Caspian 2004 Soccer 3D Team Description

Mohammad Nejad Sedaghati, Bardia Aghabeigi, Mehrdad Senobari,
Mohammad Jafar Abdi, Nina Gholami, Sina Iravanian, Arash Alikhani,Mohammad
Saeed Tabatabaee, Ehsan Rafiee, Omid Mehdi Zadeh, Sommayeh Zahedian,
Leila Pakravan Nejad, Hamid Hamraaz, Shams Feyzabaadi, Reza Seyed Khamooshi,
Mohammad Reza Kangavari

mnsedaghat@yahoo.com

Intelligent Systems Lab, Computer Engineering Department
Iran University of Science and Technology, Tehran, Iran

http://caspian.iust.ac.ir

Abstract. Soccer Simulation League a transition to 3D states brings out new research problems and equally makes it more appealing to the people who are involved in soccer simulation researches. To develop an intelligent agent in 3D Soccer Simulation environment, two fundamental challenges should be addressed: World Model enhancement, and Skill development. In this paper we propose a method to enhance the World Model including agent localization and velocity estimation. In addition, the skills which are available to the agents of Caspian Soccer Simulation Team are presented.

1. Introduction

For eight years now Soccer Server [1] exists as a test bed for evaluating multi-agent systems and has inspired a lot of researchers from Computer Science and Artificial Intelligence to compare their approaches [2]. Soccer Simulation League a transition to 3D states brings out new research problems and equally makes it more appealing to the people who are involved in soccer simulation researches.

3D Soccer Server is a simulation system in which the agents are represented by individual entities which receive sensor information through the considered sensors and perform an action using the defined actuators in a certain amount of time. To develop an intelligent agent two fundamental challenges should be addressed:

1. World Model enhancement
2. Skill development

In this paper first we describe our approach to enhance the World Model including agent localization and velocity estimation. In the next section, the skills which are available to the agents of Caspian Soccer Simulation Team are presented. Finally the potential improvements to the current approach in Conclusion and Future Work section are given.

2. World Modeling

World Model is the agent's understanding from his environment. The agent's World Model contains information about his own status and also the status of the other agents in the surrounding environment. Currently the agent status in Soccer 3D mainly includes the agent position and velocity. Since the received information is relative to the agent, a transformation from relative coordinates to global coordinates should be performed. In this section we describe our approach to estimate global position and velocity based on the data received from the vision sensor.

2.1. Agent Localization

Agent Localization is defined as estimating the agent's global position based on the data received from the agent's sensors. Currently the only agent sensor implemented in 3D Soccer Server is the vision sensor. It provides the agent with its position and velocity relative to predefined flags on the field. The important point is that the received information is not noise-free.

In order to estimate the agent's position, first we extract the two closest flags relative to the agent. The distance from the agent to the F flag is called Fr. Respectively the distance from the agent to the G flag is called Gr. The flag information in the vision sense is in the following form $F(Fr, \theta_1, \Phi_1)$, $G(Gr, \theta_2, \Phi_2)$. In addition, the agent also knows the global positions (Fx,Fy,Fz) and (Gx,Gy,Gz) of F and G since these are stored in his world model. Combining the distance to a flag with the known global position of this flag gives a circle of possible agent positions. This circle has the flag as its center and the perceived distance as its radius. In the same way the second flag also defines a circle of possible positions. Clearly, the correct agent position must lie on both circles. In general, the circles will have two intersection points one of which denotes the actual position of the agent. It can be determined simply because the other point is out of range. An example situation is shown in Figure 1. Following this approach, the agent localization problem thus amounts to finding the correct intersection point *p* of the two circles. In order to determine this point, we first calculate the orthogonal projection P1 of *p* onto the line segment that connects the two flags. To this end we compute the distance *d* between *F* and G by making use of their known global positions [3].

$$d = \sqrt{(Fx - Gx)^2 + (Fy - Gy)^2}$$

This distance can be seen to consist of two parts: one from *F* to *P1* and one from P1 to *G.* Let *a* denote the distance from *F* to *P1, b* the distance from *P1* to G and *h* the distance from *p* to P1. When we consider the triangles FPP1 and GPP1 it follows from the Pythagorean Theorem that:

$$a = \frac{Fr^2 Cos^2 \Phi_1 - Gr^2 Cos^2 \Phi_2 + d^2}{2d}$$

$$h = \sqrt{F^2 Cos^2 \Phi_1 - a^2}$$

The global position of agent becomes:

$$Px = Fx \pm h$$
$$Py = Fy \pm h$$
$$Pz = Fr \cdot \left| Sin\Phi_1 \right|$$

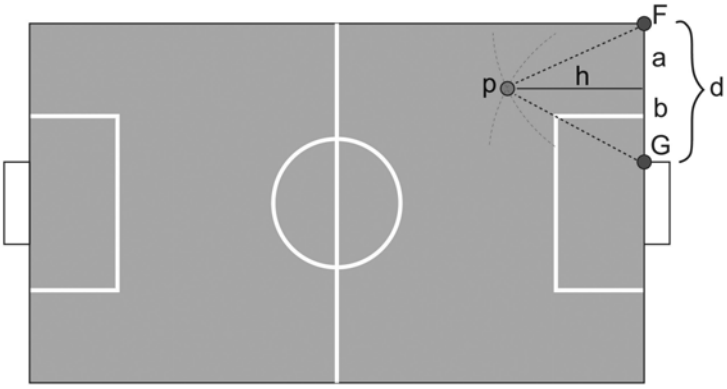

Figure 1. Agent localization using two flags

2.2. Velocity Estimation

Our approach to estimate the agent's velocity is based on the agent's position in a sequence of cycles. The approach uses the difference between the global position $(q_x^{t-1}, q_y^{t-1}, q_z^{t-1})$ of the dynamic object in previous cycle and its current global position (q_x^t, q_y^t, q_z^t) to estimate the global dynamic velocity (v_x^t, v_y^t, v_z^t) in the current cycle.

When the dynamic object has zero acceleration the movement (u_x^t, u_y^t, u_z^t) from cycle $t-1$ to Cycle t is equal to the velocity $(v_x^{t-1}, v_y^{t-1}, v_z^{t-1})$ in cycle $t-1$. From this it can be concluded that $(v_x^{t-1}, v_y^{t-1}, v_z^{t-1})$ equals the difference between (q_x^t, q_y^t, q_z^t) and $(q_x^{t-1}, q_y^{t-1}, q_z^{t-1})$. In order to compute the velocity in cycle t, we have to multiply the velocity in cycle $t-1$ by the velocity decay rate of the dynamic object. This gives the following formula for calculating the velocity (v_x^t, v_y^t, v_z^t) based on consecutive positions:

$$(v_x^t, v_y^t, v_z^t) = ((q_x^t, q_y^t, q_z^t) - (q_x^{t-1}, q_y^{t-1}, q_z^{t-1})).dynamic\_object\_decay$$

3. Skills

A skill can be regarded as the ability to execute a certain action. The behavior of an agent depends on the individual skills that this agent can perform. In general these skills can be divided into simple skills that correspond to basic actions and more advanced skills that use the simple skills as parts of more complex behaviors [3]. In this section we will describe the skills which are available to the agents of Caspian Soccer Simulation Team.

3.1. Moving to a Position

This skill enables an agent to move to a global position in the field. In our approach, the distance between the agent and the target is divided into several intervals, in each of which the agent runs at a constant speed with respect to its distance to the target.

3.2. Kicking the Ball toward a Point

This skill is applicable only if the distance between the agent and the ball is smaller than a specific value (5 meters in our implementation). The skill enables the agent to get near enough to the ball, so that it can kick the ball. Since the agent cannot specify x-y plane angle for the kick command, selecting a correct position near the ball is very important. To achieve this goal, the agent first tries to move into the ball's kickable area, and then moves round the ball until it reaches the most suitable position, then kicks the ball using an appropriate power.

3.3. Running at a Constant Speed

This skill enables an agent to run at a constant speed. Using this skill the agent can control its velocity. Moving in a constant speed leads to more accurate estimation of velocity.

3.4. Intercepting the Ball

This skill enables an agent to intercept the ball. If a player wants to kick the ball, he should first intercept it. The main objective is determining the optimal interception point based on the ball's current position and velocity, then moving to that point as fast as possible in order to reach the ball before an opponent does.

This skill predicts the position of ball after i cycles, and determines whether the agent is able to reach the predicted position of the ball within i cycles.

If the player can reach the predicted position of the ball within i cycles, the move-to-position skill is called, and otherwise the intercept action is cancelled.

3.5. Passing the Ball to a Teammate

This skill enables an agent to pass the ball to one of his teammates. There are two kinds of passes: *Fast* and *Normal*. This classification is done based on the speed of the ball at the destination point. The player that the ball is passed to must be the safest player to receive the pass. To speed up the attack, the passes should be in the form of a leading pass. That is the ball is kicked to an open space in front of the target player. In this case, the pass receiver must run to the ball and intercept it.

4. Conclusion and Future Work

3D Soccer Server is a simulation system in which the agents are represented by individual entities which receive sensor information through the considered sensors and perform an action using the defined actuators in a certain amount of time. Currently a great effort goes into building a reliable World Model and effective agent skills. In fact, accuracy of World Model has a great impact on the performance of the skills. In current approach, the built-in World Model methods generally perform basic analysis related to agent's position and velocity. In order to implement more efficient and well-defined skills, we need to develop high-level prediction methods. The prediction methods can be used to predict future state of the world based on past perceptions. These methods are important for the action-selection process. Two main prediction methods should be developed. One is for predicting the state of the agent after performing a certain action like kick and drive, and the other is for predicting how long it takes for an agent to reach a certain position on the field. The former method receives an action command as input and predicts the agent's state after the given cycles. The latter receives the target position and a player, as input and estimates the number of cycles it takes the agent to reach the given position. This method is very useful for the goalkeeper to find the best position for catching the ball. In addition to the prediction methods, we need to integrate some high-level methods into the World Model. The following is some of these high-level methods:

- Methods that give information about the number of players in a certain area
- Methods that return the closest player to a certain position or object
- Methods that indicate whether specific objects satisfies a certain high-level constraint
- Methods for computing the actual arguments that should be supplied to an action command in order to achieve the desired result such as: the power for kick or drive

The behavior of an agent depends on the individual skills that this agent can perform. In fact effective skills lead to efficient agent behaviors. In order to enhance the agent's behavior we need to develop more high-level skills that use the simple skills as parts of more complex behavior. Some of these high-level skills are listed below:

- **Marking an opponent:** This skill enables an agent to mark an opponent, i.e. to guard him one-on-one with the purpose to minimize his usefulness for the opponent team.

- Clearing **the ball:** This skill enables an agent to clear the ball to a certain area on the field. It is useful, for example, when a defender cannot dribble or pass the ball to a teammate in a dangerous position. Using this skill it can then kick the ball up to the field away from the defensive zone.

- **Passing the ball into the depth (through pass):** This skill enables an agent to give a more advanced type of pass called through pass. With a through pass the ball is not passed directly to another player, or just ahead of him, but it is kicked into open space between the opponent defender, and the opponent goalkeeper in such a way that a teammate will be able to reach the ball before an opponent does.

- **Dribbling:** This skill enables an agent to dribble with the ball, i.e. to move the ball while keeping it within a certain distance. This requires repeatedly kicking the ball at a certain speed into a desired direction and then intercepting it again.

References

1. Noda, I.: Soccer Server: A Simulator of RoboCup. In proceeding of AI Symposium ’95, pages 29-34. Japanese Society for Artificial Intelligence, December 95.
2. Kögler, M., Obst, O.: Simulation League: The Next Generation. In proceeding of RoboCup 2003 Symposium. 2003.
3. de Boer, R., Kok, J.: The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team. February 28, 2002.