Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem

Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme

Robotics Research Laboratories, Department of Computer Science, University of Southern California, Los Angeles, CA 90089-0781
ahoward@usc.edu, mataric@usc.edu, gaurav@usc.edu

Abstract. This paper considers the problem of deploying a mobile sensor network in an unknown environment. A mobile sensor network is composed of a distributed collection of nodes, each of which has sensing, computation, communication and locomotion capabilities. Such networks are capable of self-deployment; i.e., starting from some compact initial configuration, the nodes in the network can spread out such that the area 'covered' by the network is maximized. In this paper, we present a potential-field-based approach to deployment. The fields are constructed such that each node is repelled by both obstacles and by other nodes, thereby forcing the network to spread itself throughout the environment. The approach is both distributed and scalable.

Keywords: Distributed robotic systems, sensor networks, deployment, potential fields.

1 Introduction

This paper considers the problem of deploying a mobile sensor network in an unknown environment. A mobile sensor network is composed of a distributed collection of *nodes*, each of which has sensing, computation, communication and locomotion capabilities. It is this latter capability that distinguishes a *mobile* sensor network from its more conventional static cousins. Locomotion facilitates a number of useful network capabilities, including the ability to *self-deploy*; that is, starting from some compact initial configuration, the nodes in the network can spread out such that the area 'covered' by the network is maximized.

Our approach is motivated by the need to deploy sensor networks in environments that may be both hostile and dynamic. Consider, for example, a scenario involving a hazardous materials leak in a damaged structure. We would like our sensor network, whose nodes are equipped with chemical sensors, to rapidly deploy throughout the environment and return real-time data indicating the location and concentration of hazards. This kinds of scenario imposes two important constraints on our deployment algorithm: prior models of the environment are either incomplete, inaccurate or unavailable, and network nodes may be lost or destroyed.

In this paper, we describe a potential-field-based approach to deployment, in which nodes are treated as virtual particles, subject to virtual forces. These forces repel the nodes from each other and from obstacles, and ensure that an initial, compact configuration of nodes will quickly spread out to maximize the coverage area

H. Asama et al. (eds.), *Distributed Autonomous Robotic Systems 5*

of the network (it should be noted that nowhere do we reason about coverage explicitly; rather, coverage is an emergent property of the algorithm). In addition to these repulsive forces, nodes are also subject to a viscous friction force. This force is used to ensure that the network will eventually reach a state of static equilibrium; i.e., all nodes will ultimately come to a complete stop. The viscous force does not, however, prevent the network from reacting to *changes* in the environment; if something is moved, the network will automatically reconfigure itself for the modified environment before return once again to a static equilibrium. Thus, nodes move only when it is necessary to do so, saving a great deal of energy.

The potential field approach described in this paper relies on only one assumption: that each node is equipped with a sensor that allows it to determine the range and bearing of both nearby nodes and obstacles (suitable sensors can be constructed using scanning laser range-finder or omni-camera). Using this information, the node can determine the virtual forces acting it, and convert this information into a control vector to be sent to its motors. No other information is required. It should be emphasized that this approach *does not* require models of the environment, localization, or communication between nodes. As a result, the algorithm is both robust and highly scalable.

In the remainder of this paper, we develop the potential field theory underlying the deployment algorithm, and demonstrate that this algorithm has the desired property that the network will converge to a state of static equilibrium. We describe a series of simulation experiments that both validate the general approach and reveal some of its emergent properties. These experiments include realistic sensor-based simulations of a network containing 100 nodes in a large, complex environment.

2 Related Work

The concept of *coverage* as a paradigm for evaluating many-robot systems was introduced by Gage [5]. Gage defines three basic types of coverage: blanket coverage, where the objective is to achieve a static arrangement of nodes that maximizes the total detection area; barrier coverage, where the objective is to minimize the probability of undetected penetration through the barrier; and sweep coverage, which is more-or-less equivalent to a moving barrier. According to this taxonomy, the deployment problem described in this paper is a blanket coverage problem.

Potential field techniques for robotic applications were first described by Khatib [10] and have since been widely used in the mobile robotics community for tasks such as local navigation and obstacle avoidance. The related concept of 'motor schemas', which utilizes the super-position of spatial vector fields to generate behavior was introduced by Arkin [1]. Both techniques have since been applied to the problem of formation control for groups of mobile robots [13,2]. The formation problem is similar, in some respects, to the deployment problem described in this paper, in that the robots will attempt to maintain a formation based on local sensing and computation. A key difference, however, is that there is no requirement that the formation reach a state of static equilibrium.

The deployment problem also is also similar, in some respects, to the multi-robot exploration and mapping problem. Here, the aim is to build a global map of the environment by sequentially visiting each location with one or more robots. This problem has been considered by a number of authors [4,15,14,3] who use a variety of techniques ranging from topological matching [4] to fuzzy inference [11] and particle filters [16]. Two good examples are provided by Simmons [14] and Burgard [3], both of whom build global maps, apply heuristics to select goal locations for exploration, and use explicit communication to prevent more than one robot from heading for the same goal. This approach to *exploration* contrasts markedly with the approach to *deployment* described in this paper. As we will show in Section 4, potential field methods are able to achieve good coverage without global maps, without communication, and without explicit reasoning. Instead, area coverage is an emergent, system-level property.

Finally, we note that the problem of deployment is related to the traditional *art gallery* problem in computational geometry [12]. The art gallery problem seeks to determine, for some polygonal environment, the minimum number of cameras that can be placed such that the entire environment is observed. While there exist a number of algorithms designed to solve the art gallery problem, all of these assume that we possess good prior models of the environment.

3 Potential Fields

Potential fields are a commonly used and well understood method in mobile robotics, where they are typically applied to tasks such as local navigation and obstacle avoidance. In this paper, we apply potential fields to the deployment problem. The fields are constructed in such a way that each node is repelled by both obstacles and by other nodes, thereby forcing the network to spread itself throughout the environment.

The basic potential field method is as follows. Each node is subject to a force $\mathbf{F}$ that is the gradient of a scalar potential field U; i.e.,

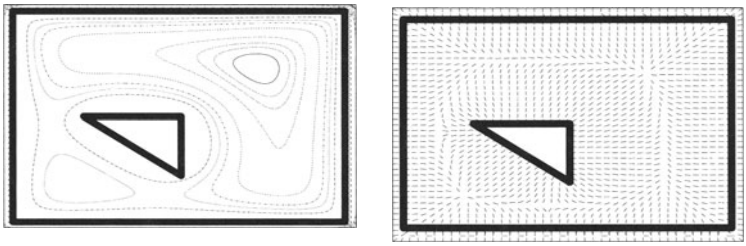
$$\mathbf{F} = -\nabla U \tag{1}$$

We divide the potential field into two components: the field U_o due to obstacles, and the field U_n due to other nodes; these fields give rise to repulsive forces $\mathbf{F}_o$ and $\mathbf{F}_n$, respectively. Thus $U = U_o + U_n$ and $\mathbf{F} = \mathbf{F}_o + \mathbf{F}_n$.

Consider the potential field due to obstacles. If we imagine that each node and each obstacle carries an electric change, we can write down an expression for the resultant 'electrostatic' potential:

$$U_o = k_o \sum_i \frac{1}{r_i}. \tag{2}$$

The summation is over all obstacles that can be seen by the node, k_o is a constant describing the strength of the field, and r_i is the Euclidean distance between the node and obstacle i. Let $\mathbf{x}$ denote the position of the node and let $\mathbf{x}_i$ denote the position of obstacle i. The distance r_i is then given by $r_i = \mid \mathbf{x}_i - \mathbf{x} \mid$. Using these



(a) (b)

Fig. 1. (a) Potential field generated by a simple environment; the contours show the lines of equal potential. (b) Force fields generated by this potential; the arrows indicate the direction (but not magnitude) of the force.

definitions, the total force F_o due to obstacles can be computed using Equation 1. We re-write the equation and expand using the chain rule, as follows:

$$\mathbf{F}_o = -\frac{\mathrm{d}U_o}{\mathrm{d}\mathbf{x}} = -\sum_i \frac{\mathrm{d}U_o}{\mathrm{d}r_i} \cdot \frac{\mathrm{d}r_i}{\mathrm{d}\mathbf{x}}. \tag{3}$$

We then insert the appropriate derivatives to obtain:

$$\mathbf{F}_o = -k_o \sum_i \frac{1}{r_i^2} \cdot \frac{\mathbf{r}_i}{r_i} \tag{4}$$

where $\mathbf{r}_i = \mathbf{x}_i - \mathbf{x}$. Note that the force is expressed entirely in terms of the relative positions $\mathbf{r}_i$ of obstacles, rather than their absolute positions $\mathbf{x}_i$. This allows us to compute the force directly from sensor data, without the need for global localization. Figure 1 shows the potential field U_o and force field $\mathbf{F}_o$ generated by a simple environment.

Consider now the potential field U_n due to other nodes. By analogy with the obstacle field, we can derive expressions for the potential U_n and force $\mathbf{F}_n$ by replacing a summation over visible obstacles with a summation over visible nodes; thus:

$$U_n = -k_n \sum_i \frac{1}{r_i} \qquad \text{and} \qquad \mathbf{F}_n = -k_n \sum_i \frac{1}{r_i^2} \cdot \frac{\mathbf{r}_i}{r_i} \tag{5}$$

where $\mathbf{r}_i$ is the relative position of node i.

3.1 The Equation of Motion and The Control Law

The trajectory of a node subject to force $\mathbf{F}$ can be computed using an appropriate *Equation of Motion*. We use an equation of the following form:

$$\ddot{\mathbf{x}} = (\mathbf{F} - \nu\dot{\mathbf{x}})/m \tag{6}$$

where $\ddot{\mathbf{x}}$ denotes the acceleration of the node and m denotes its mass. The second term on the right hand side of this equation is a *viscous friction* term in which ν is the viscosity coefficient and $\dot{\mathbf{x}}$ is the node velocity. This term is used to ensure that, in the absence of external forces, the node will eventually come to a standstill. [1] In Section 3.2 we will show that the viscous friction term also guarantees that the network *as a whole* will ultimately reach a state of static equilibrium (i.e., a state in which all nodes have stopped moving).

Our discussion up to this point has focused entirely on a *virtual* physical system, i.e., one in which the forces, accelerations, masses, etc. are entirely imaginary. This virtual physical system must, however, be mapped onto a *real* physical system made up of real nodes. Typically, these nodes will have some form of velocity controller; consequently, the mapping from virtual to real physical system is achieved by defining a *control law* that maps a virtual force onto a velocity *control vector*. In most applications using potential field techniques, such as single-robot obstacle avoidance, the control law can be entirely arbitrary; in this case, there is little to gain from preserving the correspondence between a robot's real and virtual dynamics. In our application however, it is extremely useful to retain this correspondence: if our control law is such that the nodes obey the Equation of Motion, we can be certain that the network *will* reach static equilibrium (see Section 3.2). This is not the case for any arbitrary control law.

In deriving a control law for a real node, we must be cognizant of the fact that real nodes are not 'free particles': they have both kinematic and dynamic constraints. The kinematic constraints can be largely ignored if we make the assumption that the nodes have holonomic drive mechanisms (i.e. they can move equally well in any direction). [2] The dynamic constraints, however, cannot be ignored: the node will have both a maximum velocity and a maximum acceleration, which must be captured by the control law.

Our control law can be expressed algorithmically as follows. Let $\mathbf{v}$ denote the commanded velocity at some time t, and let $\Delta\mathbf{v}$ denote the change in the commanded velocity between times t and $t + \Delta t$. The change in commanded velocity is determined using a piecewise-constant approximation to the Equation of Motion (Equation 6):

$$\Delta\mathbf{v} \longleftarrow (\mathbf{F} - \nu\mathbf{v})/m \cdot \Delta t. \tag{7}$$

The x and y components of $\Delta\mathbf{v}$ are subsequently 'clipped' such that $-a_{\max} \leq \Delta\mathbf{v} \leq a_{\max}$ where $a_{\max}$ denotes the largest allowable change in velocity. The commanded velocity $\mathbf{v}$ is determined using:

$$\mathbf{v} \longleftarrow \mathbf{v} + \Delta\mathbf{v} \tag{8}$$

[1] Strictly speaking, the node will never come to a complete stop; rather, its velocity will approach zero asymptotically.

[2] Even a standard differential drive mechanism can be treated as a holonomic platform if one is prepared to sacrifice the rotational degree of freedom. Furthermore, if one has omnidirectional sensors, this sacrifice has no functional impact.

and is then clipped to the domain $-v_{\max} \leq \mathbf{v} \leq v_{\max}$ where $v_{\max}$ is maximum allowed velocity.

Using this control law, the real node dynamics will closely approximate those described by the Equation of Motion. There are, however, two regimes in which the correspondence will fail. Firstly, for small $\mathbf{v}$, the viscous friction term will tend to produce oscillation rather than asymptotic convergence to zero velocity; this kind of behavior is typical of discrete control systems and can be eliminated by introducing a velocity 'dead-band'. Secondly, large accelerations and velocities will simply be clipped, in which case the deviation from the virtual dynamics may become arbitrarily large. This deviation is significant only if it prevents the network as a whole from reaching static equilibrium, or if it significantly increases the time taken to reach this equilibrium. We assert (without proof) that the acceleration and velocity limits act like additional non-linear friction terms, and that therefore, these limits will not prevent the system from reaching static equilibrium. The limits may, however, impact on the time taken to reach equilibrium, and this impact must be determined empirically.

3.2 Static Equilibrium

One can show that the network as a whole will reach a static equilibrium (i.e. a situation in which all nodes are stationary) by considering the total energy of the system. Each node has both potential and kinetic energy: the former arises from the node's interaction with the potential field, the latter from the node's motion. The total energy of the system is determined by summing these energies for all nodes. The Equation of Motion includes a viscous friction term that has the effect of removing energy from the system; i.e., the system is said to be *dissipative* [8]. For such systems, the total energy will decrease monotonically over time, and since the potential energy of the system is bounded from below, this necessarily implies that the total kinetic energy of the system will asymptote to zero. Clearly, therefore, the network as a whole must asymptotically approach static equilibrium.

This argument rests on the assumption that the environment itself is static, and therefore does not introduce additional energy into the system or modify the space of reachable states. In a dynamic environment, however, energy may be added to or subtracted from the system whenever an object is moved by some agency other than the network itself. Furthermore, states which where previously unreachable may now become reachable, and vice-versa. As a consequence, in an environment that is continually changing, we do not expect the network to reach a state of static equilibrium. If, however, the environment is changing *periodically* or *intermittently*, the network will reach static equilibrium, but the equilibrium state may be different after each change. Consider, for example, a network placed in a closed room: the network will deploy to fill the room and then stop. If the door to the room is now opened, the network will start deploying again, spreading beyond its original confines to seek a new equilibrium state.

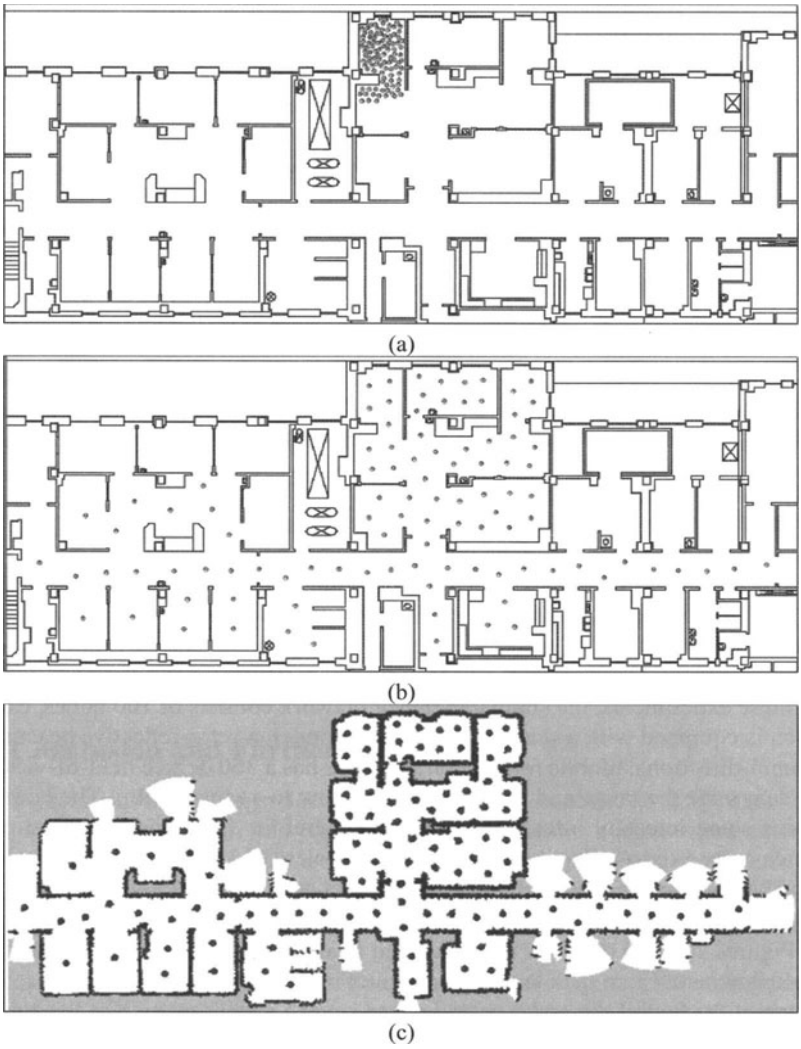


Fig. 2. A proto-typical deployment experiment for a 100-node network. (a) Initial network configuration. (b) Final configuration after 300 seconds. (c) Occupancy grid generated for the final configuration; visible space is marked in black (occupied) or white (free); unseen space is marked in gray.

4 Experiments

We have conducted a series of simulation experiments aimed at both validating and investigating the use of potential fields for the sensor network deployment problem.

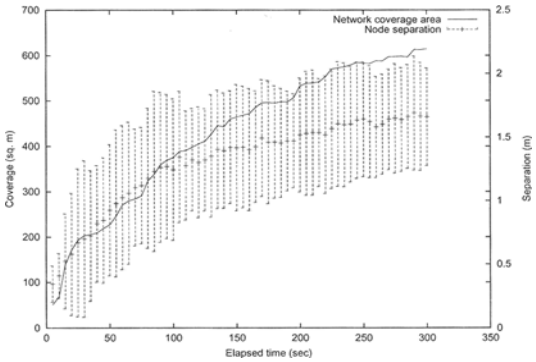


Fig. 3. Network coverage area and average node separation as function of time for a 100-node deployment experiment. The coverage and separation are plotted on different scales.

Two metrics are of particular interest: coverage (i.e., what is the area covered by the network) and time (i.e., how long does the network take to deploy).

Our experiments were conducted using the Player robot server [7] in combination with the Stage [17,6] multi-agent simulator. Stage simulates the behavior of real sensors and actuators with a high degree of fidelity, and algorithms developed using Stage can usually be transferred to real hardware with little or no modification. For these experiments, the simulated sensor network consists of 100 nodes, each of which is equipped with a scanning laser range finder, a retro-reflective beacon and an omni-directional mobile robot base. The laser has a 360 degree field-of-view and can determine the range and bearing of objects out to a range of 4m. The laser also returns some intensity information, and can therefore distinguish between nodes (which carry a retro-reflective beacon) and obstacles (which do not). The network is placed in a complex simulated environment that represents a single floor in a large hospital.

Figures 2(a) and (b) show the initial and final network configurations for a typical deployment. From their starting configuration (crammed into the single room at the top of the figure) the nodes spread out to cover a sizable portion of the environment; the coverage area in the final configuration is in excess of 500 m^2, a 10-fold improvement over the initial coverage of around 50 m^2. The temporal behavior of the network is captured Figure 3(a), which shows a plot of coverage versus deployment time. From this plot, it is apparent that the rate of coverage decreases with time, and that the total coverage was still increasing when the experiment was terminated after 300 seconds. We plan to investigate and characterize this curve more carefully in future experiments.

The deployment is quite fast. The elapsed time for this experiment is 300 seconds, in which time the nodes on the network boundary have traveled a distance of around 40 m. Since the maximum permitted velocity in this experiment is 0.5 m/s, the average velocity of the boundary nodes during deployment is just under 15% of

the theoretical maximum. If we restict ourselves to the early phase of the deployment, the average velocity is higher still: over the first 180 seconds, the nodes reach 35% of the theoretical maximum velocity.

An unexpected, but appealing, feature of the deployment is the evenness of the node spacing: the average nearest-neighbor separation in the final configuration is $1.6 \pm 0.4m$. The variance is surprisingly low given the lack explicit coordination between nodes and the structural variability of the environment. As for the separation distance, it is unclear, at this point, whether the distance is related to external phenomena, such as the scale of features in the environment, or is a function of purely internal factors, such as the relative weights on the potential fields. In an open environment, we might expect the average separation to approach 4 m, corresponding to the range limit on the laser's field-of-view; it is not obvious, however, what value we should expect in a highly structured environment such as the one used in this experiments.

The network coverage produced in this experiment is of a very high quality. Figure 2(c) shows an occupancy grid generated for the final configuration: areas that can be seen by the network are shown in black (for obstacles) or white (for open space); unseen areas are shown in gray. Note that there are no gaps or breaks in the coverage. The high quality of this coverage can be attributed to the even spacing of nodes, combined with the fact that the average node separation is about half the sensor range. This effectively creates a dense, highly redundant network.

Animations of this and other experiments can be found at:
`http://robotics.usc.edu/~ahoward/movies.html`.

5 Conclusion and Further Work

The experiments described in Section 4 are far from complete. To fully characterize the approach described in this paper, we need to perform a much more extensive series of experiments, in which we vary both external factors (such as network size, environment, and initial conditions) and internal factors (such as the weights k_o and k_n, the node mass m and viscosity coefficient ν). We are currently in the process of conducting such experiments.

The experiments described in this paper are, however, quite sufficient to demonstrate that a potential field approach *can* be used to deploy mobile sensor networks. The approach has the advantage that it does not require centralized control, localization or communication, and will therefore scale to very large networks. Furthermore, as demonstrated in Section 3.2, this approach has provable convergence characteristics.

There are a number of directions in which we would like to expand this research. We are interested, for example, in how one might apply this approach to coverage problems in which *line-of-sight connectivity* is important [9]. For these problems, would like the deployment to proceed such that the network is fully connected at all times by line-of-sight relationships. In principle, this requires form of communication between nodes; in practice, however, it may be the case that connectivity, like area coverage, can emerge from a combination of purely local rules.

References

1. R. C. Arkin. Motor schema based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112, 1989.
2. T. Balch and M. Hybinette. Behavior-based coordination of large-scale robot formations. In *Proceedings of the Fourth International Conference on Multiagent Systems (ICMAS '00)*, pages 363–364, Boston, MA, USA, July 2000.
3. W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. of IEEE International Conferenceon Robotics and Automation (ICRA)*, volume 1, pages 476–81, 2000.
4. G. Dedeoglu and G. S. Sukhatme. Landmark-based matching algorithms for cooperative mapping by autonomous robots. In L. E. Parker, G. W. Bekey, and J. Barhen, editors, *Distributed Autonomous Robotics Systems*, volume 4, pages 251–260. Springer, 2000.
5. D. W. Gage. Command control for many-robot systems. In *AUVS-92, the Nineteenth Annual AUVS Technical Symposium*, pages 22–24, Hunstville Alabama, USA, June 1992. Reprinted in Unmanned Systems Magazine, Fall 1992, Volume 10, Number 4, pp 28-34.
6. B. Gerkey, R. Vaughan, and A. Howard. Player/Stage homepage. http://robotics.usc.edu/player/, September 2001.
7. B. P. Gerkey, R. T. Vaughan, K. Støy, A. Howard, G. S. Sukhatme, and M. J. Matarić. Most valuable player: A robot device server for distributed control. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS01)*, pages 1226–1231, Wailea, Hawaii, Oct. 2001.
8. H. Goldstein. *Classical mechanics*. Addison-Wesley, 1980.
9. A. Howard, M. J. Matarić, and G. S. Sukhatme. Localization for mobile robot teams: A maximum likelihood approach. Technical Report IRIS-01-407, Institute for Robotics and Intelligent Systems Technical Report, University of Sourthern California, 2001.
10. O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.
11. M. López-Sánchez, F. Esteva, R. L. de Mántaras, C. Sierra, and J. Amat. Map generation by cooperative low-cost robots in structured unknown environments. *Autonomous Robots*, 5(1):53–61, 1998.
12. J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, August 1987.
13. F. E. Scheider, D. Wildermuth, and H.-L. Wolf. Motion coordination in formations of multiple mobile robots using a potential field approach. In L. E. Parker, G. W. Bekey, and J. Barhen, editors, *Distributed Autonomous Robotics Systems*, volume 4, pages 305–314. Springer, 2000.
14. R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proc. of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 852–858, 2000.
15. S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2000)*, volume 1, pages 321–328, 2000.
16. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence Journal*, 128(1–2):99–141, 2001.
17. R. T. Vaughan. Stage: a multiple robot simulator. Technical Report IRIS-00-393, Institute for Robotics and Intelligent Systems, University of Southern California, 2000.