



HAL
open science

Recalage de nuages de points 3D issus d'acquisitions LiDAR de scènes structurées fondé sur des a priori géométriques

Julia Sanchez, Paul Checchin, F. Denis, Florent Dupont, L. Trassoudaine

► To cite this version:

Julia Sanchez, Paul Checchin, F. Denis, Florent Dupont, L. Trassoudaine. Recalage de nuages de points 3D issus d'acquisitions LiDAR de scènes structurées fondé sur des a priori géométriques. COM-
presseion et REprésentation des Signaux Audiovisuels, CORESA 2017, Nov 2017, Caen, France. hal-
02095671

HAL Id: hal-02095671

<https://hal.science/hal-02095671v1>

Submitted on 23 Apr 2019

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 de nuages de points 3D issus d'acquisitions LiDAR de scènes structurées fondé sur des a priori géométriques

J. Sanchez¹, P. Checchin², F. Denis¹, F. Dupont¹, L. Trassoudaine²

¹Univ. Lyon-LIRIS UMR 5205 CNRS

Université Claude Bernard Lyon 1

43, bd du 11 novembre 1918

F. 69622 VILLEURBANNE CEDEX

julia.sanchez@univ-lyon1.fr, paul.checchin@uca.fr, florence.denis@liris.cnrs.fr,

florent.dupont@liris.cnrs.fr, laurent.trassoudaine@univ-bpclermont.fr

²Institut Pascal UMR 6602 CNRS

Campus universitaire des Cézeaux

4 Avenue Blaise Pascal

F. 63178 AUBIERES CEDEX

Résumé

L'utilisation de capteurs LiDAR pour obtenir des données 3D implique l'acquisition de scans suivant différents points de vue. Dans les systèmes actuels, l'algorithme d'ICP (Iterative Closest Point) est largement utilisé pour recalibrer les scans entre eux. Cependant, cette méthode se heurte à des problèmes de minima locaux et ne fonctionne que pour de faibles mouvements. Cet article développe une nouvelle méthode de recalage adaptée aux environnements structurés et basée sur des caractéristiques géométriques. La rotation et la translation de la transformation totale recherchée sont calculées de manière successive en utilisant respectivement l'image Gaussienne des nuages de points et une corrélation d'histogrammes. L'évaluation de notre algorithme sur deux ensembles de scans 3D comparé à six méthodes existantes montre que la méthode proposée est plus robuste à de faibles résolutions de scans, à la complexité de la scène et au bruit du capteur. De plus la faible durée de notre algorithme permet une implémentation temps réel du recalage.

Mots clefs

LiDAR; Recalage; Image gaussienne; Nuages de points.

1 Introduction

Les acquisitions 3D sont utilisées dans de nombreux domaines industriels tels que l'inspection de bâtiments [1], ou la navigation autonome [2]. Les données sont collectées sous forme de nuages de points mesurés dans le repère capteur dans des poses différentes. Le recalage est le processus ayant pour but d'estimer la transformation rigide qui aligne un nuage appelé source sur un nuage appelé cible. Cela permet de lier les scans afin de ne former qu'un seul nuage de points qui pourra être analysé par la suite. La phase d'acquisition doit être rapide, nécessiter peu de mémoire et consommer un minimum d'énergie afin que le système puisse être embarqué. De nombreux capteurs sont apparus sur le marché récemment pour répondre à ces critères [2],

mais les algorithmes de recalage limitent encore les performances du système complet. En environnement intérieur, ce qui constitue le contexte majeur de cet article, les algorithmes existants présentent un manque de précision et un taux de réussite du recalage trop bas. La plupart sont très coûteux en temps et nécessitent une qualité de données généralement élevée.

L'algorithme le plus utilisé dans les chaînes de traitement commercialisées est ICP (Iterative Closest Point) [3]. Cet algorithme attribue itérativement des correspondances entre les points de la source et les points de la cible, filtre ces correspondances et minimise la distance entre les points de chaque paire créée. Le processus itératif de l'algorithme rend l'implémentation en temps réel compliquée à mettre en place. De plus, la fonction de coût n'étant pas convexe, la minimisation peut mener à un minimum local. Pour répondre à ce problème, une solution est d'assurer un mouvement faible entre les scans en entrée de l'algorithme, soit en adaptant le processus d'acquisition, soit en cherchant un alignement approximatif initial avant le traitement par ICP. Cependant, aucune méthode proposée jusqu'à aujourd'hui ne permet de résoudre des problèmes de recalage quel que soit le capteur, le chevauchement des nuages de points et leur position initiale. Dans cet article, nous cherchons à développer une nouvelle méthode pour réaliser un recalage sans utiliser de correspondances locales ni d'alignement initial et qui pourrait être utilisé sur des nuages présentant un faible chevauchement et des motifs répétitifs. La section 2 décrit les méthodes connexes au travail réalisé dans cet article. La section 3 développe la méthode proposée. Enfin, l'algorithme est évalué sur deux jeux de données dans la section 4.

2 Etat de l'art

La première méthode proposée pour recalibrer des nuages de points était fondée sur la projection sur la sphère gaussienne [4]. Cette projection correspond à la représentation des normales comme des points sur une sphère de

rayon unité. L'idée principale est d'échantillonner respectivement l'image gaussienne et l'espace des rotations et d'évaluer la similarité des sphères après transformation par chacune des rotations. Cette méthode reste très approximative, la difficulté majeure résidant dans l'échantillonnage de l'espace des rotations qui doit être le plus uniforme possible et couvrir un maximum de rotations. Les méthodes EGI (Extended Gaussian Sphere) et CEGI (Complex Extended Gaussian Sphere) [5] permettent de retrouver successivement la rotation comme décrit précédemment et la translation entre nuages de points en ajoutant une information de distance des plans à l'origine. Cependant, ces méthodes ne peuvent pas prendre en compte un chevauchement partiel et ne peuvent donc pas être utilisées dans le contexte de notre travail.

L'algorithme ICP a rapidement surpassé les méthodes précédentes. Par la suite, de nombreux travaux ont eu pour but d'améliorer la robustesse de cet algorithme. La plupart d'entre eux sont présentés dans [6]. Certaines méthodes se concentrent sur la métrique de la fonction de coût. En effet, les correspondances trouvées entre points ne peuvent être exactes à cause de la différence des échantillonnages et du bruit introduit par les capteurs. Les variantes ICP *point à plan* [7] et ICP *généralisé* [8] ont été proposées pour remédier à ce problème. D'autres méthodes améliorent le processus de minimisation pour le rendre plus robuste aux minima locaux et pour accélérer l'optimisation. Dans [9] par exemple, l'algorithme de *Levenberg-Marquardt* est utilisé pour trouver l'optimum de la fonction. Récemment, l'algorithme Go-ICP [10] a été introduit pour régler le problème de non convexité. Il permet une alternance entre une phase ICP et un processus fondé sur une approche *branch and bound* pour réaliser une recherche dans l'espace des transformations. Il est prouvé que cet algorithme converge vers la solution optimale [10]. Cependant, ce dernier peut se révéler très coûteux en temps de calcul à cause des recherches du plus proche voisin répétées, impliquées par ICP et de la recherche dans l'espace des transformations. En pratique, cet algorithme reste difficile à utiliser sur des données réelles de scènes d'intérieur de plusieurs dizaines de milliers de points.

Holz et al. ont créé une chaîne de traitement en intégrant ICP afin d'assurer la convergence de l'algorithme [11]. Premièrement, un alignement approximatif est déterminé, puis, ICP (ou une de ses variantes) est exécuté pour ajuster le résultat. Pour trouver un alignement approximatif, des points clés sont tout d'abord sélectionnés [12], puis, leur descripteur local est calculé grâce à leur voisinage. Les descripteurs locaux les plus utilisés sont FPFH [13], SPIN [14], et SHOT [15]. Dans l'étape suivante, les points de la source et de la cible sont mis en correspondance par similarité de leur descripteur. Enfin, un alignement optimal est déduit. L'erreur principale vient des fausses correspondances attribuées dans ce processus. Différentes techniques ont été implémentées pour gérer ces incohérences. Certaines permettent de rejeter des mau-

vaises associations avec un *pipeline* de différents critères [11]. Zhou et al. sont allés plus loin avec la méthode FGR (Fast Global Registration) [16] en introduisant une nouvelle fonction de coût qui permet d'attribuer des poids aux correspondances de manière itérative pour leur donner plus ou moins d'importance dans la minimisation suivant leur cohérence. Cette méthode est un recalage global qui ne nécessite pas d'étape d'ajustement. Sa rapidité est remarquable et la majorité des résultats atteint une bonne précision. Néanmoins, toutes les méthodes fondées sur des descripteurs nécessitent que la scène étudiée ait des détails permettant de différencier les points localement. Elles sont moins adaptées à des scènes d'intérieur dans lesquelles beaucoup de points ont le même type de voisinage, notamment sur des plans. Dans certains cas, la convergence peut ne pas être atteinte si les correspondances ont un trop fort taux d'erreur.

Un autre moyen d'obtenir un recalage initial est d'utiliser RANSAC (RANdom SAMple Consensus). La méthode a tout d'abord été introduite dans [17]. Le principe est d'attribuer des correspondances aléatoires entre des ensembles de trois points provenant de chacun des nuages. Une transformation est déduite pour chaque correspondance et une évaluation permet de déterminer la meilleure. Super4PCS [18] a été proposé pour accélérer le traitement et le rendre plus robuste en utilisant des ensembles de quatre points et en leur appliquant différents filtres. La méthode a une convergence optimale. Cependant, le temps de calcul peut se révéler très important lorsque des données de plusieurs milliers de points sont traitées car le nombre de correspondances à tester pour obtenir une précision suffisante devient très élevé. Cette méthode ne peut donc pas être utilisée en pratique dans le contexte de notre travail.

Par ailleurs, les plans présents dans un environnement intérieur ont déjà été utilisés pour réaliser un recalage dans [19]. Cette méthode est inspirée de la NDT (Normal Distribution Transform) [20] qui est utilisée pour mettre en correspondance des densités locales de points entre les nuages. Des correspondances sont attribuées entre les plans extraits de la cible et ceux extraits de la source sur des critères géométriques. Puis la probabilité de localisation des points sur des plans après transformation est maximisée. Cette méthode établit des correspondances entre plans et non entre points ce qui se révèle plus efficace. Cependant, elle repose sur l'algorithme qui extrait les plans des nuages de points qui peut s'avérer très long dans un environnement complexe.

Au vue des difficultés rencontrées pour assurer un bon fonctionnement des méthodes citées précédemment, nous souhaitons introduire une nouvelle méthode inspirée de considérations géométriques, au processus non itératif et suffisamment rapide pour pouvoir envisager une implémentation temps réel. La méthode proposée, dénommée Structured Scene Features based Registration (SSFR) dans cet article, est décrite dans la prochaine section.

3 Méthode proposée

Cette méthode de recalage est adaptée à la reconstruction d'environnements structurés et est basée sur des caractéristiques géométriques de ce type de scène. Les trois murs principaux non parallèles du nuage cible doivent avoir des équivalents dans le nuage source. Cet algorithme permet d'obtenir successivement la rotation et la translation de la transformation recherchée en deux processus distincts.

3.1 Recherche de la rotation

La première étape consiste à projeter les deux nuages de points sur la sphère Gaussienne i.e. , on représente les normales en chaque point comme un point sur la sphère de rayon unité. Un exemple de projection est donné Figure 1. Dans cette représentation, l'orientation des plans est mise en valeur par des régions plus denses de points. La propriété principale de l'image Gaussienne est que la rotation a le même effet sur la projection que sur le nuage de points initial [5]. On va donc chercher à faire correspondre l'image gaussienne de la source et l'image gaussienne de la cible pour déduire la rotation entre la source et la cible. Pour détecter les clusters de points dans l'image gaussienne, on applique un filtre par densité. On sélectionne les six zones de plus forte densité avec leur voisinage. Puis, l'algorithme du *mean shift* [21] est utilisé avec un noyau d'*Epanechnikov* pour détecter les modes des clusters extraits précédemment. Les modes obtenus sont les normales des plans principaux du nuage et sont nommés \vec{n}_i avec $i = 1, \dots, N$ avec N le nombre de normales.

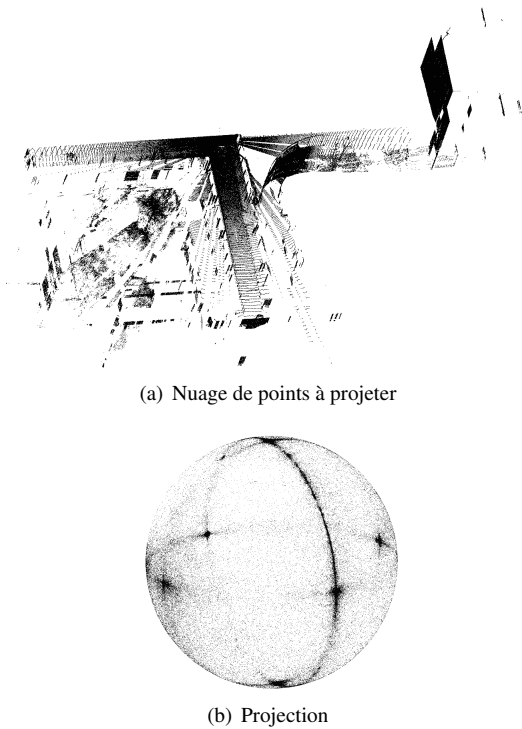


Figure 1 – Exemple de projection sur la sphère gaussienne d'un scan d'intérieur, DS2 scan 1 (cf section 4)

Un exemple de résultat de ce traitement est donné Figure 2.

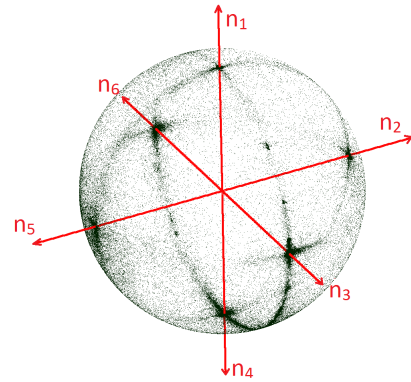


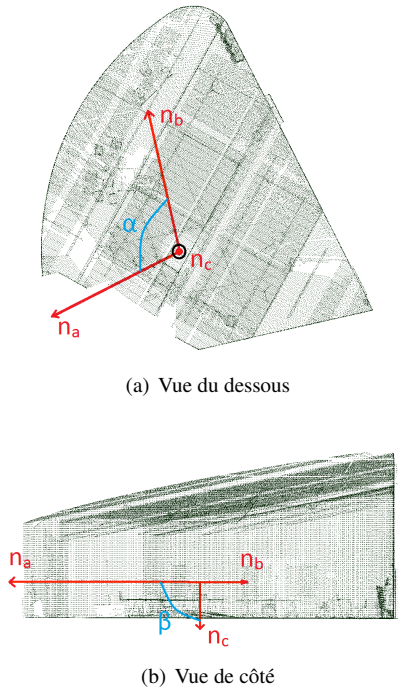
Figure 2 – Normales des plans extraites de l'image gaussienne.

Une fois que les normales des murs principaux sont obtenues, toutes les paires de normales $(\vec{n}_i, \vec{n}_j)_C$, $(\vec{n}_k, \vec{n}_l)_S$ avec $i \neq j$ et $k \neq l$ sont formées respectivement dans la cible (indice C) et dans la source (indice S). Le but est alors de lier une paire extraite de la source à la paire correspondante extraite de la cible. Toutes les correspondances sont d'abord établies puis filtrées. Pour ce faire, on ne garde que les correspondances pour lesquelles les angles $(\vec{n}_i, \vec{n}_j)_C$ et $(\vec{n}_k, \vec{n}_l)_S$ sont similaires. Pour trouver la meilleure combinaison, toutes les correspondances restantes sont testées et, après calcul de la translation, les résultats des différents recalages sont comparés comme expliqué dans la section 3.3.

3.2 Recherche de la translation

Premièrement, trois normales $\vec{n}_a, \vec{n}_b, \vec{n}_c$, sont sélectionnées parmi les modes calculés à l'étape précédente (décrite section 3.1) et définissent des axes de translation a, b, c . Les axes a et b correspondent aux normales \vec{n}_a et \vec{n}_b utilisées pour réaliser la rotation. L'axe c est déduit après avoir aligné toutes les normales principales par la rotation. Il correspond à la normale de la cible la mieux alignée avec une de la source. Un exemple de sélection d'axes est donné Figure 3. Les nuages de points sont alors projetés sur chacun des axes et des histogrammes sont construits. Le maximum de la corrélation entre les histogrammes d'un même axe correspond alors au déplacement à réaliser sur l'axe étudié pour aligner les murs.

Si les murs sont perpendiculaires, les déplacements obtenus peuvent être appliqués directement sur chacun des axes a, b, c . Cependant, si les murs ne sont pas perpendiculaires, les déplacements sont corrélés les uns aux autres. On peut alors définir un nouveau repère d'axes orthogonaux menant à 3 translations indépendantes. Ces axes sont nommés x, y, z , et sont représentés par les vecteurs $\vec{n}_x, \vec{n}_y, \vec{n}_z$. Ils sont définis comme suit : \vec{n}_x et \vec{n}_a sont superposés; \vec{n}_z est le produit vectoriel de \vec{n}_a et \vec{n}_b ; \vec{n}_y est le produit vectoriel de \vec{n}_z et \vec{n}_a .



(a) Vue du dessous

(b) Vue de côté

Figure 3 – Axes sélectionnés par la méthode (en rouge) dans le scan d'une pièce.

Les déplacements sur les axes a , b et c sont nommés Δa , Δb et Δc . Les déplacements à calculer sur les axes x , y et z sont nommés Δx , Δy and Δz . Une illustration 2D d'une configuration avec deux murs est donnée Figure 4. Sur cette figure simplifiée, on ne considère aucun déplacement selon l'axe z . Les axes de translation initiaux sont donc a et b . Les équations (1) et (2) permettent de calculer Δx et Δy avec α l'angle non signé entre les axes x et b . α appartient à l'intervalle $[0, \pi]$.

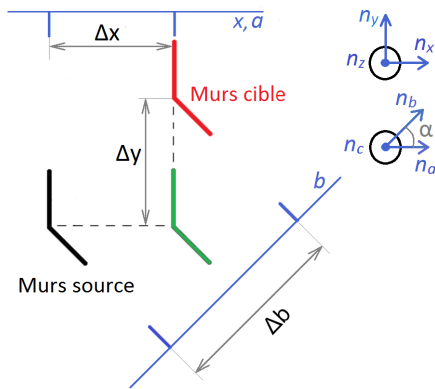


Figure 4 – Représentation schématique vue du dessus de la translation de deux murs depuis leur position dans la source (en noir) jusqu'à leur position dans la cible (en rouge). En vert, les murs sont représentés après translation sur l'axe x .

L'équivalence pour le calcul du déplacement selon l'axe z est donnée par l'équation (3) avec β l'angle non signé entre

les axes x et c et γ l'angle non signé entre y et c . β et γ appartiennent à l'intervalle $[0, \pi]$.

$$\Delta x = \Delta a \quad (1)$$

$$\Delta y = \frac{\Delta b - \Delta x * \cos(\alpha)}{\sin(\alpha)} \quad (2)$$

$$\Delta z = \frac{\frac{\Delta c - \Delta x * \cos(\beta)}{\sin(\beta)} - \Delta y * \cos(\gamma)}{\sin(\gamma)} \quad (3)$$

3.3 Sélection de la transformation

Comme décrit section 3.1, toutes les possibilités de rotation pour aligner les paires de normales sont testées. Les translations correspondantes sont calculées puis les transformations sont comparées pour sélectionner la meilleure. Pour cela, nous utilisons la valeur LCP (Largest Common Pointset) [18] qui quantifie le chevauchement des nuages de points. Cette valeur correspond au pourcentage de points de la source ayant un voisin dans la cible. Le rayon du voisinage est défini comme la résolution du nuage de points. La résolution est ici définie comme la distance moyenne entre les points d'un nuage.

4 Validation

Pour évaluer notre méthode, nous utilisons deux ensembles de données de scans. Le premier est appelé "apartment" et est disponible en ligne sur le site de l'ASL (Autonomous system Lab) [22]. Il est référencé ici comme "DS1". Nous avons constitué le deuxième ensemble, référencé "DS2", à l'Institut Pascal à Clermont-Ferrand afin de pouvoir travailler sur un environnement intérieur complexe avec une vérité terrain. La complexité de ce jeu de données provient 1) de ses motifs tels que des ouvertures, des arbres ou des murs courbes, 2) du nombre de points traités, 3) de sa structure car les scans ont été acquis sur deux étages avec de grands déplacements entre chaque acquisition. Les principales caractéristiques des deux ensembles de données sont présentées dans le tableau 1.

Tableau 1 – Caractéristiques des jeux de données avec le nombre moyen de points par scan, le nombre de scans, la distance maximale entre les extrémités des nuages de points, la résolution moyenne des nuages, la qualification des déplacements entre chaque acquisition de scan.

| | DS1 | DS2 |
|------------------|-----------------|-----------------|
| Capteur | Hokuyo UTM-30LX | Leica P20 |
| Nombre de points | 365 000 | 9×10^6 |
| Nombre de scans | 45 | 6 |
| Taille max (m) | 11 | 70 |
| Résolution (cm) | 0.61 | 0.35 |
| Déplacements | Faible | Fort |

Des transformations issues d'une vérité terrain sont disponibles pour chacune d'elles. Dans DS2, elles proviennent d'un recalage réalisé avec des cibles physiques. La

figure 5 présente un exemple de recalage avec notre méthode pour des nuages de points extraits de DS2.

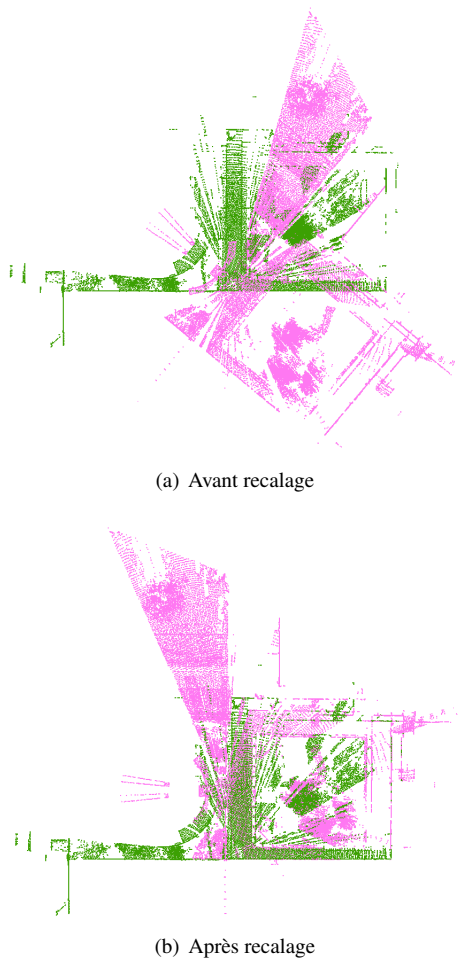


Figure 5 – Exemple de résultat obtenu : recalage du scan 2 (rose) sur le scan 1 (vert) dans DS2

Pour valider la méthode, nous avons sélectionné trois critères : 1) l'erreur RMS (*Root Mean Square*) entre source et cible. Cette erreur est calculée à partir de la vérité terrain et dépend de la résolution des nuages de points étudiés. Les correspondances de référence entre source et cible sont définies grâce à la vérité terrain. Des paires sont formées entre les points de la source (après transformation par la référence) et les points de la cible si ces points sont assez proches. Puis la distance entre les points d'une même paire est évaluée après recalage par chacune des méthodes. 2) le taux de réussite de recalage sur un ensemble de données. On suppose que le recalage est réussi quand l'erreur RMS est inférieure à 10 cm. 3) le temps de calcul de l'algorithme (processeur 8 cœurs Intel Xeon-E5620, 2.4GHz).

Notre algorithme SSFR est comparé aux méthodes existantes sur les deux ensembles de données. Les résultats sont disponibles dans le tableau 2.

Tableau 2 – Evaluation des méthodes sur DS1 et DS2. ICP1 : ICP point-to-point, ICP2 : ICP point-to-plane, NDT : Normal Distribution Transform, FGR : Fast Global Registration, Leica : méthode semi-manuelle fondée sur ICP, SSFR : notre méthode, VT : Vérité Terrain. RMSE : Root Mean Square Error; S : taux de réussite; T : time; nd : pas de données; la croix indique que le taux de réussite n'est pas suffisant.

| | DS1 | | | DS2 | | |
|-------|-----------|-------|-------|-----------|-------|-------|
| | RMSE (cm) | S (%) | T (s) | RMSE (cm) | S (%) | T (s) |
| ICP1 | 2.0 | 54 | 7.4 | x | 0 | x |
| ICP2 | 1.7 | 82 | 3.2 | x | 0 | x |
| NDT | 2.4 | 61 | 5.2 | x | 0 | x |
| FGR | 2.3 | 100 | 3.9 | x | 0 | x |
| Leica | nd | nd | nd | 2.7 | 100 | nd |
| SSFR | 1.8 | 100 | 2.4 | 2.9 | 100 | 24.0 |
| VT | 1.4 | 100 | nd | 2.66 | 100 | nd |

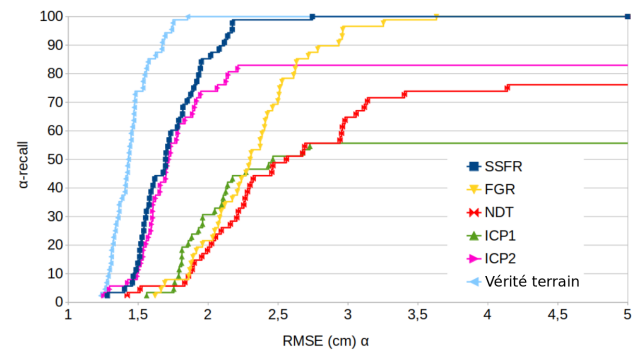


Figure 6 – Evaluation sur DS1. α -recall est le pourcentage de scans pour lesquels une méthode donnée mène à un recalage avec une RMSE $< \alpha$. SSFR est notre méthode. Comparaison avec algorithmes ICP point-to-point (ICP1), ICP point-to-plane (ICP2), Fast Global Registration (FGR) et Normal Distribution Transform (NDT).

Les nuages de points ont été sous-échantillonnés de manière uniforme afin d'obtenir une résolution de, respectivement, 3.6 cm et 6.1 cm, (ce qui limite le nombre de points à quelques dizaines de milliers) et de comparer avec différentes méthodes existantes : ICP point-to-point, ICP point-to-plane, NDT et FGR. Les méthodes Go-ICP et Super4PCS n'ont pas été incluses dans l'évaluation car leur temps de calcul est trop long pour des données aussi volumineuses que des scans de bâtiments. Les algorithmes ICP et NDT ont été testés en utilisant la bibliothèque PCL et FGR, Go-ICP et Super4PCS ont été évalués avec le code disponible en ligne. Chaque scan a été recalé sur son prédécesseur dans l'ordre d'acquisition. Notre méthode est la seule qui fonctionne sur les deux ensembles de données. Les méthodes fondées sur ICP ont un taux de réussite faible dès lors que les déplacements sont trop grands. On re-

marque que sur DS1, l'algorithme proposé dans cet article atteint une précision meilleure que les autres algorithmes testés (1.8 cm) en un temps plus court (2.4 s). La figure 6 détaille les résultats de tous les recalages avec les différents algorithmes étudiés sur DS1 et confirme les bonnes performances de notre algorithme SSFR.

5 Conclusions et perspectives

Nous avons présenté un nouvel algorithme de recalage global de scènes d'intérieur avec chevauchement partiel. Il fonctionne sans recalage initial et peut être utilisé sur des scènes géométriques avec peu de détails car il s'affranchit d'une recherche de descripteurs locaux. Son fonctionnement non itératif permet d'obtenir un résultat fiable pour une durée bornée. Il présente une bonne précision pour des environnements complexes et sa rapidité est compatible avec une implémentation en temps réel. À l'avenir, nous souhaitons enrichir et mettre à disposition le jeu de données réalisé pour cet article. La méthode va être évaluée sur de nouveaux ensembles de données pour assurer sa bonne adaptabilité aux différents contextes. De plus, un travail de modélisation de la scène sera réalisé en utilisant l'extraction des plans proposée dans ce travail afin de mettre en évidence des primitives de plus haut niveau.

References

- [1] Sebastian Ochmann, Richard Vock, Raoul Wessel, et Reinhard Klein. Automatic reconstruction of parametric building models from indoor point clouds. *Computers and Graphics (Pergamon)*, 54:94–103, 2016.
- [2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, et J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.
- [3] Paul Besl et Neil McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [4] Philippe Brou. Using the Gaussian Image to Find the Orientation of Objects. *The International Journal of Robotics Research*, 3(4):89–125, 1984.
- [5] Sang Bum Kang et Katsushi Ikeuchi. Determining 3-D object pose using the complex extended Gaussian image. Dans *Computer Vision and Pattern Recognition (CVPR)*, pages 580–585, 1991.
- [6] François Pomerleau, Francis Colas, et Roland Siegwart. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015.
- [7] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.
- [8] A Segal, D Haehnel, et S Thrun. Generalized-ICP. *Robotics: Science and Systems*, 5:168–176, 2009.
- [9] Andrew W Fitzgibbon. Robust registration of 2d and 3d point sets. 21:1145–1153, 04 2002.
- [10] Jiaolong Yang, Hongdong Li, et Yunde Jia. Go-ICP: Solving 3D registration efficiently and globally optimally. Dans *International Conference on Computer Vision (ICCV)*, pages 1457–1464, 2013.
- [11] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, et S. Behnke. Registration with the point cloud library: A modular framework for aligning in 3-d. *IEEE Robotics Automation Magazine*, 22(4):110–124, 12 2015.
- [12] Silvio Filipe et Luís A. Alexandre. A comparative evaluation of 3d keypoint detectors in a rgb-d object dataset. Dans *International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 1, pages 476–483, 1 2014.
- [13] Radu Bogdan Rusu, Nico Blodow, et Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D registration. *IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.
- [14] Andrew E. Johnson et Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(5):433–449, 1999.
- [15] Federico Tombari, Samuele Salti, et Luigi Di Stefano. Unique signatures of histograms for local surface description. Dans *European Conference on Computer Vision (ECCV)*, pages 356–369, 2010.
- [16] Qian-Yi Zhou, Jaesik Park, et Vladlen Koltun. *Fast Global Registration*, pages 766–782. 2016.
- [17] Martin A Fischler et Robert C Bolles. Paradigm for Model. *Communications of the ACM*, 24(6):381–395, 1981.
- [18] Nicolas Mellado, Dror Aiger, et Niloy J. Mitra. Super 4PCS fast global pointcloud registration via smart indexing. *Computer Graphics Forum*, 33(5):205–215, 2014.
- [19] Kaustubh Pathak et Andreas Birk. Fast Registration Based on Noisy Planes with Unknown Correspondences for 3D Mapping. *IEEE Transactions on Robotics*, 26(3):424–441, 2010.
- [20] Martin Magnusson. *The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. Thèse de doctorat, Orebro University, 2009.

- [21] Cheng Yizong. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8):790–799, 8 1995.
- [22] François Pomerleau, M. Liu, Francis Colas, et Roland Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, 12 2012.