Dynamic Scheduling of Cover-Sets in Randomly Deployed Wireless Video Sensor Networks for Surveillance Applications

Abdallah Makhoul and Congduc Pham

Computer Science Laboratory (LIUPPA), University of Pau

{abdallah.makhoul, congduc.pham}@univ-pau.fr

UFR Sciences et Techniques, Avenue de l'Université, BP : 1155, 64013 Pau CEDEX France

Abstract

A Wireless Video Sensor Network (WVSN) consists of a set of sensor nodes equipped with miniaturized video cameras. Unlike omni-directional sensors, the sensing region of a video node is limited to the field of view of its camera. Power conservation and coverage is an important issue in such wireless video networks, especially in the context of surveillance applications which is the focus of the article. In this paper, we address the area coverage problem of scheduling the activity of randomly deployed nodes to extend the network lifetime. We present a distributed algorithm for area coverage (no known targets). Moreover, we show that our approach reduces inherent ambiguities when it is necessary. Simulation results are also presented to verify the performance of the proposed approach.

I. Introduction

A Wireless Video Sensor Networks (WVSN) consists of a set of sensors nodes equipped with miniaturized video cameras. This type of networks covers a very large field of applications. In this paper, we are interested more particularly on WVSN for surveillance applications. Traditional vision systems for surveillance applications are built essentially from distributed high resolution video cameras and powerful processing units which communicate in general with central servers via a high bandwidth network. The target application of these systems is mainly infrastructure-oriented surveillance applications where deployment is performed manually on place of particular interest which are well-identified: parking lots, building entrances, hospitals, airports,... Using autonomous and small wireless video nodes can add a much higher level of flexibility, therefore extending the range of surveillance applications that could be considered and, more interestingly, can support dynamic deployment scenario even in so-called object and obstacle-rich environments or hard-to-access areas. These are the main advantages of using wireless video sensor nodes which can in addition be thrown in mass to constitute a large scale surveillance infrastructure. In these scenarios, hundreds or thousands of video nodes of low capacity (resolution, processing and storage) of a same or similar type can be deployed in an area of interest. These nodes would use collaboration mechanisms to ensure a surveillance task according to a given application and to transmit, via an ad-hoc network (mostly wireless), useful video data to one or more base stations. Desired features of such a surveillance infrastructures are high reliability and availability, largest coverage, disambiguation capabilities to name a few [1].

Central to the WSN research domain is the organization of sensor nodes to ensure correct coverage. The problem of coverage in many-robot system or WSN was largely studied and very interesting results were published. Most of the recent existing works on the connected coverage problem in sensor networks [2], [3], [4], [5] typically assume omnidirectional sensors with disk-like sensing coverage. To preserve coverage due to dynamical network topology changes, redundancy is introduced, so-called k-coverage [6], to ensure fault-tolerance and to increase network lifetime. Thus, two scalar nodes are likely to be redundant if they are close to each other. However, in wireless video sensor networks, video nodes possess "limited" sensing coverage area (sector coverage) due to the camera constraints and its Field of View (FoV). A number of studies have considered the directional sensor category in which video sensors fall into [7], [8], [9], [10]. In this paper, we propose a distributed approach dedicated to save energy and reduce ambiguities in WVSN. The main contributions are: **(a)** a video sensing model that allows us to simply define the redundancy of a FoV based on geometric computations. Based on this method every node can compute the set of nodes that cover its FoV. **(b)** a new distributed algorithm to manage the activity of randomly deployed video nodes while ensuring the whole area coverage. The novelty of this algorithm comes from the fact that it allows nodes to construct non-disjoint cover sets. **(c)** a disambiguation feature based on the availability of multiple cover sets that allows the WVSN infrastructure to take into account the surveillance application criticality.

The rest of this paper is organized as follows: in the next section we present a review of some previous related work. Section III presents a video sensing model. Section IV discusses the details of the proposed coverage scheme for video surveillance. A simple FoV coverage model is introduced for the purpose of building multiple cover sets. Then we describe the scheduling algorithm and the ambiguity reduction method based on the availability of cover sets. In section V, we describe simulation and results of simulation experiments. Finally, we end the paper by conclusions.

II. The Coverage and Scheduling problem

The coverage problem for wireless video sensor networks can be categorized as:

- *Known-Targets Coverage Problem,* which seeks to determine a subset of connected video nodes that covers a given set of target-locations scattered in a 2D plane.
- *Region-Coverage Problem,* which seeks to find a subset of connected video nodes that ensures the coverage of the entire region of deployment in a 2D plane.

Most of the previous works have considered the known-targets coverage problem [10], [11], [12], [13]. The objective is to ensure at all-time the coverage of some targets with known locations which are deployed in a two-dimensional plane.

Concerning the area coverage problem, the most existing works focus on finding an efficient deployment pattern so that the average overlapping area of each sensor is bounded. The authors in [7] analyze new deployment strategies for satisfying given coverage probability requirements with directional sensing models. A model of directed communications is introduced to ensure and repair the network connectivity. Based on a rotatable directional sensing model, the authors in [8] present a method to deterministically estimate the amount of directional nodes for a given coverage rate. A sensing connected sub-graph accompanied with a convex hull method is introduced to model a directional sensor network into several parts in a distributed manner. With adjustable sensing directions, the coverage algorithm tries to minimize the overlapping sensing area of directional sensors only with local topology information.

Different from the above works, our paper mainly focuses on the area coverage problem (no known targets a priori) and more precisely on scheduling for randomly deployed video sensor nodes. The objective is to schedule video nodes in a way to guarantee the coverage of the initial covered area and the network connectivity. A high number of video nodes are randomly scattered in a determined region. We assume the sensing region (FoV of the camera) of each node to be a sector of the sensing disk centered at the sensor. However, the algorithm proposed in this paper does not assume rotation capabilities to put restrictions on the overlaps between nodes. Our approach is based on a distributed algorithm that helps each node to organize its neighbors into non-disjoint subsets, each of which being a cover set that overlaps its FoV. Then, based on neighbors activity, a node decides to be active or in sleep mode. In a next step, our algorithm re-organizes the active nodes in a manner to reduce images ambiguities and to improve the detection capabilities.

III. Video Sensing Model

A video sensor node v is represented by the FoV of its camera. In our approach, we consider a 2-D model of a video sensor node where the FoV is defined as a sector denoted by a 4-tuple $v(P, R_s, \overrightarrow{V}, \alpha)$. Here P is the position of v, R_s is its sensing range, $\overrightarrow{V}$ is the vector representing the line of sight of the camera's FoV which determines the sensing direction, and α is the offset angle of the FoV on both sides of $\overrightarrow{V}$. Figure 1a illustrates the FoV of a video sensor node in our model.

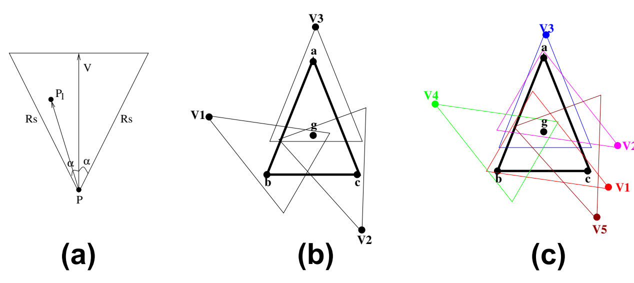

Fig. 1. Video sensing and coverage model

A point P_1 is said to be in the FoV of a video sensor v if and only if the two following conditions are satisfied:

1) $d(P, P_1) \leq R_s$, where $d(P, P_1)$ is the Euclidean distance between P and P_1.
2) The angle between $\overrightarrow{PP_1}$ and $\overrightarrow{V}$ must be within $[-\alpha, +\alpha]$.

In other words, this two conditions are met if:

$$\|\overrightarrow{PP_1}\| \leq R_s \text{ and } \overrightarrow{PP_1}.\overrightarrow{V} \geq \|\overrightarrow{PP_1}\| cos\alpha. \quad (1)$$

In the reminder of this paper, we consider that all video nodes have the same characteristics: same sensing range R_s and same offset angle α.

IV. Coverage and nodes scheduling

As we mentioned before, our approach is completely distributed and every video node computes its local solution for coverage. Each node v covers a sector area thanks to its FoV, which we call v's FoV area. Then, its local coverage objective is to ensure, at all time, the coverage of this area either by itself (if it is active) or by a subset of its neighbors. If every video node ensure that its local coverage objective is satisfied then the global coverage is also satisfied.

A. Area coverage and sensors cover set

In our approach we consider that video nodes are randomly deployed in a given area (called area of interest). Finding the optimal pattern to cover a 2D plane with video nodes (with sector sensing area) still an open problem. It usually needs a large number of video nodes ensuring complete coverage. Hence, this can lead to redundant nodes (nodes that monitor the same region) and overlaps among the FoV areas. In this section, we present a simple FoV coverage model. It allows every video node v to compute non-disjoint subsets of nodes, each subset covering its FoV area.

Definition 1: We define a cover $Co_i(v)$ of a video node v as a subset of video nodes such that: $\bigcup_{v' \in Co_i(v)}(v'$'s FoV area$)$ covers v's FoV area.

Definition 2: $Co(v)$ is defined as the set of all the covers of node v.

An example of FoV coverage is shown in Figure 1b, where nodes v_1, v_2 and v_3 cover the FoV area of node v, represented by abc. In the case of an omnidirectional sensing, a node can simply determine what parts of the coverage disc is covered by its neighbors [14]. For the FoV coverage the task is more complex. Therefore, to compute $Co(v)$, we propose a simple model based on four distinctive points: a, b, c and g (the center of gravity of (abc)) to represent the FoV of v as shown in Figure 1b. Then, we say that v's FoV is covered by a set $Co_i(v) \in Co(v)$ if the two following conditions are satisfied[1]:

1) $\forall\ v' \in Co_i(v)$, v' covers the point g and at least one of the points $\{a, b, c\}$
2) a, b, c and g are covered by the elements of $Co_i(v)$.

In other terms, to compute $Co(v)$, a node v finds the following sets, where $N(v)$ represents the set of neighbors of node v:

- $A = \{v' \in N(v) : v'$ covers point a of the FoV$\}$
- $B = \{v' \in N(v) : v'$ covers point b of the FoV$\}$
- $C = \{v' \in N(v) : v'$ covers point c of the FoV$\}$
- $G = \{v' \in N(v) : v'$ covers point g of the FoV$\}$
- $AG = \{A \cap G\}$, $BG = \{B \cap G\}$, $CG = \{C \cap G\}$

Then, $Co(v)$ is set to the Cartesian product of sets AG, BG and CG ($\{AG \times BG \times CG\}$). Note that, the set-intersection function generates $n + m$ recursive calls in the worst case. Therefore, the intersection of 2 sets can be done with complexity of $O(n + m)$, where m and n are the cardinals of the two sets respectively. As the size of sets A, B, C and G is limited, a video node can easily computes the intersections.

In the example illustrated by Figure 1c, v's FoV is represented by $(abcg)$. To find the set of covers, node v finds the sets: $AG = \{v_2, v_3\}$, $BG = \{v_1, v_4\}$ and $CG = \{v_1, v_5\}$. Then, following the above method, the set of possible covers for v is: $Co(v) = \{\{v\}, \{v_2, v_1\}, \{v_3, v_1\}, \{v_2, v_4, v_5\}, \{v_3, v_4, v_5\}\}$.

This model allows a node v to construct $Co(v)$ of its FoV area. Hence, in some cases (e.g. when there are occlusions), it can occur that a cover does not ensure the complete FoV coverage. For example, it can happen that a cover satisfies the above conditions but it does not ensure the coverage of the entire FoV. In this case, one could consider more points in the previous model such as the midpoints of segments [ab], [ac] and [bc]. On the other hand, doing so will reduce the number of covers and consequently increase the number of active nodes. Once again, the main contribution here mainly relies on the dynamic scheduling of cover sets and the coverage model we proposed in this paper is a simple way to obtain such cover sets.

B. Scheduling randomly deployed nodes with cover-sets

We consider that all video nodes have the same communication range modeled by a disc of radius R_c. Two video nodes are called neighbors if there exist an edge between them in graph G.

Our framework for video surveillance operates in three phases. The first is a setup phase where each node v constructs its set of covers $Co(v)$. The second phase is the scheduling phase where each node decides to be active or in sleep mode. Our objective is to minimize the number of active nodes while ensuring the whole coverage area. The

[1]this assumption allows us to construct the set of covers in order to apply the scheduling algorithm

third phase is devoted to reduce the inherent ambiguities in case of intrusion detection.

1) Construction of $Co(v)$: At this step, each video node v constructs all possible covers ($Co(v)$) that satisfy its local coverage objective (e.g. covering its FoV area). These sets are constructed by considering all the communication graph neighbors. Each node diffuses its position P and its direction $\overrightarrow{V}$ to its neighbors and receives their informations. According to equation (1) each node constructs the sets A, B and C as explained in section IV-A. Then, it computes $Co_i(v)$ that overlap its FoV and ensure its coverage (cf section IV-A).

In the literature, most of existing omni-directional sensing coverage works try to construct disjoint sets of active nodes [2], [3], [4], [5]. In our case, we have the possibility that two or more covers have some video nodes in common. This dependency must be taken into account in the scheduling phase. Hence, selecting one cover also reduces the life time of the sensor it has in common with another cover. In this case we can consider the level of energy as a criteria while choosing the active cover. The energy level of the lowest node in $Co_i(v)$ determines $Co_i(v)$'s energy level.

2) Video node scheduling: The activity of video sensor nodes operates in rounds. For each round, every node decides to be active or not based on the activity messages received from its neighbors. Every node orders its sets of covers according to their cardinality, and gives priority to the covers with minimum cardinality. If two sets or more in Co have same cardinality, priority is then given to the cover with the highest level of energy. Note that, another criteria can be defined.

A video node v receives the activity decisions of its neighbors. Then, it tests if the active nodes belongs to a cover $Co_i(v)$. If yes, it goes in sleep mode after sending its decision to its neighbors. In the case where no $Co_i(v)$ is satisfied, node v decides to remain in active mode and diffuses its decision.

A video node v orders the set $Co(v)$ according to their priorities. Then, it starts with the first cover $Co_1(v) \in Co(v)$ (which has the lowest cardinality) and tests if it is satisfied. A cover is comprised by a set of video nodes and if one of these switches off, this cover cannot be satisfied.

The node's scheduling process is summarized in Algorithm 1. Node v receives an activity message from its neighbor v', if $v' \in Co_1(v)$ decided to be inactive, then v goes to the next cover and so on until it finds an active cover or decides to be active itself. If node $v' \in Co_1(v)$ is active then node v check whether all nodes of $Co_1(v)$ are in active mode in order to go in sleep mode. This process is repeated for each cover and at every round.

3) Reducing ambiguities: Some applications such as surveillance and security applications, emergency detection in clinical environments and intrusion detection and tracking need more comprehensive interpretation of events or gestures. Access to multiple sources of visual data often allows for reducing ambiguities to allow for more accurate interpretation. Multiview has several advantages. First, the multi-view cameras can help circumvent occlusions. Second, even without occlusions, the information obtained from a single camera may be ambiguous for decision making, whereas a combination of information from multiple views may convey a higher confidence interpretation. Therefore, in our approach we allow collaboration among multiple cameras to reduce ambiguities, by adapting the activity nodes scheduling in a way to obtain more information about a target when it is necessary. In other words, to obtain multi-view of a target coming from more than one video node. Our approach does not address the way for processing data in a collaborative manner. For this task, one can find multiple works in the literature for centralized or distributed approaches [15].

Algorithm 1 Scheduling of node v

v is active
v orders its covers $Co_i(v) \in Co(v)$ $i = 1, 2, ...|Co(v)|$
$i \leftarrow 1$
while $i \leq |Co(v)|$ **do**
  v begins with the cover with highest priority $Co_i(v)$
  if neighbor v' decides to go in sleep mode **then**
    if $v' \notin Co_i(v)$ **then**
      continue with $Co_i(v)$
  else
    v chooses the next best priority cover $Co_{i+1}(v)$
    $i \leftarrow i + 1$
  if v' decides to remain active **then**
    if $v' \in Co_i(v)$ **then**
      continue with $Co_i(v)$
  if $\forall v'$, $v' \in Co_i(v)$, v' is active **then**
    v becomes inactive and sends its decision to its neighbors
if no $Co_i(v)$ is found **then**
  v remains active and sends its decision to its neighbors

The problem studied here is known in the literature as the k-coverage problem [6]. Redundant sensing capabilities are usually required in sensor network applications for robustness, fault tolerance, or increased accuracy features. At the same time high sensor redundancy offers the possibility of increasing network lifetime by scheduling sleep intervals for some sensors and still providing continuous service with help of the remaining active sensors. In our approach, we use the k-coverage feature provided by the availability of multiple cover sets in order to reduce ambiguities by allowing video nodes to see the critical object from different perspectives. When a node detects a critical event (e.g. an intrusion) it sends an urgent message to its neighbors to end the current round and begin a new one with a new scheduling scheme. The new scheduling must ensure that the target is covered by at least two or more video nodes. Therefore, from $Co(v)$, video node v selects the cover that ensure the target's multi-coverage to be active. Then, it goes to sleep mode after sending its decision to its neighbors which in their turn schedule their activity, and a new round starts.

V. Experimental Results

To evaluate our approach we conducted a series of simulations based on the discrete event simulator OMNet++ (http://www.omnetpp.org/). The results were obtained from iterations with various densities on a $100 * 100m^2$ area. Nodes have equal communication ranges of $30m$, an offset angle α of $\pi/6$, a battery of 100 units, random position P and random direction $\overrightarrow{V}$. A simulation starts by a neighborhood discovery. Each node gathers positions and directions of its neighbors and finds the sets AG, BG and CG. Then, round by round each node decides to be active or not. At the end of a round, active nodes decrease their batteries by one unit. Simulation ends as soon as the subset of nodes with power left is disconnected (where all nodes don't have anymore neighbors). We run each simulation 15 times to reduce the impact of randomness.

A. Proportion of Active Nodes

In those series of experiments, we varied the deployed nodes density from 50 to 200 nodes in a $100m * 100m$ area. We noted at each round the percentage of active nodes, which is the average number of nodes involved in the active set over initial number of deployed sensors. This metric reflects, to a certain extent, the effectiveness of the proposed scheduling approach. Figures 2 and 3 show the evolution of this ratio, round by round, for each density and for $R_S = 15m$ and $25m$ respectively.

Before the first drop at round 100, which corresponds to the extinction of a subset of nodes (the nodes that initially didn't have redundant covers set), the number of active nodes varies from less than 67%, for density 50, to 36% for density 200. We notice that, increasing the sensing range decreases the percentage of active nodes at each round and increases the network lifetime.

B. Percentage of Area Coverage

The main objective of our approach is to maintain a full area coverage at each round. We define the full coverage area as the region covered initially by the whole network (i.e when all the deployed nodes are active). This area represents the union of all FoV areas of the deployed nodes. Figures 4 and 5 show the percentage of area coverage round by round for different nodes density and for $R_S = 15m$ and $25m$ respectively. This percentage is the ratio between the area covered by the set of active nodes over the initial coverage area.

As we noticed, the initial sensing coverage is preserved for 100 rounds which is equal to a node's lifetime. At round 100, as expected, a set of nodes run out of energy. We can observe that in all the cases our algorithm guarantees a sensing coverage of at least 67% of the deployment area.

C. Disambiguation feature

To test the disambiguation feature, we consider a rectangular object which traverses the area of interest from left to right. The objective of this experiment is to determine the object trajectory and the identification time. In a surveillance application, faster the identification is, faster the interaction with the user is.

The rectangle as shown in Figure 6 is composed of 8 points, and is said to be fully identified when all 8 points are identified. In figure 6, we observe that node v_1 can detect three points $\{b, d, h\}$, while node v_2 detects only $\{a\}$(considering that point a hides b and c) and v_3 sees $\{e, f\}$ (g is not well detected). In our simulation, v_2 and v_3 consider the object as critical, so the first node that detects the object enforces its neighbors to become active in order to have multiple views of the object.

The rectangular object ($4 \times 2m^2$) traverses a $100m * 100m$ area where we have randomly dispersed 150 video nodes. Then, we picked up the time taken by the network to identify the object, while varying its velocity. Figure 7 shows the variation of the identification time over the velocity of the object. Without the ambiguity reduction scheduling the identification time greatly depends on the velocity of the object while it is almost constant with the ambiguity reduction scheduling.

VI. Conclusion

Scheduling algorithms to save energy and prolong the network lifetime are of prime importance for sensor networks. However, algorithms designed for omni-directional sensor networks may not be suitable for video sensor networks. In this paper, we study the problem of coverage by video sensors in randomly deployed WVSN. We first present a model to find subsets of nodes that cover the FoV area of a given node. Then, we provide a distributed algorithm that allows nodes to decide to be active or in sleep mode, in order to maximize the network lifetime. Furthermore, to eliminate ambiguities and improve the quality of intrusion detection, our algorithm enforces a new

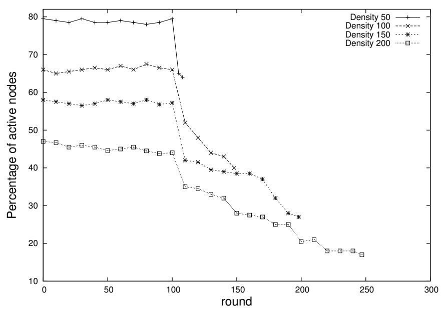

Fig. 2. $Rs = 15$

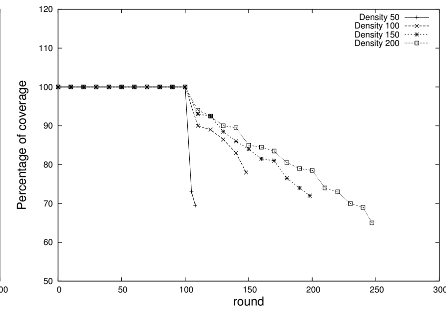

Fig. 4. $Rs = 15$

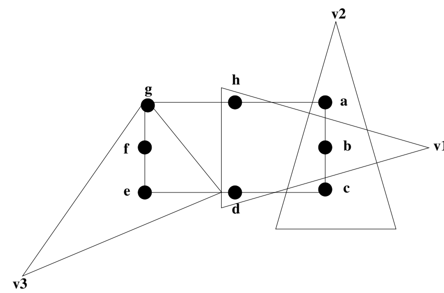

Fig. 6. Rectangular object in the sensors field

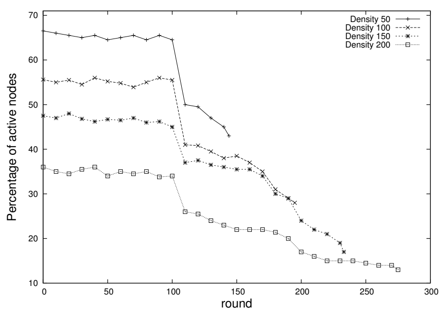

Fig. 3. $Rs = 25$

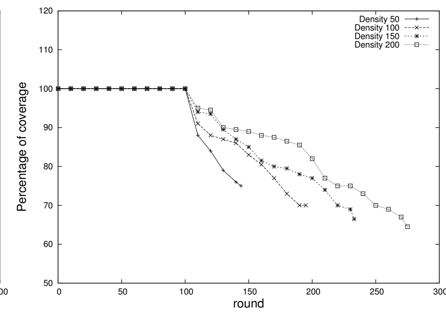

Fig. 5. $Rs = 25$

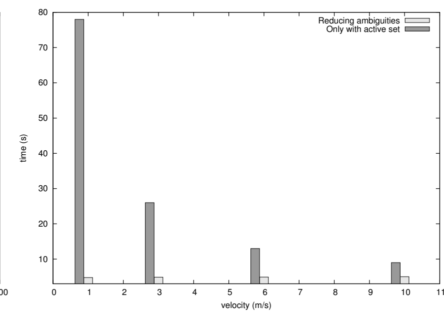

Fig. 7. Time of identification

scheduling. Finally, we evaluate the performance of the proposed approach through series of simulations. We show that our approach saves energy and improves the network lifetime while ensuring the area coverage. For instance, in high density networks our algorithm increases the network lifetime of more than 2.5 times. On the other hand, we showed that our proposed scheduling algorithm decreases the time to identify an object.

ACKNOWLEDGMENT

This work is partially supported by the ANR JCJC TCAP project of the French National Research Agency and by the Pyrnes-Atlantique Regional Council.

References

[1] S. Oh, P. Chen, M. Manzo, and S. Sastry, "Instrumenting wireless sensor networks for real-time surveillance," in *Proc. of the International Conference on Robotics and Automation*, May 2006.

[2] C. Wang, M. T. Thai, Y. Li, F. Wang, and W. Wu, "Coverage based information retrieval for lifetime maximization in sensor networks," *In IEEE GlobeCom*, Nov. 2007.

[3] J. Bahi, A. Makhoul, and A. Mostefaoui, "Hilbert mobile beacon for localization and coverage in sensor networks," *Int. Jour. of Systems Science*, vol. 39, no. 11, pp. 1081–1094, Novembre 2008.

[4] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Localized sensor area coverage with low communication overhead," *IEEE Trans. on Mobile Computing*, vol. 7, no. 5, pp. 661–672, 2008.

[5] J. Bahi, A. Makhoul, and A. Mostefaoui, "Improving lifetime and coverage through a mobile beacon for high density sensor networks," *SENSORCOMM08*, pp. 335–341, august 2008.

[6] W. Mo, D. Qiao, and Z. Wang, "Mostly-sleeping wireless sensor networks: Connectivity, k-coverage, and α-lifetime," *the Allerton Conference, UIUC*, 2005.

[7] H. Ma and Y. Liu, "Some problems of directional sensor networks," *Inter. Jour. of Sensor Networks*, vol. 2, no. 1-2, pp. 44–52, 2007.

[8] D. Tao, H. Ma, and L. Liu, "Coverage-enhancing algorithm for directional sensor networks," *LNCS - Springer*, pp. 256–267, November 2006.

[9] Y. Cai, W. Lou, M. Li, and X.-Y. Li, "Target-oriented scheduling in directional sensor networks," *26th IEEE INFOCOM*, pp. 1550–1558, 2007.

[10] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 1, pp. 21–41, 2006.

[11] J. Wang, C. Niu, , and R. Shen, "Randomized approach for target coverage scheduling in directional sensor network," *LNCS - Springer*, pp. 379–390, 2007.

[12] M. X. Cheng, L. Ruan, and W. Wu, "Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks," *in IEEE INFOCOM*, 2005.

[13] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," *IEEE INFOCOM*, pp. 1976–1984, 2005.

[14] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 32–41, 2002.

[15] T. Ahmedali and J. J. Clark, "Collaborative multi-camera surveillance with automated person detection," *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, p. 39, 2006.