



HAL
open science

SLAM visuel monoculaire par caméra infrarouge

Aurélien Gonzalez, Michel Devy, Joan Solà

► **To cite this version:**

Aurélien Gonzalez, Michel Devy, Joan Solà. SLAM visuel monoculaire par caméra infrarouge. ORASIS - Congrès des jeunes chercheurs en vision par ordinateur, INRIA Grenoble Rhône-Alpes, Jun 2011, Praz-sur-Arly, France. inria-00610516

HAL Id: inria-00610516

<https://inria.hal.science/inria-00610516>

Submitted on 22 Jul 2011

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.

SLAM visuel monoculaire par caméra infrarouge

A. Gonzalez^{1,2}

M. Devy^{1,2}

J. Solà Ortega^{1,2*}

¹ CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse Cedex 4, France

² Université de Toulouse ; UPS, INSA, INP, ISAE ; UT1, UTM, LAAS ; F-31077 Toulouse Cedex 4, France
{aurelien.gonzalez, michel.devy, jsola}@laas.fr

Résumé

De nombreuses applications nécessitent la mise en oeuvre d'algorithmes de vision sur véhicule, afin d'estimer des déplacements ou de localiser ce véhicule dans l'environnement. Cet article présente un algorithme de localisation et cartographie simultanée (SLAM) 3D que nous devons à terme intégrer dans un système embarqué sur un véhicule. Bien que de nombreuses nouvelles méthodes de SLAM se développent, nous avons choisi une méthode fondée sur un filtre de Kalman étendu (EKF) pour sa facilité de mise en oeuvre, de portage sur architecture embarqué ainsi que pour sa vitesse de traitement. Nous présentons les résultats de cet algorithme développé sur un PC standard, appliqué à des images acquises depuis une caméra visible capable de fournir des images à 60 Hz et depuis une caméra infrarouge basse résolution (160x120).

Mots Clef

SLAM, filtre de Kalman, infrarouge.

Abstract

Nowadays many applications require the execution of visual algorithms on a vehicle, for motion estimation or for self-localization. This paper presents an algorithm for a 3D Simultaneous Localization And Mapping (SLAM) that will be later integrated on a vehicle. Even if more efficient SLAM methods have been proposed recently, a method based on the Extended Kalman Filter (EKF), has been chosen for its easier implementation, for the easiness of migration on embedded system, and also for its speed processing. We will present the results of this algorithm, developed on a standard PC, applied to a 60 Hz visible camera and a low resolution infrared camera (160x120).

Keywords

SLAM, Extended Kalman Filter, infrared.

1 Introduction

Poussées par une forte demande des roboticiens, les méthodes de localisation et cartographie simultanées (SLAM) se sont grandement développées depuis les premiers travaux de Smith&Cheeseman [5], notamment autour de capteurs tels que des lasers [6] ou des sonars. Initialement, la

vision n'était pas considérée car jugée trop complexe, alors que c'est pourtant le capteur qui donne les informations les plus riches pour un robot. Les progrès faits en Vision d'une part (détection et suivi de points d'intérêt), puis les travaux de Davison en 2003 [1] sur une implémentation temps réel du SLAM visuel monoculaire, ont grandement influencé la communauté robotique ; le problème du SLAM par vision a atteint une certaine maturité en 2005 avec l'arrivée de techniques d'initialisations non retardées [2], et finalement résolu en 2006 avec la représentation d'amers ponctuels de type *inverse depth parametrisation* (IDP) [3]. Aujourd'hui, de nombreuses méthodes pour la description de points et de lignes sont recensées [4].

L'utilisation du Filtre de Kalman Étendu (EKF) dans les problèmes de localisation et cartographie simultanées en robotique a été introduit par Smith et al. [5], puis appliquée par Moutarlier et Chatila [6] sur des données laser. Les succès de cette approche grandissants, le filtre de Kalman étendu s'est imposé comme une référence dans le domaine, et les techniques de localisation et cartographie simultanées par vision monoculaire tournent, depuis 2007, avec des caméra visible à 30 Hz [7].

Cependant, les capteurs monoculaires infrarouges sont très peu utilisés dans le SLAM, du fait des contraintes matérielles et de coût de tels capteurs. Par exemple, les capteurs refroidis offrent une bonne qualité et résolution d'image mais ils sont très encombrants, surtout pour des systèmes embarqués. Les caméras non refroidies, quant à elles, ont l'encombrement d'une caméra classique, mais offrent très peu de contraste et une très faible résolution d'image. Toutefois, la technologie infrarouge commence à avoir une certaine maturité et fait son apparition dans les systèmes embarqués. En 2004, J. Nygård, P. Skoglar et, M. Ulvklö fusionnent les données d'un GPS-RTK (GPS centimétrique), une centrale inertielle et un système de navigation embarqué au sein d'un même filtre de Kalman pour géolocaliser et suivre les objets détectés avec les informations issues d'une caméra couleur et d'une caméra infrarouge lointain (bande III : 8-12 μm) embarqués dans leur drone [8]. En 2007, J. Folkesson et H. Christensen ont été les premiers à introduire cette technologie sur un robot de type Packbot pour une application de SLAM [9]. Même si le détecteur SIFT qu'ils ont employé est intéressant pour l'extraction de points d'intérêt dans l'infrarouge, il est également reconnu pour ne pas être compatible avec

*Joan Solà est maintenant à Ictineu Submarins, Barcelone, Catalogne.

les applications temps-réel. La bande infrarouge de leur caméra ($0,78 \mu\text{m} - 1,4 \mu\text{m}$) diffère également de la notre, et est limitée à 10 Hz.

Notre projet concerne l'estimation des déplacements d'un véhicule évoluant dans un environnement structuré, puis la localisation du dit véhicule par rapport à une carte a priori de cet environnement. Dans cet article, nous présentons la méthode développée pour construire une carte stochastique permettant d'estimer la position de la caméra et d'un ensemble de primitives visuelles, ici des points ; ces primitives seront exploitées ensuite pour la localisation. Notre travail a consisté à faire tourner, avec des caméras et des machines modernes, un SLAM à 60 Hz. Cet objectif a pu être réalisé grâce à une paramétrisation initiales des points détectés dans chaque image, par la représentation de type homogène ancrée. Ces points sont ensuite recherchés de manière active dans les images suivantes. Puis ils sont reparamétrisés en euclidien dès qu'ils ont convergé ; nous exploitons un filtre de Kalman étendu à chaque étape pour fusionner les nouvelles observations dans une carte stochastique contenant les positions de la caméra et des points. Nous montrerons la généralité de notre algorithme en l'appliquant à une gamme plus étendue de capteurs, tels qu'une caméra monoculaire visible fournissant des images à 60 Hz, et une caméra infrarouge lointain ($8 \mu\text{m} - 12 \mu\text{m}$) basse résolution.

La section suivante décrit les algorithmes et représentations mises en oeuvre. Dans la section 3 nous présentons et analysons des résultats expérimentaux, tandis que la section 4 résume les contributions du papier et évoque nos travaux en cours.

2 Méthode

2.1 Présentation du système

Dans un problème de SLAM classique, un ou plusieurs robots naviguent dans un environnement, découvrant et cartographiant en même temps des primitives visuelles, ici des points d'intérêt, à l'aide de leurs capteurs embarqués. Plusieurs robots de différents types peuvent être présents, portant chacun de multiples capteurs, qui eux-même fournissent des informations distinctes. En traitant les informations fournies par ces capteurs, les robots peuvent observer plusieurs points d'intérêt, qui ont des paramétrisations différentes dans la carte. Cette variété des objets à manipuler pour un tel problème, est gérée de manière transparente par le module "rtslam", développé au sein de l'environnement de développement interne au LAAS "jafar". Aujourd'hui, le code est en *open source*, disponible à l'adresse : <http://www.openrobots.org/wiki/rtslam>. Le module manipule des variables de cinq classes différentes : cartes, robots, capteurs, points d'intérêt et observation (Fig. 1). La hiérarchie de ces éléments entre eux adopte une forme d'arbre généalogique (Fig. 2). Ici nous considérons uniquement une carte construite par un seul robot, équipé d'un seul capteur, depuis lequel sont détectés des amers d'un seul type, des points. Pour d'autres travaux,

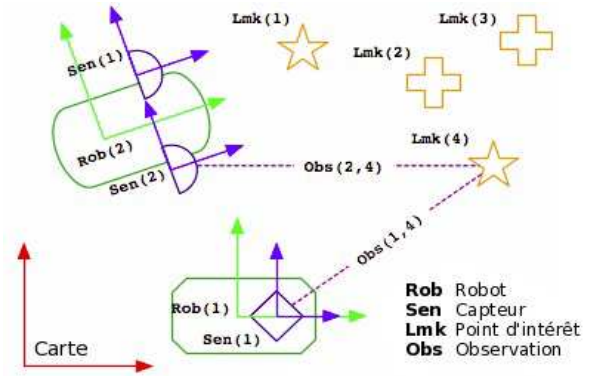


FIG. 1 – Exemples d'éléments inclus dans rtslam.

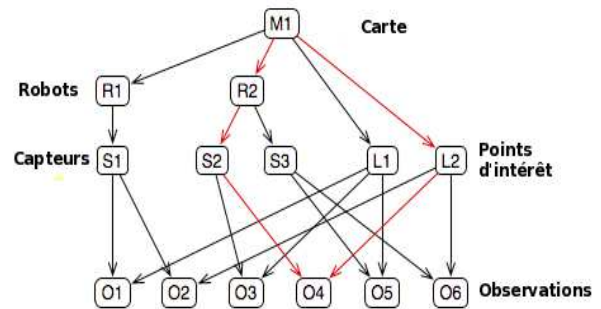


FIG. 2 – Hiérarchie des éléments sous forme d'arbre généalogique.

le même module sera exploité dans un contexte de sous-cartes, construites par plusieurs robots, équipés de plusieurs capteurs, détectant plusieurs types d'amers (points, droites, plans...).

2.2 Extraction des points

Le but est de pouvoir identifier, au fil des images et du mouvement de la caméra, les mêmes indices visuels. Pour cela, nous utilisons un détecteur de Harris [10], choisi à la fois pour sa robustesse et parce qu'il est capable d'opérer en temps-réel en mobilisant très peu de ressources. Nous adoptons ensuite l'approche de Davison et Murray [1], [11], en utilisant des descripteurs autour des points extraits. De tels descripteurs sont à la fois propres au point détecté et beaucoup plus précis que cet unique point. Ils servent également à caractériser le point sur le long terme.

2.3 Paramétrisation des points

Les points adoptent deux formes de paramétrisation : euclidien et homogène ancrée (ou *Anchored Homogeneous Point (AHP)*).

A. Les points euclidiens (EP)

Les points décrits de manière homogène ancrée (cf. section 2.3 B.), sur lesquels on a une grande précision et qui satisfont un test de linéarité après plusieurs observations, peuvent être reparamétrisés en points euclidiens (cf. section 2.4). Un point euclidien \mathbf{p} est représenté basiquement par

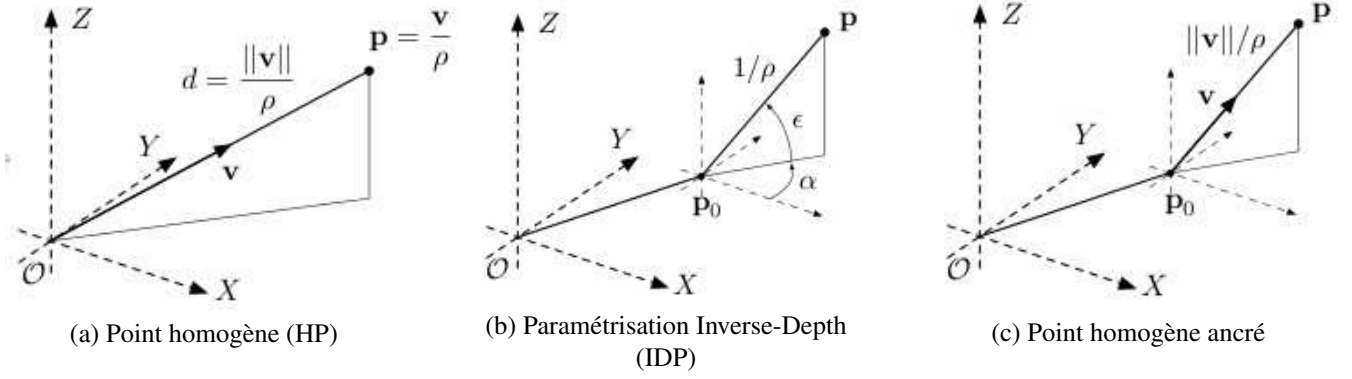


FIG. 3 – Les différentes paramétrisations des points (reproduit depuis [4]).

trois coordonnées cartésiennes

$$\mathcal{P}_{\mathbb{E}} = \mathbf{p} = [x \quad y \quad z]^T \in \mathbb{R}^3$$

La transformation de ce point dans le repère caméra, suivie de sa projection sur le plan image, est définie par

$$\underline{\mathbf{u}} = \mathbf{K}\mathbf{R}^T(\mathbf{u} - \mathbf{T}) \in \mathbb{P}^2 \quad (1)$$

ce qui nous permet d'introduire quelques notations. \mathbf{K} est la matrice des paramètres intrinsèques,

$$\mathbf{K} \triangleq \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{R} = \mathbf{R}(\mathbf{Q})$ et \mathbf{T} sont la matrice de rotation et le vecteur de translation de la caméra C , par rapport au repère du monde. La position de la caméra est donc représenté par le vecteur $C = (\mathbf{T}, \mathbf{Q})$. \mathbf{Q} représentant l'orientation, un modèle convenable pour l'EKF peut être choisi, tel que des quaternions.

L'opération de rétro-projection et transformation inverse est faite par

$$\mathbf{p} = \mathbf{R}\mathbf{K}^{-1}\underline{\mathbf{u}} + \mathbf{T} \quad (2)$$

B. Les points homogènes ancrés (AHP)

Lors de l'initialisation de nouveaux points, ceux-ci adoptent une paramétrisation homogène ancrée (AHP). En effet, les points euclidiens correspondent à des fonctions d'observation fortement non linéaires dans les systèmes *Bearing Only*, et ne sont pas conseillés pour l'initialisation de nouveaux points. La paramétrisation homogène ancrée consiste à ajouter une ancre à une paramétrisation de type homogène (HP, Fig. 3(a)) afin d'améliorer la linéarité, comme cela est fait pour la paramétrisation "Inverse-depth" (IDP, Fig. 3(b)) [12]. Ceci signifie que l'on ancre un point homogène \mathbf{p}_0 dans un espace 3D ailleurs qu'à l'origine. L'ancre du point \mathbf{p}_0 est choisi au centre optique lors de son initialisation, et le vecteur $\mathbf{v} = (u, v, w)$ est le vecteur directeur de l'axe optique. Lorsque ce vecteur est unitaire,

ajouter l'inverse de la distance ρ donne un point homogène $(u, v, w, \rho) \in \mathbb{P}^3$. Nous avons donc un point homogène ancré (AHP, Fig. 3(c)), paramétré par un vecteur de dimension 7

$$\mathcal{P}_{\text{AHP}} = \begin{bmatrix} \mathbf{p}_0^T & \mathbf{v}^T & \rho \end{bmatrix}^T = [x_0 \quad y_0 \quad z_0 \quad u \quad v \quad w \quad \rho]^T \in \mathbb{R}^7 \quad (3)$$

Ancrer un point a pour effet, lors de l'étape de la mise à jour du filtre de Kalman, qu'uniquement les erreurs accumulées sur l'ancre \mathbf{p}_0 par rapport à la position courante de la caméra T seront considérées. Contrairement à la paramétrisation homogène HP où les erreurs de la position absolue du capteur par rapport à l'origine sont cumulées.

Un point AHP se réfère au point Euclidien suivant :

$$\mathbf{p} = \mathbf{p}_0 + \mathbf{v}/\rho \quad (4)$$

La transformation et la projection de ce point sur le plan caméra sont définies par

$$\underline{\mathbf{u}} = \mathbf{K}\mathbf{R}^T(\mathbf{v} - \rho(\mathbf{T} - \mathbf{P}_0)) \in \mathbb{P}^2. \quad (5)$$

Un point AHP est initialisé par une opération de rétro-projection et transformation, à partir de son observation par un point d'intérêt détecté dans une image. Cela est réalisé par

$$\mathcal{P}_{\text{AHP}} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{v} \\ \rho \end{bmatrix} = \begin{bmatrix} \mathbf{T} \\ \mathbf{R}\mathbf{K}^{-1}\underline{\mathbf{u}} \\ \rho^C \end{bmatrix}, \quad (6)$$

où ρ^C doit être donné en tant qu'estimée de l'inverse de la distance du point au centre optique. Sa relation à la distance d est donné par la relation $\rho^C = \|\mathbf{K}^{-1}\underline{\mathbf{u}}\|/d$.

2.4 La reparamétrisation des points

Chaque point reparamétré en euclidien n'utilise plus que 3 éléments pour être représenté, au lieu des 7 de sa représentation sous forme homogène ancré. Le traitement est donc plus rapide, le SLAM ayant une complexité en O^3 , ainsi que la possibilité d'initialiser de nouveaux points à l'aide

de l'espace libéré. Un point homogène ancré sera reparamétrisé si sa prédiction est jugée suffisamment bonne, et son évolution linéaire au fil du temps. Nous effectuons donc le test de linéarité proposé par J.M.M. Montiel, J. Civera et A. J. Davison [13]. Pour cela, nous évaluons l'expression

$$L_d = \frac{4\sigma_d}{d_1} |\cos\alpha| \quad (7)$$

où $\sigma_d = \frac{\sqrt{\text{Var}(\rho)}}{\rho^2}$ peut être considéré comme un terme de normalisation de l'écart-type par rapport à la distance,

$$d_1 = \|(\mathbf{p}_0 + \mathbf{v}/\rho) - C_0\| \quad (8)$$

la distance entre l'estimée du point AHP dans la carte et l'estimée du centre optique de la caméra C_0 , et

$$\cos\alpha = \frac{\mathbf{v}^T \times ((\mathbf{p}_0 + \mathbf{v}/\rho) - C_0)}{d_1 \times \|\mathbf{v}\|} \quad (9)$$

représentant le cosinus de l'angle entre les droites formées d'une part par le centre optique et le point lors de son initialisation et d'autre part par le centre optique et le point à l'instant courant.

Si l'estimation a évolué de manière linéaire, alors $L_d \approx 0$ [13]. Il suffit alors de tester si la valeur absolue de L_d est inférieure à un seuil très faible (dans notre cas 0,1). Dans ce cas, on estime que ce point évolue de manière linéaire et son codage en AHP ne se justifie plus. Le point est reparamétrisé en euclidien par

$$\mathbf{p} = \mathbf{p}_0 + \mathbf{m}/\rho. \quad (10)$$

2.5 La carte probabiliste

L'un des points clé de notre approche est la gestion d'une carte probabiliste de l'ensemble des éléments de la scène, dans la version complète de *rtslam*, des sous-cartes, des robots, des capteurs et des amers visuels. Dans notre application, nous ne faisons que de l'estimation du mouvement; de ce fait, les amers ne sont pas conservés sur le très long terme. On peut donc ne construire qu'une seule carte; nous n'avons qu'un seul robot à localiser; ce robot n'est équipé que d'une caméra, et nous n'exploitons que des amers ponctuels, observés dans les images par des points d'intérêt.

La carte est représentée par un vecteur d'état contenant le robot et les amers ponctuels

$$X = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} \quad (11)$$

\mathcal{R} est le modèle de mouvement du robot de type *vitesse constante*, défini par

$$\mathcal{R} = \begin{bmatrix} p & q & v & w \end{bmatrix}^T$$

p et q étant la position et l'orientation du robot définie par un quaternion

$$p = \begin{bmatrix} x & y & z \end{bmatrix}^T \quad \text{et} \quad q = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^T,$$

et v et w sont chacun des vecteurs de dimension 3 définissant les vitesses linéaires et angulaires

$$v = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T \quad \text{et} \quad w = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^T.$$

Au final, pour décrire l'état d'un robot à partir d'un modèle de mouvement de type *vitesse constante*, il faut un vecteur de dimension 13.

\mathcal{M} contient l'ensemble des amers ponctuels. Chaque amer est codé sur 7 ou 3 coordonnées en fonction de sa paramétrisation en AHP ou en euclidien

$$\mathcal{M} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n \end{bmatrix} \quad (12)$$

Dans le cadre du filtre de Kalman étendu, la densité *a posteriori* est approximée par des densités Gaussiennes avec des matrices de moyennes et covariances définies par

$$\hat{X} = \begin{bmatrix} \hat{\mathcal{R}} \\ \hat{\mathcal{M}} \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{M}} \\ \mathbf{P}_{\mathcal{M}\mathcal{R}} & \mathbf{P}_{\mathcal{M}\mathcal{M}} \end{bmatrix} \quad (13)$$

où \mathbf{P} est carée et symétrique.

2.6 Filtre de Kalman étendu

Une fois le système lancé, une étape de prédiction suivie d'une étape de mise à jour s'alternent. L'évolution de la position du robot pendant un pas de temps Δt est décrite par la fonction

$$\mathcal{R}^+ = \mathbf{f}(\mathcal{R}, \mathbf{u}) \quad (14)$$

Dans notre cas

$$\mathcal{R}^+ = \begin{bmatrix} p^+ \\ q^+ \\ v^+ \\ w^+ \end{bmatrix} = \begin{bmatrix} p + v \times \Delta t \\ q + \frac{1}{2} \Omega(w) \cdot \mathbf{q} \cdot \Delta t \\ v + v_i \\ w + w_i \end{bmatrix} \quad (15)$$

où v_i et w_i représentent les perturbations sur les vitesses respectivement linéaires et angulaires et $\Omega(w)$ une matrice anti-symétrique construite à l'aide des éléments de w

$$\Omega(w) = \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} \quad (16)$$

D'après la formulation du filtre de Kalman étendu, l'étape de prédiction est définie par

$$\hat{\mathcal{R}}^+ = \mathbf{f}(\hat{\mathcal{R}}, \hat{\mathbf{u}}) \quad (17)$$

$$\mathbf{P}_{\mathcal{R}\mathcal{R}}^+ = \mathbf{F}_{\mathcal{R}} \mathbf{P}_{\mathcal{R}\mathcal{R}} \mathbf{F}_{\mathcal{R}}^T + \mathbf{F}_{\mathbf{u}} \mathbf{U} \mathbf{F}_{\mathbf{u}}^T \quad (18)$$

$$\mathbf{P}_{\mathcal{R}\mathcal{M}}^+ = \mathbf{F}_{\mathcal{R}} \mathbf{P}_{\mathcal{R}\mathcal{M}} \quad (19)$$

$$\mathbf{P}_{\mathcal{M}\mathcal{M}}^+ = \mathbf{P}_{\mathcal{M}\mathcal{M}} \quad (20)$$

où les matrices Jacobiennes sont définies par

$$\mathbf{P}_{\mathcal{R}} = \left. \frac{\partial \mathbf{f}}{\partial \mathcal{R}^T} \right|_{\hat{\mathcal{R}}, \hat{\mathbf{u}}} \quad \mathbf{F}_{\mathbf{u}} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}^T} \right|_{\hat{\mathcal{R}}, \hat{\mathbf{u}}} \quad (21)$$



(a) Robot Mana.



(b) Caméra visible Flea2.



(c) Caméra infrarouge Vitana.

FIG. 4 – Equipements expérimentaux grâce auxquels ont été réalisés les acquisitions.

2.7 Recherche active et correction de la carte

Dans ce paragraphe, nous allons nous intéresser à l'étape de correction de l'état par le filtre de Kalman. A ce stade, nous avons une prédiction gaussienne sur la localisation du ou des amers ponctuels. Pour chaque amer, cette prédiction donne une zone dans l'espace définie par une surface d'équiprobabilité qui a une forme d'ellipse 3D dans la carte comme illustré en Fig. 6. Pour un amer, nous avons un descripteur dans l'image, à savoir le descripteur associé au point d'intérêt de cet amer.

A l'aide des formules de projection (cf. section 2.3), nous connaissons ainsi leurs positions prédites dans le repère image. De plus, grâce à cette prédiction, nous avons une information sur la nouvelle représentation que doit avoir le descripteur du point en terme d'orientation et d'échelle. Il suffit alors d'appliquer ces transformations au descripteur d'origine, et de rechercher dans l'ellipse d'incertitude de la nouvelle image, le pixel qui a le descripteur le plus proche. Le pixel qui obtient le meilleur score de corrélation par ZNCC nous donne une nouvelle observation de l'amer, définie dans le repère image

$$ZNCC = \frac{1}{N} \sum_W \frac{(I - \bar{I})(J - \bar{J})}{\sigma_I \sigma_J} \quad (22)$$

où W est l'ensemble des pixels composant le descripteur, I le descripteur d'origine, J celui de l'image courante, et \bar{I} et \bar{J} leurs moyennes respectives.

On peut ensuite appliquer l'étape de rétro-projection pour mettre à jour l'état par filtrage de Kalman.

3 Résultats expérimentaux

3.1 Conditions expérimentales

Pour réaliser les acquisitions, les deux caméras ont été montées sur le robot "Mana" du LAAS (Fig.4(a)), robot tout-terrain dédié à l'exploration. Les acquisitions ont été faites de jour et de nuit. Les images sont enregistrées et datées très précisément (à la nanoseconde) pour pouvoir ensuite être rejouées avec une grande précision. Le Robot

dispose également d'un GPS-RTK faisant office de vérité de terrain.

Le programme est lancé sur un processeur Inter Core 2 Duo T9600 à 2.8 GHz. Un cœur du processeur est dédié à la partie traitement d'image et SLAM, et l'autre cœur uniquement pour l'affichage 2D et 3D. Le cœur affecté à l'algorithmie dispose donc moins de 17 ms pour effectuer l'ensemble des calculs afin d'assurer un SLAM tournant à 60 Hz. Lors du jeu des séquences, le calcul ne prenait qu'environ 40% du processeur pour une carte stochastique de 1000 éléments.

3.2 Caméra monoculaire visible

La caméra visible utilisée est une caméra IEEE1394 Flea2, développé par Point Grey Research (Fig. 4(b)). Elle renvoie des images de 640×480 en niveau de gris. L'intérêt de cette caméra est qu'elle est capable de fournir des images à 60 Hz.

Tout au long de la séquence (Fig. 5), nous pouvons remarquer que l'incertitude sur les amers (représenté par la largeur des ellipses) reste très faible tout au long du parcours. L'intérêt d'un SLAM à 60 Hz est que la prédiction du mouvement est beaucoup plus facile à estimer, et également plus précise car le déplacement entre deux images est moindre que par rapport à un SLAM monoculaire avec une caméra de faible fréquence. On peut également remarquer que la majorité des amers a rapidement convergée et adoptée une paramétrisation de type euclidien. La figure 6(a) représente la trajectoire estimée issue de notre algorithme de SLAM, et la figure 7(b) la vérité de terrain issue de la fusion entre un GPS centimétrique, odomètres et centrale inertielle. Bien qu'étant à un facteur d'échelle près, car la distance est non observable par des capteurs de type *Bearing Only*, et à un facteur d'orientation près, la forme de la trajectoire est sensiblement la même. La figure 6(c) représente la dérive prise en altitude le long de la séquence. La prise étant faite sur un robot évoluant sur un sol plan, cette dérive devrait être quasi-nulle. Celle-ci reste malgré tout bornée entre -55 cm et 70 cm.

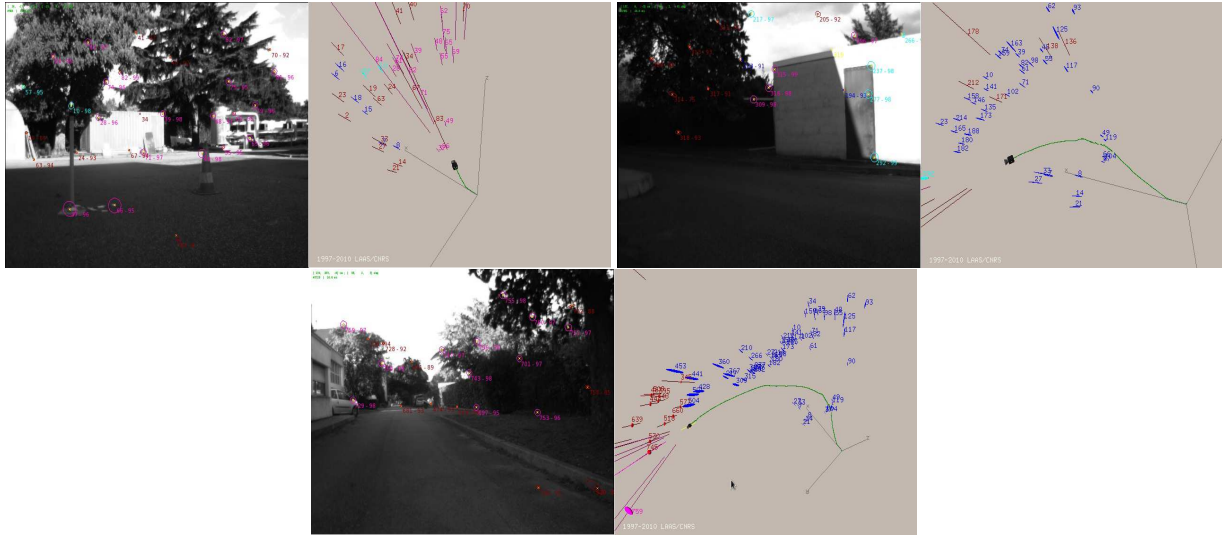
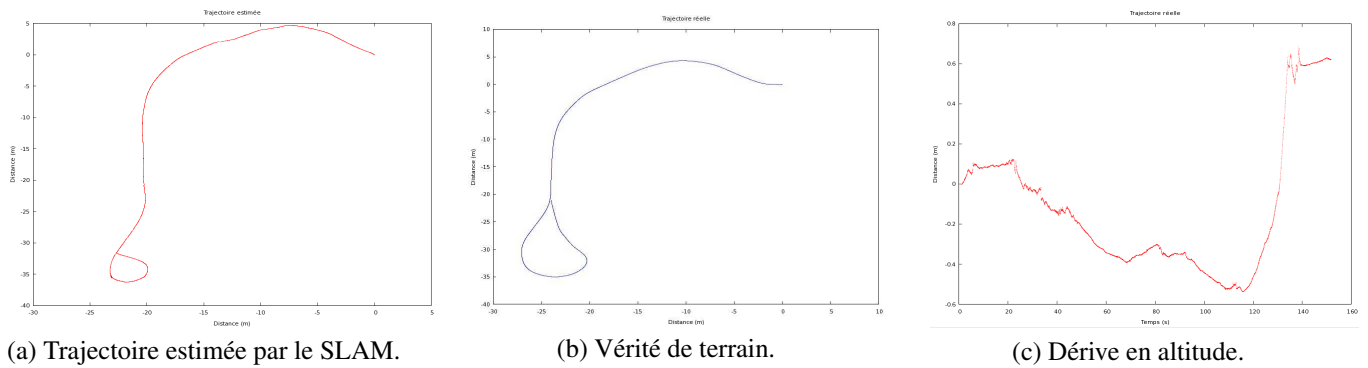


FIG. 5 – Résultats du *rtslam* en visible à différents points du parcours du robot. Les ellipses rouges représentent les amers ayant une paraétrisation AHP et les bleus une paramétrisation euclidien. Sur les images de gauche sont superposées les points des amers prédits et leurs ellipses d'incertitude à l'image visible. Les images de droite sont la représentation en 3D de la position de la caméra sur le robot, sa trajectoire en vert, et l'ensemble des amers avec leurs incertitudes respectives.



(a) Trajectoire estimée par le SLAM.

(b) Vérité de terrain.

(c) Dérive en altitude.

FIG. 6 – La figure (a) montre la trajectoire estimée par le SLAM via la caméra visible par rapport à la vérité de terrain (b). La figure (c) montre la dérive prise en altitude.

3.3 Caméra monoculaire infrarouge

La caméra infrarouge lointain ($8 \mu\text{m} - 12 \mu\text{m}$) est une caméra IEEE1394 Vitana, développée par Indigo (Fig. 4(c)). Elle renvoie des images de 160×120 en niveaux de gris mais uniquement à 30 Hz. Il est cependant nécessaire d'avoir des scènes contrastées en température, mais elle peut être d'une grande utilité dans des cas comme de nuit ou par météorologie dégradée.

Dans le cas de l'infrarouge, on remarque que les incertitudes sur les mesures sont beaucoup plus importantes que dans le cas du visible, moins de points sont reparamétrés en euclidien et la trajectoire est plus bruitée (Fig. 7). De ce fait, la trajectoire est plus bruitée (Fig. 8(a)), et est également moins proche de la vérité de terrain (Fig. 8(b)), même si elle en adopte la forme générale, toujours à un facteur d'échelle et à une orientation près. La dérive en altitude reste relativement stable, bornée entre 2 cm et -16 cm.

L'ensemble de ces défauts sont reliés et viennent en partie de la faible résolution de la caméra, du faible contraste présent dans les images infrarouges, ainsi que des bruits au niveau du capteur. Pour un SLAM monoculaire, il est intéressant d'avoir suffisamment de points au niveau du sol pour avoir une bonne estimation de la dynamique de notre mobile. Dans le cas des images infrarouges, on remarque que le sol est uniforme en température, et donc ne possède pas suffisamment de contraste permettant une bonne extraction des points d'intérêt. Cependant, l'ensemble des autres points d'intérêt permettent une bonne estimation du cap. Il est alors intéressant d'associer par exemple ce type de caméra avec une centrale inertielle pour obtenir la dynamique de notre mobile.

De plus, toutes les minutes, la caméra applique une correction de "Non Uniform Table (NUC)" positionnant pendant une seconde devant l'objectif une mire de température uni-

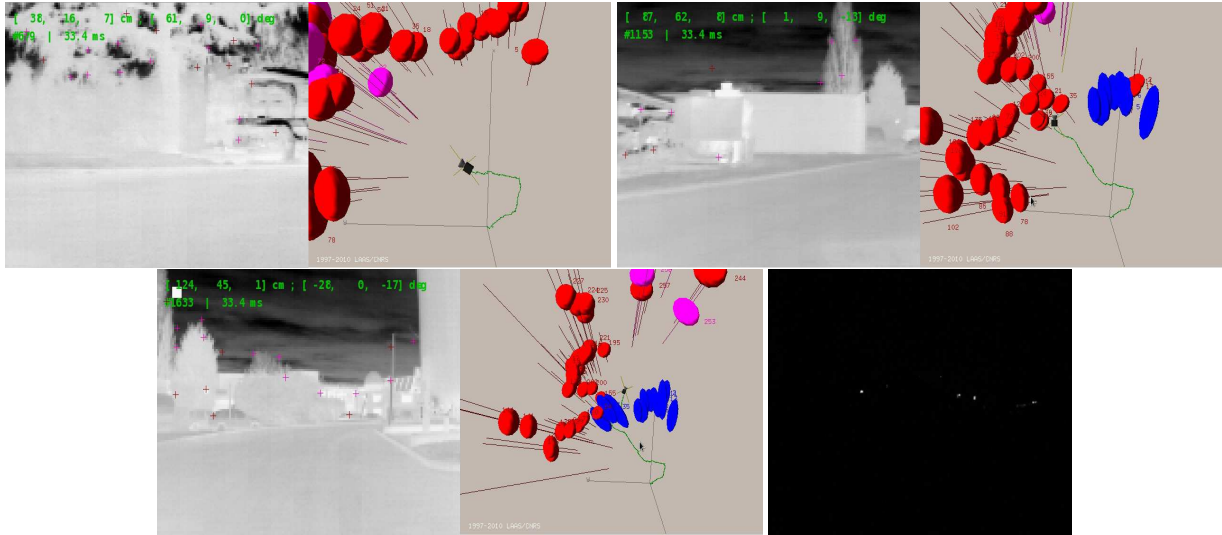


Fig. 7 – Résultats du *rtSLAM* en infrarouge à différents points du parcours du robot, et en bas à droite, ce que voit la caméra visible au même instant.

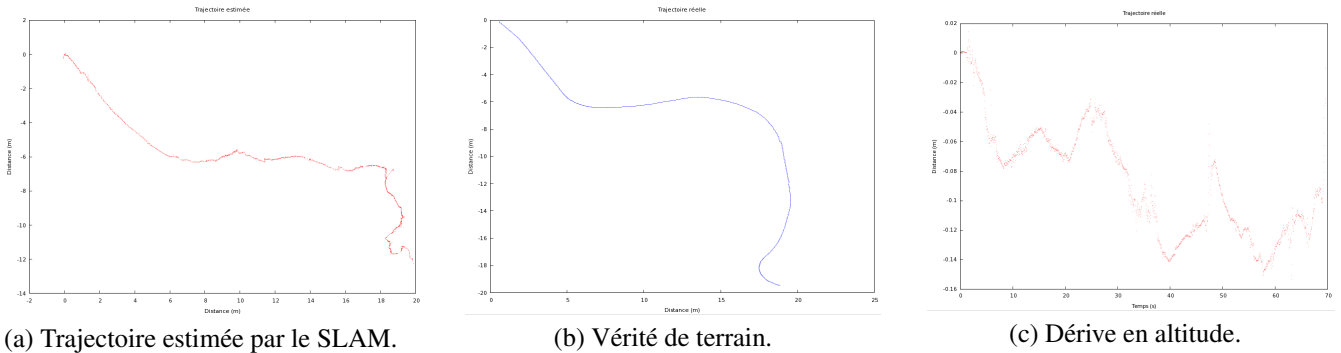


Fig. 8 – La figure (a) montre la trajectoire estimée par le SLAM via la caméra infrarouge par rapport à la vérité de terrain (b). La figure (c) montre la dérive prise en altitude.

forme permettant de recalibrer la caméra. Malgré tout, la technologie de vision infrarouge reste un choix intéressant dans des conditions nocturnes ou de météorologie dégradée.

4 Conclusions et perspectives

Dans cet article, nous avons présenté notre algorithme "*rtSLAM*". Nous avons démontré sa capacité à traiter un problème de localisation et de cartographie simultanée avec une caméra visible à 60 Hz, et avec une caméra infrarouge lointain. Cette rapidité de traitement vient du choix et de la vitesse d'exécution de chacun des composants : l'extracteur de coins de Harris, la reparamétrisation des amers, le filtre de Kalman étendu et la méthode de recherche active.

Nos recherches futures viseront à intégrer plusieurs types de capteurs hétérogène au sein d'une même application de localisation et cartographie simultanées, tels qu'une combinaison de caméras visibles et infrarouges, une centrale inertielle et un GPS. En ayant une connaissance *a priori*

de la scène, celle-ci pourrait être également incluse dans le filtre pour améliorer les résultats de l'algorithme. Dans [14], nous montrons comment nous avons intégré une centrale inertielle dans l'équation d'état du robot, et les résultats obtenus dans l'hypothèse d'un mouvement sur un plan. Nous résolvons ainsi le problème du facteur d'échelle, et limitons les divergences du SLAM monoculaire.

Concernant l'infrarouge, nous avons remarqué que le détecteur de Harris, même s'il fournit de bons résultats, n'est pas optimal pour les images thermiques qui ont des contours plutôt diffus. Dans ce cadre là, on peut noter l'utilisation de SIFT [9], ou de détecteurs de coins de SUSAN [8]. Dans sa thèse, K. Hajebi[15] a comparés différents algorithmes d'extraction de points d'intérêt dans des images infrarouges (Harris, KLT, Sobel, ...). L'un des mieux adaptés à ces images est la Congruence de Phases, algorithme que nous sommes en train de mettre en place.

Remerciements

Cet algorithme a été développé par une équipe du LAAS composée de six personnes : Cyril Roussillon, Nicolas Mansard, Jean-Marie Codol, David Marquez, Aurélien Gonzalez et Joan Solà qui était également responsable de ce projet et qui s'y est grandement investi. Nous les remercions tous très chaleureusement. Nous remercions à nouveau Cyril Roussillon pour son aide précieuse sur la prise des différentes séquences à bord du robot Mana.

Références

- [1] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," dans *Proc. Int. Conf. on Computer Vision*, Vol. 2, pp. 1403-1410, Nice, Octobre 2003.
- [2] J. Solà, A. Monin, M. Devy, et T. Lemaire, "Undelayed initialization in bearing only SLAM," dans *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2499-2504, Edmonton, Canada, 2005.
- [3] J. Montiel, J. Civera, et A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," dans *Proc. Robotics : Science and Systems*, Philadelphia, USA, Août 2006.
- [4] J. Solà, "Consistency of the monocular EKF-SLAM algorithm for 3 different landmark parametrizations," dans *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3513-3518, Anchorage, Alaska, Mai 2010.
- [5] R. Smith, M. Self, et P. Cheeseman, "A Stochastic Map for Uncertain Spatial Relationships," dans *Proc. Fourth Int. Symp. Robotics Research*, 1987.
- [6] P. Moutarlier et R. Chatila, "Stochastic Multisensory Data Fusion for Mobile Robot Location and Environment Modelling," dans *Proc. Int. Symp. Robotics Research*, 1989.
- [7] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM : Real-Time Single Camera SLAM," dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, no. 6, pp. 1052-1067, Juin 2007.
- [8] J. Nygård, P. Skoglar et, M. Ulvklo, "Navigation Aided Image Processing in UAV Surveillance : Preliminary Results and Design of an Airborne Experimental System" dans *Journal of Robotic Systems*, Vol.21, no. 2, pp. 63-72, 2004.
- [9] J. Folkesson et H. Christensen, "SIFT Based Graphical SLAM on a Packbot," dans *Proc. 6th Int. Conf. Field and Service Robotics (FSR)*, Chamonix, France, 2007.
- [10] C. Harris et M. Stephens, "A combined corner and edge detector," dans *Proc. 4th Alvey Vision Conference*, pp.147-151, 1988.
- [11] A. J. Davison and D.W. Murray, "Mobile Robot Localisation Using Active Vision," dans *Proc. Fifth European Conf. Computer Vision*, pp. 809-825, 1998.
- [12] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," dans *IEEE Trans. on Robotics*, vol. 24, no. 5, 2008.
- [13] J.M.M. Montiel, J. Civera, et A.J. Davison, "Unified Inverse Depth Parametrization for Monocular SLAM," dans *Proc. Robotics : Science and Systems*, 2006.
- [14] A. Gonzalez, dans "SLAM visuel monoculaire par caméra infrarouge dans un contexte aéroport," dans *12e Congrès des Doctorants EDSYS*, 2011.
- [15] Kiana Hajebi, dans "Structure from Infrared Stereo Images," *Thèse, University de Waterloo*, Waterloo, Ontario, Canada, 2007.