Un nouvel algorithme EM pour le recalage de nuages de points 2D–3D avec association de données probabiliste

Younes BOUTIYARZIST[1,2,4] Jean-Yves TOURNERET[2,3] François VINCENT[1] Philippe SALMON[4]

[1]ISAE-SUPAERO, Université de Toulouse, Toulouse, France

[2] Laboratoire TéSA, Toulouse, France

[3]Université de Toulouse, Institut de Recherche en Informatique de Toulouse (IRIT), Toulouse INP-ENSEEIHT, Toulouse

[4]Collins Aerospace, Blagnac, France

Résumé – Cet article présente un nouvel algorithme EM (Expectation-Maximization) pour le recalage robuste de nuages de points 2D–3D issus d'une caméra et d'une carte de référence. Nous nous intéressons à l'estimation conjointe des paramètres d'intérêt (i.e., orientation et position de la caméra), de la proportion d'observations aberrantes et de la variance du bruit de mesure. L'approche proposée repose sur un modèle statistique intégrant des variables latentes permettant de gérer les associations inconnues entre points 2D, points 3D et observations aberrantes, via un modèle de mélange. Des résultats obtenus à partir de données synthétiques montrent l'intérêt de cette démarche en termes de rapidité de convergence de l'algorithme proposé et de robustesse face aux mesures aberrantes.

Abstract – This paper studies a new Expectation-Maximization (EM) algorithm for robust 2D–3D point cloud registration resulting from a camera and a 3D reference map. We focus on the joint estimation of the parameters of interest (i.e., orientation and position of the camera), the proportion of outliers and the variance of the measurement noise. Results obtained on synthetic data demonstrate the effectiveness of this approach in terms of convergence speed and robustness to outlier measurements.

1 Introduction

Le recalage de nuages de points 2D–3D, également appelé estimation de pose de caméra, consiste à déterminer la position et l'orientation d'une caméra par rapport à un repère de référence à partir de points 2D acquis avec une caméra et de points 3D issus d'un modèle de la scène, généralement obtenu par LiDAR, photogrammétrie ou fusion de capteurs. Cette carte de référence inclut des éléments sémantiques tels que des routes, des bâtiments ou des objets d'intérêt. Ce problème trouve des applications dans divers domaines, notamment la réalité augmentée [1], la reconstruction 3D [2], la localisation en robotique [3], ou encore l'imagerie médicale [4].

En robotique [3], l'estimation de la pose d'une caméra repose généralement sur la mise en correspondance de points caractéristiques 2D extraits d'une image avec des points 3D connus, puis sur la résolution d'un problème de type Perspective-n-Point (PnP). Toutefois, lorsque les conditions météorologiques se dégradent, le nombre de points d'intérêt détectés diminue et les associations deviennent incertaines, ce qui limite l'efficacité des approches classiques. Pour répondre à ces difficultés, le problème a récemment été formulé dans [5] à l'aide d'une matrice décrivant explicitement les associations inconnues entre points 2D et points 3D, associée à un modèle de mélange intégrant une classe regroupant les observations aberrantes. L'estimation de l'orientation et de la position de la caméra est réalisée à l'aide de l'algorithme EM (Expectation-Maximization) présenté dans [5], dans lequel les paramètres du modèle, c'est-à-dire la proportion d'observations aberrantes et la variance du bruit de mesure, sont supposés connus et fixés par l'utilisateur. Cet article propose une modification de l'algorithme présenté dans [5], permettant d'estimer les paramètres du modèle statistique directement au sein de l'étape M de l'algorithme EM.

Cet article est organisé comme suit. Les sections 2 et 3 présentent le modèle statistique utilisé pour le recalage de points 2D-3D et l'algorithme EM associé. La section 4 les performances de cet algorithme de recalage à l'aide de données synthétiques. Les conclusions de cette étude et quelques perspectives sont présentées dans la section 5.

2 Modèle Statistique

Considérons deux nuages de points d'une même scène. Le premier nuage est constitué de n points 2D bruités détectés dans une image à l'aide d'une caméra notés $\mathbf{X}_1 = \{\mathbf{x}_1^i\}_{i=1}^n$, $\mathbf{x}_1^i \in \mathbb{R}^2$. Le second nuage de points contient m points 3D notés $\mathbf{X}_2 = \{\mathbf{x}_2^j\}_{j=1}^m$, $\mathbf{x}_2^j \in \mathbb{R}^3$ obtenus à l'aide de la carte de référence. Lorsque les associations entre ces deux ensembles de points sont connues, la relation entre un point $\boldsymbol{x}_1^i$ et le point 3D correspondant $\boldsymbol{x}_2^j$ peut être modélisée par un modèle de caméra sténopé [6] :

$$\begin{bmatrix} \boldsymbol{x}_1^i \\ 1 \end{bmatrix} = \underbrace{\mathbf{K}\,\frac{\mathbf{R}\boldsymbol{x}_2^j + \mathbf{t}}{\left(\mathbf{R}\boldsymbol{x}_2^j + \mathbf{t}\right)_3}}_{\pi(\boldsymbol{R}\boldsymbol{x}_2^j + \boldsymbol{t})} + \begin{bmatrix} \boldsymbol{e}_{i,j} \\ 0 \end{bmatrix}, \tag{1}$$

où $\mathbf{K}$ désigne la matrice intrinsèque de la caméra définie par :

$$\mathbf{K} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix},$$

avec α_x et α_y les distances focales exprimées en pixels selon les axes x et y, et (x_0, y_0) le point principal. La notation $(.)_3$ in-

dique la troisième composante d'un vecteur. L'orientation et la position de la caméra sont définies par une matrice de rotation $\mathbf{R} \in SO(3)$ (Special Orthogonal Group) et une translation $\mathbf{t} \in \mathbb{R}^3$. L'opérateur π représente la projection induite par la caméra. Le terme d'erreur de mesure $\boldsymbol{e}_{i,j}$ est supposé gaussien, de loi normale $\mathcal{N}(\mathbf{0}_2, \sigma^2 \boldsymbol{I}_2)$. Pour prendre en compte les points aberrants issus des algorithmes de détection, nous supposons qu'ils sont uniformément répartis dans l'image [7] :

$$p(\boldsymbol{x}_1^i|\text{outlier}) = \frac{1}{\Delta}\mathcal{I}_I(\boldsymbol{x}_1^i), \tag{2}$$

où Δ désigne l'aire totale de l'image en pixels carrés, et $\mathcal{I}_I$ est la fonction indicatrice définie sur le domaine de l'image I.

Pour gérer les associations inconnues entre les points 2D et 3D, nous introduisons comme dans [8] une matrice binaire latente $\boldsymbol{A} \in \{0,1\}^{n\times(m+1)}$. Chaque élément binaire $a_{i,j}$ indique si le point 2D $\boldsymbol{x}_1^i$ est associé au point 3D $\boldsymbol{x}_2^j$ ($a_{i,j} = 1$) ou non ($a_{i,j} = 0$). Une colonne supplémentaire de cette matrice permet également d'identifier les points aberrants. Ainsi, chaque ligne de $\boldsymbol{A}$ satisfait la contrainte $\sum_{j=1}^{m+1} a_{i,j} = 1$, garantissant qu'un point 2D est soit associé à un unique point 3D, soit classé comme une observation aberrante. En l'absence d'information a priori sur les correspondances, nous utilisons une loi uniforme pour modéliser les associations entre les points 2D et 3D. La classe des observations aberrantes est pondérée par leur proportion dans les données notée ρ. La loi a priori des associations est alors définie par :

$$P(a_{i,m+1} = 1) = \rho, \quad P(a_{i,j} = 1) = \frac{1-\rho}{m}, \tag{3}$$

où $j = 1, \ldots, m$ avec m désignant le nombre de points 3D et ρ la proportion d'observations aberrantes parmi les points 2D.

Le problème de recalage consiste alors à estimer le vecteur de paramètres $\boldsymbol{\Theta} = \{\boldsymbol{\Phi}, \boldsymbol{\zeta}\}$ regroupant les paramètres de pose de la caméra $\boldsymbol{\Phi} = \{\boldsymbol{R}, \boldsymbol{t}\}$ (rotation et translation), et $\boldsymbol{\zeta} = \{\sigma^2, \rho\}$ correspond aux autres paramètres du modèle à partir des deux nuages de points 2D et 3D.

3 Algorithme EM proposé

La vraisemblance marginale du modèle proposé est :

$$\mathcal{L}(\boldsymbol{X_1}|\boldsymbol{\Theta}, \boldsymbol{X_2}) = \sum_{\boldsymbol{A}\in\boldsymbol{\Psi}} p\left(\boldsymbol{X_1}, \boldsymbol{A} \mid \boldsymbol{\Theta}, \boldsymbol{X_2}\right), \tag{4}$$

où $\boldsymbol{\Psi}$ représente l'ensemble des matrices d'association valides. Le nombre total d'associations possibles étant $(m+1)^n$, une estimation par maximum de vraisemblance (MLE) devient rapidement trop coûteuse. Nous proposons donc d'utiliser un algorithme EM afin d'estimer $\boldsymbol{\Theta}$ en se basant sur la vraisemblance complète suivante :

$$\begin{aligned}\mathcal{L}_c(\boldsymbol{X_1}, \boldsymbol{A}|\boldsymbol{\Theta}, \boldsymbol{X_2}) &= p(\boldsymbol{X_1} \mid \boldsymbol{A}, \boldsymbol{X_2}, \boldsymbol{\Theta}) P(\boldsymbol{A} \mid \boldsymbol{X_2}, \boldsymbol{\Theta}) \\ &= \prod_{i=1}^{n}\prod_{j=1}^{m}\left[p(\boldsymbol{x}_1^i|\boldsymbol{x}_2^j, \boldsymbol{\Theta})P(a_{i,j})\right]^{a_{i,j}} \\ &\times \prod_{i=1}^{n}\left[p(\boldsymbol{x}_1^i|\text{outlier})P(a_{i,m+1})\right]^{a_{i,m+1}}.\end{aligned} \tag{5}$$

L'algorithme EM est une méthode itérative permettant d'estimer les paramètres d'un modèle comportant des variables latentes [9]. L'algorithme alterne entre deux étapes, *expectation* (E) et *maximization* (M), appliquées à chaque itération $t+1$ comme suit :

3.1 Étape E

Cette étape consiste à calculer $Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(t)})$, qui représente l'espérance de la log-vraisemblance complète conditionnellement aux données observées et aux paramètres estimés lors de l'itération précédente $\boldsymbol{\Theta}^{(t)}$:

$$Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(t)}) = \mathbb{E}_{\boldsymbol{A}|\boldsymbol{X_1},\boldsymbol{X_2},\boldsymbol{\Theta}^{(t)}}\left[\log \mathcal{L}_c(\boldsymbol{X_1}, \boldsymbol{A} \mid \boldsymbol{\Theta}, \boldsymbol{X_2})\right]. \tag{6}$$

À partir de (5), la log-vraisemblance s'exprime comme suit :

$$\begin{aligned}\log \mathcal{L}_c(\boldsymbol{X_1}, \boldsymbol{A} \mid \boldsymbol{\Theta}, \boldsymbol{X_2}) &= \sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j} \log p(\boldsymbol{x}_1^i \mid \boldsymbol{x}_2^j, \boldsymbol{\Theta}) \\ &+ \sum_{i=1}^{n} a_{i,m+1} \log p(\boldsymbol{x}_1^i \mid \text{outlier}) \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j} \log\left(\frac{1-\rho}{m}\right) + \sum_{i=1}^{n} a_{i,m+1}\log(\rho).\end{aligned} \tag{7}$$

D'après le théorème de Bayes, la loi conditionnelle de $a_{i,j} \mid \boldsymbol{x}_1^i, \boldsymbol{x}_2^j, \boldsymbol{\Theta}^{(t)}$ s'écrit, pour tout $(i,j) \in \{1,...,n\} \times \{1,...,m\}$:

$$\gamma_{i,j}^{(t)} = P(a_{i,j} = 1 \mid \boldsymbol{x}_1^i, \boldsymbol{x}_2^j, \boldsymbol{\Theta}^{(t)}), \tag{8}$$

avec

$$\gamma_{i,j}^{(t)} = \frac{p(\boldsymbol{x}_1^i \mid \boldsymbol{x}_2^j, \boldsymbol{\Theta}^{(t)})P(a_{i,j}=1)}{\sum_{j=1}^{m} p\left(\boldsymbol{x}_1^i \mid \boldsymbol{x}_2^j, \boldsymbol{\Theta}^{(t)}\right)P(a_{i,j}=1) + \frac{1}{\Delta}P(a_{i,m+1}=1)}. \tag{9}$$

De plus, la probabilité qu'un point soit une observation aberrante est donnée, pour $i = 1, ..., n$, par :

$$\gamma_{i,m+1}^{(t)} = \frac{\frac{1}{\Delta}P(a_{i,m+1}=1)}{\sum_{j=1}^{m} p\left(\boldsymbol{x}_1^i \mid \boldsymbol{x}_2^j, \boldsymbol{\Theta}^{(t)}\right)P(a_{i,j}=1) + \frac{1}{\Delta}P(a_{i,m+1}=1)}. \tag{10}$$

Pour calculer $p(\boldsymbol{x}_1^i \mid \boldsymbol{x}_2^j, \boldsymbol{\Theta}^{(t)})$, un algorithme de *ray-casting* [10] est utilisé afin d'identifier les points visibles de la carte 3D $\boldsymbol{X_2}$ pour une pose caméra donnée $\boldsymbol{\Theta}^{(t)}$. Ces points sont ensuite projetés sur le plan image en appliquant l'équation (1).

3.2 Étape M

L'étape M consiste à maximiser la fonction $Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(t)})$ définie dans (6) par rapport à $\boldsymbol{\Theta}$. Pour $\boldsymbol{\Phi}$, cette maximisation revient à minimiser l'erreur de reprojection pondérée :

$$\arg\min_{\boldsymbol{\Phi}} \overbrace{\sum_{i=1}^{n}\sum_{j=1}^{m} \underbrace{\gamma_{i,j}\|\boldsymbol{x}_1^i - \pi(\boldsymbol{R}^{(t)}\boldsymbol{x}_2^j + \boldsymbol{t}^{(t)})\|^2}_{r_{i,j}^2(\boldsymbol{\Phi}^{(t)})}}^{L_{\boldsymbol{\Phi}}(t)}, \tag{11}$$

où π représente l'opération de projection définie dans (1), $L_{\mathbf{\Phi}}$ est la fonction de coût liée à la pose de la caméra $\mathbf{\Phi}$, et $r_{i,j}$ désigne le résidu associé à la correspondance (i, j) qui est inclus dans le vecteur de résidus $\boldsymbol{r} = [r_{1,1}, ..., r_{n,m}]$. Le problème (11) étant non linéaire, nous utilisons une variante de l'algorithme EM, appelée *gradient* EM, dont les propriétés de convergence ont été largement étudiées [11]. Cette approche remplace l'étape M classique par une unique itération de descente de gradient.

L'algorithme de Gauss-Newton est utilisé pour mettre à jour la rotation $\boldsymbol{R}$ paramétrisée à l'aide de la formule de Rodrigues [12] et le vecteur de translation $\boldsymbol{t}$ selon l'équation suivante :

$$\mathbf{\Phi}^{(t+1)} = \mathbf{\Phi}^{(t)} + \Delta\mathbf{\Phi}, \tag{12}$$

avec

$$\Delta\mathbf{\Phi} = -(\boldsymbol{J}^\top \boldsymbol{J})^{-1} \boldsymbol{J}^\top \boldsymbol{r}, \tag{13}$$

où $\boldsymbol{J}$ est la matrice Jacobienne des résidus de reprojection par rapport aux paramètres $\boldsymbol{R}$ et $\boldsymbol{t}$. Pour initialiser la descente de gradient (12), nous définissons la pose initiale $\mathbf{\Phi}^{(0)}$ à l'aide d'un autre système de localisation tel qu'une prédiction par filtre de Kalman intégrant des capteurs additionnels (e.g., GPS, centrale inertielle). En l'absence d'information externe, une initialisation peut être obtenue via des techniques d'alignement géométrique, comme des solveurs PnP exploitant un sous-ensemble de correspondances fiables [13].

3.3 Estimation des paramètres du modèle

En plus de l'estimation de la pose $\mathbf{\Phi}$, la mise à jour des paramètres du modèle σ^2 et ρ est essentielle pour affiner l'estimation de la vraisemblance. Cependant, l'optimisation simultanée de la pose et des paramètres du modèle avec l'algorithme Gauss-Newton alourdit l'étape M en augmentant le coût calculatoire. Pour surmonter cette difficulté, nous proposons d'utiliser un algorithme ECM (*Expectation Conditional Maximization*) [14], dans lequel la pose $\mathbf{\Phi}$ est fixée à sa valeur estimée à l'itération précédente, permettant ainsi une mise à jour séquentielle des paramètres σ^2 et ρ. Plus précisément, l'algorithme ECM utilise les mises à jour suivantes :

Mise à jour de σ^2

Le paramètre σ^2 est mis à jour en fixant $\mathbf{\Phi}$ et ρ et en maximisant la fonction Q :

$$\sigma^{2,(t+1)} = \arg\max_{\sigma^2} Q(\sigma^2 \mid \rho^{(t)}, \mathbf{\Phi}^{(t)}). \tag{14}$$

En utilisant (6), (7), (9) et (11), ce problème se réduit à :

$$\arg\min_{\sigma^2} \overbrace{\frac{L_{\mathbf{\Phi}}(t)}{\sigma^{2,(t)}} + \log \sigma^{2,(t)} \sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{i,j}}^{L_{\sigma^2}(t)}, \tag{15}$$

qui admet la solution explicite :

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{i,j} \|\boldsymbol{x}_1^i - \pi(\boldsymbol{R}^{(t)}\boldsymbol{x}_2^j + \boldsymbol{t}^{(t)})\|^2}{\sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{i,j}}. \tag{16}$$

Mise à jour de ρ

De même, ρ est mis à jour en maximisant :

$$\rho^{(t+1)} = \arg\max_{\rho} Q(\rho \mid \sigma^{2,(t+1)}, \mathbf{\Phi}^{(t)}). \tag{17}$$

Les équations (6), (7) et (9) conduisent au problème :

$$\arg\min_{\rho} \overbrace{\log(1-\rho) \sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{i,j} + \log(\rho) \sum_{i=1}^{n} \gamma_{i,m+1}}^{L_\rho(t)}. \tag{18}$$

La solution explicite de l'équation (18) est :

$$\hat{\rho} = \frac{\sum_{i=1}^{n} \gamma_{i,m+1}}{n}. \tag{19}$$

Les valeurs initiales de σ^2 et ρ peuvent être fixées à partir d'informations a priori sur la variance du bruit et la quantité d'éléments aberrants attendus pour ces nuages de points. Une autre manière de procéder est de choisir une valeur élevée de $\sigma^{2,(0)}$ (permettant d'assurer une exploration suffisante de la loi d'intérêt) et une valeur faible de $\rho^{(0)}$ afin d'éviter qu'un grand nombre de données soient détectées comme aberrantes.

4 Résultats expérimentaux

Des expériences ont été menées afin d'évaluer les performances de l'algorithme EM proposé. Cette section commence par décrire le scénario de simulation utilisé, avant d'analyser les résultats obtenus.

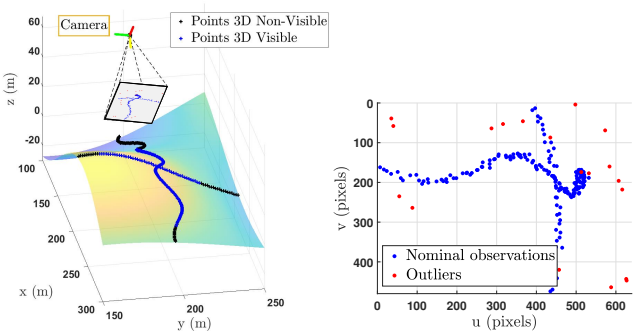
4.1 Scénario de simulation

Le scénario présenté est un carrefour visible par une caméra installée sur un drone, illustré dans la Fig. 1a. La caméra est située à la position $[120m, 200m, 60m]$ selon les axes x, y et z. Son orientation est définie par les angles d'Euler $[0°, -60°, -170°]$. La Fig. 1a présente la carte où les routes sont représentées sous forme d'un nuage de points 3D. Un algorithme de *ray-casting* a été utilisé pour déterminer les points visibles et non visibles par la caméra (indiqués respectivement par des croix bleues et noires). Le nuage de points 2D est obtenu en utilisant l'équation de projection de la caméra (1) avec un bruit de variance σ_v^2 et le modèle de données aberrantes (2) avec une proportion ρ_v. Les points 2D de la caméra et les données aberrantes sont représentées respectivement en bleu et rouge dans la Fig. 1b.

4.2 Analyse de performance

Cette section compare les performances des algorithmes EM avec et sans estimation des paramètres du modèle, respectivement notés ECM et EM, à l'aide de simulations de Monte Carlo. L'évaluation est réalisée sur un jeu de données contenant $n = 200$ observations 2D nominales. La pose initiale des deux algorithmes est fixée à $\boldsymbol{t} = [125, 195, 65]$ avec des orientations $\boldsymbol{\phi} = [2°, -58°, -168°]$. Le nombre d'itérations maximal des algorithmes EM est fixé à $n_{\max} = 50$.

Une première expérience analyse la vitesse de convergence des algorithmes pour des données avec un bruit de variance $\sigma_v^2 = 25$ et sans données aberrantes ($\rho_v = 0\%$). La valeur de la variance σ^2 pour l'algorithme EM et l'initialisation de



(a) Scène 3D (b) Image 2D

FIGURE 1 : Nuage de points 3D du scénario étudié, présenté dans la partie 4 (gauche), et nuage de points 2D distinguant les valeurs aberrantes en rouge et non aberrantes en bleu (droite).

l'algorithme ECM est fixée à $\sigma^2 = 25$. La figure 2 montre que l'algorithme ECM converge plus rapidement que l'algorithme EM avec $\sigma^2 = 25$. Une seconde expérience compare les algorithmes ECM et EM en fonction de ρ_v. Pour EM et l'initialisation d'ECM, σ^2 est fixé à 25 et ρ à 10%. La figure 3 montre que plus la valeur de ρ utilisée pour EM est proche de ρ_v, plus la différence de performance est faible. À l'inverse, un mauvais choix de ρ augmente l'écart de MSE entre les algorithmes ECM et EM. Ces résultats illustrent l'importance d'estimer conjointement le paramètre ρ avec les autres paramètres du modèle.

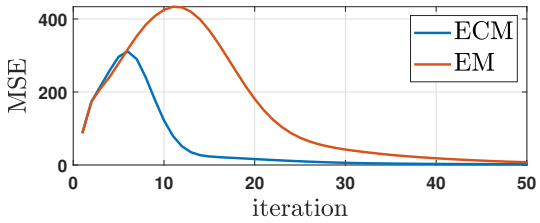


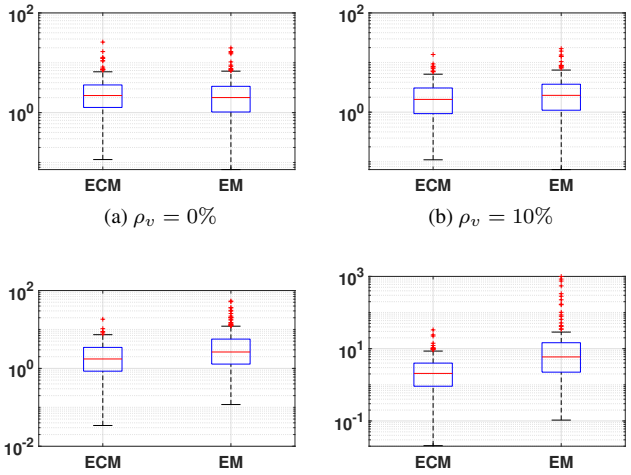
FIGURE 2 : MSE de la position et de l'orientation $\mathbf{\Phi}$ au cours des itérations.

5 Conclusion et perspectives

Nous avons proposé une extension de l'algorithme EM proposé dans [5] pour le recalage 2D–3D, intégrant l'estimation de la proportion de données aberrantes et la variance du bruit de mesure. L'utilisation du cadre ECM permet une mise à jour séquentielle efficace de ces paramètres, améliorant la robustesse aux points aberrants et accélérant la convergence de l'algorithme EM, comme le montrent nos expériences. Les perspectives incluent une meilleure gestion des données aberrantes, l'étude de la sensibilité de l'algorithme aux conditions initiales, ainsi que et des validations sur données réelles pour des applications en navigation autonome.

Références

[1] E. Marchand *et al.*, "Pose estimation for augmented reality : a hands-on survey," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 12, pp. 2633–2651, 2015.



(a) $\rho_v = 0\%$ (b) $\rho_v = 10\%$

(c) $\rho_v = 30\%$ (d) $\rho_v = 50\%$

FIGURE 3 : Boxplots des MSE du vecteur (position, orientation) $\mathbf{\Phi}$ pour différentes valeurs de la proportion de données aberrantes ρ_v.

[2] J. L. Schonberger *et al.*, "Structure-From-Motion Revisited," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, (Las Vegas, USA), June 2016.

[3] C. Campos *et al.*, "ORB-SLAM3 : An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

[4] J. Song *et al.*, "DynaWeightPnP : Toward global real-time 3D-2D solver in PnP without correspondences," *arXiv preprint arXiv :2409.18457*, 2024.

[5] Y. Boutiyarzist *et al.*, "A New EM algorithm for 2D-3D Point cloud registration with Probabilistic Data Association," in *IEEE Stat. Signal Process. Workshop (SSP)*, (Edinbourg, UK), 2025. submitted.

[6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, UK : Cambridge university press, 2000.

[7] J. Lesouple *et al.*, "Robust Hypersphere Fitting from Noisy Data Using an EM Algorithm," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, (Dublin, Ireland), pp. 1–5, 2021.

[8] G. Agamennoni *et al.*, "Point Clouds Registration with Probabilistic Data Association," in *Proc. Int. Conf. Intell. Robots Syst. (IROS)*, (Daejeon, South Korea), 2016.

[9] A. P. Dempster *et al.*, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Roy. Stat. Soc. Ser. B (Methodol.)*, vol. 39, no. 1, pp. 1–38, 1977.

[10] S. D. Roth, "Ray casting for modeling solids," *Comput. Graph. and Image Process.*, vol. 18, no. 2, pp. 109–144, 1982.

[11] W. Xu *et al.*, "Toward Global Convergence of Gradient EM for Over-Parameterized Gaussian Mixture Models," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, (Vancouver, Canada), 2024.

[12] J. S. Dai, "Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections," *Mech. Mach. Theory*, vol. 92, pp. 144–152, 2015.

[13] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP : An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, Feb. 2009.

[14] X.-l. Meng *et al.*, "Maximum likelihood estimation via the ECM algorithm : A general framework," *Biometrika*, vol. 80, pp. 267–278, 06 1993.