

Évaluation de deux algorithmes de partage de données en Cartographie et Localisation Simultanées visuelles-inertielles multi-robot et décentralisées

Rodolphe Dubois^(1,2)

Alexandre Eudes⁽¹⁾

Vincent Frémont⁽²⁾

⁽¹⁾ DTIS, ONERA, Université Paris Saclay, F-91123 Palaiseau, France

⁽²⁾ Centrale Nantes, LS2N, UMR 6004, Nantes, France

E-mail : ⁽¹⁾ prénom.nom@onera.fr, ⁽²⁾ prénom.nom@ls2n.fr

Résumé

Cet article présente et compare deux méthodes décentralisées de partage de cartes et de trajectoires en SLAM visuel-inertiel multi-robot. La première méthode repose sur le calcul de facteurs visuel-inertiels marginalisés et sparsifiés, associés à des informations visuelles locales, tandis que la seconde s'appuie sur l'échange de sous-cartes associées à des informations purement structurelles. Ces deux méthodes résultent de la transposition de deux algorithmes développés par Paull et al. [1] et Schuster et al. [2] respectivement pour du SLAM sous-marin acoustique et du SLAM terrestre stéréo. Leurs performances en termes de précision et de quantité de données échangées sont évaluées sur des scénarios multi-robot élaborés sur le jeu de données EUROCC [3].

Mots Clefs

Robotique, Cartographie et Localisation Simultanées (SLAM) visuelles-inertielles décentralisées

Abstract

This article introduces and evaluates two decentralized data sharing algorithms for multi-robot monocular visual-inertial simultaneous localization and mapping. One is based on the exchange of marginalized and consistently sparsified packets associated to local visual information, and the other one, on the exchange of submaps associated with purely structural information. Those two methods have been adapted from two distinct algorithms developed by Paull et al. [1] and Schuster et al. [2] respectively for underwater acoustic SLAM and terrestrial stereo-SLAM. Their performances in terms of accuracy and communication loads are evaluated on multi-robot scenarios built on the EUROCC dataset [3].

Keywords

Robotics, Decentralized Visual-Inertial Simultaneous Localization And Mapping (SLAM).

1 Introduction

1.1 Problème considéré

L'utilisation d'une flotte de robots est de plus en plus envisagée pour des applications telles que la surveillance ou l'inspection de zones à grande échelle. Pour de telles tâches, les méthodes de cartographie et localisation simultanées (*Simultaneous Localization And Mapping* (SLAM) en anglais) constituent une brique algorithmique essentielle en support des algorithmes de contrôle et de décision haut-niveau. Le SLAM consiste, pour un robot, en l'exploitation des mesures acquises par ses capteurs embarqués pour cartographier en temps réel l'environnement dans lequel il évolue tout en s'y localisant.

Bien que le SLAM mono-robot ait fait l'objet d'études approfondies [4], son extension au cas multi-robot introduit de nouvelles problématiques relatives au partage de données entre les agents. Ceux-ci s'échangent séquentiellement des informations sur des portions de leur cartes sous la forme d'une succession de paquets de données, qu'ils construisent de façon à fidèlement restituer le plus d'informations possible sur leurs estimées et leur modèle d'estimation sous-jacent. L'objectif vis-à-vis du robot destinataire est double : i) enrichir sa propre carte en recalant correctement les informations reçues ; ii) affiner ses estimées sans compromettre leur consistance (*i.e.* la caractérisation correcte de leur covariance) en corrélant les données reçues avec son propre modèle probabiliste.

En outre, les interactions entre les robots sont contraintes en communication : portée limitée, bande passante bornée, latences et pertes de paquets. Pour s'y adapter, les architectures multi-robot doivent concilier l'autonomie des agents avec des solutions flexibles et robustes quant à la quantité et l'intégration des données échangées.

1.2 État de l'art associé

1) SLAM multi-robot centralisé. La plupart des méthodes de partage de données introduites jusqu'à présent dans la littérature sont centralisées puisqu'elles n'admettent que des échanges directs entre des agents et un serveur central. Forster *et al.* [5] ont développé une mé-

thode centralisée en SLAM visuel-inertiel monoculaire qui attribue des tâches de bas niveau telles que la sélection et l'analyse d'images-clés aux agents, tandis que les tâches haut niveau comme la cartographie, la fermeture de boucles et la mise en correspondance des cartes sont exécutées par une station au sol. Karrer *et al.* [6] ont récemment introduit un système similaire appelé CVI-SLAM. En SLAM visuel, Li *et al.* [7] ont proposé une version centralisée de l'algorithme ORB-SLAM [8]. Cependant, ces méthodes nécessitent souvent d'échanger une grande quantité de données (*e.g.* amers 3D, points d'intérêt 2D et descripteurs associés), quitte à excéder la bande-passante disponible. De plus, contraints de rester à portée de communication du serveur, les agents voient leur autonomie restreinte. Enfin, on notera que les architectures centralisées sont particulièrement vulnérables à toute défaillance du serveur central.

2) SLAM multi-robot décentralisé. Face à ces inconvénients, de nombreux auteurs se sont intéressés aux architectures décentralisées. Schuster *et al.* [2] ont proposé une approche basée sur l'échange de sous-cartes locales dont les nuages de points sont ensuite recalés les uns par rapport aux autres par ICP (*Iterative Closest Point*). Contreras *et al.* [9] ont également proposé une méthode décentralisée pour du SLAM basé LIDAR. Pour faire face aux limitations de bande passante, d'autres méthodes font s'échanger des représentations condensées des cartes et des trajectoires. Ces méthodes reposent souvent sur des techniques récentes de marginalisation et de *sparsification* [10, 11, 12, 13]. Par exemple, Cunningham *et al.* [14] ont proposé l'algorithme DDF-SAM dans lequel chaque robot communique des facteurs marginalisés liant des amers 3D communément observés. De la même manière, Lazaro *et al.* [15] font s'échanger des mesures condensées résultant de marginalisations sur des variables partagées par les deux robots. En SLAM sous-marin acoustique, Paull *et al.* [1] ont proposé une méthode robuste et décentralisée basée sur le maintien d'un historique des échanges inter-robots. Néanmoins, peu de méthodes de ce type ont été proposées pour le SLAM visuel-inertiel monoculaire.

1.3 Contributions

Dans cet article, nous proposons et comparons deux méthodes décentralisées de partage de cartes et de trajectoires, inspirées des méthodes développées par [1] et [2] et transposées en contexte visuel-inertiel monoculaire. La section II formule le problème du SLAM visuel-inertiel monoculaire dans le formalisme graphique. Les sections III et IV décrivent les deux méthodes proposées, dont les performances sont évaluées dans la section VI sur des scénarios multi-robot construits sur le jeu de données EUROC [3].

2 SLAM visuel-inertiel monoculaire

En SLAM visuel-inertiel monoculaire, chaque robot dispose d'une caméra monoculaire et d'une centrale inertielle (IMU pour *Inertial Measurement Unit*). Celles-ci acquièrent respectivement une séquence d'images ainsi que

des mesures d'accélération spécifique et de vitesse angulaire. Le robot interprète ces mesures pour construire en ligne un modèle probabiliste graphique ou *graphe de facteurs* (*c.f.* figure 1) sur lequel il infère pour estimer sa trajectoire et construire un modèle de son environnement.

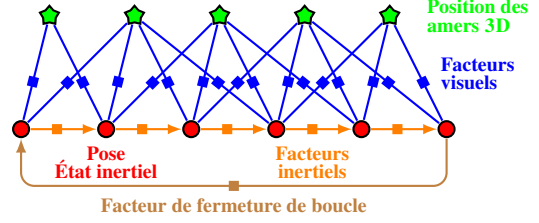


FIGURE 1 – Modèle probabiliste graphique pour le SLAM

La trajectoire est représentée par un ensemble \mathcal{X} discret de vertex. Chaque vertex est caractérisé par une pose $T_{WB_k} \in SE(3)$ – où l'indice k renvoie à l'instant t_k , W désigne le repère monde aligné avec la gravité et dont l'origine coïncide avec la position du premier vertex, et B le repère attaché au robot – et d'états inertiels ${}^{WB_k}\xi_k = [{}^W v_k^T \ {}^{B_k} b_{ak}^T \ {}^{B_k} b_{\omega k}^T]^T$ dont les composantes respectives sont la vitesse, le biais de l'accéléromètre et le biais du gyromètre. L'environnement est modélisé par un nuage \mathcal{L} d'amers 3D triangulés à partir de leurs observations suivies dans les images successives. Comme illustré dans la figure 1, toutes ces variables sont mutuellement contraintes par des facteurs gaussiens dérivés des mesures visuelles et inertielle. À chaque mesure IMU ${}^{B_k}u_k$ correspond le facteur suivant :

$$p({}^{WB_k}x_{k+1} | {}^{WB_k}x_k, {}^{B_k}u_k) \propto \exp\left(-\frac{1}{2} \left\| {}^{WB_k}x_{k+1} \ominus f_k({}^{WB_k}x_k, {}^{B_k}u_k) \right\|_{\Sigma_u}^2\right) \quad (1)$$

où ${}^{WB_k}x_k = (T_{WB_k}, {}^{WB_k}\xi_k) \in \mathcal{X}$, f_k est la dynamique discrétisée du robot et Σ_u la matrice de covariance du résidu inertiel, propagée selon le modèle IMU. L'opérateur \ominus renvoie le résidu dans un espace tangent au groupe de Lie composite dans lequel x_k évolue [16]. Pareillement, chaque observation z_k^i d'un amer 3D l_i produit le facteur :

$$p(z_k^i | T_{WB_k}, {}^W l_i) \propto \exp\left(-\frac{1}{2} \left\| z_k^i - \pi_c(T_{WB_k}, {}^W l_i) \right\|_{\Sigma_v}^2\right) \quad (2)$$

où ${}^W l_i \in \mathcal{L}$, Σ_v est la matrice de covariance du résidu visuel, et π_c la fonction de projection du référentiel monde vers l'image de la caméra. Enfin, des contraintes de fermetures de boucles sont ajoutées lorsqu'un robot associe la scène qu'il observe à une portion de sa carte déjà cartographiée. Il peut alors en déduire des correspondances 2D-3D, fusionner des amers 3D et corriger la dérive odométrique qu'il a accumulée entre temps.

La carte et la trajectoire sont finalement estimées par inférence sur le graphe de facteurs, à l'aide d'un estimateur de maximum de vraisemblance (*Maximum Likelihood (ML)*) :

$$\{\mathcal{X}_{ML}^*, \mathcal{L}_{ML}^*\} = \arg \min_{\mathcal{X}, \mathcal{L}} -\log p(\mathcal{Z} | \mathcal{X}, \mathcal{L}) \quad (3)$$

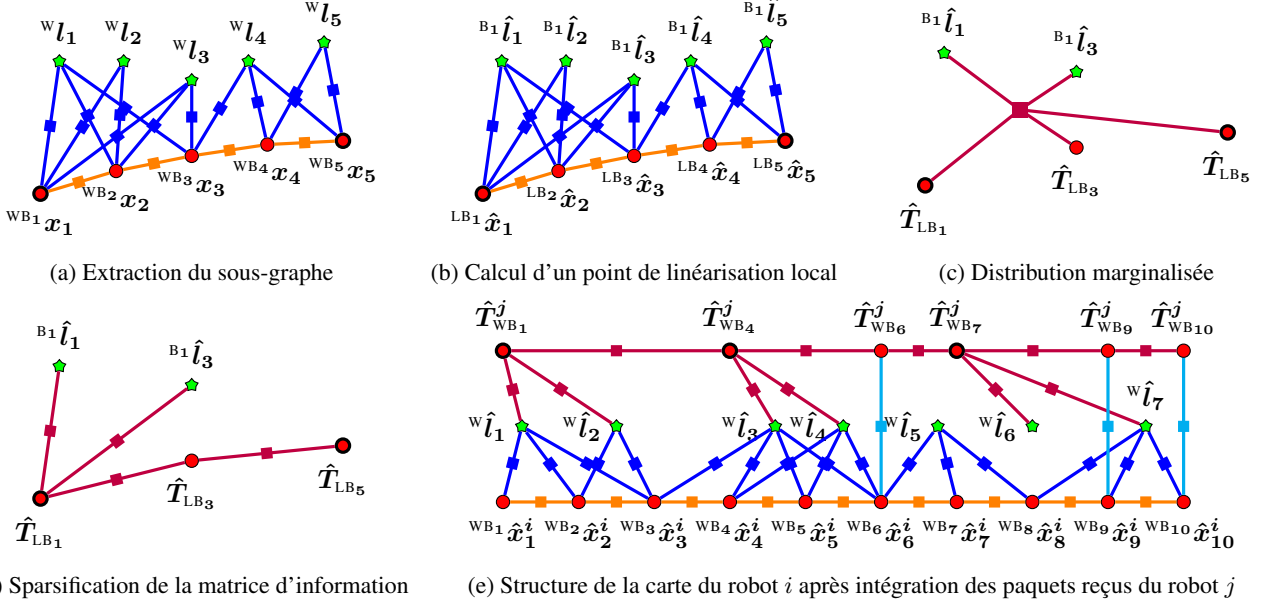


FIGURE 3 – Représentation du processus de marginalisation et de sparsification consistante pour construire un paquet.

les poses $\mathbf{T}_{LB_k} \in \mathcal{T}_S$ des vertex marqués ainsi que les positions ${}^w\mathbf{l}_i \in \mathcal{L}_S$ des amers 3D sélectionnés :

$$\mathcal{I}_D = \mathcal{I}_{\mathcal{Z}}(\mathcal{T}_S, \mathcal{L}_S) = \mathcal{I}_{SS} - \mathcal{I}_{SM}\mathcal{I}_{MM}^{-1}\mathcal{I}_{MS} \quad (5)$$

où l'indice S renvoie aux variables sélectionnées et l'indice M aux variables marginalisées. Comme illustré par la figure (3c), on obtient une distribution complexe sur les variables retenues, car elle inclut leurs corrélations croisées.

5) Sparsification consistante pour le sous-graphe. La distribution $\mathcal{N}(\hat{\mathbf{m}}_s, \mathcal{I}_D^{-1})$, où $\hat{\mathbf{m}}_s$ est le vecteur des variables sélectionnées, est trop complexe pour être communiquée en l'état. On souhaite donc calculer un jeu de facteurs binaires gaussiens, décorrélés, de poses et de positions relatives, que nous appellerons *mesures virtuelles*. Ils devront contraindre mutuellement les variables sélectionnées selon une topologie simplifiée telle que représentée dans la figure (3d), engendrer la même estimée locale $\hat{\mathbf{m}}_s$ et approcher au mieux la matrice d'information des paramètres \mathcal{I}_D . La matrice d'information des mesures virtuelles Ω_S est calculée comme la solution d'un problème d'optimisation qui minimise la perte d'information sous une contrainte de consistance :

$$\hat{\Omega}_S = \arg \min_{\Omega_S \in \mathcal{D}_+} \mathcal{D}_{KL}(\mathcal{N}(\hat{\mathbf{m}}_s, (\mathbf{J}_S^T \Omega_S \mathbf{J}_S)^{-1}) || \mathcal{N}(\hat{\mathbf{m}}_s, \mathcal{I}_D^{-1}))$$

$$\text{sous la contrainte } \mathcal{I}_S = \mathbf{J}_S^T \Omega_S \mathbf{J}_S \leq \mathcal{I}_D \quad (6)$$

où \mathcal{D}_{KL} est la divergence de Kullback-Leibler, \mathcal{D}_+ est l'ensemble des matrices semi-définies positives correspondant à la topologie recherchée et \mathcal{I}_S est la matrice caractérisant l'information apportée par les mesures virtuelles sur les variables estimées. \mathbf{J}_S est la matrice jacobienne des mesures virtuelles évaluée en $\hat{\mathbf{m}}_s$:

$$\mathbf{J}_S = \frac{\partial (\{\mathbf{T}_{B_i B_j}\}_{i \in \mathcal{T}_S}, \{\mathbf{l}_i\}_{i \in \mathcal{L}_S})}{\partial (\{\mathbf{T}_{LB_i}\}_{i \in \mathcal{T}_S}, \{\mathbf{l}_i\}_{i \in \mathcal{L}_S})} \quad (7)$$

C'est une matrice clairsemée, et les matrices jacobiennes locales pour les poses relatives valent :

$$\mathbf{J}_{\{\mathbf{T}_{LB_i}, \mathbf{T}_{LB_j}\}}^{\mathbf{T}_{B_i B_j}} = \begin{bmatrix} -\text{Ad}_{\mathbf{T}_{B_i B_j}}^{-1} & \mathbf{I}_{6 \times 6} \end{bmatrix} \quad (8)$$

où $\text{Ad}_{(\cdot)}$ est la matrice adjointe sur $\text{SE}(3)$. La contrainte de consistance permet d'assurer que la distribution issue de la sparsification n'ajoute pas d'informations artefactuelles *i.e.* qu'elle ne réduise pas artificiellement l'entropie jointe sur certains sous-groupes de variables. Comme démontré dans [11], comme \mathbf{J}_S est inversible par construction, la solution au problème relaxé de la contrainte de consistance est :

$$[\Omega_S^*]_i = (\{\mathbf{J}_S \mathcal{I}_D^{-1} \mathbf{J}_S^T\}_i)^{-1} \quad (9)$$

où l'indice i renvoie au $i^{\text{ème}}$ bloc diagonal associé au $i^{\text{ème}}$ facteur de la topologie désirée.

Il nous faut maintenant imposer la contrainte de consistance à partir de cette solution au problème relaxé. Les méthodes classiques pour résoudre le problème (6), basées sur la résolution de problèmes de programmation quadratique semi-définie par des méthodes de points-intérieurs qui modélisent la contrainte de consistance par des barrières de type log-det, sont inapplicables en temps réel. On applique ici une méthode non optimale pour imposer cette contrainte mais avec un coût plus réduit. Soit λ_{\min}^D la plus petite valeur propre de \mathcal{I}_D . Chaque valeur propre λ_i des blocs calculés est comparée à λ_{\min}^D et remplacée par elle si elle l'excède. On obtient ainsi une matrice d'information très conservatrice mais consistante et par une méthode à la complexité raisonnable.

3.3 Envoi et intégration du paquet

Chaque robot maintient un historique de ses communications avec les autres robots. Il vérifie périodiquement quels sont les robots avec lesquels il peut communiquer, et envoie

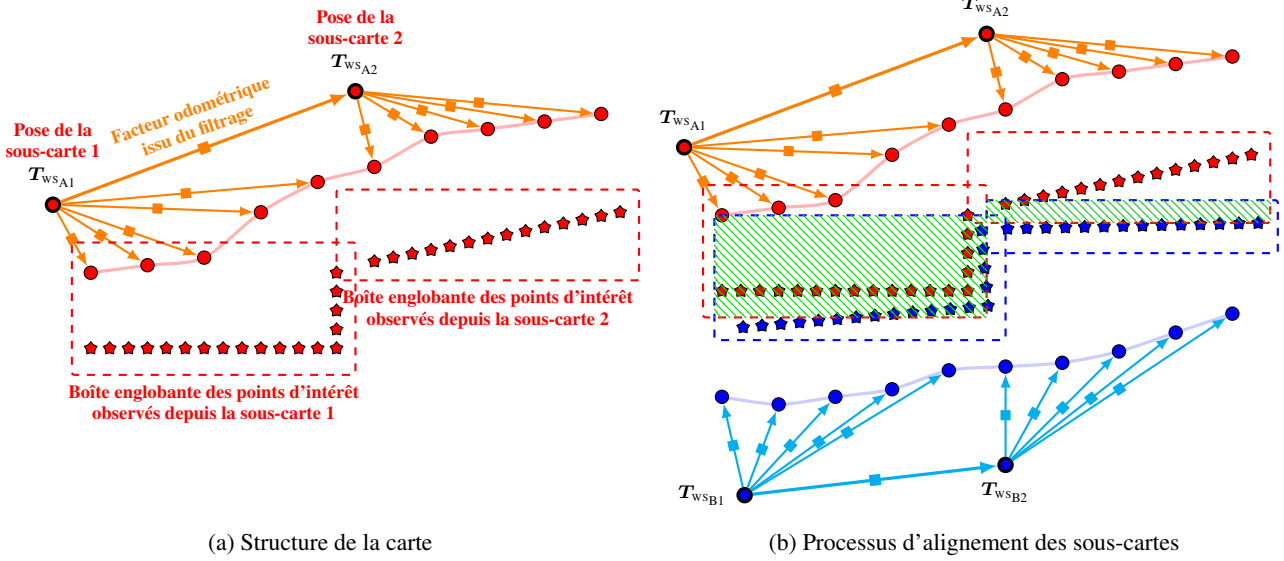


FIGURE 4 – Algorithme de partage de sous-cartes introduit dans [2]

à chacun d’entre eux, séquentiellement, les paquets qu’il ne leur a pas encore transmis. Comme illustré par la figure (3e), les paquets reçus sont intégrés dans la carte du robot récepteur. L’information visuelle communiquée sur les vertex sélectionnés est utilisée pour détecter et caractériser des correspondances 2D-3D avec la carte du récepteur. Enfin, l’émetteur communique également les observations inter-robot (*e.g.* estimation de poses relatives par détection de marqueurs AprilTags [18]) associées aux vertex sélectionnés. La détection de ces fermetures de boucle inter-robot ainsi que la bonne corrélation des paquets reçus avec la carte du récepteur doit aider ce dernier à bien les recalcr et éventuellement corriger sa propre dérive odométrique.

4 Méthode 2 – Échange de sous-cartes associées à des informations structurelles

La méthode décrite ci-dessous est inspirée de celle développée par Schuster *et al.* [2] pour du SLAM basé sur de la vision stéréo, et ici transposée en contexte visuel-inertiel monoculaire.

4.1 Méthode originale

Comme représenté sur la figure (4a), la carte de chaque robot est constituée d’une chaîne linéaire de sous-cartes. Chaque sous-carte est caractérisée par une pose qui lui attache un référentiel S aligné avec le vecteur gravité. Dans [2], un filtre de Kalman étendu (EKF) à référence locale, est utilisé pour construire séquentiellement ces sous-cartes. Pour chaque vertex de la sous-carte, ce filtre fournit une estimation de sa pose dans le référentiel attaché à la sous-carte ainsi qu’une matrice de covariance caractérisant son incertitude. Lorsque la distance parcourue au sein de la sous-carte ou que la variation cumulée d’angle de lacet excèdent des seuils respectifs d_{\max} et θ_{\max} définis par l’utilisateur,

une nouvelle sous-carte est créée. Le filtre est alors réinitialisé tandis qu’un facteur odométrique lie la pose de la nouvelle sous-carte à la précédente. Chaque sous-carte est enfin associée à un nuage de point reconstruit à partir des observations du banc stéréo qui équipe le robot.

Les sous-cartes constituent l’unité élémentaire d’échange d’informations entre les robots. Chaque robot de la flotte vérifie régulièrement quels sont les robots avec lesquels il peut communiquer et leur transmet toutes les sous-cartes qu’il ne leur avait pas encore transmises, ainsi que toutes les observations inter-robot qui implique l’émetteur et le récepteur. Ces observations inter-robot jouent un rôle essentiel dans le pré-alignement des sous-cartes. Les auteurs de [2] proposent de calculer des contraintes analogues à des fermetures de boucles en recalant les nuages de points des différentes sous-cartes entre elles. Le processus est le suivant : on affine le pré-alignement des nuages de points dont les boîtes englobantes s’intersectent suffisamment – comme représenté dans la figure (4b) en calculant des descripteurs géométriques de type C-SHOT et en cherchant des correspondances entre les deux nuages de points sur la base de ces descripteurs. Enfin, une pose relative est estimée entre les deux nuages de points par ICP, ce qui permet de rajouter une contrainte entre les poses de deux sous-cartes impliquées. La covariance associée à cette estimation est approximée par une matrice sphérique dont les coefficients diagonaux sont calculés selon une méthode proposée dans [19] :

$$\sigma_x = \sigma_y = \sigma_z = \text{RMSE} \quad (10)$$

$$\sigma_{\{\text{roulis}, \text{tangage}, \text{lacet}\}} = \arctan \left(\frac{2 \cdot \text{RMSE}}{d_{\{yz, xz, xy\}}} \right) \quad (11)$$

où RMSE désigne la racine carrée de l’erreur quadratique moyenne des distances aux plus proches voisins dans les deux nuages de points alignés, et d_{yz} , d_{xz} et d_{xy} désignent respectivement le diamètre des intersections des

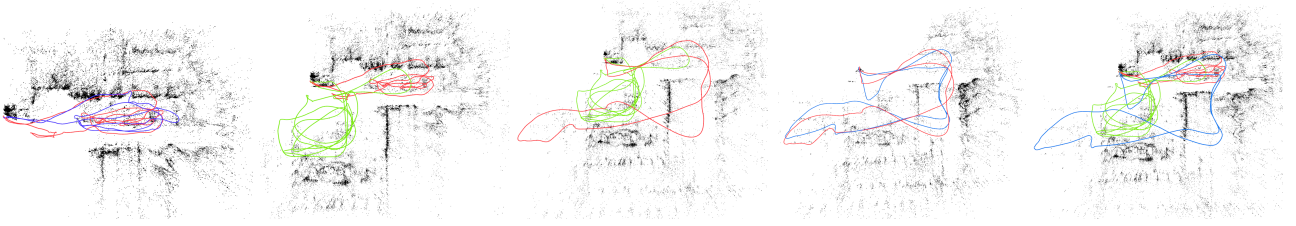


FIGURE 5 – Trajectoires des scénarios multi-robot utilisés. De gauche à droite : scénario 1 (MH1/MH2), scénario 2 (MH1/MH3), scénario 3 (MH3/MH4), scénario 4 (MH4/MH5) et scénario 5 (MH1/MH3/MH5)

deux nuages de points projetés sur les plans respectifs.

Le modèle résultant est un graphe de poses non-séquentiel présentant différents types de facteurs : facteurs odométriques, facteurs de pose relative entre les sous-cartes successives, facteurs dérivant des observations inter-robot et facteurs associés à l’alignement des nuages de points des différentes sous-cartes.

4.2 Problématiques associées au contexte visuel-inertiel monoculaire

Le contexte visuel-inertiel monoculaire impose des contraintes supplémentaires liées à la structure éparse des nuages de points estimés. Alors qu’en vision stéréo, tout amer 3D peut être instantanément triangulé, ce n’est pas le cas en vision monoculaire où plusieurs observations successives du même point depuis des points de vues différents sont nécessaires, et où le facteur d’échelle n’est rendu observable que grâce aux contraintes inertielles. On a donc intérêt à privilégier la construction de sous-cartes plus grandes afin que le nuage de points et la trajectoires soient suffisamment contraints. En visuel-inertiel monoculaire, on pourra par exemple utiliser un filtre tel que le *Multi-State Constraint Kalman Filter* (MSCKF) [20].

En conséquence, il est difficile d’extraire des descripteurs géométriques de tels nuages, à l’instar des descripteurs C-SHOT mentionnés précédemment, car ils ne sont pas suffisamment denses pour permettre l’extraction des normales nécessaires à leur calcul. Ainsi, les observations inter-robot doivent jouer un rôle prépondérant dans le pré-alignement des sous-cartes en vue de l’ICP. De plus, les nuages monoculaires étant moins denses et moins précis que leurs analogues stéréo, leur alignement par ICP s’en retrouvera moins concluant, et l’incertitude sur les transformations estimées calculée selon les équations (10) et (11) en cas d’ICP fructueuse s’en retrouvera accrue.

5 Évaluation des performances

Les performances des deux méthodes présentées ci-dessus ont été implémentées à l’aide du framework C++ MAPLAB [17] développée par l’ETH de Zürich, dans le cadre de scénarios multi-robot construits à partir du jeu de données EUROC [3]. Dans la suite, on se réfère à la méthode introduite dans la section III comme la méthode I, et à la méthode

introduite dans la section IV comme la méthode II.

5.1 Scénarios étudiés

À partir des séquences *Machine Hall* d’EUROC, on construit 5 scénarios multi-robot en confrontant les trajectoires suivantes tel qu’indiqué dans le tableau 1. Les trajectoires correspondantes sont représentées dans la figure (5). Pour chaque scénario, les observations inter-robots sont simulées en parcourant conjointement les trajectoires des robots en présence et en testant à chaque pas de temps si les conditions d’observabilité entre chaque paire de robot – à savoir distance inter-robot et visibilité – sont satisfaites. Si c’est le cas, une mesure de pose relative est générée selon une distribution gaussienne multivariée caractérisée par des écarts types σ_{position} et $\sigma_{\text{orientation}}$ sur la position relative et l’orientation relative.

Dans les scénarios 1 et 4, les deux trajectoires sont similaires et riches en fermetures de boucles intra et inter-robot, tandis que les scénarios 2, 3 et 5 confrontent des trajectoires plus hétérogènes et moins riches en fermetures de boucles inter-robot. Les trajectoires des robots sont estimées par l’algorithme d’odométrie visuelle-inertielle ROVIO [17] et affinées par la détection de fermetures de boucles et leur optimisation par ajustement de faisceaux (*Bundle Adjustment*). Les performances d’estimation de chaque robot sur sa trajectoire ainsi que sa distance parcourue et sa durée de parcours figurent dans le tableau 1.

L’objectif de la simulation basée sur le rejeu des séquences EUROC est de tester la capacité des différents robots à corréler les informations qu’ils reçoivent des autres robots à leur propre carte afin d’estimer correctement leurs trajectoires et de recalibrer les représentations de l’environnement communiquées. Les trajectoires estimées seront évaluées en calculant leurs erreurs quadratiques moyennes en position (*Root Mean Squared Error* (RMSE)) par rapport à la vérité terrain.

5.2 Paramètres de simulation

Les valeurs des paramètres introduits précédents pour les deux méthodes décrites et utilisées lors des simulations sont récapitulées dans le tableau 2.

Séq.	Distance Durée RMSE			Scénarios multi-robot				
	[m]	[s]	[m]	1	2	3	4	5
MH1	76	149	0.013	✗	✗			✗
MH2	69	183	0.016	✗				
MH3	132	87	0.011		✗	✗		✗
MH4	92	87	0.094			✗	✗	
MH5	100	98	0.032				✗	✗

TABLE 1 – Caractéristiques des séquences, RMSE moyens sur les trajectoires mono-robot et scénarios multi-robot

MÉTHODE 1		MÉTHODE 2	
Paramètre	Valeur	Paramètre	Valeur
n_c^{lim}	20	d_{max}	6.0 [m]
r_c^{lim}	0.2	θ_{max}	90.0 [deg]
n_{amers}	100		
$n_{\text{observations}}$	100		

TABLE 2 – Paramètres utilisés lors des simulations

5.3 Résultats de simulation

Les RMSE moyens associés aux trajectoires estimées dans les cartes des différents robots après intégration des données reçues par eux, pour les différents scénarios, sont donnés dans les tableaux 3 et 4. Le tableau 5 renseigne les performances de corrélation entre les informations reçues par chaque robot dans les différents scénarios et sa propre trajectoire.

		Robot 1	Robot 2	Robot 3	Robot 4	Robot 5
MÉTHODE 1	Sc. 1	0.013	0.016	—	—	—
	Sc. 2	0.018	—	0.015	—	—
	Sc. 3	—	—	0.011	0.090	—
	Sc. 4	—	—	—	0.038	0.032
	Sc. 5	0.014	—	0.018	—	0.019
MÉTHODE 2	Sc. 1	0.055	0.114	—	—	—
	Sc. 2	0.063	—	0.220	—	—
	Sc. 3	—	—	0.254	0.309	—
	Sc. 4	—	—	—	0.350	0.066
	Sc. 5	0.066	—	0.350	—	0.163

TABLE 3 – RMSE moyens [m] en position absolue sur l’estimation par chaque robot de sa propre trajectoire dans chaque scénario et chaque méthode. Les indices des robots sont relatifs au numéro de leur séquence d’EUROC associée.

Tout d’abord, les trajectoires issues d’EUROC étant déjà très riches en fermetures de boucles intra-robot, elles peuvent déjà être estimées avec une bonne précision comme le montrent les valeurs de RMSE du tableau 1. Dans ces conditions, les informations reçues de la part des autres robots ne leur permettent pas d’améliorer sensiblement cette estimation. En revanche, les deux méthodes devraient pouvoir y contribuer dans le cas de trajectoires pauvres en fermetures de boucle intra-robots mais potentiellement riches en fermetures de boucles inter-robot.

Si l’on s’intéresse aux performances d’estimation des tra-

Traj.	Mét. 1	Mét. 2	Traj.	Mét. 1	Mét. 2
Scénario 1			Scénario 4		
R1 \Leftarrow R2	0.116	0.736	R4 \Leftarrow R5	0.103	0.443
R2 \Leftarrow R1	0.056	0.720	R5 \Leftarrow R4	0.152	0.347
Scénario 2			Scénario 5		
R1 \Leftarrow R3	0.063	0.251	R1 \Leftarrow R3	0.223	0.222
R3 \Leftarrow R1	0.220	0.084	R1 \Leftarrow R4	0.511	0.329
Scénario 3			R3 \Leftarrow R1	0.436	0.197
R1 \Leftarrow R3	0.303	0.300	R3 \Leftarrow R5	0.066	0.101
R3 \Leftarrow R1	0.250	0.316	R5 \Leftarrow R1	0.273	0.243
			R5 \Leftarrow R3	0.063	0.063

TABLE 4 – RMSE moyens [m] en position absolue sur l’estimation par chaque robot des trajectoires des autres robots dans chaque scénario et chaque méthode. La notation $Ri \Leftarrow Rj$ désigne la trajectoire du robot j estimée dans la carte du robot i . Les indices des robots sont relatifs au numéro de leur séquence d’EUROC associée.

Robot	MÉTHODE 1	MÉTHODE 2
	Nombre d’amers 3D reçus et fusionnés	Nombre d’ICP réussies avec les sous-cartes reçues
Scénario 1		
Robot 1	888 (37%)	22
Robot 2	857 (29%)	21
Scénario 2		
Robot 1	200 (5%)	12
Robot 3	397 (13%)	12
Scénario 3		
Robot 3	59 (3%)	1
Robot 4	89 (2%)	1
Scénario 4		
Robot 4	802 (29%)	3
Robot 5	731 (30%)	3
Scénario 5		
Robot 1	221 (3%)	16
Robot 3	448 (8%)	14
Robot 5	253 (4%)	16

TABLE 5 – Nombre d’amers 3D reçus et fusionnés avec la méthode 1, et nombre de contraintes détectées avec les sous-cartes reçues

jectoires des autres robots, les tableaux 3 et 4 montrent la méthode 1 permet globalement d’obtenir de meilleurs résultats que la méthode 2. À cela, plusieurs explications. Tout d’abord, la méthode 1 repose sur la détection de correspondances inter-robot à l’aide des échantillons d’informations visuelles échangés entre les robots pour ultérieurement fusionner des amers 3D. Ces méthodes de corrélation s’avèrent plus robustes que les techniques de correspondances structurelles et géométriques employées dans la méthode 2. En particulier, cette dernière est tributaire de la qualité des pré-alignement des sous-cartes à l’aide des mesures inter-robots disponibles. Par ailleurs, on note que la quantité d’ICP fructueuses qui permettent d’aboutir à la

formulation de contraintes entre les poses des différentes sous-cartes, est très inégale entre les différents scénarios.

6 Conclusion

Dans cet article, on a proposé et simulé deux méthodes décentralisées de partage de cartes et de trajectoires en contexte visuel-inertiel monoculaire. Ces deux méthodes sont respectivement inspirées de méthodes développées pour le SLAM sous-marin acoustique et le SLAM visuel stéréo. Si ces méthodes ne permettent pas d'améliorer la qualité de la carte des robots récepteurs dans les scénarios testés, elles permettent néanmoins de recalculer correctement les trajectoires des autres robots et les informations cartographiques qui y sont associées, tout en transmettant une quantité de données moindre.

Remerciements

Nous remercions la Direction Générale de l'Armement (DGA) pour le financement de ces travaux.

Références

- [1] L. Paull, G. Huang, M. Seto, and J. J. Leonard, "Communication-constrained multi-robot cooperative slam," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 509–516.
- [2] M. J. Schuster, C. Brand, H. Hirschmüller, M. Suppa, and M. Beetz, "Multi-robot 6d graph slam connecting decoupled local reference filters," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 5093–5100.
- [3] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
- [4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping : Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [5] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3962–3970.
- [6] M. Karrer, P. Schmuck, and M. Chli, "Cvi-slam collaborative visual-inertial slam," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2762–2769, 2018.
- [7] F. Li, S. Yang, X. Yi, and X. Yang, "Corb-slam : a collaborative visual slam system for multiple robots."
- [8] R. Mur-Artal and J. D. Tardós, "Orb-slam2 : An open-source slam system for monocular, stereo, and rgbd cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [9] L. Contreras, O. Kermorgant, and P. Martinet, "Efficient decentralized collaborative mapping for outdoor environments," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2018, pp. 56–63.
- [10] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, "Generic node removal for factor-graph slam," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1371–1385, 2014.
- [11] M. Mazuran, W. Burgard, and G. D. Tipaldi, "Nonlinear factor recovery for long-term slam," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 50–72, 2016.
- [12] J. Vallvé, J. Solà, and J. Andrade-Cetto, "Graph slam sparsification with populated topologies using factor descent optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1322–1329, 2018.
- [13] K. Eickenhoff, L. Paull, and G. Huang, "Decoupled, consistent node removal and edge sparsification for graph-based slam," in *IROS*, 2016, pp. 3275–3282.
- [14] A. Cunningham, V. Indelman, and F. Dellaert, "Ddf-sam 2.0 : Consistent distributed smoothing and mapping," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5220–5227.
- [15] M. T. Lazaro, L. M. Paz, P. Pinies, J. A. Castellanos, and G. Grisetti, "Multi-robot slam using condensed measurements," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1069–1076.
- [16] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv :1812.01537*, 2018.
- [17] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab : An open framework for research in visual-inertial mapping and localization," *arXiv preprint arXiv :1711.10250*, 2017.
- [18] J. Wang and E. Olson, "AprilTag 2 : Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 4193–4198.
- [19] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa, "Submap matching for stereo-vision based indoor/outdoor slam," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 5670–5677.
- [20] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Robotics and automation, 2007 IEEE international conference on*. IEEE, 2007, pp. 3565–3572.