Visualizing and Debugging Complex Multi-Agent Soccer Scenes in Real Time

Justin Stoecker, Ubbo Visser

Department of Computer Science
University of Miami, Coral Gables FL,
{justin,visser}@cs.miami.edu

Abstract. The RoboCup Soccer environment is one of the most difficult scenarios for autonomous agents. With the potential for so many things to go wrong, debugging and analyzing agents' behaviors becomes a significant task. We propose RoboViz, an open-source program for integrating agent-driven visualizations into a real-time, 3D rendered environment; the scene becomes a shared, interactive whiteboard for all agents, and the user can moderate by filtering drawings they are interested in. Visualization is an effective tool for tracking down errant behaviors and explaining algorithms. RoboViz is embraced by the RoboCup Soccer Simulation 3D sub-league as the de facto monitor application, and the latest revision makes it useful for other leagues as well. We are currently testing RoboViz in the Standard Platform League (SPL).

1 Introduction

A significant challenge in developing a robotic agent is debugging and evaluating its behavior. The RoboCup Soccer scenario is one of the most difficult environments for intelligent agents, and presents several hurdles: an uncertain and dynamic world, multiple competitive and cooperative agents, physics, and the need for high-level strategy. With such a challenging environment, it is inevitable that teams developing soccer agents for RoboCup will stumble over bugs and struggle with solutions. As might be expected, teams competing in the various RoboCup leagues develop their own specialized tools, in isolation, to optimize and debug agent code. However, there is generally a lack of tools that are truly beneficial to all teams participating in a league. In this paper, we focus on the development and debugging issues shared by teams in the RoboCup Soccer Simulation 3D sub-league; however, the same issues are relevant to other RoboCup Soccer sub-leagues.

There are several challenges to overcome when interpreting robot behaviors, such as localization or task planning, and simple approaches are often inadequate in diagnosing problems. The real-time nature of the environment is the most notable complication: agents generally process input and act within milliseconds, so outputting values on a console provides an incomprehensible amount of information. Logging this information and parsing it later has two serious drawbacks: the data can fill volumes very quickly, and it can be difficult to synchronize one