Affinement de nuages de points par correction de la trajectoire

Sommaire

Introduction

Dans les chapitres 3 et 4, nous avons présenté deux algorithmes d'optimisation des paramètres extrinsèques et intrinsèques avec comme objectif l'affinement de nuages de points par optimisation des paramètres de calibrage. Dans les deux cas, nous supposions que la trajectoire du véhicule mobile d'acquisition était parfaitement connue et correcte, ce qui n'est pas certain notamment en environnement urbain où les capteurs GNSS ne sont pas capables de mesurer en permanence la position du véhicule. Si on se réfère à la figure 2.6, nous avons traité l'optimisation des paramètres de deux calibrages différents, qui représentent les transformations des données brutes acquises par le système mobile vers le repère body dont le centre est la position du véhicule dans un repère monde à tout instant.
Dans ce chapitre, nous allons nous intéresser à l'optimisation de la dernière transformation nécessaire au géo-référencement : il s'agit de l'optimisation de la trajectoire du véhicule mobile. Le but est toujours d'aller dans le sens de ce qui a été présenté jusqu'à maintenant : nous reprenons la même approche, avec la même fonctionnelle à minimiser. La solution que l'on va proposer dans ce chapitre est applicable dans la méthodologie à l'optimisation de la trajectoire d'un système mobile équipé soit d'un LIDAR multi-couches, soit de plusieurs LIDARs ; en effet, nous allons aussi utiliser la redondance de données entre fibres du même capteur, et les résultats seront présentés pour des données acquises avec un capteur LIDAR Velodyne HDL-32E.

5.1 Etat de l'art sur l'optimisation de trajectoire

Pour la cartographie mobile, la connaissance de la trajectoire est indispensable pour correctement construire des cartes 3D de l'environnement. Nous avons vu dans le chapitre 2 que le géoréférencement des données est très important dans de nombreux cas d'application, et les systèmes mobiles

d'acquisition embarquent de nombreux capteurs pour connaître leur position par rapport à une référence de façon très précise : GPS, centrale inertielle, odomètres, et les données de ces différents capteurs sont fusionnées pour avoir le positionnement du véhicule.
Un domaine de recherche pour lequel le positionnement du véhicule est très important est le SLAM : en effet, pour de la « Localisation et Cartographie Simultanée », la localisation précise du véhicule est un des résultats espérés, en plus de la cartographie précise. Ces algorithmes peuvent être séparés en plusieurs catégories, comme le fait Gérossier dans sa thèse [Gérossier, 2012]. Nous pouvons réaliser une classification de ces algorithmes similaire :

- Le SLAM avec une représentation par grilles d'occupation. Cette approche permet d'obtenir une carte divisée en plusieurs cellules, et qui fournit une information de probabilité d'occupation des différentes cellules de la carte, permettant la planification de trajectoires par exemple. Ce type de SLAM est généralement utilisé en amont d'une autre méthode de SLAM, notamment l'EKF-SLAM, car il permet d'avoir une estimation de la trajectoire, qui est ensuite affinée.
- Le SLAM fondé sur une représentation par cartes topologiques. Cette fois-ci, l'environnement est représenté par des noeuds, qui représentent les différentes positions du système mobile, et des arcs qui illustrent les trajets entre deux nœuds. C'est une approche simple à mettre en œuvre, puisqu'il n'y a plus de problèmes d'incertitudes sur la position du robot, la navigation de celui-ci étant locale. Toutefois, cette approche n'est pas optimale et ne prend en compte aucune notion géométrique : c'est pourquoi il vaut mieux l'utiliser de pair avec un autre type d'approche.

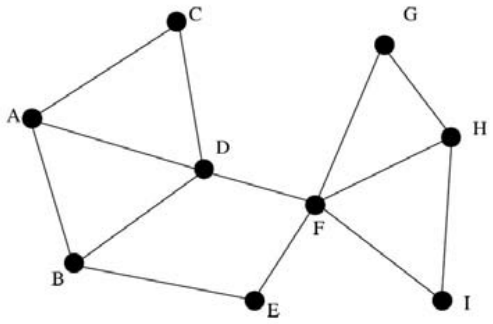


FIGURE 5.1 – Carte topologique [Bailey, 2002]

- Le Graph SLAM, qui dans l'idée est assez proche du SLAM basé sur une représentation par cartes topologiques ; les deux approches sont basées sur la construction d'un graphe avec les données acquises. Cependant, ici, les nœuds représentent aussi bien les positions successives du véhicule que les points d'intérêt extraits de l'environnement, et les arcs traduisent la présence de recouvrement entre nœuds. C'est une approche d'optimisation, car les recouvrements permettent de simplifier le Graphe construit, ce qui conduit à un problème de plus faible dimension, finalement solvable avec des techniques d'optimisations standards, comme la minimisation des moindres carrés par exemple.
- Le SLAM avec une carte d'amers. Cette approche utilise des points d'intérêts extraits des nuages de points - ou plus généralement des données acquises à l'aide des capteurs montés sur le véhicule - que l'on nomme « amers ». Le plus souvent, cette approche est utilisée dans une optique temps réel, notamment avec un filtre de Kalman, ou un filtre de Kalman étendu (EKF-SLAM), car la fusion de données est le plus souvent non linéaire. Dans une telle approche, les capteurs extéroceptifs type GPS et proprioceptifs type odomètre, centrale inertielle ont un rôle très important, car ils rentrent à part entière dans l'estimation de la localisation et la correction de celle-ci à l'aide du filtre de Kalman. Dans cette approche, le vecteur d'état

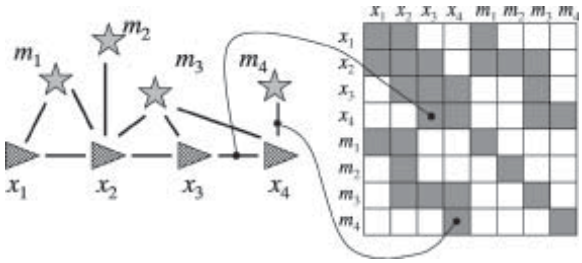


FIGURE 5.2 – Exemple de graphe utilisé dans une approche de Graph SLAM [Thrun et Montemerlo, 2005]

à chaque instant de contrôle contient les positions du véhicule et les positions des amers. C'est un SLAM très utilisé pour des applications temps réel, notamment grâce à l'utilisation de l'EKF pour la fusion de données issues des capteurs du véhicule. Tim Bailey et Hugh Durrant-Whyte, dans deux articles publiés la même année [Bailey et Durrant-Whyte, 2006a], [Bailey et Durrant-Whyte, 2006b], proposent un historique très complet sur le SLAM probabiliste, basé amers. Plusieurs solutions au problème du SLAM sont proposées, dont l'EKF, et un autre filtre, le filtre particulaire. Pour résumer la méthodologie de ce type d'approche, on peut distinguer deux temps dans l'algorithme de SLAM avec une carte d'amers : tout d'abord, le vecteur d'état du véhicule est estimé à l'aide des capteurs de mesure, puis, grâce au filtre de Kalman (par exemple), les données issues des capteurs et les amers extraits des cartes produites sont fusionnés pour affiner la localisation, et la précision des cartes dans le même temps.

- Le SLAM orienté trajectoire. Dans cette dernière approche, on cherche à estimer les positions successives du véhicule lors de la campagne d'acquisition, mais sans estimation de la trajectiore en s'appuyant sur des amers comme dans le SLAM avec une carte d'amers. Cette fois-ci, le vecteur d'état à chaque instant de contrôle ne contient que les positions du véhicule, qui sont au nombre de six : trois positions et trois rotations. Historiquement, dans ce genre d'approche, on ne cherchait à trouver un vecteur d'état composé que de trois paramètres, en supposant que le véhicule n'avait que trois degrés de libertés, qui sont les deux positions dans le plan de déplacement du véhicule et la rotation autour de l'axe vertical. Cependant, grâce à l'évolution des systèmes d'acquisitions mobiles, et surtout pour répondre à de nouvelles problématiques de cartographie comme l'explique Nüchter [Nüchter *et al.*, 2004], le vecteur d'état est maintenant composé de six paramètres : c'est ce que l'on appelle notamment le 6D-SLAM. En effet, dans son article de 2004, Nüchter présente le 6D-SLAM, qui voit deux méthodes d'optimisation distinctes : d'une part, on a le recalage en tant que problème d'optimisation. C'est le « scan matching », tel qu'il a été introduit par Besl et Mckay en 1992 [Besl et McKay, 1992], et qui consiste à trouver le vecteur d'état entre deux scans qui minimise l'énergie qu'ils définissent. D'autre part, on a aussi une optimisation orientée "scan matching", mais basée sur l'extraction d'amers : cette fois-ci, le problème à résoudre consiste à recaler les scans concernés sur la base du recouvrement de points d'intérêts extraits de chacun des scans.

A côté de ces approches de SLAM, on peut trouver d'autres approches plus spécifiques, qui dépendent des systèmes d'acquisitions utilisés, comme par exemple le visual-SLAM (V-SLAM), où une approche « image » est considérée [Lategahn *et al.*, 2011]. Effectivement, ce type de SLAM est particulier car le type de capteurs utilisé est une caméra, mais cette approche possède l'avantage de permettre l'extraction de points d'intérêts - ce que l'on appelle des amers visuels dans ce cas-là - plus facilement qu'avec un nuage de points. Un autre type de SLAM rencontré est le Cooperative-SLAM (C-SLAM) dans lequel plusieurs robots sont utilisés, comme présenté dans [Heon-Cheol *et al.*, 2011] et [Mourikis et Roumeliotis, 2006]. Ces robots vont coopérer afin de créer les cartes des environs et se localiser. Dans le cas où l'information de positionnement absolu est manquante, les robots peuvent tout de même améliorer la précision de leur positionnement en prenant en compte la position relative

par rapport aux autres robots.

5.2 Méthode d'optimisation proposée

La figure 2.6 présente les principales étapes de géoréférencement des données lors d'une cartographie mobile. Dans les chapitres 3 et 4, nous avons présenté des méthodes d'affinement de nuages de points par optimisation des paramètres de calibrage extrinsèque et intrinsèque. Ces optimisations ont été effectuées avec l'hypothèse que la position du véhicule était connue à tout instant de façon précise : cette hypothèse n'est pas complètement vérifiée, puisque la centrale inertielle et le système GPS ont des erreurs de mesures non négligeables avec le temps. C'est pour cela que nous nous sommes intéressés à l'optimisation de la trajectoire du véhicule au cours de l'acquisition. L'article qui traite de ce sujet qui nous a le plus intéressé est le suivant, [Monnier *et al.*, 2013], dans lequel une méthode pour recaler un nuage de point mobile avec une vérité terrain appartenant à une base de données est proposée. La méthode proposée est du recalage non-rigide : en effet, le caractère non-rigide du recalage est important car au cours de l'acquisition, l'erreur sur la trajectoire n'est pas constante, et il est alors préférable de ne pas appliquer la même transformation à l'ensemble des données pour avoir un meilleur recalage avec la vérité terrain.
La méthode que l'on propose pour la correction de la trajectoire va dans la continuité de ce que l'on a proposé dans les chapitres 3 et 4. En effet, nous voulons corriger les translations de la trajectoire du véhicule, et pour cela, nous supposons que les paramètres de calibrages extrinsèques et intrinsèques sont soit corrects, soit déjà corrigés.

5.2.1 Présentation de l'algorithme mis au point et optimisation

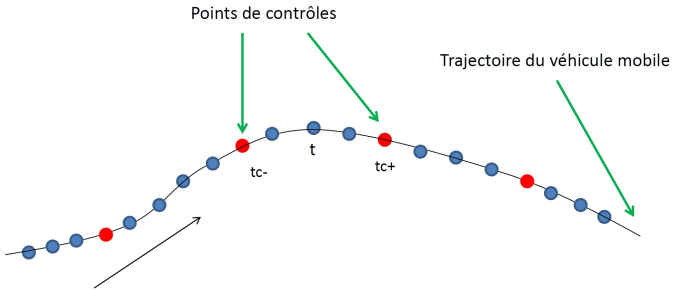


FIGURE 5.3 – Schéma d'une acquisition avec plusieurs points de contrôles

La figure 5.3 présente une acquisition avec plusieurs « points de contrôles » : un point de contrôle est un point qui sert de repère dans une acquisition. Il peut être **virtuel**, interpolé entre deux points de l'acquisition, ou **réel**. Ce point de contrôle possède un « temps de contrôle », et est en général choisi pour sa position 3D ou son temps d'acquisition : dans notre cas, et pour l'application de correction de la trajectoire, nous avons choisi des points de contrôles en fonction de leur temps d'acquisition, afin d'avoir une trajectoire subdivisée en « morceaux » de plages temporelles égales. Comme nous l'avons dit dans la section 5.2, un de nos objectifs avec l'optimisation de la trajectoire du véhicule est de proposer une méthode qui aille dans la continuité de ce qui a été fait pour l'instant. Nous cherchons à corriger la trajectoire en utilisant la structure du capteur LIDAR multi-couches et en exploitant la redondance de données entre les différentes couches. Dans ce chapitre, nous ne présentons que l'optimisation des paramètres de translation de la trajectoire : c'est une première approche pour l'affinement des nuages de points par correction de la trajectoire. Pour réduire la complexité de la correction de la trajectoire, seuls certains points sont concernés par la correction de la position du véhicule, les points de contrôles : pour les positions du véhicule mobile

sur le reste de l'acquisition, une interpolation linéaire sur les positions du véhicule aux 2 points de contrôles « encadrants » chaque point permet d'appliquer une correction de trajectoire à l'ensemble de l'acquisition.

La trajectoire que l'on cherche à corriger peut être supposée à peu près correcte : en effet, celle-ci provient de la fusion des données venant du GPS et de la centrale inertielle, qui donnent des informations précises. Certaines erreurs de positionnement par GPS et une dérive de la centrale inertielle peuvent fausser la trajectoire du véhicule, qui a besoin d'être affinée. Toutefois, ces erreurs peuvent être supposées faibles : c'est pour cela que la fonctionnelle que l'on cherche à minimiser pour la correction de la trajectoire se décompose en deux parties : une première **énergie d'attache aux données J** qui permet de corriger la trajectoire du véhicule, et une deuxième **énergie de déformation** E_{def} qui permet de contrôler l'amplitude des modifications apportées à la trajectoire, comme présenté en équation 5.1 :

$$E_{tot}(\delta X) = J(\delta X) + \lambda_{rigid} * E_{def}(\delta X) \tag{5.1}$$

La correction que l'on apporte à la translation de la trajectoire en chaque point de contrôle est la suivante : $\delta X_{t_c} = (\delta x_{t_c} \quad \delta y_{t_c} \quad \delta z_{t_c})^T$, où t_c est un temps de contrôle ; il s'agit du temps d'acquisition d'un point de contrôle. L'énergie d'attache aux données est définie en équation 5.2 :

$$J(\delta X) = \frac{\sum_{i=1}^{B} \sum_{j=i-N}^{i+N} \sum_k w_{i,j,k} * d_{i,j,k}^2(\delta X)}{\sum_{i=1}^{B} \sum_{j=i-N}^{i+N} \sum_k w_{i,j,k}} \tag{5.2}$$

avec :

$$\begin{cases} d_{i,j,k}(\delta X) = n_{i,k}^T \times (p_{i,k}(\delta X) - m_{j,k}(\delta X)) \\ p_{i,k}(\delta X) = R_{nav}(p'_{i,k}) \times (R \times p'_{i,k} + T) + Xp_{i,k} + \delta Xp_{i,k} \\ m_{j,k}(\delta X) = R_{nav}(m'_{j,k}) \times (R \times m'_{j,k} + T) + Xm_{j,k} + \delta Xm_{j,k} \\ Xp(i,k) = T_{nav}(p'_{i,k}) \\ Xm(j,k) = T_{nav}(m'_{j,k}) \\ \\ \delta Xp_{i,k} = (1 - \alpha(t_p)) * \delta X_{t_{p,c-}} + \alpha(t_p) * \delta X_{t_{p,c+}} \\ \delta Xm_{j,k} = (1 - \alpha(t_m)) * \delta X_{t_{p,c-}} + \alpha(t_m) * \delta X_{t_{p,c+}} \\ \alpha(t) = (t - t_{c-})/(t_{c+} - t_{c-}) \end{cases}$$

Les corrections sur la position du véhicule en chaque point issus de l'acquisition sont calculées à partir d'une interpolation linéaire sur les corrections des positions aux 2 points de contrôles encadrants chaque point, comme présenté dans la même équation. Pour le terme $\delta Xp_{i,k}$, t_p est le temps d'acquisition du point $p_{i,k}$, et $t_{p,c+}$ et $t_{p,c-}$ représentent respectivement les temps de contrôles supérieur et inférieur au temps t_p ; de même pour le temps t_m. Cette interpolation linéaire décrit le caractère non-rigide du recalage, puisque chaque point du nuage aura sa propre correction de trajectoire.
Comme nous l'avons évoqué, nous avons aussi ajouté une énergie de déformation contrôlant l'amplitude des corrections apportées à la trajectoire, définie en équation 5.3 :

$$E_{def}(\delta X) = \sum_{i=1}^{N_{t_c}} \|\delta X_{t_{c,i}}\|^2 \tag{5.3}$$

Dans l'équation 5.3, l'énergie E_{def} est définie comme la somme des corrections apportées aux positions du véhicule en chaque point de contrôle. N_{t_c} donne le nombre de points de contrôles gardés pour l'optimisation. Dans l'article de Monnier [Monnier *et al.*, 2013], l'énergie de déformation choisie vaut $\sum_{i=1}^{N_{t_c}-1} \|\delta X_{t_{c,i+1}} - \delta X_{t_{c,i}}\|^2$, ce qui permet de contrôler l'amplitude des corrections entre 2

positions de contrôle successives. Comme expliqué dans l'article, les auteurs cherchent à recaler une nouvelle acquisition à une référence géo-référencée, et la distance entre les deux nuages de points peut-être élevée : la seule contrainte est d'avoir une continuité sur la trajectoire du nuage de point qui est recalé, d'où une rigidité sur les corrections entre deux positions de contrôle successives. Pour notre approche d'optimisation, nous partons de l'hypothèse que la trajectoire actuelle est proche de la trajectoire réelle : nous cherchons essentiellement à corriger des inconsistances sur la trajectoire du véhicule au sein d'une même acquisition. C'est pour cela que nous avons choisi le terme de déformation présenté en équation 5.3 : entre deux positions de contrôles successives, la correction à apporter à la translation de la trajectoire peut-être différente, mais elle reste assez faible.

Pour contrôler les corrections qui sont apportées à la trajectoire, en plus de l'énergie de déformation, un terme de rigidité λ_{rigid} adimensionnel est ajouté à l'optimisation, qui permet de contrôler l'amplitude des déformations. Selon sa valeur, le résultat d'optimisation sera différent :

- si le λ_{rigid} est faible, des amplitudes de corrections plus élevées seront permises, ce qui permet de corriger les erreurs de trajectoires importantes, au risque de trop s'éloigner de la trajectoire réelle du véhicule.
- si le λ_{rigid} est élevé, les corrections apportées seront plus petites, mais le nuage affiné aura une trajectoire corrigée proche de celle d'origine, ce qui peut être préférable dans certains cas. L'optimisation est aussi plus lente puisque les corrections voulues sont faibles.

Comme nous l'avons évoqué précédemment, nous prenons un certain nombre de points de contrôle N_{t_c} pour optimiser la trajectoire, et l'équation 5.1 peut alors se réécrire sous la forme :

$$E_{tot}(\delta X) = \frac{\sum_{i=1}^{B}\sum_{j=i-N}^{i+N}\sum_k w_{i,j,k} * (D_{i,j,k} + C^T_{i,j,k} \times \delta X)^2}{\sum_{i=1}^{B}\sum_{j=i-N}^{i+N}\sum_k w_{i,j,k}} + \lambda_{rigid} * \sum_{i'=1}^{N_{t_c}} \|\delta X_{t_{c,i'}}\|^2 \qquad (5.4)$$

avec :

$$\begin{cases} \delta X = ([\delta x_l \quad \delta y_l \quad \delta z_l]_{,l\in[\![1,N_{t_c}]\!]})^T \\ D_{i,j,k} = n^T_{i,k} \times \big(T_{nav}(p'_{i,k}) - T_{nav}(m'_{j,k}) + (R_{nav}(p'_{i,k}) \times (R \times p'_{i,k} + T)) - (R_{nav}(m'_{j,k}) \times (R \times m'_{j,k} + T))\big) \\ C_{i,j,k} = \begin{bmatrix} \dots \\ (1-\alpha(tp_{i,k})) * n_{i,k} \\ \alpha(tp_{i,k}) * n_{i,k} \\ \dots \\ -(1-\alpha(tm_{j,k})) * n_{i,k} \\ -\alpha(tm_{j,k}) * n_{i,k} \\ \dots \end{bmatrix} \end{cases}$$

Dans l'équation 5.4, δX représente le vecteur des corrections que l'on apporte à la trajectoire en chaque point de contrôle sélectionné : il a une taille de $3 * N_{t_c}$, car pour chaque point de contrôle, les 3 paramètres de translations sont optimisés. $C_{i,j,k}$ est un vecteur de taille $3 * N_{t_c}$ (3 paramètres de position par temps de contrôle), avec entre 6 et 12 termes au maximum non nuls, selon le temps de contrôle considéré. Pour le point p, 6 termes sont non nuls, comme indiqué dans le vecteur $C_{i,j,k}$, et proviennent de l'interpolation linéaire effectuée entre les temps de contrôle supérieur et inférieur au point p. Les 6 termes non nuls du vecteur sont les termes $3 * t_{p,c-}$, $3 * t_{p,c-} + 1$ et $3 * t_{p,c-} + 2$, ainsi que $3 * t_{p,c+}$, $3 * t_{p,c+} + 1$ et $3 * t_{p,c+} + 2$. Nous avons la même construction pour le point m, avec les temps de contrôles $t_{m,c-}$ et $t_{m,c+}$ et où 6 termes sont aussi non nuls. Dans le cas où $t_{p,c-} = t_{m,c-}$ et $t_{p,c+} = t_{m,c+}$, seuls 6 termes sont non nuls (au lieu de 12). $n_{i,k}$ est la normale au plan passant par le point k acquis par la fibre i.

Finalement, la solution qui minimise l'équation (5.4) est la solution du système linéaire suivant :

$$C \times \delta X = -V \tag{5.5}$$

avec :

$$\begin{cases} C = \sum_{i=1}^{B} \sum_{j=i-N}^{i+N} \sum_{k} w_{i,j,k} * (C_{i,j,k} \times C_{i,j,k}^{T}) + \lambda_{rigid} * I_{3n} \\ V = \sum_{i=1}^{B} \sum_{j=i-N}^{i+N} \sum_{k} w_{i,j,k} * (D_{i,j,k} \times C_{i,j,k}) \end{cases}$$

5.2.2 Validation du résultat de l'optimisation

Contrairement aux chapitres 3 et 4, nous ne pouvons pas directement utiliser la valeur finale de l'énergie pour valider le résultat d'optimisation. En effet, la fonctionnelle que l'on cherche à minimiser pour l'optimisation de la trajectoire est composée de 2 termes :

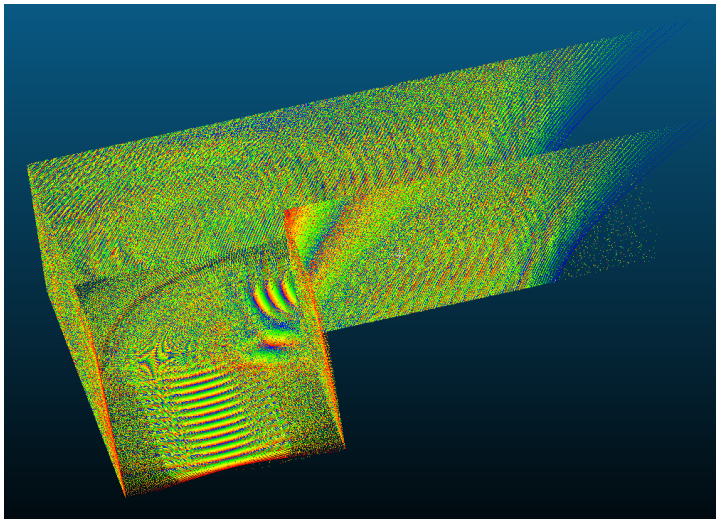
- un terme d'attache aux données, qui est similaire aux énergies minimisées dans les deux chapitres précédents.
- un terme de déformation, qui permet de contrôler l'amplitude des déformations que l'on applique à la trajectoire. Ce terme ne permet plus d'utiliser la valeur de l'énergie J après convergence comme critère de validation de l'optimisation, parce que ce terme peut être élevé en fonction de la valeur de λ_{rigid} que l'on choisit.

L'énergie que l'on a définie n'a plus le même sens physique qu'avant. Aussi, pour valider notre résultat d'optimisation, nous mesurons des distances nuage à nuage : pour le jeu de données simulées, nous comparons les distances entre la vérité terrain et le nuage avec une mauvaise trajectoire d'un côté, et entre la vérité terrain et le nuage dont la trajectoire est optimisée de l'autre ; le résultat attendu est une distance par rapport à la vérité terrain plus faible dans le cas où la trajectoire est optimisée. Pour le jeu de données réelles, la validation est visuelle, car nous n'avons pas de vérité terrain avec laquelle comparer le résultat d'optimisation.

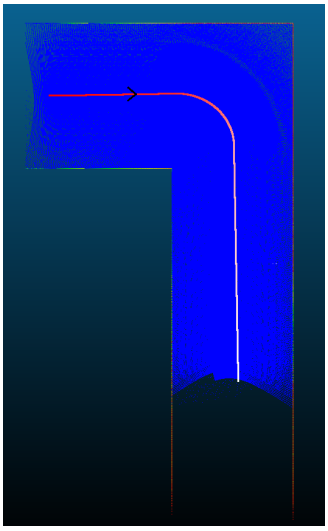
5.3 Résultats expérimentaux

Pour les paramètres à régler dans notre optimisation (nombres de fibres voisines recalées entre elles ; taille du voisinage pour le calcul des normales), nous avons utilisé les mêmes valeurs que celles retenues pour les sections 3.4.2 et 4.4. Pour l'expérimentation réelle, un sous-échantillonnage plus important a été effectué car le nuage est composé d'un très grand nombre de points et possède de nombreux recouvrements entre différents instants d'acquisitions. Deux nouveaux paramètres apparaissent dans cette optimisation : d'une part, nous avons le nombre de temps de contrôle à prendre en compte pour la correction des paramètres de translation de la trajectoire, et d'autre part le paramètre λ_{rigid} de rigidité. Différents tests ont montré qu'un espacement de l'ordre de la seconde des temps de contrôles donnait les résultats optimaux. Pour le paramètre d'attache aux données, son ordre de grandeur détermine la « liberté » laissée aux données pour se déformer ou non : plus la valeur est grande, plus faiblement les paramètres de trajectoires seront modifiés, et inversement. Dans [Monnier *et al.*, 2013], une description de son ordre de grandeur est donnée : dans notre optimisation, nous donnerons moins d'importance à l'attache aux données qu'à la correction de la trajectoire. Aussi, en fonction des caractéristiques de notre système d'acquisition et du résultat que l'on souhaite avoir, l'ordre de grandeur que l'on a retenu pour λ_{rigid} est de 10^2.

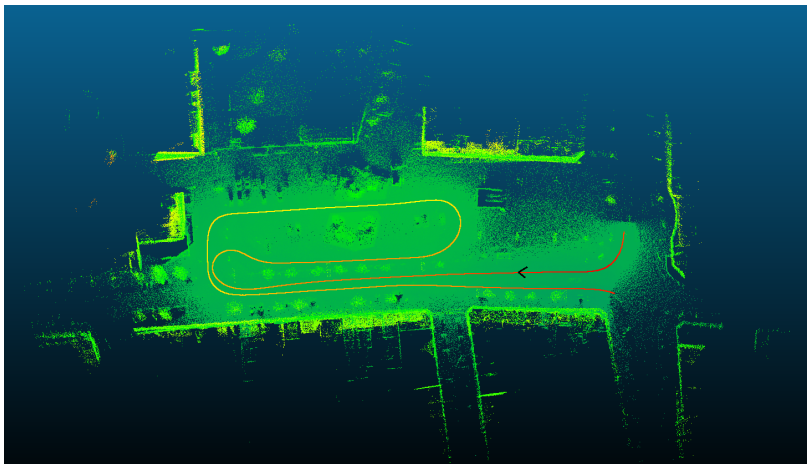
Dans la suite de cette section, nous allons présenter 2 résultats d'optimisation des paramètres de translations de la trajectoire : nous avons un nuage simulé et un nuage réel, qui ont les propriétés suivantes :



(a) Jeu de données simulées #6



(b) Trajectoire du nuage #6



(c) Jeu de données réelles #7, avec sa trajectoire

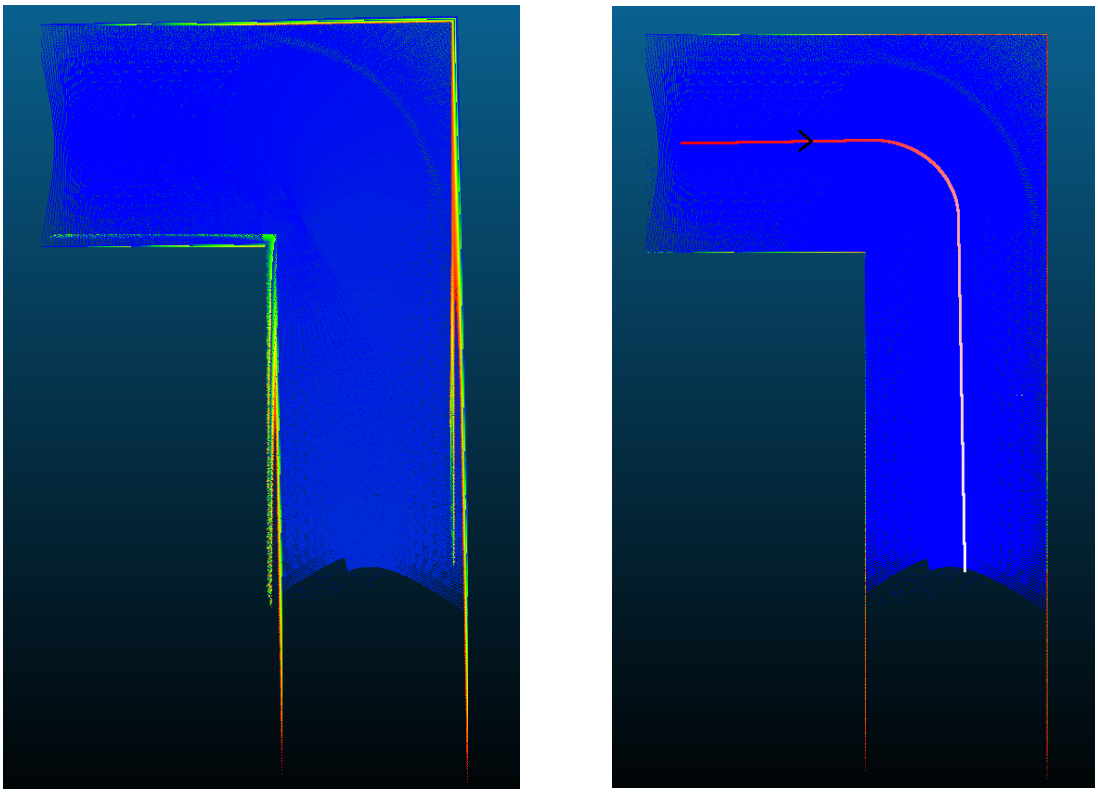
FIGURE 5.4 – Jeux de données utilisés pour l'optimisation de la trajectoire

- la figure 5.4 a) présente le jeu de données simulées utilisé pour valider notre approche d'optimisation. La scène est composé d'un sol et de 4 plans verticaux représentant des façades, et le véhicule effectue un virage lors de son déplacement. Enfin, il n'y a pas de variation d'altitude. Le nuage est composé d'environ 4 millions de points. Un nombre de 5 points de contrôles a été retenu pour ce nuage simulé, afin que l'on ait des points de contrôles dont les temps d'acquisitions soient espacés d'environ 1 seconde. La figure 5.4 b) présente la trajectoire du nuage simulé #6.

- la figure 5.4 c) présente le nuage réel utilisé pour l'optimisation de la trajectoire. Il s'agit d'une acquisition effectuée à Lille, et la portion de l'acquisition utilisée représente une place de marché sur laquelle le véhicule d'acquisition effectue une boucle. Cette fois-ci, nous prenons 50 points de contrôles, car le nuage possède environ 50 millions de points, et que cela nous permet d'avoir un intervalle entre les temps d'acquisitions des points de contrôles de l'ordre de la seconde.

5.3.1 Résultats sur un jeu de données simulées

Pour le jeu de données simulées #6, nous avons la vérité terrain : il est donc possible de comparer notre résultat d'optimisation à cette vérité. Pour les tests, nous sommes partis de la vérité terrain à laquelle nous avons rajouté une erreur linéaire par morceaux à la trajectoire. Pour vérifier le résultat d'optimisation, c'est-à-dire si la trajectoire a correctement été optimisée pour que le nuage après optimisation soit très proche de la vérité terrain, nous avons mesuré la distance nuage à nuage entre la vérité terrain et le nuage avant optimisation d'une part, puis entre la vérité terrain et le nuage après optimisation. Avant optimisation, nous avons une distance de 13.36 cm environ, alors qu'après optimisation, cette distance vaut environ 6 mm : il y a une très nette amélioration de la qualité du nuage. La figure 5.5 présente visuellement le résultat de l'affinement par optimisation de la trajectoire, et on peut voir que l'on a bien une nette amélioration par rapport au nuage de départ avec une trajectoire erronée.
Le temps de calcul pour cette optimisation est d'environ 2 minutes car le nombre de temps de contrôles est assez faible : nous avons 15 paramètres seulement à optimiser.



(a) (b)

FIGURE 5.5 – Résultats de l'optimisation de la trajectoire pour le nuage #6 : a) jeu de données avant optimisation ; b) jeu de données après optimisation

5.3.2 Résultats sur un jeu de données réelles

Le jeu de données réelles #7 provient d'une acquisition effectuée à Lille, dans le cadre du projet Terramobilita. La zone présentée sur la figure 5.4 c) est une place dans laquelle le véhicule d'acquisition a effectué un aller-retour, comme le montre la trajectoire présentée sur la même figure. Ce jeu de données est intéressant parce que la trajectoire n'est pas optimale, et les recouvrements dus à l'aller-retour lors de la cartographie ne se superposent pas comme ils devraient, comme le montrent

les figures 5.7 et 5.8, qui sont des vues de dessus de l'acquisition. Pour ce jeu de données, nous n'avons pas de vérité terrain, et la validation est visuelle : nous savons qu'avec la trajectoire avant optimisation, certaines zones du nuage ont des problèmes, et nous voulons que l'optimisation des paramètres de trajectoire règle en grande partie ces problèmes de recouvrement. C'est le cas, comme le montrent les figures 5.7 b) et 5.8 b) : en effet, on voit une nette amélioration au niveau des zones entourées, avec un meilleur recouvrement des données issues de différents instants d'acquisitions. Le temps de calcul pour cette optimisation est d'environ 15 minutes.

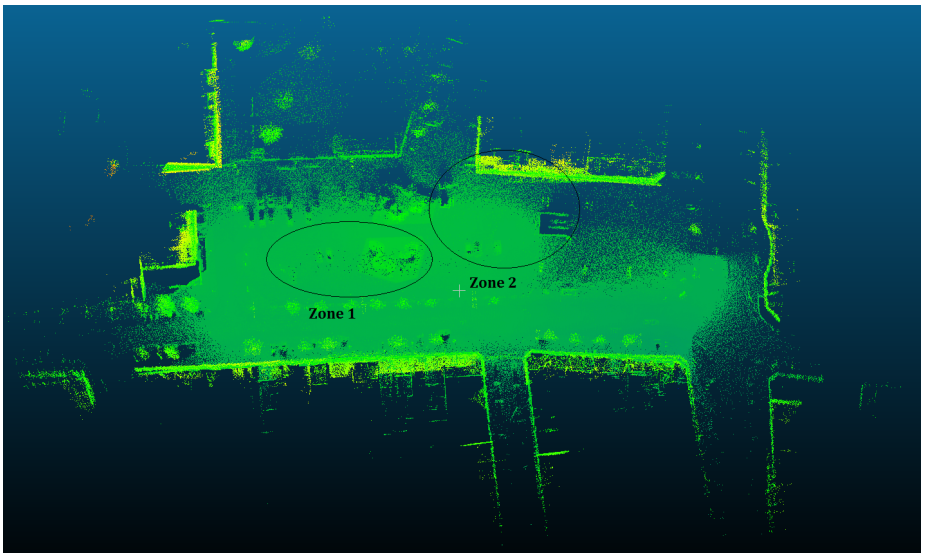
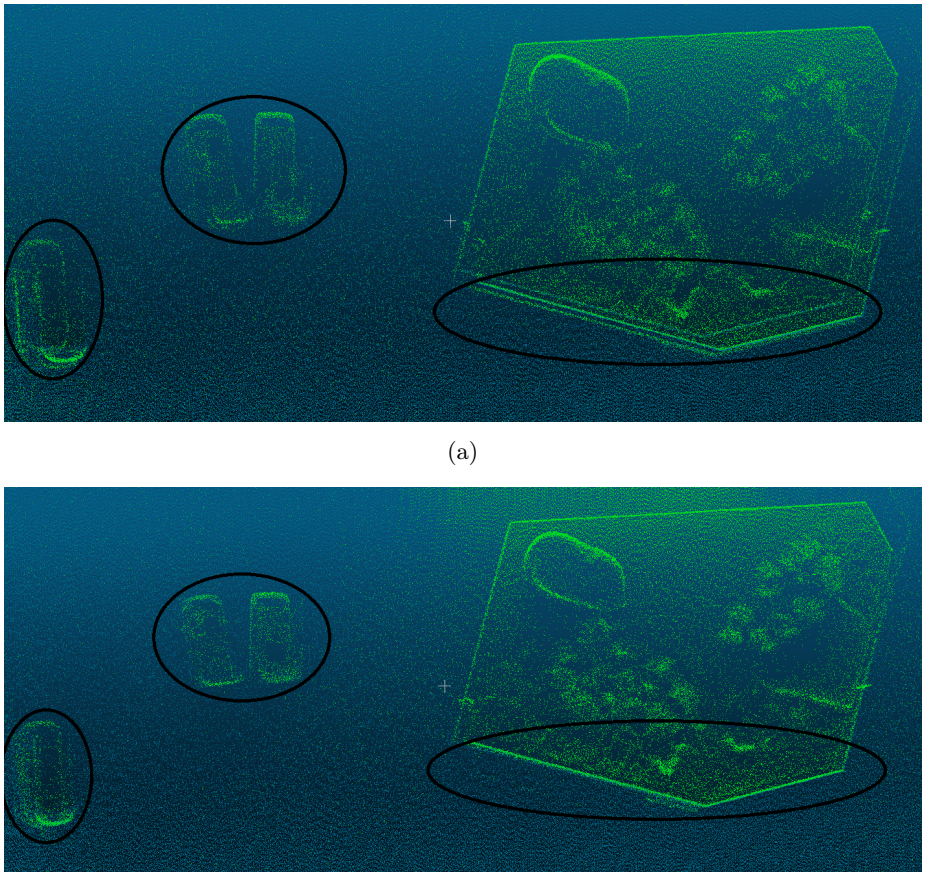


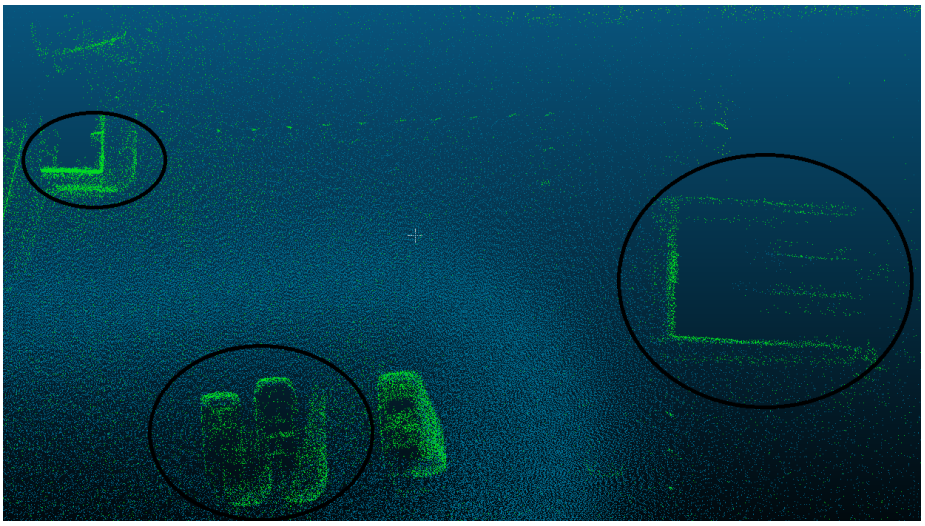
FIGURE 5.6 – Nuage #7, présentant les deux zones que l'on compare avec l'optimisation



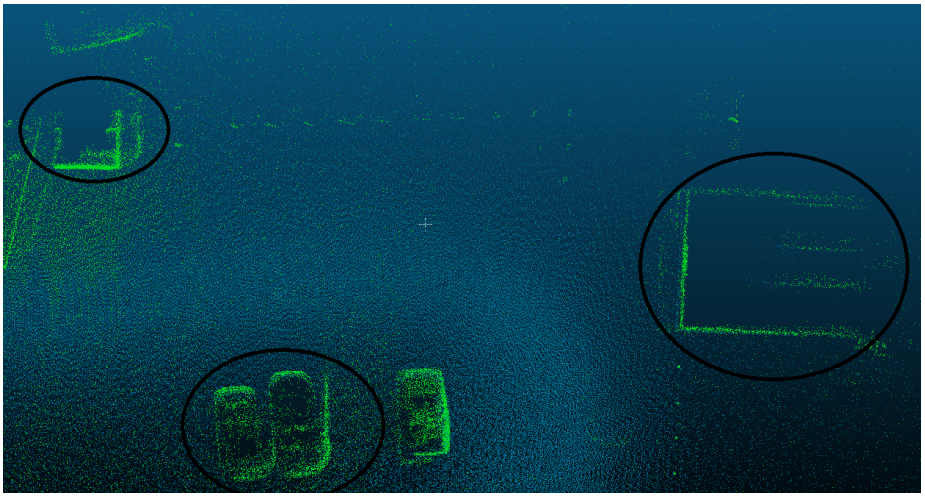
(a)

(b)

FIGURE 5.7 – Résultats de l'optimisation de la trajectoire pour le nuage #7 sur une première zone du nuage : a) jeu de données avant optimisation ; b) jeu de données après optimisation



(a)



(b)

FIGURE 5.8 – Résultats de l'optimisation de la trajectoire pour le nuage #7 sur une autre zone du nuage : a) jeu de données avant optimisation ; b) jeu de données après optimisation

5.4 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'affinement de nuages de points par optimisation de la trajectoire du véhicule lors de l'acquisition. Nous n'avons traité que de l'optimisation des paramètres de translations de la trajectoire pour avoir une première approche d'optimisation de la trajectoire. Au vu des résultats obtenus, l'affinement par optimisation de la trajectoire donne des résultats satisfaisants, qui nous permettent de valider l'approche. Les résultats sont satisfaisants pour l'approche que l'on a retenue : par rapport aux approches qui ont été présentées dans les chapitres 3 et 4, nous ne nous sommes intéressés qu'à l'optimisation des paramètres de translations. Pour le jeu de données réelles #7, les figures 5.7 et 5.8 présentent des améliorations, mais une erreur de recalage persiste, et cette erreur semble due à un décalage angulaire : la centrale inertielle donne des valeurs de roulis et de tangage précises, mais le lacet est plus difficilement mesurable par celle-ci. Aussi, comme indiqué dans la section 5.3.2, les images des résultats sont présentés en vues de dessus, et le décalage angulaire est alors essentiellement représenté par un biais au niveau du lacet. C'est pour cela qu'une piste d'amélioration de notre approche d'optimisation de la trajectoire est de prendre en compte les angles : les premiers résultats obtenus indiquent assez clairement qu'une optimisation des angles de rotations du véhicule permettrait d'affiner le nuage réel #7 de façon plus significative.