

PMT²: A Predictive Mobile Target Tracking Algorithm in Wireless Multimedia Sensor Networks

Ibtissem Boulanouar, Stéphane Lohier, Abderrezak Rachedi, Gilles Roussel

► **To cite this version:**

Ibtissem Boulanouar, Stéphane Lohier, Abderrezak Rachedi, Gilles Roussel. PMT²: A Predictive Mobile Target Tracking Algorithm in Wireless Multimedia Sensor Networks. IEEE ISCC, Jun 2014, Madeira, Portugal. 2014, <10.1109/ISCC.2014.6912601>. <hal-00962255>

HAL Id: hal-00962255

<https://hal-upec-upem.archives-ouvertes.fr/hal-00962255>

Submitted on 24 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PMT^2 : A Predictive Mobile Target Tracking Algorithm in Wireless Multimedia Sensor Networks

Ibtissem Boulanouar, Stéphane Lohier, Abderrezak Rachedi and Gilles Roussel

Université Paris-Est - Laboratoire de l'Institut Gaspard Monge (LIGM)

77454 Marne la Vallée Cedex 2, France

{FirstName. LastName}@univ-mlv.fr

Abstract—In this work, we propose a new Predictive-based Mobile Target Tracking Algorithm for Wireless Multimedia Sensor Networks called PMT^2 . Resource management being a critical feature of this kind of networks, the main aim of PMT^2 is to handle the trade-off between the accuracy of the tracking and the energy conservation. Prediction approach seems to be the best candidate to reach this objective. For this purpose, we introduce an enhanced version of the Extended Kalman Filter combined with a change detection mechanism named CuSum for Cumulative Summary. We also propose a deployment strategy to improve the efficiency of the tracking algorithm. Using simulations, we show the performances of the proposed coupled mechanism in the trajectory prediction and in the reactivity to abrupt direction changes. Moreover, we perform a comparative study between PMT^2 and existing works: 1) BASIC where all the Cameras Sensors are always in active mode; 2) OCNS for Optimal Camera Node Selection, a cluster-based solution with a probabilistic sensor selection; 3) PTA, another predictive solution based on standard Kalman Filter. The obtained results illustrate that PMT^2 improves the quality of tracking by up to 35% compared to existing works, while reducing energy consumption by up to 55%.

I. INTRODUCTION

Recent advances in wireless communication and micro-electronics have allowed the emergence of low-power multimedia sensors. These multimedia sensors can be fitted with small microphones and cameras in order to handle multimedia content. Wireless Multimedia Sensor Networks (WMSN) are a set of tiny interconnected Multimedia Sensors scattered in a region of interest to handle specific tasks. Applications of WMSN are typically target tracking, multimedia surveillance, habitat and health care monitoring [1]. WMSN have specific characteristics such as QoS requirements, and adjustable fields of view.

In this work, we investigate the problem of single mobile target tracking using WMSN. Specifically, we focus on non-cooperative target, where no communication is possible between the target and network infrastructure (e.g. humans, animals). Multimedia target tracking consists in finding target's location in consecutive camera frames. Having the coordinates of the dynamic target, and thus, the record of displacement can be very useful for many security applications. However, performing target tracking on WMSN has some limitations: wireless links are not reliable and image capture and transmission are greedy processes in term of energy. To overcome

this power limitation, only the nodes on target pathway should be activated, the rest should stay in idle mode. Therefore, we propose PMT^2 : a Predictive Mobile Target Tracking for WMSN.

PMT^2 is a predictive, distributed and cooperative algorithm. It is composed of five main steps (see Fig.2): Wake up, detection, localization, prediction and finally, next sensor selection. Prediction is the most important step in our algorithm. It serves as a basis to choose the node to wake up. It is performed using an Enhanced version of the Extended Kalman Filter (EKF). Unlike standard Kalman filter, EKF is designed for non linear systems. Therefore, it allows to handle more realistic target trajectories. It is usually described as a two steps algorithm: prediction and update. Based on a target mobility model, we have added a verification step in its process. This third step aims to enhance trajectory prediction by checking if the predicted target location belongs to the target mobility graph. This graph is designed by using Voronoi diagram [2]. Moreover, we coupled the enhanced EKF with a change detection mechanism Cumulative Summary (CuSum). CuSum is in charge of detecting abrupt target direction changes.

The performance of PMT^2 is closely related to the deployment strategy. For this purpose, we have proposed a deployment strategy in addition to the tracking algorithm. We have enhanced Virtual Forces Algorithm [3] for weighted areas. In real indoor environment, some areas are more important to monitor than others. For example in a building, entrances, exits and corridors have higher priorities than individual offices. These important areas are weighted according to their importance. In summary PMT^2 :

- Implements a deployment strategy for weighted areas.
- Improves the EKF in order to increase prediction accuracy.
- Uses CuSum to detect and relay sudden and abrupt direction changes.

The rest of this paper is organized as described below: In the next section, we present an overview of existing works. In section III, we introduce the proposed algorithm PMT^2 . Simulation set up and results are discussed in section IV. Finally, section V concludes the paper.

II. RELATED WORK

Existing works about target tracking in Wireless Sensor Networks (WSN), can be classified in three classes: cluster-based, structure-less and predictive-based. The two first classes are related to network architecture while the last one is related to tracking relaying strategy.

A. Structure-less class

In structure-less approach, no network organization is set up. All the nodes have the same level with no hierarchy between them. The tracking is performed in reactively manner at each stage of target evolution inside the region of interest. In [4], the authors propose a Collaborative Tracking Algorithm (CTA) in Wireless Sensor Networks. The main contribution of this work is that the algorithm runs on Heterogeneous WSN composed of two types of sensors: Motion Sensors and Camera Sensors. When a Motion Sensor detects the target, it computes the probability of its detection by a Camera Sensor. If this probability is higher than a predefined threshold, the concerned Camera Sensor is activated to handle the localization. Although CTA presents good results, the heterogeneous network is not easy to set up.

In [5], the authors proposed Dynamical Object Tracking (DOT). A mobile node is in charge of tracking process. Two phases are implemented: target discovery and tracking. In the first one, the mobile node requests all the sensors. The nodes closer to the target replies with tracking information as target position. These closer sensors are kept active. In the second phase, when a sensor detects the target, it sends to the mobile node a message containing target information.

B. Cluster-based class

In cluster-based class, the network is organized in clusters of nodes. A cluster is composed of a cluster head and cluster members. In [6] the authors study the trade off between energy consumption and tracking precision in WSN. For that purpose, they propose a hierarchical, distributed and cluster network architecture. When a node detects the target, it reports its location and the sensed signal strength to the cluster head. Based on the received information, the cluster head selects the three sensors with top signal strength to perform localization. In [7] Bayesian estimation via Quantized Variational Filtering is used to choose the cluster head at each step of tracking process. Once the cluster head is chosen, it selects a group of candidate nodes using a Multi-Objective Genetic Algorithm. This group is in charge of estimating the target position. The main disadvantage of the solutions proposed in [6] and [7], is that too many nodes are involved in the localization process.

In [8], a collaborative algorithm to address the node selection problem in WMSN is proposed. The main goal of this algorithm is to keep the desired density of active nodes while the others remain in sleeping mode. The algorithm starts when the target is detected by a Camera Sensor; it broadcasts its own location to the sensors within its transmission range. Then, each of them computes the probability of detecting the target. If the probability reaches a predefined threshold, the sensor

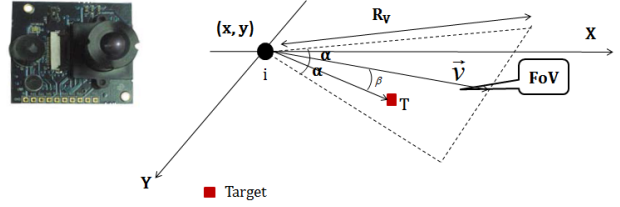


Fig. 1. Imote2 and Camera Sensor's FoV

activates its camera to handle the localization phase. The main drawback of cluster-based approach is that the clusters may overlap each other. Indeed, a cluster member can have more than one cluster head causing unnecessary and redundant data.

C. Predictive-based class

In predictive-based class, models or mechanisms are used to proactively estimate and predict the target movement. This approach can be established on cluster-based or structure-less classes. The solution proposed in [9] aims to conserve energy by limiting the number of active nodes, and decreasing the transmission distance between transmitter and receiver nodes. For that purpose, a predictive-based method is implemented. It consists in a set of linear equation that predicts the upcoming target location based on previous and current ones. Due to the linearity of the proposed predictive method, realistic target movement cannot be captured. In [10], a mobility model based on Autoregressive model is proposed. It allows an accurate prediction of target displacement in wireless networks. This solution requires training data to initialize some parameters of the equation model. This phase has a non-negligible cost.

PMT^2 , assumes predictive-based approach and is not dependent of a specific topology. It is a complete tracking algorithm which uses a solution for every step of the tracking process.

III. PMT^2 : A PREDICTIVE MOBILE TARGET TRACKING ALGORITHM IN WMSN

In this section, we present our proposed tracking algorithm. We start by specifying the application environment. Then, we present the algorithm.

A. Preliminaries and Assumptions

The assumptions made by PMT^2 are enumerated below: The network is composed of a set of Wireless Multimedia Sensors embedding cameras, named Camera Sensors (CS). Each sensor has a conical and directional Field of View (FoV) with opening angle 2α , video sensing radius R_V , and communication range R_T . As an example of existing CS: MEMSIC (formally CrossBow) [11] proposes the imote2 sensor which embed a low-resolution (640x480) camera. An imote2 and its FoV are represented in Fig.1. All the deployed CSs have the same transmission and sensing characteristics.

A single and dynamic target is expected to cross the region of interest. It moves in random manner through the graph designed using obstacle mobility model [2]. In this model, the

topology of the area is considered in modeling the target movement path. Topology includes real world obstructions such as buildings, walls and other structures which are considered as barriers to target movement. Once positions, shapes and sizes of the obstacles are entered, Voronoi Diagram is used to determine the pathways between the obstacles. It is known that Voronoi Diagram is a fundamental graph constructor. In 2D plane, considering a set of location points, this algorithm divides the plane into a group of convex polygons cells, such as all points inside a cell are closest to only one location point. The resulting topological graph summarizes possible target trajectories. It is used in target prediction phase, to check if the target belongs to the designed graph.

B. Deployment strategy

The network is deployed following a Virtual Force Algorithm (VFA) [3]. VFA attempts to enhance the sensors sensing coverage. Repulsive and attractive forces are utilized to determine the new position of sensors. We assume that the deployment algorithm is executed on a powerful device, without energy constraint. Then, we use the algorithm's outputs to deploy the sensors in the real area. We consider that priori informations about the covered area are available, and consequently, we choose a planned initial deployment instead of random one. In planned deployment, the nodes are placed regularly following the topology, avoiding obstructions.

Let our WMSN consists of N Cameras Sensors with IDs $\{CS_1, CS_2, \dots, CS_n\}$. After a planned deployment, each CS_i computes its total Force \vec{F}_i . \vec{F}_i represents the total repulsive and attractive forces applied on CS_i . In the initial work, the sensors have omnidirectional sensing field represented by a circle (360°). However, in our work, we assume directional and limited one. Indeed, as mentioned in the previous subsection, the CS's visual sensing field is represented by a cone (60°).

Thus, \vec{F}_i is applied on the gravity center of this FoV instead on the center of the circle. In this case, the movement of the CS includes both re-positioning and re-orientation. \vec{F}_i is calculated as follow:

$$\vec{F}_i = \sum_{\substack{j=1 \\ j \neq i}}^N \vec{F}_{ij} + \vec{F}_{obs} \quad (1)$$

Where \vec{F}_{ij} is the force between CS_i and CS_j . \vec{F}_{obs} represents the total repulsive forces applied on CS_i by the obstacles. We specify then how to calculate \vec{F}_{ij} :

$$\vec{F}_{ij} = \begin{cases} (W_A(d_{ij} - d_{th}, \alpha_{ij})), & \text{if } d_{ij} > d_{th} \\ 0, & \text{if } d_{ij} = d_{th} \\ (W_R \frac{1}{d_{ij}}, \alpha_{ij} + \pi) & \text{if } \textit{Otherwise} \end{cases} \quad (2)$$

Where W_A and W_R are respectively the measure of attractive and repulsive forces. α_{ij} represents the direction of \vec{F}_i . d_{ij} is the Euclidean distance between the gravity centers of CS_i and CS_j while d_{th} is the threshold distance which controls how close CSs get to each other. Its value is determined based

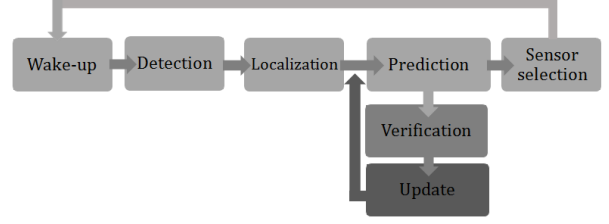


Fig. 2. The Predictive Mobile Target Tracking algorithm process

on the sensing range R_V . With a similar computing principle, \vec{F}_{obs} is calculated based on the distance between the sensor CS_i and the center of obstacles.

In addition to the VFA proposal, we introduce the critical sub-area concept: in real indoor environment, some areas are more important to cover than other. They are selected following their priority in a particular environment. For example in a building, entrances, exits and corridors have higher priorities than individual offices. The critical sub-areas are weighted according to their importance. In this work, we do not study how to weight the critical sub-areas; this is designed by an expert such as a security system designer or an architect. The weight attribute to each critical sub-area is considered in deployment process. Therefore, in our work, \vec{F}_i is as follow:

$$\vec{F}_i = \sum_{\substack{j=1 \\ j \neq i}}^N \vec{F}_{ij} + \vec{F}_{obs} + W_{gt} \vec{F}_{csa} \quad (3)$$

\vec{F}_{csa} is the total attractive of repulsive forces assigned on CS_i by the critical sub-areas. W_{gt} is the weight assign to each one. With this new proposed approach, we have enhanced VFA to W-VFA which handles directional sensors and weighted deployment areas.

C. PMT^2 : Predictive Mobile Target Tracking algorithm

In this sub-section, we present the proposed algorithm: PMT^2 . Fig.2 summarizes its five main steps.

1) *Wake up*: in most tracking application, the target irregularly appears in the area. Thus, putting all the nodes in active mode is unnecessary and too costly. In this work, after the deployment phase, all the CSs are in sleeping mode (sensing channel in hibernation). The communication channel is kept in active mode for communication and collaboration purposes. Periodically, a sub-set of CSs is chosen to be activated. To guarantee the quality of monitoring, this sub-set is selected based on the wake up rate and the number of nodes proposed in [12]. These values depend on network density. In our work, we consider CS's location and orientation in selection process. CSs near the border are given priority over sensors closer to the center of the deployment area. Finally, when a selected sub-set returns in sleeping mode; another relevant one wakes up.

2) *Detection*: when a CS is in active mode, it captures images of the region of interest. Afterward, using the background subtraction method [13], it checks if the target is in

its FoV. Let us remind that this method consists in comparing the background frame, captured during the deployment phase, and the current one. If the moving object has a different color and shapes than the background, the CS concludes that there is a target in its FoV. If the target is detected, the next phase starts. Otherwise, the CS returns to the sleeping mode.

3) *Localization*: in most tracking solutions in WSN, localization is inferred using the Received Signal Strength Indicator (RSSI) from at least 3 nodes [6] [7]. In our work, we use an image processing solution [14]. When a CS detects the target, it captures images, and uses them to perform localization. For that purpose, it computes the size of the target on the captured frame by using CS features such as dimensions and focal length. Then, based on its own location it calculates the distance between the target and itself and thus, target location.

4) *Prediction*: in our tracking algorithm, prediction is the most important step. Target movement is anticipated, and tracking process is relayed from node to node. To accomplish this task, the current activated CS uses the proposed enhanced Extended Kalman Filter (EKF) coupled with a change detection mechanism. The EKF is an extension of the standard Kalman Filter for non-linear models. It is an efficient mathematical tool which uses current information to predict future ones. Usually, it is described as a system state (4) and measurement update model (5).

$$X_{t+1} = f_t(X_t) + w_t \quad (4)$$

$$Z_t = h_t(X_t) + v_t \quad (5)$$

Where X_{t+1} is the mobility state vector at time $t + 1$:

$$X_{t+1} = [x_{t+1}, v_{x_{t+1}}, a_{x_{t+1}}, y_{t+1}, v_{y_{t+1}}, a_{y_{t+1}}]'$$

x_{t+1} and y_{t+1} are target's 2D coordinates at time $t + 1$. $v_{x_{t+1}}$ and $v_{y_{t+1}}$ specify its velocity while $a_{x_{t+1}}$ and $a_{y_{t+1}}$ denote its acceleration. Z_t is the measurement update vector:

$$Z_t = [x_{z_t}, y_{z_t}]'$$

Where x_{z_t} and y_{z_t} specify the measured target coordinates. $f_t(\cdot)$ is a nonlinear representation and $h_t(\cdot)$ is a nonlinear observation function. w_t and v_t are white Gaussian noise with zero mean and respectively Q_{w_t} and Q_{v_t} variance. We assume that they are independent of each other. EKF [15] is divided in two main steps: prediction and update steps. In this work, we have added another step: verification (Fig.2).

a) *Prediction step*: During this phase, $\hat{X}_{t+1|t}$ which is the estimated mobility state vector at time $t + 1$ is computed using equation (6). Also $P_{t+1|t}$, the covariance matrix associated to the predicted mobility state vector is evaluated from the previous estimated $P_{t|t}$ and process noise covariance matrix Q_{wt} using equation (7).

$$\hat{X}_{t+1|t} = f_t \hat{X}_{t|t} \quad (6)$$

$$P_{t+1|t} = F_t P_{t|t} F_t' + Q_{wt} \quad (7)$$

Where F_t is the Jacobian matrix of the state transition function f_t . We obtain it as described below:

$$F_t = \frac{\partial f}{\partial X} \Big|_{\hat{X}_{t|t}}$$

b) *Verification step*: The main objective of the verification step is to check if the predicted coordinates, obtained in the previous step, belong to the topological graph constructed using Voronoi Diagram. This graph is constructed as described in section III.A. To achieve this task, we must determine if the predicted coordinates belong to one of the edges which form the graph. Each edge being a straight line, this amounts to check if one point belongs to one of the straight lines. If this is true, we directly perform update step. Otherwise, we correct these coordinates by replacing them by the closest coordinates on the graph.

c) *Update step*: This step aims to correct the predicted coordinates. While in the verification step, the coordinates are corrected following the mobility graph. In this step, the coordinates are corrected based on the measurement of real target location. We firstly compute K_{t+1} , which represents the Kalman gain.

$$K_{t+1} = P_{t+1|t} H_{t+1}' [H_{t+1} P_{t+1|t} H_{t+1}' + Q_{v_{t+1}}]^{-1} \quad (8)$$

Where H_t is the Jacobian matrix of observation function h_t . We obtain it as described below:

$$H_t = \frac{\partial h}{\partial X} \Big|_{\hat{X}_{t|t+1}}$$

Finally, $\hat{X}_{t+1|t+1}$, the mobility state estimate, and $P_{t+1|t+1}$, the corresponding covariance matrix are calculated by correcting the predicted ones as follow:

$$\hat{X}_{t+1|t+1} = \hat{X}_{t+1|t} + K_{t+1} [Z_{t+1} - H_{t+1} \hat{X}_{t+1|t}] \quad (9)$$

$$P_{t+1|t+1} = [I - K_{t+1} H_{t+1}] P_{t+1|t} \quad (10)$$

As we said previously, we have associated to the EKF a change detection mechanism. In such realistic deployment area, where obstacle exists, abrupt and sudden target direction changes may occur. Therefore, we propose to use CuSum [16] for Cumulative Summary, a change detection mechanism, to detect and relay them. We use CuSum test instead of another detection change mechanism for its efficiency in our tracking context with numerous obstacles.

CuSum test is described as follow:

$$g_{t+1} = g_t + S_{t+1} - v \quad (11)$$

As initial condition $g_{t+1} = 0$. S_{t+1} is the normalized innovation process of the EKF. In our work, we define it as described below:

$$S_{t+1} = \frac{Z_{t+1} - H_{t+1} \hat{X}_{t+1|t}}{\sqrt{(P_{t+1|t} + Q_{wt}) H_{t+1}' + Q_{vt}}} \quad (12)$$

There are two important parameters in CuSum test, the drift parameter v and the alarm threshold h . The value of v is subtracted at each iteration to prevent positive drifts, that may yield a false alarm. h is called alarm threshold because

a direction change is detected when $g_{t+1} > h$. This condition is considered as the stopping rule of the CuSum test. After an alarm, the value of g_{t+1} is reset to zero. The smaller the values of v and h , the more sensitive the test is. Their values is chosen according to the application context.

Two particular cases may occur. The first one is an error in next target location prediction. The second one is a target which enters in a CS's FoV and stays static for a certain period of time. The solution for both cases is to send back a *Re-Wake up* message to the last CS which detects and observes the target. At the end of this phase, the upcoming target location is available. Using this information, the current active CS selects the succeeding one.

5) *Next sensor selection*: once the future target location is predicted; the next sensor selection phase starts based on two criteria: location and orientation angle. In this phase, the objective is to relay the tracking from node to node until the target leaves the area. To be chosen, a CS must satisfy Target in Sector (TIS) test [17]:

$$\begin{cases} d_{iT} \leq R_v \\ \beta \in [-\alpha, \alpha] \end{cases} \quad (13)$$

Where d_{iT} denotes the distance between the node i and the target T . β is the angle between $i\vec{T}$ and \vec{v} . \vec{v} is a sensing vector which divide the FoV into half (see Fig.1). 2α represents the opening angle of the CS. The sensor with the smallest value of d_{iT} , which means is closest to the target, is selected to be the succeeding one. As we said previously, initially all the CSs are in idle mode. Then, the active CS, which has already detected the target and computed its future location, activates the succeeding one by sending a *Wake up* message. A *Wake up* message includes sender ID. Each target coordinates captured by a CS are sent to the sink which reconstitutes the target pathway.

TABLE I
PARAMETERS VALUES OF THE SIMULATION

Mac Layer protocol	IEEE 802.15.4
Access Mode	CSMA/CA non-beaconed
Area size	100m x 100m
Target speed	1.38 m/s (pedestrian)
Number of nodes	10, 20, 30, 40, 50
Simulation Time	200 s
Transmission range (R_T)	30 m
Depth of view (R_V) of CS	20 m
Angle of view of CS (2α)	$\pi/3$
Size of messages	100 bytes
Height of CS	2 m

TABLE II
ENERGY MODEL PARAMETERS (IMOTE2 [11])

Initial node energy	3 AAA
Active power	0.279 Joule/Second
Idle power (Radio on Camera off)	0.226 Joule/Second
Sleep power (Radio off Camera off)	0.015 Joule/Second
rx/tx Power (Frequency 104 MHz)	0.078 Joule/Second
Camera Power	0.044 Joule/Second

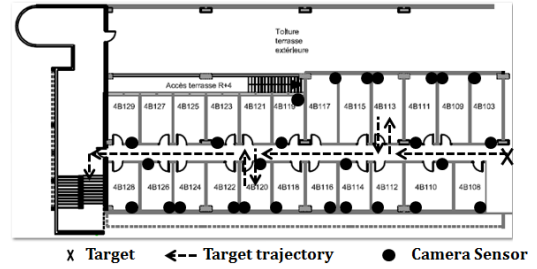


Fig. 3. Indoor deployment area (part of Paris-Est research laboratory)

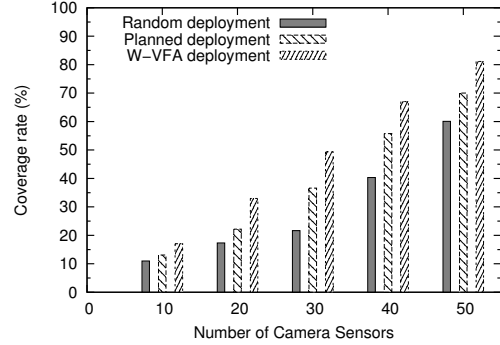


Fig. 4. Coverage rate vs. number of Camera Sensors

IV. PERFORMANCE EVALUATION

In this section, simulation setting, evaluation metrics and results are discussed. In order to evaluate PMT^2 , we have used NS-2 simulator [18]. Table 1 summarizes the simulation parameters used. The values of R_T , R_V and 2α are selected according to Imote2 features [11]. Based on preliminaries simulations, the configuration parameters for CuSum test are fixed to $v = 0.5$ and $h = 5$. We choose these values because they better handle the trade-off between the tracking accuracy and energy consumption.

We start by evaluating the deployment strategy. Then, we evaluate the proposed tracking algorithm PMT^2 . Fig.3 illustrates the deployment area (Paris-Est computing research laboratory) used to achieve these tasks. Based on this indoor environment, five random target trajectories with frequent and sudden direction changes are generated and used for the evaluation. Three metrics are used as well: tracking accuracy, energy consumption and number of exchanged messages. We compare the performance of our proposed algorithm to three other ones: 1) BASIC, as its name suggests it is a basic scheme where all the Camera nodes are always in active mode, the localization is performed using an image processing solution [14], 2) the cluster-based solution OCNS described in [8], more details are given in section II, and 3) PTA, another predictive-based algorithm where the standard Kalman Filter for linear systems is used [19].

A. Deployment strategy

The deployment strategy is evaluated by comparing the coverage rate of the area before and after the deployment

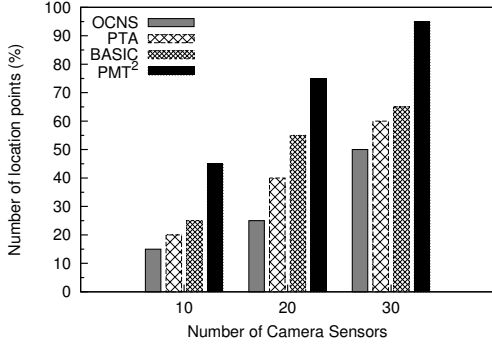


Fig. 5. Number of location points vs. number of Camera Sensors

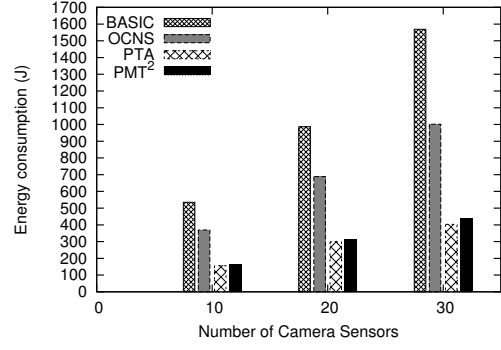


Fig. 7. Energy consumption vs. number of Camera Sensors

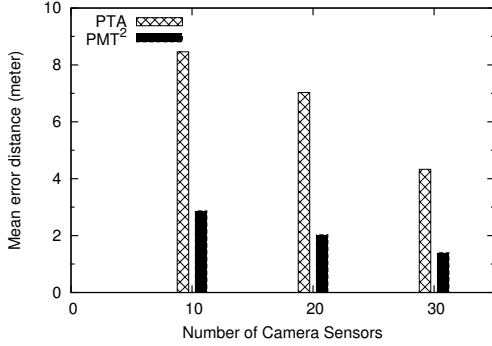


Fig. 6. Mean error distance vs. number of Camera Sensors

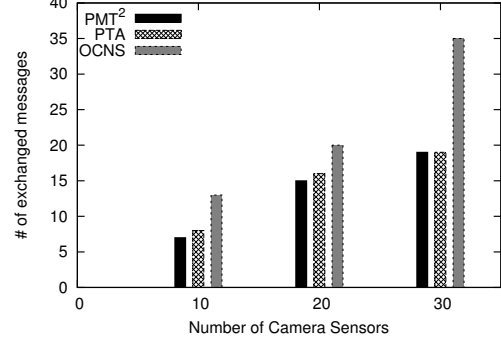


Fig. 8. Exchanged messages vs. number of Camera Sensors

algorithm. We compare the performance of the proposed deployment strategy vs. random and planned ones. In the Random deployment, the CSs are scattered with random location and orientation, while in the Planned deployment the CSs are placed regularly following the topology of the area. Fig.4. shows the coverage rate vs. the number of Camera Sensors. We observe that W-VFA, the proposed VFA algorithm for weighted critical-sub-areas, has the best coverage performances. Indeed, the proposed strategy based on attractive and repulsive forces, allows to calculate both the new location and orientation of the CSs. W-VFA covers up to 12% more than Planned deployment and up to 27% more than Random one.

B. Tracking accuracy

We calculate tracking accuracy in two different ways. In the first one, we consider the number of location points captured along a defined pathway. Based on the size of the area of interest, target speed, the maximal number of deployed nodes and their sensing range, the best tracking precision (100%) is reached when one location point is reported every five meters. Fig.5 shows the average number of location points captured vs. the number of Camera Sensors. Due to the Cameras, which are always kept active, BASIC solution has better results than OCNS and PTA. Nevertheless, it is outperformed by PMT^2 . Indeed, PMT^2 has the best results. It reaches 95% for 30 nodes. The Enhanced Extended Kalman Filter coupled with CuSum allows to predict realistic target behavior, with possible sudden direction changes.

Moreover, unlike PMT^2 , where the CSs are deployed following W-VFA strategy, in BASIC solution as well as OCNS and PTA, the CSs are scattered in random manner. We conclude, that the deployment strategy has a positive impact on the quality of tracking. This impact is more important than the number of active CSs. OCNS has the worst results because of the wake up and relaying strategy. A target can enter the area of interest without being detected immediately.

The second way to calculate the tracking accuracy is to compare the predicted target trajectory to the real one. The results are obtained by computing the mean error distance between them. Fig.6 summarizes the obtained results. Only predictive algorithms are concerned by this metric. Thus, BASIC and OCNS are not represented on the figure. We observe that the predicted trajectory by PMT^2 is closer to the real one, compared to the trajectory predicted by PTA. Indeed, PTA is designed for linear systems. It cannot handle non-linear and realistic target trajectory.

C. Energy consumption

The energy consumption is evaluated based on NS-2 energy model [20] and Imote2 [11] [21] power consumption summarized in Table II. To evaluate energy consumption, we calculate the energy cost of cameras activation, active period duration, predictive algorithm computation, and communication. Fig.7 shows the mean energy consumption of the whole network in Joule (J) during one simulation (150 seconds) depending on the number of Camera Sensors. Obviously, BASIC is an unrealizable solution. Because of the permanent active state

of CSs, it consumes up to 1568.77 J. Due to the proposed predictive mechanism, PMT^2 consumes up to 567.29 Joules less than OCNS. Thanks to the coupled predictive mechanism, only the nodes that are on target pathway are activated. Furthermore, the node selection process in OCNS has a non negligible communication cost. Due to the nearly similar processing and communication principal, PMT^2 and PTA consume approximatively the same energy.

D. Exchanged messages

This metric represents the communication overhead. We calculate it by considering the number of collaborative messages exchanged during the tracking process. Fig.8 shows the mean number of exchanged messages vs. the number of Camera Sensors. BASIC solution is not represented in this figure because there is no communication between nodes. PMT^2 and PTA have the same communication rules. As explained in section III.C.5, *Wake up* messages are exchanged when the future location of the target is calculated. Therefore, PMT^2 and PTA have approximatively the same communication cost. PMT^2 outperforms OCNS. Indeed, while in PMT^2 the only messages exchanged are *Wake up* ones. In OCNS, two types of messages are exchanged: *Wake up* and *Collaborative* ones. *Collaborative* messages are exchanged during the node election process.

V. CONCLUSION

In this paper, we have tackled the problem of target tracking in Wireless Multimedia Sensor Networks. We propose a Predictive Mobile Target Tracking Algorithm called PMT^2 . Prediction is performed using an Enhanced Extended Kalman Filter associated with a change detection mechanism called Cumulative Summary. This combined mechanism allows to track and capture very realistic target behavior. Based on the predicted information, PMT^2 reduces the number of nodes participating in the tracking process, and thus saving resources. The simulation results show that PMT^2 performs target tracking with the best results, up to 35% more than existing works. Moreover, it saves up to 55% more energy compared to other works. Object identification as well as multi-target tracking, are left as future work.

REFERENCES

- [1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer networks*, vol. 51, no. 4, pp. 921–960, 2007.
- [2] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks," in *International conference on Mobile computing and networking (MobiCom)*, San Diego, California, USA, 2003.
- [3] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *INFOCOM*, San Francisco, California, USA, 2003.
- [4] I. Boulanouar, S. Lohier, A. Rachedi, and G. Roussel, "Cta: a collaborative tracking algorithm in wireless sensor networks," in *International Conference on Computing, Networking and Communications (ICNC)*, San Diego, California, USA, 2013.
- [5] H.-W. Tsai, C.-P. Chu, and T.-S. Chen, "Mobile object tracking in wireless sensor networks," *Computer communications*, vol. 30, no. 8, pp. 1811–1825, 2007.

- [6] S. Deshpande and K. M. Sivalingam, "A study of energy vs. quality of tracking trade-off in wireless sensor networks," in *International Conference on Advanced Networks and Telecommunication Systems (ANTS)*, Bengaluru (Bangalore), India, 2011.
- [7] M. Mansouri, L. Khoukhi, H. Nounou, and M. Nounou, "Genetic algorithm optimization for quantized target tracking in wireless sensor networks," in *Global Telecommunications Conference (GLOBECOM)*, Houston, Texas, USA, 2011.
- [8] L. Liu, X. Zhang, and H. Ma, "Optimal node selection for target localization in wireless camera sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 7, pp. 3562–3576, 2010.
- [9] F. Deldar and M. H. Yaghmaee, "Energy efficient prediction-based clustering algorithm for target tracking in wireless sensor networks," in *International Conference on Intelligent Networking and Collaborative Systems (INCOS)*, Thessaloniki, Greece, 2010.
- [10] Z. R. Zaidi and B. L. Mark, "Mobility tracking based on autoregressive models," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 1, pp. 32–43, 2011.
- [11] Imote2. [Online]. Available: <http://www.memsic.com>
- [12] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "Peas: A robust energy conserving protocol for long-lived sensor networks," in *International Conference on Distributed Computing Systems (ICDCS)*, Providence, Rhode Island, USA, 2003.
- [13] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *International Conference on Pattern Recognition (ICPR)*, Tampa, FL, USA, 2008.
- [14] H. Oztarak, K. Akkaya, and A. Yazici, "Efficient localization and tracking of multiple objects in wireless multimedia sensor networks," *Ad Hoc and Wireless Sensor Networks*, vol. 19, 2013.
- [15] P. Zarchan and H. Musoff, *Fundamentals of Kalman filtering: a practical approach*. Aiaa, 2005, vol. 208.
- [16] F. Gustafsson, *Adaptive filtering and change detection*. Wiley New York, 2000, vol. 1.
- [17] Y. E. Osais, M. St-Hilaire, and R. Y. Fei, "Directional sensor placement with optimal sensing range, field of view and orientation," *Mobile Networks and Applications*, vol. 15, no. 2, pp. 216–225, 2010.
- [18] The NS-2 website. [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [19] I. Boulanouar, S. Lohier, A. Rachedi, and G. Roussel, "PTA: A predictive tracking algorithm in wireless multimedia sensor networks," in *International Conference on Global Information Infrastructure and Networking Symposium (GIIS)*, Trento, Italy, To appear.
- [20] I. T. Downard, "Simulating sensor networks in ns-2," DTIC Document, Tech. Rep., 2004.
- [21] D. J. A. Barton-Sweeney and A. Savvides, "imote2 node and enable camera module power measurements," in *ENALAB Technical Report: 090601*, 2006.