

UI-AI3D 2005 Team Description

Mohammad Babaeizadeh¹, Mohammad Amin Jashki¹, Hamid Mohammadi¹,
Esmail MollaAhmadi¹, Mohsen Mossavi², Zahed Rahmati¹, Hamed Shahbazi¹,
Abbas Yazdanpanah¹, and Reza Zafarani¹
<http://eng.ui.ac.ir/uisim>

¹ Computer Engineering Department , University of Isfahan ,
81745 Isfahan, Iran
babaei_mbz@yahoo.com, hardchice@yahoo.com, hmohammadi110@gmail.com,
empoly_2004@yahoo.com, zahed114@yahoo.com, demahpilot@yahoo.com,
yazdanpanah_a998@yahoo.com, rzafarani@eng.ui.ac.ir

² Faculty of Engineering, University of Isfahan,
81745 Isfahan, Iran
mohsen@mossavi.com

Abstract. UI-AI was one of the first teams that participated in the Robocup 3D competition's first year in Portugal, the 3D soccer simulation environment[1] is introducing a new test bed for creating and simulating 3D soccer teams. Team agents are based on spades' communication[2] and synchronization and the new 3D simulator born. The new UI-AI team is completely created with a structural and multi layer architecture.

1 Introduction

The main reason for creating the new team was to reorganize the old works done on the UI-AI3D 2004 and creating a base source code for 3D soccer simulation, therefore we created a new source code, and the communications is based on the old simple agent in the 3D server. We also added some useful techniques from the old 2D server to our code to simplify some works here.

2 Team Architecture

The team consists of three major layers, communication, world model and action layer. The communication layer does the low level data sending as well as converting the high level actions in higher layers to lower level actions, which should be sent. The layers does some blocking functions and some signaling functions to reduce the size of I/O, although a multi-threaded system is not used here , the signaling we used improved out the performance drastically. In the next layer (World Model), the first act is to do some localization algorithms to find out the exact place of the agent. This is done by analyzing the different errors created by the three factors (Phi, Theta, D) received in the visionSense from the server. We therefore analyzed the errors and selected the best localization

algorithm. After localizing, the players do some world model calculations such as speed and acceleration vectors, place of others players and their relative speed and acceleration. All these data plus some additional datas, needed for further considerations, are kept in the world model layer. In the action layer the players traverse a decision tree and a fuzzy function is used to select the best action for the current agent. Actions selected currently depend on small amount of factors such as distance to ball, speeds, the TYPE of player in formation, and game situations. Formation is also implemented in this layer.

3 Localization

Having Knowledge of the exact position of objects, is the most important parameter in the World Model layer. Obviously, the algorithm which is used for localization should have high accuracy and speed (The order function should converge in less than the order of $O(n^4)$ in our tests). Finding the place of objects, different kinds of errors are analyzed and various methods like *Maximum Likelihood* , *RBFNN* and Neurofuzzy are tested. Although machine learning algorithms are tested, the need of some handcoded algorithms were essential, therefore UI-AI agents are using some hybrid (handcoded and machine learning). The UI-AI3D 2005 team is now using an algorithm which an agent can find his accurate location, rapidly. Having the exact position of himself, the agent will be able to do other computations in the game more subtle.

4 Timing, Velocity and Acceleration

Next important parameter in World Model is timing and velocity vectors. By changing engine force of agents, acceleration changes dynamically. So agents should calculate their acceleration vectors, too. Discrete time and timing problems, made us not to use real time which presented by SPADES. UI-AI3D 2005 uses a virtual timing system for timing and these kind of problems, which makes velocity calculation more accurate, using physics terms. It also enables agents to predicate objects' velocities in up coming times, accurately. These parameters will be used in the action layer has the implementation of move, kick and other soccer actions.

5 Improving performance using Signaling

Although the time blocked by receiving the data input is not reduced (Multi-threading is not used), the resources such as world model, the output from the fuzzy selector, and the layers are having owners at a time. In each virtual time generated by the agent each resource can have one owner. The number of times each owner can keep the resource is fixed dynamically. The algorithm is currently based on common operating system scheduling algorithms, but the required time for each owner is currently based on the knowledge we got during

our tests. Although, the algorithm is in its primary stages of development, it has improved the flexibility of our team.

6 Formation and Decision Tree

The last stop in our modular code for the agents is the Action layer which formation and decision tree are implemented in. This layer is not completely carried out yet. Formation is very simple right now, which currently contains AGENT.TYPE, before kickoff standing positions, attractions and action areas and other simple parameters. So agents follow a coach-like decision tree which is pruned using soft fuzzy functions. Formation file is used as low level input for high level actions generated by fuzzy function.

7 Conclusion and Future Work

The main aspect for the UIAI3D-2005 was to create a base source code for the 3D soccer simulation. This base stands on a objective three layer Model. A powerful Communication and accurate World Model and simplified version of important actions such as moving to a position and kicking the ball was implemented. Our future work is based on continue working on base code, improve low levels such as creating a threaded binary, more accurate movement model and better high levels and decision modelling. Using our base, 2D Soccer experiences and 3D Soccer capabilities will lead us through a more better 3D Soccer team.

References

1. Kogler, M.: Simulation and Visualization of Agents in 3D Environments.
2. Riley, P.: SPADES System for Parallel Agent Discrete Event Simulation, User's Guide and Reference Manual For Version 0.91