Real world multiagent systems: information sharing, coordination and planning

Frans C.A. Groen, Matthijs T.J. Spaan, and Jelle R. Kok

Informatics Institute, University of Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Abstract. Applying multiagent systems in real world scenarios requires some essential research questions to be answered. International challenges play an important role in comparing different approaches to these questions. One such a challenge in this field is RoboCup. In this paper we focus on three of these research questions: A shared world model which estimates the global positions of objects in order to reduce the uncertainty in the environment. It allows us to fuse observations made by different agents and improve position estimates of each agent. Second, we show how coordination graphs can be applied to multi-robot teams to allow for efficient coordination. Third, we present work in the POMDP framework for agent planning in uncertain environments, in which the agent only receives partial information (through its sensors) regarding the true state of environment.

1 Introduction

In the future intelligent multiagent systems will be deployed in real world situations, for instance, as service robots, transportation systems, exploration of hazardous environments, homeland security and rescue in disaster scenarios [20]. The societal and economical benefits of building such systems are huge, while at the same time there are still important research questions to be answered before these systems can be applied. This requires the integration of many technologies such as mechatronics, control theory, computer vision, self-learning systems and cooperative autonomous systems [11]. These agents are "intelligent on-line embedded systems" which are able to operate in human habited dynamic environments. Local intelligence and mutual communication make systems robust to erroneous perception or malfunctioning of one or more robots.

How to evaluate these complex systems is not an easy question. The current trend to enable comparison of different algorithms is to make the data available on Internet, apart from only publishing the results obtained from them in articles. However, the evaluation of real world multiagent systems is much more complex. Simulation is certainly useful in this respect, but real comparisons require the deployment of systems in real world scenarios. It has been discerned that international challenges may play an important role in those evaluations. A challenge should be sufficiently rich so that the different aspects of the problem are well represented. Challenges should not change every year but should have a stable component so that ideas or even best algorithms can be adopted by other competitors, ensuring that a rapid development takes place over the years and incorporating all groups involved. An example is the DARPA Grand Challenge: a race for autonomous ground vehicles through desert-like terrain. A challenge formulated in multiagent collaboration is the RoboCup challenge [5, 10]: to have in 2050 a team of humanoid robots playing a soccer match against a human team. Robot soccer is quite representative for the problems occurring when multiagent systems are applied in real world scenarios. There are a number of possibly heterogeneous robots that have to work together toward a common goal. The domain is continuous and dynamic, there are opponents whose behavior will not be fully predictable. Another challenge is Robot Rescue: the search and rescue for large scale disasters, e.g., searching for survivors. It started as a simulation project but now also involves a real environment developed by NIST.

In section 2 we will discuss the RoboCup challenge and the topics addressed in this challenge. Three of those topics are addressed in the successive sections in more detail. In section 3 we will discuss a distributed world model, which forms the basis for planning and learning to coordinate the multi-robot team. We consider how the agents can build and maintain shared models of the environment. We have been studying the problem of robot localization, that is, how a robot can find its position in the environment under conditions of uncertainty in its motion and sensor measurements. We are using particle filters for this problem. Section 4 explores the framework of coordination graphs for solving multiagent coordination problems in continuous environments such as RoboCup, as well as how learning can be performed in such settings. Section 5 addresses a second problem, planning under uncertainty, and here we are investigating solution techniques for partially observable Markov decision processes. Finally, section 6 wraps up with conclusions and avenues for future developments.

2 RoboCup, a challenge for real world multiagent systems

In 1997 the RoboCup Federation organized the first RoboCup competition in Nagoya followed by yearly world championships [5]. RoboCup's main challenge is to develop a team of humanoid robots playing soccer that are able to defeat the human world champion in 2050. Competitions in multiple leagues offer the possibility to focus research on different aspects of this challenge in the different leagues. It involves different multidisciplinary topics which are reflected and addressed in separate leagues: the simulation league, the small size league, the middle size league, the legged league and more recently the humanoid league.

Small-size robot league The small-size league is played on an enlarged table tennis-sized field. Each team consists of five small robots of about 15 centimeters in diameter. A camera above the field is used to get a complete view of the game, which is sent to the computers of the teams on the side of the field. From this image a world model is constructed using the color

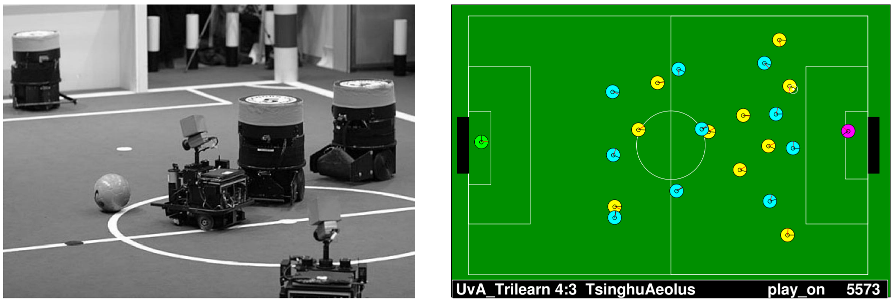

Fig. 1. Two RoboCup leagues: on the left the middle-size robots, on the right the simulated soccer agents.

coding of the ball and the different robots. Based on this world model the actions of the different robots are determined and sent to the robots by way of wireless communication. Since the world model is complete and quite accurate research focuses on robot coordination, team behavior and real time control.

Middle-size robot league The robots in the Middle-size league are about 50 centimeters (1.5 feet) in diameter, see Fig. 1 (left). They compete on a field of 10 meters long and 5 meters wide. The objects are colour coded: the ball is orange, the goals are yellow and dark blue, the robots are black with a light blue or pink hat, the field is green and the lines are white. The main difference with the small-size league is that there is no global vision of the field. Visual information is received from a camera on board of each robot. The robots can communicate with each other by way of wireless communication. To create cooperative team behavior robots have to know where they are on the field and self-localization is a key issue. Currently omni-directional vision systems that give a 360 degree view of the field are used by most teams to facilitate self-localization.

Sony Legged robot league On a field slightly larger than the small-size league, teams of four Sony AIBO's (the well-known robotic toy dogs) compete. These robots walk on four legs, and are thus the first 'step' toward a league of biped humanoid robots. Since every team uses the same robots, the only difference between the teams is in the software.

Simulation league This looks like a standard computer game (see Fig. 1 (right)), but the essential difference is that each player is its own simulated robot, driven by its own program. Each agent has to decide on its own next move. Because simulation frees the researchers from inherent physical limitations of 3-dimensional robots as mechanical parts, wheel control and other functions, these screen players are able to perform on a far more advanced level. They are able to interact and cooperate, even changing play strategies from defense to offense, or from forward to defender. A team consists of 11 players, each of which can be either homogeneous or heterogeneous and has various properties such as dexterity and pace. Since there is only a limited amount of uncertainty in the information agents receive, it is relatively easy to generate a complete and accurate world model, compared to the other leagues. This enables the teams to concentrate on cooperative team behavior and tactics.

Key topics represented with the RoboCup challenge are

- Robot development for the different leagues, with as final goal a humanoid robot capable of playing soccer.
- Perception of the real world from a moving platform in a dynamic scenario. This involves both self-localization and object recognition.
- Communication and fusion of information obtained at different moments by different robots to create a dynamic model of the world.
- Reactive behavior, to create the basic actions of the robot.
- Planning to decide what actions are optimal in a given state.
- Team coordination which involves learning and opponent modeling.

In the rest of this paper we will address several of these topics.

3 Distributed shared world models

A Shared World Model (SWM) estimates the global positions of objects in order to reduce the uncertainty in the environment [17]. It allows for the fusing of observations made by different agents and estimates new attributes such as the speed of objects for extrapolation. This requires each robot to accurately localize itself, i.e., to find its most likely position given a sequence of robot displacements and observations. To create a distributed system robust in case of failure of an agent, each agent computes the SWM by itself. There is no central control which makes the system robust. The disadvantage of this is that the SWM on each agent can differ due to different order and time of arrival of the local world models. The higher levels in planning and coordination should be robust against these differences.

In the RoboCup middle-size league the observations mainly consist of camera images which are compared to a model of the environment. We have taken an iterative approach in which after each displacement and observation the probability distribution representing the uncertainty over the current position of the robot is updated. This probability distribution is represented by a set of candidate positions of the robot which are called particles [16].

The SWM is realized by first building a local world model on each agent which tracks the position of objects relative to the agent. Second, the local world model is sent to all agents, including itself, together with the agent's multiple position hypotheses: its particles. Each agent then updates its SWM by projecting the local world models it receives on the global position of the agent which supplied it. For this it needs to select the most likely position hypothesis of the other agent and the certainty associated with this hypothesis. When no acceptable match is found the local world model is discarded. Additionally, the SWM is used to improve the position estimate of each agent. This is realized by building on each agent an additional SWM composed only of local world models received from other agents. By matching the local world model of the agent with this SWM, new position candidates can be deduced. These position candidates are used, together with the position candidates derived from the static objects (the lines in the environments and the goals), to improve the position estimate.

In a command and control setting a visualization and interaction environment should incorporate the shared world model. This can be the SWM of each robot or a centralized variant in which all information is fused. The user can select the robot he is interested in. The visualization will show the objects in this robot's world model alongside the camera images and localization uncertainty of a robot. The visualization environment allows for human interaction and even direct control of the robot, which could be presented in downscaled form on a PDA. We developed an interactive robot 3D visualization that gives a nice overview and provides a great experimental environment. It has provided new insights into the behavior and limitations of the current RoboCup Middle-size software base [8].

4 Coordinating a multi-robot team

How can intelligent real world multiagent systems cooperatively solve a task? A multiagent (or multi-robot) system is a collection of agents that coexist in an environment and interact with each other. We are interested in fully cooperative multi-robot systems in which all robots share a common goal. A shared world model can facilitate the cooperation within such robot teams. We have shown how to coordinate the actions of a multi-robot team by assigning roles to the robots and applying a coordination graph to the problem [6]. Roles are a natural way of introducing domain prior knowledge to a multiagent problem and provide a flexible solution to the problem of distributing the global task of a team among its members. In the soccer domain for instance one can easily identify several roles ranging from 'active' or 'passive' depending on whether an agent is in control of the ball or not, to more specialized ones like 'striker', 'defender', 'goalkeeper', etc. Such an assignment of roles provides a natural way to parametrize a coordination structure over a continuous domain. The intuition is that, instead of directly coordinating the agents in a particular situation, we assign roles to the agents based on this situation and subsequently try to 'coordinate' the set of roles.

One approach to efficiently perform this coordination involves the use of a *coordination graph (CG)* [2]. In this graph, each node represents an agent, and an edge indicates that the corresponding agents have to coordinate their actions. In order to reach a jointly optimal action, a variable elimination algorithm is applied that iteratively solves the local coordination problems one by one and propagates the result through the graph using a message passing scheme. In a context-specific CG the topology of the graph is first dynamically updated based on the current state of the world before the elimination algorithm is applied [3]. Figure 2 shows such an updated coordination for a typical RoboCup situation, where the

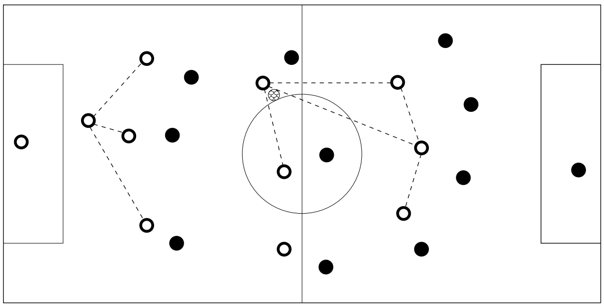

Fig. 2. Coordination graph for a typical RoboCup soccer simulation situation. On the left a coordinated defense is shown, and on the right an offense maneuver is planned.

defense and offense of the game are automatically separated by conditioning on the context: the location of the ball.

We applied coordination graphs successfully in our RoboCup simulation team by manually specifying both the coordination dependencies and the associated payoffs using value rules [6]. This resulted in the world champion title in the RoboCup-2003 soccer simulation league, illustrating that such a representation can capture very complex and effective policies.

Recently we extended this work by allowing the agents to *learn* the value of the different coordination rules [7]. We have demonstrated how Q-learning, a well known reinforcement learning technique [15], can be efficiently applied to such multiagent coordination problems. In many problems agents only have to coordinate with a subset of the agents when in a certain state (e.g., two cleaning robots cleaning the same room). We have proposed a multiagent Q-learning technique, *Sparse Cooperative Q-learning*, that allows a group of agents to learn how to jointly solve a task given the global coordination requirements of the system.

5 Robotic planning in uncertain environments

As autonomous robots are being applied in more and more complex domains the need grows for tractable ways of planning under uncertainty. In order for a robot to execute its task well in a real world scenario it has to deal properly with different types of uncertainty: a robot is unsure about the exact consequence of executing a certain action and its sensor observations are noisy. Robotic planning becomes even harder when different parts of the environment appear similar to the sensor system of the robot. In these partially observable domains a robot needs to explicitly reason with uncertainty in order to successfully carry out a given task.

As such this planning problem can be seen as a Partially Observable Markov Decision Process (POMDPs) [4], with several applications in operations research [13], artificial intelligence [4], and robotics [12, 1, 18]. The POMDP defines

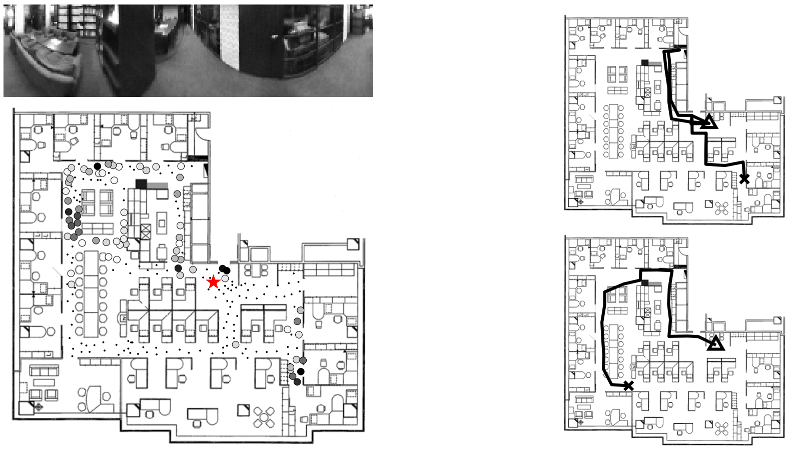

Fig. 3. Delivery task in an office environment. On the top left an example observation, below the corresponding observation model, relating observations to states. The darker the dot, the higher the probability. On the right example trajectories computed by Perseus. Start positions are marked with $\times$ and the last state of each trajectory is denoted by a $\triangle$.

a sensor model specifying the probability of observing a particular sensor reading in a specific state, and a stochastic transition model which captures the uncertain outcome of executing an action. In many situations a single sensor reading does not provide enough evidence to determine the complete and true state of the system. The POMDP framework allows for successfully handling such situations by defining and operating on the *belief state* of a robot. A belief is a probability distribution over all states and summarizes all information regarding the past. Solving a POMDP now means computing a policy—i.e., a mapping from belief states to actions—that maximizes the average collected reward of the robot in the task at hand. Such a policy prescribes for every belief state the action that maximizes the expected reward a robot can obtain in the future.

Unfortunately, solving a POMDP in an exact fashion is an intractable problem. Intuitively speaking, looking one time step deeper into the future requires considering each possible action and each possible observation. A recent line of research on approximate POMDP algorithms involves the use of a sampled set of *belief points* on which planning is performed (see e.g., [9]). The idea is that instead of planning over the complete belief space of the robot (which is intractable for large state spaces), planning is carried out only on a limited set of prototype beliefs that have been sampled by letting the robot interact with the environment. We have developed along this line a simple randomized approximate algorithm called *Perseus* that is very competitive to other state-of-the-art methods in terms of computation time and solution quality [14, 19].

We applied this approach to an office delivery task involving a mobile robot with omnidirectional vision in a highly perceptually aliased office environment, where the number of possible robot locations is in the order of hundreds. Figure 5 (left) shows the office environment, together with one of the omnidirectional camera images. We have shown how Perseus can be applied to such robotic planning problems. Robots typically have to deal with large state spaces, high dimensional sensor readings, perceptual aliasing and uncertain actions. We defined a mail delivery task in which a simulated robot has to deliver mail in an office environment. We used PCA to project the omnidirectional camera images the robot observes to a low-dimensional space, in order to able to handle them efficiently. The POMDP requires a discrete observation space, thus we perform clustering in the projected space to extract observation prototypes. We have shown our algorithm can successfully solve the resulting POMDP model. Figure 5 (right) plots two example trajectories. They show the computed policy directs the robot to first move to the pickup states, pick up the mail, and then move to the delivery locations in order to successfully deliver the mail.

6 Conclusions and future developments

In this paper we have reported on research on several aspects of cooperative real world multiagent systems. In this field robot soccer can be seen as a real scientific challenge, which is representative for the application of real world multiagent systems in practical dynamic situations. robot soccer competitions form a platform to compare different approaches to these problems and to evaluate them in practice.

We presented three of these problems. First the shared world model of the robot, which fuses observations from different robots and improves their position estimates. We presented our research on coordination within teams of robots which focuses on the use of coordination graphs [6] and extended it by allowing the agents to learn the value of coordination rules [7]. We described our algorithm for planning in environment in which a robot is unsure about the exact consequence of executing a certain action and in which its sensor observations are noisy [14].

At the moment we are working on the following directions: First, we study anytime algorithms for multiagent action selection in coordination graphs, in particular algorithms that involve distributed message-passing techniques. Second, we conduct experiments to determine the usefulness of humans interacting with the team of robots through visualization in improving the accuracy of the shared world model. Finally, we are planning to extend our work on efficient POMDP algorithms to multiagent teams.

Acknowledgments

This work is supported by PROGRESS, the embedded systems research program of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the Technology Foundation STW, project AES 5414.

References

1. Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.
2. C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems 14*. The MIT Press, 2002.
3. C. Guestrin, S. Venkataraman, and D. Koller. Context-specific multiagent coordination and planning with factored MDPs. In *Proc. 8th Nation. Conf. on Artificial Intelligence*, Edmonton, Canada, July 2002.
4. L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.
5. Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. RoboCup: The robot world cup initiative. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 340–347. ACM Press, 1997.
6. Jelle R. Kok, Matthijs T. J. Spaan, and Nikos Vlassis. Noncommunicative multi-robot coordination in dynamic environments. *Robotics and Autonomous Systems*, 2004. In press.
7. Jelle R. Kok and Nikos Vlassis. Sparse cooperative Q-learning. In Russ Greiner and Dale Schuurmans, editors, *Proc. of the 21st Int. Conf. on Machine Learning*, pages 481–488, Banff, Canada, July 2004. ACM.
8. M. Koutek, M. T. J. Spaan, and B. Terwijn. Construction and 3d visualization of the shared world model in the robosoccer context. Technical Report NIFEWVU20082004, Faculty of Sciences, Vrije Universiteit, 2004. In preparation.
9. J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc. Int. Joint Conf. on Artificial Intelligence*, Acapulco, Mexico, August 2003.
10. RoboCup official site. http://www.robocup.org.
11. R. Siegwart and I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.
12. Reid Simmons and Sven Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1080–1087, 1995.
13. E. J. Sondik. *The optimal control of partially observable Markov decision processes*. PhD thesis, Stanford University, 1971.
14. Matthijs T. J. Spaan and Nikos Vlassis. A point-based POMDP algorithm for robot planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2399–2404, New Orleans, Louisiana, 2004.
15. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
16. B. Terwijn, J.M. Porta, and B.J.A. Kröse. A particle filter to estimate non-markovian states. In F.C.A. Groen, editor, *International Conference on Intelligent Autonomous Systems, IAS'04*, pages 1062–1069. IOS Press, March 2004. ISBN 1-58603-414-6.
17. A. Tews and G.F. Wyeth. Thinking as one: Coordination of multiple mobile robots by shared representations. In *International Conference on Robotics and Systems (IROS 2000)*, pages 1391–1396, 2000.
18. G. Theocharous and S. Mahadevan. Approximate planning with hierarchical partially observable Markov decision processes for robot navigation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington D.C., 2002.

19. Nikos Vlassis and Matthijs T. J. Spaan. A fast point-based algorithm for POMDPs. In *Benelearn 2004: Proceedings of the Annual Machine Learning Conference of Belgium and the Netherlands*, pages 170–176, Brussels, Belgium, January 2004. (Also presented at the NIPS 16 workshop 'Planning for the Real-World', Whistler, Canada, Dec 2003).
20. G. Weiss, editor. *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.