

Systèmes Cyber-Physiques autonomes et communicants en milieux hostiles. Application à l'exploration par robots mobiles.

VIRGILE DAUGÉ

L.O.R.I.A.

virgile.dauge@inria.fr

5 février 2018

Résumé

L'objectif de ce rapport est de présenter un résumé de l'avancement de mes travaux. Premièrement, un état de l'art évoquant les principales technologies permettant l'exploration autonome sera dressé, suivie par une analyse des solutions existantes ainsi qu'une présentation des axes de développement dégagés.

I. INTRODUCTION

LA robotique autonome est en plein essor, rendu possible à la fois par l'arrivée de capteurs compacts et performants (caméras stéréos, Lidars...) et l'augmentation de la puissance de calcul embarquée. Avant même d'effectuer une mission spécifique, un robot doit être capable de percevoir et d'interagir avec son environnement. Dans une grande majorité des systèmes actuels, cette perception de l'environnement est déléguée à un ou plusieurs éléments extérieurs (GPS, Systèmes de captures de mouvements, connaissances à priori, intervention humaine). Ceci n'est pas envisageable dans de nombreuses situations, pour des raisons d'inaccessibilité, de coûts, de dangerosité ou encore d'absence de connaissances préalables. Afin de rendre un système réellement autonome, il est nécessaire de mettre en place des mécanismes de perception basés uniquement sur les capacités internes du système, les différents capteurs intégrables ainsi que la puissance de calcul disponible. En effets, cer-

taines tâches nécessitent d'être effectuées en permanence à une fréquence suffisamment élevée. C'est le cas par exemple de la détection et l'évitement d'obstacles, où la fréquence de rafraîchissement est directement liée à la vitesse de déplacement du système. La première nécessité est de donner au système cyber-physique la capacité de cartographier son environnement (même à un niveau de détail faible) et de se positionner au sein de cette carte¹

II. ETAT DE L'ART

Le SLAM étant une tâche complexe, il est souvent composé de nombreux algorithmes interagissant ensemble. Une solution complète repose souvent sur le travail de plusieurs équipes, et est souvent adaptée à un cadre bien particulier. La première difficulté à franchir consiste à estimer correctement sa position, et par conséquent les déplacements successifs de notre robot. Un rapide aperçu des techniques d'odome-

1. Principalement appelé Simultaneous Localisation and Mapping [1][2]

trie sera donc premièrement proposé. L'enchevêtrement de solutions actuellement connues forme un ensemble très hétérogène et difficile à analyser. Il m'a cependant paru pertinent de les diviser en deux grandes catégories : Plausibilité maximum (Maximum likelihood) et Intégration au sein d'un graphe (Graph SLAM aussi appelé Smoothing and mapping).

i. Techniques d'odometrie

Il existe un vaste panel de combinaison algorithmiques permettant d'estimer un déplacement à partir des capteurs embarquables dans un système cyber-physique mobile. Ne seront abordées ici que les solutions jugées pertinentes, adaptées à nos besoins. Il existe un capteur que l'on retrouve sur tous les drones, la centrale inertielle. Cette dernière permet de mesurer les accélérations subies par le système, donnant ainsi une indication de déplacement. Bien que difficiles à calibrer et présentant une estimation assez grossière et fortement sujette aux dérives temporelles², les IMU servent souvent de base à d'autres estimateurs, fournissant ainsi une solution de départ, permettant de gagner du temps de calcul et parfois d'améliorer grandement la précision du résultat final. Autre capteur abordable et fréquemment présent sur des drones, la caméra (qu'elle soit monoculaire, stéréo ou omnidirectionnelle) permet de déterminer une estimation de mouvement, basée sur l'analyse d'amers³[3] puis sur l'appairage de ces amers[4] entre les différentes images. Capteur en pleine démocratisation, le Lidar (laser detection and ranging) produit désormais des nuages de points denses, à une fréquence élevée (10Hz), avec une portée d'une centaine de mètres sur 360 degrés avec une précision de $\pm 3cm$. L'appairage de points est par la suite appliqué pour estimer la transformation entre deux scans successifs. L'algorithme le plus couramment utilisé est Iterative Closest Point, il

consiste à appairer les points puis à minimiser de manière itérative l'écart de position entre les couples de points. Il existe en revanche une multitude d'implémentations, et les résultats sont fortement dépendants de l'estimation initiale ainsi que des nombreux paramètres du processus (échantillonnage, exclusion des points aberrants etc...)[5].

ii. Maximum Likelihood

L'approche de plausibilité maximale consiste à analyser des données de capteurs à chaque étape, afin de déterminer la solution la plus probable et de l'intégrer directement à la carte, souvent par fusion directe, interdisant tout repositionnement ou calcul postérieur. C'est le cas notamment de la solution la plus performante [6] eu égard des critères du benchmark de référence (KITTI) [7]. Cette solution est basée sur une estimation de déplacement combinant l'analyse des frames d'une caméra par Visual Odometry [8] ainsi qu'une comparaison de nuages de points de Lidars par l'algorithme d'Iterative Closest Point [5]. L'avantage de ce type de méthode consiste en sa "simplicité", et son coût réduit en termes calculs. L'inconvénient majeur réside dans l'impossibilité de corriger une erreur passée. En effet, une fois intégrée à la carte, une mesure est considérée comme valide et servira de référence pour les estimations futures, pouvant mener à une accumulation d'erreurs plus importante.

iii. Graph SLAM

Afin de permettre une intégration raffinable des estimations successives de mouvements, une approche probabiliste a été développée. L'idée principale est d'appliquer une loi de probabilité à plusieurs variables (loi jointe) pour chaque étape temporelle (extended Kalman filters)[9]. Permettant ainsi de minimiser l'impact du bruit présent dans les mesures, tout en raffinant l'estimation de carte et de trajectoire au fur et à mesure de la progression. L'inconvénient principal de cette méthode réside dans la nécessité de mettre à jour l'intégralité du réseau à chaque étape, menant à un coût en

2. Les erreurs s'accumulent au fur et à mesure sans possibilité de correction, pouvant mener à des erreurs de grande amplitude.

3. Ensemble de pixels formant une structure facilement identifiable, et stable malgré les potentiels changements de points de vue.

calculs ne permettant pas son application en temps réel avec les capacités limitées embarquées dans un robot, surtout si l'on considère l'accumulation des mesures au fur et à mesure de l'exploration. Afin de pallier ce problème, une équipe du Georgia Institute of Technology a développé une méthode itérative[10] basée sur des réseaux Bayésiens et utilisant un graphe de factorisation factorisant la distribution de probabilités. En exploitant de surcroît les possibilités d'optimisation offertes par les matrices creuses, permettant de mettre à jour plus efficacement et rapidement le graphe (Certaines optimisations ont été apportées dans la V2 [11]). De plus, il est seulement nécessaire de réaliser des calculs pour les nouvelles mesures et non plus l'intégralité des observations depuis le début de l'expérience. S'il est possible d'utiliser ces algorithmes en temps réel, il reste toutefois difficile d'obtenir une estimation correcte de la covariance nécessaire à l'ajustement des mesures entre elles. Dans le cas de l'ICP, certaines solutions analytiques permettent d'obtenir une estimation en 2D [12] et en 3D [13]. Cependant, les covariances estimées sont parfois erronées, tant en direction qu'en échelle. Ce qui conduira nécessairement à une mauvaise intégration des mesures dans le graphe. Différents algorithmes basés sur des graphes exploitent des capteurs différents : une centrale inertielle (IMU) [14], une simple caméra monoculaire [15][16], une caméra stéréo [17] ou encore omnidirectionnelle [18]. Il est également possible d'ajouter un mécanisme de "loop closure", consistant à détecter un élément de l'environnement déjà connu et ajouter un lien avec la position fraîchement mesurée. Cela permet parfois de corriger une partie de la dérive (accumulation d'erreurs au fil du temps) et ainsi maintenir l'erreur de positionnement dans une intervalle acceptable.

III. ANALYSE

Comme nous l'avons montré précédemment, les différentes techniques de SLAM permettent d'exploiter différents capteurs et de produire une estimation de position plus ou moins précise, mais toujours soumise aux dérives.

En effet, même les techniques basées sur des graphes, censées corriger ce problème de dérive, souffrent d'une mauvaise estimation de la covariance. La position ainsi estimée ne permet pas d'établir une cartographie précise sur une grande échelle, subissant directement les effets de la dérive. Cette estimation de position de notre système est donc le goulet d'étranglement actuel de la robotique autonome, tout particulièrement quand la cartographie ne sert pas uniquement pour la navigation mais est également l'objectif final de la mission. Afin de réduire au maximum cette erreur, plusieurs pistes sont envisageables. Réduire directement les erreurs de mesures, en introduisant un mécanisme de mesure de position plus précis. Autre piste, être capables d'estimer de manière précise l'erreur de chaque mesure, afin de maximiser l'impact du graphe Bayésien sur la réduction de la dérive. Enfin, s'attaquer à la source même du problème de dérive, en essayant de contourner la nature dynamique de nos mesures.

IV. NOTRE APPROCHE

Nous développons actuellement un système de positionnement tentant de pallier les difficultés évoquées précédemment. La première étape consiste à évaluer l'erreur de chaque mesure en fonction de la position du drone dans son environnement et du modèle d'erreur des capteurs employés, couplés avec leurs positions. Le résultat de cette opération est construit sous forme de "Heatmap" d'erreur, dont la précision est réglable selon la puissance de calcul disponible. La figure 1 illustre un exemple construit pour une erreur de capteur *erreur* (en mètres) dépendante uniquement de la distance de mesure d (en mètres) tel que :

$$erreur = 0.01 + \frac{0.02 * d}{5} \quad (1)$$

Ce calcul d'erreur embrasse un triple emploi. Il permettra à la fois de vérifier/d'infirmer la validité de notre solution selon les capacités actuelles des capteurs en le comparant aux algorithmes existants, de minimiser l'erreur de

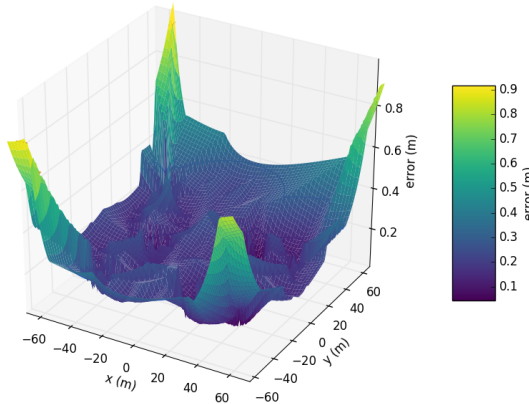


Figure 1 – Heatmap d'erreur générée (le modèle d'erreur du capteur est ici arbitraire : $1\text{cm} + 2\text{cm}$ tous les 5m)

positionnement en guidant l'algorithme d'exploration, évitant ainsi les mesures approximatives et enfin de renseigner plus précisément le graphe de factorisation sur le degré d'exactitude de nos mesures. Notre système de positionnement est en plein développement et fera l'objet de publications ultérieures.

RÉFÉRENCES

- [1] H. DURRANT-WHYTE et T. BAILEY, « Simultaneous localization and mapping : part i », *IEEE Robotics Automation Magazine*, t. 13, n° 2, p. 99–110, juin 2006, ISSN : 1070-9932. DOI : 10 . 1109 / MRA . 2006 . 1638022.
- [2] T. BAILEY et H. DURRANT-WHYTE, « Simultaneous localization and mapping (SLAM) : part II », *IEEE Robotics Automation Magazine*, t. 13, n° 3, p. 108–117, sept. 2006, ISSN : 1070-9932. DOI : 10 . 1109 / MRA . 2006 . 1678144.
- [3] T. LINDBERG, « Feature detection with automatic scale selection », *International journal of computer vision*, t. 30, n° 2, p. 79–116, 1998.
- [4] A. BAUMBERG, « Reliable feature matching across widely separated views », in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, t. 1, 2000, 774–781 vol.1. DOI : 10.1109/CVPR.2000.855899.
- [5] F. POMERLEAU, F. COLAS et R. SIEGWART, « A review of point cloud registration algorithms for mobile robotics », *Foundations and Trends® in Robotics*, t. 4, n° 1, p. 1–104, 27 mai 2015, ISSN : 1935-8253, 1935-8261. DOI : 10 . 1561 / 23000000035. adresse : <http://ftp.nowpublishers.com/article/Details/ROB-035> (visité le 07/06/2017).
- [6] J. ZHANG et S. SINGH, « Visual-lidar odometry and mapping : low-drift, robust, and fast », in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, mai 2015, p. 2174–2181. DOI : 10.1109/ICRA.2015.7139486.
- [7] (). The KITTI vision benchmark suite, adresse : http://www.cvlibs.net/datasets/kitti/eval_odometry.php (visité le 09/11/2017).
- [8] D. NISTER, O. NARODITSKY et J. BERGEN, « Visual odometry », in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, t. 1, juin 2004, I–652–I–659 Vol.1. DOI : 10 . 1109 / CVPR . 2004 . 1315094.
- [9] M. S. ARULAMPALAM, S. MASKELL, N. GORDON et T. CLAPP, « A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking », *IEEE Transactions on Signal Processing*, t. 50, n° 2, p. 174–188, fév. 2002, ISSN : 1053-587X. DOI : 10.1109/78.978374.
- [10] M. KAESSE, A. RANGANATHAN et F. DELLAERT, « iSAM : incremental smoothing and mapping », *IEEE Transactions on Robotics*, t. 24, n° 6, p. 1365–1378, déc. 2008, ISSN : 1552-3098. DOI : 10 . 1109 / TR0 . 2008 . 2006706.
- [11] M. KAESSE, H. JOHANSSON, R. ROBERTS, V. ILA, J. J. LEONARD et F. DELLAERT, « iSAM2 : incremental smoothing and mapping using the bayes

- tree », *The International Journal of Robotics Research*, t. 31, n° 2, p. 216–235, fév. 2012, ISSN : 0278-3649, 1741-3176. DOI : 10.1177/0278364911430419. adresse : <http://journals.sagepub.com/doi/10.1177/0278364911430419> (visité le 11/07/2017).
- [12] A. CENSI, « An accurate closed-form estimate of ICP's covariance », in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, avr. 2007, p. 3167–3172. DOI : 10.1109/ROBOT.2007.363961.
- [13] S. M. PRAKHYA, L. BINGBING, Y. RUI et W. LIN, « A closed-form estimate of 3d ICP covariance », in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, mai 2015, p. 526–529. DOI : 10.1109/MVA.2015.7153246.
- [14] P. ROBERTSON, M. ANGERMANN et B. KRACH, « Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors », in *Proceedings of the 11th international conference on Ubiquitous computing*, ACM, 2009, p. 93–96.
- [15] R. MUR-ARTAL, J. M. M. MONTIEL et J. D. TARDÓS, « ORB-SLAM : a versatile and accurate monocular SLAM system », *IEEE Transactions on Robotics*, t. 31, n° 5, p. 1147–1163, oct. 2015, ISSN : 1552-3098. DOI : 10.1109/TR0.2015.2463671.
- [16] J. ENGEL, T. SCHÖPS et D. CREMERS, « LSD-SLAM : large-scale direct monocular SLAM », in *European Conference on Computer Vision*, Springer, 2014, p. 834–849. adresse : http://link.springer.com/chapter/10.1007/978-3-319-10605-2_54 (visité le 02/05/2017).
- [17] J. ENGEL, J. STÜCKLER et D. CREMERS, « Large-scale direct SLAM with stereo cameras », in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, p. 1935–1942. adresse : <http://ieeexplore.ieee.org/abstract/document/7353631/> (visité le 02/05/2017).
- [18] D. CARUSO, J. ENGEL et D. CREMERS, « Large-scale direct slam for omnidirectional cameras », in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, p. 141–148. adresse : <http://ieeexplore.ieee.org/abstract/document/7353366/> (visité le 02/05/2017).