



HAL
open science

Evaluation de localisation et cartographie simultanées de robot mobile par caméra couleur-profondeur embarquée

Eva Goichon, Fumio Kanehiro, P Vasseur, G. Caron

► To cite this version:

Eva Goichon, Fumio Kanehiro, P Vasseur, G. Caron. Evaluation de localisation et cartographie simultanées de robot mobile par caméra couleur-profondeur embarquée. ORASIS 2023, Laboratoire LIS, UMR 7020, May 2023, Carqueiranne, France. hal-04219455

HAL Id: hal-04219455

<https://hal.science/hal-04219455v1>

Submitted on 27 Sep 2023

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.

Evaluation de localisation et cartographie simultanées de robot mobile par caméra couleur-profondeur embarquée

E. Goichon^{1,2}, F. Kanehiro², P. Vasseur¹, G. Caron^{1,2}

¹Laboratoire MIS, Université de Picardie Jules Verne, Amiens, France

²CNRS-AIST JRL (Joint Robotics Laboratory), IRL, Tsukuba, Japon
eva.goichon@u-picardie.fr

Résumé

Cet article évalue le SLAM visuel RGB-D RTAB-Map selon les champs de vue de caméra de profondeur, jusque 120°.

Mots-Clés

Vision par ordinateur, VSLAM, RTAB-Map, Azure Kinect

1 Introduction

Les algorithmes de localisation et de cartographie simultanées visuelles (VSLAM) estiment le mouvement de caméra et reconstruisent la structure d'une scène inconnue [1]. Un VSLAM se compose de trois éléments : le suivi de pose de la caméra, la cartographie de la scène et la fermeture de boucle. L'image d'entrée est donc déterminante quant à la précision du VSLAM. Il est courant d'utiliser des caméras couleur-profondeur donnant directement des informations géométriques sur l'environnement, contrairement à une caméra couleur classique. Il existe une multitude d'algorithmes adaptés à l'utilisation de ces caméras, tels que OpenVSLAM [2], RS-SLAM [3], [4], ORB-SLAM 3 [5] ou encore RTAB-Map SLAM [6].

RTAB-Map est un algorithme pouvant être utilisé en intérieur ou en extérieur, robuste aux scènes dynamiques et multi-session fonctionnant sous le système ROS (Robotic operating system). De plus, il a l'avantage de fonctionner avec de nombreuses caméras dont notamment la caméra Azure Kinect DK de Microsoft. La suite de l'article illustrera le comportement de RTAB-Map en fonction du champ de vue de la caméra couleur-profondeur. Cela est permis expérimentalement grâce à la caméra Azure Kinect qui permet de capturer un champ de profondeurs jusque 120°.

2 Problématique

La caméra Azure Kinect DK est choisie pour cette étude car elle possède quatre modes de profondeur : un champ de vision étroit (NFOV, Fig. 1) et un champ de vision large (WFOV, Fig. 2), chacun avec deux résolutions (maximale ou *pixels* combinés par blocs de 2×2).

RTAB-Map, comme tous les VSLAM RGB-D, prend en entrée l'intersection des images en couleur et des images de profondeur (vue couleur à profondeur).



FIGURE 1 – NFOV

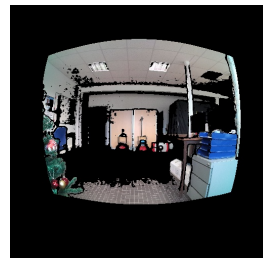


FIGURE 2 – WFOV

Les pixels noirs des Figures 1 et 2 illustrent l'absence de mesure de profondeur. La quantité de pixels réellement utilisée pour le SLAM est donc bien inférieure à la quantité d'information capturée. Cette différence est mise en évidence dans le Tableau 1. Les taux de pixels utilisés par rapport à la résolution de chaque mode de vue y sont regroupés. La première colonne met en lumière les défauts de mesure de profondeur. La deuxième montre comme le champ de vue étroit de la caméra couleur limite le champ de vue initialement plus large de la caméra de profondeur. La dernière colonne montre que la résolution de la caméra couleur est sous exploitée.

3 Expérimentations

Afin d'évaluer l'impact des différents modes de profondeur sur la trajectoire estimée par RTAB-Map, un comparatif sur robot mobile Pioneer 3 AT est réalisé. La caméra est montée sur le robot à l'aide d'un mat à 122 cm de hauteur du sol. Des trajectoires approximativement circulaires d'environ 12 m sont réalisées par pilotage manuel dans le volume

	Pixels utilisables de la vue couleur à profondeur	Pixels utilisables de la vue de profondeur	Pixels utilisables de la vue couleur
NFOV pixels non combinés	87,25 %	87,51 %	8,42 %
NFOV pixels combinés	91,53 %	91,45 %	2,22 %
WFOV pixels non combinés	74,36 %	42,21 %	11,06 %
WFOV pixels combinés	81,78 %	45,81 %	3,04 %

TABLE 1 – Proportions de pixels réellement utilisables dans l'image en fonction des différentes vues



FIGURE 3 – Pioneer 3AT équipé d’une Azure Kinect DK évoluant dans le volume du système Optitrack

de mesure d’un système de *capture de mouvement* Optitrack afin d’obtenir une vérité terrain précise et de comparer les résultats de localisation de RTAB-Map. L’ensemble des données nécessaires est enregistré via ROS sous forme de fichier rosbag. On réalise un parcours pour chaque mode de profondeur, disponible à 15Hz et 30Hz chacun excepté pour le WFOV à pixels non combinés qui n’existe pas à 30Hz.

4 Résultats

L’analyse des données est réalisée via l’outil EVO [7]. Un exemple en Figure 4 montre la trajectoire estimée par RTAB-Map, après fermeture de boucle (cf. *camera_poses*), et la pose déterminée par le système Optitrack comme référence (cf. *mocap_pose*).

Les erreurs de poses absolues (APE) et relatives (RPE) sont respectivement regroupées dans les Tableaux 2 et 3. RTAB-Map fonctionne sur toutes les trajectoires, sauf avec le mode de profondeur à large champ (environ 20% de la trajectoire estimée) dont les résultats ne sont donc pas reportés.

5 Conclusion et perspectives

Dans notre étude, le champ de vue large de caméra de profondeur à 15Hz donne les meilleures poses absolues et en moyenne les meilleures poses relatives. Cependant, la précision reste faible. La quantité de pixels utilisée par RTAB-Map étant infime en comparaison des données capturées

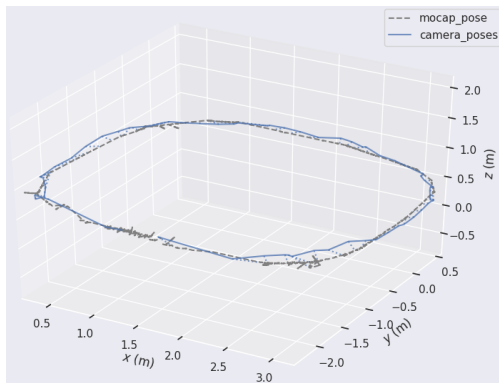


FIGURE 4 – Trajectoire calculée pour le mode de profondeur WFOV à pixels combinés acquis à 15Hz.

APE en mm	max	mean	median	min	rmse	sse	std
NFOV combiné 15Hz	484	214	211	47	239	4160	106
NFOV combiné 30Hz	575	253	269	35	283	5194	126
NFOV non combiné 15Hz	603	211	187	34	240	3524	115
NFOV non combiné 30Hz	291	164	155	30	177	2170	68
WFOV combiné 15Hz	262	136	120	58	145	1357	49
WFOV combiné 30Hz	354	196	193	46	208	2897	70

TABLE 2 – Erreurs de positions absolues de chaque mode de profondeurs en millimètres (en gras, les meilleures).

RPE en mm	max	mean	median	min	rmse	sse	std
NFOV combiné 15Hz	1147	298	273	19	354	9021	192
NFOV combiné 30Hz	2005	342	286	24	452	13057	295
NFOV non combiné 15Hz	1298	285	270	24	337	6821	180
NFOV non combiné 30Hz	1216	275	245	39	334	7569	188
WFOV combiné 15Hz	1688	265	238	34	354	8024	235
WFOV combiné 30Hz	1571	277	252	35	345	7877	207

TABLE 3 – Erreurs de positions relatives de chaque mode de profondeurs en millimètres (en gras, les meilleures).

par la caméra Azure Kinect, une nouvelle approche multi-caméra utilisant toutes les données capturées est en cours d’étude.

Références

- [1] Takafumi TAKETOMI, Hideaki UCHIYAMA et Sei IKEDA. “Visual SLAM algorithms : a survey from 2010 to 2016”. In : (2017).
- [2] Shinya SUMIKURA, Mikiya SHIBUYA et Ken SAKURADA. “OpenVSLAM : A Versatile Visual SLAM Framework”. In : 2019.
- [3] Teng RAN et al. “RS-SLAM : A Robust Semantic SLAM in Dynamic Environments Based on RGB-D Sensor”. In : (2021).
- [4] Chenyang ZHANG. “PL-GM :RGB-D SLAM With a Novel 2D and 3D Geometric Constraint Model of Point and Line Features”. In : (2021).
- [5] Carlos CAMPOS et al. “ORB-SLAM3 : An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM”. In : (2021).
- [6] Mathieu LABBÉ et François MICHAUD. “RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation”. In : (2019).
- [7] Michael GRUPP. *evo : Python package for the evaluation of odometry and SLAM*. <https://github.com/MichaelGrupp/evo>. 2017.