Ajustement de faisceaux du SLAM revisité en utilisant un capteur RGB-D

Kathia Melbouci, Sylvie Naudet Collette, Vincent Gay-Bellile, Omar Ait Aider, Mathieu Carrier, Michel Dhome

Ajustement de faisceaux du SLAM revisité en utilisant un capteur RGB-D

Kathia Melbouci[1] Sylvie Naudet Collette[1] Vincent Gay-Bellile[1] Omar Ait-Aider[2]
Mathieu Carrier [1] Michel Dhome [2]

[1] CEA, LIST, Laboratoire Vision et Ingénierie des Contenus
Point Courrier 94, Gif-sur-Yvette, F-91191 France
[2] Clermont Université, Université Blaise Pascal, LASMEA, BP 10448
Clermont-Ferrand / CNRS, UMR 6602, LASMEA, AUBIERE

kathia.melbouci@cea.fr

Résumé

Nous présentons dans ce papier une méthode qui intègre l'information de profondeur fournie par un capteur RGB-D, pour la cartographie et la localisation simultanée ou (Simultanuous Localization And Mapping, SLAM) afin d'améliorer la précision de la localisation. Nous présentons un nouvel ajustement de faisceaux local qui permet de combiner des données ayant une information de profondeur et des données visuelles dans une même fonction de coût totalement exprimée en pixels. L'approche proposée est évaluée sur des séquences de benchmark et comparée aux méthodes de l'état de l'art.

Mots Clef

Capteur RGB-D, Reconstruction et Localisation simultanée, RGB-D benchmark.

Abstract

We present a method of using depth information provided by an RGB-D sensor, for visual simultaneous localization and mapping (SLAM), in order to improve the localization accuracy. We present a new local bundle adjustment which allows to easily combine depth and visual data in cost function entirely expressed in pixel. The proposed approach is evaluated on a public benchmark dataset and compared to the state of art methods.

Keywords

RGB-D sensor, SLAM, RGB-D benchmark.

1 Introduction

Le problème de localisation et de cartographie simultanée (SLAM) consiste à construire incrémentalement une carte d'un environnement inconnu à partir d'une séquence d'images, et d'utiliser cette carte pour localiser la caméra. Couramment, les méthodes de SLAM visuel [1, 2, 4, 5, 10, 15] construisent une carte de l'environnement à partir de primitives mises en correspondance dans des images successives. Le mouvement de la caméra est simultanément éstimé à partir de cette carte souvent éparse.

Dans les méthodes de SLAM basé images clés, la pose de la caméra et le nuage de points 3D sont simultanément optimisés par un ajustement de faisceaux (Bundle Adjustment BA). Cependant le SLAM visuel présente plusieurs inconvénients, le SLAM est un processus incrémental donc sujet à l'accumulation d'erreurs. De plus, avec une seule caméra et sans connaissance a priori sur la géométrie de la scène, le SLAM permet d'estimer la rotation, mais la translation n'est connue qu'à un facteur d'échelle près. Ce dernier est fixé initialement mais dérive au cours du temps. Il est possible de fixer le facteur d'échelle et ainsi minimiser l'effet de la dérive en utilisant un capteur RGB-D. Ce type de capteur fournit en plus des images RGB, des images de profondeur synchronisées. Avec l'apparition des caméras Kinect de Microsoft, Primesense Carmine ou Asus Xtion [1] qui intègrent des capteurs 3D légers et de faible coût, les recherches exploitant des capteurs RGB-D ont connu un grand essor [3, 5, 7, 8, 16, 20].

La principale contribution de ce papier est de proposer une technique qui intègre la mesure de profondeur dans un algorithme de SLAM visuel monoculaire. Ceci implique plusieurs modifications dans l'algorithme. La plus importante se trouve dans la manière d'utiliser l'information de profondeur comme contrainte additionnelle dans l'ajustement de faisceaux.

Une approche similaire a été proposée par *Scherer et al.* [17, 18] qui ont revisité l'algorithme PTAM [10] afin d'y intégrer la profondeur. Cependant cette méthode repose sur un ajustement de faisceau minimisant des erreurs de reprojection et des erreurs sur les profondeurs et nécessite d'utiliser un facteur de pondération entre ces deux termes. Dans ce papier, nous proposons de revisiter l'algorithme du SLAM basé images clés proposé par [15]. Nous proposons une nouvelle fonction de coût pour l'ajustement de faisceaux, totalement exprimée en pixels, permettant de combiner d'une manière simple les informations visuelles et les

1. http ://www.asus.com/fr/Multimedia/Xtion PRO

informations de profondeur sans pondérer ces deux termes. Nous détaillons cette nouvelle approche, nous l'évaluons sur une base de données RGB-D publique [2]. Nous réalisons également une étude comparative par rapport aux approches de l'état de l'art.
La suite de ce papier est organisée comme suit.
Dans la section 2 nous analysons les travaux antérieurs. La section 3 explique la manière dont nous intégrons l'information de profondeur dans l'algorithme du SLAM visuel. Les résultats expérimentaux sont présentés dans la section 4. La conclusion et les perspectives sont présentés dans la section 5.

2 Travaux antérieurs

L'utilisation d'un capteur RGB-D dans les algorithmes de cartographie et de localisation simultanées a été ces dernières années un axe de recherche très actif. Ces capteurs fournissent en plus de l'image couleur, la mesure de profondeur de chaque pixel. Contrairement aux caméras stéréoscopiques, l'environnement ne nécessite pas d'être texturé pour avoir une carte de profondeur dense. Grâce à cette information de profondeur l'ambiguïté sur l'échelle dans ce type d'algorithmes peut être résolue, l'initialisation de la carte de l'environnement et l'estimation de la pose de la caméra sont simplifiés [7, 17].

Une des approches les plus connues utilisant le capteur RGB-D est KinectFusion proposée par [16]. Les auteurs définissent un volume statique dans lequel ils fusionnent des cartes de profondeurs en discrétisant une fonction de distance signée. La valeur de cette distance en chaque voxel correspond à la distance signée du voxel à la surface la plus proche. Les poses des caméras sont estimées en alignant deux nuages de points par l'algorithme (Iterative closest point ICP). *Whelan et al*. [20, 21] ont montré comment KinectFusion pouvait être étendu pour les grands environnement grâce à une représentation plus efficace de la mémoire, en déplaçant le volume représentant la scène avec le déplacement de la caméra.
Meilland et al. dans [13] proposent un modèle qui unifie les avantages d'une représentation volumétrique dense avec une représentation basée sur des images clés permettant une cartographie de l'environnement dense et précise sur de grandes trajectoires, sans avoir besoin de corriger la dérive. Ils proposent de fusionner des images clés qui vont représenter un modèle 3D, afin de prédire une seule image de référence. La pose de la caméra est estimées en minimisant une erreur photométrique et une erreur géométrique.

Les méthodes citées précédemment se concentrent essentiellement sur la reconstruction du modèle de l'environnement, de ce fait elles peuvent fournir une cartographie 3D précise de la scène observée mais elles sont couteuses en temps de calcul, et nécessitent souvent un puissant processeur graphique pour être temps réel. Les auteurs dans [9], ont proposé une approche temps réel en CPU, où la pose de la caméra est estimée par alignement d'images. Ils minimisent une erreur photométrique entre deux images successives en intégrant l'information de profondeur. *Henry et al.*[7] ont proposé une approche hybride qui combine des informations visuelles et des informations géométriques. Pour le calcul de la pose de la caméra, ils proposent de combiner un système basé sur les points d'intérêt détectés et mis en correspondance, avec un alignement de deux nuages de points denses. A chaque image, ils extraient des amers visuels par l'algorithme SIFT [12]. Ces primitives sont mises en correspondance et utilisées pour estimer une pose par un RANSAC [6]. Un ICP est initialisé avec cette pose qui sera ensuite améliorée en minimisant une erreur d'alignement. Cette erreur est une combinaison d'une distance entre deux nuages de points denses et d'une distance entre les primitives visuelles. Cette méthode fournit une carte peu précise, mais elle est capable de traiter de grandes trajectoires sans dérive importante. Une autre approche similaire proposée par [3], exploite aussi des points d'intérêt mis en correspondance pour initialiser un ICP. Une optimisation globale par graphe de poses est appliquée pour raffiner la pose de la caméra. *Huang et al.* ont proposé une méthode d'odometry visuelle connue sous le nom de (Fast Odometry From Vision FOVIS) [8] inspirée des travaux de [7]. Ils calculent une pose initiale à partir de données visuelles et la raffine en utilisant des informations 3D.

Les méthodes citées précédemment utilisent uniquement les points avec des profondeurs associées, tous les autres sont rejetés. Généralement, les caméras RGB-D ont des portées limitées, des problèmes avec la lumière du soleil, les discontinuités dans les cartes de profondeur et les surfaces réfléchissantes ou très absorbantes. Ce qui signifie que la mesure de profondeur n'est pas disponible sur tous les points de l'image. Pour pallier à cette limite, [17, 18, 22] proposent des approches qui exploitent l'information de profondeur lorsque celle-ci est accessible et utilisent que les informations visuelles dans le cas où les mesures de profondeurs sont indisponibles.

Scherer et al. [17] ont récemment proposé une extension de l'algorithme PTAM [10] pour prendre en compte les données de profondeur. Ils ont suggéré plusieurs modifications du PTAM, la plus importante concerne l'ajustement de faisceaux. En effet ils ont étendu l'optimisation par ajustement de faisceaux en prenant en compte la profondeur des points dans la fonction de coût à minimiser. Cette dernière combine l'erreur de reprojection classique avec une erreur sur les profondeurs. Pour être combinées, ces erreur de différentes métriques (la première en pixel et la deuxième en mètre) doivent être pondérées selon leurs incertitudes. L'incertitude sur les données de profondeur est fonction d'un facteur de pondération, que les auteurs ont fixé expérimentalement, mentionnant que ce paramètre doit être adapté selon la complexité de la scène. Les mêmes auteurs ont proposé dans un second papier [18] une extension de ces travaux. En effet une des limitations du PTAM est le temps de calcul de l'ajustement de faisceaux global. Ce temps est conséquent quand le nombre de points 3D à

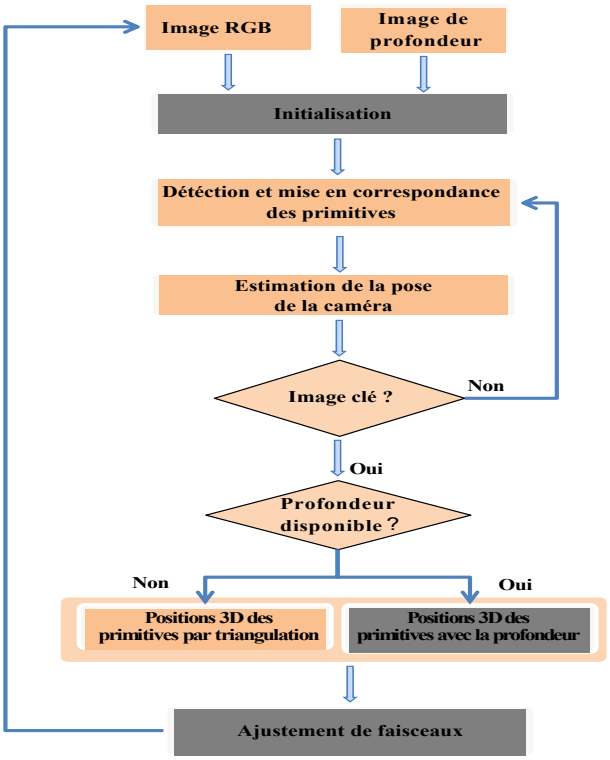

FIGURE 1 – Schéma récapitulatif de notre approche

optimiser devient important. Pour atténuer cette limite, ils proposent dans [18] de remplacer l'ajustement de faisceaux global par une optimisation par graphe de poses, où la fonction de coût combinant une erreur visuelle et une erreur sur les profondeurs est utilisée pour raffiner les branches du graphe. Ces branches représentent la pose relative entre chaque paire d'images clés.

Contribution

Notre méthode est une amélioration du SLAM visuel monoculaire basé sur des images clés. Les différences majeurs avec les deux solutions proposées par *Scherer et al.* dans [17] et [18] se trouvent dans :

- Le processus d'optimisation : dans notre méthode nous proposons un ajustement de faisceaux temporel local, alors que [17] propose un ajustement de faisceaux global, et que [18] proposent une optimisation par graphe de poses que sur deux images clés.
- La fonction de coût dans l'ajustement de faisceaux : on propose de combiner deux erreurs exprimées en pixels ; l'une dépend des informations visuelles seulement, et l'autre est fonction des profondeurs. [17, 18] proposent de combiner deux erreurs de métriques différentes.

Les performances de notre approche sont évaluées sur une base de données publique [2]. Elle est comparée aux méthodes de l'état de l'art : FOVIS [8], [18], et [17] que nous avons revisitées.

2. http ://vision.in.tum.de/data/datasets/rgbd-dataset

La solution proposée est illustrée dans la Figure 1. Elle comprend toutes les étapes du SLAM visuel (mise en correspondance des points d'intérêt, calcul de pose, enrichissement de la carte de l'environnement, et ajustement de faisceaux). Les modifications apportées au SLAM visuel [15] sont illustrées en gris.

A partir d'une seule image initiale, nous créons une carte de l'environnement : la position 3D de chaque primitive est obtenue à partir de ses coordonnées image et sa profondeur. A chaque nouvelle image des points d'intérêt sont extraits et mis en correspondance avec les points 3D reconstruits pour obtenir un ensemble d'associations $2D-3D$. A partir de ces associations, nous calculons la pose de la caméra courante avec l'algorithme d'estimation de pose de Grunert [11] dans un processus de RANSAC [6]. Alors que le calcul du mouvement de la caméra se fait à chaque image, l'enrichissement de la carte par de nouveaux points 3D ne se fait qu'aux images clés.

Ces nouveaux points 3D sont rajoutés à partir de leurs observations et de leurs profondeurs fournies par le capteur RGB-D. Cependant comme expliqué précédemment, la profondeur de certaines primitives est inexistante. Dans ce cas, les points 3D correspondants à ces primitives sont reconstruits par triangulation. La pose de la caméra et les points 3D reconstruits sont ensuite optimisés dans un ajustement de faisceaux.

Pour atteindre les performances du temps réel comme proposé dans [15], nous utilisons un ajustement de faisceaux local. Ce dernier optimise un nombre limité de caméras et de points 3D, observés sur une fenêtre temporelle de N caméras.

L'intégration de la mesure de profondeur dans l'ajustement de faisceaux est explicitée dans le section suivante.

3 Intégration de la profondeur dans l'ajustement de faisceaux

Nous proposons ici une nouvelle fonction de coût qui permet de combiner aisément l'information visuelle et l'information de profondeur, sans utiliser un facteur de pondération additionnel comme dans [17, 18].

Notation

Nous utilisons un modèle de caméra sténopé pour décrire la projection des points $3D$ exprimés dans le repère global, dans le repère $2D$ de l'image. On note P, la matrice de transformation qui permet de passer du repère monde au repère caméra.

$$P = \begin{bmatrix} R & t \\ 0_{1\times 3} & 1 \end{bmatrix} \quad (1)$$

Un point 3D en cordonnées homogènes est définie par : $Q = (X, Y, Z, 1)^T$. Ce point 3D est reconstruit à partir de ses coordonnées pixel $q = (u, v, 1)^T$ et de sa mesure de profondeur associée d, en utilisant la fonction de rétro-projection π^{-1} :

$$\pi^{-1}(q, d) = dK^{-1}q \quad (2)$$

K décrit les paramètres intrinsèques de la caméra :

$$K = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

avec :
f_u, f_v : les distances focales.
u_0, v_0 : les coordonnées du point principal.
Les coordonnées image du point Q peuvent être calculées avec la fonction de projection π :

$$\pi((x, y, z)^T) = (x/z, y/z)^T \quad (4)$$

$$q = \pi(KPQ) \quad (5)$$

L'ajustement de faisceaux classique

L'ajustement de faisceaux local optimise les poses des caméras et la structure 3D de la scène en minimisant une erreur de projection 2D dans les N dernières images clés. Cette erreur est la différence entre la projection du point Q_i dans l'image I_j, et son observation correspondante $q_{i,j}$ sur cette même image. Pour permettre des traitement en temps réel, l'ajustement de faisceaux local optimise seulement les N_c dernièrse caméras clés et les N_p points 3D observés dans les N dernières images clés, avec $N \geqslant N_c$.
La fonction de coût est donc définie par :

$$\varepsilon_{\text{slam}}\Big(\{P_j\}_{j=0}^{N_c}, \{Q_i\}_{i=0}^{N_p}\Big) = \sum_{i=0}^{N_p} \sum_{j \in A_i} \rho_s\Big(q_{i,j} - \pi(KP_jQ_i),\ a_s\Big)$$

Où A_i est l'ensemble des images clés observant le point Q_i.
$\rho_s(., a_s)$: l'estimateur de Geman-McClure, avec a_s le seuil de rejet estimé par l'écart médian absolu (Absolute Median Deviation MAD).

L'ajustement de faisceaux avec l'erreur sur la profondeur

Nous nous sommes inspirés des travaux de [17] en rajoutant l'erreur sur la profondeur à l'erreur de reprojection 2D. La nouvelle fonction de coût pour l'ajustement de faisceaux est illustrée dans les Équations 6 et 7. Nous l'avons intégrée dans notre algorithme de SLAM revisité. Nous appelons cette méthode DSLAM. Comme mentionnée dans [17, 18] l'erreur de projection classque ($\varepsilon_{\text{slam}}$), et l'erreur sur les profondeurs des primitives (ε_{d}) sont de métriques différentes. Pour être combinées, ces deux erreurs doivent être pondérées selon leurs incertitudes. L'incertitude sur les profondeurs est fonction d'un facteur a. Pour les expérimentations nous allons utiliser le facteur de pondération recommandé par [17], $(a = 3.33110^{-3})$.

$$\varepsilon_{\text{d}}\Big(\{P_j\}_{j=0}^{N_c}, \{Q_i\}_{i=0}^{N_p}\Big) = \sum_{i=0}^{N_p} \sum_{j \in A_i} \rho_d\Big(\frac{d_{i,j} - [P_jQ_i]_z}{ad_{i,j}^2},\ a_d\Big) \quad (6)$$

Avec $d_{i,j}$ la profondeur du point Q_i mesurée par le capteur RGB-D, dans l'image j.

$$\varepsilon_{dslam}\Big(\{P_j\}_{j=0}^{N_c}, \{Q_i\}_{i=0}^{N_p}\Big) = \varepsilon_{\text{d}} + \varepsilon_{\text{slam}} \quad (7)$$

L'ajustement de faisceaux proposé

En plus de l'erreur de reprojection 2D classique mentionnée plus haut, nous intégrons la mesure de profondeur comme contrainte supplémentaire dans l'ajustement de faisceaux pour améliorer sa précision et sa robustesse. Contrairement à [17] qui est une combinaison d'une erreur de reprojection 2D conventionnelle et d'une contrainte 1D sur la profondeur, nous proposons de combiner des données visuelles et des données de profondeur dans une fonction de coût totalement exprimée en pixel.

Pour cela, nous mesurons la position 3D pour chaque primitive 2D détectée dans l'image I_k et ayant des correspondants sur au moins 2 images clés. A partir de sa coordonnée image $q_{i,k}$ et sa profondeur $d_{i,k}$. Nous transformons ce point dans le repère global à partir de l'Équation 9.

$$\pi^{-1}(q_{i,k}, d_k) = d_{i,k}K^{-1}q_{i,k} \quad (8)$$

$$Q_{i,k} = P_k^{-1}\pi^{-1}(q_{i,k}, d_{i,k}) \quad (9)$$

Pour chaque point $3D$ $Q_{i,k}$, généré à partir de sa profondeur mesurée dans la caméra clé k, on calcule son erreur de projection dans chaque image clef qui l'observe, avec $(j \in A_i \text{ et } j \neq k)$.

La fonction de cout résultante est décrite dans l'Équation 10 :

$$\varepsilon_{\text{depth}}\Big(\{P_j\}_{j=0}^{N_c}\Big) = \sum_{i=0}^{N_p} \sum_{j \in A_i} \sum_{k \in A_i}^{k \neq j} \rho_d\Big(q_{i,j} - \pi\Big(KP_jP_k^{-1}\pi^{-1}(q_{i,k}, d_{i,k})\Big),\ a_d\Big) \quad (10)$$

Pour réduire l'influence des points aberrants, dues aux erreurs de mise en correspondance ou aux bruits sur les profondeurs, on applique un estimateur robuste de type Geman-McClure $\rho_d(., a_d)$, avec a_d définissant le seuil de rejet estimé par le MAD(Absolute Median Deviation).
La fonction de coût résultante, qui prend en compte l'erreur de projection et la contrainte sur les profondeurs des primitives est la suivante :

$$\varepsilon\Big(\{P_j\}_{j=0}^{N_c}, \{Q_i\}_{i=0}^{N_p}\Big) = \varepsilon_{\text{depth}} + \varepsilon_{\text{slam}} \quad (11)$$

L'Équation 11 est minimisée par l'algorithme Levenberg-Marquart [14]. Notons que l'ajustement de faisceaux proposé conserve la structure creuse par blocs des matrices impliquées par l'optimisation. Celle-ci est donc implémentée de manière efficace en tenant compte des ces structures creuses comme dans [19].

4 Évaluation et résultats expérimentaux

L'approche proposée a été évaluée et comparée au SLAM visuel de [15] et au DSLAM cité précédemment, sur une séquence de synthèse $(154m)$ représentant des couloirs et des bureaux. Les résultats illustrés dans la Figure 2 et le Tableau 1 montrent que l'intégration de l'information de profondeur dans le SLAM visuel réduit significativement sa dérive. Nous obtenons une erreur moyenne sur

Méthode \ Erreur [m]	RMSE	STD
OURS	0.228	0.187
DSLAM	0.209	0.192
VISUAL-SLAM	1,101	0.575

TABLE 1 – Erreurs de position absolue sur la séquence de synthèse .

toute la séquence de 0.2m avec notre approche alors que l'erreur du SLAM visuel dépasse le mètre.
Pour étudier plus en détail l'influence de notre solution sur de larges environnements, nous l'avons évaluée sur une séquence réelle tirée de la base de données "TUM RGB-D benchmark" [3]. Cette base de donnée contient des images couleurs et des images de profondeurs acquises avec une caméra Microsoft Kinect à 30Hz avec une résolution de (640×480). Elle comporte aussi les trajectoires de la vérité de terrain du capteur, obtenues à partir d'un système de capture de mouvement de haute précision grâce à huit caméras de suivi à grande vitesse (100 Hz).
Sur cette même séquence, nous avons également comparé notre méthode avec FOVIS [8] qui est publiquement disponible [3] et aux travaux publiés par *Scherer* dans [18].

Nous avons comparé les trajectoires estimées par chaque méthode avec celles données dans la vérité terrain. Pour cette comparaison nous avons utilisé l'erreur de translation absolue (Absolute Translation Error ATE) fournie avec les outils de la base de données.
Les résultats présentés dans la Figure 3 et le Tableau 2 montrent que rajouter l'information de profondeur comme contrainte supplémentaire dans l'ajustement de faisceaux du processus du SLAM visuel améliore significativement sa précision : L'erreur de position est réduite d'un facteur 10. Notre solution présente de meilleurs résultats que la méthode FOVIS, l'erreur de position est réduite de $13cm$. Notre résultat est également meilleur que celui présenté par la méthode de *Scherer* dans [18]. Cependant, les résultats obtenus avec notre solution sont similaires aux résultats obtenus avec le DSLAM. La seule différence entre ces deux approches se trouve dans la fonction de coût de l'ajustement de faisceaux. En effet DSLAM nécessite un facteur de pondération additionnel pour pouvoir combiner deux erreurs de métriques différentes : une erreur 2D en pixel et une erreur 1D en mètre, ce facteur dépend de l'incertitude sur chaque erreur, et doit être adapté selon la complexité de la scène. Notre approche combine deux erreurs de même métrique par conséquent elles ne nécessite pas un facteur de pondération supplémentaire.

Sur la séquence Freiburg3, on a aussi évalué les temps de calcul de notre système sur un Intel Xeon W3570 à 3.20 GHz en utilisant 1 cœur.

3. https ://code.google.com/p/fovis

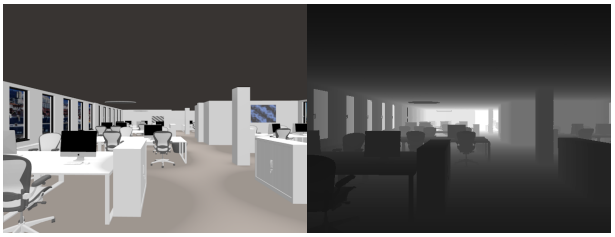

Image couleur Image de profondeur

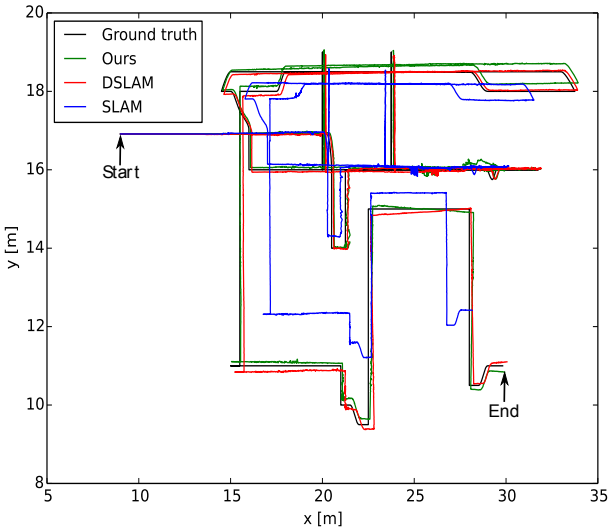

FIGURE 2 – Projection dans le plan-xy de la vérité terrain et des trajectoires estimées par le SLAM et le DSLAM, sur la séquence de synthèse.

Nous avons mesuré que le temps de calcul moyen nécessaire pour traiter chaque image est de ($\sim 25ms$), l'estimation du mouvement de chaque caméra est de ($\sim 20ms$) en moyenne. L'ajustement de faisceaux requière ($38ms$) en moyenne.

5 Conclusion

Ce papier présente une technique pour intégrer l'information de profondeur dans un algorithme de SLAM visuel monoculaire. L'idée majeure dans ce papier est d'utiliser les profondeurs des primitives comme contraintes additionnelles dans l'ajustement de faisceaux. Une idée similaire a été investiguée dans [17], où les auteurs ont revisité l'algorithme PTAM, en rajoutant la mesure de profondeur fournie par un capteur RGB-D. Dans ce papier, nous proposons de modifier l'algorithme du SLAM visuel proposé par [15]. Pour cela nous décrivons une nouvelle fonction de coût qui permet de grouper d'une manière efficace et simple, une erreur sur les données visuelles et une erreur qui dépend des données visuelles et de leurs profondeurs. Ces deux erreurs sont exprimées en pixel, par conséquent elles ne nécessitent pas de facteur de pondération pour être combinées contrairement à l'approche proposée par [17]. Cette méthode est évaluée sur une base de données de bench-

Erreur [m] / Méthode	RMSE	STD
OURS	0.068	0.039
DSLAM	0.069	0.052
Scherer in[18]	0.136	-
FOVIS	0.207	0.117
VISUAL-SLAM	0.652	0.204

TABLE 2 – Erreurs de position absolue sur la séquence Freiburg3-long-office-household.

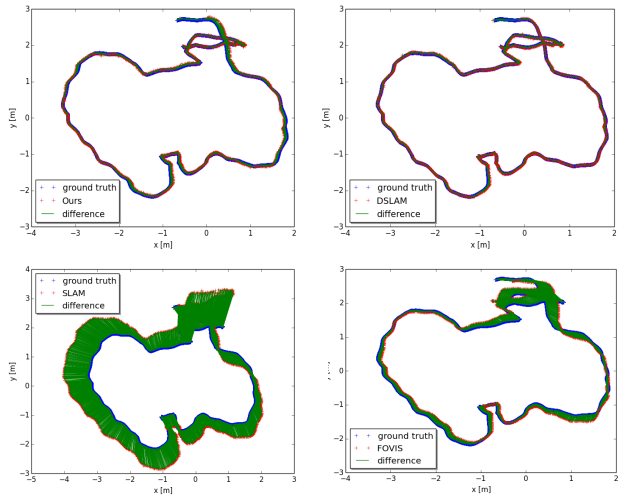

FIGURE 3 – Projection dans le plan-xy de la vérité terrain et des trajectoires estimées par les différentes approches, sur la séquence freiburg3-long-office-household.

mark et comparée aux récentes méthodes de l'état de l'art incluant [17, 18]. Cette évaluation montre que l'exploitation de l'information de profondeur réduit la dérive en facteur d'échelle du SLAM visuel et améliore sa précision. Cette nouvelle fonction de coût permet de garder la structure creuse des matrices engendrées par l'ajustement de faisceaux et par conséquent d'obtenir des temps de traitement encourageants (25ms en moyenne).

Nos futures travaux porteront sur l'amélioration de la robustesse de notre algorithme dans des zones avec peu de texture. Notons que la solution proposée ne se limite pas qu'aux caméras RGB-D. Elle peut être utilisée avec d'autres types de capteurs qui fournissent l'information de profondeur comme un laser couplé à une caméra.

Références

[1] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410. IEEE, 2003.

[2] Ethan Eade and Tom Drummond. Scalable monocular slam. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 469–476. IEEE, 2006.

[3] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1691–1696. IEEE, 2012.

[4] Jakob Engel, Jürgen Sturm, and Daniel Cremers. Camera-based navigation of a low-cost quadrocopter. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2815–2821. IEEE, 2012.

[5] Nikolas Engelhard, Felix Endres, Jürgen Hess, Jürgen Sturm, and Wolfram Burgard. Real-time 3d visual slam with a hand-held rgb-d camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*, volume 180, 2011.

[6] Martin A Fischler and Robert C Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981.

[7] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping : Using depth cameras for dense 3d modeling of indoor environments. In *In the 12th International Symposium on Experimental Robotics (ISER*. Citeseer, 2010.

[8] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *International Symposium on Robotics Research (ISRR)*, pages 1–16, 2011.

[9] Christian Kerl, Jurgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3748–3754. IEEE, 2013.

[10] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[11] Mingyang Li and Anastasios I Mourikis. High-precision, consistent ekf-based visual–inertial odometry. *The International Journal of Robotics Research*, 32(6) :690–711, 2013.

[12] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110, 2004.

[13] Maxime Meilland and Andrew I Comport. On unifying key-frame and voxel-based dense visual slam

at large scales. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3677–3683. IEEE, 2013.

[14] Jorge J Moré. The levenberg-marquardt algorithm : implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[15] E. Mouragnon, Maxime Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. CVPR, 2006.

[16] Richard A Newcombe, Andrew J Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. Kinectfusion : Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[17] Sebastian A Scherer, Daniel Dube, and Andreas Zell. Using depth in visual simultaneous localisation and mapping. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 5216–5221. IEEE, 2012.

[18] Sebastian A Scherer and Andreas Zell. Efficient onbard rgbd-slam for autonomous mavs. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1062–1068. IEEE, 2013.

[19] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision algorithms : theory and practice*, pages 298–372. Springer, 2000.

[20] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous : Spatially extended kinectfusion. 2012.

[21] Thomas Whelan, Michael Kaess, John J Leonard, and John McDonald. Deformation-based loop closure for large scale dense rgb-d slam. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 548–555. IEEE, 2013.

[22] Ji Zhang, M. Kaess, and S. S. Real-time depth enhanced monocular odometry.