The Sixteenth International Symposium on Artificial Life and Robotics 2011 (AROB 16th '11),
B-Con Plaza, Beppu,Oita, Japan, January 27-29, 2011

On the Use of Human Instruction for Improving the Behavior of RoboCup Soccer Agents

Yosuke Nakamura and Tomoharu Nakashima

Osaka Prefecture University, 1-1 Gakuen-cho,Nakaku,Sakai,Osaka
(Tel : 81-72-254-9351; Fax : 81-72-254-9915)
(nakashi@cs.osakafu-u.ac.jp)

Abstract: In this paper, we propose a behavior generation approach from human instruction to improve the strategy of RoboCup soccer 3D simulation team. Many teams implement their strategies based on the programmers' own knowledge about soccer. That is, the programmers have to write action rules that cover any situations of the soccer field. Although it is clear that this is not the best approach, there are only a few research works that tackle this problem. In this paper, we solve this problem using human instruction to improve the manually implemented behavior of soccer robots. It is shown that the team performance is improved by the generated rules by this approach.

Keywords: RoboCup, soccer robot, learning, multi-agent system

I. INTRODUCTION

RoboCup is an international project which aims at building autonomous soccer robots. RoboCup has some main leagues such as Soccer, Rescue, @Home and Junior. We focus on the soccer simulation league, which is a subleagues of the RoboCup Soccer League. The soccer simulation league is one of the oldest leagues in the RoboCup competitions.

Fig. 1. 2D Simulation

There are two categories in the soccer simulation league. One is 2D league where all objects such as the ball, players, flags, and goal posts are modeled as a circle. The other is 3D league where humanoid robots with 22 degrees of freedom are autonomously controlled in a three-dimensional field. Figure 1 shows the snapshot of the 2D simulation game. In the 2D simulation league, all objects are realized in a two-dimensional space. This league is valuable as a test bed for high level decision making systems. There are many famous papers about 2D league. Gabel et al.[1] considered a defense scenario of crucial importance and employed a reinforcement learning methodology to autonomously acquire an aggressive duel behavior. Kyrylov and Hou [2] treated optimal defensive positioning as a multi-criteria assignment problem and demonstrated that pareto-optimal collaborative positioning yields good results. Kalyanakrishnan and Stone [3] introduced a policy search method for a keepaway task, which is a popular benchmark for multiagent reinforcement learning from the simulation soccer domain.

Fig. 2. 3D Simulation

On the other hand, the 3D simulation league includes the concept of height and can simulate the real world better than the 2D league. Figure 2 shows a game of the 3D simulation league. We can watch the game of the 3D soccer simulation league through the soccer monitor, which is included in the package of the soccer server [4]. The first prototype of the 3D soccer agent was proposed in 2003 [5]. In the early stage of the 3D simulation league, the soccer agents were modeled as a sphere object with a kick device. In 2007, a bipedal humanoid robot model was employed for soccer agents for the first time in the league. This made the development of soccer agents quite challenging because not only intelligent decision making but also low level skills such as the movement of joints have to be considered when developing the controller of the robot. Shafii et al.[6] employed a truncated fourier series approach for a stable biped walking of a humanoid robot and optimized it by using particle swarm optimization. Warden et al.[7] proposed a framework for spatio-temporal real-time analysis of dynamic scenes to improve the grounding situation of autonomous agents in physical domains. Recently low level skills have been significantly improved by top teams in the world. In addition, the number of agents in one team is increasing: one team had three agents in RoboCup 2009, and it was increased to six in RoboCup 2010. It will finally become 11 in the near future. Therefore, it is getting more important to implement team strategy to win a game. In this paper, we propose a method that generates action rules automatically from human instruction. A human instructor is expected to give more appropriate actions to the soccer agents. In the proposed method, the instructions are recorded and converted to action rules after selecting any useful instructions.

II. BEHAVIOR GENERATION USING HUMAN INSTRUCTION

1. Overview

Many teams implement their strategies based on the programmers' own knowledge about soccer. That is, the programmers have to write action rules that cover any situations of the soccer field. Although it is clear that this is not the best approach, there are only a few research works that tackle this problem. In this paper, we solve this problem using human instruction to improve the already implemented behavior of soccer robots. For this purpose, we developed a human-agent interface. In this system, a gamepad is used to send human instructions to agents. The human instructions are then converted to a set of action rules that are used to modify the behavior of the soccer robots. The process of our approach is the following:

i. Recording instructions.

ii. Reducing and clustering instructions.

iii. Generating action rules.

In the following subsection, we explain our approach in detail.

2. Recording instructions

The action of the soccer agent is semi-automatically determined. That is, the soccer agent has its own decision based on the sensory information. However, if the human instructor thinks that the action currently taken by the soccer agent is not appropriate, the action of the soccer agent is overruled by the instruction from the human instructor. Each time a human sends an instruction to the soccer agent, the instruction is recorded along

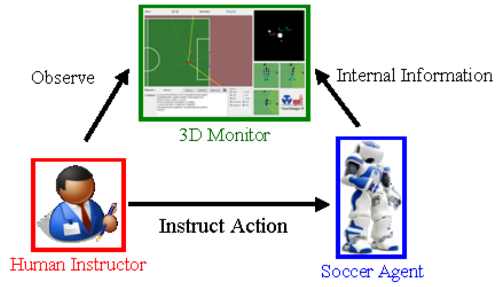

Fig. 3. Human instruction

Fig. 4. 3D Monitor for instruction

with sensory information that the soccer agent receives at that time. A user interface is used to monitor the sensory information that the soccer agent is currently receiving. The above process is graphically shown in Fig. 3. The 3D monitor in Fig. 3 is developed for the purpose of this paper. The snapshot of the 3D monitor is shown in Fig. 4. The 3D monitor allows human instructors to check the internal status of the soccer agents since the sensory information sent to the soccer agent is limited to the front area of its head.

When the human instructor sends an instruction to the soccer agent, the 3D monitor records it along with the internal status of the soccer agent. The internal status recorded with the action instruction consists of the positions of the ball and five soccer agents (three opponents, the other mate attacker, and itself). There are three actions available for the human instructors: kick, dribble, and wait. For the kick and the dribble actions, the human instructors also have to send the action direction. Although the human instructor can specify any direction for the two actions, the 3D monitor quantizes it into one of the eight directions such as up, down, right, left, up-right, up-left, down-right, down-left, and toward-opponent-goal. The nearest direction out of the nine to the specified one is selected and recorded in the 3D monitor.

3. Reducing and clustering instructions

Since a huge number of instructions are sent from humans during a match, it is not practical to use all the instructions. Also, some instructions are useful while others are not helpful for better strategies (e.g., scoring a goal). In this paper, we only use the recorded instructions that led to a goal while discarding the other instructions. Thus only helpful action rules are generated to improve the behavior of the soccer agent.

Since there are still a large number of instructions after removing not successful instructions, we apply a clustering method to compress the information contained in the instructions. We apply an incremental clustering method to the field status for each action. In the incremental clustering method, a pair of two instructions with the minimum distance in the field status space is combined and the average is used as the representative of the pair. This process is iterated until the number of clusters becomes a pre-specified number. In the computational experiments of this paper, we applied the clustering method to obtain 100 clusters (i.e., 100 representatives of the cluster) for each action. During the paring process, the distance between two clusters is measured as the minimum distance among all possible combinations of the elements within the clusters.

4. Generating action rules

The representatives of each cluster obtained in the previous subsection are converted into a set of action rules. As described in the previous subsection, only successful instructions leading to a score are converted to action rules after clustering. For each cluster center, an action rule of the following form is generated:

$$\begin{aligned} &\text{R: If the current status is } P \text{ then the action is } A,\\ &P\text{: } (x_{\text{self}}, y_{\text{self}}, x_{\text{ball}}, y_{\text{ball}}, x_{\text{opp1}}, y_{\text{opp1}},\\ &\qquad x_{\text{opp2}}, y_{\text{opp2}}, x_{\text{opp3}}, y_{\text{opp3}}, x_{\text{mate}}, y_{\text{mate}}) \quad (1)\\ &A\text{: Instructed action,} \end{aligned}$$

where P is the status of the field. The status of the field contains the positions of the opponent agents, the other mate agent, and the ball. The action in the consequent part of the action rule is the instruction that was specified by the human instructor.

III. PERFORMANCE EVALUATION

1. Experimental settings

In the computational experiments in this paper, a set of action rules are generated from human instructions in 3-on-3 soccer matches. That is, a team consists of three soccer agents (two attackers and a goal keeper). The strategy of the team is manually written beforehand. While the soccer agents autonomously play according to the written strategy, a human instructor can overrule the agent's action if the action is thought to be not appropriate for the instructor. In this paper, the action taken by the main attacker (i.e., the player nearest to the ball) is the focus of the overrule by the human instructor.

Fig. 5. Instructing scene

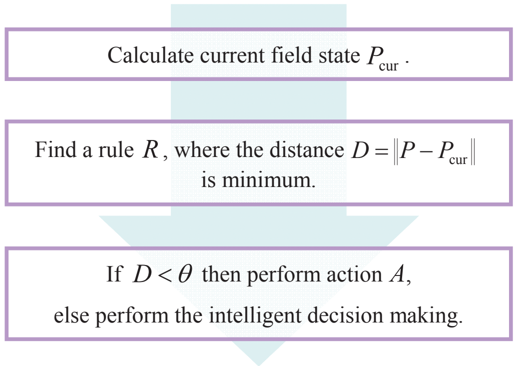

Fig. 6. Decision making process

Figure 5 shows a snapshot of the interface for the human instructor. This interface is used to send instructions that overrule the currently executed actions of the main attacker. The instructions are then converted to action rules as described Section II. The generated action rules are added to our team, opuCI_3D_2010, which participated in RoboCup 2010 Singapore. The agent first looks at the rules that are manually written. Then the fittest rule with the current field status is chosen to select an action. The rule R which has the nearest P to the current field status is selected and the distance D between P and the current field status is calculated as follows:

$$\begin{aligned} D &= \|P - P_{\text{cur}}\| && (2)\\ &= [\sum_{i \in Obj} \{(x_i - x_i^{\text{cur}})^2 + (y_i - y_i^{\text{cur}})^2\}]^{\frac{1}{2}} && (3) \end{aligned}$$

where P_{cur} is the current state vector and *Obj* includes self, ball, opp1, opp2, opp3, and mate. The distance D is calculated in Euclidean distance. If D is smaller than a certain threshold value θ, the behavior A is executed. Otherwise, an agent makes a decision according to manually written action rules. Figure 6 shows the decision making process of an agent using action rules.

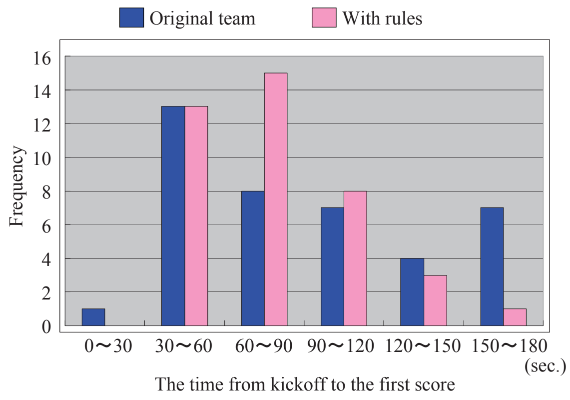

Fig. 7. Performance comparison between the original team and the experimental team

Table 1. Mean and variance of the time necessary for scoring

	Original team	With rules
Mean	90.20	75.04
Variance	2333.1	946.92

In the performance evaluation, the team with the action rules generated by the proposed method played against team opuCI_3D_2010. A game starts by the experimental team's kickoff and the time from kickoff to the first score by the experimental team is measured. 40 games are played for the performance evaluation.

2. Results

The results of the performance evaluation are shown in Fig. 7. In Fig. 7, the vertical axis means the frequency and the horizontal axis means the time necessary to get a score. Table 1 shows the mean and the variance of the time necessary for scoring a goal. From Fig. 7 and Table 1, we can find that the generated rules, that is generated behaviors, lead to the decrease in the time from kickoff to a goal. In order to show the statistical significance of our method, we performed a one sided t-test in order to show that there is a significant difference between these two means. The null hypothesis H_o and the alternative hypothesis H_a are the following:

$$H_{\rm o} : u_{\rm t} = u_{\rm r} \tag{4}$$

$$H_{\rm a} : u_{\rm t} > u_{\rm r} \tag{5}$$

where $u_{\rm t}$ is the mean of the original team's first score time and $u_{\rm r}$ is the mean of the rule experimental team's first score time. The result of the t-test is shown in Table 2. We can find the fact that the p-value $P(T \leq t)$ associated with the t-test is smaller than α from Table 2 and there is evidence to reject the null hypothesis $H_{\rm o}$ in favor of the alternative hypothesis $H_{\rm a}$. Therefore we can say that the proposed method effectively decreased the time necessary to score a goal.

Table 2. The result of t-test

α	0.05
t-value	1.674
$P(T \leq t)$	0.0494

IV. CONCLUSIONS

In this paper, we introduced the behavior generation approach from human instruction. The proposed method enables us to improve the behavior of autonomous soccer agents through human instructions. The results of computational experiments showed that the agents with human instructions are superior to the original ones in terms of time from kickoff to the first score. Now top level teams which participate in the world competition have developed highly sophisticated skills, and it becomes more important to improve the team strategy. Our method can be expanded to apply to defender's behavior to improve the defense ability.

REFERENCES

[1] Gabel T, Riedmiller M, Trost F (2008), A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach. RoboCup 2008: Robot Soccer World Cup XII:61–72

[2] Kyrylov V, Hou E (2009), Pareto-Optimal Collaborative Defensive Player Positioning in Simulated Soccer. RoboCup 2009: Robt Soccer World Cup XIII:179–191

[3] Kalyanakrishnan S, Stone P (2009), Learning Complementary Multiagent Behaviors: A Case Study. RoboCup 2009: Robt Soccer World Cup XIII:153–165

[4] Spark - A generic physical simulator, Sourceforge project, http://sourceforge.net/projects/simspark/files

[5] Kogler M and Obst O (2003), Simulation League: The Next Generation, RoboCup 2003: Robot Soccer World Cup VII:458–469

[6] Shafii N, Nezami O M, Aslani S et al. (2009), Evolution of Biped Walking using Truncated Fourier Series and Particle Swarm Optimization. RoboCup 2009: Robot Soccer World Cup XIII:344–354

[7] Warden T, Lattner A, Visser U (2008), Real-Time Spatio-Temporal Analysis of Dynamic Scenes in 3D Soccer Simulation. RoboCup 2008: Robot Soccer World Cup XII:366–378