Workshop on Robot Motion Planning:
Online, Reactive, and in Real-time
2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems, IROS 2012
Vilamoura, Algarve, Portugal, October 7-12, 2012

Online, Adaptive, and Distributed Multi-Robot Motion Planning for Collaborative Patrolling of Sparse Sensor Networks

Theofanis P. Lambrou and Christos G. Panayiotou

Abstract— **This paper presents an online, adaptive, distributive and collaborative path planning method for a team of autonomous mobile sensors that enables them to navigate through a sparse network of stationary sensors to search for events and improve the spatio-temporal coverage of the sensor field. The mobile sensor nodes have limited communication and sensing ranges and collaborate to *autonomously* plan their trajectories, adapt to the local region they monitor and enhance the area coverage over time under constrains like obstacles, collisions and limited communication. In this context, this paper addresses the trade off between area coverage and mobiles' travelled distance and proposes an adaptive speed model to minimize the distance the mobiles travelled and hence the energy needed for mobility. Finally, simulation results indicate the effectiveness of the proposed approach over a centralized partitioning approach under mobile sensors failures.**

I. INTRODUCTION

This article investigates the path planning problem for improving the coverage and detection performance of mixed WSNs consisting of both static and mobile nodes. With recent advances in distributed robotics and low power embedded systems, such mixed WSNs are becoming attractive as covering complectly a large region of interest with static sensors requires excessively dense deployments which implies prohibitive cost. However, controlling the motion of mobile sensor-robots in such distributed environments is often complicated by factors as resource constraints on sensing, motion, communication and computation capabilities, uncertain nature of the environment (e.g. obstacles, hazards, node failures) and distributed-asynchronous information sharing.

Such mixed WSNs are expected to find potential applications in environmental monitoring (e.g. water bodies monitoring) as well as search and surveillance operations. Search and surveillance is a problem that has attracted significant attention over the past years, however, there is significant focus on how to allocate search effort across the environment instead of finding the best search path to follow [1], [2]. Recently, wireless sensor networks (WSNs) have been proposed to address the area monitoring or surveillance problem with either stationary nodes [3], [4], mobile [5], [6], [7], [8], [9] or both types of nodes [10], [11], [12], [13]. Mixed/Mobile WSNs is a new area of research and methods proposed usually considered random mobility models [5], [11], [14] or they do not even consider the actual path that mobile nodes should follow [6],[15] (e.g. solve the redeployment problem). Moreover, other methods proposed for finding the worst-case coverage path [16] do not consider the complete coverage-search problem and provide only a single path between two given points in a centralized and static manner (do not consider changes in the field) and hence do not support multiple mobile nodes.

In [12], an architecture is developed that enables the collaboration of mobile and stationary sensor nodes in WSNs. Mobile sensors plan their trajectories to sample the least covered areas by the stationary sensor nodes. The framework developed is easily scalable to large numbers of mobile sensor nodes and for different WSNs deployments and enables mobile sensors to compute their path on-line using only "local" information and adapt to the sensor field changes. This paper extends and generalizes the framework proposed in [12] by incorporating a probabilistic sensing model and a dynamic speed policy. In addition, the approach is now applicable for mobile nodes with variable speed and sensor fields that include obstacles. The main contribution of this paper is the development of an adaptive speed policy that maximizes the area coverage and at the same time minimizes the total distance travelled by mobiles (energy needed for mobility).

The remaining of the paper is organized as follows. Section II presents the distributed-collaborative path planning framework for mixed WSNs. Section III investigates the performance of the proposed framework and presents the simulation results. Finally, the paper concludes with Section IV.

II. PATH PLANNING FRAMEWORK

In this section we present a collaborative framework where the mobiles nodes autonomously decide their path to sample the areas least covered. In this architecture, at every step, the mobiles define a "local" area around their current location and identify the biggest coverage hole which becomes their next target point. Target points are then updated in a receding horizon like scheme. This approach works well and given enough time complete area coverage can be achieved.

At this point its worth pointing out that alternative path planning approaches like potential function techniques usually fail to address the problem under consideration as they get stuck in local minima or oscillate between two closest points [17]. Solutions provided to overcome the problem of local minima, when planning with potential functions, like wave-front planner [18], and *navigation functions* [19], [20] still fail to address the problem. Wave-front planner needs to search the entire space for a path each time the path is updated which is computationally intractable. On the

T. Lambrou and C. Panayiotou are with the KIOS Research Center for Intelligent Systems and Networks and the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus. {faniseng,christosp}@ucy.ac.cy

other hand, navigation functions assumed that obstacles are circular disks that do not intersect and the configuration space is bounded by a sphere (or a star) space. These assumptions are not satisfied in our setting.

Therefore, efficient path planning under random static sensor deployments is complicated, especially when the algorithm is intended to be robust to the sensor field density. To resolve these issues we have developed dynamic receding horizon policy which constitutes of appropriate normalizing path cost functions. It should be pointed out that the proposed path planing method is also applicable when static sensors are absent and thus mobile robots plan there trajectories to patrol or cover the given area under surveillance.

A. Sensor Network Model

We consider a mixed sensor network made of a large number of sensor nodes deployed in a large region $\mathcal{A}$ as shown in Fig. 1.

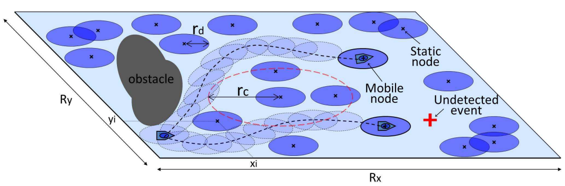


Fig. 1. Mixed sensor network model.

We assume that the region under monitored is a large rectangular area $\mathcal{A} = \mathcal{R}_x \times \mathcal{R}_y$ and a large set $\mathcal{S}$ of $S = |\mathcal{S}|$ static sensor nodes are randomly placed in the area $\mathcal{A}$, at positions $\mathbf{x}_i = (x_i, y_i)$, $i = 1, \cdots, S$. In addition, we assume that a small set $\mathcal{M}$ of $M = |\mathcal{M}|$ mobile sensor nodes are available and their position after the k-th time step is $\mathbf{x}_i(k) = (x_i(k), y_i(k))$, $i = 1, \cdots, M$, $k = 0, 1, \cdots$. For notational convenience, we define the set of *all* sensor nodes $\mathcal{N} = \mathcal{S} \cup \mathcal{M}$ and in this set the mobile nodes are re-indexed as $m = S+1, \cdots, N$, where $N = S + M$.

We assume that all sensors sense the environment according to the probabilistic sensing model [21]. This model is more realistic compared to the Boolean sensing model as it can capture the degradation of a sensor's sensing capability as the distance between the sensor and measuring point increases. In this model, a quantity r_u is defined in order to capture the uncertainty in sensor detection. According to this model, the initial (given one sample) probability that a sensor $s \in \mathcal{N}$ detects an event to a distance r is

$$p_s(r) = \begin{cases} 1, & r \leq r_u \\ e^{-\beta(r - r_u)^\gamma}, & r_u < r < r_d \\ 0, & r \geq r_d \end{cases} \qquad (1)$$

where, r_u defines the starting of uncertainty in sensor detection, r_d is the maximum sensing range of the node and the parameters β and γ are adjusted according to the physical properties of the sensor and the environment. This model is more general because it becomes Boolean sensing model when $r_u = r_d$. It is also assumed that all static and mobile nodes have common communication ranges $r_c > r_d$ as well as sensing characteristics and know their location through a combination of GPS and localization algorithms.

The neighborhood of a sensor $s \in \mathcal{N}$ is the set of all sensors nodes that are one hop away, i.e., the nodes that are located at a distance less than or equal to r_c from s. This set is denoted by

$$\mathcal{H}_{r_c}(s) = \{j \ : \ \|\mathbf{x}_s - \mathbf{x}_j\| \leq r_c, \ j \in \mathcal{N}, j \neq s\} \qquad (2)$$

where $\|\cdot\|$ denotes the Euclidean norm.

In addition, we consider a set $\mathcal{E}$ of $E = |\mathcal{E}|$ point events that can occur in $\mathcal{A}$ at positions $\mathbf{e}_i = (x_i^e, y_i^e)$, $i = 1, \cdots, E$. These events are uniformly distributed in the areas not monitored by the static sensor nodes and are temporally static, i.e. they occurred continuously in time. The case of temporally dynamic events is also addressed however it is omitted due to space limitations.

B. Event Detection

As previously implied, all static and mobile nodes sense the environment according to the probabilistic sensing model and it is assumed that all sensors' samples are temporally and spatially independent. An event is considered as detected (found) when is occurred at a point that falls within the sensing range r_d of a mobile sensor and the corresponding occurrence point is sensed with probability close to 1, given any concurrent measurements of neighboring static and mobile sensors. In other words, a binary variable $I_D(\mathbf{e}_j)$ is used to indicate whether an event $\mathbf{e}_j$ has been detected or not by a mobile sensor at the current step as follows:

$$I_D(\mathbf{e}_j) = \begin{cases} 1 & if \ P_D(\mathbf{e}_j) \geqslant \tau_d \\ 0 & otherwise \end{cases} \qquad (3)$$

where $P_D(\mathbf{e}_j)$ is defined below and τ_d is a pre-defined threshold close to 1

$$P_D(\mathbf{e}_j) = 1 - \prod_{i \in \mathcal{H}_{r_d}(\mathbf{e}_j)} (1 - p_i(r_{ij})) \qquad (4)$$

where $\mathcal{H}_{r_d}(\mathbf{e}_j) = \{i \ : \ \|\mathbf{e}_j - \mathbf{x}_i\| \leq r_d, \ j \in \mathcal{E}, \ i \in \mathcal{N}\}$ defines all sensors that are located at a distance less than or equal to r_d from the event $\mathbf{e}_j$[1].

C. Dynamic Coverage $\mathcal{C}_k$

To study the coverage of such mixed sensor networks we define a coverage measure called the dynamic coverage. Unlike static coverage (defined as the instantaneous ratio of covered area by the sensor network to the area of interest), dynamic coverage $\mathcal{C}_k$ is defined as the ratio of covered area by the sensor network to the area of interest during a time interval $[0, k]$. In other words, it defines the probability that a temporally static point event can be detected within a time interval $[0, k]$ by at least one sensor node in the sensor field. A similar measure of dynamic coverage is also considered in [5], [11], [14]. The dynamic coverage depends not only on

[1]Given that $\mathbf{e}_j$ is unknown, the evaluation of eq.4 requires each mobile to receive $p_i(.)$ of its neighbors and assumes that two events occurred at least $2r_d$ apart.

the sensing model, the number of nodes and node placement strategy, but also depends on the mobility behavior of the nodes. Hence, proper motion planning is required to exploit the full advantage of mobile sensors.

To make the concept of dynamic coverage a computationally measurable objective, the entire sensor field area $\mathcal{A}$ is discretized into an $X \times Y$ matrix G_k, $k = 0, 1, \cdots$. Initially, a zero value is assigned at each cell $G_0(i, j) = 0$, $i = 1, \cdots, X$, $j = 1, \cdots, Y$ and then at $k = 1$ (first sample) a value is assigned at each cell $G_1(i, j)$ depending on its distance from the stationary sensors as well as its distance for the initial position of the mobile sensors. Then, at each step k every sensor is sampling the environment and mobile sensors are moving around as well and thus the following updating rule is used for the G_k matrix,

$$G_{k+1}(i,j) = \begin{cases} 1 - (1 - G_k(i,j)) \prod_s (1 - p_s(\bar{r})), \\ \qquad \text{if } (i,j) \in D_{\bar{r}_d}(\bar{\mathbf{x}}_s),\ s \in \mathcal{N} \\ G_k(i,j), \\ \qquad \text{otherwise} \end{cases} \tag{5}$$

where $\bar{\mathbf{x}}_s$ are the coordinates of sensor s (mobile or static) in the grid G_k, $D_{\bar{r}_d}(\bar{\mathbf{x}}_s)$ is the set of grid cells covered by sensor $s \in \mathcal{N}$ with sensing range r_d, $\bar{r}$ is the discretized distance of cell $G_k(i, j)$ from $\bar{\mathbf{x}}_s$ and $p_s(\bar{r})$ is given by eq. (1).

The $\mathcal{C}_k$ represents the *dynamic coverage* over a time interval $[0, k]$ and it is an appropriate quality metric for applications that require coverage of all locations within some time interval. $\mathcal{C}_k$ also represents the probability of detection of static events existing in the sensor field within a time interval $[0, k]$

$$\mathcal{C}_k = \frac{1}{X \times Y} \times \sum_{i=1}^{X} \sum_{j=1}^{Y} G_k(i,j) \tag{6}$$

Therefore, when temporally static events are considered, the objective is to maximize the dynamic coverage rate over a time interval, this objective can be satisfied by finding the near-optimal paths to be followed by the mobile sensors in the sensor field in a distributive and collaborative manner. Its worth pointing out that finding optimal solutions to any arbitrary problem instance is not possible due to the complexity of the problem.

D. Mobile Sensor Node Model

Mobile sensor nodes can move in the sensor field and autonomously plan their trajectories to enhance the dynamic coverage and minimized the detection latency. The state of the m-th mobile node at time k is denoted by its position $\mathbf{x}_m(k)$ and its heading direction $\theta_m(k)$. Each mobile node m is capable to move with variable speed $\upsilon_m(k) \in [\upsilon_{min} \quad \upsilon_{max}]$ and make path planning decisions at discrete time intervals. Note that this model also considers the maneuverability constraints of the mobile platform using some angle ϕ which constrains the maximum allowed difference between $\theta_m(k)$ and $\theta_m(k + 1)$ and allows variable speed with maximum velocity of υ_{max}.

Finally, we describe the information required by each mobile in order to run the proposed path planning algorithm. Each mobile uses a *coverage cognitive map*, an $X \times Y$ matrix P_k^m, $m \in \mathcal{M}$ where it keeps the state of the field. Ideally P_k^m should remain $P_k^m = G_k$ at all times k, since the matrix G_k represents the accurate global state of the field which is used for the computation of the dynamic coverage $\mathcal{C}_k$. Clearly, in a dynamic environment where several sensors move, fail or more sensors are added as well as due to limited communication between mobile nodes, it is impossible to guarantee that $P_k^m = G_k$ at all times. However, we emphasize, that the proposed algorithm, that will run by a mobile located at some position $\mathbf{x}_m(k)$, computes its path based *only* on local information, i.e., information in the submatrix of P_k^m that corresponds to the cells $\mathcal{D}_{\bar{r}_c}(\bar{\mathbf{x}}_m(k))$, and thus, it is sufficient to have accurate information only for the $\mathcal{D}_{\bar{r}_c}(\bar{\mathbf{x}}_m(k))$ submatrix. This is easily attainable since the required information can be obtained from the one-hop neighbors.

E. Distributed Path Planning

The path planning method is based on Receding-Horizon approach where at each step the mobile's controller evaluates the cost of moving to a finite set of candidate positions and moves to the one that minimizes an overall cost.

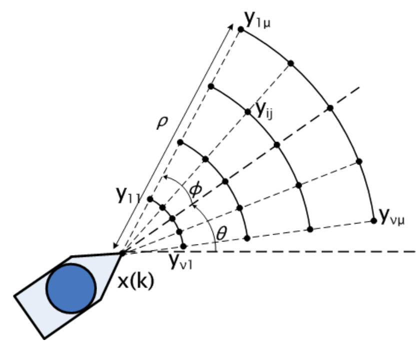


Fig. 2. Evaluation of the mobile node's next step.

As shown in Fig. 2, suppose that during the kth step, the mobile node is at position $\mathbf{x}(k)$ and its heading to a direction θ. The next candidate positions are the $\nu\mu$ points $\mathbf{y}_{11}, \cdots, \mathbf{y}_{\nu\mu}$ that are distributed on a circular sector with center $\mathbf{x}(k)$, radius ρ and angle $\theta - \phi$ and $\theta + \phi$, where $\nu \in \{2n + 1, \forall n \in \mathbb{Z}^+\}$. The mobile node evaluates a cost function $J(\mathbf{y}_{ij})$ for all candidate locations $(\mathbf{y}_{11}, \cdots, \mathbf{y}_{\nu\mu})$ and moves to the location $\mathbf{x}(k+1) = \mathbf{y}_{i^*j^*} = \mathbf{x}(k) + \frac{j^*\rho}{\mu}.e^{\mathrm{i}(\theta + \varphi_{i^*})}$ where i is the imaginary unit and i^*j^* are the indexes that minimize $J(\mathbf{y}_{ij})$,

$$i^*j^* = \arg \min_{\substack{1 \le i \le \nu \\ 1 \le j \le \mu}} \{J(y_{ij})\} \tag{7}$$

In this model, θ is the direction that the mobile is heading, ϕ is the maximum angle that the mobile can turn in a single step, $\nu \times \mu$ is the number of candidate positions that are being evaluated for the next step and ρ is the maximum distance that the mobile can cover in one time step when the mobile is moving with its maximum velocity υ_{max}. This model allows the mobile node to select an appropriate speed level at each time step and thus to adapt its speed based on the objectives that its tries to achieve. Assuming that the time t_s between two consecutive time steps k and $k+1$ is constant, we get $t_s = \frac{\rho}{\upsilon_{\max}} = \frac{j\rho}{\mu\upsilon_i}$. Hence, these discretized speed levels are defined by

$$\upsilon_i = \frac{j}{\mu}\upsilon_{\max},\ j = 1, \cdots, \mu \tag{8}$$

The objective function $J(y_{ij})$ that each mobile is trying to minimize is of the form

$$J(\mathbf{y}_{ij}) = \sum_{o \in \mathcal{O}} w_o J_o(\mathbf{y}_{ij}) \tag{9}$$

where $\mathcal{O}$ is a set of indexes such that the functions J_o, $o \in \mathcal{O}$ are normalized cost functions with $0 \leq J_o \leq 1$ and are defined to achieve certain objectives. w_o are non-negative constant weights that are used to trade off these objectives. For the purposes of this section, $\mathcal{O} = \{t, s, c, a, b\}$ but other functions can also be included (e.g. a cost function that depend on time or the residual energy of mobile nodes). In order to improve the area coverage, the mobiles should move towards large uncovered regions and on their path, they should try (to the extend possible) to avoid areas that are covered by static sensors or have been covered by other mobile nodes. For the purposes of this paper the following normalized functions have been used: $J_t(\cdot)$ which penalizes positions that are away from large coverage holes, $J_s(\cdot)$ and $J_c(\cdot)$ which penalize positions that are close to regions been covered by other sensors (stationary or mobile), $J_a(\cdot)$ which enable mobiles to avoid obstacles and $J_b(\cdot)$ which prevents mobiles moving outside the region under monitored. Next, we present the formulas of these functions.

a) Target Cost Function: At each step k, the mobile node m uses the information stored in its P_k^m matrix to search for the center of the biggest coverage hole (uncovered region) at a radius r_z from its current location. This can be done efficiently using the zoom algorithm [12]. The zoom algorithm divides the submatrix of P_k^m that corresponds to the cells $\mathcal{D}_{r_z}(\bar{\mathbf{x}}_m(k))$ in a 2D divide-and-conquer manner and outputs the hole center position. The center of the hole becomes the current target destination point $\mathbf{x}_t$ of the mobile. The cost $J_t(\mathbf{y})$ is a function that pulls the mobile towards its target and is a function of the distance between the mobile and the target position. This cost function is given by

$$J_t(\mathbf{y}) = \frac{\|\mathbf{y} - \mathbf{x}_t\|}{r_z} \tag{10}$$

In this function, r_z is the maximum distance between the mobile node and its target and is used for normalization. The radius r_z is an important parameter of the path planning algorithm and previous results [22] indicate that is more beneficial (achieves better area coverage and event detection time) if the dynamic target is determined more closer ("locally") to the mobile as opposed to more "globally". Thus r_z range must be fairly small compared to the sensor field area. Note that since $r_z \leq r_c - r_d$ in order to have accurate information, a smaller r_z is advantageous as it implies that less information (i.e. less computation and communication) is needed for the coverage hole estimation.

b) Neighboring Sensor Cost Function: The objective of this function is to push the mobile away from areas covered by other sensors. The cost function $J_s(\mathbf{y})$ used involves a repulsion force that pushes the mobile away from its closest neighbor. The form of this function is given by

$$J_s(\mathbf{y}) = \max_{j \in \mathcal{H}_{r_c}(m)} \left\{ \exp\left(-\frac{\|\mathbf{y} - \mathbf{x}_j\|^2}{r_s^2} \right) \right\} \tag{11}$$

where $\mathcal{H}_{r_c}(m)$ is the set of all nodes in the communication range r_c of the mobile m. The detection range r_d quantifies the size of the region around the mobile m to be repelled by its neighbors.

c) Coverage Cost Function: The cost function $J_c(\mathbf{y})$, similarly to J_s, is designed to push the mobile away from areas that have been covered by other sensors (stationary or mobile) or by itself using the relevant information from the cognitive map of the mobile node. This function takes a larger value if the candidate position is adequately covered by other sensors and a small value otherwise. This cost function is given by

$$J_c(\mathbf{y}) = \frac{1}{\pi r_d^2} \sum_{\{i,j\} \in \mathcal{D}_{\bar{r}_d}(\bar{\mathbf{y}})} P_k(i,j) \tag{12}$$

where $\mathcal{D}_{\bar{r}_d}$ is the set of cells that exist in a discretized disk of the mobile's P_k matrix, centered at the position $\bar{\mathbf{y}}$ with radius of $\bar{r}_d$.

d) Obstacle Avoidance Cost Function: This function enable the robots to avoid hitting obstacles that exist in the environment. The obstacle avoidance cost function J_a is similar to J_s and its form is given by

$$J_a(\mathbf{y}) = \exp\left(-\frac{\big(r_o - \|\mathbf{y} - \mathbf{x}(k)\|\big)^{10}}{r_d^{10}} \right) \tag{13}$$

where r_d is the detection range and r_o indicates the distance of the obstacle's boundary from the mobile's current position $\mathbf{x}(k)$ and its provided by the mobile's on board range-finding sensors such as low cost ultrasonic sensors or infrared sensors. The information provided by range-finding sensors can be combined with the model presented in Fig. 2 to associate each candidate location $\mathbf{y}_{ij}$ with a cost. For instance, the geometry of detectors can be combined with each candidate direction φ_i, $i = 1, \cdots, \nu$ to provide the distance to obstacles associated with the candidate direction.

e) Boundaries Cost Function: For completeness, note that another cost function is used that prevents mobiles from stepping outside the field along with projection which means that mobile sensors return to the interior of the field whenever they reach to boundaries in a manner similar to that of a light wave reflecting on a mirror. This boundary cost function $J_b(\mathbf{y})$ penalizes all candidate positions $\mathbf{y}$ that are not included in the field area $\mathcal{A}$ and is given by

$$J_b(\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{y} \notin \mathcal{A} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

F. Distributed Asynchronous Collaboration Scheme

Since each mobile determines its path autonomously, when two or more mobiles come close to each other it is very likely that the information they will use to estimate the next target position will be the same and as a result they will all estimate the same target location. To avoid this problem we utilize a collaboration protocol that enables mobile nodes to exchange some information in order to avoid moving towards to the same point and search different areas.

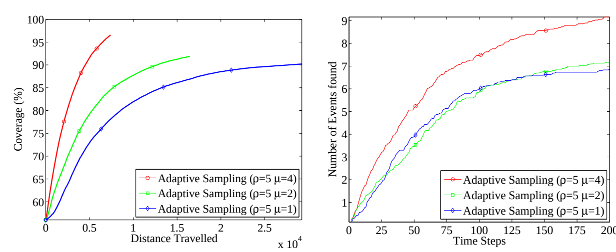
The collaboration protocol developed is as follows: If at step k, the mobile nodes come into communication range r_c when they were out of range in step $k-1$, they exchange their *entire maps* P_k^m *only* if their coverage difference exceeds a predefined threshold τ_C since the last time communicated [2]. If they are in r_c at time $k-1$ then they only exchange their *positions* $\mathbf{x}^m(k)$ and dynamic *target coordinates* $\mathbf{x}_t^m(k)$. Note that the P_k^m is exchanged only in an event driven way.

After a mobile node i has exchanged collaboration messages with its neighboring mobiles it has all the necessary information to execute the collaboration protocol. Thus at first, it merges its cognitive map P_k^i with the cognitive maps P_k^j, $j \neq i$ received from its "new" neighbors, so that it does not explore areas already explored by other mobile nodes. Merging policy is based on a cell value maximization rule. Afterwards, the mobile node i utilizes the current locations $\mathbf{x}^j(k)$ and dynamic target coordinates $\mathbf{x}_t^j(k)$ received by its neighboring mobiles $j \neq i$ (as well as the locations received by its neighboring stationary nodes) in order to update its P_k^i cognitive map and to avoid going towards the same point. Thus, once the mobile i has received all target points from its neighbors, it forms the matrix $D_{r_z}(\bar{\mathbf{x}}_i(k))$ (which is a copy of the set of P_k^i cells that corresponds to the distance r_z from the current position of mobile i) and updates the $D_{r_z}(\bar{\mathbf{x}}_i(k))$ matrix by assuming that these targets points constituted covered areas. Finally, it executes the zoom algorithm [12] where the input is the $D_{r_z}(\bar{\mathbf{x}}_i(k))$ matrix and the output is the dynamic target point $\mathbf{x}_t^i(k)$ of the mobile node i which is definitely different that the target points of its neighboring mobiles. As mobile nodes remain in communication range there is no need to exchange their cognitive maps since their maps are updated accurately using the positions of neighbors. It should be pointed out that proposed scheme is distributed (no need for a central controller) and utilizes only local information available in the neighborhood of the mobile node.

III. Simulation Results

In the first simulation, we investigate the parameters of the adaptive speed policy with respect to the average dynamic coverage and average number of static events detected using monte carlo simulations. We assumed 100 sensor fields with 300 randomly distributed stationary sensors and in each field 10 static events not initially detected exist. The objective is to investigate the adaptive versus the constant speed policy. The key parameter here is μ, for $\mu = 1$ mobiles are moving with constant maximum speed and as μ increases more speed levels and thus speed adaptivity is allowed, however as μ increases computation overhead also increases. These experiments refer to a square sensor field of area $A = 40000m^2$ and the sensors parameters are set to $r_d = 6m$ with $r_u = 4m$, $\beta = \gamma = 1$. The radius r_z where the dynamic target is found is set to $r_z = 19m$ and $r_c = r_z + r_d = 25m$. The weights are set to $w_t = 0.5, w_s = 0.2, w_c = 0.3, w_a = w_b = 1$ and thresholds for event detection and exchange of cognitive maps are set to $\tau_d = 0.999$ and $\tau_C = 5\%$ respectively. The mobile maneuverability parameters are set to $\rho = 5m$ and $\phi = 40°$ while for every decision $\nu\mu$ candidate next positions are considered with $\nu = 5$. Fig. 3 depicts the results for $\mu = 1, 2, 4$.

As shown in Fig. 3 the performance in terms of average dynamic coverage as well as the number of events detected increases but the total distance travelled by mobile nodes decreases!. This is an advantageous and desirable behavior and can be justified because using an adaptive speed policy (i.e. modifying the speed of the mobile at each step) enable mobiles to make more precise movements, which decreases the distance the mobiles moved (i.e. the energy needed for mobility) and simultaneously increases the dynamic coverage performance over time. In other words, if a mobile considers more candidate positions including positions that fall very near to it, (meaning going slower or make more precise navigation) it enables the mobile to go slower when needed but still have the option to go fast and thus decide its next position more accurately.



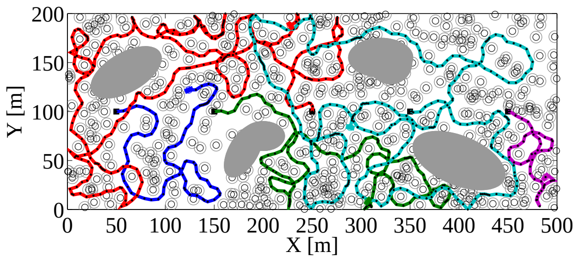
(a) Coverage Vs Distance. (b) Detected static events Vs Time.

Fig. 3. The average coverage and average number of detected static events accomplished by $M = 5$ mobile nodes after 200 moving steps for the adaptive and constant maximum speed policies.

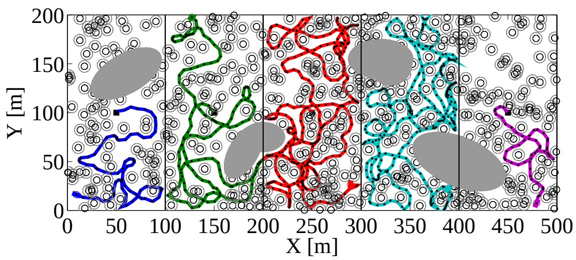
Finally, the last simulation evaluates the robustness of the proposed Distributed Collaborative coverage path planning Algorithm (DCA) with respect to the failures of mobile sensors. The lifetime of mobile sensors is modeled as an exponential distribution with failure rate $\lambda = 1/T$, where T denotes the simulation time. To illustrate the effectiveness of the proposed DCA, we have implemented another Centralized Partitioning coverage path planning Algorithm (CPA). In CPA is assumed that a central controller partitions the area under monitored into m equal partitions, where m denotes the number of mobile sensors, and assigns each mobile to a different partition. Fig. 4 illustrates the paths followed using

[2]Each mobile must keep in its memory a communication matrix where it tracks with which mobiles was in communication during the previous step as well as what was its coverage value since the last time it communicates with another mobile.

the DCA and CPA on the same sparse sensor field with 500 stationary sensors and 4 obstacles. The other parameters used for this simulation are the same as in the previous simulation. Note that in this scenario three mobile sensors fail before the end of the simulation time $T = 300$ time steps.



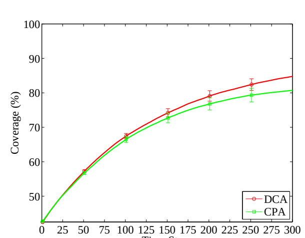
(a) Coverage path planning using DCA.



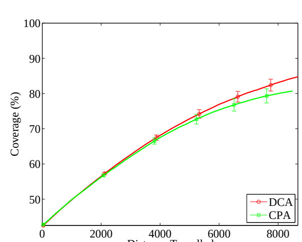
(b) Coverage path planning using the CPA.

Fig. 4. Paths followed using DCA and CPA under mobile sensor failures in a sparse sensor field with obstacles.

The two approaches are evaluated/compared using extensive monte carlo simulations for the case when mobiles failed according to the exponential distribution with failure rate $\lambda = 1/300$. We assumed 100 sensor fields with 500 randomly distributed stationary sensors and for each field identical initial positions and failures of mobile sensors are considered when simulating each algorithm. Results are shown in Fig. 5. As expected DCA outperforms CPA in both coverage and distance travelled performance due to its adaptive and distributed behavior. Therefore the proposed DCA is robust and adaptive to sensor node failures.



(a) Coverage Vs Time.



(b) Coverage Vs Distance.

Fig. 5. DCA vs CPA under mobile sensor failures: Average coverage accomplished by $M = 5$ mobile nodes after 300 moving steps.

IV. Conclusion

This paper presents an efficient adaptive-distributed-collaborative framework for mixed WSNs where autonomous mobile sensors navigate through a sparse stationary WSN searching for events and improving area coverage. An adaptive speed policy have been proposed to improve the performance and the energy consumption of mobile sensors and the robustness of the proposed approach has been evaluated under mobile sensor failures.

Acknowledgments

This work is partly supported by the Cyprus Research Promotion Foundation under grant ΤΠΕ/ΟΡΙΖΟ/0609 (ΒΕ) /06 and co-funded by the Republic of Cyprus and the European Regional Development Fund.

References

[1] A. R. Washburn and M. Kress, *Combat Models*. Springer, 2009.
[2] S. J. Benkoski, M. G. Monticino, and J. R. Weisinger, "A survey of the search theory literature," *Naval Research Logistics*, vol. 38, no. 4, pp. 469–494, 1991.
[3] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *in Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.
[4] M. Cardei and J. Wu, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press, 2004, ch. 19.
[5] B. Liu, O. Dousse, P. Nain, and D. F. Towsley, "Dynamic coverage of mobile sensor networks," *CoRR*, vol. abs/1101.0376, 2011.
[6] J. Cortes, "Coverage optimization and spatial load balancing by robotic sensor networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 749 –754, 2010.
[7] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *Proceedings of 44rd IEEE CDC*, 2005.
[8] A. Deshpande, S. Poduri, D. Rus, and G. S. Sukhatme, "Distributed coverage control for mobile sensors with location-dependent sensing models," in *Proceedings of the IEEE ICRA'09*, 2009, pp. 3493–3498.
[9] A. Howard, M. Mataric, and G. Sukhatme, "Mobile sensor network deployment using potential fields:a distributed, scalable solution to the area coverage problem," in *DARS*, 2002.
[10] A. Ghosh, "Estimating coverage holes and enhancing coverage in mixed sensor networks," in *Local Computer Networks*, 16-18 Nov. 2004, pp. 68–76.
[11] T. Wimalajeewa and S. K. Jayaweera, "Impact of mobile node density on detection performance measures in a hybrid sensor network," *IEEE Trans. Wireless. Comm.*, vol. 9, no. 5, pp. 1760–1769, 2010.
[12] T. P. Lambrou, C. G. Panayiotou, S. Felici-Castell, and B. Beferull-Lozano, "Exploiting mobility for efficient coverage in sparse wireless sensor networks," *Wireless Personal Communications Journal*, vol. 54, no. 1, pp. 187–201, 2010, iSSN: 0929-6212.
[13] T. P. Lambrou and C. G. Panayiotou, "A testbed for coverage control using mixed wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 527–537, 2012.
[14] E. Yanmaz and H. Guclu, "Stationary and mobile target detection using mobile wireless sensor networks," in *IEEE INFOCOM 2010 Workshop on Computer Communications*.
[15] N. Bartolini, T. Calamoneri, T. La Porta, and S. Silvestri, "Mobile sensor deployment in unknown fields," in *IEEE INFOCOM 2010*.
[16] S. Megerian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Worst and best-case coverage in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 84 – 92, 2005.
[17] T. P. Lambrou and C. G. Panayiotou, "Improving area coverage using mobility in sensor networks," in *ISYC 2006*, Ayia Napa, Cyprus.
[18] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 2, pp. 224 –241, 1992.
[19] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential fields," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
[20] S. G. Loizou and K. J. Kyriakopoulos, "Navigation of multiple kinematically constrained robots," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 221–231, 2008.
[21] A. Hossain, P. Biswas, and S. Chakrabarti, "Sensing models and its impact on network coverage in wireless sensor network," in *the 3rd IEEE Conference on Industrial and Information Systems*, Dec 2008.
[22] T. P. Lambrou and C. G. Panayiotou, "On the optimal search neighborhood in mixed wireless sensor networks," in *50th IEEE Conference on Decision and Control*, Orlando, Florida, USA, Dec. 2011.