UvA Trilearn 2002 Team Description

Jelle Kok, Remco de Boer, Nikos Vlassis, and Frans Groen

Faculty of Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{jellekok,remdboer,vlassis,groen}@science.uva.nl

Abstract. This paper describes the main features of the *UvA Trilearn 2002* soccer simulation team. This team is an extension over *UvA Trilearn 2001* which participated for the first time at the RoboCup-2001 competition. The main concepts of *UvA Trilearn 2001* will be addressed briefly, followed by the improvements introduced in *UvA Trilearn 2002*. These include the improved localization methods using particle filters and the action selection method based on a priority-confidence model. Finally we will give some conclusions and describe future research directions.

1 Introduction

The *UvA Trilearn 2001* [1, 2] soccer simulation team was built by two masters students from the Intelligent Autonomous Systems Group at the University of Amsterdam for their graduation project. It had been built from scratch and did not contain any code copied from other RoboCup teams. Much of the initial effort had gone into getting the lower levels to work, since we felt that these would be the most crucial for the success of the team. Furthermore, low-level imperfections cannot be compensated for by high-level behavior added later on. This has among other things led to a multi-threaded three-layer architecture with an advanced synchronization method and a probabilistic world model from which several high-level conclusions could be derived. At the higher levels we encoded several skills that each player could execute and a fast-play strategy in which the players tried to pass the ball quickly to each other. In addition, they often tried to pass the ball into the depth in front of the wing attackers at the side of the field thereby cutting through the opponents' defense and disorganizing their team. The effectiveness of this strategy was greatly enhanced by the use of heterogeneous players on the wings. The best heterogeneous player for this position was selected using a utility function, which combined the maximum speed and the stamina loss when moving at this maximum speed. Although these players became tired more quickly, it was their faster speed which made the difference. Furthermore, an optimal scoring policy was implemented which returned the point in the goal with the highest probability of scoring[1]. During the project we also paid much attention to software engineering issues to facilitate

[1] For details see [4], which was accepted at the 7th International Conference on Intelligent Autonomous Systems (IAS-7).

future use. This led to highly modular object oriented code and to a multi-level log system for quick debugging (similar to [8]). Much effort had also gone into documenting our code for future release using the documentation system Doxygen [10]. *UvA Trilearn 2001* reached 5th place at the German Open 2001 and 4th place at RoboCup-2001. The main extensions resulting in *UvA Trilearn 2002* will be discussed in the remainder of this paper.

2 Localization using particle filters

We have implemented the concept of particle filtering to improve the position and velocity estimation of the agent and the ball. The particle filter is an attractive simulation-based approach to the problem of computing intractable posterior distributions in Bayesian filtering [3]. The idea is to determine a discrete approximation of the continuous posterior density by using a set of N particles x_t^i with associated probability masses π_t^i for $i = 1, ..., N$. For the agent localization each particle is represented by a quadruple (x, y, v_x, v_y) which contains the global position and velocity in Cartesian coordinates. The particle set thus resembles the current state of the agent and is updated each time new information is received. There are four different phases in which the particles are updated:

Initialization phase At the arrival of the first visual information or when no particles are left after the verification phase, all particles are initialized using the perceived information of the closest flag. This information contains the distance and angle relative to the neck of the agent. Noise is incorporated into these values by quantizing the real value r. This quantization procedure effectively means that a range of real values $r \in [r_{min}, r_{max}]$ is mapped to the same quantized value r'. Given the quantized value r' we can determine the range of values from which the real value must originate[2]. The ranges for the relative angle and relative distance are used to initialize each particle. A value is selected randomly from each range and using the known global position of the perceived flag and the calculated global neck angle of the agent (which also introduces an additional noise range) a possible global position of the agent is calculated. The velocity of the agent is initialized by setting it to the received velocity from the body sensor information (which hardly contains any noise). After all particles are initialized they are verified with the information of the other flags also contained in the visual information.

Shifting phase At the start of the next cycle, the position and velocity information contained in each particle is updated according to the performed action by the agent and the known soccer server dynamics. For this we use the exact same equations (including the noise) that the server uses.

Verification phase After the arrival of new visual information, the information for each flag is used to verify which particles are impossible. All particles are

[2] This method of *inverse quantization* is also used by Lucky Lübeck [6]. For each flag they use the returned ranges to approximate the area of possible player positions by a polygon. The global position of the agent is calculated by intersecting all polygons.

removed from the set that fall outside the range from which the perceived relative angle or relative distance could originate. After this phase, only those particles remain that are consistent with all flag information.

Resampling phase After removing the particles in the verification phase, the particle set is resampled by randomly selecting one of the remaining particles and making an exact copy of it. This is repeated until the original number of particles has been restored. Note that identical particles that result from the resampling step will be spread out again in the next cycle due to the fact that the noise added during the shifting phase has a random character.

Since the range from which the values are initialized and updated is uniformly distributed, the probability mass π_t^i for each particle equals $1/N$. The position and velocity estimate can thus easily be derived by taking the average of all particles currently contained in the set. Tests are performed where both the global position and global velocity of an agent are recorded in each cycle and compared to the exact values recorded by the coach. The average error over 10 matches (60,000 samples) equals 4.51cm for the global position (with a standard deviation of 0.67cm) on a field of 105m$\times$68m and 0.276cm/cycle (with a standard deviation of $1.04 \cdot 10^{-5}$) for the global velocity. Note that these values are gathered with the view cone of the agent set to normal, meaning that no visual information was received once every three cycles. This error is considerably smaller than reported by other teams [6, 9]. For the ball position and velocity the same approach is followed. The only difference is that to calculate the range of possible values more factors have to be taken into account.

3 Priority-Confidence Model

Since much of the initial effort for *UvA Trilearn 2001* had gone into getting the lower levels to work, not much time had been spent on the action selection procedure. Depending on the position on the field a priority list of actions was traversed and the first action with a high enough success rate was selected. Although this procedure proved to be very effective, it lacked flexibility. In *UvA Trilearn 2002* we have implemented the concept of a priority-confidence model [5] to determine a new action. In this framework the different possible actions (passing, dribbling, etc.) are compared using a confidence measure which is based on the importance of the action (priority) in combination with the satisfaction of the a priori conditions (confidence). The action with the highest confidence measure is executed. The main difference with our previous approach is the fact that now all actions are taken into consideration.

4 Conclusion and Future Directions

We have finalized the implementation of our lower levels after the implementation of the particle filters. This has resulted in a source code release of our low-level software (including the synchronization, world model and basis agent skills)[3]. Furthermore it contains a simple high-level strategy, similar to a simple team released by *FC Portugal* [7]. With the same high level decision procedure our team defeats *Simple Portugal* by an average score of 5-0 indicating that our lower levels outperform theirs. In combination with the extensive documentation throughout the code, we believe this is a good starting point for new teams. From this point forward, we will focus more on the higher levels and in particular on multi-agent modeling using the coach. In our current implementation, individual agent decisions are often affected by actions from opponents, but strategy decisions for the team as a whole are mostly fixed. Ideally however the team strategy should be adjusted in response to adversary behavior. We want to use the coach to analyze the strengths and weaknesses of the enemy team and to give advice on the best possible strategy. To this end, the coach must be able to classify enemy behavior based on certain features. He can then decide which strategy is most appropriate for the given behavior class. The coach can for instance recommend to change the formation or switch player types for a certain position inside the current formation. In turn the players must know how to interpret the coach's advice and how to act upon it.

References

1. R. de Boer and J. Kok. The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team. Master's thesis, University of Amsterdam, The Netherlands, Feb. 2002.
2. R. de Boer, J. Kok, and F. Groen. UvA Trilearn 2001 Team Description. In *Robocup-2001: Robot Soccer World Cup V*. Springer Verlag, Berlin, 2002.
3. A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.
4. J. Kok, R. de Boer, and N. Vlassis. Towards an Optimal Scoring Policy for Simulated Soccer Agents. In *Proc. 7th Int. Conf. on Intelligent Autonomous Systems*, Marina del Rey, California, Mar. 2001. To appear.
5. J. Lubbers and R. R. Spaans. The Priority/Confidence Model as a Framework for Soccer Agents. In *Proceedings of the Second RoboCup Workshop*, pages 99–108, Paris, 1998.
6. D. Polani and T. Martinetz. Team Description for Lucky Lübeck - Evidence-Based World State Estimation. In P. Stone, T. Balch, and G. Kraetszchmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*. Springer Verlag, Berlin, 2001.
7. L. P. Reis and J. N. Lau. FC Portugal Team Description: RoboCup-2000 Simulation League Champion. In P. Stone, T. Balch, and G. Kraetszchmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 29–40. Springer Verlag, Berlin, 2001.
8. P. Riley, P. Stone, and M. Veloso. Layered Disclosure: Revealing Agents' Internals. In *Proceedings of the Seventh International Workshop on Agent Theories, Architectures and Languages (ATAL-2000)*, 2000.
9. T. Suzuki. Team YowAI Description. In *RoboCup-99 Team Descriptions for the Simulation League*, pages 31–35. Linköping University Press, 1999.
10. D. van Heesch. The Documentation System Doxygen, 1997. At http://www.doxygen.org.

[3] Available from our website: http://www.science.uva.nl/~jellekok/robocup/