Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Distributed Coverage Control in Sensor Network Environments with Polygonal Obstacles ⋆

Minyi Zhong and Christos G. Cassandras

Dept. of Manufacturing Engineering
and Center for Information and Systems Engineering
Boston University, Brookline, MA 02446
mzhong@bu.edu,cgc@bu.edu

Abstract: We consider the problem of distributed coverage control for mobile sensor networks operating in environments cluttered with polygonal obstacles which interfere with both the navigation and sensing by the nodes. A gradient-based motion control scheme is developed to maximize the joint detection probability of random events in such mission spaces, taking into account the discontinuities that are introduced by obstacles in the sensing probability models. The optimization scheme requires only local information at each node. We also propose a modified objective function which allows a more balanced coverage when necessary. An interactive simulation environment has been developed through which we illustrate the adaptive and distributed properties of the coverage algorithm in a variety of mission spaces with obstacles.

Keywords: sensor networks; distributed control; mission planning and decision making

1. INTRODUCTION

The performance of a sensor network depends on how its nodes are located within a "mission space", which gives rise to the fundamental problem of *coverage control* or *active sensing*; see Meguerdichian et al. (2001), Cortes et al. (2004) and Mihaylova et al. (2002). In particular, nodes must be deployed so as to maximize the information extracted from the mission space while maintaining acceptable levels of communication and energy consumption. The *static* version of this problem involves positioning sensors without any further mobility and optimal locations can be determined by an off-line scheme which is akin to the widely studied facility location optimization problem. The *dynamic* version allows the coordinated movement of sensors, which may adapt to changing conditions in the mission space, typically deploying them into geographical areas with the highest information density; see, for example, Cortes et al. (2004), Meguerdichian et al. (2001)and Zou and Chakrabarty (2003).

Solutions of the coverage control problem based on partitioning the mission space overlook the fact that the overall sensing performance may be improved by sharing the observations made by multiple sensors. In addition, many approaches assume uniform sensing quality and an unlimited sensing range, which is not the case for most sensing devices used in practice. A number of solution techniques are also based on a centralized controller, which is inconsistent with the distributed communication and computation structure of sensor networks. Moreover, the combinatorial complexity of the problem constrains the application of such schemes to limited-size networks. Finally, another issue that appears to be neglected is the movement of sensors, which not only impacts sensing performance but it also influences other quality-of-service aspects in a sensor network, especially those related to wireless communication: because of the limited on-board power and computational capacity, a sensor network is not only required to sense but also to collect and transmit data as well. For this reason, both sensing quality and communication performance need to be jointly considered when controlling the deployment of sensors. In order to address all these issues, a distributed coverage control algorithm was developed in Li and Cassandras (2005) and Cassandras and Li (2005) which uses a distance-dependent probabilistic sensing model and incorporates communication constraints.

The environment assumed in Cassandras and Li (2005) does not take into account boundaries in the mission space and allows no obstacles. Introducing obstacles has two ramifications. First, the sensor nodes can obviously not be located or navigate in the space occupied by an obstacle. Second, the obstacles interfere with the sensing process. Unfortunately, the second difficulty is a serious one. The reason is that, in the presence of obstacles, the detection probability of an event at some point $x \in \mathbb{R}^2$ from a sensor node at $s \in \mathbb{R}^2$ is no longer a continuous function of s.

In this paper, we develop a distributed gradient-based coverage control scheme for a finite mission space with polygonal obstacles where we aim to maximize the joint probability of detecting events in the mission space with a given event density function. This objective is in contrast to the classic Art Gallery problem setting which

⋆ Supported in part by the National Science Foundation under Grants DMI-0330171 and EFRI-0735974, by AFOSR under grants FA9550-07-1-0213 and FA9550-07-1-0361, and by DOE under grant DE-FG52-06NA27490.

focuses on ensuring that every point in the mission space is "visible" by at least one guard with unlimited range; see Urrutia (2000), O'Rourke (1987), Ganguli et al. (2006a), Ganguli et al. (2006b). In addition, we allow nodes to have different sensing characteristics and their range to be generally limited. Our approach requires gradients of the objective function evaluated at each node using only local information along the lines of Cassandras and Li (2005). The main contribution of the paper is this gradient derivation, rigorously incorporating the effect of discontinuities in the detection probability functions mentioned above, which is caused by the obstacles. In addition, we address the problem of imbalances in the mission space coverage; in particular, maximizing the joint detection probability can result in a deployment configuration which achieves extremely high event detection performance in certain regions while leaving others virtually uncovered. Through an appropriate transformation of our objective function, we show that we can in fact achieve a balanced performance to any degree desired.

2. PROBLEM FORMULATION

We model the *mission space* $\Omega \subset \mathbb{R}^2$ as a non-self-intersecting polygon, i.e., a polygon such that any two non-consecutive edges do not intersect. An event density function $R(x) : \Omega \to \mathbb{R}$ captures the frequency of random event occurrences at some point $x \in \Omega$. $R(x)$ satisfies $R(x) \geq 0$ for all $x \in \Omega$ and $\int_\Omega R(x)\,dx < \infty$. We assume that when an event takes place, it will emit some event signal which may be observed by some sensor nodes.

The mission space may contain *obstacles* which can interfere with the movement of sensor nodes and the propagation of event signals. We model the boundaries of these obstacles as m non-self-intersecting polygons properly contained in Ω and denote them by $P_j \subset \Omega$, $j = 1, ..., m$. Each P_j divides Ω into two disjoint regions, P_j's exterior and interior. The interior of P_j, denoted by $\mathring{P}_j$, is infeasible for the sensor nodes to navigate in. Thus, the overall feasible (or navigable) subspace of Ω is $F = \Omega \backslash \left(\mathring{P}_1 \cup \cdots \cup \mathring{P}_m\right)$.

We will assume that $R(x) = 0$ for $x \notin F$ either because there is "nothing interesting" happening outside of F or because the sensor nodes ignore all points outside of F. In our coverage control problem, the number of sensor nodes deployed into Ω may be fixed, N, or time-varying, $N(t)$, if nodes are allowed to "die" or new nodes are occasionally introduced. When the number of nodes is N their location is denoted by a $2N$-dimensional vector $\mathbf{s} = (s_1, ..., s_N)$ with $s_i \in F$, $i = 1, ..., N$.

Next, we discuss the sensing model used. If there is no obstacle and Ω is convex (thus, there is a clear line of sight between any two points in Ω), the probability that sensor node i detects an event occurring at $x \in F$ is denoted by $p_i(x, s_i)$. The received signal strength generally decays with $\|x - s_i\|$, the Euclidean distance between the source and the sensor. We represent this degradation by a monotonically decreasing differentiable function $p_i(x, s_i)$. An example of such a function is

$$p_i(x, s_i) = p_{0i} e^{-\lambda_i \|x - s_i\|} \tag{1}$$

Notice that (1) is an omnidirectional model and can be viewed as a function of $\|x - s_i\|$.

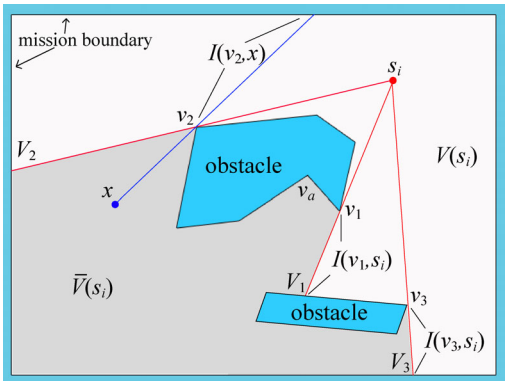


Fig. 1. Mission space with two polygonal obstacles

Similar to the geometric terms in Ganguli et al. (2006a), we use the prefix ∂ to denote the boundary of a topological set, so that $\partial F = \partial\Omega \cup P_1 \cup \cdots \cup P_m$ and let T be the set of vertices of ∂F. A vertex v in T is a *reflex vertex* if the two edges incident at v form an angle strictly greater than π in the interior of F. In Fig. 1, for example, all vertices of the two obstacles, except v_a, are reflex vertices.

A point $x \in F$ is *visible* from $y \in F$ if the line segment defined by x and y is contained in F, i.e., $(\lambda x + (1 - \lambda)\, y) \in F$, for all $\lambda \in [0, 1]$. The *visibility polygon* $V(x) \subset F$ from a point $x \in F$ is the set of points in F visible from x. Let $\overline{V}(x) = F \backslash V(x)$ denote the invisibility polygon with respect to x. For example, in Fig. 1, the white area is the visibility polygon $V(s_i)$ of s_i and the grey area is $\overline{V}(s_i)$.

Let v be a reflex vertex and let $x \in F$ be a point visible from v. We define a set of points $I(v, x)$ as follows:

$$I(v, x) = \{q \in V(v) : q = \lambda v + (1 - \lambda)\, x, \quad \lambda > 1\}$$

To give a graphical interpretation of $I(v, x)$, consider a ray extending from v in the direction of $v - x$. This ray travels inside F until it hits ∂F at an *impact point*. The line segment between v and the impact point is $I(v, x)$ (see also Fig. 1).

Sensing model. If $x \notin V(s_i)$, then the ability of node i to detect an event occurring at x will be reduced, possibly to zero. In general, an event signal might still be able to travel through obstacles, but its intensity will be attenuated more so than in open space depending on factors such as the size of the obstacle and the number of obstacles in the line of sight. Thus, the modified probability that sensor node i detects an event occurring at x is

$$\widehat{p}_i(x, s_i) = \begin{cases} p_i(x, s_i) & \text{if } x \in V(s_i) \\ \widetilde{p}_i(x, s_i) & \text{if } x \in \overline{V}(s_i) \end{cases} \tag{2}$$

where $\widetilde{p}_i(x, s_i) \leq p_i(x, s_i)$. For a totally "opaque" obstacle, we have $\widetilde{p}_i(x, s_i) = 0$.

Note that for a fixed point x, $\widehat{p}_i(x, s_i)$ is not a continuous function of s_i. There is a jump whenever s_i crosses a line $I(v, x)$, where v is a reflex vertex visible from x. For example, in Fig. 1 assuming "opaque" obstacles, s_i cannot see x, so $\widehat{p}_i(x, s_i) = 0$. But if s_i moves left and eventually reaches $I(v_2, x)$, then x becomes visible with respect to s_i and $\widehat{p}_i(x, s_i)$ jumps to a nonzero value $p_i(x, s_i)$.

Coverage optimization problem. Since multiple sensor nodes are deployed to cover the mission space, the joint probability that an event occurring at x is detected,

denoted by $P(x,\mathbf{s})$, is given by

$$P(x,\mathbf{s}) = 1 - \prod_{i=1}^{N} [1 - \widehat{p}_i(x, s_i)] \tag{3}$$

and the optimization problem of interest is

$$\max_{\mathbf{s}} \int_{\Omega} R(x) P(x,\mathbf{s}) dx \tag{4}$$
$$\text{s.t. } s_i \in F,\ i = 1, ..., N$$

In (4) we use the locations of the sensor nodes as decision variables to maximize the frequency of event detection in Ω. Our goal is to develop a distributed algorithm to solve this optimization problem with each node performing a limited number of computations based on local information only. This eliminates the communication burden of transferring information to and from a central controller and the vulnerability of the whole system which would be entirely dependent on this controller.

Since we have assumed $R(x) = 0$ for $x \notin F$, we rewrite the objective function in (4) as

$$H(\mathbf{s}) = \int_{F} R(x) P(x,\mathbf{s}) dx \tag{5}$$

3. CONVEX MISSION SPACE WITH NO OBSTACLES

If the mission space is convex and contains no obstacles, we have $F = \Omega$ and $x \in V(s_i)$ for any $x, s_i \in F$. Thus, $\widehat{p}_i(x, s_i)$ in (3) can be replaced by $p_i(x, s_i)$ and $P(x,\mathbf{s})$ is a continuous function of s_i if $p_i(x, s_i)$ is continuous. Therefore,

$$\frac{\partial H(\mathbf{s})}{\partial s_i} = \int_{\Omega} R(x) \frac{\partial P(x,\mathbf{s})}{\partial s_i} dx \tag{6}$$

The following is obtained in Cassandras and Li (2005):

$$\frac{\partial H(\mathbf{s})}{\partial s_i} = \int_{\Omega_i} R(x) \prod_{k \in \mathcal{B}_i} [1 - p_k(x, s_k)] \frac{dp_i(x, s_i)}{dd_i(x)} \frac{s_i - x}{d_i(x)} dx \tag{7}$$

where $d_i(x) \equiv \|x - s_i\|$. If δ denotes the sensing radius of node i, then the node's region of coverage is represented by $\Omega_i = \{x : d_i(x) \leq \delta\}$. $\mathcal{B}_i$ is a set of neighbor nodes with respect to i:

$$\mathcal{B}_i = \{k : \|s_i - s_k\| < 2\delta,\ k = 1, \ldots, N,\ k \neq i\}$$

In (7), all information is locally available to node i and the gradient can be used to determine the next waypoint on the ith mobile sensor's trajectory through

$$s_i^{k+1} = s_i^k + \eta_k \frac{\partial H(\mathbf{s})}{\partial s_i^k} \tag{8}$$

where k is an iteration index, and the step size sequence $\{\eta_k\}$ is selected according to standard rules (e.g., see Bertsekas (1995)) when the convergence of motion trajectories must be guaranteed.

4. A MISSION SPACE WITH OBSTACLES

Node i's position in $\mathbb{R}^2$ is represented by $s_i = (s_{x_i}, s_{y_i})$. For simplicity, we will drop the subscript i in s_i, s_{x_i}, and s_{y_i} and focus on a typical node whose position is denoted by s. In addition, in the following discussion, we will assume that obstacles are fully opaque, i.e. $\widetilde{p}_i(x, s_i) = 0$, for all i, x and s_i, to simplify the derivation.

As already pointed out, for a given $x \in F$, $\widehat{p}(x, s)$ is not a continuous function of s and so neither is $P(x,\mathbf{s})$ in (3). Thus, we cannot interchange the order of differentiation and integration as in (6). A natural way to proceed is to separate the integration area F in (5). In particular, we will separate F into $V(s)$ and $\overline{V}(s)$, and define:

$$P_1(x,\mathbf{s}) = 1 - \prod_{k=1, k\neq i}^{N} [1 - \widehat{p}_k(x, s_k)] (1 - p_i(x, s_i))$$

$$P_2(x,\mathbf{s}) = 1 - \prod_{k=1, k\neq i}^{N} [1 - \widehat{p}_k(x, s_k)] (1 - \widetilde{p}_i(x, s_i))$$

so that

$$H(\mathbf{s}) = \int_{V(s)} R(x) P_1(x,\mathbf{s}) dx + \int_{\overline{V}(s)} R(x) P_2(x,\mathbf{s}) dx \tag{9}$$

Gradient derivation. We will apply an extension of the Leibnitz rule (Flanders (1973)) to evaluate $\partial H(\mathbf{s})/\partial s_x$, since in (9), both the integrand and the domain of integration are functions of s. For the first term in the right hand side of (9), we have

$$\frac{d}{ds_x} \int_{V(s)} R(x) P_1(x,\mathbf{s}) dx = \int_{V(s)} R(x) \frac{\partial P_1(x,\mathbf{s})}{\partial s_x} dx + \int_{\partial V(s)} R(x) P_1(x,\mathbf{s}) (u_x dx_y - u_y dx_x) \tag{10}$$

where (u_x, u_y) denotes the "velocity" vector at a boundary point $x = (x_x, x_y)$ of $V(s)$. The first term in the right hand side of (10) does not involve any variation of the integration domain and can be evaluated similarly to (7). To evaluate the second term, we need to introduce some more definitions related to the geometry of Fig. 1.

A reflex vertex v is an *anchor* of s if it is visible from s and $I(v, s) = \{q \in V(v) \,|q = \lambda v + (1 - \lambda) s, \lambda > 1\}$ is not empty, see also Ganguli et al. (2006a). Denote the anchors of s by v_j, $j = 1, ..., Q(s)$, where $Q(s)$ is the number of anchors of s. An *impact point* of v_j, denoted by V_j, is the intersection of $I(v_j, s)$ and ∂F. Let $D_j = \|s - v_j\|$ and $d_j = \|V_j - v_j\|$. Define θ_j to be the angle formed by $s - v_j$ and the x-axis which satisfies $\theta_j \in [0, \frac{\pi}{2}]$.

For simplicity, we will abbreviate $I(v_j, s)$ by I_j in what follows. Observing that I_j is always on the boundary of $V(s)$ and $\overline{V}(s)$, we define n_j to be the unit normal vector on I_j which points to the interior of $V(s)$. Assuming that s is not on a reflex vertex, a polygonal inflection, or a bitangent (see definitions of these terms in LaValle and Hinrichsen (2001)), on the boundary of the visibility polygon of s, only the points on I_j, $j \in 1, ..., Q(s)$, will have a nonzero "velocity" when s is perturbed; see Fig. 2 for an illustration. Thus, the second term in the right hand side of (10) becomes

$$\sum_{j=1}^{Q(s)} \int_{I_j} R(x) P_1(x,\mathbf{s}) (u_x dx_y - u_y dx_x) \tag{11}$$

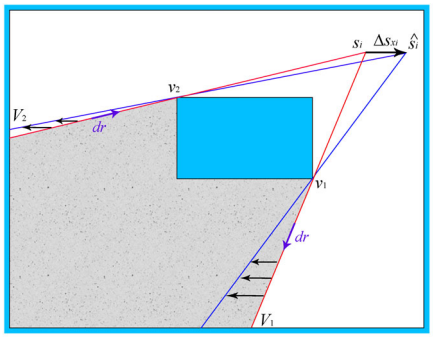


Fig. 2. Only points on I_j will be affected as s_i is perturbed

Since the original contour integral on $\partial V(s)$ is counterclockwise, the orientation of the line integral above associated with I_j can be determined accordingly. Without loss of generality, we further assume that a point on I_j will only move along the x-axis under a small perturbation in s_x. Thus we have $u_y = 0$ and $u_x = \frac{-\|v_j - x\|}{D_j}$, for a point $x \in I_j$. After skipping some calculations (which may be found in Zhong and Cassandras (2007)), (11) can be rewritten as

$$\sum_{j=1}^{Q(s)} \int_{I_j} R(x) P_1(x, \mathbf{s}) u_x dx_y = \sum_{j=1}^{Q(s)} sgn(n_{jx}) \frac{\sin\theta_j}{D_j} \int_0^{d_j} R(\rho_j(r)) P_1(\rho_j(r), \mathbf{s}) r dr$$

where

$$\rho_j(r) = (V_j - v_j)\frac{r}{d_j} + v_j \tag{12}$$

After evaluating the second term in (9) in a similar way and combining the results, we obtain

$$\frac{\partial H(\mathbf{s})}{\partial s_x} = \int_{V(s)} R(x) \frac{\partial P_1(x, \mathbf{s})}{\partial s_x} dx + \sum_{j=1}^{Q(s)} sgn(n_{jx}) \frac{\sin\theta_j}{D_j} \int_0^{d_j} R(\rho_j(r)) \Phi_{i,j}^N(r) p_i(\rho_j(r), s_i) r dr \tag{13}$$

where $\rho_j(r)$ was defined in (12) and we set

$$\Phi_{i,j}^N(r) = \prod_{k=1, k\neq i}^{N} [1 - \widehat{p}_k(\rho_j(r), s_k)]$$

Proceeding in the exact same manner, we obtain

$$\frac{\partial H(\mathbf{s})}{\partial s_y} = \int_{V(s)} R(x) \frac{\partial P_1(x, \mathbf{s})}{\partial s_y} dx + \sum_{j=1}^{Q(s)} sgn(n_{jy}) \frac{\cos\theta_j}{D_j} \int_0^{d_j} R(\rho_j(r)) \Phi_{i,j}^N(r) p_i(\rho_j(r), s_i) r dr \tag{14}$$

A more elaborate derivation of (13)-(14) using purely geometric arguments (instead of the Leibnitz rule) is also possible and can be found in Zhong and Cassandras (2007).

Distributed algorithm. The derivatives in (13)-(14) can now be used in the motion control scheme (8) with the inclusion of a standard projection mechanism so that if a node points immediately into an edge of an obstacle or of the mission space boundary, we project $\partial H(\mathbf{s})/\partial s_i$ onto that edge, thus forcing the node to "slide" along the associated motion constraint. As in the case of no obstacles, $\partial H(\mathbf{s})/\partial s_i$ can be evaluated using only information locally available to node i. Let Ω_i, $\mathcal{B}_i$ and δ be defined as in Section 3. We can then rewrite (13) as

$$\frac{\partial H(\mathbf{s})}{\partial s_{x_i}} = \int_{V(s)\cap\Omega_i} R(x) \prod_{k\in\mathcal{B}_i} [1 - \widehat{p}_k(x, s_k)] \frac{dp_i(x, s_i)}{dd_i(x)} \frac{(s_i - x)_x}{d_i(x)} dx + \sum_{j\in\Gamma_i} sgn(n_{jx}) \frac{\sin\theta_j}{D_j} \int_0^{z_j} R(\rho_j(r)) \widehat{\Phi}_{i,j}^N(r) p_i(\rho_j(r), s_i) r dr \tag{15}$$

where $\Gamma_i = \{j : D_j < \delta,\ j = 1, \ldots, Q(s_i)\}$; $z_j = \min(d_j, \delta - D_j)$ and we define

$$\widehat{\Phi}_{i,j}^N(r) = \prod_{k\in\mathcal{B}_i} [1 - \widehat{p}_k(\rho_j(r), s_k)]$$

The computation of the integrals in (15) is quite involved. Thus, we resort to the same mission space discretization as in in Cassandras and Li (2005), which reduces the evaluation to a worst-case computation of order $O(N_{\mathcal{B}} W^2 + |\Gamma_i| N_{\mathcal{B}} U)$ where $N_{\mathcal{B}} - 1$ is the number of neighbors of i, while W and U are controllable resolution parameters to discretize the surface integration and line integration, respectively.

Recall that in the derivation of (13)-(14) we assumed that the controllable node location s_i does not coincide with a reflex vertex, a polygonal inflection, or a bitangent, at which points $H(\mathbf{s})$ is generally not differentiable. To take these points into account, one can modify the standard gradient-based algorithm in (8) resorting to nonsmooth optimization methods. Since our objective function is nonconcave, subgradient algorithms with diminishing stepsize (e.g., Shor (1985)) do not guarantee convergence. Instead, we may use *bundle methods* (Kiwiel (1985)) which aggregate the subgradient information in the past iterations and find a descent direction at each step (in contrast, the subgradient method does not always follow a descent direction). For a locally Lipschitz problem with constraints, there are results for these methods guaranteeing convergence to stationary points. However, efficient implementation of bundle methods is fairly complicated.

Whereas such nonsmooth optimization methods are invaluable in optimization problems where the points of nondifferentiability are hard to determine in advance, the geometric properties of the coverage problem greatly simplify this concern. In particular, since we assume the topology of the mission space is known, it is straightforward to determine all reflex vertices, polygonal inflection points, or bitangents. Therefore, upon execution of the gradient-based algorithm (8), it is easy to detect when s_i^k is in the vicinity of such a point at any iteration k and simply adjust the corresponding step size η_k so as to bypass this point. Moreover, in practice numerical errors normally have the same effect. The only case where this approach is not as simple is when a local optimum in fact coincides with a reflex vertex, polygonal inflection, or bitangent (which may not be that uncommon). In this case, this approach will

lead to oscillatory behavior of a node around such a point. If convergence must be guaranteed, then a local nonsmooth optimization method may be used.

We should also point out that convergence of (8) often requires a step size sequence such that $\eta_k \to 0$ as $k \to \infty$. This, however, is not always desirable if the mission space is not stationary and one needs to track changes in the density function or the addition/removal of sensor nodes. One may trade off oscillatory behavior around local optima for the benefit of reacting to such changes. It is also obvious that (8) may lead only to local optima since our objective function is generally not concave.

5. MODIFIED COVERAGE OBJECTIVE FOR BALANCED DETECTION

As defined in (4), the coverage objective function aims at maximizing the joint event detection probability without considering the issue of maintaining a balance between a region which may be extremely well covered and others which may not. As shown by the examples in the next section, an optimal coverage solution may lead to a part of the mission space having a detection probability near 1, while other parts are covered with a small detection probability. In order to address this issue, we introduce a modification to the objective function as follows:

$$H_M(\mathbf{s}) = \int_\Omega R(x)\, M(P(x,\mathbf{s}))\, dx \tag{16}$$

where $M(P) : [0,1] \to \mathbb{R}$ is a (possibly piecewise) differentiable concave non-decreasing function of the joint detection probability P. Clearly, $M(P)$ may be selected so that the same amount of marginal gain in $P(x,\mathbf{s})$ is assigned a higher reward at a lower value of P. Letting

$$\Psi = M[1-\Phi_{i,j}^N(r)(1-p_i(\rho_j(r)))] - M[1-\Phi_{i,j}^N(r)]$$

the gradient of $H_M(\mathbf{s})$ is given by:

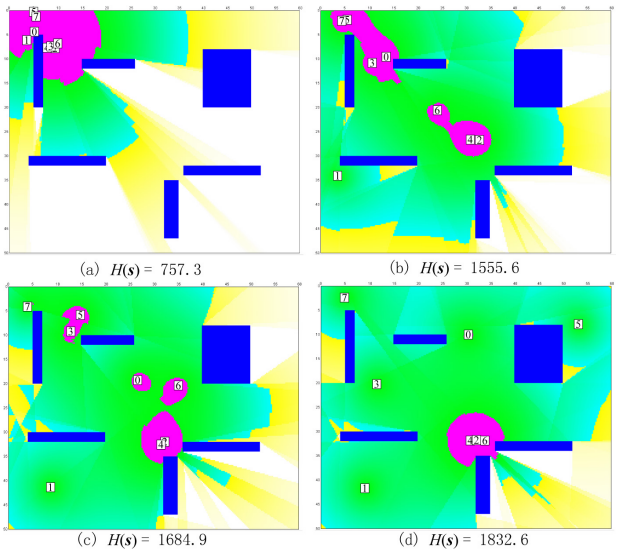
$$\frac{\partial H_M(\mathbf{s})}{\partial s_x} = \sum_{j=1}^{Q(s)} sgn(n_{jx}) \frac{\sin\theta_j}{D_j} \int_0^{d_j} R(\rho_j(r))\, \Psi r dr + \int_{V(s)} R(x)\, M'(P_1(x,\mathbf{s})) \frac{\partial P_1(x,\mathbf{s})}{\partial s_x} dx$$

$$\frac{\partial H_M(\mathbf{s})}{\partial s_y} = \sum_{j=1}^{Q(s)} sgn(n_{jy}) \frac{\cos\theta_j}{D_j} \int_0^{d_j} R(\rho_j(r))\, \Psi r dr + \int_{V(s)} R(x)\, M'(P_1(x,\mathbf{s})) \frac{\partial P_1(x,\mathbf{s})}{\partial s_y} dx$$

Obviously, using $\partial H_M(\mathbf{s})/\partial s_i$ instead of $\partial H(\mathbf{s})/\partial s_i$ in (8) will lead to different local optima. One can use $H(\mathbf{s})$ to guide the nodes to such a point and assess whether the coverage is sufficiently balanced. If not, switching to $H_M(\mathbf{s})$ induces the nodes to move to a different stationary configuration at which, interestingly, $H(\mathbf{s})$ may in fact be higher than before; see examples in the next section.

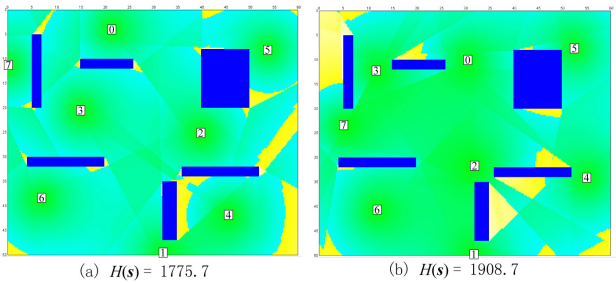
6. SIMULATION RESULTS

An interactive Java-based simulation environment has been developed and may be found (along with instructions) at *http://codescolor.bu.edu/coverage.*



(a) $H(\mathbf{s}) = 757.3$ (b) $H(\mathbf{s}) = 1555.6$ (c) $H(\mathbf{s}) = 1684.9$ (d) $H(\mathbf{s}) = 1832.6$

Fig. 3. Coverage control using the sensor model (1), $p_{0i} = 1$, $\lambda_i = 0.08$, 8 nodes deployed.



(a) $H(\mathbf{s}) = 1775.7$ (b) $H(\mathbf{s}) = 1908.7$

Fig. 4. (a) Modified coverage objective used. (b) Original objective used after reaching (a).

Fig. 3 shows snapshots of an optimal coverage deployment trajectory generated under (8) with the gradient evaluated through (13)-(14). In this example, there are eight sensor nodes in a bounded mission space with uniform event density. The dark polygons represent obstacles which are totally opaque to the sensors. The numbered small white rectangles indicate the locations of the nodes, all initially starting at the upper-left corner. The mission space is color-coded from darker to lighter as the detection probability decreases. In Fig. 3(d), note that the equilibrium configuration includes three nodes (2, 4 and 6) which are very close to each other; local optimality prevents them from navigating into the lower right corner, which is only partially covered. If the modified objective function (16) is used with all other settings unchanged, the nodes spread more evenly as shown in Fig. 4(a). After an equilibrium is reached using (16), we change the objective function to the original one and obtain the coverage of Fig. 4(b) where the $H(\mathbf{s})$ value achieved is higher than that in Fig. 3(d). This suggests that alternating between the two objective criteria provides opportunities to escape local optima which provide imbalanced coverage and possibly improve the original objective as well.

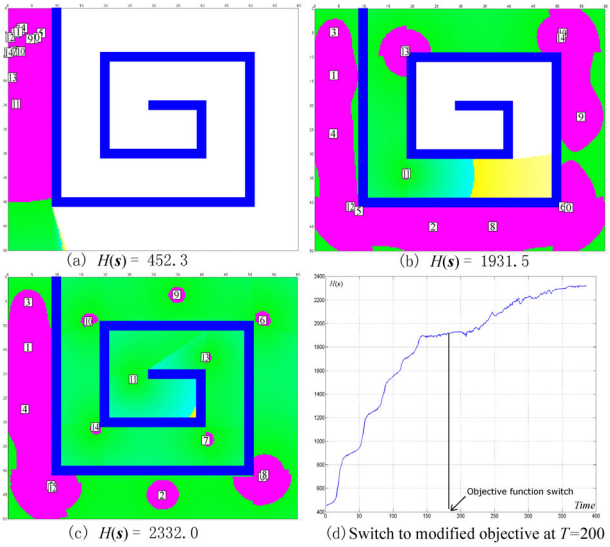


Fig. 5. Coverage control of maze with $\lambda_i = 0.05$, 15 nodes

Fig. 5 shows snapshots of an optimal coverage deployment trajectory in a maze-like environment, which is harder to cover. At equilibrium, the corner points are attractive node locations because a node at such a point has the benefit of seeing two "corridors" at the same time. Again, the 15 nodes do not cover the whole mission space evenly, with the majority placed so that some extremely well covered areas are created. After reaching equilibrium as shown in Fig. 5(b), we switch to the modified objective function and the final result is provided in Fig. 5(c).

We should point out that in these simulated coverage missions, we observe nodes oscillating around local optima, consistent with the discussion of the distributed algorithm in Section 4. We can easily eliminate such oscillations by using a decreasing step size sequence. However, in practice, maintaining a large step size is desirable for faster deployment, thus giving rise to a tradeoff between deployment speed and oscillatory behavior. If, for example, the nodes are deployed into the mission space from a single point, a large step size is critical for preventing the nodes from being trapped in a configuration where they remain close to each other. Using a large step size at the expense of a convergence guarantee in effect encourages some "random exploration" in order to bypass local optima near the initial state. The simulation results shown here are in fact making use of such a scheme.

7. CONCLUSIONS AND FUTURE WORK

We have presented a gradient-based distributed coverage control scheme for mobile sensor networks operating in environments cluttered with polygonal obstacles. The main technical challenge has been the evaluation of the objective function derivatives with respect to node positions in the presence of discontinuities in the sensing probability models due to the obstacles. We also propose a modified version of our approach when the objective includes balancing the mission space coverage by skewing the objective function so as to assign higher rewards for coverage improvement in low detection probability areas. The algorithm provably converges to local optima under well-known conditions, although in practice a certain amount of oscillatory behavior around such points is desirable in order to trigger response to changes in the mission space. Methods that can guarantee attaining a global optimum are highly desirable but remain elusive except for special cases.

Future work aims at incorporating communication costs into the objective function and allowing for sensor nodes with limited fields of view or application-specific sensing models. More importantly, the algorithm presented in this paper is synchronized in the sense that when a node updates its heading in its motion control, it requires the most up-to-date information from its neighbors. This results in a high volume of local communication. An *asynchronous* version of the algorithm aiming at reducing communication frequency is the subject of our current work.

REFERENCES

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

C. G. Cassandras and W. Li. Sensor networks and cooperative control. *European Journal of Control*, 11 (4-5):436–463, 2005.

J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Trans. on Robotics and Automation*, 20(2), 2004.

H. Flanders. Differentiation under the integral sign. *The American Mathematical Monthly*, 80(6):615–627, 1973.

A. Ganguli, J. Cortes, and F. Bullo. Maximizing visibility in nonconvex polygons: nonsmooth analysis and gradient algorithm design. *SIAM Journal on Control and Optimization*, 45, 2006a.

A. Ganguli, J. Cortes, and F. Bullo. Distributed deployment of asynchronous guards in art galleries. In *Proc. of the American Control Conf.*, 2006b.

K. C. Kiwiel. *Methods of descent for nondifferentiable optimization*. Springer-Verlag, Berlin, 1985.

S. M. LaValle and J. E. Hinrichsen. Visibility-Based Pursuit-Evasion: the case of curved environments. *IEEE Trans. on robotics and automation*, 17(2), April 2001.

W. Li and C. G. Cassandras. Coverage control of sensor networks. In *Proc. of 44th IEEE Conf. on Decision and Control*, 2005.

S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proc. of IEEE INFOCOM*, 2001.

L. Mihaylova, T. Lefebvre, H. Bruyninckx, and K. Gadeyne. Active sensing for robotics-a survey. In *Proc. of the 5th Intl. Conf. on Numerical Methods and Applications*, Borovets, Bulgaria, 2002.

J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, 1987.

N. Z. Shor. *Minimization methods for non-differentiable functions*. Springer-Verlag, Berlin, 1985.

J. Urrutia. Art Gallery and Illumination Problems. In J. R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*. North-Holland, 2000.

M. Zhong and C. G. Cassandras. Converage control in environments with polygonal obstacles. Technical report, Boston University, 2007. http://codescolor.bu.edu/coverage.

Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In *Proc. of IEEE INFOCOM*, pages 1293–1303, 2003.