Semi-Flocking Algorithm for Motion Control of Mobile Sensors in Large-Scale Surveillance Systems

Samaneh Hosseini Semnani and Otman A. Basir

***Abstract*—The ability of sensors to self-organize is an important asset in surveillance sensor networks. Self-organize implies self-control at the sensor level and coordination at the network level. Biologically inspired approaches have recently gained significant attention as a tool to address the issue of sensor control and coordination in sensor networks. These approaches are exemplified by the two well-known algorithms, namely, the Flocking algorithm and the Anti-Flocking algorithm. Generally speaking, although these two biologically inspired algorithms have demonstrated promising performance, they expose deficiencies when it comes to their ability to maintain simultaneous robust dynamic area coverage and target coverage. These two coverage performance objectives are inherently conflicting. This paper presents Semi-Flocking, a biologically inspired algorithm that benefits from key characteristics of both the Flocking and Anti-Flocking algorithms. The Semi-Flocking algorithm approaches the problem by assigning a small flock of sensors to each target, while at the same time leaving some sensors free to explore the environment. This allows the algorithm to strike balance between robust area coverage and target coverage. Such balance is facilitated via flock-sensor coordination. The performance of the proposed Semi-Flocking algorithm is examined and compared with other two flocking-based algorithms once using randomly moving targets and once using a standard walking pedestrian dataset. The results of both experiments show that the Semi-Flocking algorithm outperforms both the Flocking algorithm and the Anti-Flocking algorithm with respect to the area of coverage and the target coverage objectives. Furthermore, the results show that the proposed algorithm demonstrates shorter target detection time and fewer undetected targets than the other two flocking-based algorithms.**

***Index Terms*—Biologically inspired computing, distributed control, mobility control, sensor coordination, sensor networks, surveillance systems.**

I. Introduction

SENSOR networks is an emerging field of study that is expected to touch many aspects of our life. Research in this area was originally motivated by military applications. Afterward sensor networks have demonstrated tremendous promise in many other applications such as infrastructure security, environment and habitat monitoring, industrial sensing, traffic control, and surveillance applications [21]. Recent years have witnessed exponential growth in this field, and this trend is expected to intensify as Sensor Networks advance in capabilities.

Sensor Networks have demonstrated noticeable success in mobile surveillance applications [1]–[4], showing advanced capabilities to self-organize, and to cooperate and coordinate their activities to collect information about targets and events in a given volume of interest (VOI). The information collected by the sensors is often fused to obtain a complete picture of the environment and assess situations of interest. Due to communication and energy restrictions, centralized data fusion algorithms are not efficient and there is a need for distributed algorithms that restrict the communication between neighbors [22]. The ability to self-organize constitutes an indispensable attribute in surveillance applications where target mobility increases surveillance complexity. In this case, sensor mobility comes handy to enable the network to achieve dynamic area coverage and robust target detection.

An important challenge in self-organizing surveillance systems is the control and coordination of sensor mobility. This problem concerns the optimal movement of a set of mobile sensors to achieve maximum area and/or target coverage [5], maximum radio coverage between the sensors [6], [7], or improved target coverage over maneuverable targets [8], etc. This paper addresses the issue of sensor control and coordination for maximum area and target coverage.

The Flocking Algorithm is one of the approaches recently reported in the literature that address the issue of sensor control and coordination in sensor networks. This algorithm has attracted significant interest in recent years in the field of mobility control [9]–[15]. Flocking is a biologically inspired behavior that embodies a form of cooperative behavior of a large number of autonomous interacting agents to achieve a coordinated group behavior. Group movements of birds, fishes, insects, and bacteria are examples of the flocking behavior in nature. To conceive flocking behavior, each agent follows a set of flocking rules and maintains some sort of communication with its neighboring agents. Self-organization and local communication requirements of the flocking process provide an inspiring behavior in the management of sensors in mobile sensor networks.

This paper introduces the Semi-Flocking algorithm, a modified version of the Flocking algorithm [16]. Two other flocking-based algorithms are discussed and used as benchmarks to study the performance of the proposed Semi-flocking

Manuscript received July 19, 2013; revised March 26, 2014; accepted May 19, 2014. Date of publication July 1, 2014; date of current version December 15, 2014. This work was supported by the Ontario Research Fund-Research Excellence Program through the project “MUltimodal-SurvEillance System for SECurity-RElaTed applications (MUSES SECRET)” funded by the Government of Ontario, Canada. This paper was recommended by Associate Editor S. X. Yang.

The authors are with the Electrical and Computer Engineering Department, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: s29hosse@uwaterloo.ca; obasir@uwaterloo.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2328659

algorithm. These two algorithms are Flocking [16] and Anti-Flocking [12]. The paper demonstrates how the proposed Semi-Flocking algorithm provides a better alternative to conceiving self-organizing capabilities in mobile sensor networks contemplating robust surveillance performance.

The outline of the paper is as follows. Section II discusses two main flocking algorithms (Flocking and Anti-Flocking) for mobility control of sensors in surveillance applications. Section III introduces the concept of Semi-Flocking motion modeling. Section IV presents an experimental study to evaluate the proposed Semi-Flocking algorithm. This section discusses the experimental setup, evaluation parameters, simulation results and analysis. Finally, concluding remarks and recommendations for future direction are given in Section V.

II. Flocking-Based Algorithms for Mobility Control

This section discusses the Flocking and Anti-Flocking algorithms as two flocking-based algorithms that can be used for mobility control of sensors in surveillance systems and applications. Flocking-based algorithms have several advantages that make them suitable for use in sensor management. First, they are completely distributed algorithms; therefore, they are highly compatible with the distributed nature of sensor management in sensor networks. Second, in flocking-based algorithms, each particle needs to communicate only with its neighbors; thus, using flocking-based algorithms for sensor management requires only local communication between sensors. Third, because in flocking-based algorithms, particles apply simple flocking rules, using this type of algorithms for sensor management has low computation overhead for the sensors. In addition, flocking-based algorithms are inspired from nature, and have been shown to behave well in self-organized networks. Furthermore, they are highly flexible and scalable, and thus, they are suitable for large sensor networks. The advantages of flocking-based algorithms and their high compatibility with mobile sensor networks motivate us to focus on them as a useful approach to tackle the sensor management problem in a self-organizing surveillance system. The following assumptions have been made in this paper about the surveillance system.

1) The surveillance system consists of n mobile sensors deployed in a 2-D geographical region with width w and length l.
2) Communication ability: each sensor can communicate with all its neighboring sensors by exchanging messages through a communication network.
3) Sensing ability: each sensor can sense all the targets that are within distance r from its position, and thus the sensing range of each sensor is a circle with radius r centered at the sensor position. Targets that come within this range are always detected.
4) Motion ability: the motion of each sensor is controlled independently but in coordination with the motion of other sensors. Let $q_i,\ p_i \in \mathbb{R}^2$ denote the position and velocity of sensor i, respectively. The motion of sensor i is governed by the following equation:
$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i \end{cases} \qquad \text{where} \quad q_i,\ p_i,\ u_i \in \mathbb{R}^2.$$
5) The surveillance system consists of m mobile targets (n > m) randomly entering and leaving the area of interest (AOI). Let $q_{tj},\ p_{tj} \in \mathbb{R}^2$ denote the position and velocity of target j, respectively. All targets follow the following equation of motion:
$$\begin{cases} \dot{q}_{tj} = p_{tj} \\ \dot{p}_{tj} = u_{tj} \end{cases} \qquad \text{where} \quad q_{tj},\ p_{tj},\ u_{tj} \in \mathbb{R}^2.$$
6) Knowledge of sensors about targets is limited to targets positions and velocities.

A. Flocking Algorithm

Flocking is a process that embodies a form of collective behavior of distributed group of autonomous agents. This behavior is accomplished via simple local interactions, a common group objective, and without any global information. The Flocking algorithm is a biologically inspired process that mimics the collective behavior of birds. This algorithm is based on three main rules [9].

1) *Flock Centering:* Stay close to nearby flock-mates.
2) *Collision Avoidance:* Avoid collision with nearby flock-mates.
3) *Velocity Matching:* Match velocity with nearby flock-mates.

Olfati-Saber [10] proposed a theoretical framework for the design and analysis of a distributed Flocking algorithm. He subsequently applied this algorithm to control the movement of sensors [11], [14], [16]. In this algorithm each sensor applies a control input that consists of three components

$$u_i = f_i^g + f_i^d + f_i^\gamma$$

where f_i^g is a gradient-based component, f_i^d is a velocity consensus component and f_i^γ is navigational feedback due to a group objective.

The control function for each sensor i (u_i) composed of two elements $u_i^\alpha + u_i^\gamma$ ($u_i = u_i^\alpha + u_i^\gamma$) in which

$$u_i^a = \underbrace{\sum_{j\in Ni} \emptyset_\alpha \left(||q_j - q_i||_\sigma\right) n_{ij}}_{\text{Gradient-based term}} + \underbrace{\sum_{j\in Ni} a_{ij}(q)(p_j - p_i)}_{\text{Consensus term}} \tag{1}$$

where, in (1), N_i represents the set of neighbors of sensor i. $\emptyset_\alpha(z)$ is an action function that is defined in [10] as follows:

$$\emptyset_\alpha(z) = \rho_h(z/r_\alpha)\,\emptyset(z - d_\alpha) \tag{2}$$
$$\emptyset(z) = \frac{1}{2}\left[(a+b)\,\sigma_1(z+c) + (a-b)\right] \tag{3}$$

where r_α and d_α are constant parameters of α-lattice. $\sigma_1(z) = z/\sqrt{1+z^2}$; $\emptyset(z)$ is an uneven sigmoidal function with parameters $a, b,\ c$ such that

$$0 < a \le b,\ \ c = |a-b|/\sqrt{4ab}.$$

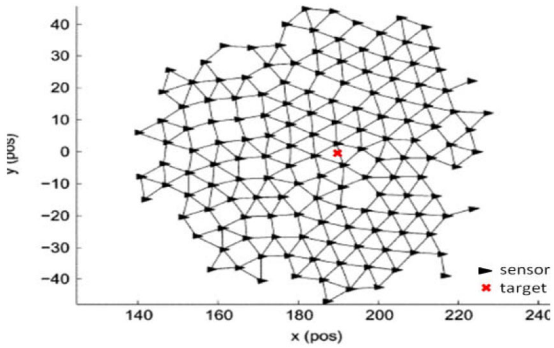

Fig. 1. Flocking for n = 100 agent [10].

$\rho_h(z)$ is a bump function that smoothly varies between 0 and 1 and is defined in (4) [10]

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h) \\ \frac{1}{2}\left[1 + \cos\left(\pi \frac{(z-h)}{(1-h)}\right)\right], & z \in [h, 1] \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The other parameters of (1) are defined as follows: $\|q_j - q_i\|_\alpha$ represents the σ-norm of a vector connecting q_i to q_j defined as [10]. This σ-norm is defined as

$$\|z\|_\sigma = \frac{1}{\varepsilon}\left[\sqrt{1 + \varepsilon\|z\|^2} - 1\right] \quad (5)$$

$$n_{ij} = \nabla\|q_j - q_i\|_\sigma = \frac{q_j - q_i}{\sqrt{1 + \varepsilon\|q_j - q_i\|^2}} \quad (6)$$

where n_{ij} is a vector along the line connecting q_i to q_j, and $\varepsilon \in (0, 1)$ is a fixed parameter of the σ-norm.

Finally, $a_{ij}(q)$, in the consensus term of (1), is an element of the spatial adjacency matrix and is defined as follows [10]:

$$a_{ij}(q) = \rho_h\left(\|q_j - q_i\|_\sigma / r_\alpha\right) \in [0, 1]. \quad (7)$$

The second part of $u_i = u_i^\alpha + u_i^\gamma$, i.e., u_i^γ, is the navigational feedback and is defined based on the sensors' group objective. If the group objective is to track a target at position q_t moving with velocity p_t, u_i^γ is as defined in (8). In this equation c_1 and c_2 are positive constant values

$$u_i^\gamma = f_i^\gamma(q_i, p_i, q_t, p_t) = \underbrace{-c_1(q_i - q_t) - c_2(p_i - p_t)}_{\text{Navigational feedback}}. \quad (8)$$

To use the Flocking algorithm for target tracking, each sensor i needs to apply u_i as an input control. The result of this application is a mass of sensors around the target. Fig. 1 shows the final position of sensors applying this algorithm [10]. It is shown that creating this mass improves the performance of target tracking [12].

Applying the Flocking algorithm in a surveillance application results in the creation of a flock of sensors around the first entered target. This algorithm provides robust target coverage for the first target, but as all the sensors are engaged in tracking the first target, none remains to track other targets (this problem is depicted in Fig. 2). In addition, none of the sensors remains to search the surveillance area to detect new targets or events. This drawback decreases the area coverage and, as a result, increases the detection time for new targets. Therefore, the Flocking algorithm is not only unable to provide acceptable dynamic area coverage, but also cannot obtain robust target coverage in a multitarget system.

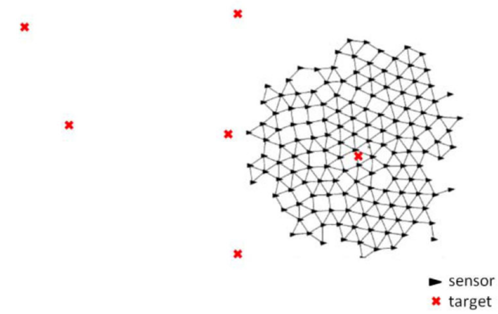

Fig. 2. Flocking algorithm drawbacks in surveillance system.

B. Anti-Flocking Algorithm

The Flocking algorithm is a well-known algorithm with sound mathematical foundation, and noticeable promising results when applied in target tracking. However, this algorithm has limitations with respect to a number of sensor management objectives, especially when applied in surveillance systems. This includes the objective that the sensors of the surveillance system must simultaneously achieve acceptable dynamic area coverage and robust target coverage. This limitation is one of the main motivations for the Anti-Flocking algorithm [12].

The Anti-Flocking algorithm is based on three main rules [12].

1) *De-Centering:* Attempt to move apart from neighbors.
2) *Collision Avoidance:* Stay away from the nearest obstacle that is within a safe distance.
3) *Selfishness:* If neither of the above two situations happens, move on a direction that maximizes one's own gains.

Since the objective of the Anti-Flocking algorithm is to maintain high area coverage, the gain of a mobile sensor's optimal moving direction is set to be the area that has the longest time being unvisited [12]. The result of applying the Anti-Flocking algorithm in a surveillance application is illustrated in Fig. 3. As depicted in this figure, and also by experimental results reported in [12], the Anti-Flocking approach achieves acceptable dynamic area coverage.

Although the Anti-Flocking algorithm achieves acceptable dynamic area coverage, which is essential for surveillance applications, it does not assemble a mass of mobile sensors around each target. This behavior is mostly due to the de-centering rule that scatters sensors over the whole volume of the monitoring area.

The lack of an adequate number of sensors around a target in the Anti-Flocking algorithm greatly increases the network's chance of missing already detected targets. Furthermore, even if the sensor network is to employ a prediction algorithm (e.g., Kalman Filter [17]) to predict future positions of targets, the lack of adequate sensors flocking around the target would result in poor prediction.

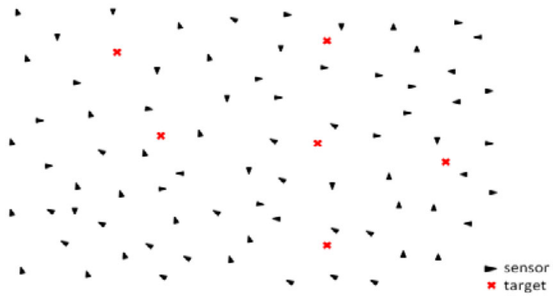

Fig. 3. Anti-Flocking algorithm in surveillance application.

III. Semi-Flocking Approach

Section II described the Flocking algorithm and the Anti-Flocking algorithm as they pertain to sensor management in sensor networks. Although each algorithm possesses intriguing characteristics, each has indispensable drawbacks. The proposed Semi-Flocking algorithm attempts to address these drawbacks. This section introduces this algorithm and shows how self-control and coordination is facilitated so as to ensure high level of area coverage and target coverage.

Analysis of the Flocking algorithm and the Anti-Flocking algorithm in the context of surveillance applications reveals that their drawbacks are expected due to their emphasis on one aspect of performance, and not the two. The Flocking algorithm focuses only on robust target coverage, while the Anti-Flocking algorithm focuses only on dynamic area coverage. As mentioned in Section I, surveillance, by definition, is to achieve both dynamic area coverage and robust target detection. Ignoring either one of these two objects compromises the intended purpose of surveillance. The proposed Semi-Flocking algorithm presented in this paper attempts to fill this gap. The main idea of the Semi-Flocking algorithm is to create small flocks of sensors around each target while still leaving some sensors free to search the surveillance environment for detection of new targets. The Semi-Flocking concept is depicted in Fig. 4.

As illustrated in Fig. 4, while the Semi-Flocking approach obtains acceptable dynamic area coverage, it still creates small flocks of sensors around the position of each target. Although these flocks are smaller than those in the Flocking algorithm, they still can efficiently avoid missing targets. Furthermore, the Semi-Flocking algorithm is able to cover targets (on average) better than the Flocking algorithm. Another interesting feature of the Semi-Flocking approach is its ability to allow sensors to switch between two modes, i.e., tracking mode and searching mode. For example if a target leaves the AOI, then the members of the flock around it join the other flocks to increase the strength of target coverage, or join the searching sensors to increase the chance of detecting new targets.

There are many questions to be answered regarding the Semi-Flocking algorithm. For example, how does each sensor select the best target for tracking (decide which flock to belong to)? How do different flock members avoid collision? How many sensors track the targets and how many of them search the AOI? What happens when a new target comes into the AOI? Which particles should create the flock for a new target? What happens when a target leaves the AOI? What is the job of free sensors? Part A of this section introduces the sensor motion control model that captures these questions.

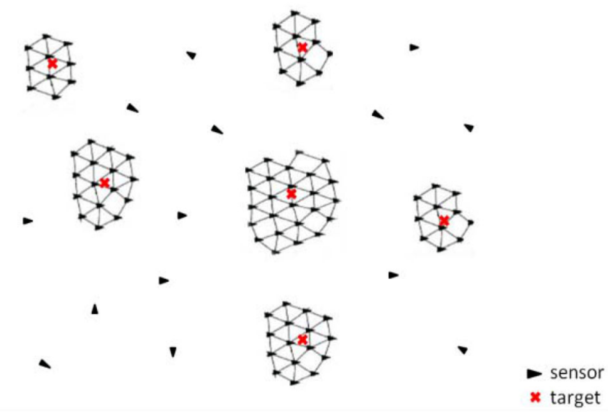

Fig. 4. Semi-Flocking algorithm for sensor management in surveillance application.

A. Semi-Flocking Algorithm

Semi-Flocking behavior is a result of applying simple rules by each sensor. Although these rules are very simple and can be represented as an input vector for each sensor, the result is a complicated group behavior which we call Semi-Flocking behavior. This section describes how this input vector is calculated for Semi-Flocking algorithms and how it is different from the Flocking and Anti-Flocking rules.

In the Flocking algorithm [10], as we stated in Section II, sensors apply a control input vector i.e., $u_i = f_i^g + f_i^d + f_i^\gamma$, in which the first two terms are related to the three Reynold's rules: flock centering, collision avoidance, and velocity matching. The third term (f_i^γ) is a navigational feedback that attracts all the sensors toward one target in the Flocking algorithm. In the Semi-Flocking algorithm, each sensor i applies a control input vector $u_i = f_i^g + f_i^d + f_i^\gamma$, similar to the control vector in the Flocking algorithm except that it makes essential modification in the third term (f_i^γ) to modify the navigational method of the Flocking algorithm. This modification induces the behavior where sensors either get attracted toward one of the surrounding targets, or alternatively emerge free to search the AOI to look for new targets. The sensors are selected to track a target based on two important factors: 1) distance between the sensor and the target and 2) the number of sensors already tracking the target. Each sensor applies Equation (9) to calculate its navigational part

$$\begin{aligned} u_i^\gamma &= f_i^\gamma\left(q_i, p_i, q_{t1}, \ \ldots, \ q_{tm}\right) \\ &= \sum_{j=1}^{m} \varphi\left(q_{tj} - q_i\right) \frac{c_{1j}\left(q_{tj} - q_i\right)}{n_{tj}} \\ &\quad + \sum_{j=1}^{m} \varphi\left(q_{tj} - q_i\right) \frac{c_{2j}\left(p_{tj} - p_i\right)}{n_{tj}} \\ &= \sum_{j=1}^{m} \varphi\left(q_{tj} - q_i\right) \frac{c_{1j}\left(q_{tj} - q_i\right) + c_{2j}\left(p_{tj} - p_i\right)}{n_{tj}} \end{aligned} \tag{9}$$

(1) $u_i^\gamma = 0;$

(2) **for** target j=0 to m **do**

(3) **if** $\|q_{tj} - q_i\|_\sigma \le \theta_j$ **then**

(4) $u_{i,tj}^\gamma = \frac{c_{1j}(q_{tj}-q_i)+c_{2j}(p_{tj}-p_i)}{n_{tj}}$

(5) //where n_{tj} is the number of sensors already tracking target tj

(6) $u_i^\gamma = u_i^\gamma + u_{i,t}^\gamma;$

(7) **end if**

(8) **end for**

(9) **if** $u_i^\gamma == 0$ **then** // the sensor is in searching mode

(10) $q_{w,l}$= center of adjacent areas that has least visited times

(11) $u_i^\gamma = c \times (q_{w,l} - q_i)$

(12) //toward the area that has longest time not being visited

(13) **end if**

Fig. 5. Semi-Flocking navigational-control pseudocode for sensor i.

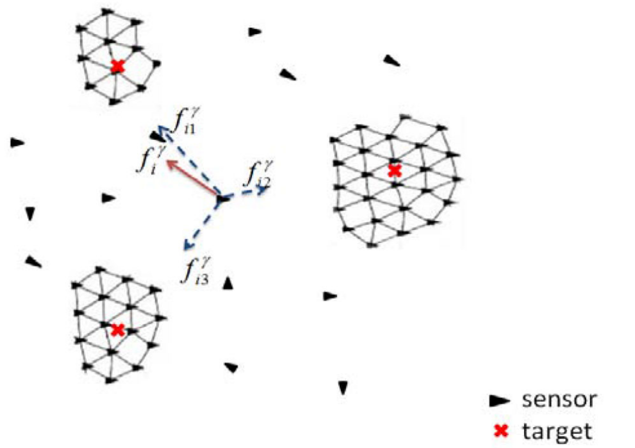

Fig. 6. Multitarget navigational-control in Semi-Flocking algorithm.

where m represents the number of targets, c_{1j}, c_{2j} are positive constant values, $q_{tj} - q_i$ is a vector along a line connecting sensor i to target tj; n_{tj} represents the number of sensors currently tracking target tj, $p_{tj} - p_i$ represents the difference between the velocity of sensor i and target tj and $\varphi\left(q_{tj} - q_i\right)$ is a switching function taking 0–1 values defined by

$$\varphi\left(q_{tj} - q_i\right) = \begin{cases} 1 & q_{tj} - q_i < \theta_j \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 5 illustrates the pseudocode of the navigational control applied by each sensor i in the Semi-Flocking algorithm for computing the size and direction of vector $f_i^\gamma = u_i^\gamma$.

As illustrated in Fig. 5, all the targets whose Euclidean distance from the position of the sensor i is less than a threshold value θ_j (line 3 of Fig. 5) participate in calculation of u_i^γ. The threshold value for each target (θ_j) depends on the number of sensors around that. At the beginning of the algorithm θ_j is set to a default value which depends on the total number of participating sensors and targets. This value is same for all the targets. Later the default value will be corrected for each target based on the number of sensors that are supporting that target. The higher the number of supporters the lower the value of θ_j and vice versa. The contribution of each target tj is calculated by

$$u_{i,tj}^\gamma = \frac{c_{1j}\left(q_{tj} - q_i\right) + c_{2j}\left(p_{tj} - p_i\right)}{n_{tj}}. \tag{10}$$

The size of the vector $u_{i,tj}^\gamma$ is inversely proportional to the size of the flock around each target. The larger the size of the flock, the smaller the navigational vector. The multitarget navigational feedback concept is represented in Fig. 6. In this figure f_i^γ represents the summation of three vectors: f_{i1}^γ ($u_{i,t1}^\gamma$), f_{i2}^γ ($u_{i,t2}^\gamma$), and f_{i3}^γ($u_{i,t3}^\gamma$) that are the control functions applied to a sample sensor.

In the navigational part of the Semi-Flocking algorithm, if no target exists close to sensor i (within distance θ), the sensor searches the AOI to detect incoming targets. In this part, each sensor, instead of doing random search, moves toward the surrounding area that has the longest time not being visited. Assuming the targets have a uniform distribution, it is more likely to detect a target in such an area.

To find least visited adjacent area for each sensor a counter counts the number of times each area is covered by sensors. Suppose that $q_{w,l}$ represents the center of adjacent area that has least visited times, then $q_{w,l} - q_i$ represents a vector along the line connecting current position of the sensor to the center of the least visited area. In the Semi-Flocking algorithm, if a sensor is not selected to track a target, then it will be attracted to the least visited area by calculating the u_i^γ vector using (11). In this equation c is a positive constant value that adjusts size of the vector

$$u_i^\gamma = c \times \left(q_{w,l} - q_i\right). \tag{11}$$

IV. Experiments and Discussion

A. Evaluation Parameters

Sensor management in the context of a surveillance application has a twofold objective. First, it should demonstrate robust target coverage, and second, it should be able to obtain acceptable dynamic area coverage. Based on these requirements we defined four parameters: target coverage (TC), target detection time (TDT), percentage of nondetected targets (PNDT) and cumulative area coverage (AC) as the main evaluation parameters. These parameters are very similar to the ones applied for evaluation of Anti-Flocking algorithm [12] and can be categorized into two groups: the first three parameters evaluate algorithms over target coverage and the last one evaluates their area coverage.

1) *Target Coverage:* Target coverage for a target is the percentage of its lifetime in which it is covered by at least k sensors. Assume that k is the minimum number of sensors required for each target's complete coverage. Equation (12) represents the formula for computing target coverage of target i

$$TC_i = \frac{\sum t_i}{lifetime_i} * 100 \tag{12}$$

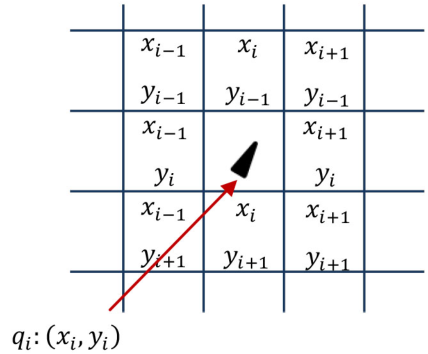

Fig. 7. Moving toward least visited area in navigational-control of Semi-Flocking algorithm.

where i represents a target, each t_i is a fraction of time that target i is covered at least by k sensors, and $lifetime_i$ is the time that target i exists in the AOI.

2) *Target Detection Time:* The detection time of a target is defined as the time passing until the target is covered by at least k sensors.
3) *Percentage of Nondetected Targets:* This parameter represents the percentage of targets that have never been covered by at least k sensors during their lifetime.
4) *Cumulative Area Coverage [18]:* The cumulative area coverage for time interval $[0, \ t)$ in a surveillance system is the fraction of the AOI that is covered at least once by at least one sensor within time interval $[0, \ t)$.

B. Experimental Setup

We implemented a Java version of the Flocking algorithm in the framework presented by Olfati-Saber [10]. We also implemented Semi-Flocking and Anti-Flocking algorithms in the same framework. The following parameters remained fixed through the implementation of all three flocking-based algorithms: $d = 20$, $r = 1.2d$, $\varepsilon = 0.1$ (for σ-norm), $a = b = 5$ for $\varphi(z)$, $h = 0.2$ for the bump function of $\varphi_\alpha(z)$ and the step-size in all simulations is 0.02 s.

1) Random Moving Targets: In this experiment the AOI of the surveillance system is a 1250×665 rectangle. There are 150 mobile sensors in the system, and the detection radius of each sensor is 30. The number of critical targets that need to be tracked by the sensors varied from 0 to 9 to show low to high density problems. For each number of targets, 10 random instances were generated. The reported results are the average over these instances. All the targets are mobile and have a constant speed. In all the instances, the initial position, entrance time, and lifetime of each target were selected randomly by a uniform distribution. The initial position and velocity of all sensors are also selected using a uniform random distribution. We used the same random seeds to generate the same instances in the three algorithms: Flocking, Anti-Flocking and Semi-Flocking. For each instance, we continued the monitoring time for 360 s (6 min). After entering, a target moves randomly around until the end of its lifetime. We assume a target is covered if it is in the field of view of at least three sensors ($k = 3$). This assumption matches with the requirement of many applications of sensor networks [19], [20].

Fig. 8. Sample frame from the walking pedestrian dataset [23].

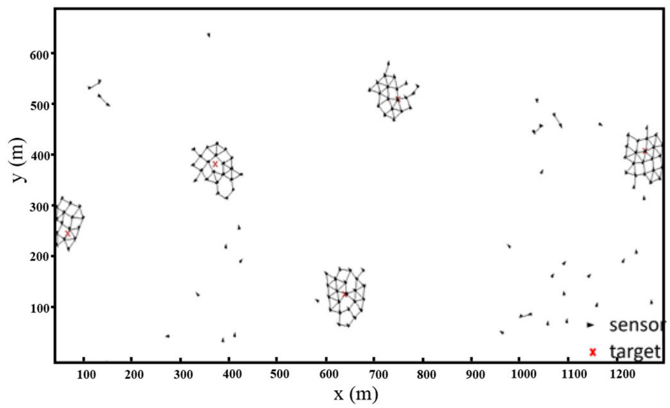

Fig. 9. Snapshot of executing Semi-Flocking algorithm by 150 sensors.

2) Walking Pedestrian Dataset: In the second experiment we used a standard dataset which is collected from digital video sequences of actual walking pedestrians in a busy scenario from a bird eye view. The scene used for data collection was filmed from the 4th floor of a hotel in Zurich in 2009 [23]. This dataset contains around 18061 position and velocity observations for 420 individuals which are manually annotated. Fig. 8 represents a sample frame of this dataset.

C. Simulation Results and Analysis

1) Random Moving Targets: Fig. 9 shows a snapshot of the proximity structure for 150 sensors applying Semi-Flocking algorithm after a few seconds of starting the algorithm. The sensors are supposed to track five mobile targets in this example. As this figure demonstrates a small flock is formed and maintained around each target. At the same time there are some free sensors exploring the AOI. The observations are in close agreement with our expectation of the behavior of the Semi-Flocking algorithm depicted in Fig. 4.

The result of evaluation of Semi-Flocking algorithm and its comparison with Flocking and Anti-Flocking algorithms with

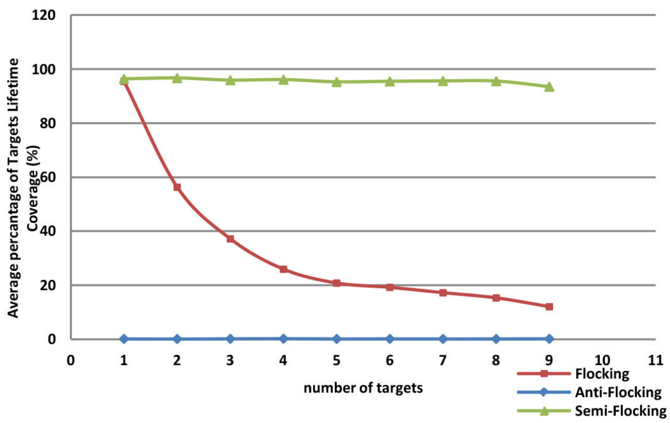

Fig. 10. Average percentage target coverage (TC) in three flocking-based algorithms for random moving targets.

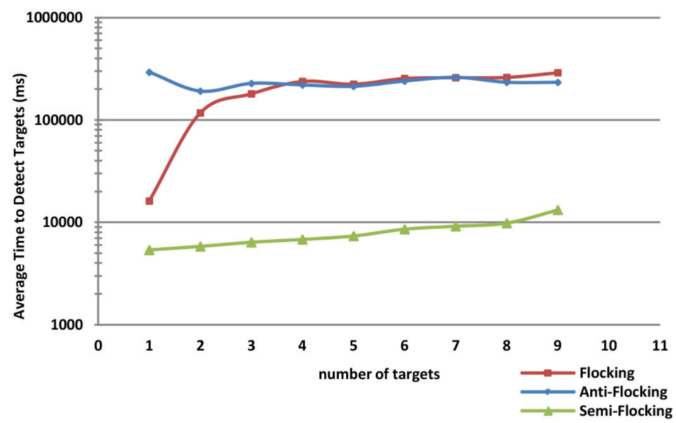

Fig. 11. Average target detection time (TDT) in three flocking-based algorithms for random moving targets.

respect to the parameters introduced in Section IV-A are as follows.

a) Target coverage: Fig. 10 shows the average of 10 runs of target coverage for three flocking-based approaches. Each point in the graph represents the average TC over the number of targets. For example if the number of targets is 2, then the diagram shows the value of $(TC_1 + TC_2)/2$.

As Fig. 10 shows, the Semi-Flocking algorithm demonstrates higher target coverage than the other two algorithms for all numbers of targets, and it is the most interesting aspect of this algorithm. The Semi-Flocking algorithm creates smaller flocks than Flocking algorithm, but it forms a flock around each target. If the number of targets increases, the size of flock around each target decreases automatically. This reduction slightly increases the chance of missing a target and, as a result, average target coverage decreases by increasing the number of targets.

The Flocking algorithm works well for a small number of targets, but as the number of targets increases the average target coverage drops rapidly. This behavior is due to creation of a flock (containing all the sensors) around the first target by Flocking algorithm. Therefore, its target coverage for one target is perfect. However, it does not create a flock around the other targets (Fig. 3. represents this problem). Although other targets may be covered, by chance, when they are placed inside the flock that is tracking the first target, there is no plan in this algorithm for their coverage. The result is a high value for TC_1 and low values for TC_2 to TC_{10}. Thus, for the Flocking algorithm the average value of TC decreases rapidly by increasing the number of targets.

The Anti-Flocking algorithm does not demonstrate acceptable target coverage and this matter is not relevant to the number of targets. This behavior can be explained by highlighting the fact that this algorithm does not create any flock around targets. This behavior decreases the chance of a target being covered by at least k sensors in its lifetime; and this chance does not change by increasing the number of targets.

b) Target detection time: Fig. 11 shows the average of 10 runs of TDT for three flocking-based approaches. Each point in the diagram represents the average of TDT over the number of targets.

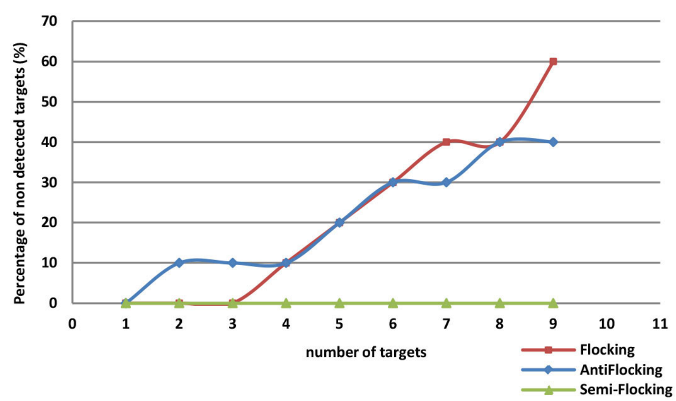

Fig. 12. Percentage of nondetected targets (PNDT) in three flocking-based algorithms for random moving targets.

As Fig. 11 illustrates, Semi-Flocking shows suitable results. Flocking algorithm works well for the small number of targets but its results drop rapidly by increasing the number of targets. Anti-Flocking algorithm fails to demonstrate acceptable average TDT for all the cases. The results are a reflection of the flocking behavior of these algorithms. The Flocking algorithm has the best TDT for the first target, but since it has no plan for detecting other targets, the TDT value for others is high. Thus, by increasing the number of targets, the average of TDT increases rapidly. The Semi-Flocking algorithm demonstrates the best results. The results are excellent when there are few targets in the surveillance area. However, as the number of targets increases, the number of free sensors that search the AOI for detection of new targets decreases slightly. As a result, the TDT for next targets increases and, consequently, the average TDT increases a little. The Anti-Flocking algorithm has a high and almost constant TDT for all numbers of targets. Because this algorithm does not create a flock around each target and it takes a long time for k sensors to be placed around each target simultaneously and eventually be able to detect the target.

c) Number of nondetected targets: Fig. 12 shows the average PNDTs in three flocking-based approaches. Each point in the diagram represents the average of PNDT over the number of targets.

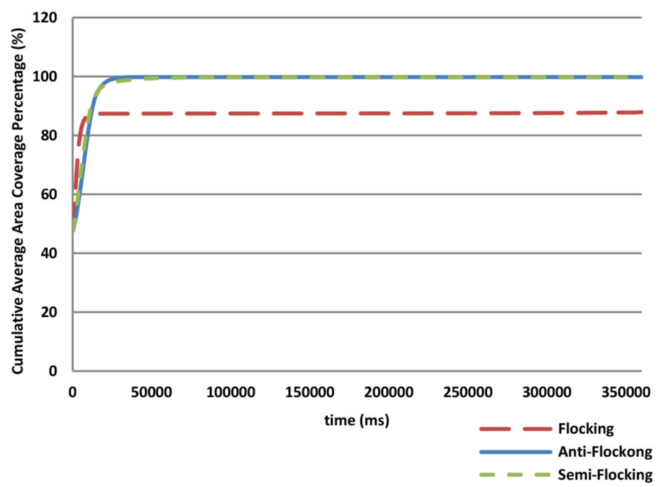

Fig. 13. Cumulative area coverage (AC) in time interval [0, 360000) (ms) in three flocking-based algorithms for random moving targets.

As illustrated in Fig. 12, Semi-Flocking miss less than 5% of targets on average for all numbers of targets. The Flocking and Anti-Flocking algorithms miss more targets. The same reasons presented to explain TDT results (Fig. 11) can explain PNDT results too, since both are related to the target detection parameter.

d) Cumulative area coverage: Fig. 13 shows the average of 10 runs of area coverage for three flocking-based approaches in a time interval [0, 360000) milliseconds.

As shown in this figure, the cumulative area coverage increases over time for all three algorithms. However, the area coverage of the Anti-Flocking algorithm is higher than that of the two other algorithms most of the times. This occurs because the Anti-Flocking algorithm is almost a search algorithm and only emphasizes on increasing area coverage. However, the Semi-Flocking algorithm aims to strike a balance between robust area coverage and target coverage. Furthermore, as Fig. 13 shows, the result of Semi-Flocking algorithm, in area coverage, are extremely close to the results of Anti-Flocking algorithm. The Flocking algorithm represents lower area-coverage results. First, because it applies all the sensors for creation of a flock and does not let any of them free to explore the AOI. Second, it creates a big flock, in which most of the sensors revisit areas previously observed by their front sensors.

2) Walking Pedestrian Dataset: The results obtained from this part of the test are more reliable because this experiment is conducted over real data so the speed, position and the number of pedestrians (targets) in the scene varies more realistically. Table I illustrates the result of applying three flocking-based algorithms on the walking pedestrian dataset [23]. This results include: Average percentage of target coverage (TC), Average TDT, and PNDT. As represented in this table, Semi-Flocking algorithm shows highest percentage of target coverage, lowest TDT and fewer nondetected targets. These results soundly confirm previous results on random moving targets.

Fig. 14 shows the result of area coverage for three flocking-based approaches. As this figure illustrates, Anti-Flocking shows the best results, the results of Semi-Flocking is very close to Anti-Flocking and the Flocking algorithm has the worst performance over this parameter. These results are very similar to the results obtained for the random walking target data as represented in Fig. 13.

TABLE I
RESULTS OF WALKING PEDESTRIANS DATASET

algorithm / parameter	**Semi-Flocking**	**Flocking**	**Anti-Flocking**
Average percentage of target coverage(TC)	85.65755406	18.43532294	1.916780043
Average target detection time (*TDT*)	8317.737789 ms	104751.5244 ms	17466.37275 ms
Percentage of non-detected targets (PNDT)	0	41.64524422	18.76606684

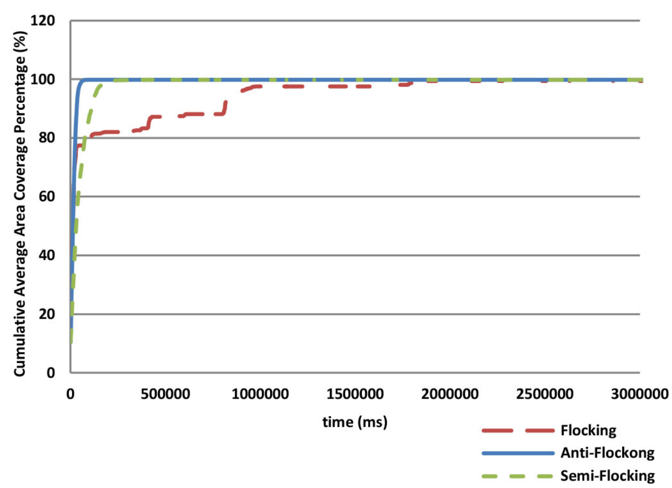

Fig. 14. Cumulative area coverage (AC) in time interval [0, 3000000) (ms) in three flocking-based algorithms for walking pedestrian dataset.

Summarizing the results of the three flocking-based algorithms over the four evaluated parameters shows that, in target coverage, the Semi-Flocking algorithm demonstrates unrivaled results, especially when the number of targets increases. In the area coverage, the results of the Semi-Flocking algorithm are very close to the results of the Anti-Flocking algorithm. Considering the objective function of surveillance systems which is optimizing both target and area coverage, we can say the Semi-Flocking algorithm is the most suitable algorithm among flocking-based algorithms for management of sensors in this application.

V. CONCLUSION

This paper introduced the Semi-Flocking algorithm, an approach for controlling the movement of mobile sensors in surveillance applications. The Semi-Flocking algorithm combines features of Flocking and Anti-Flocking algorithms, and so inhabits a position between these two extremes. In the next step, Flocking, Anti-Flocking and Semi-Flocking algorithms were examined as guidance strategies once for a set of mobile sensors tracking randomly moving targets in the surveillance system and once for a standard walking pedestrian dataset. It has been found that the target coverage, detection effectiveness and area coverage of these mobile sensors vary with the mobility-control algorithm used. The proposed Semi-Flocking algorithm exhibits outstanding performance in meeting the objectives of surveillance application in both test cases.

As a future work, we will consider decreasing the chance of missing already detected targets in the Semi-Flocking algorithm (especially in problems with a high number of targets), by applying techniques that predict the next positions of targets and then guide small flocks of sensors toward such positions.

References

[1] Y. Tseng, Y. Wang, K. Cheng, and Y. Hsieh, "iMouse: An integrated mobile surveillance and wireless sensor system," *IEEE Computer*, vol. 40, no. 6, pp. 60–66, Jun. 2007.

[2] T. He *et al.*, "Energy-efficient surveillance system using wireless sensor networks," in *Proc. 2nd Int. Conf. Mobile Syst. Appl. Services*, Boston, MA, USA, 2004, pp. 270–283.

[3] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "SensEye: A multi-tier camera sensor network," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, Singapore, 2005, pp. 229–238.

[4] F. Dressler, *Self-Organization in Sensor and Actor Networks*. Chichester, U.K.: Wiley, 2007.

[5] L. Hodge and M. Kamel, "An agent-based approach to multi-sensor coordination," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 4, pp. 648–661, Sep. 2003.

[6] M. Batalin and G. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *Proc. 6th Int. Conf. Distributed Autonomous Robot. Syst. (DSRS)*, 2002.

[7] A. Howard, M. Mataric, and G. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. DARS*, 2002.

[8] S. Li, C. Xu, W. Pan, and Y. Pan, "Sensor deployment optimization for detecting maneuvering targets," in *Proc. 7th Int. Conf. Information Fusion (FUSION)*, 2005, pp. 1629–1635.

[9] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *ACM Comput. Graph.*, vol. 21, no. 4, pp. 25–34, 1987.

[10] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[11] R. Olfati-Saber and P. Jalalkamali, "Coupled distributed estimation and control for mobile sensor networks," *IEEE Trans. Autom. Control*, vol. 57, no. 10, pp. 2609–2614, Sep. 2012.

[12] Y. Miao, A. M. Khamis, and M. S. Kamel, "Applying anti-flocking model in mobile surveillance systems," in *Proc. AIS*, 2010, Povoa de Varzim, Portugal, pp. 1–6.

[13] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. 46th IEEE Conf. Decision Control*, New Orleans, LA, USA, 2007, pp. 5492–5498.

[14] P. Jalalkamali and R. Olfati-Saber, "Information-driven self-deployment and dynamic sensor coverage for mobile sensor networks," in *Proc. Amer. Control Conf. (ACC)*, Montreal, QC, Canada, 2012, pp. 4933–4938.

[15] H. M. La and W. Sheng, "Dynamic target tracking and observing in a mobile sensor network," *Robot. Autonom. Syst.*, vol. 60, no. 7, pp. 996–1009, Jul. 2012.

[16] R. Olfati-Saber, "Distributed tracking for mobile sensor networks with information-driven mobility," in *Proc. Amer. Control Conf.*, New York, NY, USA, 2007.

[17] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. 46th IEEE Conf. Decision Control*, New Orleans, LA, USA, 2007, pp. 5492–5498.

[18] B. Liu, P. Brass, and O. Dousse, "Mobility improves coverage of sensor networks," in *Proc. MobiHoc*, Urbana-Champaign, IL, USA, 2005.

[19] B. Krishnamachari, S. B. Wicker, and R. Bejar, "Phase transition phenomena in wireless ad hoc networks," in *Proc. Global Telecommunications Conf. (GLOBECOM)*, vol. 5. 2001, pp. 2921–2925.

[20] L. K. Soh and C. Tsatsoulis, "Reflective negotiating agents for real-time multisensor target tracking," *Int. Joint Conf. Artif. Intell.*, vol. 17, no. 1, pp. 1121–1127, 2001.

[21] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[22] M. Vinyals, J. A. Rodriguez-Aguilar, and J. Cerquides, "A survey on sensor networks from a multiagent perspective, " *Comput. J.*, vol. 54, no. 3, pp. 455–470, 2011.

[23] (2014, Mar. 19). *BIWI Walking Pedestrians Dataset* [Online]. Available: http://www.vision.ee.ethz.ch/datasets

Authors' photographs and biographies not available at the time of publication.