Low-Overhead Cooperative Detection of False Sensor Readings

Alessandro Farinelli
Dipartimento di Informatica e Sistemistica
University of Rome "La Sapienza"
Rome, Italy
farinelli@dis.uniroma1.it

Paul Scerri
Robotics Intitute
Carnegie Mellon University
Pittsburgh, PA, USA
pscerri@cs.cmu.edu

ABSTRACT

A key problem for multi-robot teams is to correctly perceive features in their environment. When sensors of individual robots provide incorrect information, the team should not unnecessarily waste resources acting on the incorrect information. In this paper we propose a novel approach to cooperatively finding false sensor readings in domains such as rescue response. Our approach uses three key ideas. First, we use decision-theoretic reasoning to explicitly and continuously balance the benefits and costs of acting, considering the current estimate that a sensor reading was wrong. Second, assumptions underlying joint actions are attached to coordination messages, giving team mates an opportunity to refute those assumptions, if required. This process consistently reduces the communication overhead needed to detect incorrect sensor readings. Finally, we allow robots to use previously made sensor readings to provide arguments for or against current assumptions allowing the team to leverage the sensing it has already performed. Results in a simulation environment show that the approach is efficient, lightweight and reliable.

1. INTRODUCTION

Autonomous multi-robot systems have the potential to dramatically improve the efficiency and safety with which some tasks, e.g., disaster response, can be completed while also reducing costs and risk[6, 10]. However, for some domains, reliable sensing of key events in the environment is still not possible. When a robot falsely detects something, i.e., a *false positive*, and initiates coordinated action based on that false reading, the team incurs an unnecessary cost. On the other hand failing to detect something (false negatives) can be more costly because the team fails to act. Even when sensor readings themselves are reliable, automatically extracting high level data from a particular set of sensor readings can be a difficult task. An important example of this problem is the detection of injured civilians in a disaster response scenario[5]. The presence of heat, dust, smoke and noise can make robotic sensing of these civilians difficult.

Previous work on distributed Markov Decision Processes (MDPs) and Partially Observable MDPs has been effective at dealing with uncertain sensors in the context of coordination[11]. However, their computational complexity is extremely high, making them infeasible for all but small teams. Other approaches to coordination of robots in dynamic domains have either ignored uncertainty or taken "pre-processing" steps to remove uncertainty before coordinating and then excluded uncertainty in coordination[13, 2, 14, 8, 1, 15, 3]. Some approaches centralize and combine sensor readings assuming that the integrated result is correct[2, 14]. Other approaches assume individual sensor readings are correct until contradictory sensor readings are made[8, 15, 3]. For our domains of interest these approaches are unacceptably expensive or inefficient.

We propose a novel approach to this problem embodying three key ideas. First, robots use decision theoretic to continually balance the need to act against the possibility action is based on incorrect sensor readings. As new information is received, the robot updates its confidence in the assumptions that led it to act. By explicitly modeling sensor failure rates and having an explicit model of the costs of acting incorrectly, the robot can perform a straightforward calculation about whether to continue to act. If later information drops the confidence in the information below some level, action will be halted. This approach address the trade-off between the costs and risks of acting based on uncertain sensors.

Second, when it sends messages to other team members to coordinate action, the robot attaches the key beliefs that led it to that course of action. If another robot has current or previous sensor readings that refute or support those beliefs they are required to send back a message with the sensor reading. The robot making the original reading can reasses its confidence in its beliefs and determine whether the benefits of acting continue to exceed the risks. A Bayesian Filter is used to combine sensor readings into a numerical confidence value which is used by the decision-theoretic reasoning. Notice that there is minimal extra communication required to implement this approach.

Finally, we allow temporally earlier sensor readings (from other robots) to be used to improve confidence in current readings. By explicitly modeling world dynamics (as well as sensor failure rates) the conditional probability that a sensor reading is incorrect can take into account how the world

AAMAS'05 July 25-29,2005, Utrecht, Netherlands

was perceived earlier. For example, if many robots have traversed a hall in a disaster response domain without sensing any prone civilians, a single reading indicating the presence of such a civilian should be treated with more scepticism than if it was the only reading made. The use of these previous measurements can dramatically and cheaply decrease the cost of false postive sensor readings.

Our results in a simulation environment show that this approach is efficient, lightweight and reliable. The approach finds many false positive measurements when the density of the robots is reasonably high and the dynamics of the world is reasonably low. Although sometimes action is stopped errornously because team mates have incorrect readings, an overwhelming percentage of the stopped actions were founded on incorrect sensor readings.

2. PROBLEM

We assume a set of robots $R = \{r_1, \cdots, r_s\}$, that can navigate and acquire information from an environment, and a set of objects $O = \{o_1, \cdots, o_l\}$ that can be detected by the robots. An object is a general representation of an interesting feature (or event) for the robots; an object can be a human being, a device; another robot or even a fire. The environment is represented using a regular $N \times M$ grid, E. In each cell, $e_{i,j}$, one of the objects O may be present. We assume that environment is dynamic, therefore objects may change position, appear and disappear from the world.

A random variable $O_{e,t} = \{O \cup \epsilon\}$ represents whether an object is present in the cell e of the environment at time step t; the ϵ value represents no object is present. The event $O_{k,e,t}$ is defined as an object appearing ($k \in [1, l]$) or not ($k = \epsilon$) in cell e at time t and has probability $Pr(O_{e,t} = O_k)$. A second random variable $Z^i_{e,t}\{O \cup \epsilon\}$ represents the observation of robot i at time step t for cell e; similarly we define the event $Z^i_{k,e,t} = Pr(Z_{e,t} = O_k)$.

Robots have a model of how the environment evolves and a model of observations. In particular robots know the transition probabilities of the environment: $Pr(O_{k,e,t}|O_{k,e,t-1})$ and the probabilities of obtaining an observation from the environment: $Pr(Z^i_{k,e,t}|O_{k,e,t})$.

The feature level extraction process can result in two types of error: false negatives and false positives. When an object is perceived in a given position and it is not present we have a false positive, while when an object is not detected we have a false negative. The probabilities of both these types of errors can be expressed as: $Pr(Z^i_{k,e,t}|\neg O_k, e, t)$ and $Pr(\neg Z^i_{k,e,t}|O_k, e, t)$, respectively. We assume independence between the failure rates of different robots. This assumption is well suited for team of heterogeneous robots which have different sensors, but is more reasonable for false positive readings than for false negative ones. We do not currently explicitly model movement of objects in the environment, i.e., an object is no more likely to be near where it was previously than in any other location.

The costs and benefits of acting are modeled using decision theory. The team pays a cost, C_{WIP}, when a plan is incorrectly instantiated (WPI) and gets a reward R_{CPI} when a plan is correctly initiated (CPI). Each robot, when it detects (correctly or not) a feature that requires action can choose from three possible coordination actions: initiate a relevant plan (IP), ask team mates whether they can provide useful information (ATM) or instantiate an active perception plan (IAP). The IP action will entail messages being exchanged with team mates to coordinate the execution of the plan and an initial commitment of resources. The ATM action entails a communication and processing cost for the team and may not be successful, if other team mates do not have related information. The IAP action is a high cost action that initiates a joint effort to establish the validity of the sensor reading. While the cost is high, we assume that IAP has high gain in terms of information acquisition. The costs for these actions are written C_{IP}, C_{ATM} and C_{IAP}, respectively.

The information gain is represented by conditioning the probability of success of the action on the expected volume of readings resulting from the action. For example, for IP the expected number of readings is proportional to the number of robots coming to know about the plan and the ratio of robots that have related information. We represent this value as Z_{IP} and thus after performing the coordination action IP the probability of CPI will be given by $P(CPI|Z_{IP})$.

2.1 Example

A motivating example is a set of heterogeneous robots, $R = \{r_1, \ldots, r_n\}$ tasked to search a building engulfed by fire. O are trapped civilians and the environment is an office building. Due to smoke and heat, sensing is difficult, thus $Pr(Z^i_{O,e,t}|O_O, e, t) < 1.0$ and $Pr(Z^i_{\epsilon,e,t}|O_\epsilon, e, t) < 1.0$. When a civilian is detected a plan is initiated to either lead the person to a safe area of the building or to direct human fire fighters to the victim. If either civilians are not correctly sensed or fire fighters are unnecessarily sent into the building (WPI) there is a high cost. If $|R|$ is large then the ATM action may be useful since other robots may have passed the same location and used different sensor suites to draw a different conclusion. Finally, because of the time-critical nature of the rescue task and the advantage to spreading out robots for search, the IAP action has high cost.

3. ALGORITHM

The key to this approach is to use input from team mates to improve a model of the world, while continuing to act on initial sensor readings. When a robot initiates a plan, messages are sent to its teammates to coordinate execution. Our approach requires that the assumptions leading to the plan initiation are appended to these coordination messages. When a team member receives a coordination message, it checks whether it has recent sensor readings to refute or support the rationale for the plan. If it does, it sends a message containing the information for or against the assumptions back to the initiator of the plan. The plan initiating robot uses this information to update its confidence in its beliefs and reconsider whether to continue to act. Notice that this method for cooperatively detecting false positive sensor readings makes minimal assumptions about the underlying coordination algorithm.

Algorithm *ObsManagement* shows pseudo code for a robot. Line 1 initialize the world state WS and the set of instantiated plan set $InstPlan$. The WS is a representation each agent maintains of E. Each cell of WS contains a tuple representing the state of the cell and the last time the cell has been updated, while $InstPlan$ represents all the coordination plans instantiated by the agent. Whenever the robot receives a message msg it checks whether msg is a valid observation (Line 4). An observation is valid if it is inside a specified time window, T, which is computed by determining whether the probability that the world changed since the time the observation was made has been received, is less than a specified threshold VOT (Valid Observation Threshold). If the observation is valid the robot updates its world state using a Bayesian filter which also factors in the evolving world state. Next the robot determines, using a decision theoretic approach, whether to instantiate a plan related to the new observation (line 12), $EU(a)$ returns the expected utility of performing coordination action a. The expected utility of a coordination action is based on robots knowledge about the environment, and thus it can change as robots acquire new information. If the robots decide to instantiate a plan, it checks whether the plan has been already started and if not it starts the plan (line 14). Using the same reasoning, continuing execution of an initiated plan maybe determined to be a wrong decision (line 17); in this case the plan is stopped. Alternatively, the decision can be to ask team mates for more information about a particular part of the environment (line 19) or whether to initiate a IAP (line 21).

```
ObsManagement()
(1)    WS ←< ε, 0 >, InstPlan ← ∅
(2)    while true
(3)      msg ← getMsg()
(4)      if msg is ValidObs
(5)        UpdateWS(msg.obs)
(6)    else
(7)      /*Msg is coordination Msg*/
(8)      if Relevant(WS, Msg)
(9)        send(obs,msg.sender)
(10)     evolveWS()
(11)     foreach e ∈ WS
(12)       if EU(IP(e)) > EU(¬IP(e))
(13)         if Plan(e) ∉ InstPlans
(14)           InstantiatePlan(msg)
(15)       else
(16)         if Plan(e) ∈ InstPlans
(17)           StopPlan(msg)
(18)         else if EU(ATM(e)) > EU(¬ATM(e))
(19)           DoPlan(ATM(e))
(20)         else if EU(IAP(e)) > EU(¬IAP(e))
(21)           if Plan(IAP(e)) ∉ InstPlans
(22)             InstantiatePlan(IAP(e))
(23)           else if Plan(IAP(e)) ∈ InstPlans
(24)             StopPlan(IAP(e))
```

Bayesian update of agents knowledge

The belief update of each agent is done using a Bayes filter. Specifically, a filter is instantiated for each cell of E, as indicated in equation 1.

$$Bel'(O_{e,t}) = \sum_{O_{e,t-1}} Pr(O_{e_t}|O_{e,t-1})Bel(O_{e,t-1}) \tag{1}$$

$$Bel(O_{e,t}) = \eta \prod_i Pr(Z^i_{e,t'}|O_{e,t})Bel'(O_{e,t}) \tag{2}$$

Since readings obtained from team mates $Z^i_{e,t'}$ maybe older than present time $t' < t$, they cannot be directly integrated using the filter equation because it may refer to a time step in the past and therefore should have influenced state probabilities up to the present time. We address this problem maintaining an history of observations $ZH = \{Z^i_{e,t}\}$ acquired by agents along with corresponding cell state $OH = \{O_{e,t}\}$ and reinitializing the filter when an old reading is obtained. When a reading referring to a past time t' is obtained we reinitialize the filter with a state $O_e, \bar{t}$ where $\bar{t} = Max\{t \mid O_{e,t} \in OH \wedge \bar{t} < t'\}$ and incorporate all observation present in ZH from $\bar{t}$ to the present time. However, maintaining the history of observations and states probabilities for all the process has a cost in term of memory that grows with time. To limit such cost we define a valid time window for observations T that goes from the current time t_c back to $t_c - T$ time units. The time window's size can be defined according to the evolution model of the environment. In particular, assuming objects are all of one type, then state of cell can be only present P or not present NP. Assuming $Pr(O_{e,t} = P|O_{e,t-1} = NP) = Pr(O_{e,t} = NP|O_{e,t-1} = P) = C$ then we can write the state evolution as: $Pr(O_{e,t} = P) = C + (1-2C)Pr(O_{e,t-1} = P)$.
Let $X_t = Pr(O_{e,t} = P)$ then we have:

$$X_0 = Pr(O_{e,0} = P)$$

$$X_1 = C + (1-2C)X_0$$

$$\vdots$$

$$X_n = C + (1-2C)X_{n-1}$$

Let $1 - 2C = a$ We can re-write this as

$$X_n = (1-a)/2 \sum_{i=0}^{n-1} a^i + a^n X_0$$

We can now find the n for which $Pr(O_{e,t} = P|O_{e,0} = P) \geq VOT$ as

$$Pr(O_{e,t} = P|O_{e,0} = P) = (1-a)/2 \sum_{i=0}^{n-1} a^i + a^n \geq VOT$$

The series $\sum_{i=0}^{n-1} a^i$ is a geometric series with $|a| \in [0,1]$ therefore from series theory we can write

$$VOT \leq (1-a^n)/2 + a^n = (1+a^n)/2$$

Thus,

$$n \leq log_a(2VOT - 1) \Rightarrow n = \lfloor (log_a(2VOT-1)) \rfloor \tag{3}$$

Notice that the same calculation can be done considering $X_t = Pr(O_{e,t} = NP)$, therefore the computed n represent the time window for which an information (e.g. presence or absence of an object) can be considered relevant given our assumptions on the world model.

4. EXPERIMENTS AND RESULTS

To evaluate our approach we implemented the proposed method in a simulator that captures key features of the environment while being sufficiently abstract to test a wide range of parameters and configurations efficiently. The simulated environment is composed of a grid world where objects appear and disappear spontaneously. The simulated robots have limited knowledge of the overall team state and can communicate with only a subset of the overall team. When a robot perceives an object it instantiates a coordinated plan for that feature. A task assignment algorithm[12] is initiated to allocate tasks in that plan. The algorithm uses *tokens*, representing each task, that get propagated through the team until a robot accepts that task. To each task assignment token an argument for the plan instantiation is

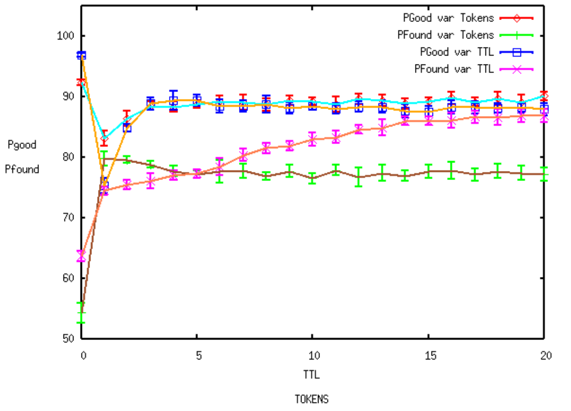

Figure 1: Varying TTL and number of Tokens

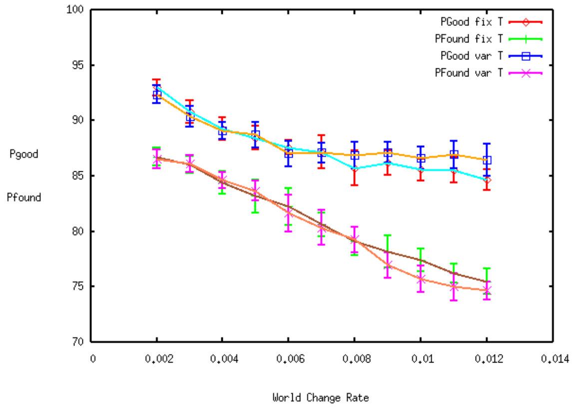

Figure 2: Comparison among fixed observation time window (T) and computed T according to theory, varying world dynamics

attached, stating why the plan has been started (e.g. object found in cell i). We model both the number of tokens sent (equal to the number of tasks in the joint action) and the number of steps the token takes through the team (TTL).

To evaluate the performance of the distributed sensing procedure we measure the percentage of stopped plans out of the total amount of plans incorrectly instantiated (*percentage found*) and the percentage of correctly stopped plans out of all the plans that have been stopped (*percentage good*). The first metric measures how well the approach works at finding the incorrectly started plans and the second measures how well the approach works at stopping only those plans that should be stopped. In both cases, higher is better and 100% is perfect. Notice, that in these experiments the robot always initiates a plan when it detects an object, hence there is no measure of how many plans were not started despite a "false positive".

In each experiment there were 100 simulated robots, each with the same perception model. Specifically, the probability of a correct positive reading was $0.5 + (0.4 * \exp^{-0.3 \times dist})$ and for a correct negative reading $0.5 + (0.4 * \exp^{-0.2 \times dist})$,

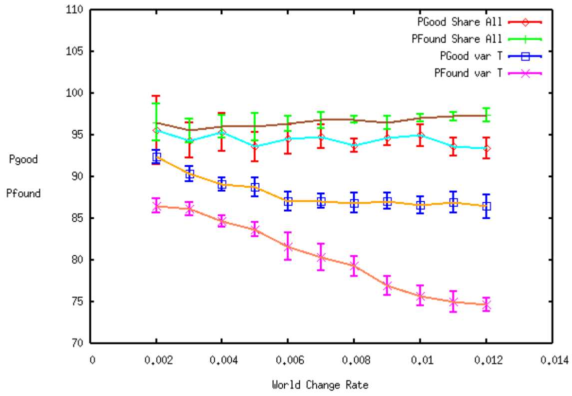

Figure 3: Comparison among Share All and our approach, varying world dynamics

where $dist$ is the robot's distance from the object. Even up close robots will made false positive readings 10% of the time. The world changed at the rate $WCR = 0.005$. These models are accurately known by each of the robots. Each graph reports values averaged over 10 runs.

Figure 1 shows the percentage of incorrectly started plans that are stopped and the percentage that are correctly stopped as the number of tokens (i.e., tasks) and their TTL (i.e., how many team members each token visited) is varied. When varying the number of tokens, TTL is 10, when varying TTL number of tokens is 5. The approach is able to detect at least 75% of the incorrectly initiated plans and is about 85% reliable at stopping plans correctly. Moreover, notice that only a small number of tokens and relatively small TTL are required to get good performance.

Figure 2 shows the results as the rate of change in the world is varied. The Figure shows the ability to find incorrect plan deteriorates as the world changes more quickly because old measurements provided by the team are less useful. However, since the filter explicitly takes into account these dynamics, there is a less pronounced fall off in the accuracy with which plans are stopped. Moreover, in the Figure we compare results obtained considering all observations robots receive with results obtained considering only observations referring to a valid time window. The valid time window is computed for each world change rate according to the theory presented above (see section 3). Results show that the performance of the approach using the valid time window computation, closely match the performance of the method using all available observations. However, using a valid observation time window, allows us to save a consistent amount of memory and computation time, because we have to store only a limited observation and cell state history.

To determine whether appending assumptions to coordination messages was reasonable, we compare to a benchmark strategy, called *Share All* where each robot shares all its sensor readings with all other robots at each step. Clearly, this is infeasible for large teams, but it provides an upper bound on the performance that can be achieved. Figure 3 shows that *Share All* does perform better and the difference increases as the world becomes more dynamic. However, for environments that are not too dynamic, our much less communication intensive approach performs almost as well using

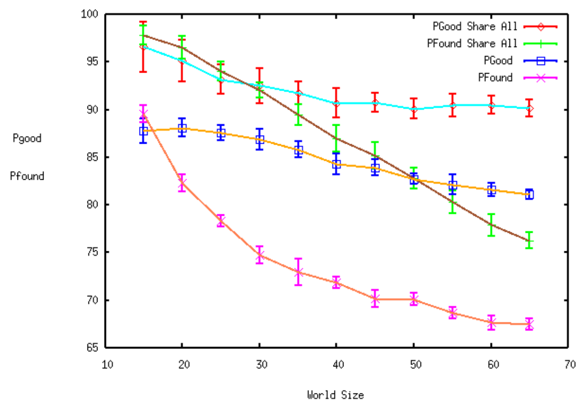

Figure 4: Comparison among Share All and our approach, varying world size

two orders of magnitude less messages.

As the size of the environment increases, the density of robots decreases hence providing less opportunity for supporting or refuting information to be provided. Figure 4 shows performance as the size of the environment is varied while keeping the number of robots and the sensor range constant. As expected, performance falls away as the robot density becomes lower.

Finally, Figure 5 shows the performance as the sensor model is varied. The x-axis shows the probability of correct sensing when very close to the object. Unsurprisingly, the better the sensors the easier it was to correctly overcome a single false measurement.

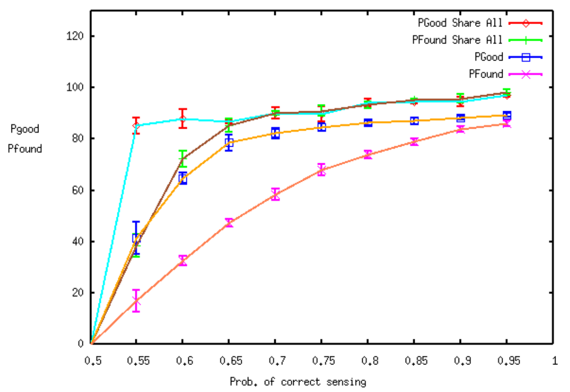

Figure 5: Varying sensor model

5. RELATED WORK

Several approaches have been used to address the problem of coordination for robotic agents embedded in dynamic environments [2, 14, 7, 4, 8, 1, 15, 9]. Some of them are explicitly focused on the problem of distributed, cooperative sensing[2, 14] and are clearly related to this work. In particular in [2] a Bayesian framework is used to address the problem of multi object tracking by a team of robots in the RoboCup soccer scenario. The authors provide both a multi object tracking algorithm based on Kalman Filter, and a single-object tracking method performed using a combination of Kalman filter and Markov localization for outliers detection. Also in [14] a Kalman filter is involved to estimate the state of a moving object, in the same domain. Both such approaches however, require a central unit to process data coming from the robotic agents. With respect to these kind of approaches our method performs the distributed sensing procedure in a fully decentralized fashion, requiring a very low communication overhead.

Many other approaches do not directly address the problem of noisy sensing at the coordination level, e.g., [4, 8, 1, 15, 3, 9]. Low-level routines are sometimes required to filter out incorrect sensor readings and at the coordination level data are considered to be reliable. In [1, 15] all tasks present in the system are reallocated at a fixed time step, so that transient false readings of team members negatively influence the performance of the system only for a limited time. However such approaches require a very high communication overhead which becomes prohibitive for large teams.

6. CONCLUSIONS

In this paper we propose a novel approach to deal with distributed noisy and unreliable perception in dynamic environments. Our approach uses a Bayesian framework to integrate readings from different robots, and is explicitly designed to operate in large scale teams. The approach enables robots to integrate previously made sensor readings to help refute incorrect sensing. Moreover, robots refine their world knowledge while executing their actions; using a decision theoretic approach robots evaluate whether the action they are performing are worth being accomplished continuously monitoring the world state. This work is a first important step to explicitly consider uncertainty at the coordination level in a tractable way, however more work is needed. Importantly, some of the independence assumptions made in this work will need to be relaxed for more realistic domain.

7. ACKNOWLEDGMENT

Alessandro Farinelli is supported by the European Office of Aerospace Research and Development under grant number 053015. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the European Office of Aerospace Research and Development.

8. REFERENCES

[1] C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo, and A. Sgorbissa. Coordination among heterogenous robotic soccer players. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS'00)*, volume 2, pages 1385–1390, 2000.

[2] Dietl, J. S. Gutmann, and B. Nebel. Cooperative sensing in dynamic environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01)*, 2001.

[3] B. Gerkey and M. J. Matarić. Sold!:auction method for multi-robot coordination. In *IEEE Transaction on Robotics and Automation*, volume 18(5), pages 758–768, October 2002.

[4] B. Horling, R. Mailler, J. Shen, R. Vincent, and V. Lesser. *Distributed Sensor Networks A multiagent perspective*, chapter Using Autonomy, Organizational Design and Negotiation in a DSN, pages 139–184. Kluwer Academic, 2003.

[5] Hiroaki Kitano, Satoshi Tadokoro, Itsuki Noda, Hitoshi Matsubara, Tomoichi Takahashi, Atsushi Shinjoh, and Susumu Shimada. Robocup rescue: Searh and rescue in large-scale disasters as a domain for autonomous agents research. In *Proc. 1999 IEEE Intl. Conf. on Systems, Man and Cybernetics*, volume VI, pages 739–743, Tokyo, October 1999.

[6] K. Konolige et al. Centibots: Large scale robot teams. In A. C. Shultz and L. E. Parker, editors, *Proc. of the NRL Workshop on Multi-Robot Systems*, volume II, pages 193–204, Washington, DC, 2003. Kluwer Academic Publishers.

[7] P. J. Modi, P. Scerri, Shen W. M., and M. Tambe. *Distributed Sensor Networks A multiagent perspective*, chapter Distributed Resource Allocation, pages 219–256. Kluwer Academic, 2003.

[8] L. E. Parker. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, April 1998.

[9] L. E. Parker. Distributed algorithms for multi-robot observation of multiple moving target. *Autonomous Robots*, 12(3), 2002.

[10] L. E. Parker. The effect of heterogeneity in teams of 100+ mobile robots. In A. C. Shultz and L. E. Parker, editors, *Proc. of the NRL Workshop on Multi-Robot Systems*, volume II, pages 205–215, Washington, DC, 2003. Kluwer Academic Publishers.

[11] David Pynadath and Milind Tambe. Multiagent teamwork: Analyzing the optimality and complexity of key theories and models. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, 2002.

[12] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe. Token approach for role allocation in extreme teams: analysis and experimental evaluation. In *13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2004).*, pages 397–402, Modena, Italy, 2004.

[13] Paul Scerri, Yang Xu, Elizabeth Liao, Guoming Lai, and Katia Sycara. Scaling teamwork to very large teams. In *Proceedings of AAMAS'04*, 2004.

[14] A.W. Stroupe, M.C. Martin, and T. Balch. Distributed sensor fusion for object position estimation by multi-robot systems. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on Volume: 2*, pages 1092 – 1098, 2001.

[15] B. B. Werger and M. J. Mataric. Broadcast of local eligibility for multi-target observation. In *DARS00*, pages 347–356, 2000.