A single-target tracking algorithm in controlled mobility sensor networks

F. Mourad, H. Chehade, H. Snoussi, F. Yalaoui, L. Amodeo
ICD (UMR STMR CNRS 6279)
Université de Technologie de Troyes
12, rue Marie Curie, 10010, France
Email: {mouradfa, chehadeh, snoussi, yalaoui,amodeo}@utt.fr

C. Richard
Laboratoire FIZEAU (UMR CNRS 6525)
Université de Nice Sophia-Antipolis
Parc de Valrose, 06108 Nice, France
Email: cedric.richard@unice.fr

***Abstract*—This paper deals with the problem of single target tracking in controlled mobility sensor networks. It thus proposes an original strategy to manage the mobility of sensor nodes, in order to improve the estimation of the positions of the target. The proposed method consists of estimating the actual position of the target, then it uses position estimates to predict a region where the target is assumed to fall in the next time-step. A structure of triangle is then defined, aiming at covering at best the predicted area. Each mobile sensor is then assigned a position at the vertices of the structure, in the way to minimize the total traveled distance by the nodes. One constraint for this technique is to maintain the total coverage of the network. For this reason, we propose to use a hybrid network, including static nodes, as well, to insure continuously the coverage of the network. Simulation results corroborate the efficiency of the proposed method compared to the target tracking methods considered for networks with static nodes.**

***Index Terms*—Controlled mobility, interval analysis, network coverage, optimization, state estimation, target tracking.**

I. INTRODUCTION

In recent years, advances in miniaturization, low-power circuit design and efficient wireless communication have produced a new technological innovation, the Mobile Sensor Networks (MSN). These networks are composed of a large number of mobile devices having sensing, processing and communication capabilities [1], [2]. The mobility of these devices could be either passive or controlled. In the case of passive mobility, sensors are moved in an uncontrollable manner in response to external forces, whereas in the case of controlled mobility, sensors are moved in response to internal or external commands. In this paper, we consider the case of mobile sensors having a controlled mobility. In such a case, one could take advantage of the mobility of the nodes to improve the accuracy of the sensed data in the network.

One interesting application of MSN is target tracking. It consists of estimating instantly the position of a moving target. It is of great importance in surveillance and security especially in military applications. This problem has been mainly considered for networks having static nodes [3], [4]. However, when sensors are able to move, it is important to take advantage of their mobility in order to improve the position estimation. Different techniques have been proposed to manage the mobility of the nodes [5], [6]. These techniques have mainly focused on improving the area coverage or increasing the lifetime of the network, etc. A few methods have been developed for target tracking in MSN [7], [8]. For instance, researchers in [7] have proposed a mobility management scheme based on the Bayesian estimation theory. Note that many assumptions are made in this method such that both the target and the sensor nodes are supposed having constant velocities.

This contribution focuses on target tracking in MSN where nodes have a controlled mobility. A novel strategy for managing sensors mobility is thus proposed. It aims at moving the sensors to improve the tracking of a single target. The main constraint of this method is to minimize the energy consumption in order to maximize the lifetime of the network. At each time step, the proposed method consists of i) estimating the current position of the target, ii) predicting the next-step position of the target, and then iii) computing a set of new locations to be taken by the nodes in the way to improve the estimation process. The proposed strategy should also maintain the total coverage of the network. Beside the mobile nodes, we propose to use static nodes aiming at ensuring a continuous coverage of the network independently of the movement of the mobile ones.

The rest of the paper is organized as follows. In Section 2, we describe the method. Section 3 introduces the different steps of the algorithm. Simulation results are given in Section 4. Section 5 concludes the paper.

II. DESCRIPTION OF THE METHOD

The aim of the method is to manage the mobility of the nodes at each time-step in the way to improve the estimation of the next-step position of the target. The main constraint for this method is to minimize the energy consumption while moving the sensors. Another constraint consists of maintaining the total coverage of the surveillance area, in the way to be able to detect any intrusion. For this reason, we propose to use a hybrid network, that is, a network composed of mobile and static sensors. While mobile sensors follow the target at each time-step, static sensors are uniformly deployed in the way to maintain the total coverage of the network. Assume that the sensing ranges of static sensors are circular having r_s as radii. Then, in order to cover the total area with the static sensing disks, one is able to use the disk packing theory [9].

A simpler solution consists of using the ($\sqrt{2}r_s$)-side squares contained in the sensing disks to cover the area. Practically, if the surveillance area is a $\mathcal{W}_1 \times \mathcal{W}_2$ square area, then the number of static sensors needed to cover the whole area is equal to $K_s = K_{s,1} * K_{s,2}$, where

$$K_{s,1} = \text{Inth}(\frac{\mathcal{W}_1}{\sqrt{2}r_s}) \text{ and } K_{s,2} = \text{Inth}(\frac{\mathcal{W}_2}{\sqrt{2}r_s}), \quad (1)$$

with Inth(x) yielding the smallest integer equal or higher than x. The positions of the static sensors would then be given by the combinations of the following coordinates,

$$\begin{cases} A_{(s,1),p} = o_1 + \frac{\sqrt{2}r_s}{2} + (p-1)\sqrt{2}r_s, & p \in \{1, ..., K_{s,1}\}, \\ A_{(s,2),q} = o_2 + \frac{\sqrt{2}r_s}{2} + (q-1)\sqrt{2}r_s, & q \in \{1, ..., K_{s,2}\}, \end{cases} \quad (2)$$

where $o_1 = \frac{\mathcal{W}_1 - \sqrt{2}r_s K_{s,1}}{2}$ and $o_2 = \frac{\mathcal{W}_2 - \sqrt{2}r_s K_{s,2}}{2}$. In other words, each static sensor would have a coordinate vector denoted $\boldsymbol{a}_{s,i}$, equal to a couple of coordinates $(A_{(s,1),p}, A_{(s,2),q})$ with $i = p + (q-1)K_{s,1}$, $i \in \{1, ..., K_s\}$..

Now that the coverage of the network is continuously maintained, the objective of the method consists of moving the mobile sensors, as minimal as possible, in the way to follow the target at each time-step. For these reasons, the method would be composed of the following steps at each iteration:

1) Estimating the current position of the target;
2) Predicting the next-step position of the target using the estimated positions;
3) Computing the following positions of the mobile sensors using the predicted position.

The management of the mobility of the sensors should satisfy the energy consumption minimization constraint.

III. Algorithm

The proposed method is composed of different techniques for estimating and predicting the positions of the target. It also includes an approach for computing the next-step positions of the mobile sensors in the way to cover at best the predicted location of the target. It is worth noting that the proposed method is centralized. In other words, all the sensed data are communicated, via the static nodes, to a central base-station. Using all information, the algorithm computes the positions of the target and the movements of each mobile sensors, and then, the computed information is communicated back to the sensors. In the following, $\boldsymbol{x}(t) = (x_1(t), x_2(t))$ would denote the target position at time t, $\boldsymbol{a}_{m,i}(t) = (a_{m,i,1}(t), a_{m,i,2}(t))$, $i \in \{1, ..., K_m\}$, would denote the mobile sensors coordinates and $\boldsymbol{a}_{s,i} = (a_{s,i,1}, a_{s,i,2})$, $i \in \{1, ..., K_s\}$, would represent static sensors.

A. Estimation of the current position of the target

In order to estimate the position of the target, we propose to use a connectivity-based approach. In other words, all sensors (mobile or static) detecting the target generate a one-bit information, equal to 1. These measurements are then communicated to the base-station. Let $I_m(t)$ and $I_s(t)$ be the sets of indices of the mobile and static sensors detecting the target at time t, respectively. If we assume that the sensing ranges of the mobile sensors are circular having r_m as radii, then the distances separating the target to the mobile sensors detecting it are less than r_m. This is also true for static sensors detecting the target with distances to the target less than r_s. The estimation model would thus be given by the following,

$$\begin{array}{ll} (x_1(t) - a_{m,i,1}(t))^2 + (x_2(t) - a_{m,i,2}(t))^2 \leq r_m^2, & i \in I_m(t), \\ (x_1(t) - a_{s,i,1})^2 + (x_2(t) - a_{s,i,2})^2 \leq r_s^2, & i \in I_s(t). \end{array} \quad (3)$$

In order to resolve this problem, we use the interval theory [10]. For this reason, we briefly recall the basic definitions of interval analysis. A *real interval*, denoted $[x]$, is a closed subset of $\mathbb{R}$ given as follows,

$$[x] = [\underline{x}, \overline{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \overline{x}\}, \quad (4)$$

where $\underline{x}$ and $\overline{x}$ are the lower and upper scalar endpoints of the interval respectively. A multidimensional interval of $\mathbb{R}^n$, also called *box*, is given by the cartesian product of n real intervals, $[\boldsymbol{x}] = [x_1] \times \cdots \times [x_n]$. An interval has a dual nature as sets and real numbers. The interval theory takes advantage of this duality to extend all arithmetic and set operations to intervals.

The key idea of the method consists of considering the target position as a two-dimensional box $[\boldsymbol{x}](t) = [x_1](t) \times [x_2](t)$ [11]. In other words, the proposed method aims at computing the minimal box $[\boldsymbol{x}](t)$ that includes all possible solutions of the problem. In this way, the target position is a rectangular area including the unknown location of the target and all incertitude over its value. In order to resolve the estimation problem, all constraints should be reformulated using intervals as follows,

$$\begin{array}{ll} ([x_1](t) - a_{m,i,1}(t))^2 + ([x_2](t) - a_{m,i,2}(t))^2 \subseteq [0, r_m^2], & i \in I_m(t), \\ ([x_1](t) - a_{s,i,1})^2 + ([x_2](t) - a_{s,i,2})^2 \subseteq [0, r_s^2], & i \in I_s(t). \end{array} \quad (5)$$

The problem is then defined as a constraint satisfaction problem. An initial domain, for instance the whole deployment area, is thus contracted in order to obtain the smallest box including the exact scalar solution. The algorithm used to perform the contraction is called the Waltz contractor [10]. It is a forward-backward algorithm that iterates all constraints without any prior order until no contraction is possible. The computed solution, called *boxed estimate*, is at best the box including the overlapping area of all disk constraints. The punctual estimate would be the center of the boxed estimate of the position of the target. Fig. 1 shows an illustration of the estimation phase.

B. Prediction of the next-step position

The aim of this section is to propose a technique for computing a box where the target might fall in the next time-step. For this reason, we propose to use a second order prediction model, using intervals. Indeed, the considered model uses the boxed estimates at times t, $t-1$ and $t-2$ to compute a predicted position box for time-step $t+1$. The prediction model is formulated as follows,

$$[\widehat{\boldsymbol{x}}](t+1) = [\boldsymbol{x}](t) + \Delta t.[\boldsymbol{v}](t) + \frac{\Delta t^2}{2}.[\gamma](t), \quad (6)$$

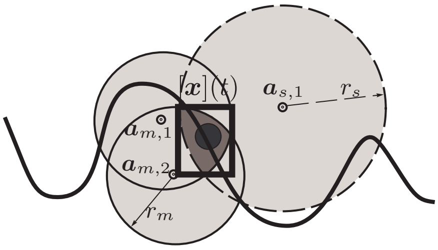

Fig. 1. An illustration of the estimation phase.

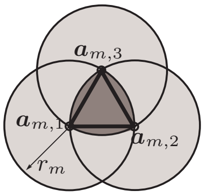

Fig. 2. A triangle of sensors having the side lengths equal to the sensing range.

where $[\widehat{\boldsymbol{x}}](t+1)$ is the predicted position box of the target, Δt is the time period falling between two time-steps, $[\boldsymbol{v}](t) = \frac{[\boldsymbol{x}](t)-[\boldsymbol{x}](t-1)}{\Delta t}$ and $[\gamma](t) = \frac{[\boldsymbol{v}](t)-[\boldsymbol{v}](t-1)}{\Delta t}$. Using intervals, the prediction phase yields a box, called *prediction box*, including the next-step position of the target and the incertitude brought by measurements over its value.

C. Definition of the mobile sensors positions

The goal of the method consists of moving the sensors in an energy-aware manner in order to improve the coverage of the prediction box. For this reason, we propose to use the triangulation principle. In other words, if we consider only three mobile nodes, the triangulation-based idea consists of constructing an equilateral triangle with the sensors. The intersection area of the sensing disks of the mobile nodes should cover the target at the next time-step. Let the triangle sides be equal to the sensing range r_m of the mobile nodes, as shown in Fig. 2. If the target is detected by the three sensors (considering only mobile nodes), then the estimation phase leads to the smallest box covering the intersection area of their sensing disks. The accuracy of the estimation would depend on the sensing range of the mobile sensors. Indeed, with smaller sensing disks, smaller overlapping areas are obtained leading to smaller boxed estimates.

Having more than three sensors, one is able to construct a structure of triangles in the way to cover the whole prediction box. Using triangle structures, every single point of the prediction box could be covered by at least three mobile sensors. An example of a structure of seven mobile sensors is given in Fig. 3. Having r_m as triangle side length, one could compute the required number of mobile sensors depending on the size of the prediction box. Let $K_{m,1}$ be the number of the sensors required in the first down row and let $K_{m,2}$ be

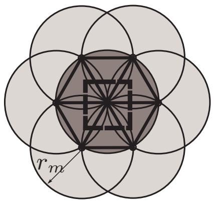

Fig. 3. An example of a structure of seven mobile sensors with the prediction box.

the number of required rows. Having the type of structures of Fig. 3, rows having odd indices are thus composed of $K_{m,1}$ sensors whereas those of even indices have $K_{m,1}+1$ sensors. In order to guarantee a coverage of the whole box even for points close to the borders, $K_{m,1}$ and $K_{m,2}2$ are set as follows,

$$K_{m,1} = \text{Inth}(\frac{\widehat{\mathcal{W}}_1}{r_m}) + 1 \quad \text{and} \quad K_{m,2} = \text{Inth}(\frac{\widehat{\mathcal{W}}_2}{\frac{\sqrt{3}}{2} r_m}) + 1, \quad (7)$$

where $\widehat{\mathcal{W}}_1$ and $\widehat{\mathcal{W}}_2$ are the widths of $[\widehat{x}_1](t+1)$ and $[\widehat{x}_2](t+1)$ respectively, $\frac{\sqrt{3}}{2} r_m$ is the height of the sensor triangle and Inth(x) is the smallest integer equal or greater than x. The total number of mobile sensors to be moved is then equal to $K_m^* = K_{m,1}\frac{K_{m,2}+\delta}{2} + (K_{m,1}+1)\frac{K_{m,2}-\delta}{2}$, where $\delta = 1$ if $K_{m,2}$ is odd and 0 otherwise. The coordinates of the vertices of the structure would thus be defined by combinations of the following coordinates,

$$\begin{cases} A_{(m,1),p} = \begin{cases} \widehat{b}_{o,1} + (p-1) r_m, & \text{if } q \text{ is odd } (1 \le p \le K_{m,1}), \\ \widehat{b}_{e,1} + (p-1) r_m, & \text{if } q \text{ is even } (1 \le p \le K_{m,1}+1), \end{cases} \\ A_{(m,2),q} = \widehat{b}_2 + (q-1)\frac{\sqrt{3}}{2} r_m, \ 1 \le q \le K_{m,2}, \end{cases} \quad (8)$$

where $\widehat{b}_{o,1} = \widehat{\underline{x}}_1 - \frac{K_{m,1} r_m - \widehat{\mathcal{W}}_1}{2}$, $\widehat{b}_{e,1} = \widehat{\underline{x}}_1 - \frac{(K_{m,1}+1) r_m - \widehat{\mathcal{W}}_1}{2}$ and $\widehat{b}_2 = \widehat{\underline{x}}_2 - \frac{K_{m,2}\frac{\sqrt{3}}{2} r_m - \widehat{\mathcal{W}}_2}{2}$. The obtained positions $\boldsymbol{A}_{m,k}$, $k \in \{1, ..., K_m^*\}$, are thus symmetric with respect to the prediction box. These positions should then be assigned to the K_m mobile sensors. However, when $K_m \neq K_m^*$, two cases could be encountered. If $K_m^* < K_m$, we propose to use the closest K_m^* mobile nodes to the center of the prediction box. Otherwise, if $K_m^* > K_m$, we propose to keep the closest K_m positions to the center of the prediction box. In the following, we will assume that K_m^* is the final number of positions, and thus the number of mobile sensors to be moved. These sensors are assumed to be at the positions $\boldsymbol{a}_{m,1}(t), ..., \boldsymbol{a}_{m,K_m^*}(t)$, as well.

Once the positions are defined, the goal of the method is to assign each mobile node one position, in the way to minimize the total distance to be traveled. Practically, each mobile node being at the position $\boldsymbol{a}_{m,i}(t)$ should move to a new position $\boldsymbol{A}_{m,i}$, chosen within the set defined above, while minimizing $\sum_{i=1}^{K_m^*} \|\boldsymbol{a}_{m,i}(t) - \boldsymbol{A}_{m,i}\|$. Using a deterministic (exact) method to perform this assignment could be high-energy consuming especially when the number of considered positions is high. Indeed, the number of all possible solutions is equal to $K_m^*!$. For this reason, the assignment problem is resolved using a metaheuristic algorithm, the Ant Colony Optimization approach (ACO) [12], [13]. ACO is a probabilistic method for solving complex computational problems. Having a set of n variables, to be assigned n values, and an objective function to be minimized, ACO starts with an initial solution, and then it moves towards optimal solutions using an efficient memory-based search technique. Using ACO, one is able to find the optimal assignment in a small computational time, compared to exact methods.

IV. Simulations

In order to evaluate the effectiveness of the proposed method, we consider a target moving over 100-steps trajectory in a $[0, 100m] \times [0, 100m]$ surveillance area. The sensing range r_s of static sensors is set to $10m$. The number of required static nodes is thus equal to $K_{s,1}.K_{s,2}$ where $K_{s,1} = K_{s,2} = \text{Inth}(\frac{100}{10.\sqrt{2}}) = 8$. The sensing range r_m of mobile sensors is set to $5m$. It is worth noting that all simulations are performed on an Intel(R) Core(TM)2 CPU ($2.40GHz$, $1.00GB$ RAM) using MATLAB 7.9.

In order to illustrate the performances of our method, we compare it to a target tracking method developed for static sensor networks. We thus propose an interval-based method performing a similar estimation as our method. The sensors are deployed uniformly for the static method whereas for our method, the mobile nodes are initially deployed in a random manner. 100 sensors are used for both methods. In particular, 64 static sensors and 36 mobile ones are considered for our method. Fig. 4 shows the estimated boxes obtained with both methods. It also shows the target trajectory and the static sensors locations for both methods. It is obvious that moving the nodes leads to more accurate estimates. The average areas of boxes are equal to $15m^2$ and $117m^2$ with our method and the static method respectively. Let the estimation error be the average distance between the centers of the boxed estimates and the exact positions of the target. Then it is equal to $0.92m$ with our method whereas it is equal to $3.33m$ with the static method. The average distance traveled by the nodes is equal to $5m$ per time-step with a target velocity equal to $3.9m$. It is worth noting that the computation time highly increases from $0.0006s$ to $0.38s$ while moving the sensors. This is mainly due to the optimization algorithm.

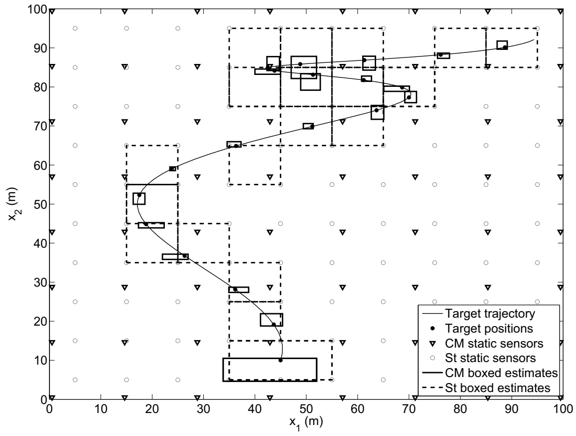

Fig. 4. An illustration of the estimated boxes obtained with both our method (CM) and the static method (St).

V. Conclusion

In this contribution, we proposed a novel technique for target tracking in mobile sensor networks. We thus proposed a strategy for managing the mobility of sensors in order to improve the accuracy of target position estimates. The method consists of estimating the current position of the target and then predicting its following position using a second-order prediction model. A relocation of sensors is then performed in order to optimize the target localization for the following time-step. In order to maintain the total coverage of the network, the proposed approach uses a hybrid sensor network composed of both static and mobile nodes. While mobile nodes are moved to improve the target tracking, static nodes ensure the total coverage of the network. Simulation results illustrate the efficiency of the proposed method compared to algorithms developed for static sensor networks. Future works will handle the problem of multi-target tracking.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[2] D. Zeinalipour-Yazti and P. K. Chrysanthis, "Mobile sensor network data management," in *Encyclopedia of Database Systems*, 2009, pp. 1755–1759.

[3] P. M. Djurić, M. Vemula, and M. F. Bugallo, "Target tracking by particle filtering in binary sensor networks," *IEEE Transactions on Signal Processing*, 2008.

[4] J. Teng, H. Snoussi, and C. Richard, "Decentralized variational filtering for target tracking in binary sensor networks," *IEEE Transactions on Mobile Computing*, 2010.

[5] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, "Controlled sink mobility for prolonging wireless sensor networks lifetime," *Wirel. Netw.*, vol. 14, no. 6, pp. 831–858, 2008.

[6] T. P. Lambrou, C. G. Panayiotou, S. Felici, and B. Beferull, "Exploiting mobility for efficient coverage in sparse wireless sensor networks," in *Wireless Personal Communications*, S. Netherlands, Ed., April 2009.

[7] Y. Zou and K. Chakrabarty, "Distributed mobility management for target tracking in mobile sensor networks," *IEEE Transactions on Mobile Computing*, 2007.

[8] J.-C. Chin, Y. Dong, W.-K. Hon, C. Y. T. Ma, and D. K. Y. Yau, "Detection of intelligent mobile target in a mobile sensor network," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 41–52, February 2010.

[9] M. Locatelli and U. Raber, "Packing equal circles in a square: a deterministic global optimization approach," *Discrete Applied Mathematics*, vol. 122, no. 1-3, pp. 139 – 166, 2002.

[10] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis*. Springer, 2001.

[11] F. Mourad, H. Snoussi, F. Abdallah, and C. Richard, "Anchor-based localization via interval analysis for mobile ad-hoc sensor networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3226–3239, 2009.

[12] M. Dorigo and L. Gambardella, "Ant colonies for the traveling salesman problem," *Biosystems*, 1997.

[13] H. Chehade, L. Amodeo, and F. Yalaoui, "A new efficient hybrid method for selecting machines and sizing buffers in assembly lines," *Journal of Operations and Logistics*, 2009.