Real world multi-agent systems: information sharing, coordination and planning

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

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

Abstract. Applying multi-agent systems in real world scenarios requires several essential research questions to be answered. Agents have to perceive their environment in order to take useful actions. In a multi-agent system this results in a distributed perception of partial information, which has to be fused. Based on the perceived environment the agents have to plan and coordinate their actions. The relation between action and perception, which forms the basis for planning, can be learned by perceiving the result of an action. In this paper we focus these three major research questions

First, we investigate distributed world models that describe the aspects of the world that are relevant for the problem at hand. Distributed Perception Networks are introduced to fuse observations to obtain robust and efficient situation assessments. Second, we show how coordination graphs can be applied to multi-robot teams to allow for efficient coordination. Third, we present techniques 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

Service robots, transportation systems, exploration of hazardous environments, homeland security and rescue in disaster scenarios [23] are examples where intelligent multi-agent systems could be deployed in real world situations. The societal and economical benefits of making such systems are huge, while at the same time there are still important research questions to be answered before these systems can be applied. Building these systems requires the integration of many technologies such as mechatronics, control theory, computer vision, self-learning systems and cooperative autonomous systems [16]. These agents are "intelligent on-line embedded systems" which are able to operate in dynamic environments inhabited by humans. Local intelligence and mutual communication make systems robust to erroneous perception or malfunctioning of robots.

How to evaluate these complex systems is not an easy question. The current trend to enable comparison of algorithms for parts of the system is to make the data used available on Internet, besides reporting on the algorithms and their results in scientific journals. However, the evaluation of complete real world multi-agent systems is much more complex because it is almost impossible to capture dynamic real-world aspects in static data on the Internet. Simulation is certainly useful in this respect, but these are only an abstraction of reality, and robust comparisons require the deployment of systems in real world scenarios. It has been recognized that international challenges may play an important role in those evaluations. An example is the DARPA Grand Challenge: a race for autonomous ground vehicles through desert-like terrain. A challenge formulated in multi-agent collaboration is the RoboCup challenge [6, 15]: to have in 2050 a team of humanoid robots playing a soccer match against a human team.

In section 2 we will discuss challenges for real world multi-agent systems and the research topics involved. To interact with their environment agents have to perceive it. In a multi-agent system this results in a distributed perception of partial information, which has to be fused. Next, Agents have to plan and coordinate their actions, which are based on the perceived environment. The relation between action and perception, forming the basis for planning, can be learned by perceiving the result of an action. In this paper we focus these three research questions, which are addressed in the successive sections in more detail. In section 3 we will discuss distributed world models. Such models form the basis for planning and learning to coordinate the multi-robot team. In robocup these distributed models are shared maps, which form the basis of localization and navigation of the robot-agents. In crisis management scenario's distributed world models facilitate efficient and reliable situation assessment relevant for real world decision making processes. We introduce distributed perception networks [12], that use distributed causal models to interpret large amounts of information. Section 4 explores the framework of coordination graphs for solving multi-agent 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 Challenges for real world multi-agent systems

In this paper we address some of the challenges of two types of real world multi agent settings: real world robots and distributed situation assessment systems. 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.

2.1 Real World Robotics

Multi-robot systems in dynamic environments have to cope with several substantial problems. These are summarized in RoboCup which introduces standard challenge settings that allow for an objective comparison of different solutions.

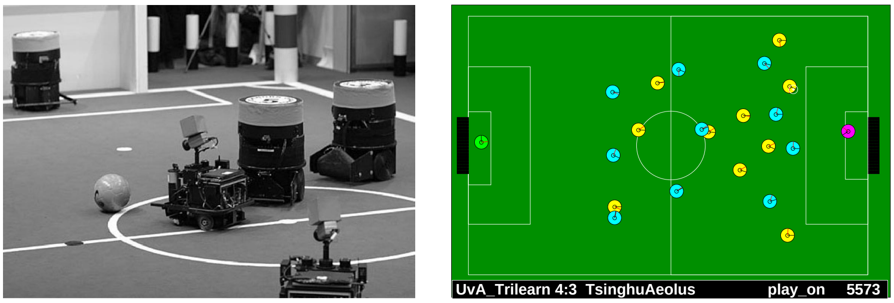

Fig. 1. Two RoboCup leagues: on the left the middle-size robots, on the right the simulated soccer agents.

RoboCup's main challenge is to develop a team of humanoid robots playing soccer that is capable of defeating the human world champion in 2050. Competitions in multiple leagues offer the possibility to focus research on different aspects of this challenge.

- In the small-size league each team consists of five small robots of about 15 centimeters in diameter. The ball and the robots are the color coded to facilitate the recognition from the images of a central camera above the field. Since the position of the robots and the ball is known quite accurately, research focuses on robot coordination, team behavior and real time control.
- The robots in the Middle-size league are bigger (about 50 centimeters), see Fig. 1 (left). The objects are again color coded. 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. To enable cooperative team behavior robots have to know where they and the other robots are on the field. So self-localization is a key issue.
- Similar research topics are present in the Sony Legged robot league, where teams of four Sony AIBO's (the well-known robotic toy dogs) compete. These robots walk on four legs. Since every team uses the same robots, the only difference between the teams is in the software.
- In the humanoid league research focuses on the development of robots with a human-like body with the abilities to play soccer against each other. There are two classes: KidSize (30-60cm height) and TeenSize (65-130cm height). Technical challenges involve topics such as penalty kicking, dynamic walking, dribbling and passing.
- The simulation league 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 these screen players are able to perform on a far more advanced level. This enables the teams to concentrate on cooperative team behavior and tactics.

2.2 Multi-agents in Automated Situation Assessment Applications

Situation assessment is indispensable for complex decision making by agents or humans. For example, consider a crisis management scenario, where the decision makers must react to a hazardous situation that takes place after a toxic gas escaped from a chemical plant. Clearly, the crisis managers must be informed about the presence of the gas as quickly as possible. Unfortunately, the gas cannot be observed directly. Instead, situation assessment, i.e. reasoning about the presence of toxic gases, requires interpretation of different types of observations that might result from hidden causes.

In a typical crisis management scenario the presence of a gas could be inferred through interpretation of large quantities of heterogeneous observations obtained through the existing sensory, communication and data storage infrastructure. For example, relevant observations could be obtained from chemical sensors installed in the plant's vicinity and through human reports about smell, haziness, irritation, etc. In addition, Unmanned Aerial Vehicles equipped with sophisticated sensor suites could provide valuable information on the gas concentration in the plant's vicinity.

Such an interpretation of the observations is not trivial, because we often have to deal with a great number of data of different types and often of a low quality. Clearly, the accuracy as well as the efficiency of such interpretation is crucial for adequate decision making where misleading or delayed state estimation can have devastating consequences.

Standardized challenge settings in this area are still being developed. An example is Robot Rescue: the search and rescue for large scale disasters, e.g., searching for survivors after earth quake disasters [12]. This challenge started as a simulation project but now also involves a real environment developed by National Institute of Standards and Technology.

3 Distributed world models

Typical multi-agent systems in real world applications interact with their environment in different ways, which requires knowledge of the relevant states in the world as well as general knowledge about the relevant processes. Such knowledge is captured in appropriate world models which, dependent on the application, make different types of knowledge explicit. To make a multi-agent system robust to failure of an agent or of the communication, world models are distributed throughout the system of communicating agents. Each agent computes a world model by itself from its limited perception and communication with other agents.

3.1 Distributed Perception Networks

Distributed world models play a central role in *Distributed perception networks*, which are multi-agent Systems for the fusion of large amounts of heterogeneous and uncertain information [12]. A Distributed Perception Network is essentially an organization of agents which support robust and efficient situation assessment through interpretation of information that can be accessed through sensory systems, databases, GSM networks and the world wide web.

The interpretation of the observations is based on causal Bayesian networks, probabilistic models which describe uncertain causal relationships between different phenomena. In a large class of situation assessment problems we can identify sequences of hidden events causing observable events [1]. For example, the presence of a toxic gas will result in a specific conductivity of ionized air which can be measured with sensors, exposed persons will perceive a typical smell and might develop certain health symptoms, which in turn will result in reports. Bayesian networks provide theoretically rigorous and compact mappings between hidden causes of interest and observable effects. By using these networks we can infer hidden causes through backward reasoning, from symptoms to their causes.

Moreover, such causal models are distributed throughout systems of communicating agents. Agents implement local world models encoded through Bayesian Networks, which represent basic modeling building blocks. In other words, each agent supports a limited expertise about the domain. Each agent updates its belief over events represented by a single variable. An agent computes a probability distribution over a local variable by using the local causal Bayesian network and a set of inputs. The inputs might be observations (e.g. sensor reports) or probability distributions over certain random variables supplied by other agents.

Belief propagation in a system of agents can be viewed as a combination of several types of algorithms, handling different types of fusion problems [2]. Such belief propagation supports exact inference which (i) is independent of the order of evidence instantiations, (ii) does not require any centralized fusion control and (iii) can efficiently cope with changing network structures at runtime. This is achieved by designing local Bayesian networks in such a way that each agent can compute a probability distribution over its fusion result by processing its local input independently of other agents.

By distributing the world models as well as the inference processes throughout systems of agents, we can often prevent processing and communication bottlenecks as well as a single point of failure.

Also, each Distributed Perception Network is specialized for a particular fusion task, which requires a specific world model that explicitly captures every piece of available evidence and maps it to the hypotheses of interest. Since we deal with applications where the information sources are not known in advance and their constellations can change at runtime, it is impossible to find an adequate causal model prior to the operation. Instead, the information sources are discovered at runtime and the agents assemble local probabilistic world models into adequate distributed Bayesian networks on the fly. In other words, a domain model is assembled out of basic building blocks with clear interfaces on an as needed basis.

---

[1] In this paper an event is synonymous to a realization of a certain situation (i.e. a state)

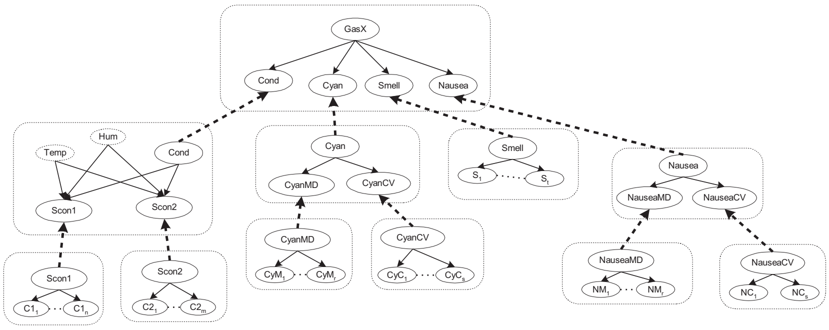

Fig. 2. A Distributed Perception Network that fuses information about the existence of high concentration of a toxic gas. Each dotted rectangle represents an agent. Thick dashed lines represent communication between agents, sharing partial fusion results. Each agent makes use of a local Bayesian Network.

In addition, through the modularity of Distributed Perception Network the design and maintenance of fusion systems are simplified. Simple partial world models can be obtained from different experts or machine learning processes. By complying to few design conventions, simpler models can easily be integrated into complex fusion structures that support very robust belief propagation. Thus, we avoid coordination of many different experts, which would be necessary if the fusion were based on centralized (not distributed) Bayesian networks. In addition, smaller models are easier to generate and fusion systems consisting of Distributed Perception Networks agents can easily be maintained. If the expertise about a certain sub-domain changes, only the local Bayesian networks implementing that expertise need to be replaced. Also, rigorous probabilistic causal models facilitate efficient approaches to distributed resource allocation [10].

Moreover, Distributed Perception Networks support accurate reasoning even if the information sources are very noisy and the modeling parameters deviate significantly from the true distributions between the modeled events. This is very relevant for real world applications, like detecting a high concentration of "Ammonia" (see Fig. 2, where we often cannot obtain precise models and information sources are not perfect. With the help of the Inference Meta Model [13], we show that Distributed Perception Network can form distributed Bayesian networks which are inherently robust w.r.t. the modeling parameters and facilitate localization of modeling parameters that do not support accurate interpretation in a given situation. Thus, we can estimate the fusion quality and signal potentially misleading results.

The assembly of theoretically sound domain models at runtime is a unique feature of Distributed Perception Networks, which allows efficient fusion of very heterogeneous information obtained from changing information source constellations.

While other recently proposed approaches to distributed information fusion [11, 24] support more general domain models than Distributed Perception Net-

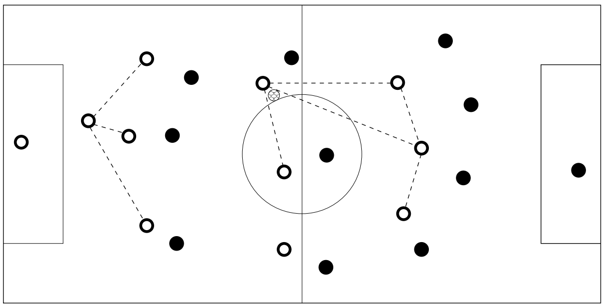

Fig. 3. Coordination graph for a typical RoboCup soccer simulation situation. On the left a coordinated defense is shown, and on the right an offense maneuver is planned.

works, they require a complete knowledge of the available information sources, which makes them unsuitable for certain types of applications, such as detection of critical situations in crisis management processes relying on ad-hoc information source constellations.

4 Coordinating a multi-robot team

How can intelligent multi-agent systems cooperatively solve a task? The agents interact with each other and coexist in an environment, that they perceive, resulting in a distributed world model. We are interested in fully cooperative multi-robot systems in which all robots have a common goal. Sharing the world model can facilitate the cooperation within such robot teams. We have shown in the past how to coordinate the actions of a multi-robot team by assigning roles to the robots and applying a coordination graph to the problem [7]. Roles are a natural way of introducing domain prior knowledge to a multi-agent problem and provide a flexible solution to the problem of distributing the global task of a team among its members. The role assignment not only reduces the number of actions that have to be considered for each agent, but can also be used to determine which agents depend on each other. 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* [3]. In this graph, each node represents an agent, and an edge indicates that the corresponding agents have to coordinate their actions. Payoff functions, defined over the actions of the connected agents, determine the effect of specific local action combinations. In order to reach a jointly (global) optimal action, a variable elimination algorithm is applied that iteratively solves the local coordination problems. For this, messages are propagated through the graph. In a context-specific coordination graph the topology of the graph is first dynamically updated based on the current state of the world before the elimination algorithm is applied [4]. Figure 3 shows such an updated coordination for a typical RoboCup situation, where the 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 [7]. 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 [8]. We have demonstrated how Q-learning, a well known reinforcement learning technique [21], can be efficiently applied to such multi-agent 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 multi-agent 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 [9].

5 Robotic planning in uncertain environments

Besides coordination agents have to plan their actions. This requires the need 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 may be noisy. Robotic planning becomes even harder when different parts of the environment cannot be distinguished by the sensor system of the robot. In these partially observable domains a robot needs to reason with uncertainty explicitly 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) [5], with several applications in operations research [18], artificial intelligence [5], and robotics [17, 1, 22]. The POMDP defines 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 framework allows for successfully handling such situations by defining and operating on the *belief state* of a robot. A belief state is a probability distribution over all states of the environment 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 aver-

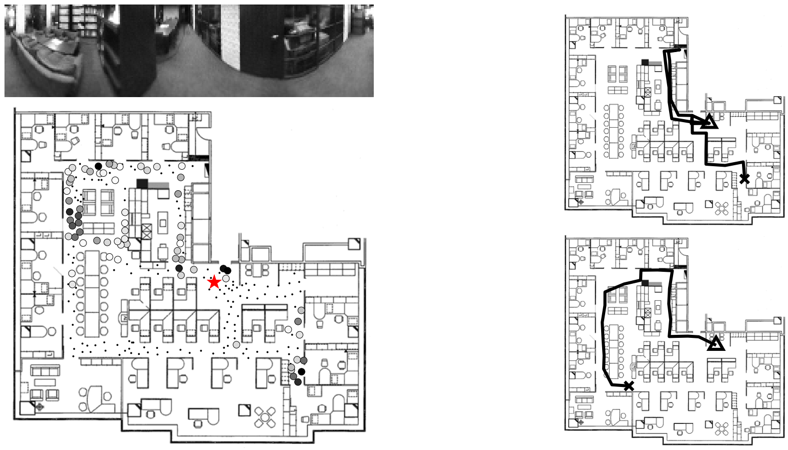

Fig. 4. Delivery task in an office environment. On the top left an example observation, below the corresponding observation model, relating observations to states. The darker the dot, the higher the probability. On the right example trajectories computed by Perseus. Start positions are marked with $\times$ and the last state of each trajectory is denoted by a $\triangle$.

age 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. The reward function encodes the robot's task and as such will be provided by the robot's designer.

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 algorithms involves the use of a sampled set of *belief points* on which planning is performed (see e.g., [14]). 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 [20].

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 [19]. Figure 4 (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 principle component analysis 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 4 (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 our research on several aspects of real world multi-agent systems.

In this field robot soccer can be seen as a real scientific challenge, which is representative for the application of real world multi-agent systems in practical dynamic situations. Robot soccer competitions is an example of a platform to compare different approaches to these problems and to evaluate them in practice.

We presented our research on coordination within teams of robots which focuses on the use of coordination graphs [7] and extended it by allowing the agents to learn the value of coordination rules [8]. We described our approach to planning in an environment in which a robot is unsure about the exact consequence of executing a certain action and in which its sensor observations are noisy [20].

A multi-agent system result in a distributed perception of partial information, which has to be fused for situation assessment in real world applications [2, 12]. We show that the distributed approaches to situation assessment (distributed perception networks), can cope with uncertain domain models and noisy/subjective information sources. In particular, we investigate how distributed causal world models can be used for efficient and reliable interpretation of large quantities of uncertain and heterogeneous information. A strong emphasis is put on the robustness of information fusion using Bayesian networks [13].

Acknowledgments

This work is supported by PROGRESS and Combined Systems. PROGRESS is 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. Combined Systems is a Senter ICT Breakthrough project, supported by the Dutch Ministry of Economic Affairs, grant number: TSIT2003. We thank the anonymous reviewer for his valuable suggestions and remarks.

References

1. Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile robot navigation. In *Proc. of International Conference on Intelligent Robots and Systems*, 1996.
2. P. de Oude, B. Ottens, and G. Pavlin. Information fusion in distributed probabilistic networks. In *Artificial Intelligence and Applications*, pages 195–201, Innsbruck, Austria, 2005.
3. C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems 14*. The MIT Press, 2002.
4. C. Guestrin, S. Venkataraman, and D. Koller. Context-specific multiagent coordination and planning with factored MDPs. In *Proc. of the National Conference on Artificial Intelligence*, Edmonton, Canada, July 2002.
5. L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.
6. 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.
7. Jelle R. Kok, Matthijs T. J. Spaan, and Nikos Vlassis. Non-communicative multi-robot coordination in dynamic environments. *Robotics and Autonomous Systems*, 50(2-3):99–114, February 2005.
8. 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.
9. Jelle R. Kok and Nikos Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, 7:1789–1828, September 2006.
10. J. Nunnink and G. Pavlin. A probabilistic approach to resource allocation in distributed fusion systems. In *Proc. Int. Conf. on Autonomous Agents and Multi-Agent Systems*, pages 846–852, Utrecht, Netherlands, 2005.
11. M. Paskin and C. Guestrin. A robust architecture for distributed inference in sensor networks. *Technical Report IRBTR -03-039, Intel Research*, 2003.
12. G. Pavlin, M. Maris, and J. Nunnink. An agent-based approach to distributed data and information fusion. In *IEEE/WIC/ACM Joint Conference on Intelligent Agent Technology*, pages 466–470, 2004.
13. G. Pavlin and J. Nunnink. Inference meta models: Towards robust information fusion with bayesian networks. In *in Proc. Int. Conf. on Information Fusion*, pages 846–852, Florence, Italy, 2006.
14. 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.
15. RoboCup official site. http://www.robocup.org.
16. R. Siegwart and I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.
17. Reid Simmons and Sven Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. Int. Joint Conf. on Artificial Intelligence*, pages 1080–1087, 1995.
18. E. J. Sondik. *The optimal control of partially observable Markov processes*. PhD thesis, Stanford University, 1971.

19. 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.
20. Matthijs T. J. Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.
21. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
22. G. Theocharous and S. Mahadevan. Approximate planning with hierarchical partially observable Markov decision processes for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington D.C., 2002.
23. G. Weiss, editor. *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.
24. Y. Xiang. *Probabilistic Reasoning in Multiagent Systems: A Graphical Models Approach*. Cambridge University Press, 2002.