1.

3 Task and Environment

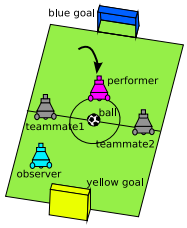


Figure 8: Environment

Figure 9: A real robot

Fig.8 shows a situation the agents are supposed to encounter. An agent shows a behavior and the observer estimates the behavior using a set of behavior modules of its own. Fig.9 shows a mobile robot we have designed and built. Fig.10 shows the viewer of our simulator for our robots and the environment. The robot has a normal perspective camera in front of its body. It has an omni-directional camera, however, it doesn't use it, here. A simple color image processing is applied to detect the ball, the interceptor, and the receivers on the image in

Table 1: Prepared Modules and their state variables

Module	State variables
GoToBall	ball position y on the image of perspective camera
GoToYellow	yellow goal position y on the image of perspective camera
GoToBlue	blue goal position y on the image of perspective camera
ShootYellow	ball position y, yellow goal position y, and angle between them θ on the image
ShootBlue	ball position y, blue goal position y, and angle between them θ on the image
PassToTeammate1	ball position y, teammate 1 position y, and angle between them θ on the image
PassToTeammate2	ball position y, teammate 2 position y, and angle between them θ on the image

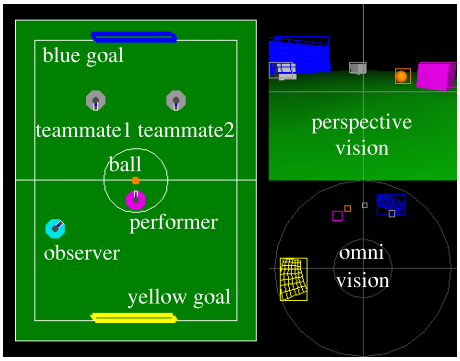


Figure 10: Viewer of simulator

real-time (every 33ms). The left of Fig.10 shows a situation the agent encounters while the top right images show the simulated ones of the normal and the bottom right omni vision systems. The mobile platform is an omni-directional vehicle (any translation and rotation on the plane). Table 1 shows a list of prepared behavior modules and their state variables. The observer has learned the behaviors and its state value estimator based on a reinforcement learning method beforehand.

3.1 State Variables and Estimation

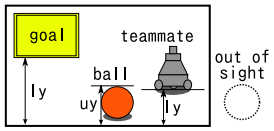


Figure 11: State variables representing distances to the objects

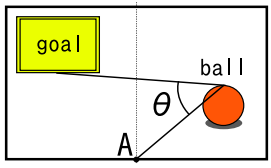


Figure 12: A state variable θ representing the positional relationship between the objects

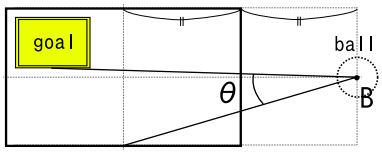


Figure 13: A state variable θ when one of the objects is out of sight

We use the distances of the ball, goal, and player from the agent and their relative angles between them on the image of the frontal camera on the robot. Figs.11 and 12 show examples of those state variables. We divide this state space into a set of region to obtain state id. The space of position value is quantized into 6 subspaces and the space of relative angle between objects into 5 spaces here. Behavior modules define their policy and state value function in this state space.

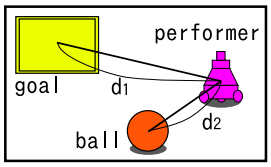


Figure 14: Estimated state variables representing distances

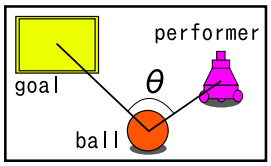


Figure 15: An estimated state variable θ representing the position relation among objects

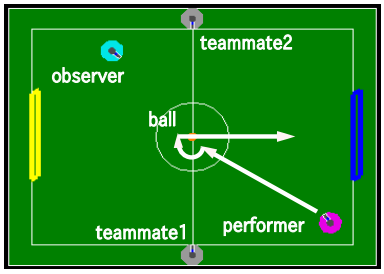
When an observer infers an intention of the performer, it has to estimate its state. Here, we introduce a simple method of state estimation of the performer. Fig.14 shows the estimated distances from the agent and the objects. Fig.15 shows the estimated angle between the objects. The observer uses these estimated state for estimation of state value instead of its own state shown in Figs.11 and 12. These estimated states with this method

are far from precise ones, however, it keeps topological information and we can acquire good estimation of temporal difference of state value with this method.

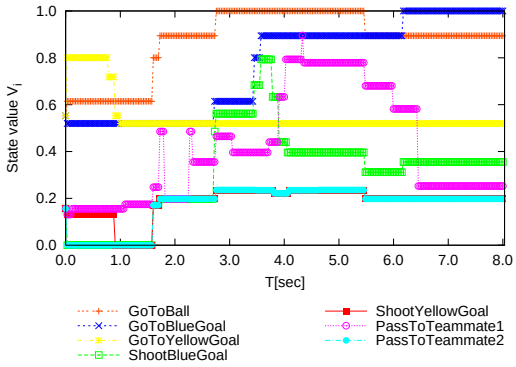
4 Experiments

The observer has learned a number of behaviors shown in Table 1 before it tried to infer the performer's intention. We gave the observer many experiences enough to cover all exploration space in state space.

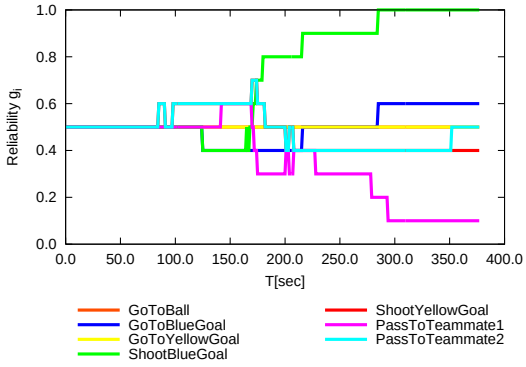
4.1 Same behavior demonstration



(a) An overview of the behavior



(b) State value V_i of each behavior module

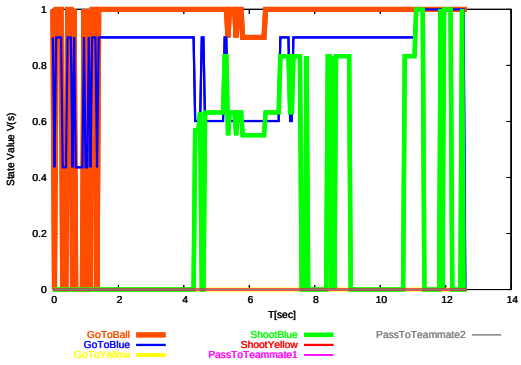


(c) Reliability g_i of each behavior module

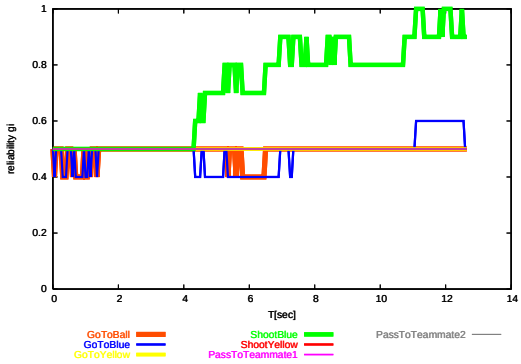
Figure 16: Inferring intention of a performer trying to shoot a ball to the blue goal

After the behavior acquisition, we let the performer play one of the behaviors from Table 1 and the observer infers which behavior the other is taking. Fig.16 shows an example behavior performed by another agent. The performer showed exactly same behavior that the observer acquired in behavior learning stage, here. The bottom right agent shows "ShootBlue" behavior and the top left observer watches the behavior. The observer

(a) An overview of the behavior



(b) State value V_i of each behavior module



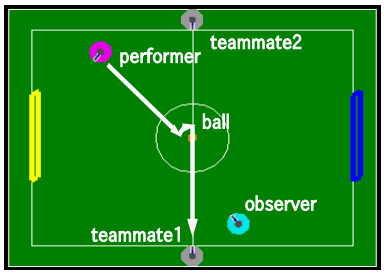
(c) Reliability g_i of each behavior module

Figure 17: Inferring intention of the performer trying to shoot ball to the blue goal in a real robot experiment

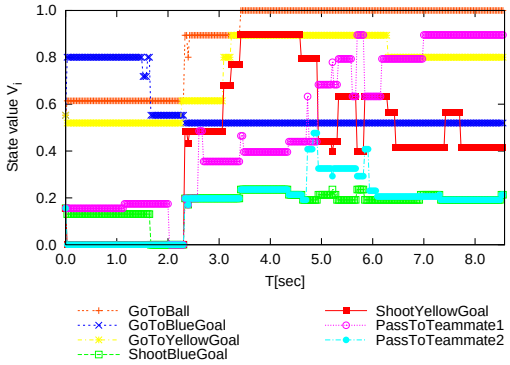
tries to keep the performer in own perspective during the behavior. Fig.16 (a) shows the sequence of the behavior. Fig.16 (b) and (c) show sequences of estimated state value and reliability of the inferred intentions of the agent, respectively. The green line indicates the behavior of shooting a ball into a blue goal and goes up during the trial. The observer successfully inferred the intention of the performer.

Figure 17 shows a result of inferring intention of the performer trying to shoot ball to the blue goal in a real robot experiment. The situation and the result are similar to the simulation and it shows successfully infer the intention of the performer.

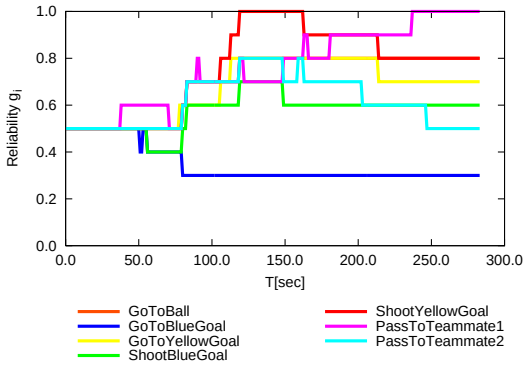
Fig.18 shows an example passing behavior performed by an agent, a sequence of estimated state value of each modules, and a sequence of reliability of inferred intention, respectively. These figures show that the observer



(a) An overview of the behavior

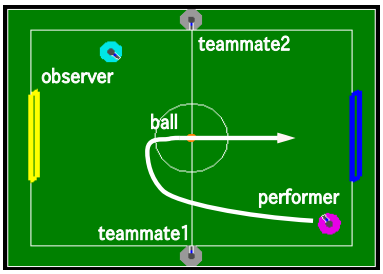


(b) State value V_i of each behavior module

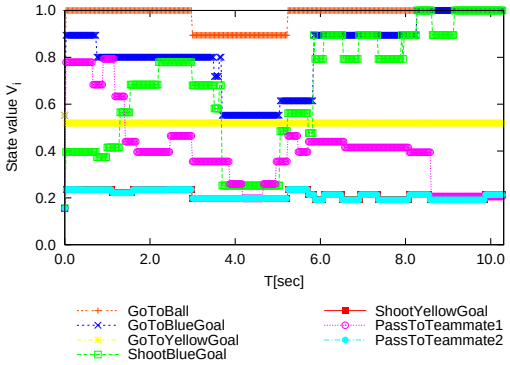


(c) Reliability g_i of each behavior module

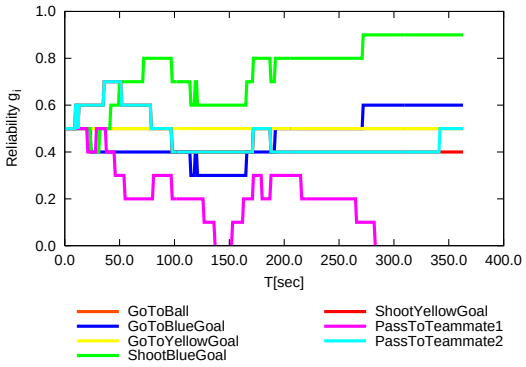
Figure 18: Inferring intention of a performer trying to pass a ball to teammate1



(a) An overview of the behavior



(b) State value V_i of each behavior module



(c) Reliability g_i of each behavior module

Figure 19: Inferring intention of a performer trying to shoot a ball to the blue goal with different manner

can infer the performer's passing behavior (purple line), too. The orange line indicates the reliability of going to a ball behavior and it also goes up during the trial because the agent is continuously approaching to the ball during the trial to pass it to the teammate.

4.2 Different behavior demonstration

The observer cannot assume that the performer will take exactly same behavior even if its intention is same. We prepare other different behaviors for the demonstration of the performer. Fig.19 (a) shows an example of the different shooting behavior demonstrated by the performer. The learned behavior by the observer is a zippy motion as shown in Fig.16. On the other hand, the demonstrated behavior is a more smooth motion. Therefore, the state transition probability will be different from each other. Figs.19 (b) and 19 (c) show sequences of estimated state value of each modules and a sequence of reliability of inferred intention, respectively. These figures show that the observer can infer the performer's intention of shooting (green line).

4.3 Comparison with System based on Coordination Translation

In this section, we compare performances between our proposed method and the one based on state estimation using coordinate transformation system and tracing state transition probability that is proposed by others, for example [2]. In order to estimate the state value of a behavior module through the observation of the performer, there must be a rough coordinate transformation matrix beforehand. Figs.20 and 21 show a rough sketch of the transformation system. In order to estimate y position on the performer's view image, the observer assumes there are tiles on the floor, maps the positions of an object and the performer, estimates rough distance between them, and maps the distance to the y position on the image of the observer's view. Fig.21 shows a

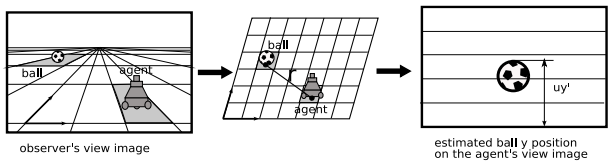


Figure 20: Estimation of y position of the performer's image

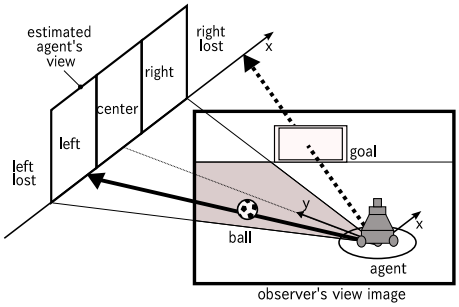


Figure 21: Estimation of direction from a performer to an object

sketch of estimation of direction from the performer to an object. The lower right rectangle shows an example image from the perspective camera and it captures the performer, a ball, and a goal. We assume that it can detect a direction of the performer on the image under a vision system. We put a potential image plane in front of the agent and estimate rough x positions of the objects on the image of the performer's view. Fig.21 shows that the ball is mapped to the left side on the image and the goal to an area of lost to the right side from the camera image. Table 2 shows the success rate

Table 2: Inferring intention performances of the proposed method and the one with coordination transformation system

	Proposed method	Method based on state trans. prob.
ShootBlueGoal1	84%	24%
ShootBlueGoal2	78%	11%
ShootYellowGoal	86%	20%
PassToTeammate1	76%	34%

of intention inference of the proposed method and the one with the coordination transformation system. The proposed method shows much better results over the behaviors than the one with the coordination transformation system. "ShootBlueGoal1" indicates a case of inference of shooting behavior identical to the observer's one. "ShootBlueGoal2" indicates a case of inference of shooting behavior but different from the observer's one.

5 Future work

This basic idea can be applied for not only intention inference but also cooperative behavior acquisition. How to define a reward function for cooperative behavior acquisition in multi-agent system is one of the most interesting issues. The proposed method can infer other's intention and estimate the reward/state value of the agent for each step. This indicates that the observer can explore some actions and evaluate how much they will contribute to the other efficiently. Then, it can learn cooperative behavior based on a certain reinforcement learning approach without any heuristic/hand-coded reward function by which it evaluates a reward of itself based on the estimated reward/state value of the other agent.

References

[1] R. Jacobs, M. Jordan, Nowlan S, and G. Hinton. Adaptive mixture of local experts. *Neural Computation*, 3:79–87, 1991.

[2] Doya K., Sugimoto N., Wolpert D.M., and Kawato M. Selecting optimal behaviors based on contexts. In *International Symposium on Emergent Mechanisms of Communication*, pages 19–23, 2003.

[3] Agapito Ledezma, Ricardo Aler, Araceli Sanchis, and Daniel Borrajo. Predicting opponent actions by observation. In D. Nardi et al., editor, *RoboCup2004*, pages 286–296. Springer-Verlag Berlin Heidelberg, 2005.

[4] Stefan Schaal, Auke Ijspeert, and Aude Billard. Computational approaches to motor learning by imitation, 2004.

[5] Yasutake Takahashi, Kazuhiro Edazawa, Kentarou Noma, and Minoru Asada. Simultaneous learning to acquire competitive behaviors in multi-agent system based on modular learning system. In *RoboCup 2005 Symposium papers and team description papers*, pages CD–ROM, Jul 2005.

[6] Yasutake Takahashi, Tomoki Nishi, and Minoru Asada. Self task decomp osition for modular learning system through interpretation of instruction by coach. In *RoboCup 2005 Symposium papers and team description papers*, pages CD–ROM, Jul 2005.