

# Recalage basé plans de nuages de points pour la navigation dans des environnements structurés

Ketty Favre, Muriel Pressigout, Eric Marchand, Luce Morin

▶ **To cite this version:**

Ketty Favre, Muriel Pressigout, Eric Marchand, Luce Morin. Recalage basé plans de nuages de points pour la navigation dans des environnements structurés. ORASIS 2021 - 18ème édition des journées francophones des jeunes chercheurs en vision par ordinateur, Centre National de la Recherche Scientifique [CNRS], Sep 2021, Saint Ferréol, France. pp.1-8. hal-03339657

**HAL Id: hal-03339657**
**https://hal.science/hal-03339657**

Submitted on 9 Sep 2021

**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.

# Recalage basé plans de nuages de points pour la navigation dans des environnements structurés

Ketty Favre[1] Muriel Pressigout[2] Eric Marchand[3] Luce Morin[2]

[1] Univ Rennes, CNRS, IETR - UMR 6164, Rennes, France.
[2] Univ Rennes, INSA Rennes, CNRS, IETR - UMR 6164, Rennes, France.
[3] Univ Rennes, Inria, CNRS, IRISA, Rennes, France.

ketty.favre@univ-rennes1.fr

## Résumé

*Les algorithmes traditionnels de recalage de nuages de points 3D, basés sur l'algorithme d'Iterative Closest Point (ICP), reposent sur la mise en correspondance de points. Dans les environnements bien structurés, tels que les bâtiments, les plans peuvent être segmentés et utilisés pour le recalage, de manière similaire à l'approche classique d'ICP basée sur les points. L'utilisation de plans réduit considérablement le nombre d'entrées de l'ICP. Cet article présente un algorithme de recalage précis et robuste basé sur les plans. La transformation optimale est estimée par une approche de minimisation en deux étapes, en effectuant successivement une minimisation robuste plan à plan et un recalage robuste non linéaire point à plan. Les expériences sur le jeu de données de référence Autonomous Systems Lab (ASL) montrent que la méthode proposée permet de recaler avec succès 100% des scans des trois séquences d'intérieur. Les expériences montrent également que la méthode proposée est robuste dans les scénarios de mouvements importants et plus précise que les autres algorithmes de l'état de l'art évalués. En outre, un nouveau jeu de données, LOOP'IN, est fourni. Il est composé de deux boucles dans des scènes d'intérieur d'environnements réels, avec un grand nombre de scans capturés avec un LiDAR 3D. Les tests menés sur le jeu de données LOOP'IN montrent que l'algorithme est capable de recaler de longues séquences, de fermer des boucles et de construire une carte incrémentale de l'environnement exploré.*

## Mots Clef

Recalage, localisation, nuage de points

## Abstract

*Traditional 3D point clouds registration algorithms, based on ICP, rely on point matching of large point clouds. In well-structured environments, such as buildings, planes can be segmented and used for registration, similarly to the classical point-based ICP approach. Using planes tremendously reduces the number of inputs. In this article, a robust and accurate plane-based registration algorithm is presented. The optimal transformation is estimated through a two-step minimization approach, successively performing robust plane-to-plane minimization and non-linear robust point-to-plane registration. Experiments on the ASL benchmark dataset show that the proposed method enables to successfully register 100% of the scans from the three indoor sequences. Experiments also show that the proposed method is robust in large motion scenarios and more accurate than other state-of-the-art algorithms. Moreover, a new challenging dataset, LOOP'IN, is provided. It is composed of two loops in real-world indoor scenes, with a large number of scans captured with a 3D LiDAR. Tests led on LOOP'IN dataset shows that the algorithm is able to register long sequences, to close loops and to build an incremental map of the explored environment.*

## Keywords

Registration, localization, point cloud

# 1 Introduction

Les progrès de la technologie LiDAR ne sont plus à démontrer. Les capteurs LiDAR permettent d'obtenir des données 3D du milieu environnant, représentées sous forme de nuages de points. Ces nuages de points sont utilisés dans de nombreux types d'applications, plus particulièrement la localisation et la reconstruction de scènes. Dans les domaines de la vision par ordinateur et de la robotique, le recalage est un problème largement étudié, mais qui reste avec de nombreux défis à relever.

L'une des approches les plus populaires pour recaler des nuages de points 3D est l'algorithme Iterative Closest Point (ICP) [1]. Il permet de calculer la transformation rigide (rotation et translation) qui relie un nuage de points 3D source et un nuage de points 3D cible. Pour ce faire, chaque point du nuage source est apparié avec son plus proche voisin dans le nuage cible. Ensuite, la transformation rigide 3D qui minimise la distance entre les points appariés est estimée. Cette opération est réalisée selon un schéma itératif jusqu'à ce que l'erreur résiduelle atteigne le seuil souhaité. Le recalage de type ICP peut être divisé en deux catégo-

ries : le recalage des caractéristiques locales et celui des caractéristiques globales. Les premières approches de l'algorithme ICP utilisaient principalement les caractéristiques locales. [1] utilise la distance point-à-point et [2] la distance point-à-plan, qui s'avère être plus robuste et converge plus rapidement que la distance point-à-point. Une résolution linéaire de la minimisation de la distance point-à-plan peut être trouvée dans [3] en utilisant l'approximation des petits angles. Une résolution non linéaire de Levenberg-Marquardt utilisant des estimateurs robustes est proposée dans [4]. Les méthodes de recalage local peuvent également extraire des points clés des nuages de points pour établir des correspondances. Dans [5], Normal Distribution Transform (NDT) prend en compte les structures de surface locales autour de chaque point. Dans l'approche d'ICP généralisé (G-ICP) proposé dans [6], le voisinage local des points est utilisé afin d'assimiler cette structure à de petits patchs planaires. De même que pour les approches point-à-plan, les normales locales du nuage de points cible sont prises en compte ainsi que celles du nuage de points source. Ces méthodes offrent des performances satisfaisantes mais prennent généralement beaucoup de temps lorsqu'il s'agit de recaler de grands nuages de points.

Une façon de surmonter le problème du nombre de points est d'utiliser des caractéristiques globales, telles que les plans. Dans ces approches, la première étape consiste à segmenter les nuages de points en patchs planaires [7] [8] [9], puis à utiliser les caractéristiques planaires pour établir les correspondances. Pour faire correspondre les patchs planaires, on peut trouver plusieurs approches. Dans le cas de [10], le débit d'acquisition des données est supposé élevé, ce qui entraîne un faible mouvement relatif d'un scan à l'autre. Ainsi, pour déterminer les correspondances de plans, les plans proches les uns des autres et presque parallèles sont appariés. Cette approche donne de bons résultats dans ce type de scénario, mais peine à estimer une trajectoire correcte en dehors de faibles déplacement entre deux scans. Dans [8], un descripteur plan/ligne est proposé pour établir les correspondances de structure. Dans [9], les formes, les surfaces et les normales des patchs planaires sont considérées pour trouver les meilleures correspondances.

L'algorithme proposé dans cet article, Plane-based Accurate Registration ICP (PAR-ICP), se situe dans les catégories des caractéristiques globales et locales. Un recalage de caractéristiques à deux niveaux est effectué, d'abord en utilisant des plans (grâce à une mise en correspondance basée sur un apprentissage), puis en utilisant des points.

Les principales contributions de cet article sont :

- un algorithme effectuant un recalage précis basé sur des plans dans des jeux de données difficiles et capable de reconstruire l'environnement via une carte incrémentale ;
- Une méthode de minimisation en deux étapes avec deux niveaux de recalage des caractéristiques, robuste aux mouvements importants ou à une initialisation imprécise ;
- Une mise en correspondance des plans effectuée par classification en utilisant des Random Forest ;
- une évaluation et une comparaison de la méthode proposée avec les algorithmes de l'état de l'art sur un jeu de données difficile ;
- un nouveau jeu de données, LOOP'IN, contenant de longues séquences et des boucles, pour la communauté utilisant les nuages de points.

Dans les sections suivantes, la structure de la méthode proposée est d'abord décrite. Puis, les définitions et notations utilisées dans cet article sont détaillées. Ensuite, l'appariement de plans à l'aide de la classification apprise et la minimisation de la distance sont présentés dans des sections dédiées. Le jeu de données fourni est décrit succinctement. Enfin, les expériences et leurs résultats sont présentés.

# 2 Vue d'ensemble de l'algorithme proposé

Cette section décrit la structure de l'algorithme proposé, nommé PAR-ICP. Il est présenté dans la Figure 1. De la même manière que l'algorithme classique ICP, la méthode effectue de manière itérative l'étape de mise en correspondance et l'étape de minimisation. Cependant, dans la méthode proposée, on effectue d'abord la mise en correspondance des plans et la minimisation de la distance plan-à-plan, puis la mise en correspondance des points et la minimisation de la distance point-à-plan. Cette dernière étape vise à raffiner l'estimation. Un exemple de recalage à l'aide de la méthode proposée est donné à la Figure 2.

Chaque étape de la structure de l'algorithme est décrite plus loin dans cet article :

- l'extraction des plans et la mise en correspondance des plans à l'aide de la classification sont décrites dans la section 4 ;
- le recalage plan-à-plan robuste est décrit dans la section 5.1 ;
- la minimisation supplémentaire point-à-plan conduisant à un recalage plus fin est détaillée dans la section 5.2.

# 3 Définitions et notations des distances de l'algorithme ICP

Dans les sections suivantes, les points source et cible seront respectivement désignés par ${}^{s}\mathbf{P} = \{{}^{s}\mathbf{p}_i\}_{i=1}^{N}$ et ${}^{t}\mathbf{P} = \{{}^{t}\mathbf{p}_j\}_{j=1}^{M}$. Dans ce problème de recalage, le nuage de points cible est fixe. Le but est de trouver la transformation rigide ${}^{t}\mathbf{T}_s$ qui recale le mieux la source à la cible. Cette transformation est définie comme suit :

$$ {}^{t}\mathbf{T}_s = \begin{bmatrix} {}^{t}\mathbf{R}_s & {}^{t}\mathbf{t}_s \\ \mathbf{0}_{3\times1} & 1 \end{bmatrix} \tag{1} $$

avec ${}^{t}\mathbf{R}_s$ et ${}^{t}\mathbf{t}_s$ respectivement une matrice de rotation $3\times3$ et un vecteur de translation $3\times1$.

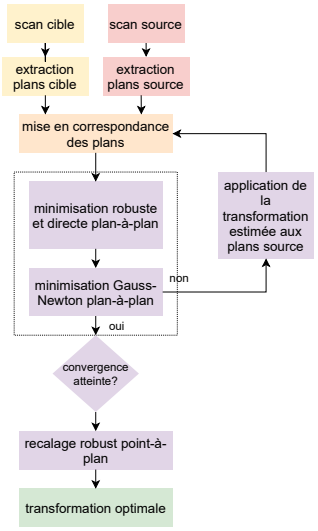



FIGURE 1 – Structure de l'algorithme PAR-ICP.

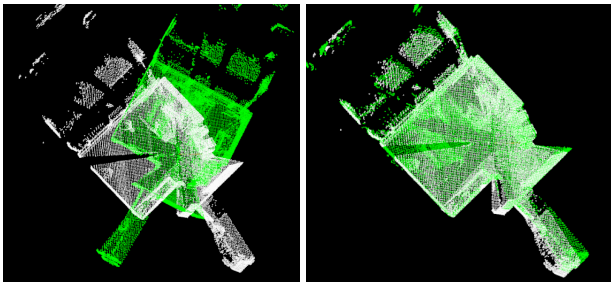


FIGURE 2 – Exemple de recalage entre deux nuages de points (scans 8 et 9 de la séquence *Apartment* du jeu de données ASL [11]). La rotation entre les nuages de points est importante (plus de 70° sur l'axe de lacet), mais la méthode proposée parvient à recaler les deux nuages de points avec précision. En blanc, le nuage de points cible - En vert, le nuage de points source. *A gauche :* avant recalage - *A droite :* après recalage.

Pour le recalage basé sur les points, chaque point ${}^s\mathbf{p}_i$ de la source est mis en correspondance avec son point le plus proche ${}^t\mathbf{p}_i$ de la cible. Ensuite, la transformation rigide minimisant une métrique de distance est estimée et ces deux étapes sont itérées jusqu'à ce qu'un seuil soit atteint. Dans la méthode originale ICP [1], la métrique de distance à minimiser est la distance euclidienne point-à-point. Comme les points correspondants d'un scan à l'autre peuvent ne pas être exactement identiques mais se trouver sur la même surface, il est préférable de choisir, comme dans [2], de minimiser la distance point-à-plan (Fig. 3), définie par :

$$d_i^{\perp} = \| {}^t\mathbf{n}_i^{\top} \cdot ({}^t\mathbf{T}_s \, {}^s\mathbf{p}_i - {}^t\mathbf{p}_i) \|^2 \qquad (2)$$

avec ${}^t\mathbf{n}_i$ la normale à la surface calculée à partir du voisinage de ${}^t\mathbf{p}_i$.
En ce qui concerne les plans, un plan $\Pi(\rho, \mathbf{n})$ est défini par l'équation $\mathbf{n}^{\top}\mathbf{p} = \rho$, où $\rho$ est la distance du plan $\Pi$ à

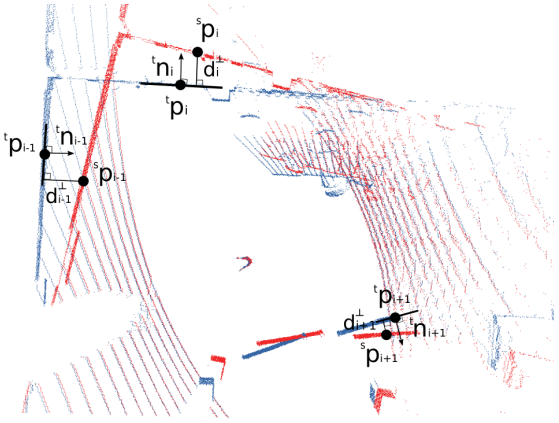



FIGURE 3 – Distance point à plan $d_i^{\perp}$ comme décrit dans [3]. En bleu, le nuage de points cible et les normales de surface liées à ses points. En rouge, le nuage de points source à recaler.

l'origine du capteur. La distance entre deux plans correspondants ${}^s\mathbf{\Pi}_i({}^s\rho_i, {}^s\mathbf{n}_i)$ le plan source et ${}^t\mathbf{\Pi}_i({}^t\rho_i, {}^t\mathbf{n}_i)$ le plan cible, est donnée comme suit [10] [12] :

$$\mathbf{d}_i^{\Pi} = \begin{pmatrix} {}^t\mathbf{R}_s \, {}^s\mathbf{n}_i - {}^t\mathbf{n}_i \\ [{}^t\mathbf{R}_s \, {}^s\mathbf{n}_i]^{\top} \, {}^t\mathbf{t}_s + {}^s\rho_i - {}^t\rho_i \end{pmatrix} \qquad (3)$$

où ${}^s\mathbf{n}_i$ et ${}^t\mathbf{n}_i$ sont les normales à ${}^s\mathbf{\Pi}_i$ et ${}^t\mathbf{\Pi}_i$ respectivement et ${}^s\rho_i$ et ${}^t\rho_i$ leur distance respective à l'origine du capteur dans le repère cible.

# 4 Mise en correspondance des plans par apprentissage

Pour réduire la complexité de la phase de mise en correspondance, et donc le nombre de plans correspondants potentiels, un classifieur Random Forest est entraîné pour trouver les meilleures paires de plans source/cible. Les structures planaires sont extraites du nuage de points à l'aide d'un algorithme de segmentation par croissance de région basé sur [14], grâce à la bibliothèque C++ Point Cloud Library (PCL) [15]. Dans cette approche, les points d'un voisinage ayant des normales proches sont considérés comme étant sur la même surface lisse et sont rassemblés dans un cluster. Chaque cluster représente un plan. Les normales des points sont estimées en utilisant un estimateur robuste basé sur des Principal Component Analysis (PCA) itératifs sur le voisinage des points concernés [16]. Pour chaque plan extrait ${}^s\mathbf{\Pi}_i$ dans la source, une liste de plans dans la cible, qui sont des correspondances potentielles pour le plan source, est construite. On peut noter que dans la littérature on trouve des algorithmes de mise en correspondances se basant sur des Random Forest, nottament dans [13] où des segments sont appariés.

## 4.1 Principe du Random Forest

L'algorithme de classification Random Forest [17] est un algorithme d'apprentissage basé sur la notion d'ensemble,

utilisant des arbres de décisions. Il prédit la classe d'une instance cible à partir des valeurs de plusieurs variables d'entrée, appelées caractéristiques. Un classificateur Random Forest rassemble les prédictions de plusieurs arbres de décision en un seul modèle pour prendre une décision. L'ensemble d'apprentissage est composé de $n$ instances. Chaque instance est représentée comme un vecteur de $m$ caractéristiques $f_1, \ldots, f_m$ et une étiquette de classe. Tout d'abord, $k$ échantillons sont construits à partir de sous-ensembles choisis aléatoirement dans l'ensemble d'apprentissage original en utilisant le principe de bootstrapping. Ensuite, $k$ arbres de décision sont construits sur la base des $k$ échantillons. Une fois le modèle estimé, il est utilisé dans la phase de classification pour les nouvelles instances. Les $k$ arbres donnent une prédiction. La classe avec le plus grand nombre de votes gagne. En combinant plusieurs prédictions d'arbres de décision décorrélés, le modèle est moins sujet au surapprentissage qu'un seul arbre de décision.

### 4.2 Données d'entrainement et augmentation de données

Pour entraîner le classifieur, un ensemble de données étiquetées est nécessaire. Des paires de plans sont extraites des séquences intérieures du jeu de données ASL [11] et étiquetées manuellement. Un bruit gaussien est ajouté aux nuages de points afin d'utiliser des données qui ne sont pas exactement les mêmes que les données originales. Comme cet base de données donne la vérité terrain (c'est à dire le déplacement entre deux captures) pour chaque scan, il est possible de faire correspondre facilement les plans d'un scan à l'autre. Les classes choisies sont étiquetées *Match* si les plans sont des correspondances potentielles et *Not-Match* s'ils ne le sont pas.
Cependant, le jeu de données n'est pas composé de nombreux scans. Afin d'entraîner le modèle avec un plus grand ensemble, une augmentation des données est effectuée sur les données précédemment étiquetées, en appliquant des rotations sur un des nuages de point de la paire à recaler, avec un angle choisi aléatoirement dans $[-70°; +70°]$.

### 4.3 Sélection des caractéristiques

Le classificateur reçoit les caractéristiques suivantes en entrée :

— La distance $d_o$ entre les projections de l'origine sur le plan source et le plan cible :

$$d_o = \| {}^s\rho_i \, {}^s\mathbf{n}_i - {}^t\rho_j \, {}^t\mathbf{n}_j \|^2 \tag{4}$$

Les plans sont censés être proches les uns des autres, donc $d_o$ devrait être faible.

— La distance $d_c$ entre les centroïdes des plans source et cible :

$$d_c = \| {}^s\mathbf{c}_i - {}^t\mathbf{c}_j \|^2 \tag{5}$$

avec ${}^s\mathbf{c}$ et ${}^t\mathbf{c}$ les centroïdes de ${}^s\mathbf{\Pi}_i$ et ${}^t\mathbf{\Pi}_j$ respectivement. Une fois de plus, cette caractéristique vise à souligner la proximité des plans, de sorte que $d_c$ devrait être faible.

— Le rapport de surface entre les patchs planaires $S_r$ :

$$S_r = \frac{min({}^sS_i, {}^tS_j)}{max({}^sS_i, {}^tS_j)} \tag{6}$$

avec ${}^sS_i$ et ${}^tS_j$ la surface des patchs planaires source et cible respectivement. Les patchs planaires étant censés avoir des formes similaires, on s'attend à ce que $S_r$ soit proche de 1.

— Le produit scalaire des normales des plans $\phi_n$ :

$$\phi_n = {}^s\mathbf{n}_i \cdot {}^t\mathbf{n}_j \tag{7}$$

Les plans sont censés être presque parallèles (cette caractéristique suppose une faible rotation entre deux balayages), donc $\phi_n$ devrait être proche de 1.

### 4.4 Résultats de la mise en correspondance par classification

Le classifieur est évalué par sa précision, définie comme le pourcentage de correspondances valides par rapport à la vérité terrain. Afin d'éviter d'avoir des données de test dans l'ensemble d'entraînement, un processus de validation croisée à trois volets a été utilisé : deux séquences ont été utilisées pour l'entraînement tandis que la troisième a été conservée pour le test. La précision pour chaque séquence testée est présentée dans le tableau 1. Comme on peut le voir, le classifieur produit quelques fausses correspondances, mais leur occurrence est inférieure à 15% sur les données testées. Un tel pourcentage de données aberrantes est gérable pour une méthode d'estimation de pose robuste, comme nous le verrons dans la section suivante.

## 5 Recalage

### 5.1 Recalage plan-à-plan

Maintenant que la mise en correspondance des plans est établie, la minimisation de la distance plan-à-plan, qui estime la transformation rigide reliant les plans source et cible, peut être calculée.

**Méthode de minimisation directe de la distance plan-à-plan.**
Le développement de cette méthode est similaire à celui présenté dans [12]. Pour tenir compte de la présence de

TABLE 1 – Résultats de la validation croisée du processus de mise en correspondance

| k | Apartment | ETH | Stairs | Accuracy (%) |
|---|---|---|---|---|
| 1 | *testing* | training | training | 87.9 |
| 2 | training | *testing* | training | 89.3 |
| 3 | training | training | *testing* | 91.6 |

fausses correspondances dans la liste des plans correspondants, un algorithme robuste RANSAC est utilisé.

**Méthode de minimisation de la distance plan-à-plan par une approche Gauss-Newton.**
Afin d'affiner la solution obtenue par l'approche directe, les correspondances de plans identifiées comme des valeurs aberrantes par l'algorithme RANSAC sont données en entrée de l'approche itérative non linéaire de Gauss-Newton. Cette méthode nécessite une représentation minimale de la transformation à estimer ${}^t\mathbf{T}_s$. Une telle représentation est définie par un vecteur à 6 dimensions noté $\mathbf{q} = ({}^t\mathbf{t}_s, \theta\mathbf{u})^\top$ où $\theta$ et $\mathbf{u}$ sont l'angle et l'axe de la rotation ${}^t\mathbf{R}_s$.
L'erreur plan-à-plan doit être minimisée de telle sorte que :

$$\widehat{\mathbf{q}} = \underset{\mathbf{q}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{d}_i^{\Pi}\|^2 \tag{8}$$

où $\mathbf{d}_i^{\Pi}$ est défini par (3). L'équation (8) peut être résolue à l'aide de l'algorithme de Gauss-Newton. Sa résolution consiste à minimiser la fonction de coût $E(\mathbf{q}) = \|\mathbf{e}(\mathbf{q})\|$ où :

$$\mathbf{e}(\mathbf{q}) = \begin{pmatrix} \mathbf{n}(\mathbf{q}) - \mathbf{n} \\ \rho(\mathbf{q}) - \rho \end{pmatrix} \tag{9}$$

avec $\mathbf{n}(\mathbf{q}) = (..., {}^t\mathbf{R}_s {}^s\mathbf{n}_i, ...)$, $\mathbf{n} = (..., {}^t\mathbf{n}_i, ...)$, $\rho(\mathbf{q}) = (..., [{}^t\mathbf{R}_s {}^s\mathbf{n}_i]^\top {}^t\mathbf{t}_s + {}^s\rho_i, ...)$ et $\rho = {}^t\rho_i$ le vecteur d'erreur de la distance entre le nuage de points cible et le nuage de points source transformé avec la transformation estimée précédente.
Ce problème de minimisation peut être résolu par une approche itérative aux moindres carrés qui donne :

$$\delta\mathbf{q} = -\lambda \mathbf{J}(\mathbf{q})^+ \mathbf{e}(\mathbf{q}) \tag{10}$$

où $\lambda$ est un scalaire dans $]0, 1]$ et $\mathbf{J}(\mathbf{q})^+$ est la pseudo-inverse du Jacobien $\mathbf{J}(\mathbf{q})$ qui est donné dans [18]. La pose est ensuite mise à jour à chaque itération en utilisant la carte exponentielle.

## 5.2 Recalage point-à-plan

Pour assurer un recalage encore plus précis, une étape plus fine est ajoutée pour trouver la meilleure transformation rigide attendue, basée sur les données originales des nuages de points, plutôt que sur les plans qui en sont extraits. Pour ce faire, un recalage point-à-plan est ajouté à la fin du processus.
Chaque point ${}^s\mathbf{p}_i$ de ${}^s\mathbf{P}$ est mis en correspondance avec son point le plus proche ${}^t\mathbf{p}_i$ dans ${}^t\mathbf{P}$ selon la distance euclidienne. Ensuite, la transformation rigide qui recale le nuage de points source et cible est calculée en minimisant la distance point-à-plan (Eq. 2). Sa résolution consiste à minimiser la fonction de coût $E(\mathbf{q}) = \|\mathbf{e}(\mathbf{q})\|$ où $\mathbf{e}(\mathbf{q}) = (d_1^\perp, \ldots, d_n^\perp)^T$.
Le principe est le même que dans la section 5.1, avec :

$$\widehat{\mathbf{q}} = \underset{\mathbf{q}}{\operatorname{argmin}} \sum_{i=1}^{N} \|d_i^\perp\|^2 \tag{11}$$

où $d_i^\perp$ est donné par l'équation (2).
La minimisation est faite comme dans le paragraphe précédent grâce à une méthode de Gauss Newton (le Jacobien est donné dans [18]). Néanmoins, si des valeurs aberrantes sont présentes dans le jeu de données, l'estimation de la minimisation devient instable. Il est donc nécessaire de considérer un processus de minimisation robuste [19]. La méthode de Gauss-Newton permet d'utiliser les M-estimateurs dans une approche itérative des moindres carrés repondérés. L'erreur à minimiser est définie par $\mathbf{e}_\rho(\mathbf{q}) = \mathbf{D}\,\mathbf{e}(\mathbf{q})$. où $\mathbf{D}$ est une matrice diagonale contenant des poids qui reflètent la confiance dans les données (voir par exemple [19] pour plus de détails).

# 6 Jeux de données d'évaluation

Pour évaluer la robustesse et la précision de l'algorithme proposé, deux jeux de données sont utilisés. Le premier est le jeu de données ASL [11] et le second, le jeu de données LOOP'IN, est fourni avec cet article.

## 6.1 Jeu de données : Autonomous System Lab

Comme cet article se concentre sur l'utilisation des caractéristiques planaires, seules les séquences intérieures du jeu de données ASL [11] sont utilisées. L'un des principaux avantages de ce jeu de données est que chaque séquence est accompagnée des poses vérité terrain mesurées pour chaque scan avec une précision millimétrique.

- *Apartment* : cette séquence est conçue pour évaluer la robustesse de l'algorithme aux valeurs aberrantes provenant d'éléments dynamiques (par exemple, des meubles déplacés).
- *ETH* : cette séquence vise à évaluer la robustesse de recalage vis à vis d'éléments répétitifs.
- *Stairs* : cette séquence vise à évaluer la robustesse aux variations rapides des volumes scannés.

## 6.2 Jeu de donnée : LOOP'IN

Pour évaluer PAR-ICP sur de longues séquences et des fermetures de boucles, un jeu de données, LOOP'IN [1], a été créé. Il est composé de données d'intérieur d'environnement réel, chaque scan étant constitué de 25K à 30K points. L'acquisition des données a été réalisée à l'aide d'un LiDAR VLP-16 Puck Hi-Res. Le LiDAR est monté sur une plateforme mobile poussée par un opérateur. Comme les captures sont parfois effectuées dans des zones étroites, le LiDAR est incliné vers le sol, afin de s'assurer de capturer les points du sol. Le jeu de données proposé est composé de deux séquences d'intérieur :

- *Balcony Loop* : La séquence est composée d'une boucle avec un croisement de la trajectoire à la fin. La boucle se fait autour d'un balcon entourant une pièce presque carrée. La séquence est composée de 142 scans.

1. LOOP'IN dataset est disponible en ligne sur https://github.com/kfavre/LOOP-IN_dataset.

— *Coffee Hall Loop* : La séquence est composée d'une boucle commençant et se terminant au même endroit, dans une grande cafétéria. Des meubles sont présents ainsi que des personnes en mouvement. D'un côté de la salle se trouve une grande baie vitrée (qui fait apparaître des reflets sur les scans). La séquence est composée de 171 scans.

# 7 Résultats expérimentaux

Les expériences sont menées en deux étapes. Tout d'abord, PAR-ICP est comparé aux algorithmes de l'état de l'art sur le jeu de données de ASL, référence en ce qui concerne la robustesse et la précision. Ensuite, la capacité de la méthode proposée à recaler de longues séquences aussi précisément que possible est évaluée à l'aide du jeu de données LOOP'IN présenté.

## 7.1 Validation sur le jeu de données ASL

Pour évaluer la précision et la robustesse de la méthode proposée, des expériences sont réalisées sur les séquences intérieures ASL présentées précédemment. Le jeu de données est accompagné d'une vérité terrain pour la pose de chaque nuage de points, ce qui permet d'utiliser des métriques pour évaluer la précision de la pose estimée.

**Métriques.** Comme dans [20], la précision des algorithmes est évaluée par la distance euclidienne $\Delta_t$ (respectivement la distance géodésique $\Delta_r$) entre la transformation estimée et la vérité terrain pour la translation (respectivement pour la rotation) :

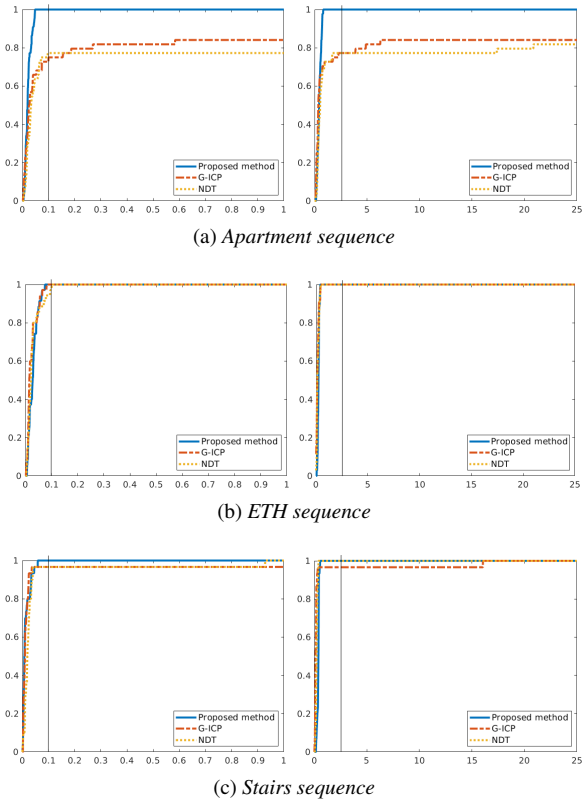
$$\Delta_t = \| {}^t\hat{\mathbf{t}}_s - {}^t\mathbf{t}_s^* \| \tag{12}$$

$$\Delta_r = arccos\left( \frac{trace({}^t\mathbf{R}_s^{*-1} {}^t\hat{\mathbf{R}}_s) - 1}{2} \right) \tag{13}$$

avec ${}^t\hat{\mathbf{t}}_s$ et ${}^t\hat{\mathbf{R}}_s$ la translation et la rotation estimées, ${}^t\mathbf{t}_s^*$ et ${}^t\mathbf{R}_s^*$ la translation et la rotation de la réalité terrain, respectivement.
Les seuils pour estimer un recalage réussi sont choisis comme suggéré dans [5], c'est-à-dire que le seuil de translation est de 0,1m et le seuil de rotation est de 2,5°. Notez que les erreurs de rotation et de translation sont présentées séparément par souci de clarté, mais un résultat n'est valable que si les erreurs de rotation et de translation sont toutes deux inférieures à leur seuil respectif.
Les nuages de points sont recalés scan à scan.

**Evaluation et comparaison avec des algorithmes de l'état de l'art.** Tout d'abord, PAR-ICP est comparé à deux algorithmes de recalage de l'état de l'art :

— G-ICP [6] a trois paramètres principaux. Le nombre maximal d'itérations est fixé à 10, le "Euclidean fitness epsilon" est fixé à $10^{-6}$ et la distance maximale de correspondance à 0,8m.
— NDT, est réglé avec les étapes recommandées dans [5]. Le "transformation epsilon" est fixé à $10^{-3}$, la taille du pas à 0,1, le nombre d'itérations maximales à 5, la résolution du premier pas 1,0m, la résolution du deuxième pas 2,0m, la résolution du troisième pas 1,0m et le dernier pas 0,5m.



(a) *Apartment sequence*

(b) *ETH sequence*

(c) *Stairs sequence*

FIGURE 4 – Probabilités cumulées des erreurs de translation et de rotation pour chaque séquence sur PAR-ICP, G-ICP [6] et NDT [5]. *A gauche :* l'erreur de translation (en mètres) sur l'axe horizontal. La barre verticale représente le seuil (0,1 m) pour un recalage réussi. *A droite :* l'erreur de rotation (en degrés) sur l'axe horizontal. La barre verticale représente le seuil (2,5°) de réussite de recalage.

Notez que dans ce contexte, pour PAR-ICP, la validation croisée est utilisée pour le classifieur Random Forest établissant les correspondances, afin d'éviter d'utiliser des données qui ont déjà été vues dans la validation. Lorsque l'algorithme est testé sur *Apartment*, il a été entraîné uniquement sur *ETH* et *Stairs*, et ainsi de suite pour les trois séquences. La précision de PAR-ICP, G-ICP et NDT sur la translation et la rotation est résumée dans la Figure 4 avec des courbes représentant les probabilités cumulées d'erreur sur la translation et la rotation. Le comportement attendu des courbes est d'atteindre 1 (ce qui signifie que tous les scans de la séquence sont recalés correctement) avant que le seuil d'erreur ne soit atteint. Comme prévu, PAR-ICP est très performant dans un environnement intérieur contenant de fortes caractéristiques planes. Sur chaque séquence, il réussit à recaler 100% des scans selon les seuils choisis. La différence majeure est visible sur la séquence *Apartment*. Alors que PAR-ICP réussit à recaler tous les scans, G-ICP

TABLE 2 – Pourcentage (%) de recalages réussis (translation et rotation combinées) pour les algorithmes évalués sur chaque séquence calculée dans [9]

| Sequence | PAR-ICP | FGR | GH-ICP | PGR |
|---|---|---|---|---|
| *Apartment* | **100** | **100** | 52 | **100** |
| *ETH* | **100** | 63 | 93 | 90 |
| *Stairs* | **100** | 3 | 94 | 97 |

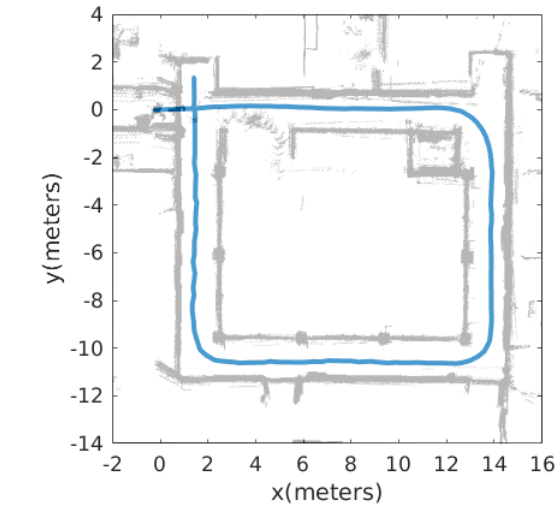
n'atteint que 75% de recalages réussis et 77% pour NDT. Cela est dû à la présence de grandes rotations dans la séquence, prouvant la robustesse de PAR-ICP à ce type de mouvement.

Dans [9], plusieurs algorithmes sont évalués en suivant la même procédure : FGR [7], GH-ICP [21] et l'algorithme proposé dans [9] noté ici PGR (Planar Global Registration). Les résultats de ces expériences sont présentés dans le tableau 2. Pour plus de détails sur les paramètres des algorithmes, veuillez vous référer à [9].

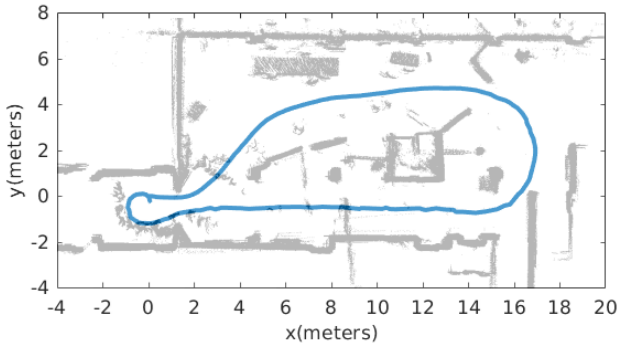
Sur *Apartment*, FGR, PGR et PAR-ICP réussissent tous à recaler 100% des scans. Les erreurs moyennes de rotation et de translation sont respectivement de $0,45°$ et $0,022m$ pour PAR-ICP, de $1,126°$ et $0,028m$ pour FGR, et de $0,303°$ et $0,009m$ pour PGR. PGR s'avère être le plus précis dans ce scénario. La raison pour laquelle GH-ICP n'atteint qu'un taux de réussite de 52% est probablement due à la présence de grandes rotations dans cette séquence. Sur *ETH* et *Stairs*, GH-ICP et PGR donnent tous deux des résultats satisfaisants (plus de 90% pour chaque algorithme sur les deux séquences), mais restent moins précis que PAR-ICP.

### 7.2 Recalage sur le jeu de données LOOP'IN et construction de cartes incrémentales avec notre approche

Dans l'expérience suivante, la trajectoire du capteur des séquences LOOP'IN est calculée en utilisant PAR-ICP. Dans ce contexte, toutes les séquences ASL ont été utilisées pour entraîner le classifieur Random Forest établissant des correspondances. Pendant le recalage de chaque nuage de points, une carte dite incrémentale est construite et mise à jour. Le recalage d'un nouveau scan source ne se fait pas scan à scan mais en recalant le scan sur la carte incrémentale. Cela permet de faire correspondre les plans de la source avec les plans qui ont été rencontrés précédemment dans la séquence, ce qui donne des informations importantes dans une longue séquence, et encore plus si une boucle est effectuée. Les plans de la source, qui sont déjà connus et identifiés comme des correspondances dans les premières étapes, sont mis à jour dans la carte, et de nouveaux plans inconnus sont ajoutés, dans l'attente d'une correspondance ultérieure dans le processus de reconstruction. La trajectoire sur les axes x et y pour chaque séquence est présentée dans la Figure 5. Dans les deux séquences, PAR-ICP est capable de fournir une trajectoire satisfaisante, et de fermer la boucle entre les poses de départ et d'arrivée, lorsqu'il utilise la carte incrémentale pour le recalage. Certaines imprécisions dans la trajectoire peuvent être observées de temps en temps, mais elles n'empêchent pas l'algorithme de fermer la boucle. Ainsi, lorsque de fortes caractéristiques planaires sont présentes, la méthode proposée donne de bons résultats.



(a)



(b)

FIGURE 5 – Trajectoire dans le plan xy calculée grâce à l'algorithme PAR-ICP pour chaque séquence. Les axes sont en mètres. L'environnement reconstruit est superposé à la main sur la trajectoire. (a) Séquence *Balcony loop*. (b) Séquence *Coffee Hall*.

## 8 Conclusion

Cet article présente un algorithme de recalage précis basé sur les plans dénommé PAR-ICP. Une approche de mise en correspondance des plans utilisant un classificateur Random Forest (RF) entraîné sur les caractéristiques des plans est introduite. La minimisation robuste plan-à-plan, suivie d'une minimisation point-à-plan, permet d'atteindre 100% de recalage réussi sur les séquences d'intérieur du jeu de données ASL.

Les expériences montrent que PAR-ICP est plus précis que les autres algorithmes de l'état de l'art dans les environnements bien structurés. Elles démontrent également qu'il est plus robuste aux mouvements importants.
Un nouveau jeu de données, LOOP'IN, est également fourni pour permettre à la communauté d'évaluer et de comparer les algorithmes de recalage sur des données réelles.
En effectuant la fermeture de boucles dans de longues séquences et en créant des cartes incrémentales, l'algorithme proposé donne de bons résultats pour les applications de localisation et de cartographie.
Une étape ultérieure consistera à intégrer PAR-ICP dans un processus de SLAM.

# Références


[1] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, Feb. 1992.

[2] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, pp. 145–155, Apr. 1992.

[3] K.-l. Low, "Linear least-squares optimization for point-toplane ICP surface registration," tech. rep., Department of Computer Science, University of North Carolina at Chapel Hill, 2004.

[4] A. Fitzgibbon, "Robust Registration of 2D and 3D Point Sets," *Image and Vision Computing*, vol. 21, Jan. 2003.

[5] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, "Beyond points : Evaluating recent 3D scan-matching algorithms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3631–3637, May 2015.

[6] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Proc. of Robotics : Science and Systems*, vol. 2, p. 4, 2009.

[7] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Computer Vision – ECCV 2016*, vol. 9906, 10 2016.

[8] S. Chen, L. Nan, R. Xia, J. Zhao, and P. Wonka, "PLADE : A Plane-Based Descriptor for Point Cloud Registration With Small Overlap," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, pp. 2530–2540, Apr. 2020.

[9] W. Zong, M. Li, Y. Zhou, L. Wang, F. Xiang, and G. Li, "A Fast and Accurate Planar-Feature-Based Global Scan Registration Method," *IEEE Sensors Journal*, vol. 19, pp. 12333–12345, Dec. 2019.

[10] W. S. Grant, R. C. Voorhies, and L. Itti, "Efficient Velodyne SLAM with point and plane features," *Autonomous Robots*, vol. 43, pp. 1207–1224, June 2019.

[11] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, "Challenging data sets for point cloud registration algorithms," *The International Journal of Robotics Research*, vol. 31, pp. 1705–1711, Dec. 2012.

[12] Y. Taguchi, Y. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *2013 IEEE International Conference on Robotics and Automation*, pp. 5182–5189, May 2013.

[13] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch : Segment based place recognition in 3d point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5266–5272, IEEE, 2017.

[14] T. Rabbani, F. A. van den Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraints," in *ISPRS 2006 : Proceedings of the ISPRS commission V symposium Vol. 35, part 6 : image engineering and vision metrology, Dresden, Germany 25-27 September 2006*, pp. 248–253, 2006.

[15] R. B. Rusu and S. Cousins, "3D is here : Point Cloud Library (PCL)," in *2011 IEEE International Conference on Robotics and Automation*, (Shanghai, China), pp. 1–4, May 2011.

[16] J. Sanchez, F. Denis, D. Coeurjolly, F. Dupont, L. Trassoudaine, and P. Checchin, "Robust normal vector estimation in 3D point clouds through iterative principal component analysis," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 163, pp. 18–35, 2020.

[17] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001.

[18] K. Favre, M. Pressigout, L. Morin, and E. Marchand, "A plane-based approach for indoor point clouds registration," in *IAPR Int. Conf on Pattern Recognition, ICPR'20*, (Milano (virtual), Italy), January 2021.

[19] E. Malis and E. Marchand, "Experiments with robust estimation techniques in real-time robot vision," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 223 – 228, 2006.

[20] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets Open-source library and experimental protocol," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[21] Z. Dong, B. Yang, Y. Liu, F. Liang, B. Li, and Y. Zang, "A novel binary shape context for 3d local surface description," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 130, pp. 431–452, 2017.