

Algorithme de suivi de cible mobile dans les réseaux de capteurs multimédia sans fil

Ibtissem Boulanouar
Laboratoire d'Informatique de
l'Institut Gaspard Monge
(LIGM - UMR 8049)
77420 Champs-sur-Marne,
France
ibtissem.boulanouar@u-
pem.fr

Abderrezak Rachedi
Laboratoire d'Informatique de
l'Institut Gaspard Monge
(LIGM - UMR 8049)
77420 Champs-sur-Marne,
France
abderrezak.rachedi@u-
pem.fr

Stéphane Lohier
Laboratoire d'Informatique de
l'Institut Gaspard Monge
(LIGM - UMR 8049)
77420 Champs-sur-Marne,
France
stephane.lohier@u-
pem.fr

Gilles Roussel
Laboratoire d'Informatique de
l'Institut Gaspard Monge
(LIGM - UMR 8049)
77420 Champs-sur-Marne,
France
gilles.roussel@u-pem.fr

ABSTRACT

Dans ce papier, nous proposons un algorithme prédictif pour le suivi de cible mobile dans le contexte des réseaux de capteurs multimédia sans fil. La gestion des ressources, particulièrement l'énergie, étant un sujet critique dans ce type de réseaux, l'objectif de la solution proposée est de gérer le compromis entre la précision du suivi et l'économie d'énergie. Nous évaluons ses performances à l'aide de simulations en le comparant à d'autres algorithmes proposés dans la littérature.

Keywords

Réseaux de capteurs sans fil, prédiction, filtre de Kalman, suivi de cible mobile

1. INTRODUCTION

Les dernières avancées technologiques dans le domaine des communications sans fil ont permis le développement de capteurs sans fil à faible consommation énergétique. Ces petits capteurs peuvent être équipés de caméras, de microphones et d'autres éléments pouvant gérer des flux multimédia. Les réseaux de capteurs multimédia sans fil correspondent à un ensemble de petits appareils interconnectés entre eux et déployés dans une zone d'intérêt afin de gérer des tâches spécifiques comme la localisation, le suivi ou l'identification de cible.

Dans ce papier, nous nous intéressons au suivi de cible mobile. Cette application consiste donc à détecter et à localiser une cible, par exemple un intrus, traversant une zone d'intérêt. Contrairement aux systèmes de surveillance classiques qui nécessitent une infrastructure fixe, les réseaux de capteurs multimédia sans fil sont aussi faciles à installer qu'à désinstaller. Cependant, étant autonomes et donc contraints énergétiquement, les capteurs ne peuvent pas rester actifs tout le temps. Afin de résoudre ce problème, l'idée simple est d'activer uniquement les capteurs qui se trouvent sur la trajectoire de la cible au moment où cette dernière est à leur portée. Nous proposons pour cela PMT^2 (*Predictive Mobile Target Tracking*), un algorithme prédictif, distribué et coopératif de suivi de cible mobile.

PMT^2 comprend cinq étapes : réveil, détection, localisation, prédiction et sélection du capteur suivant. La prédiction de la trajectoire étant l'étape la plus importante, nous utilisons une version améliorée du Filtre de Kalman Étendu (FKE). En effet, le FKE se compose habituellement de deux étapes : prédiction et correction. En se basant sur le modèle de mobilité de la cible, nous y avons ajouté l'étape de vérification. Cette dernière a pour objectif d'améliorer la prédiction en vérifiant si la position de la cible prédite appartient au graphe de mobilité. Ce graphe est réalisé à l'aide du diagramme de Voronoi [6]. Enfin, nous avons couplé cette version améliorée du FKE avec un mécanisme de détection de changements brusques de direction : CuSum (Cumulative Summary).

Les performances de PMT^2 étant fortement liée à la stratégie de déploiement, nous avons aussi proposé un algorithme de déploiement. Nous avons étendu la solution basée sur les forces virtuelles [13] afin d'y inclure le principe de zones critiques. En effet, dans des environnements réels, certaines zones sont plus importantes à couvrir que d'autres. Dans un immeuble par exemple, les couloirs, les entrées/sorties et

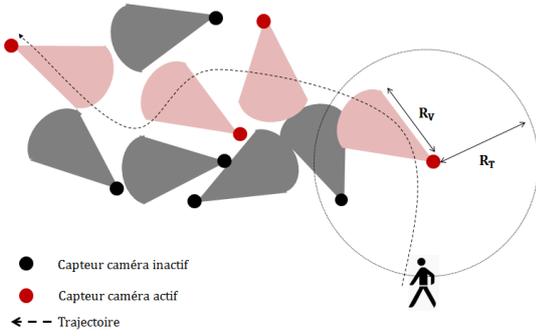


Figure 1: Réseaux de capteurs sans fil multimédia homogène

lieux de fort passages sont plus importants à couvrir qu'un bureau. Ces zones sont pondérées suivant leurs importance.

L'article est organisé comme suit : la prochaine section décrit le type de réseau déployé. La section 3 décrit la stratégie de déploiement utilisée ainsi que le modèle de mobilité de la cible. Notre solution est présentée dans la section 4. Dans la section 5, les résultats de simulations sont présentés et commentés. Enfin le papier est conclut en section 6.

2. DESCRIPTION DU SYSTÈME

Nous déployons un réseau de capteurs homogènes, composé de capteurs caméra (CC) disposant d'une portée visuelle R_V et d'une portée de communication R_T . Un exemple du réseau déployé est illustré dans la figure 1. Les CCs sont déployés en utilisant la stratégie de déploiement décrite ci-dessous. Après cette phase, chaque CC connaît sa position et son orientation. Nous supposons que chaque capteur connaît ses voisins immédiats (ID, coordonnées et orientation) grâce à un échange préalable de messages.

Une seule cible à la fois est supposée traverser la zone d'intérêt. Son modèle de mobilité est décrit ci-dessous.

2.1 Modèle de mobilité de la cible

Nous considérons dans PMT^2 que la cible se déplace suivant le modèle de mobilité proposé dans [6]. Nous supposons donc dans ce qui suit qu'une **seule** cible traverse la zone d'intérêt de manière aléatoire en empruntant une des trajectoires possibles constituant le graphe de mobilité de la cible. Ce modèle prend en considération la topologie de la zone d'intérêt pour modéliser le mouvement de la cible. La topologie d'une aire inclue les reliefs, les occlusions et tout autre élément pouvant être un obstacle au déplacement de la cible. Le principe général est décrit comme suit : une fois les positions, les formes et les tailles des obstacles saisies, le diagramme de Voronoï est utilisé pour déterminer les chemins possibles entre ces barrières. Le diagramme de Voronoï est un constructeur de graphe assez connu dans le domaine des réseaux de capteurs. Il est souvent utilisé pour modéliser le mouvement d'objet ou de nœuds dans une aire donnée et résoudre ainsi des problématiques comme le déploiement de capteurs [7]. Dans un espace en 2D, considérant un ensemble de points, ce diagramme divise cet espace en un groupe de

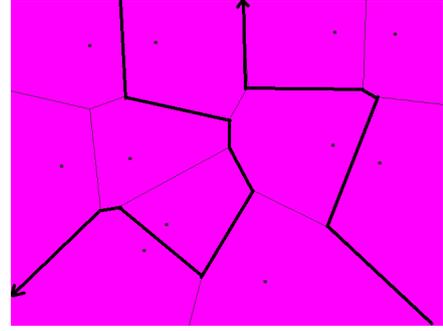


Figure 2: Exemple illustratif d'un diagramme de Voronoï

cellules de forme polygones, tels que tous les points se trouvant à l'intérieur d'une cellule sont plus proches d'un point p que de tout autre point dans l'aire principale. Le graphe obtenu décrit toutes les trajectoires possibles que la cible peut emprunter dans une zone donnée. La figure 2 illustre un exemple de diagramme de Voronoï avec des exemples de trajectoires possibles.

3. STRATÉGIE DE DÉPLOIEMENT

Comme étape préliminaire, la phase de déploiement a pour but de maximiser la couverture vidéo. Nous proposons pour cela une version adaptée à notre contexte de l'algorithme VFA (*Virtual Force Algorithm*) [13]. VFA utilise le concept de forces répulsives et attractives afin de déterminer une meilleure position pour les capteurs. Nous supposons que cette phase est exécutée en amont, sur une machine n'ayant pas de contraintes énergétiques (exemple : un ordinateur). Les résultats seront ensuite utilisés pour déployer les capteurs dans la zone d'intérêt.

Nous disposons d'informations préliminaires sur cette zone d'intérêt, nous choisissons donc pour commencer un déploiement initial planifié au lieu d'un déploiement totalement aléatoire. Dans un déploiement planifié, les capteurs sont placés de façon intelligente et régulière en fonction de la topologie de la zone, évitant les obstacles et les obstructions. Une fois que les coordonnées initiales sont obtenues, elles serviront de base pour calculer les nouvelles. C'est à ce niveau que l'algorithme de déploiement intervient.

Notre réseau se compose de N capteurs caméra ayant pour identifiants $\{CC_1, CC_2, \dots, CC_N\}$. Suite au déploiement initial, chaque CC_i calcule sa force totale \vec{F}_i . \vec{F}_i représente le total des forces attractives et répulsives appliquées sur le CC_i , elle est calculée comme suit :

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

\vec{F}_{ij} représente la force exercée entre le CC_i et le CC_j . \vec{F}_{obs} représente la force répulsive totale exercée sur le CC_i par les obstacles. Le calcul de \vec{F}_{ij} à l'aide du système d'équation 2 se fait comme suit :

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

W_A et W_R étant respectivement les mesures des forces attractives et répulsives. α_{ij} représente la direction de \vec{F}_i . d_{ij} est la distance Euclidienne entre les centres de gravité de CC_i et CC_j alors que d_{th} est le seuil de distance qui contrôle la proximité ou l'éloignement des CCs. En suivant le même principe de calcul, \vec{F}_{obs} est calculée en se basant cette fois sur la distance entre le CC_i et le centre de gravité de l'obstacle.

Dans ce travail, nous introduisons le concept de sous-zones critiques et nous considérons les forces appliquées par celles-ci. Dans des environnements réels, il existe des sous-zones plus importantes à couvrir que d'autres. Dans un immeuble par exemple, les couloirs, les entrées/sorties et les lieux de forts passages sont plus importants à couvrir que des bureaux. Autre exemple : dans un musée la priorité est donnée aux œuvres d'art. Ces sous-zones sont pondérées en fonction de leur importance. Nous ne nous attarderons pas ici sur la façon de pondérer ces zones. Cette tâche est réalisée par des professionnels tel que les architectes ou les concepteurs des systèmes de sécurité. Le poids attribué à chaque zone est considéré dans le processus de déploiement, par conséquent, \vec{F}_i est calculé en utilisant l'équation 3.

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

\vec{F}_{szc} est le total des forces appliquées par les sous-zones critiques sur le CC_i . Sa valeur est calculée en utilisant l'équation 2, la distance utilisée dans ce cas est celle qui sépare le CC_i et le centre de gravité de la sous-zone critique. W_{gt} est le poids assigné à chacune d'elles.

Grâce à cette nouvelle approche, nous améliorons l'algorithme de déploiement initial VFA en W-VFA (*Weighted-VFA*) qui prend désormais en compte les sous-zones critiques et leurs poids. Le réseau est déployé en utilisant les résultats obtenus grâce à W-VFA.

4. PMTT : PREDICTIVE MOBILE TARGET TRACKING

$PMTT^2$ est un algorithme de suivi de cible prédictif décomposé en cinq (05) principales étapes : réveil, détection, localisation, prédiction et sélection du capteur suivant. Il est adapté aux cibles à évolution non-linéaire avec de possibles changements brusques de trajectoires.

4.1 Réveil

Dans la plus part des applications de suivi, la cible apparaît de manière asynchrone dans la zone d'intérêt. Ainsi, mettre en permanence les capteurs en état actif est inutile et coûteux. Initialement, nous considérons donc que tous les capteurs sont en hibernation. Le module de communication restant toujours actif pour des raisons de collaboration. Périodiquement, un sous-ensemble de nœuds est activé. Afin de garantir la qualité de la surveillance tout en préservant au mieux les ressources, deux paramètres importants sont à

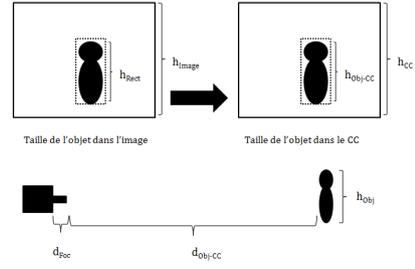


Figure 3: Processus à suivre pour calculer la distance entre le capteur caméra et la cible [11]

considérer [12] : Le nombre de capteurs composant ce sous-ensemble et leur fréquence de réveil.

Pour calculer le nombre N_{SE} de capteurs composant chaque sous-ensemble, nous nous basons sur la densité totale des capteurs déployés, la taille de la zone d'intérêt et la taille du champ visuel des CCs comme décrit dans l'équation suivante :

$$N_{SE} = TailleZoneIntert * \frac{c}{TailleChampVisuel} \quad (4)$$

Où c est une constante qui définit le taux de couverture souhaité.

La fréquence de réveil est quant à elle fixée en fonction du contexte applicatif en considérant par exemple la vitesse de déplacement moyenne de la cible. Une fois les valeurs de ces deux paramètres fixées, les capteurs à réveiller sont choisis en fonction de leurs positions. Nous supposons que la cible pénètre dans la zone d'intérêt par une de ses bordures, ainsi les capteurs se trouvant proche de cette bordure ont la priorité. Lorsque le sous-ensemble sélectionné est désactivé, un autre est activé. Le processus est répété jusqu'à ce que la cible soit détectée.

4.2 Détection

Chaque capteur multimédia réveillé capture des images de son champ de vision. En utilisant la technique du *background subtraction* [2], il vérifie si une cible se trouve à sa portée visuelle. Cette méthode consiste à superposer et à comparer l'image courante et l'initiale, prise durant la phase de déploiement. Si la cible est détectée la prochaine étape peut débuter. Autrement, le capteur se remet en hibernation.

4.3 Localisation

Contrairement aux solutions classiques utilisant la puissance du signal reçu ou tout autre technique de géo-localisation impliquant au minimum trois (03) nœuds [4] [9], dans ce travail, la localisation est effectuée à l'aide d'une solution basée sur le traitement d'image [11] et impliquant un seul nœud à la fois. Dans cette solution, l'idée est d'estimer la distance entre la cible et le capteur. En disposant de cette information clé et partant de l'hypothèse que chaque capteur connaît ses coordonnées, la position de la cible peut ainsi être déduite. Dans un premier temps, la taille de l'objet dans la camera du capteur h_{Obj-CC} doit être calculée en utilisant l'équation 5, pour cela trois (03) données sont requises : la hauteur du capteur h_{CC} , la hauteur de l'image h_{Image} et la

hauteur du rectangle bordant l'objet dans l'image h_{Rec} .

$$h_{Obj-CC} = h_{CC} \frac{h_{Rec}}{h_{Image}} \quad (5)$$

Dans un second temps, la distance d_{Obj-CC} séparant l'objet de la caméra doit être estimée en exploitant cette fois-ci la hauteur estimée de l'objet dans la réalité h_{Obj} ainsi que la distance focale de la caméra du capteur d_{Foc} . h_{Obj} est obtenu en se basant sur la hauteur de placement de la caméra.

$$d_{Obj-CC} = h_{Obj} \frac{d_{Foc}}{h_{Obj-CC}} \quad (6)$$

Enfin, la position de la cible peut être déduite en se basant sur la position du capteur. La figure 3 illustre ce mécanisme. Une fois la cible localisée le CC calcule les futures coordonnées de la cible comme décrit dans la phase suivante.

4.4 Prédiction

Cette étape est la plus importante dans le processus du suivi. Le déplacement de la cible est anticipé et le processus de suivi relayé de nœud en nœud jusqu'à ce que la cible quitte la zone d'intérêt. Pour ce faire, une version améliorée du filtre de Kalman Étendu est proposée. Le Filtre de Kalman Étendu (FKE) est une extension du Filtre de Kalman standard pour les modèles non-linéaires. Il permet de prédire les futures coordonnées de la cible en se basant sur sa position actuelle. Il se décompose habituellement en deux phases : prédiction et correction. Dans ce travail, nous proposons d'ajouter une phase de vérification.

Le FKE est décrit à l'aide d'un modèle d'évolution d'état et un autre de mesure qu'il suppose non-linéaires comme suit :

Modèle d'évolution d'état :

$$X_{t+1} = f_t(X_t) + w_t \quad (7)$$

Modèle de mesure :

$$Z_t = h_t(X_t) + v_t \quad (8)$$

X_{t+1} est le vecteur d'état. Comme le modèle est cette fois non-linéaire, l'état décrit donc la position, la vitesse et l'accélération de la cible à l'instant $t + 1$.

$$X_{t+1} = [x_{t+1}, v_{x_{t+1}}, a_{x_{t+1}}, y_{t+1}, v_{y_{t+1}}, a_{y_{t+1}}]' \quad (9)$$

Où x_{t+1} et y_{t+1} sont les coordonnées de la cible ; $v_{x_{t+1}}$ et $v_{y_{t+1}}$ représentent sa vitesse ; enfin, $a_{x_{t+1}}$ et $a_{y_{t+1}}$ représentent son accélération.

Z_t est le vecteur de mesure où x_{z_t} et y_{z_t} sont les coordonnées mesurées. Il est décrit comme suit :

$$Z_t = [x_{z_t}, y_{z_t}]' \quad (10)$$

$f_t(\cdot)$ et $h_t(\cdot)$ sont des fonctions non-linéaires ; W_t et V_t sont des bruits blancs Gaussiens à moyenne nulle et variances respectives Q_{wt} et Q_{vt} . On les suppose indépendants. Cidessous, les phases de prédiction, vérification et correction sont décrites.

Phase de prédiction. Durant cette phase, $\hat{X}_{t+1|t}$, le vecteur d'état estimé à l'instant $t + 1$ est obtenu à l'aide de l'équation 11. De même, la matrice de covariance associée au vecteur d'état prédit est obtenue à l'aide l'équation 12.

$$\hat{X}_{t+1|t} = f_t \hat{X}_{t|t} \quad (11)$$

$$P_{t+1|t} = F_t P_{t|t} F_t' + Q_{wt} \quad (12)$$

F_t est la matrice Jacobienne de la fonction de transition d'état f_t , elle est obtenue comme suit :

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

Phase de vérification. L'objectif de cette phase est de vérifier si les coordonnées prédites se trouvent bien sur une des trajectoires possibles de la cible définies à l'aide du modèle de mobilité décrit dans la section 2.1. Pour cela, on doit déterminer si les coordonnées de la cible obtenues lors de la phase de prédiction appartiennent à une des droites formant le graphe. Si c'est le cas, nous passons à la phase de correction, sinon, nous les corrigeons en les remplaçant par les coordonnées les plus proches se trouvant sur le graphe.

Phase de correction. Cette étape a pour objectif la correction des coordonnées de la cible prédite. Alors que dans la phase de vérification celles-ci sont corrigées en fonction du graphe de mobilité de la cible, dans cette étape, les coordonnées sont corrigées en fonction des mesures. Pour cela nous commençons par calculer K_{t+1} , le gain de Kalman qui décide du degré d'importance de l'intégration de la mesure.

$$K_{t+1} = P_{t+1|t} H_{t+1}' [H_{t+1} P_{t+1|t} H_{t+1}' + Q_{v_{t+1}}]^{-1} \quad (13)$$

H_t est la matrice Jacobienne de la fonction d'observation h_t , elle est obtenue comme suit :

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

Enfin, le vecteur d'état $\hat{X}_{t+1|t+1}$ et la matrice de covariance correspondante $P_{t+1|t+1}$ sont mis à jour.

$$\hat{X}_{t+1|t+1} = \hat{X}_{t+1|t} + K_{t+1} [Z_{t+1} - H_{t+1} \hat{X}_{t+1|t}] \quad (14)$$

$$P_{t+1|t+1} = [I - K_{t+1} H_{t+1}] P_{t+1|t} \quad (15)$$

Le caractère imprévisible de la cible ainsi que les différents obstacles se trouvant sur sa trajectoire peuvent provoquer des changements brusques de direction. Nous proposons donc de coupler le FKE avec un mécanisme de détection de changements nommé CuSum (*CUMulative SUMmury*) [5]. En considérant un paramètre, la moyenne par exemple, Cusum permet de détecter un changement dans sa valeur et de décider si une correction est nécessaire ou non. Nous utilisons CuSum aussi bien pour sa réactivité que pour ses performances dans un contexte de suivi où la zone de déploiement compte de nombreux obstacles contraignant la cible à les éviter. En considérant le FKE, un changement de direction brusque peut être détecté en examinant le modèle de mesure Z_{t+1} . Si l'évolution des coordonnées est constante alors Z_{t+1} est un processus à moyenne nulle et bruit blanc. Sinon,

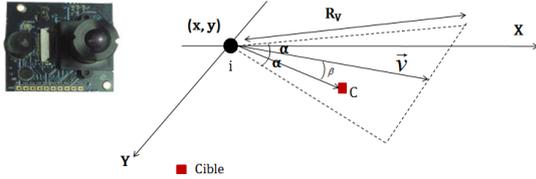


Figure 4: Imote2 et champ visuelle de la caméra

la moyenne va s'éloigner du zero. Le test est décrit comme suit :

$$g_{t+1} = g_t + S_{t+1} - v \quad (16)$$

Initialement $g_{t+1} = 0$, S_{t+1} est le processus de mesure normalisé, il est décrit comme suit :

$$S_{t+1} = \frac{Z_{t+1} - H_{t+1}\hat{X}_{t+1|t}}{\sqrt{(P_{t+1|t} + Q_{wt})H_t^2 + Q_{vt}}} \quad (17)$$

Dans l'équation 16, le paramètre v est très important, il permet d'éviter les fausses alarmes. Sa valeur est soustraite à chaque itération, plus cette valeur est petite, plus le test est sensible aux fluctuations. h représente le seuil de déclenchement de l'alarme. En effet, un changement de direction est détecté lorsque $g_{t+1} > h$. De même que pour v , plus la valeur de h est petite plus le test est sensible, il faut donc choisir ces valeurs en fonction de l'application. Cette condition est considérée comme la condition d'arrêt du test CuSum. Après le déclenchement de l'alarme la valeur de g_{t+1} est remise à zero.

4.5 Sélection du capteur suivant

Une fois que les futures coordonnées de la cible sont calculées par le capteur, ce dernier sélectionne parmi ses voisins le prochain qui lui succédera dans sa tâche. Pour être sélectionné, un capteur doit satisfaire le *Target in Sector Test* [10]. Ce dernier vise à vérifier si la cible se trouve à la portée visuelle du CC en calculant la distance entre chaque capteur voisin et les coordonnées prédites. Il est décrit comme suit :

$$\begin{cases} d_{iC} \leq R_v \\ \beta \in [-\alpha, \alpha] \end{cases} \quad (18)$$

Où d_{iC} est la distance entre la cible et le capteur ; β est l'angle entre $i\vec{C}$ et \vec{v} ; \vec{v} est le vecteur qui divise en deux le champ visuel de la caméra. La figure 4 illustre un exemple de champ visuelle d'un capteur caméra. Le capteur qui a la plus petite distance sera sélectionné. Le capteur courant l'activera en envoyant un message de **réveil**. Ce message contient l'identifiant du capteur émetteur. Chaque coordonnée calculée par les CCs est envoyée au puits qui se chargera de reconstituer la trajectoire complète de la cible.

5. SIMULATIONS ET ÉVALUATION DE PERFORMANCES

Pour évaluer les performances de PMT^2 nous avons réalisé des simulations sur NS-2[1]. Les valeurs des paramètres sont choisis en fonction des caractéristiques réelles des capteurs multimédia. Nous avons comparé notre solution à trois autres : 1)BASIC, un schéma classique où tous les capteurs sont toujours en état actif, 2) OCNS [8], un algorithme de

suivi où le réseau est organisé en cluster, avec un modèle probabiliste implémenté pour activer les capteurs qui participeront au processus de suivi de cible, et 3) PTA [3], un autre algorithme prédictif basé sur le Filtre de Kalman classique pour les modèles linéaires. Deux métriques sont utilisées pour réaliser cette comparaison : la précision du suivi et la consommation d'énergie.

5.1 Précision du suivi

La précision du suivi est une des métriques principales de l'évaluation des algorithmes de suivi. La figure 5(a) montre les résultats obtenus. PMT^2 obtient les meilleures performances, elle atteint jusqu'à 95% de précision pour 30 CC, ce qui représente 1 point chaque 5.26 mètres. Nous expliquons ces résultats par l'efficacité du mécanisme de prédiction. En effet, le couplage FKE et CuSum permet de capturer des trajectoires de cible réalistes. Pour 30 CCs, PMT^2 améliore la précision de 30% comparé à BASIC, de 45% comparé à OCNS et de 35% comparé à PTA.

De plus, contrairement aux autres solutions où le déploiement des capteurs se fait de manière aléatoire, nous utilisons dans notre solution un algorithme de déploiement W-VFA pour améliorer les performances.

5.2 Consommation d'énergie

Pour évaluer l'énergie consommée durant le processus de suivi, nous comptabilisons le coût énergétique des différentes tâches telles que l'activation des capteurs, les traitements et les échanges de messages entre capteurs. La figure 5(b) illustre les résultats obtenus. Nous observons que la solution BASIC est irréaliste. En effet, l'état d'activité permanent des capteurs provoque une consommation énergétique excessive. PMT^2 obtient de meilleures performances que OCNS. Grâce à l'efficacité du modèle prédictif seuls les capteurs se trouvant sur la trajectoire de la cible sont activés. PTA et PMT^2 ayant le même principe de collaboration, ils ont approximativement le même coût énergétique.

6. CONCLUSION

Dans cet article nous avons traité le problème de suivi de cible mobile dans les réseaux de capteurs multimédia sans fil. Nous avons pour cela proposé PMT^2 , un algorithme prédictif qui implémente une solution pour chaque étape du suivi. L'étude de performances a démontré l'efficacité de cette solution aussi bien en terme de précision du suivi qu'en consommation énergétique. Cette étude a aussi permis de prouver l'importance d'une stratégie de déploiement. En effet, l'utilisation de ce type de mécanisme améliore sensiblement les performances des algorithmes de suivi.

7. REFERENCES

- [1] <http://www.isi.edu/nsnam/ns/>.
- [2] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1-4. IEEE, 2008.
- [3] I. Boulanouar, S. Lohier, A. Rachedi, and G. Roussel. Pta : A predictive tracking algorithm in wireless multimedia sensor networks. In *Global Information*

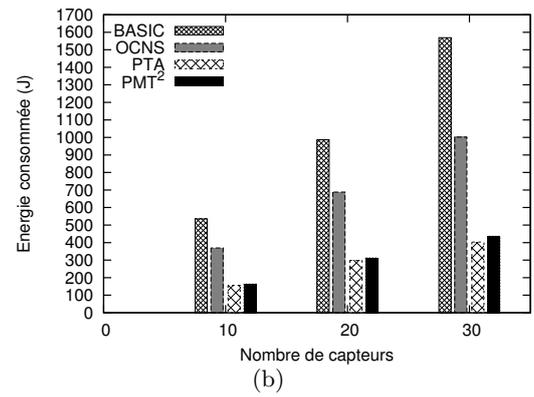
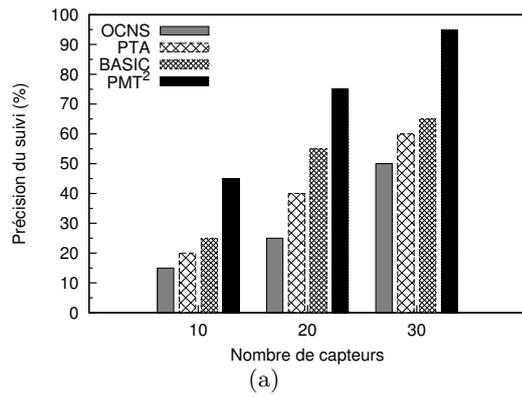


Figure 5: Évaluation de performances

Infrastructure Symposium, 2013, pages 1–6. IEEE, 2013.

INFOCOM, San Francisco, California, USA, 2003.

- [4] 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.
- [5] F. Gustafsson. *Adaptive filtering and change detection*, volume 1. Wiley New York, 2000.
- [6] 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.
- [7] C. Lee, D. Shin, S. W. Bae, and S. Choi. Best and worst-case coverage problems for arbitrary paths in wireless sensor networks. *Ad Hoc Networks*, 11(6) :1699 – 1714, 2013.
- [8] L. Liu, X. Zhang, and H. Ma. Optimal node selection for target localization in wireless camera sensor networks. *Vehicular Technology, IEEE Transactions on*, 59(7) :3562–3576, 2010.
- [9] 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 2011), 2011 IEEE*, pages 1–6. IEEE, 2011.
- [10] 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*, 15(2) :216–225, 2010.
- [11] 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*, 19, 2013.
- [12] . Y. Zhong, S. Lu, and L. Zhang. Peas : A robust energy conserving protocol for long-lived sensor networks. In *IEEE International Conference on Distributed Computing Systems, ICDCS*, Providence, USA, 2003.
- [13] Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In