

Performance study of multiple cover-set strategies for mission-critical video surveillance with wireless video sensors

Congduc Pham
University of Pau
LIUPPA Laboratory
Email: congduc.pham@univ-pau.fr

Abdallah Makhoul
University of Franche-Comté
LIFC Laboratory
Email: abdallah.makhoul@iut-bm.univ-fcomte.fr

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. In this paper, we study the problem of coverage by video sensors in randomly deployed WVSN. We focus on the performance of various fast cover set construction strategies for enabling efficient scheduling of nodes in mission-critical surveillance applications. Simulation results shows the performance of the various strategies in terms of percentage of coverage, network lifetime, intrusion stealth time and number of intrusion detection.

Index Terms—Sensor networks, video surveillance, coverage, mission-critical applications

I. INTRODUCTION

A Wireless Video Sensor Networks (WVSN) consists of a set of sensors nodes equipped with miniaturized video cameras. This type of networks is particularly suitable for applications that focuses on surveillance [1], [2], [3], [4]. In this article, we are interested more particularly on WVSN for mission-critical surveillance applications where sensors can be thrown in mass when needed. In this case, one of the first requirement is to ensure that that these randomly deployed video sensors will not land upside-down with the embedded camera turned towards the ground. This can actually be easily avoided by fitting the video sensor in a rocket-shaped case which will always touch ground in the right way as illustrated by figure 1(left) (iMote2 with IMB400 multimedia board [5]).



Fig. 1. A rocket-shaped video sensor.

Figure 1(right) shows a simple video surveillance application with this hardware that continuously takes pictures and displays both the current picture and the last picture.

Our research on wireless sensor networks focuses on mission-critical surveillance applications. Figure 2 shows a typical scenario of a random deployment of video sensor nodes for intrusion detection or disaster relief applications. In such deployment scenario, most of sensor nodes must move to a so-called *hibernate* mode in the absence of events in order to save energy. However, it is also highly desirable that some sensor nodes still keep a relatively high capture rate in order to act as sentry nodes in the surveillance system (figure 2a) to better detect intrusions/events and to alert other active nodes to move to an *alerted* mode (figure 2b). In [6] we proposed to take into account the application's criticality to define an appropriate level of service. For video sensors, the higher the capture rate is, the better relevant events could be detected and identified. Therefore a low criticality level indicates that the application does not require a high video frame capture rate while a high criticality level does. According to the application's requirements, an R^0 value that indicate the criticality level could be initialized accordingly into all sensors nodes prior to deployment. [6] also proposed to apply a risk-based approach for scheduling sensor nodes: different parts of the area of interest may have different risk levels according to the pattern of observed events such as the number of detected intrusions. In [7], the authors introduce so-called differentiated services by dynamically modify the time duration for a node to work during each round. As we directly linked the application criticality to the frame capture rate of a video sensor node, we want to impact on quality (number of frames) rather than on whole coverage as in [7]. Therefore, each sensor has a risk level noted r^0 that can have values in $[0, R^0]$. In the scenario of figure 2, after some time, an alerted node which does not detect more intrusions, should slowly go back to *hibernate* mode again by decreasing its risk level r^0 to 0 in order to save energy, see figure 2c. In this figure, we can also see that an alerted sensor node which does detect an intrusion (all sensor nodes close to the intruder's trajectory – dash line – in figure 2c) stays at r^0 close to the maximum value.

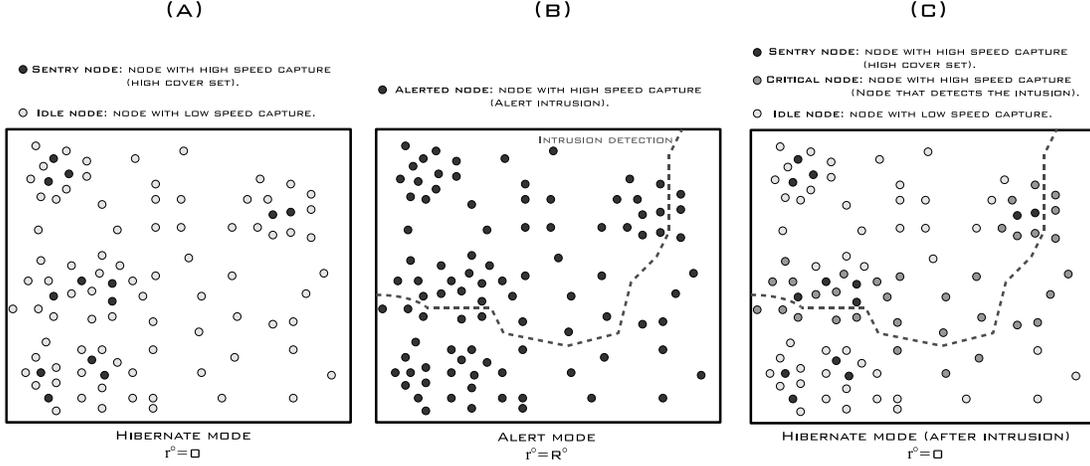


Fig. 2. Evolution of the video network nodes

However, even in the case of very mission-critical applications, it is not realistic to consider that video nodes should always capture at their maximum rate when in active mode. In randomly deployed sensor networks, provided that the node density is sufficiently high, sensor nodes can be redundant (nodes that monitor the same region) leading to overlaps among the monitored areas. Therefore, a common approach is to define a subset of the deployed nodes to be active while the other nodes can sleep. Many contributions have been made in the last few years and the authors in [8], [9], [10], to name a few, have proposed interesting energy-efficient approaches that aim at providing the highest detection quality. One obvious way of saving energy is to say that nodes that can be put in sleep mode are typically those whose sensing area are covered by others. However, in mission-critical applications where sentry nodes are desired as discussed previously, nodes that possess a high redundancy level (their sensing area are covered many times by other nodes) could rather be more active than other nodes with less redundancy level. The notion of cover set has therefore been introduced to define the redundancy level of a sensor [11]. In [6] the idea is that when a node has several covers, it can increase its frame capture rate because if it runs out of energy it can be replaced by one of its covers. The problem we specifically address in this paper is to efficiently define these cover sets in the case of video sensor for which the sensing range is defined by a field of view (FoV) and not by a disk. This task is complex and determining whether a sensor's FoV is completely covered or not by a subset of neighbor sensors is a time consuming task which is usually too resource-consuming for autonomous sensors.

This article focuses on the performance of various cover set construction strategies for enabling efficient scheduling of nodes in mission-critical surveillance applications. We present in section II some related works and our proposition of a coverage model for quickly building multiple cover sets per sensor while taking account very narrow angle of view (AoV) cameras and heterogeneous AoV deployment. The perfor-

mance of our coverage model proposition is then evaluated in detail in section III. In particular, we will present results for percentage of coverage, number of cover sets, disambiguation features, network lifetime, stealth time and intrusion detection. Conclusions are given in section IV.

II. VIDEO SENSOR COVERAGE MODEL

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 [12], [13], [14], [15] typically assumes omnidirectional sensors with disk-like sensing coverage. To preserve coverage due to dynamical network topology changes, redundancy is introduced, so-called k -coverage [16], to ensure fault-tolerance and to increase network lifetime. Thus, two scalar nodes are likely to be redundant if they are close each other. However, in wireless video sensor networks, video nodes possess "limited" sensing coverage area (sector coverage) due to the camera constraints and its FoV.

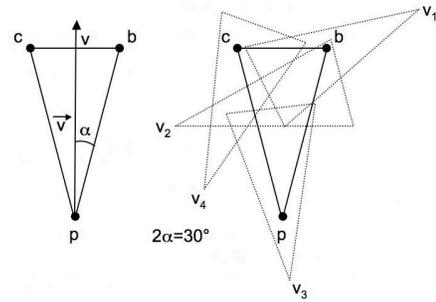


Fig. 3. Video sensing and coverage model

A video sensor node v is represented by the FoV of its camera. In our approach, we consider a commonly used 2-D model of a video sensor node where the FoV \vec{v} is defined as a triangle (abc) denoted by a 4-tuple $v(P, d, \vec{V}, \alpha)$. Here P is the position of v , d is the distance pv (depth of view,

DoV), \vec{V} is the vector representing the line of sight of the camera's FoV which determines the sensing direction, and α is the angle of the FoV on both sides of \vec{V} (2α can be denoted as the angle of view (AoV) in photography). The left side of figure 3 illustrates the FoV of a video sensor node in our model. The AoV (2α) is 30° and distance bc is the linear FoV which is usually expressed in ft/1000yd or millimeters/meter. By using simple trigonometry relations we can link bc to pv with the following relation $bc = \frac{2 \sin \alpha}{\cos \alpha} \cdot pv$.

Some wireless sensor platforms can therefore have a video camera board (iMote2 from Crossbow for instance [5]) where the embedded camera is similar to those that can be found on modern mobile phones or laptops. For instance, on an Samsung Player Addict mobile phone, distance bc is about 2.6m when pv is about 3.7m. This will result in an AoV of 38.71° . The iSight embedded camera on Apple PowerBook Pro has an AoV of about 31.68° . The IMB400 multimedia board for the Intel Mote2 sensor has an AoV of about 20° , which is rather small. Figure 1 shows a picture in the "last picture" window with a DoV that was experimentally measured to be 2.60m. The linear FoV at that distance is experimentally measured to be 0.92m. Obviously, the linear FoV and the AoV are important criteria in video sensor networks deployed for mission-critical surveillance applications. The DoV is a more subjective parameter. Technically, the DoV could be very large but practically it is limited by the fact that an observed object must be sufficiently big to be identified.

In randomly deployed sensor nodes scenarios, full coverage of the area of interest is not guaranteed, unless node mobility or camera rotation facilities are considered. In this paper, we will not consider such possibilities. However, even if full coverage is not guaranteed, if the density of node is sufficiently high, the percentage of coverage can be close to 100% with large parts of the area covered multiple times thank to redundancy of node's FoV as shown in the right part of figure 3 where the triangle abc is completely covered by the set $\{v_1, v_2, v_3, v_4\}$.

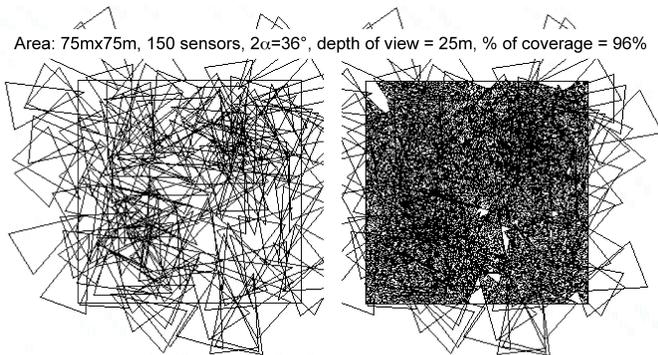


Fig. 4. 150 nodes in an 75m.75m field

Figure 4 shows a scenario where 150 video sensor are randomly deployed in an 75m.75m area (node density is $0.0266 \text{ sensor}/m^2$ or 1 sensor for $37.5m^2$). α is set to $\pi/10$ (AoV of 36°) and pv is set to 25m. The left part of the

figure shows the node's position and their FoV while the right part shows the area's covered parts. These results have been obtained by simulations where the position and the line of sight of each sensor are randomly defined. Note that the FoV of many sensors are outside the area of interest which can be useful in intrusion detection applications with barrier coverage concerns. In all the simulations the percentage of coverage has been greater than 90%. 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 [17], [18], [19], [11]. The objective is to ensure at all-time the coverage of some targets with known locations which are deployed in a two-dimensional plane. For example, the authors in [19] organize sensor nodes into mutually exclusive subsets that are activated successively, where the size of each subset is restricted and not all of the targets need to be covered by the sensors in one subset. In [11] the authors organize the sensors into a maximal number of set covers that are activated successively.

In [17], a directional sensor model is proposed, where a sensor is allowed to work in several directions. The idea behind this is to find a minimal set of directions that can cover the maximal number of targets. It is different from the one in [20] that aims to find a group of non-disjoint cover sets, each set covering all the targets to maximize the network lifetime. Concerning the area coverage problem, most existing works focus on finding an efficient deployment pattern so that the average overlapping area of each sensor is bounded. The authors in [21] 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 [22] 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 efficient scheduling of randomly deployed video sensor nodes where the final objective is to schedule video nodes in a way to guarantee a high percentage of coverage of the initial covered area while reducing the energy consumption.

A. Our FoV coverage model

As depicted earlier in figure 3(right) and figure 4, random deployment of sensor leads to a high level of redundancy. The main issue is then to quickly identify the different sets of sensors that cover v 's FoV to implement a distributed algorithm that helps each node to organize its neighbors into subsets, each of which being a cover set that overlaps its FoV. Then, based on neighbors activity, a node could decide to be active or in sleep mode. We define a cover set $Co_i(v)$ 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. $Co(v)$ is defined as the set of all the cover sets of node v .

In the case of an omnidirectional sensing, a node can simply determine what parts of the coverage disc is covered by its neighbors [23]. For the FoV coverage the task is more complex and determining whether a sensor's FoV is completely covered or not by a subset of neighbor sensors is a time consuming task which is usually too resource-consuming for autonomous sensors. To solve coverage problem in this kind of networks by handling three-dimensional geometrical forms is certainly not an easy task to achieve. In [24], video sensor nodes are deployed in a plane (noted P1) and their FoV are projected on another plane (noted P2). The covered area in this case can be represented by circles or rectangles in the P2 plane. They showed that a coverage study with such a simplification, by employing algorithms designed initially for coverage in scalar WSN, does not always give the desired performances.

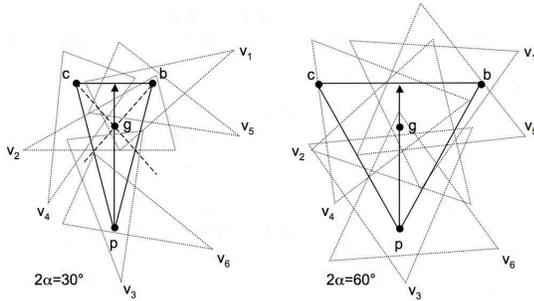


Fig. 5. Different angles of views.

Our idea is to use specific points of a sensor's FoV to quickly determine cover sets that may not completely cover sensor v 's FoV but a high percentage of it. First, sensor v can classify its neighbors into 3 categories of nodes, (i) those that cover point p , (ii) those cover point b and (iii) those that cover point c . Then, in order to avoid selecting neighbors that cover only a small portion of v 's FoV, we add a fourth point taken near the center of v 's FoV to construct a fourth set and require that candidate neighbors covers at least one of the 3 vertices and the fourth point. Figure 5(left) shows the usage of abc 's center of gravity in addition to the triangle vertices that makes sensors v_1 and v_2 eligible for covering vertex b .

The advantage of this method is that determining if a specific point x is covered by a sensor's FoV can be done in an easy and fast way. One exact method for determining whether a point is inside the abc triangle is the method described by

Moreno in [25]. This is actually the method we implemented for all the coverage simulations described in this paper.

B. Determining cover sets

Therefore, to compute $Co(v)$, we first propose a simple model based on four distinctive points: p, b, c and g (the center of gravity of (abc)) to represent the FoV of v as shown in figure 5(left). 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) $\forall v' \in Co_i(v)$, v' covers the point g and at least one of the points $\{p, b, c\}$,
- 2) p, b, c and g are covered by the elements of $Co_i(v)$.

Practically, node v computes $Co(v)$ by finding the following sets, where $N(v)$ represents the set of neighbors of node v :

- $P/B/C/G = \{v' \in N(v) : v' \text{ covers point } p/b/c/g\}$
- $PG = \{P \cap G\}$, $BG = \{B \cap G\}$, $CG = \{C \cap G\}$

Then, $Co(v)$ is set to the Cartesian product of sets PG, BG and CG ($\{PG \times BG \times CG\}$). 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 P, B, C and G is limited, a video node can rapidly computes the required intersections.

If we look carefully at figure 5(left) we can see that only sensors v_1 and v_2 do cover point g and another vertex (point b). The other sensors do not cover at the same time point g and another vertex making them not eligible for belonging to v 's coverset. This example illustrates the fact that with an AoV between 30° and 38° and a DoV of about 25m, the position of point g , which by definition is at $2/3$ of the triangle vertices, may not be the best position. As the sensor's FoV is modeled as an isosceles triangle the linear FoV (distance bc) is much shorter than the DoV (distance pv) as the AoV (2α) is smaller (remember $bc = \frac{2 \sin \alpha}{\cos \alpha} \cdot pv$). As the AoV increases this problem tends to decrease a bit although not being completely solved as the DoV usually remains the same. Figure 5(right) shows sensors at the same position but with an AoV of 60° instead of 30° . In this case we can have $PG = \{v_6\}$, $BG = \{v_1, v_5\}$ and $CG = \{v_4\}$. Therefore $Co(v) = \{\{v_6, v_1, v_4\}, \{v_6, v_5, v_4\}\}$ which is better than the previous situation where $Co(v) = \emptyset$. When 2α is small (resulting in a small linear FoV) it is necessary to adjust the position of point g (which therefore is not the center of gravity anymore) on the segment pv . Using one point is not enough as the more you move g towards p , the more g will be away of b and c , making it almost impossible to have a neighbor sensor covering both c and g or b and g !

When 2α is small we propose using 2 alternate points instead of the center of gravity g : point g_p between p and g (red point), and point g_v between g and v (yellow point). Recall that point v is the intersection between the line of sight \vec{V} and segment $[bc]$. Figure 6 depicts the case when g_p and g_v are the mid-point of their respective segment but it is possible to give a different weight. Therefore with $2\alpha = 30^\circ$, we have $PG =$

$\{v_3, v_6\}$, $BG = \{v_1, v_2, v_5\}$ and $CG = \{v_4\}$ resulting in $Co(v) = \{\{v_3, v_1, v_4\}, \{v_3, v_2, v_4\}, \{v_3, v_5, v_4\}, \{v_6, v_1, v_4\}, \{v_6, v_2, v_4\}, \{v_6, v_5, v_4\}\}$.

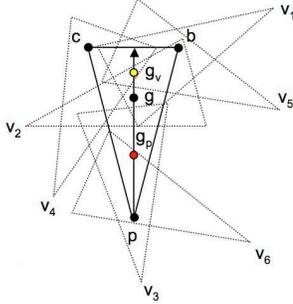


Fig. 6. Using alternate points.

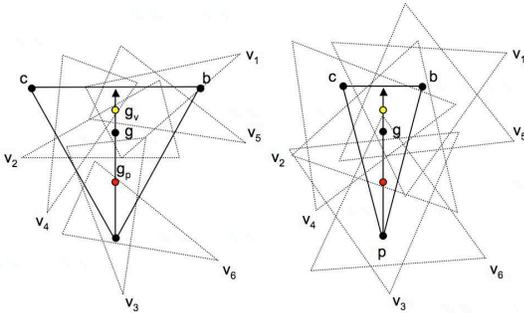


Fig. 7. Heterogeneous angles of view.

Only homogeneous scenarios have been considered so far but it is highly possible that video sensors with different angles of view are randomly deployed. In this case, a wide angle FoV could be covered by narrow angle FoV sensors and vice versa. Figure 7 shows these cases and the left part of the figure show the most problematic case when a wide angle FoV ($2\alpha = 60^\circ$) has to be covered by narrow angle FoV ($2\alpha = 30^\circ$). As we can see on the figure even with alternate points g_p (red point) and g_v (yellow point), it is difficult for a narrow angle sensor to cover one of these points and one of the b or c vertices at the same time.

The solution we propose is to use the same method as for alternate points g_p and g_v : use g_c and g_b that are set in figure 8(left) as the mid-point of segment $[cg]$ and segment $[bg]$ respectively (again it is possible to give a different weight).

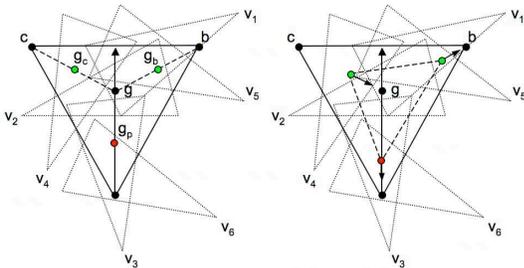


Fig. 8. Using more alternate points.

When using these additional points, it is possible to required either that a sensor v_x covers both c and g_c or g_c and g (the same for b and g_b) depending on whether the edges or the center of the FoV is privileged. Generalizing this method by using different weights to set g_c , g_b and g_p closer or farther from there respective vertices can be useful to set which parts of the FoV has more priority as depicted in the right part of figure 8 where g_c has moved closer to g , g_b closer to b and g_p closer to p .

Our simple model allows a node v to quickly construct $Co(v)$ of its FoV area. However, as said previously, this method is approximative and a cover can satisfy the specific points coverage conditions without ensuring the coverage of the entire FoV. In the next section we will show the performance in term of percentage of coverage of the various methods. In the rest of the paper, we will denote by COV_{woG} , COV_{wG} , COV_{waGpv} and COV_{waGbc} the following respective strategies: (i) only the vertices p , b or c are used, (ii) point g , which is the triangle's center of gravity, is taken into account when determining eligible neighbors to be included in a sensor's cover sets, (iii) alternates points g_p and g_v are used and, (iv) alternates points g_p , g_b and g_c are used.

III. PERFORMANCE RESULTS

To evaluate the accuracy of our cover sets techniques we conducted a series of simulations based on the discrete event simulator OMNet++ (<http://www.omnetpp.org/>).

A. Percentage of coverage and cover set size

We want to study the accuracy in terms of percentage of coverage of the initial FoV of the cover set construction strategies COV_{woG} , COV_{wG} , COV_{waGpv} , COV_{waGbc} . The results were obtained from iterations with various node populations on a $75m.75m$ area. Nodes have random position P , random line of sight \vec{V} , equal communication ranges of $30m$ (which determines neighbor nodes), equal DoV of $25m$ and an offset angle α . We will test with $2\alpha = 36^\circ$ ($\alpha = \pi/10$) and $2\alpha = 60^\circ$ ($\alpha = \pi/6$). A simulation starts by a neighborhood discovery. Each node gathers positions and directions of its neighbors and finds the sets PG , BG and CG . Then, each node decides to be active or not. We run each simulation 15 times to reduce the impact of randomness. The results (averaged over the 15 simulation runs) are summarized in table I where all sensors have the same AoV of either $2\alpha = 36^\circ$ or $2\alpha = 60^\circ$.

For each sensor v with cover sets the simulation computes the percentage of coverage of each cover set $Co_i(v)$ by sampling a large number of random points (50000 is our simulation) in v 's FoV and determining whether this point is covered by one of the sensor in $Co_i(v)$. We can see that with a small AoV, $2\alpha = 36^\circ$, using only point g creates strong constraints on neighbor selection leading to a very small number of sensor nodes with cover sets. When nodes do have cover sets, the number of cover sets is also very small. Therefore very few sensors could be in sleep mode so network lifetime is expected to be quite short. However, the good thing is that the mean percentage of coverage per

COV_{woG} 36° #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	38.22	46.96	14.40,86.27	18.08	1,45.33	9.59
100	49.67	47.67	16.58,83.05	15.53	1,33,110.66	20.31
125	64.80	48.67	15.48,89.40	15.67	1,150	29.98
150	67.78	48.81	12.77,90.88	15.67	1,66,170.33	35.93
175	75.24	47.83	18.50,89.56	13.31	1,66,412	82.24
COV_{wG} 36° #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	0	0	0,0	nan	0,0	0
100	1	92.03	89.78,98.64	0	1,1	1
125	1.87	91.45	88.83,93.15	2.97	1,33,2	1.56
150	1.78	95.06	91.47,98.29	4.06	1,3	1.94
175	3.43	94.42	87.60,99.03	4.40	1,33,2.66	1.92
COV_{waGpv} 36° #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	6.22	82.07	74.78,89.98	6.24	1,33,4	2.23
100	11	79.22	55.47,96.68	13.16	1,5,33	2.05
125	18.93	79.86	49.99,98.90	12.14	1,11,33	3.23
150	18.89	82.22	54.56,99.07	11.67	1,8,66	2.97
175	26.67	82.07	59.26,99.26	10.17	1,22,66	5.32
COV_{waGbc} 36° #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	12.44	77.48	56.46,91.81	13.33	1,33,9,33	3.62
100	20.33	79.62	53.65,98.98	12.05	1,10,66	3.94
125	30.67	76.89	50.53,97.92	11.58	1,34	5.40
150	35.11	78.47	52.07,96.09	10.60	1,31,33	6.90
175	48.57	77.76	49.97,98.20	10.54	1,50,33	11.57
COV_{woG} 60° #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	60.89	56.15	18.95,90.79	16.52	2,130	26.42
100	65.33	55.21	19.87,86.46	14.55	2,396	61.46
125	72	55.83	25.95,90.01	13.66	3,66,473.66	104.79
150	78.00	55.68	29.15,91.62	11.77	4,846.33	184.01
175	80.38	56.27	24.64,89.78	11.77	8,33,872	217.17
COV_{wG} 60° #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	4.89	94.04	90.36,98.15	3.67	1,5,66	2.20
100	7.33	94.63	86.99,98.49	4.40	1,6	2.99
125	11.73	95.06	85.20,99.52	4.12	1,13	3.53
150	17.11	95.44	84.99,82	3.98	1,16,33	4.15
175	26.29	94.64	83.57,99.89	4.01	1,35,66	6.40
COV_{waGpv} 60° #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	12.44	88.18	73.2,99.13	9.47	1,5	2.51
100	14.33	90.52	74.25,98.87	7.15	1,34,66	7.93
125	24.53	90.23	72.60,99.40	6.70	1,48	12.52
150	29.78	89.53	65.14,99.04	7.11	1,33,80	12.03
175	34.48	89.46	67.40,99.72	7.31	1,66,58	12.37
COV_{waGbc} 60° #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	35.56	77.91	58.98,94.93	10.07	1,18,66	6.68
100	50	79.18	56.00,98.57	10.38	1,59	11.40
125	58.13	80.42	57.68,98.77	8.61	1,33,130.66	27.38
150	66.89	81.32	53.9,96.46	8.34	1,33,164.66	37.63
175	73.33	81.93	53.79,98.39	8.15	1,33,260	52.45

TABLE I
RESULTS FOR COV_{woG} , COV_{wG} , COV_{waGpv} , COV_{waGbc} . $2\alpha = 36^\circ$ AND $2\alpha = 60^\circ$.

COV_{waGpv} 36°(50%) 60°(50%) #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	11.56	83.36	70.20,93.99	9.12	1,8	2.70
100	16.33	86.88	61.52,99.50	11.21	1,13.33	3.62
125	29.07	89.07	63.14,100	9.20	1,24.66	6.66
150	33.56	88.01	56.18,99.99	10.06	1,40	8.23
175	43.81	88.52	58.76,99.97	9.02	1,45.33	10.47
COV_{waGbc} 36°(50%) 60°(50%) #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	8.44	85.81	71.60,96.59	10.22	1,5.66	2.53
100	12.33	79.34	56.33,94.49	12.08	1,33,14	4.92
125	13.87	80.88	61.50,94.87	9.63	1,33,35.33	10.27
150	18.22	76.04	54.17,97.23	11.81	1,34	9.58
175	24.95	75.21	55.92,26	9.33	1,66,99.33	18.93
COV_{waGpv} 36°(80%) 60°(20%) #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	16	81.97	60.34,100	11.84	1,9	2.83
100	15	88.34	69.60,100	9.00	1,12	3.13
125	14.40	85.16	55.43,100	14.14	1,12	4.17
150	28.67	85.95	57.58,100	10.88	1,16	3.77
175	33.14	85.94	54.34,100	11.85	1,32	6.21
COV_{waGbc} 36°(80%) 60°(20%) #nodes	% nodes with coverset	mean % coverage	min,max % cov- erage/coverset	stddev of % coverage	min,max #coverset/node	mean #coverset/node
75	10.67	83.39	57.20,97.34	14.34	2,12	5.38
100	17	86.29	62.58,99.78	12.44	1,12	3.06
125	41.60	81.41	56.86,95.58	8.36	1,48	11.25
150	47.33	81.92	55.51,100	11.18	1,48	9.39
175	54.86	80.18	51.84,98.24	10.41	1,120	17.20

TABLE II
RESULTS FOR COV_{waGpv} , COV_{waGbc} . MIXED $2\alpha = 36^\circ$ AND $2\alpha = 60^\circ$.

cover set is high, more than 90% in most cases and very close of 100%, with a small standard deviation, which is not the case when point g is not required (COV_{woG}). Therefore, when α is homogeneously small, the simulation results show that the alternate points strategy (COV_{waGpv} and COV_{waGbc}) succeed in constructing cover sets that nevertheless have a high percentage of coverage. The other positive aspect of the alternate point strategy is that more nodes can have cover sets, even when node density is low (75 and 100 sensors cases). COV_{waGbc} provides more cover sets but is a bit less performant in terms of coverage than COV_{waGpv} . When the AoV is homogeneously wider, $2\alpha = 60^\circ$, using only point g can provide both a relatively high proportion of nodes with cover sets and a very high percentage of coverage. However, again, using alternate points provides much more cover sets per sensors and a much higher percentage of nodes with cover sets which has a direct consequence on the network lifetime. In some cases it may be useful to privilege larger cover sets size and sacrificing a bit on the mean percentage of coverage per cover can make sense if scheduling of nodes is more efficient as we will show later in this paper. For instance, the COV_{waGbc} strategy with an AoV of $2\alpha = 60^\circ$ can provide a percentage of coverage of more than 80% with more than 70% of nodes having cover sets.

Table II show an heterogeneous scenario with mixed AoV of 36° and 60° . The first 2 parts used 50% of 36° AoV and 50% of 60° AoV. The last 2 parts used 80% of 36° AoV and 20% of 60° AoV. The COV_{waGpv} strategy, when

compared to the COV_{waGbc} strategy, usually provides a better percentage of coverage for cover sets but produces less cover sets per sensor and a smaller proportion of sensors with cover sets, when there are a majority of narrow AoV sensors. If we look at the COV_{waGbc} strategy in the 3 tables, we can see that this strategy can provide both a high percentage of coverage (always about 80%) and a large proportion of sensors with cover sets. This is highly advantageous as more sensors with cover sets means more sensors that can stay inactive to increase the network lifetime.

B. Occlusions, disambiguation

When there are occlusions it is desirable to have many cover sets in order to cover the same point from multiple different viewpoints. When determining $Co(v)$, it can happen that some cover sets are a subset of other cover sets of larger size. For instance, we could have $PG = \{v_1\}$, $BG = \{v_1, v_3\}$ and $CG = \{v_2, v_3\}$ which would give $Co(v) = \{\{v_1, v_3\}, \{v_1, v_2\}, \{v_1, v_2, v_3\}\}$. Normally, we propose to remove $\{v_1, v_2, v_3\}$ from $Co(v)$ as either $\{v_1, v_3\}$ or $\{v_1, v_2\}$ can cover v 's FoV according to our definition of FoV coverage. However, in case of particularly difficult terrain with many potential occlusions or for disambiguation purposes, it is possible to keep all the cover sets resulting from the cartesian product. In our simulations described in this paper, we do only keep cover sets of minimum size in $Co(v)$.

For the particular case of disambiguation, we introduce a 8m.4m rectangle at random positions in the field. There are 175 sensor nodes. The rectangle has 8 significant points as

depicted in figure 9 and moves at the velocity of 5m/s in a scan line mobility model (left to right). Each time a sensor node covers at least 1 significant point or when the rectangle reaches the right boundary of the field, it appears at another random position. This process starts at time $t = 10s$ and is repeated until the simulation ends. The purpose is to determine how many significant points are covered by the initial sensor v and how many can be covered by using one of v 's cover set. For instance, figure 9 shows a scenario where v 's FoV covers 3 points, the left cover set $(\{v_3, v_1, v_4\})$ covers 5 points while the right cover set $(\{v_3, v_2, v_4\})$ covers 6 points.

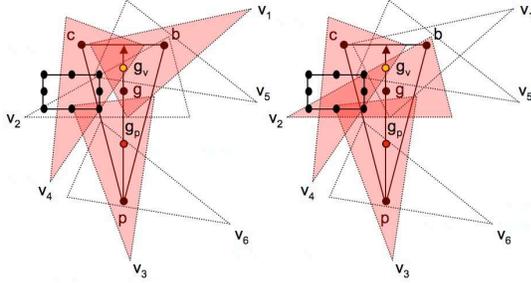


Fig. 9. Rectangle with 8 significant points. Initial sensor v and 2 different cover sets.

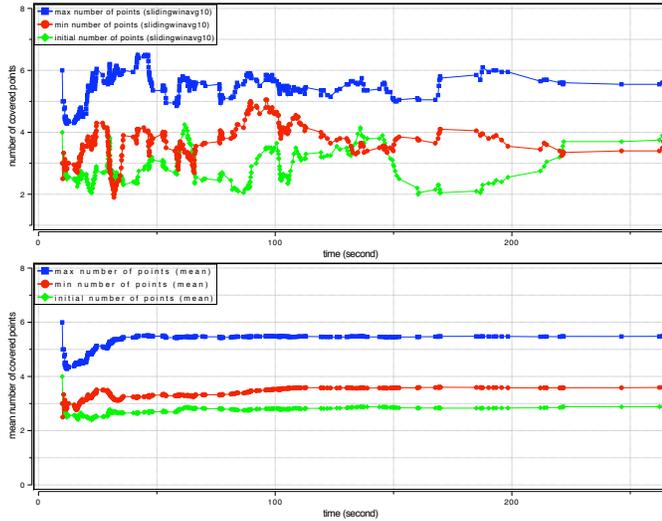


Fig. 10. Number of covered points of an intrusion rectangle. Sliding winavg of 20 (top), mean (bottom).

In the simulations, each time a sensor v covers at least 1 significant point of the intrusion rectangle, it determines how many significant points are covered by each of its cover sets. The minimum and the maximum number of significant points covered by v 's cover sets are recorded along with the number of significant points v was able to cover initially. Figure 10 shows these results. The top part shows the values using a sliding window averaging filter with a batch window of 20 samples. The bottom part shows the evolution of the mean value. We can see that using the node's cover sets always succeeds in identifying more significant points.

C. Network lifetime, stealth time and intrusion detection

In [6] we propose a criticality-based scheduling that uses the number of cover sets to define the sensor's capture rate. Behavior curves based on Bezier curves define a sensor's capture rate. The scheduling mechanism basically operates in 3 phases. The first phase is a broadcast of each sensor's position in the field. Only one message per sensor node is required at transmission. The second phase is a setup phase where each node v constructs its set of cover sets $Co(v)$. The third phase is the scheduling phase where each node decides to be active or in sleep mode. Phases 1 and 2 occur only once at the beginning of the network lifetime, unless mobility is provided, which is not the case in this paper. Our objective is to minimize the number of active nodes while ensuring the maximum coverage area. Compared to the work presented in [6], our contribution in this paper is to compare various cover set construction strategies and their impact on network lifetime, stealth time and intrusion detection.

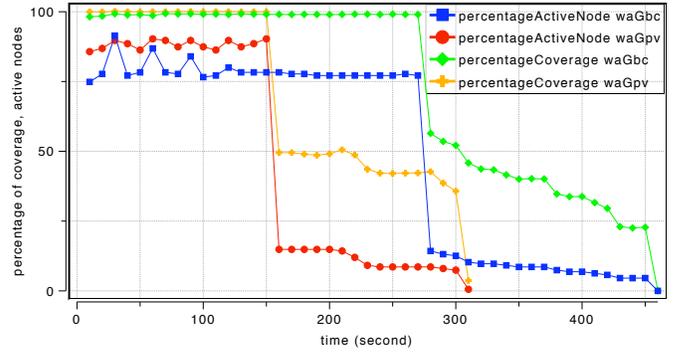


Fig. 11. Percentage of coverage and active nodes with network lifetime.

Figure 11 compares the COV_{waGpv} and the COV_{waGbc} strategies in terms of percentage of coverage, percentage of active nodes and network lifetime. The AoV is homogeneous and set to $2\alpha = 36^\circ$. There are 175 sensor nodes. We can see on the figure that under the COV_{waGbc} strategy, the network lifetime much longer as more nodes have higher number of cover sets. In addition, the percentage of active nodes is smaller while the percentage of coverage is very close to the one provided by the COV_{waGpv} strategy.

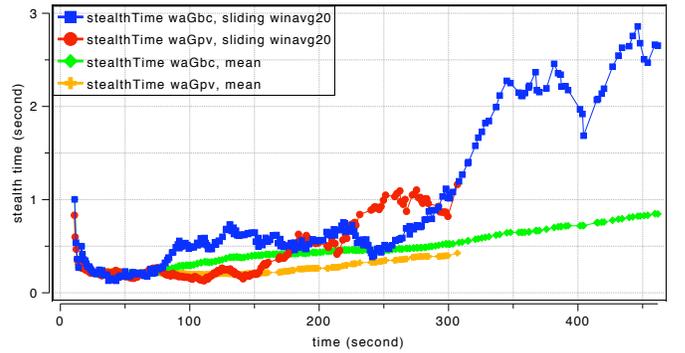


Fig. 12. Mean stealth time.

Figure 12 shows the stealth time with the intrusion rectangle scenario described previously. We plotted 2 versions of the stealth time: one version with a sliding window averaging filter with a batch window of 20 samples and one version with the mean filter. We can see that the COV_{waGpv} strategy provides a smaller stealth time compared to the COV_{waGbc} strategy. This is the main advantage of using the COV_{waGpv} strategy because, as shown in the previous figure, the network lifetime is much shorter.

Figure 13 shows that our sentry node selection strategy succeeds in enabling fast detection of intruders in the field. The left part of the figure shows the sensors' position and their respective number of cover sets in the deployment scenario depicted in figure 4. Those nodes with a high number of cover sets will capture faster according to the scheduling mechanism described in [6]. The right part of the figure shows the number of intrusions detected by each node. The bigger the dot, the higher the number of detected intrusions by that node is. We can clearly see that there is a strong relation between nodes with high number of cover sets and those that have been able to detect the intrusions.

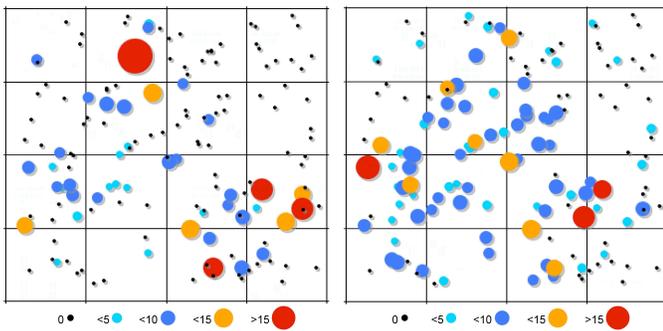


Fig. 13. Node's cover set size and node's detected intrusion number.

IV. CONCLUSIONS

Determining cover sets to define the redundancy level for scheduling and increasing the network lifetime is of prime importance for mission-critical 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 WWSN. We presented a model to find subsets of nodes that cover the FoV area of a given node and various cover set construction strategies. Our contribution in this paper is to evaluate the performance of the various strategies through simulations. We show that some strategies perform better than others depending on the angle of view and whether the angle of view are homogeneous or not. In general, using alternate points is better. Simulation results shows the performance of the strategies in terms of percentage of coverage, network lifetime, intrusion stealth time and number of intrusion detection. Depending on the focus of the application, it is possible to choose a strategy to reduce the stealth time or to increase the network lifetime. However, we showed that the COV_{waGbc}

strategy is the most generic strategy that provides small stealth time and longer network lifetime.

ACKNOWLEDGMENT

This work is supported by the PHC Tassili project 09MDU784.

REFERENCES

- [1] T. He et al, "Energy-efficient surveillance system using wireless sensor networks," in *ACM MobiSys*, 2004.
- [2] 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.
- [3] O. Dousse, C. Tavarularis, and P. Thiran, "Delay of intrusion detection in wireless sensor networks," in *ACM MobiHoc*, 2006.
- [4] Y. Zhu and L. M. Ni, "Probabilistic approach to provisioning guaranteed qos for distributed event detection," in *IEEE INFOCOM*, 2008.
- [5] Crossbow, "http://www.xbow.com," accessed 3/05/2010.
- [6] A. Makhoul, R. Saadi, and C. Pham, "Risk management in intrusion detection applications with wireless video sensor networks," in *IEEE WCNC*, 2010.
- [7] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *ACM SenSys*, 2003.
- [8] E. H. Gupta, S. R. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query," in *ACM MobiHoc*, 2003.
- [9] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Connected sensor cover: Self-organization of sensor networks for efficient query," in *ACM SenSys*, 2003.
- [10] S. Shakkottai, R. Srikant, and N. B. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," in *IEEE INFOCOM*, 2003.
- [11] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," *IEEE INFOCOM*, 2005.
- [12] 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*, 2007.
- [13] 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, 2008.
- [14] 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.
- [15] J. Bahi, A. Makhoul, and A. Mostefaoui, "Improving lifetime and coverage through a mobile beacon for high density sensor networks," *SENSORCOMM08*, 2008.
- [16] W. Mo, D. Qiao, and Z. Wang, "Mostly-sleeping wireless sensor networks: Connectivity, k-coverage, and α -lifetime," *the Allerton Conference, UIUC*, 2005.
- [17] 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.
- [18] J. Wang, C. Niu, , and R. Shen, "Randomized approach for target coverage scheduling in directional sensor network," *LNCS - Springer*, pp. 379-390, 2007.
- [19] M. X. Cheng, L. Ruan, and W. Wu, "Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks," in *IEEE INFOCOM*, 2005.
- [20] Y. Cai, W. Lou, M. Li, and X.-Y. Li, "Target-oriented scheduling in directional sensor networks," *26th IEEE INFOCOM*, 2007.
- [21] 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.
- [22] D. Tao, H. Ma, and L. Liu, "Coverage-enhancing algorithm for directional sensor networks," *LNCS - Springer*, pp. 256-267, 2006.
- [23] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," *1st ACM WSN*, 2002.
- [24] S. Soro and W. Heinzelman, "On the coverage problem in video-based wireless sensor networks," in *Broadband Networks'05*, 2005.
- [25] C. Moreno, "http://www.mochima.com/articles/cuj_geo_metry_article/cuj_geometry_article.html," accessed 3/05/2010.