SLAM visuel avec détection et suivi d'objets mobiles par une approche de segmentation/classification

David Marquez-Gamez, Michel Devy

SLAM visuel avec détection et suivi d'objets mobiles par une approche de segmentation/classification

David Márquez-Gámez Michel Devy

CNRS ; LAAS ; 7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; Toulouse, France
{dmarquez, michel}@laas.fr

Résumé

Cet article présente un système conçu pour traiter ensemble du SLAM, et de la Détection et du Suivi d'objets mobiles (DATMO), en exploitant uniquement la vision. Le but est de produire depuis des données visuelles, une description exploitable par un robot mobile, d'une scène dynamique : modélisation du monde statique, localisation du robot dans ce monde, et comment les autres objets mobiles s'y déplacent. Une approche combinant Segmentation par Clustering et Classification permet de détecter d'une part des points statiques pris en compte dans le SLAM visuel, et d'autre part des groupes de points mobiles exploités pour détecter et suivre les composantes dynamiques de la scène. L'approche globale est évaluée sur des bases d'images acquises en milieu urbain.

Mots-clés : SLAMMOT, Objets mobiles, Détection, Suivi, Segmentation, Clustering.

Abstract

In this paper, we present a system to solve the SLAM with Detection and Tracking of Moving Objects problem (SLAM-MOT) with the aid of vision. The goal is to achieve a convenient description of the visually perceived dynamic scene : how the static world is, where the robot is in this world, and how other existing bodies move. A Clustering-Classification method is proposed to detect static points, to be used by the SLAM algorithm and moreover to detect and cluster the moving ones in order to detect and track moving objects. The overall approach is evaluated with real data acquired in an urban environment.

Keywords : SLAMMOT, Moving Objects, Detection, Tracking, Clustering.

1 Introduction

Les méthodes SLAM (pour *Simultaneous Localization and Mapping*) permettent d'estimer à la fois la trajectoire d'un véhicule et la structure du monde à partir de données acquises depuis un capteur embarqué. Une hypothèse forte souvent considérée dans ces approches, est que le monde est statique, donc que les éléments modélisés dans la carte SLAM sont statiques. Cela a deux avantages : les positions de ces éléments peuvent être estimées de manière incrémentale, et ils peuvent être exploitées comme des références fixes (d'où le nom *amers*) pour estimer la position du robot dans le monde.

Dans cet article, nous considérons qu'il peut exister des objets mobiles dans le monde perçu par le robot ; malgré cela, nous voulons résoudre le problème du SLAM, et dans le même temps, estimer l'état de chaque objet mobile. Cela nécessite de traiter aussi de la détection et du suivi de ces objets mobiles, tout en conservant la capacité de détecter une nombre suffisant d'amers fixes afin de pouvoir localiser le robot. Nous avons développé un système complet qui traite des deux fonctions SLAM d'une part, et Détection et Suivi d'objets mobiles (souvent appelé DATMO, pour *Detection And Tracking of Mobile Objects*) d'autre part, en exploitant uniquement la vision : nous appelons ce système *BiCam SLAM with DATMO* car nous exploitons deux caméras montées sur notre robot.

La fonction DATMO nécessite de résoudre deux problèmes de nature très différente : la détection *Moving Objects Detection* (MOD) et le suivi, ou *Moving Objects Tracking* (MOT). Le problème MOT est résolu par des approches d'estimation. Mais, afin de rester dans la zone d'observabilité complète du monde 3D par un système BiCam [1], les objets mobiles seront uniquement suivis par le MOT dans la zone proche du robot, alors que les amers statiques pourront être extraits pour le SLAM dans tout le champ de vue (donc jusqu'à l'infini). La fonction MOD traite de la détection *initiale* des objets mobiles : la difficulté vient de l'impossibilité d'analyser de manière exhaustive le contenu de chaque image à une fréquence élevée (typiquement 30Hz). Nous proposons une méthode active, qui exploite les connaissances disponibles afin de focaliser l'analyse sur chaque image, sur les zones les plus probables dans lesquelles un objet mobile peut apparaître. Cette approche rapide et efficace permet non seulement de détecter et suivre des objets mobiles, mais encore d'en identifier la nature.

Un rapide état de l'art du SLAM-MOT est d'abord donné en section 2. Nous présenterons en section 3, comment est représenté un objet mobile à sa première détection. La section 4 décrit l'approche proposée *BiCam SLAM with DATMO*. La section 5 montre des résultats expérimentaux sur des séquences d'images acquises en milieu urbain. Enfin, la section 6 résume nos contributions et propose quelques perspectives à ce travail.

2 Etat de l'art sur le SLAM-MOT

La fonction désignée par "cartographie et localisation simultanées, avec suivi d'objets mobiles" (SLAM-MOT) combine un module de SLAM, prévu normalement pour s'exécuter en milieu statique, avec un module DATMO en charge de détecter et suivre des objets mobiles.
Le travail fondateur sur cette problématique du SLAM dans un environnement dynamique, a été présenté par B.Wang [2]. C'est une méthode classique de SLAM 2D fondé sur des données télémétriques laser, mais qui prend en compte la présence d'objets mobiles. Ces objets sont détectés en segmentant les parties de chaque coupe-laser qui ne s'apparient pas avec la carte courante. Comme ils ne peuvent servir pour estimer la position du robot dans l'environnement, les objets détectés mobiles sont laissés en dehors de la carte. Le système complet, fondé sur des télémètres laser 2D longue portée et grande vitesse, a permis de réaliser du SLAM sur de grandes surfaces à des vitesses élevée avec des fermetures de boucle de grande taille, le tout en milieu urbain avec un trafic dense.
D'autres approches intéressantes ont été proposées. Citons les travaux de Agrawal et al.[3] via des cartes denses de disparité produites par un banc stéréo calibré, une méthode RANSAC robuste appliquée pour déterminer la transformation rigide entre deux positions successives du capteur stéréo, et une approche de type *Inverse Perspective mapping* exploitée pour trouver ensuite les zones de l'image qui ont des mouvements propres indépendants du mouvement du capteur. Dans [4], un objet mobile est prédéfini et suivi en 3D, en parallèle avec une méthode de SLAM monoculaire ; cette approche ne propose pas de solution pour la détection des objets mobiles, et exploite pour suivre l'objet, une méthode basée modèle, uniquement adaptée pour suivre une classe d'objets (véhicules). Migliore et al. [5] ont proposé une approche intégrant SLAM et MOT, dans laquelle les objets mobiles sont détectés par un simple test statistique, puis suivis par des modules MOT monoculaire séparés. G.Gaté et al [6] ont proposé un système multi-capteur permettant de rendre plus robuste l'étape de détection : l'objet mobile est d'abord détecté dans les données laser, puis sa présence est confirmée par classification, dans les données visuelles.
Sola [7] a proposé une analyse d'observabilité pour la détection et le suivi d'objets mobiles depuis un capteur visuel embarqué. Il conclut que seul un système de type BiCam (deux caméras) permet de discriminer entre mouvements des objets et mouvement propre des caméras. Il propose une méthode pour détecter des points mobiles ; un suivi est lancé séparément pour chaque point ainsi détecté, représenté dans un repère robot-centré.
Lin et al. [8] ont proposé récemment une approche de SLAM-MOT par stéréovision, dans laquelle le vecteur d'état mis à jour par le SLAM est "augmenté" avec les états des objets mobiles qui ont été détectés. Ils ont montré que le MOT couplé ainsi au SLAM, permettait d'améliorer les performances du SLAM. Ils exploitent une stratégie identique à l'approche BiCam [1] ; les deux caméras sont traités comme des capteurs indépendants, dont les observations permettent de mettre à jour le vecteur d'état par EKF.

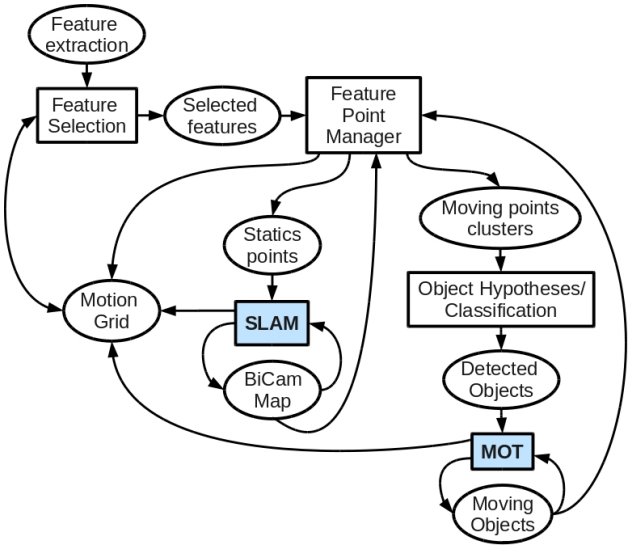

FIG. 1 – Notre système *BiCam SLAM with DATMO*.

Tous ces auteurs ont proposé des approches originales pour traiter du SLAM en présence d'objets mobiles. Mais les approches proposées apparaissent comme la juxtaposition des fonctions SLAM et MOT, avec partage minimal d'informations. Il existe généralement peu d'interactions entre *Localization and Mapping* d'un côté, et *Detection and Tracking* de l'autre.

3 Modèle des objets mobiles et représentation de l'environnement.

Dans cette section, nous décrivons comment nous définissons et nous représentons le modèle initial d'un objet mobile, modèle construit lors de sa première détection. Ce modèle sera ensuite exploité dans l'approche *BiCam with DATMO* décrite dans la section suivante.

3.1 Modèle des objets mobiles

Un simple point est représenté en $3D$ dans l'espace euclidien ; un tel point, s'il est dynamique, devra être suivi dans les images acquises depuis le robot. Ces points $\mathbf{o}_j^{\mathcal{O}_i}$,sont définis par leurs positions et leurs vitesses linéaires.
Un objet $\mathcal{O}_i$, est défini par un ensemble de points connexes qui ont des mouvements cohérents en $3D$ (vitesses très proches). Ces points sont rigidement liés, ce qui définit la structure de l'objet. Cette structure, caractéristique de l'objet, peut être décrite dans un repère propre à l'objet, par l'ensemble des points acquis sur la surface de l'objet. Un objet mobile conserve sa structure rigide, mais la position de son repère évolue dans le temps : donc, $\mathbf{o}_j^{\mathcal{O}_i} = \mathbf{o}_j^{\mathcal{O}_i}(t)$ et $\mathcal{O}_i = \mathcal{O}_i(t)$. Par la suite, nous omettons de préciser le temps (t) dans nos équations. Nous écrivons en conséquence les représentations initiales des états des *points mobiles* et des *objets mobiles* comme suit :

$$\mathbf{o}_j^{\mathcal{O}_i} = \begin{bmatrix} \mathbf{m}_j^{\mathcal{O}_i} \\ \mathbf{v}_j^{\mathcal{O}_i} \end{bmatrix}, \quad \mathcal{O}_i = \begin{bmatrix} \mathbf{o}_1^{\mathcal{O}_i} \\ \vdots \\ \mathbf{o}_n^{\mathcal{O}_i} \end{bmatrix}, \tag{1}$$

où $\mathbf{o}_j^{\mathcal{O}_i}$, est l'état d'un *point mobile j*, défini par sa position, $\mathbf{m}_j^{\mathcal{O}_i}$, et sa vitesse linéaire $\mathbf{v}_j^{\mathcal{O}_i}$, respectivement. Un ensemble de points mobiles ($\mathbf{o}_j^{\mathcal{O}_i}$), définit un *objet mobile i* ($\mathcal{O}_i$). Chaque point mobile peut être exprimé dans le repère monde *W* ($\mathbf{o}_j^{\mathcal{O}_i W}$), le repère robot *R* ($\mathbf{o}_j^{\mathcal{O}_i R}$) ou encore le repère caméra *C* ($\mathbf{o}_j^{\mathcal{O}_i C}$), en exploitant les transformations de repères usuelles.

3.2 Représentation de l'environnement.

Dans une fonction SLAM classique, le robot est localisé grâce à des observations sur des amers statiques, cela afin de garantir une bonne localisation. Les objets mobiles sont ignorés dans ce processus SLAM dédié à la localisation du robot. Les observations que le robot peut avoir sur ces objets mobiles ne doivent pas perturber la fonction SLAM, et donc le modèle du robot lui-même.
Cette approche conduit à construire des représentations des objets mobiles totalement indépendantes de la carte SLAM : on ne considére pas les corrélations entre les états des objets mobiles, du robot et de la carte SLAM. La seule relation entre les états du robot et d'un objet mobile sont décrites par les observations courantes de cet objet depuis le robot, et par un modèle dynamique choisi pour cet objet. Finalement la carte construite par notre système *BiCam SLAM with DATMO* est faite de la carte SLAM construite par l'approche BiCam et par un ensemble de vecteurs stochastiques, mis à jour par EKF, un vecteur pour chaque objet détecté. Donc :

$$X^T = \begin{bmatrix} C_L^T & C_R^T & L_1^T & \dots & L_M^T \end{bmatrix} \quad (2)$$

$$\mathcal{O}_i^T = \begin{bmatrix} \mathbf{o}_1^{\mathcal{O}_i T} \dots \mathbf{o}_n^{\mathcal{O}_i T} \end{bmatrix} \quad (3)$$

où dans l'équation 2, $C_{L/R}^T = [\mathbf{r}_{L/R}^T, \mathbf{q}_{L/R}^T] \in \mathbb{R}^7$, est l'état de la caméra, respectivement gauche et droite, donné par sa position et un quaternion pour l'orientation, plus l'ensemble des amers $L_j = i_j \in \mathbb{R}^6$ (en IDP) ou $L_j = p_j \in \mathbb{R}^3$ (en euclidien) et, dans l'équation 3, $\mathbf{o}_j^{\mathcal{O}_i T} = (\mathbf{m}_j^{\mathcal{O}_i}, \mathbf{v}_j^{\mathcal{O}_i})$, est l'état d'un point mobile défini par sa position et sa vitesse. Le modèle de mouvement d'un tel point est un modèle à vitesse constante.

4 Description du système complet

Nous proposons dans cet article un système complet pour traiter les fonctions SLAM et MOT sur un robot équipé d'un banc de stéréovision (ici traité comme un système BiCam, deux caméras indépendantes), évoluant dans un environnement dynamique. Les principales fonctions considérées dans ce système sont présentées en Figure 1.

4.1 Sélection des points d'intérêt

La fonction *Détection des Points d'intérêt* doit tirer profit de toutes les connaissances déjà issues des images précédentes, ou d'informations contextuelles liées à l'application. Aussi à chaque itération de l'algorithme de détection, des points d'intérêt doivent être d'abord extraits : cette extraction est guidée par une *Grille de Mobilité*, construite

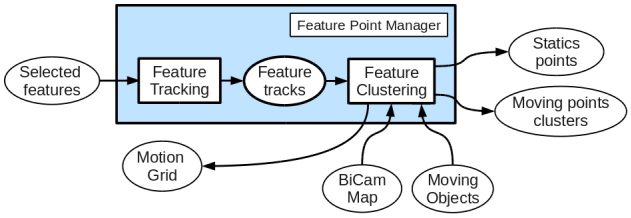

FIG. 2 – La fonction *Caractérisation des points d'intérêt*.

comme une grille d'occupation, sur la base d'une discrétisation de l'espace image en cellules (typiquement tesselation de l'image en pavés 10x10). Cette grille donne la probabilité de trouver en une telle cellule, un point d'intérêt mobile.
Initialement, en l'absence d'informations contextuelles, cette probabilité est uniforme ; ensuite la grille est mise à jour selon différents critères qui sont présentés en section 4.2. Les nouveaux points sont donc extraits dans les cellules qui ont la plus forte probabilité de présence d'un objet mobile. Vu l'initialisation de la grille, ces points sont d'abord uniformément répartis dans l'image. Ensuite au fil des déplacements de notre robot et de l'exécution des différentes fonctions de notre système, les points sont remplacés soit car ils ont été perdus par la fonction de Suivi des points, soit parce qu'ils ont été classés Statique ; les points qui sont classés Dynamique sont par contre conservés, associés à des groupes de points qui sont suivis de manière continue dans la suite des images.

4.2 Grille de Mobilité

Comment instancier et mettre à jour la *Grille de Mobilité* ?
A chaque cycle de la fonction de détection, il est important (1) de détecter de nouveaux points sur un objet mobile qui a déjà été détecté dans les images précédentes, cela pour segmenter toute la zone image qui lui correspond, et (2) de détecter de nouveaux objets mobiles qui sont entrés dans le champ de vue de la caméra, donc d'extraire de nouveaux points à classer Statique ou Dynamique dans les zones de l'image où la probabilité de présence d'un objet mobile est la plus élevée.
Pour sélectionner de manière active ces points d'intérêt qui vont être suivis avant d'évaluer la probabilité qu'ils correspondent à des points statiques ou dynamiques de la scène, nous exploitons cette *Grille de Mobilité* introduite dans [9]. Cette grille se comporte comme une grille d'occupation en robotique (*occupancy grid*), au sens où les plus fortes probabilités correspondent potentiellement aux zones mobiles (occupées) et les plus basses aux zones statiques (libres) ; il convient de focaliser la détection de nouveaux points mobiles dans les zones de la grille qui ont les plus fortes probabilités.
Cette grille est donc donnée en entrée de la fonction *Sélection de Points d'intérêt* qui extrait de nouveaux points d'intérêt ; ces points seront ensuite caractérisés comme Statique ou Dynamique, après les avoir suivis pendant N_{im} images successives. Une fois ces points caractérisés, les zones dans lesquelles ils se trouvent après ces N_{im} images, peuvent hériter de leurs labels Statique ou Dynamique.

Il faut donc mettre à jour les probabilités de ces zones dans la *Grille de Mobilité*. Elles sont modifiées selon des fonctions gaussiennes bidimensionnelles centrées sur les points qui ont été caractérisés à cette étape de la fonction de Détection. Pour un point en (u,v), cette gaussienne a des variances σ_u et σ_v qui sont fonctions de r, r étant la taille d'une cellule de la grille (ici 10 pixels). L'équation 4 montre l'évlution spatio-temporelle des probabilités pour des cellules qui sont mobiles, statiques ou inconnues dans la grille. La probabilité qu'une cellule recouvre un objet mobile dans l'image courante, est inversément proportionnelle à sa distance au point classé Dynamique le plus proche ; la décroissance est obtenue grâce à la fonction gaussienne centrée sur ce point. Et donc, dans le temps, cette probabilité évolue comme l'inverse de la valeur établie dans le premier cas de l'équation 5 :

$$p(u,v,t) = \Theta_i^{pm} = \sum_{u,v \in cell} (\alpha(u,v)_{t-1} + G(\mu_u, \mu_v, \sigma_u, \sigma_v)) \tag{4}$$

où $\alpha(u,v)$ désigne la probabilité dans une cellule centrée en (u,v). L'évolution dans le temps de cette probabilité est donnée par :

$$\alpha(u,v)_t = \begin{cases} 1 - p(u,v)_t & \text{if } \alpha(u,v)_{t-1} = moving \\ p_0 & \text{if } \alpha(u,v)_{t-1} = static \end{cases} \tag{5}$$

4.3 Caractérisation des Points d'intérêt

Nous devons classifier chaque primitive visuelle comme dynamique ou statique, avant de l'exploiter soit comme amer qui sera rajouté dans la carte SLAM pour localiser le robot, soit comme observation qui permettra au module MOT de caractériser un objet mobile. Nous exploitons la méthode proposée dans [9] afin de détecter un objet mobile : nous appelons cette procédure *Caractérisation des Points d'intérêt*. Le but de cette méthode est de discriminer les points comme Statique ou Dynamique et de regrouper les points dynamiques qui pourraient appartenir aux mêmes objets mobiles. Le processus de Caractérisation des Points d'intérêt est présenté en Figure 2.

A chaque itération de la fonction de détection, N_p points d'intérêt ont été sélectionnés de manière active en exploitant la *Grille de Mobilité* : N_p = 150 dans nos tests. Ces points sont suivis par la fonction KLT fondée sur la corrélation et l'optimisation des déformations des fenêtres de corrélation. Supposons d'abord que l'environnement est statique ; la fonction KLT boucle sur N_{Im} images successives. Ce temps de suivi est appelé temps de pistage : $N_{im} = 6$ dans nos évaluations ; c'est le temps nécessaire pour construire une *piste* pour chaque point suivi. Une piste donne les positions successives de ce point, et permet de calculer sa vitesse apparente. Certains points sont perdus durant ces N_{im} images ; ils ne sont pas remplacés de suite. L'ensemble des points suivis sera reconstruits uniquement après ces N_{im} images, surtout après mise à jour de la *Grille de Mobilité*. La fonction KLT s'exécute donc en continu, mais en remplaçant l'ensemble de points suivis après chaque période de pistage.

Seuls les points suivis par la fonction KLT sont sélectionnés pour être groupés par la méthode de clustering a contrario inspirée des travaux présentés en [10]. Cette fonction génére une liste de points statiques et une liste de M_i clusters ou objets mobiles potentiels. Chaque cluster est défini par une liste de points classés Dynamique, par le barycentre de ces points, la vitesse apparente (moyenne des vitesses apparentes de ces points) et une boîte (rectangle) englobant ces points. La fonction de Clustering prend en entrée des informations issues de la fonction SLAM, afin de compenser le mouvement de la caméra quand cela est possible (uniquement sur le plan du sol). Elle prend aussi en entrée des informations venant de la fonction MOT, requises pour mettre à jour la *Grille de Mobilité*. En effet, les caractéristiques (positions, vitesses et tailles apparentes dans l'image courante) de chaque chaque objet suivi par la suite, sont exploitées pour évaluer la probabilité de trouver des points Dynamique dans chaque cellule de la grille.

4.4 Génération des hypothéses : approche *Classification*

Des hypothéses sur la présence d'objets mobiles dans la scène, sont issues des résultats d'un détecteur fondé sur une approche *Classification*. Le système que nous proposons est indépendant du choix de la méthode de classification [11], [12], que cela soit les attributs attachés à une région à classifier, ou le choix du classifieur lui-même. Dans nos expérimentations, pour des raisons pratiques, nous exploitons un classifieur AdaBoost [12], fondé sur l'idée qu'une combinaison de classifieurs faibles permet d'obtenir un classifieur fort et efficace, cela après une phase d'apprentissage supervisée, donc sur une base d'exemples positifs et négatifs. Ici le classifieur AdaBoost a été entraîné de manière spécifique pour classer une région, soit comme *Personne*, soit comme *Véhicule*.

Chaque groupe notée M_i fournie par la fonction *Caractérisation des Points d'Intérêt*, est associée à une région d'intérêt dans l'image ; une telle région est un *Objet Potentiel*. Le classifieur sera uniquement appliqué sur ces régions d'intérêt afin d'associer une probabilité à chaque *Objet Potentiel*, cela pour éviter la phase de parcours spatial multi-échelles de l'image, étape lourde des approches de détection par classification.

Donc, la région d'intérêt M_i est décomposée en n régions différentes, $(M_i^k)_{1 \leq k \leq n}$; chacune de ces n régions est donnée à l'algorithme de classification, qui lui associe une valeur λ_i^k obtenue par une somme pondérée des résultats des N classifieurs faibles, $(w_l)_{1 \leq l \leq N}$

$$\lambda_i^k = \sum_{1 \leq l \leq N} \alpha_l w_l(M_i^k) \tag{6}$$

Finalement, pour chaque groupe M_i, un score de classification global est calculé de la manière suivante :

$$\Theta_i^c = \max_{1 \leq k \leq n} \lambda_i^k \tag{7}$$

Cette méthode de classification est seulement entraînée pour calculer des probabilités pour une région donnée de correspondre à une image d'un objet de type *Personne* ou *Véhicule* ; cela n'est pas suffisant pour conclure sur la détection ou le suivi d'un objet mobile dans la scène. Cepen-

dant, chaque groupe M_i se voit attribuer un score de classification Θ_i^c.
Pour chaque groupe ou objet potentiel i à l'instant k, ce score de classification est indépendant de la probabilité que la zone de l'image dans laquelle ce groupe se trouve, contienne un objet mobile, probabilité déjà disponible par l'approche *Grille de Mobilité*. Il reste à traiter de la fusion de ces deux informations.

4.5 Evaluation finale des hypothèses sur la détection des objets mobiles

Finalement, afin de détecter correctement les objets mobiles, l'estimée fournie par l'approche *Grille de Mobilité* doit être fusionnée avec l'estimation issue de l'approche *Classification*. Nous disposons de deux estimées différentes de la même probabilité pour une région, $P(\mathcal{O}_i|A_j)$, la probabilité de l'événement $\mathcal{O}_i$ (cette région correspond à un objet mobile) connaissant A_j. Ces deux estimées ont été obtenues par des méthodes qui sont elles-même incertaines. Aussi, chacune de ces estimées peut être réécrite de la manière suivante :

$$P(\mathcal{O}_i|A_1) = \Theta_i^{pm} \tag{8}$$

$$P(\mathcal{O}_i|A_2) = \Theta_i^{c} \tag{9}$$

où $P(\mathcal{O}_i|A_1)$ est l'estimation donnée par l' approche *Grille de Mobilité* et $P(\mathcal{O}_i|A_2)$ est l'estimation fournie par l'approche *Classification*. En faisant l'hypothése que A_1 and A_2 sont des événements indépendants, la règle de fusion suivante est obtenue des opérations de base sur les probabilités :

$$\begin{aligned} &P(\mathcal{O}_i) = P(\mathcal{O}_i \cap (A_1 \cup A_2)) + P(\mathcal{O}_i \cap (\overline{A_1 \cup A_2})) \\ &= P(\mathcal{O}_i|A_1)P(A_1)P(\overline{A_2}) + P(\mathcal{O}_i|A_2)P(\overline{A_1})P(A_2) \\ &+ P(\mathcal{O}_i|A_1 \cap A_2)P(A_1)P(A_2) + P(\mathcal{O}_i|\overline{A_1} \cap \overline{A_2})P(\overline{A_1})P(\overline{A_2}) \end{aligned} \tag{10}$$

où $P(A_j)$ désigne la probabilité fournie par le sous-système j (ici approches Grille de Mobilité ou Classification) de retourner une mauvaise estimation. $P(\mathcal{O}_i|\overline{A_1} \cap \overline{A_2})$ peut être approximée par ξ, la probabilité de classification a priori de la région dans une classe donnée. Alors la règle de fusion entre les deux approches est approximée par une somme pondérée :

$$\begin{aligned} &\Phi_{\mathcal{O}_i} = \Theta_i^{pm} P(A_1)P(\overline{A_2}) + \Theta_i^{c} P(\overline{A_1})P(A_2) \\ &+ \frac{\Theta_i^{pm} P(A_1) + \Theta_i^{c} P(A_2)}{P(A_1) + P(A_2)} P(A_1)P(A_2) + \xi P(\overline{A_1})P(\overline{A_2}) \end{aligned} \tag{11}$$

A ce niveau, l'estimée finale de la probabilité de détection est disponible. Le résultat final de la fonction MOD est donné par un seuillage sur cette probabilité. Donc, pour chaque région d'intérêt M_i,

$$M_i = \begin{cases} \textit{Moving Object} & \text{if } \Phi_{\mathcal{O}_i} \geq \text{threshold} \\ \textit{NOT Moving Object} & \text{if } \Phi_{\mathcal{O}_i} < \text{threshold} \end{cases}$$

Notre stratégie de détection est donc fondée sur la fusion de deux algorithmes de natures différentes, d'une part la *Grille de Mobilité* et l'*approche fondée sur la classification*. Comme le second algorithme dépend du processus d'apprentissage pour calculer une probabilité de classification sur certains objets (piétons, voitures...), on ne peut dire que cette méthode de détection d'objets mobiles est "generique".

4.6 Fonction MOT : suivi des objets mobiles

Création des objets mobiles depuis les groupes de points. Dès qu'un objet est détecté via un groupe de points d'intérêt dynamiques, une vecteur d'état lui est associé sous la forme d'une pdf qui sera mise à jour par EKF. Ce vecteur d'état suit donc une loi gaussienne $\mathcal{O} \sim \mathcal{N}\{\bar{\mathcal{O}}; \mathbf{P}_{\mathcal{O}}\}$. Pour sa création nous exploitons la paramétrisation par IDP, décrite dans [13]. L'estimée et la matrice de variances initiales des points dynamiques $\mathbf{m}_j^{\mathcal{O}_i}$, associés à cet objet $\mathcal{O}_i$, sont produits par la paramétrisation IDP définie dans la caméra gauche :

$$\mathbf{m}_j^{\mathcal{O}_i} = \mathbf{d}_L(C_L, \mathbf{h}_L, \rho) \tag{12}$$

où $\mathbf{d}_L(\cdot)$ est fonction de $C_L^T = [\mathbf{x}_L^T, \mathbf{q}_L^T]$ l'état de la caméra gauche, défini par sa position et son orientation, $\mathbf{h}_L \sim \mathcal{N}\{\mathbf{y}_L; \mathbf{R}\}$ l'observation du point dans la caméra gauche représentée par un vecteur gaussien, et $\rho \sim \mathcal{N}\{\bar{\rho}; \sigma_\rho^2\}$ l'inverse de la distance euclidienne entre C_L et le point. Après sa détection initiale dans l'image gauche, tous les paramètres de $\mathbf{m}_j^{\mathcal{O}_i}$ sauf ρ sont immédiatement observables, et leurs estimées et matrices de variances sont obtenues par inversion et linéarisation des fonctions d'observation.
Donc pour la position d'un point mobile $\mathbf{m}_j^{\mathcal{O}_i}$, l'estimée et la matrice de variances initiales sont donc :

$$\bar{\mathbf{m}}_j^{\mathcal{O}_i} = \mathbf{d}_L(C_L, \bar{\mathbf{h}}_L, \bar{\rho}) \tag{13}$$

$$\mathbf{P_{mm}} = \mathbf{D}_{L\mathbf{h}}\, \mathbf{R}\, \mathbf{D}_{L\mathbf{h}}^T + \mathbf{D}_{L\rho}\, \sigma_\rho^2\, \mathbf{D}_{L\rho}^T \tag{14}$$

où $\mathbf{R}$ est la matrice de rotation, tandis que les jacobiennes sont :

$$\mathbf{D}_{L\mathbf{h}} = \left.\frac{\partial \mathbf{d}_L}{\partial \mathbf{h}^T}\right|_{(\bar{C}_L, \bar{\mathbf{h}}, \bar{\rho})}, \quad \mathbf{D}_{L\rho} = \left.\frac{\partial \mathbf{d}_L}{\partial \rho^T}\right|_{(\bar{C}_L, \bar{\mathbf{h}}, \bar{\rho})} \tag{15}$$

L'estimée et la matrice de variances initiales pour la vitesse d'un point mobile $\mathbf{v}_j^{\mathcal{O}_i} \sim \mathcal{N}\{\bar{\mathbf{v}}_j^{\mathcal{O}_i}; \mathbf{P_{vv}}\}$ sont déterminées de manière heuristique comme dans [8].
Finalement, le vecteur d'état d'un objet mobile est donnée par la pdf définie par le couple :

$$\bar{\mathcal{O}}_i = \begin{bmatrix} \bar{\mathbf{m}}_j^{\mathcal{O}_i} \\ \bar{\mathbf{v}}_j^{\mathcal{O}_i} \end{bmatrix}, \quad \mathbf{P}_{\mathcal{O}_i} = \begin{bmatrix} \mathbf{P_{mm}} & 0 \\ 0 & \mathbf{P_{vv}} \end{bmatrix}, \tag{16}$$

Dès sa création, cet état est mis à jour par EKF avec l'observation issue de la caméra droite. La fonction d'observation pour la caméra droite est donnée par la fonction de projection, en fait le modèle pin-hole de la caméra, exprimé dans son propre repère par :

$$\mathbf{y}_R = \mathbf{h}_R(C_R, \mathbf{m}_j^{\mathcal{O}_i}) + v_R \tag{17}$$

où $v_R \sim \mathcal{N}\{0;\mathbf{R}\}$ est le bruit d'observation, supposé être un bruit gaussien blanc.

Donc, à partir de son observation dans la caméra droite, considérant la pose incertaine de cette caméra $C_R \sim \mathcal{N}\{\bar{C}_R;\mathbf{P}_{C_R}\}$, la *pdf* donnant l'état de l'objet mobile est mis à jour par EKF.

Le test de linéarité introduit par [14] est évalué après chaque exécution du filtre. Si ce test est positif, la position de l'objet peut être reparamétrée de IDP en coordonnées euclidiennes 3D.

Mise à jour des objets mobiles par compensation du mouvement du robot. Le modèle dynamique du robot exploite des mesures de déplacement données par odométrie. Ce modèle dynamique exploite donc le modèle de l'odométrie, pou le modèle unicycle de robot qui correspond à notre plateforme $R^+ = \mathbf{f}_R(R,\mathbf{u})$, où $\mathbf{u}$ est le vecteur de contrôle du robot, ou les mesures odométriques :

$$\mathbf{u} = \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{e} \end{bmatrix} = [\delta x, \delta y, \delta z, \delta\phi, \delta\theta, \delta\psi]^T \in \mathbb{R}^6 \qquad (18)$$

avec la *pdf* gaussienne $\mathbf{u} \sim \mathcal{N}\{\bar{\mathbf{u}};\mathbf{U}\}$. Dès que la position du robot a été mise à jour, les positions des objets mobiles doivent être transformées, pour être exprimées vis-à-vis de la nouvelle position du repère du robot. Donc l'équation $\mathcal{O}_i^+ = \mathbf{j}_\mathcal{O}(\mathcal{O}_i,\mathbf{u})$, exprime simplement un changement de coordonnées qui est spécifié à partir des mesures odométriques $\mathbf{u}^T = [\delta\mathbf{x}^T, \delta\mathbf{e}^T]$. Nous pouvons donc représenter les objets mobiles par :

$$\mathbf{m}_j^{\mathcal{O}_i+} = \mathbf{R}^T\delta(e)\cdot(\mathbf{m}_j^{\mathcal{O}_i} - \delta x) \qquad (19)$$

$$\mathbf{v}_j^{\mathcal{O}_i+} = \mathbf{R}^T\delta(e)\cdot\mathbf{v}_j^{\mathcal{O}_i} \qquad (20)$$

où $\mathbf{R}^T\delta(e)$ est une matrice de rotation calculée à partir des incréments de rotation exprimés en angles d'Euler $\delta(e)$. Donc en considérant ces équations de compensation appliquées aux positions et vitesses, la *pdf* représentant chaque objet mobile peut être mise à jour par EKF.

Estimation du mouvement propre des objets mobiles. Nous n'avons aucune observation directe sur le mouvement des objets mobiles. Des modèles de mouvement à vitesse constante fournissent des trajectoires plus ou moins lisses en fonction de la quantité de bruit pris en compte dans le système. Par exemple, le mouvement d'un objet de type *Personne* peut être considéré avec une vitesse constante dans le plan 2D du sol. S'il est supposé qu'il ne peut pas s'arrêter brutalement, il suffit de considérer un simple bruit gaussien assez faible. Mais un modèle de bruit plus fort devra être considéré si on veut prendre en compte des changements de dynamique brusques, de type *Stop and Go* ; mais ce bruit plus fort dans le même temps va rendre la trajectoire moins lisse et va tendre à éliminer la différence entre les états *Stop* et *Go*. Le modèle le plus satisfaisant est un modèle à deux modes, associés à des probabilités a priori de transition :

A chaque instant :
- *if* $\mathcal{O}_i$ *est mobile* — il reste mobile, mais il peut stopper.
- *if* $\mathcal{O}_i$ *est statique* — il reste statique, mais il peut démarrer.

A ce point, des bruits gaussiens de différentes amplitudes peuvent être introduits dans le système. Néanmoins pour l'instant dans notre système, pour des raisons pratiques, nos objets ont tous des modèles à vitesse constante. Donc le modèle dynamique de nos objets est donné par $\mathcal{O}_i^+ = \mathbf{f}_\mathcal{O}(\mathcal{O}_i,\boldsymbol{\omega})$, qui est un modèle à vitesse constante sans rotation donné par

$$\mathbf{m}_j^{\mathcal{O}_i+} = \mathbf{m}_j^{\mathcal{O}_i} + T_s\mathbf{v}_j^{\mathcal{O}_i} \qquad (21)$$

$$\mathbf{v}_j^{\mathcal{O}_i+} = \mathbf{v}_j^{\mathcal{O}_i} + \boldsymbol{\omega} \qquad (22)$$

où T_s est le temps d'échantillonnage du filtre et $\boldsymbol{\omega} = [\omega_x,\omega_y,\omega_z]^T \in \mathbb{R}^3$ est un bruit blanc gaussien, qui est une perturbation sur la vitesse $\boldsymbol{\omega} \sim \mathcal{N}\{0,\mathbf{Q}\}$.

4.7 BiCam SLAM

BiCam SLAM [1] vient du constat initial que les systèmes SLAM stéréo sont très myopes, puisqu'ils ne prennent en compte que les amers détectés dans le champ d'observabilité 3D du capteur stéréovision ; l'idée est de combiner SLAM stéréo et SLAM mono qui permet de prendre en compte des amers lointains, éventuellement sur la ligne d'horizon. Donc les images acquises par le capteur stéréo sont traitées comme un SLAM coopératif centralisé avec deux caméras indépendantes (mais synchronisées) : la caméra maître est responsable de la détection et de l'initialisation des amers, tandis que la seconde caméra, dite esclave, permet de réobserver la carte SLAM construite par la caméra maître. Se faisant, elle fournit un deuxième point de vue sur les amers. Lorsque ceux-ci se trouvent dans la zone d'observabilité 3D des deux caméras, la triangulation est faite et cet amer peut être reparamétrisé avec une représentation euclidienne.

La fonction *BiCam SLAM* réalise donc un SLAM coopératif, qui peut exploiter plus de deux caméras éventuellement non rigidement liées. Le coeur est la fonction EKF-SLAM exploitée pour mettre à jour une carte stochastique qui contient les états de toutes les caméras C_i et de tous les amers détectés jusque là L_j, $X^T = [C_1^T \ \ldots \ C_N^T \ L_1^T \ \ldots \ L_M^T]$. Le vecteur d'état d'une caméra C_i contient sa position et son orientation décrite par un quaternion $[C_i = (\mathbf{r}_i,\mathbf{q}_i) \in \mathbb{R}^7]$, tandis que l'état d'un amer L_j peut être paramétré soit en *inverse depth* $(L_j = i_j \in \mathbb{R}^6)$ soit en Euclidien $(L_j = p_j \in \mathbb{R}^3)$. Cette représentation doit être incrémentée et mise à jour à chaque acquisition d'images ; N_{obs} amers déjà connus sont traités par une fonction de recherche active, tandis que N_{new} nouveaux amers sont détectés et rajoutés à la carte. La complexité de cet algorithme augmente de manière linéaire en fonction du nombre de caméras et des paramètres N_{obs} and N_{new}, et de manière quadratique en fonction du nombre

d'amers dans la carte. Des capteurs proprioceptifs (odomètre, gyro) fournissent les estimées des déplacements effectués par les caméras. Les événements "mouvement caméra", "initialisation d'amers" et "mise-à-jour d'amers" sont traités comme dans le EKF-SLAM en sélectionnant le bloc adéquat dans la carte stochastique et dans la matrice de variance, puis en appliquant les modèles de mouvement du robot et d'observation de la caméra.

5 Résultats expérimentaux.

Nous présentons des résultats obtenus sur des séquences acquises depuis une plateforme Segway 4-roues, équipée avec un banc calibré de stéréovision intégrant des caméras de résolution 1024 x 768 et une base de 0.35 *m*. Notre méthode *BiCam SLAM with DATMO* est validée sur deux séquences acquises dans des environnements intérieur et extérieur, avec différents types d'objets mobiles.
Les figures 3 et 4 montrent les résultats, avec en colonne de gauche, des images significatives de ces séquences, et en colonne de droite, les vues de dessus de la carte 3D générée après traitement de ces images. Dans les images, nous superposons les points exploités par les fonctions SLAM et MOT : en *cyan*, les observations des amers mis à jour par le SLAM ; en *bleu*, les observations prédites des amers ; en *jaunes*, les observations de points mobiles mis à jour par le MOT. Les boîtes *rouge* englobent les groupes de points extraits sur les objets mobiles. Dans les deux cas, ces objets (des personnes en intérieur, un cycliste en extérieur) sont correctement détectés et suivis, et leurs positions sont bien estimées, comme cela peut être vu sur les vues de dessus.

5.1 Séquence acquise en intérieur.

Notre robot équipé de ces deux caméras orientées vers l'avant, exécute un mouvement en ligne droite dans un grand espace public (ici la salle Robotique du LAAS). L'odométrie est exploitée pour prédire la position du robot à chaque itération du notre méthode BiCam SLAM. Deux objets mobiles apparaissent dans la séquence : d'abord un piéton traversant le champ visuel de la gauche vers la droite, puis une personne poussant un caddy qui va de la droite vers la gauche.
La figure 3 montre les résultats de notre approche *BiCam SLAM with DATMO* sur cette séquence. Les deux personnes sont correctement détectées et suivies dans la séquence. Même dans cette situation dans laquelle les zones dynamiques occupent une large part des images, la carte stochastique peut être construite et mise à jour par la fonction BiCam SLAM, sans introduire dans la carte des points dynamiques mal classifiés ; de même la fonction MOT permet de suivre les objets dynamiques et de les positionner dans l'espace 3D.

5.2 Séquence acquise en extérieur.

Ensuite nous avons voulu tester notre approche en milieu extérieur, plus difficile du fait des conditions d'illumination. En ce cas, un banc stéréo calibré est fixé sur une poussette déplacée à la main en quasi ligne droite, sur les bords du *Canal du Midi*. La prédiction est obtenue par un modèle de mouvement à vitesse constante. Un seul objet

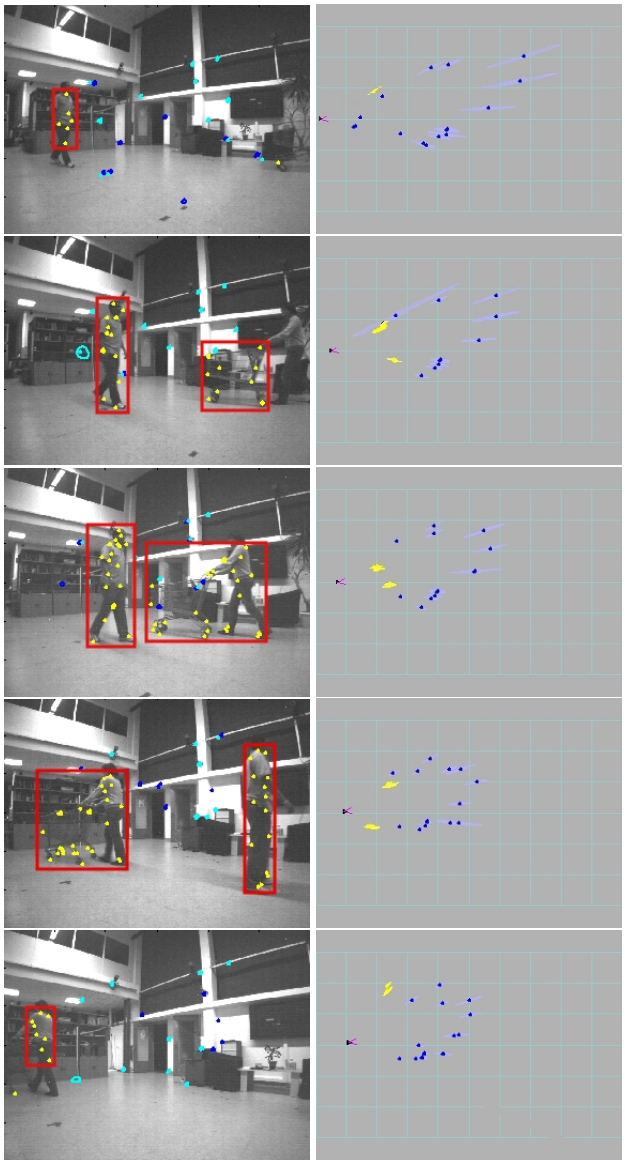

FIG. 3 – Test sur une séquence acquise en milieu intérieur.

mobile apparaît dans la séquence, un cycliste qui dépasse notre poussette, donc qui se dirige vers l'avant dans le sens positif de l'axe *z*.
La figure 4 montre ici aussi les résultats de l'approche *BiCam SLAM with DATMO*. Le cycliste est correctement détecté et suivi ; les fonctions SLAM, DATMO et MOT se comportent de manière satisfaisante.

6 Conclusions

Dans cet article nous avons présenté un système qui traite de manière efficace le problème de la navigation dans des environnements dynamiques, donc de la cartographie et localisation simultanées (SLAM) avec détection et suivi d'objets mobiles (DATMO). Ce système exploite uniquement la vision depuis deux caméras, selon le principe *BiCam*. La détection des objets mobiles est réalisée par une approche active, capable d'anticiper la possible apparition d'un objet mobile, en sélectionnant dans l'image courante,

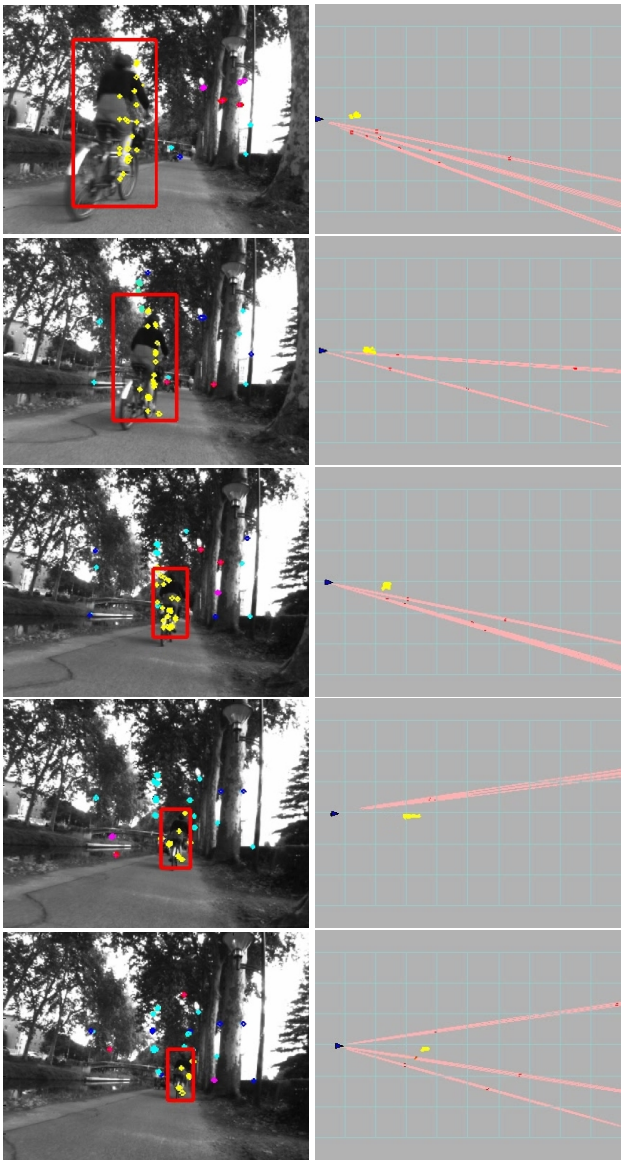

FIG. 4 – Test sur une séquence acquise en milieu extérieur.

les zones les plus intéressantes dans lesquelles détecter un tel objet. Il est nécessaire d'intégrer dans un même système de nombreuses fonctions pour exécuter le SLAM visuel et le DATMO. Nous avons montré comment ces fonctions interagissent. Des résultats sur des séquences d'images réelles, acquises depuis des caméras embarquées sur un robot, nous ont permis de prouver la pertinence de l'approche proposée de manière qualitative ; des évaluations quantitatives sont en cours.

Références

[1] J. Sola, A. Monin, M. Devy, and T. Vidal-Calleja, "Fusing monocular information in multicamera slam," *IEEE Trans. on Robotics*, vol. 24(5), pp. 958–968, 2008.

[2] C.-C. Wang, "Simultaneous localization, mapping and moving object tracking," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2004.

[3] M. Agrawal, K. Konolige, and L. Iocchi, "Real-time detection of independent motion using stereo," in *Proceedings IEEE workshop on visual motion*, 2005.

[4] S. Wangsiripitak and D. Murray, "Avoiding moving outliers in visual slam by tracking moving objects," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 375 –380.

[5] D. Migliore, R. Rigamonti, D. Marzorati, M. Matteucci, and D. Sorrenti, "Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments," in *In proceedings of International workshop on Safe navigation in open and dynamic environments application to autonomous vehicles*, May 2009.

[6] G. Gate, A. Breheret, and F. Nashashibi, "Fast pedestrian detection in dense environment with a laser scanner and a camera," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, april 2009, pp. 1 –6.

[7] J. Sola, "Towards visual localization, mapping and moving objects tracking by a mobile robot : a geometric and probabilistic approach," Ph.D. dissertation, Institut National Politechnique de Toulouse, Toulouse, February 2007.

[8] K.-H. Lin and C.-C. Wang, "Stereo-based simultaneous localization, mapping and moving object tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[9] D. L. Almanza-Ojeda, M. Devy, and A. Herbulot, "Active method for mobile object detection from an embedded camera, based on a contrario clustering," in *Informatics in Control, Automation and Robotics*, 2011, vol. 89, pp. 267–280.

[10] T. Veit, F. Cao, and P. Bouthemy, "Space-time a contrario clustering for detecting coherent motions," in *Robotics and Automation, 2007 IEEE International Conference on*, april 2007, pp. 33 –39.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05, Washington, DC, USA, 2005, pp. 886–893.

[12] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2002.

[13] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular slam," *IEEE Trans. on Robotics*, vol. 24, 2008.

[14] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth to depth conversion for monocular slam," in *International Conference on Robotics and Automation*, 2007, pp. 2778–2783.