



# **Application et analyse de GNSS dans l'industrie forestière**

**Mémoire**

**Philippe Dandurand**

**Maîtrise en informatique - avec mémoire**  
Maître ès sciences (M. Sc.)

Québec, Canada

# **Application et analyse de GNSS dans l'industrie forestière**

**Mémoire**

**Philippe Dandurand**

Sous la direction de:

Philippe Giguère  
François Pomerleau

## Résumé

La localisation est un problème qui affecte beaucoup d'industries. Comme l'automatisation devient de plus en plus importante, la localisation doit aussi devenir de plus en plus précise pour assurer une bonne efficacité des opérations et pour assurer la sécurité des employés qui vont devoir travailler avec cette automatisation. Notre étude se concentre sur la problématique de la localisation, appliquée à l'industrie forestière. Comme cela demande l'automatisation de machinerie lourde et la manipulation de l'inventaire, il est important d'avoir une localisation précise sur une échelle du décimètre. Une solution possible pour la localisation de précision est via les récepteurs Global Navigation Satellite System (GNSS) Real Time Kinematic (RTK), qui permettent de faire la localisation à une dizaine de centimètres près. Par contre, l'industrie forestière travaille dans des environnements divers qui peuvent causer des difficultés pour ces récepteurs GNSS. Par exemple dans des forêts denses ou près de bâtiments qui peuvent cacher des satellites pour la localisation. Comme l'environnement du récepteur et la qualité de la réception sont mesurables, nous avons aussi développé une méthode permettant de faire une relation entre les deux.

Ce mémoire tentera d'expliquer la problématique de l'application et l'analyse de GNSS pour faire la localisation dans des environnements divers. Ensuite, le mémoire va montrer comment les récepteurs GNSS sont affectés par divers environnements en montrant des résultats d'expériences dans ces derniers. Le tout sera démontré dans le chapitre 1 et la section 4.1 Comme l'environnement est important pour estimer son influence sur les receveurs GNSS, la génération de carte 3D est importante. Cette génération est expliquée dans le chapitre 2 et la section 4.2. Une plate-forme de récolte de données que nous avons conçue sera expliquée en grand détail dans le chapitre 3. Ensuite, nous avons fait une estimation de la qualité de la visibilité de satellite avec un article scientifique dans le chapitre 5. Les résultats y montrent que notre méthode obtient une bonne relation entre l'environnement et la performance des récepteurs GNSS. Ces résultats sont tirés d'expériences réalisées dans divers environnements.

# Table des matières

Résumé	ii
Table des matières	iii
Liste des figures	v
Liste des tableaux	vi
Liste des abréviations	vii
Remerciements	viii
Avant-propos	ix
Introduction	1
<b>1 Localisation GNSS</b>	<b>3</b>
1.1 Utilisation de la technologie GNSS dans l'industrie forestière . . . . .	3
1.2 La technologie GNSS . . . . .	4
1.3 Source d'erreurs des systèmes GNSS . . . . .	5
1.4 Systèmes de correction d'erreurs GNSS . . . . .	8
<b>2 Cartographie 3D via lidar</b>	<b>16</b>
2.1 Données . . . . .	16
2.2 Théorie . . . . .	17
2.3 Outil de génération . . . . .	18
<b>3 Plate-forme de saisie de données</b>	<b>24</b>
3.1 Caractéristique matérielle de l'ordinateur . . . . .	25
3.2 Solution logicielle . . . . .	26
3.3 Capteurs . . . . .	26
3.4 Plate-forme finale . . . . .	27
<b>4 Résultats pour la validation du matériel</b>	<b>30</b>
4.1 Résultats sur les données GNSS . . . . .	30
4.2 Résultats sur la génération de cartes . . . . .	45
4.3 Conclusion . . . . .	48

<b>5</b>	<b>Article FSR 2019</b>	<b>50</b>
5.1	Résumé . . . . .	50
5.2	Abstract . . . . .	51
5.3	Introduction . . . . .	51
5.4	Related work . . . . .	53
5.5	Theory . . . . .	55
5.6	Experimental Setup and Methodology . . . . .	61
5.7	Results . . . . .	65
5.8	Conclusion . . . . .	69
	<b>Conclusion</b>	<b>71</b>
	<b>Bibliographie</b>	<b>72</b>

# Liste des figures

1.1	Système et équations du GNSS . . . . .	5
1.2	Couverture courante du système Wide Area Augmentation System (WAAS)	9
1.3	Illustration de la méthode RTK . . . . .	12
1.4	Exemple de données en format National Marine Electronics Association (NMEA). . . . .	13
1.5	Emlid Reach . . . . .	15
2.1	Exemple de données lidar . . . . .	17
2.2	Exemple d'erreur entre deux surfaces . . . . .	18
2.3	Étapes de l'algorithme de cartographie . . . . .	20
2.4	Normale de surfaces des points . . . . .	22
2.5	Réorientation des normales de surface . . . . .	23
3.1	Plate-forme finale : structure extérieure. . . . .	28
3.2	Plate-forme finale : structure intérieure . . . . .	29
4.1	Robot <i>Husky</i> utilisé pour la collecte de données. . . . .	31
4.2	Station de référence utilisée pour la collecte de données. . . . .	32
4.3	Résultat de post-traitement . . . . .	34
4.4	Comparaison avec et sans correction . . . . .	35
4.5	Comparaison odomètre et GNSS avec correction . . . . .	36
4.6	Comparaison odomètre et GNSS sans correction . . . . .	37
4.7	Exemple de T-shed . . . . .	38
4.8	Résultat de trajectoire obstruée avec et sans correction . . . . .	38
4.9	Comparaison odomètre et GNSS avec correction . . . . .	39
4.10	Comparaison odomètre et GNSS sans correction . . . . .	39
4.11	Résultat de post-traitement avant chute de feuilles . . . . .	41
4.12	Résultat de post-traitement après chute de feuilles . . . . .	42
4.13	Histogramme de distance des diverses trajectoires. . . . .	43
4.14	Histogramme de nombre de satellites de la première trajectoire (Figure 4.11), avant la chute des feuilles, pour chacune des catégories de positionnement. . . . .	44
4.15	Histogrammes du nombre de satellites de la deuxième trajectoire (Figure 4.12), après la chute des feuilles, pour chacune des catégories de positionnement. . . . .	44

4.16	Histogrammes du nombre de satellites de la trajectoire Figure 4.3, pour chacune des catégories de solution. . . . .	45
4.17	Carte 3D du stationnement du Pouliot . . . . .	47
4.18	Carte 3D du grand axe de l’université Laval . . . . .	47
4.19	Carte du garage de la forêt Montmorency . . . . .	48
4.20	Carte 3D du pavillon Paul Comtois . . . . .	48
5.1	Side view of a complex environment . . . . .	53
5.2	Different uncertainty information is accessible through NMEA sentences.	57
5.3	Example of signal reduction . . . . .	60
5.4	Different information extracted from the geometry of a point cloud for the absorption model and the ground segmentation. . . . .	61
5.5	Experimental setups . . . . .	62
5.6	Overview of all trajectories used during our experiments, all of them in Quebec, Canada. . . . .	64
5.7	Viable satellite model comparison . . . . .	66
5.8	Relation between satellite visibility and the number of satellites seen. . .	67
5.9	Validation trajectory results . . . . .	68
5.10	Predicted coverage on the ground plane. . . . .	68
5.11	Effect of the resolution on the binary mask <b>B</b> . . . . .	69

## Liste des tableaux

1.1	Liste de différentes solutions RTK commerciales. . . . .	14
5.1	List of parameters used for our results. . . . .	65

# Liste des abréviations

- Differential Global Navigation Satellite System (DGNSS)
- Global Navigation Satellite System (GNSS)
- Global Positioning System (GPS)
- Globalnaya Navigazionnaya Sputnikovaya Sistema (GLONASS)
- Real Time Kinematic (RTK)
- Dilution Of Precision (DOP)
- Signal to Noise Ratio (SNR)
- East North Up (ENU)
- Inertial Measurement Unit (IMU)
- Iterative Closest Point (ICP)
- National Marine Electronics Association (NMEA)
- Geographic Information System (GIS)
- Robot Operating System (ROS)
- Wide Area Augmentation System (WAAS)
- Temps Universel Coordonné (UTC)
- Field-Programmable Gate Array (FPGA)
- Field and Service Robotics (FSR)

# Remerciements

J'aimerais remercier Philippe Giguère pour l'aide et la confiance qu'il a mise en moi. Je remercie François Pomerleau pour les conseils et toute l'aide qu'il m'a apportée lors de mon projet. Je remercie tout le monde du laboratoire Norlab : Philippe Babin, Jean-François Tremblay, Maxime Vaidis, David Landry et Vladimír Kubelka, pour tout l'aide qu'ils m'ont donné lors de ma maîtrise. Je remercie le personnel et les techniciens de la forêt Motmonrency pour les formations et le matériel qui nous a été prêté.

# Avant-propos

Pour introduire la motivation, les notions et l'utilisation de DGNSS dans l'industrie forestière, un rapport technique a été réalisé. Le chapitre 1 de ce mémoire reprend l'essentiel de ce rapport technique réalisé pour le consortium de recherche FORAC. Ce consortium regroupe l'Université Laval avec des organisations et des gouvernements pour améliorer l'industrie forestière au Québec.

Comme l'analyse de l'environnement est un aspect important de ce document, des cartes sur la forme de nuage de point 3D contenant l'information sur la géométrie de l'environnement sont générées. Le chapitre 2 va donner une base sur la génération des cartes en format de nuage de point 3D venant des données d'une plate-forme expliqué dans le chapitre 3. De plus, le chapitre discute des outils et paramètres utilisés.

Le chapitre 3 explique en détail le développement et le fonctionnement de la plate-forme que nous avons conçu pour faire des récoltes de données. Cette plate-forme a été utilisée pour les tests du chapitre 5.

Le chapitre 4 montre les résultats venant des expériences de l'utilisation de DGNSS dans des environnements divers et de génération de carte comme expliquée dans le chapitre 2.

Le chapitre 5 de ce mémoire est un article inséré. Cet article a été publié à la 12<sup>e</sup> conférence FSR. En attendant que l'article scientifique soit publié, l'article est disponible sur un dépôt *arXiv* avec l'identifiant 1904.07837 . Pour cet article, j'étais responsable de faire la conception et l'exécution des expériences. En tant que premier auteur, l'écriture de l'article était aussi une de mes responsabilités. Les professeurs Pomerleau et

Giguère ont donné des conseils qui ont été d'une grande importance tout au long du processus et Philippe Babin a participé à la conception des expériences. Cela depuis le début du projet en janvier 2019 jusqu'à la soumission du 10 avril. De cet article, nous avons obtenu plusieurs méthodes. Premièrement, une méthode d'extraction d'obstacles structurés, tel un bâtiment, est non structurée (telles des forêts) dans l'environnement d'un GNSS. Deuxièmement, une méthode d'estimation du nombre de satellites visibles pour la localisation du GNSS. Ces deux résultats sont discutés dans le chapitre 5.

# Introduction

Depuis quelques années, une nouvelle révolution industrielle, nommée *Industrie 4.0*, est en train de modifier le mode de fonctionnement de nombreuses industries. On peut voir ces changements chez *Amazon* avec leurs entrepôts intelligents, et même dans l'industrie automobile. Ces changements vont aussi affecter des industries plus traditionnelles, par exemple l'agriculture et l'industrie forestière. Pour cette dernière, il y a peu de méthodes en ce moment pour faire le suivi des opérations. Comme une étape essentielle du suivi des opérations en foresterie implique la saisie de données géolocalisées, il est important de déterminer les bons outils pour résoudre cette partie du problème.

Le but de ce document est d'apporter des pistes de solution à ce problème, via la localisation de précision avec des appareils de récolte de données munis d'un *Real Time Kinematic (RTK) Global Navigation Satellite System (GNSS)*, ainsi que de mieux comprendre l'effet de l'environnement sur la qualité de cette localisation. L'une des principales composantes des approches *Industries 4.0* est l'analyse des données d'opérations. Nous avons donc conçu à cet effet une plate-forme informatique de récolte de données pouvant accommoder une variété de capteurs. Cette plate-forme, décrite au chapitre 3 a été développée pour une diversité d'environnement, car la machinerie lourde opère généralement en milieu extérieur. Aussi, il était important de minimiser l'impact sur l'opérateur de cette machinerie. Finalement, cette plate-forme devait être indépendante du véhicule, pouvant être installée en son extérieur.

En ce qui a trait à la localisation des équipements en milieu extérieur, nous ferons appel à la technologie GNSS, notamment celles exploitant le RTK. La méthode de RTK implique qu'un récepteur GNSS reçoive des messages de corrections d'une base

de référence, d'où le nom Differential Global Navigation Satellite System (DGNSS).

Le chapitre chapitre 1 se concentre principalement sur l'utilisation de la technologie GNSS dans les environnements utilisés par l'industrie forestière. Nous présenterons et analyserons à la section 4.1 des données provenant de récoltes visant à 1) évaluer la performance du GNSS en général et 2) l'impact de différents environnements sur la précision de la localisation. Ce dernier point permettra de mieux juger de l'importance de la correction des solutions appliquées aux données GNSS à l'aide d'une base de référence, tant au niveau de la précision que de la robustesse. Naturellement, pour avoir une idée de l'impact de l'environnement, nous avons besoin d'avoir de l'information sur celui-ci sous un format compréhensible. Cette information sera sous la forme d'une carte faite d'un nuage de points 3D. Le chapitre 2 explique les outils utilisés pour la génération de carte utilisée. Dans la section 4.2, nous pouvons voir les résultats de la génération de carte.

Finalement, le chapitre 5 présent notre article, qui utilise les données récoltées par la plate-forme pour prédire la visibilité des satellites en utilisant une constellation de satellites et un nuage de points 3D représentant l'environnement. Comme la visibilité de satellites est un bon indicateur de la qualité de la localisation, cela implique que nous pourrions prédire la précision de localisation d'un récepteur GNSS.

# Chapitre 1

## Localisation GNSS

### 1.1 Utilisation de la technologie GNSS dans l'industrie forestière

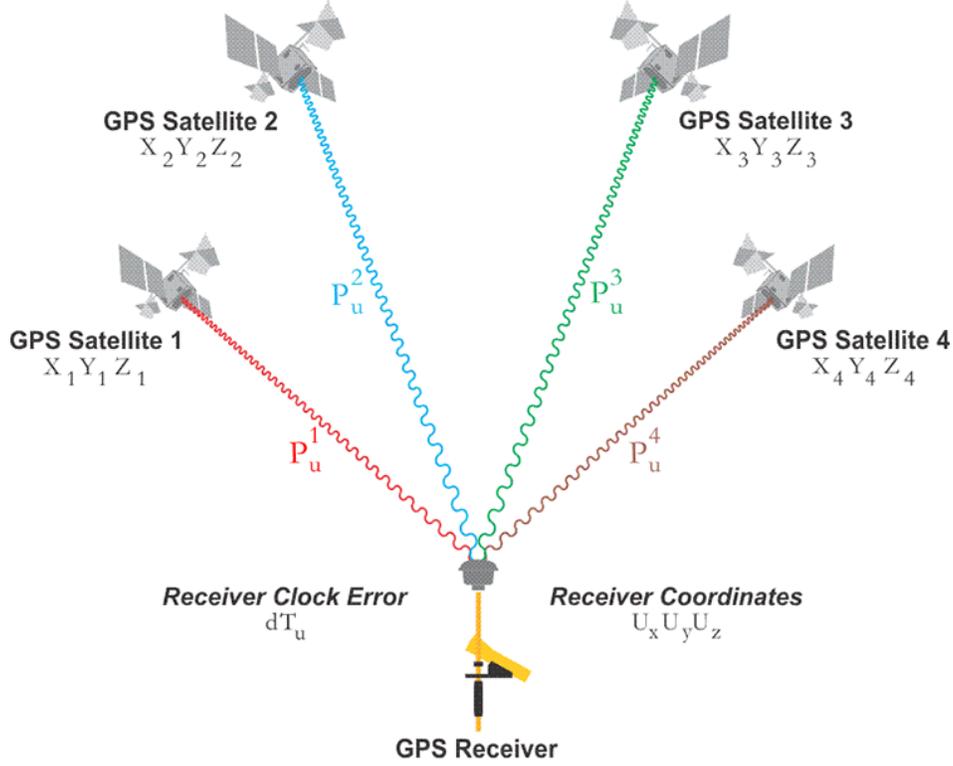
La logistique de la cour à bois est un problème important dans l'industrie forestière. L'automatisation de cette logistique est donc un objectif important pour cette industrie, car elle pourra permettre un meilleur suivi et l'optimisation du processus d'affaires via une réduction des temps d'opération. Une première étape pour permettre cette automatisation est d'analyser les déplacements des équipements. Pour accomplir ceci, nous avons développé une plate-forme pouvant être installée sur de la machinerie lourde, et enregistrer des données provenant de divers capteurs. Pour la localisation, divers capteurs peuvent être utilisés (par exemple : une centrale inertielle, un détecteur de distance 3D, ou l'odométrie du véhicule). Dans ce chapitre, nous allons nous concentrer sur une solution : le GNSS. Les capteurs GNSS sont très utilisés pour la localisation dans d'autres industries. Par exemple en aviation pour la navigation, en automobile pour faire le guidage ou en agriculture pour optimiser les opérations [1]. Dans les prochaines sections, nous expliquons en plus grand détail son fonctionnement et ses limites.

## 1.2 La technologie GNSS

La technologie du GNSS est devenue progressivement facile d'accès avec son avancement technologique. De plus, la demande pour une localisation robuste et précise en automatisation alimente ces progrès [2]. Concrètement, les solutions GNSS sont basées sur des récepteurs terrestres qui reçoivent et analysent des signaux provenant de satellites. Ces signaux contiennent notamment la position de ces satellites dans l'orbite de la terre, qui sont aussi appelés éphémérides, et le décalage de l'horloge du récepteur par rapport à celle du satellite. En ayant ces informations et le temps d'arrivée des signaux provenant d'un minimum de quatre satellites, un récepteur peut calculer une distance pour chaque satellite en se basant sur le temps de vol. Ensuite, en sachant la distance entre chaque satellite et le récepteur, un système d'équations non linéaires peut être généré et résolu ensuite avec la méthode de Gauss-Newton. La Figure 1.1 illustre ce problème, ainsi que le détail des équations à résoudre. Cette résolution va donc donner une réponse au problème de moindres carrés et donner une solution représentant le positionnement d'un récepteur. Pour trouver une solution à ce système, il faut avoir au moins 4 équations, d'où le requit d'observer un minimum de 4 satellites. De surcroît, plus le récepteur verra de satellites, plus d'équations seront ajoutées au système, et conséquemment la précision de la solution et du positionnement s'en trouvera généralement améliorée. Un exemple de système de calcul peut être vu dans la Figure 1.1. Dans cet exemple, on peut voir qu'avec quatre coordonnées de satellites, il est possible de résoudre le système pour obtenir les coordonnées du récepteur.

Auparavant, les GNSS étaient appelés *Global Positioning System (GPS)*. Il était nommé ainsi, parce que le seul système de satellites en marche était celui des États-Unis, et il portait le nom de GPS. Maintenant, il y a plusieurs constellations de satellites. En effet, il y a Galileo par l'Union européenne, Globalnaya Navigazionnaya Sputnikovaya Sistema (GLONASS) par la Russie et BeiDou-3 par la Chine. C'est grâce à ces nouveaux systèmes et à la nouvelle capacité de recevoir des messages de différents systèmes que ces capteurs ont commencé à être renommés GNSS. Une partie de l'amélioration récente de la précision de la localisation pour GNSS s'explique donc par l'utilisation combinée

de ces multiples constellations de satellites. Ces constellations sont en constantes améliorations depuis leur création. Il est donc à prévoir que la performance de tels systèmes n'ira qu'en augmentant.



$$p_u^1 = \sqrt{(X_1 - U_x)^2 + (Y_1 - U_y)^2 + (Z_1 - U_z)^2} + c(dT_u)$$

$$p_u^2 = \sqrt{(X_2 - U_x)^2 + (Y_2 - U_y)^2 + (Z_2 - U_z)^2} + c(dT_u)$$

$$p_u^3 = \sqrt{(X_3 - U_x)^2 + (Y_3 - U_y)^2 + (Z_3 - U_z)^2} + c(dT_u)$$

$$p_u^4 = \sqrt{(X_4 - U_x)^2 + (Y_4 - U_y)^2 + (Z_4 - U_z)^2} + c(dT_u)$$

FIGURE 1.1 – Exemple de système GNSS et du système d'équations généré. Le symbole  $dT_u$  est l'erreur d'horloge entre le satellite et le récepteur. Le symbole  $c$  est pour la vitesse de la lumière. Figure tirée de [3].

### 1.3 Source d'erreurs des systèmes GNSS

Les systèmes GNSS sont susceptibles à plusieurs sources d'erreurs. Ces sources d'erreurs peuvent venir autant du système lui-même que de l'environnement. C'est des erreurs

qui sont souvent corrigibles par un système externe, tel que le Wide Area Augmentation System (WAAS) ou le RTK, tous deux décrits à la section 1.4. Ces sources d'erreurs sont détaillées plus bas.

### **1.3.1 Vision réduite du ciel**

Avant tout, pour qu'un système de localisation GNSS soit capable de donner une position, le récepteur doit avoir une ligne de vue directe avec au moins quatre satellites. Aussi, plus le récepteur verra de satellites, plus le positionnement pourra être précis. Surtout, si le récepteur ne voit pas beaucoup de satellites, le positionnement pourrait tout simplement ne pas se faire. Cette visibilité dépend à la fois de la position des satellites dans le ciel (éphémérides), mais aussi des obstacles présents dans l'environnement. La couverture satellitaire étant généralement assez uniforme, l'environnement dictera le plus ce type de bruit. Par exemple, des endroits présentant de nombreux édifices en hauteur, tels que les villes, souffriront davantage de ce problème que des environnements ouverts comme les champs. Dans le chapitre 5 montre une bonne idée de l'effet de cette source d'erreur.

### **1.3.2 Horloge de satellite et récepteur**

Le calcul de positionnement nécessite une mesure précise des temps de vol des signaux. Ces temps de vols sont calculés à partir des signaux d'horloges internes, autant du côté des satellites que des récepteurs. Même si les satellites sont munis d'horloges atomiques réputées très précises, ces dernières peuvent quand même dériver dans le temps. De même, le récepteur souffrira d'incertitude quant à la mesure de l'arrivée de ces signaux.

### **1.3.3 Erreur d'orbite**

Comme le calcul de positionnement utilise des satellites, le récepteur a besoin d'avoir une bonne précision sur l'orbite et la vitesse de ces derniers. Ces deux choses n'étant pas constantes, un certain niveau d'incertitude sera induit par le décalage entre la

position estimée et réelle des satellites. Une bonne partie de cette incertitude est due aux éphémérides imprécises et à l'erreur d'horloge expliquée plus haut.

### 1.3.4 Réflexion de signal

Dans le meilleur des cas, le signal du satellite arrive directement au récepteur. Malheureusement, dans des contextes urbains avec de grands bâtiments vitrés, le signal d'un satellite peut être réfléchi par un bâtiment, et ensuite reçu par le récepteur. Ceci cause une erreur sur le calcul de distance entre le satellite et le récepteur, car la trajectoire empruntée par le signal a été allongée. Cette erreur est souvent minimisée avec une bonne physionomie de l'antenne, afin d'atténuer les signaux autres que ceux provenant directement au-dessus de l'antenne. En effet, il est fort probable que les signaux provenant du bas de l'antenne ont été réfléchis par une surface. De plus, il est possible de filtrer les satellites en examinant leur éphéméride. En acceptant les signaux où le satellite est à au moins 15 degrés au-dessus de l'horizon, on diminue le risque d'incorporer des mesures associées à des signaux qui ont probablement souffert de réflexions.

### 1.3.5 Délai ionosphérique

Lorsque les signaux électromagnétiques passent au travers de l'ionosphère (50-1000 km d'altitude), ceux-ci vont être perturbés par des ralentissements et des déviations. Ces perturbations sont reliées aux propriétés électriques de l'ionosphère, qui fluctuent dans le temps. De plus, les signaux des satellites sont de plus en plus affectés lorsque le satellite est proche de l'horizon. Pour faire une estimation de cette erreur, une station de base (soit grâce au WAAS ou DGNS) envoie des messages de correction aux récepteurs environnants. Le récepteur avec l'estimation de cette erreur peut corriger cette source d'erreurs, à un certain degré. Par contre, même avec les messages de corrections, il est possible de corriger seulement jusqu'à approximativement 75% de l'erreur [4].

## 1.4 Systèmes de correction d'erreurs GNSS

Nous présentons ici deux systèmes de correction d'erreur de positionnement pour les systèmes GNSS. Le premier, le système WAAS, possède une précision plus limitée, mais mérite quand même sa description à titre indicatif. Le deuxième système, dit RTK, est nettement plus précis, et fera l'objet d'une étude plus approfondie. Il est aussi le système que nous avons choisi pour conduire des expériences sur ses performances en positionnement. Ces deux systèmes sont décrits plus bas.

### 1.4.1 Wide Area Augmentation System (WAAS)

Depuis 2003, il existe un système de correction du GNSS, le WAAS, basé sur le concept de *diffenrential GPS*. Le WAAS couvre la totalité de l'Amérique du Nord (voir Figure 1.2) pour la correction de positionnement GNSS, via 38 stations fixes réparties sur ce territoire. Il n'est donc pas disponible pour d'autres régions. Ces stations ont comme objectif d'envoyer des messages de corrections aux modules GNSS mobiles utilisés par des particuliers. Ces messages contiennent des correctifs (différences) à apporter aux positions estimées par les stations mobiles, afin de compenser les erreurs de délai ionosphérique, d'orbite et d'horloge de satellites. Dans le dernier rapport de performance [5], on y dénote une précision de moins d'un mètre latéralement, et 1.5 mètre verticalement. Cette précision peut cependant être jugée insuffisante pour faire un suivi serré du positionnement des équipements, particulièrement dans des endroits complexes comme les cours à bois. Seuls certains appareils GNSS ont accès à ce réseau, étant donné que la connexion au WAAS demande un abonnement pour recevoir les signaux de correction.

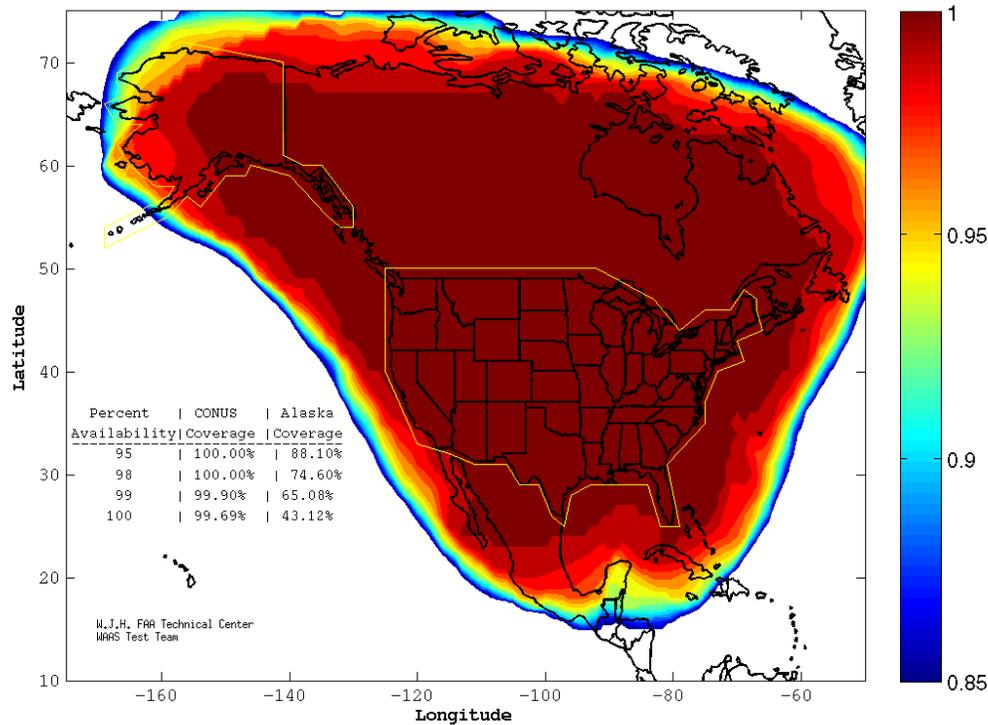


FIGURE 1.2 – Couverture courante du système WAAS. Figure tirée de [6]. La barre de couleur représente la qualité du signal.

### 1.4.2 Real-Time Kinematic (RTK)

La méthode de positionnement RTK est différente de la méthode de calcul GNSS, tout en s'apparentant à la méthode de correction du système WAAS. En effet, la méthode RTK utilise aussi une base de référence fixe pour aider à la correction du positionnement de la solution mobile (telle qu'illustré à la Figure 1.3), tout comme le système WAAS. Cette station fixe est utilisée pour calculer les messages de corrections à apporter, faisant d'elle une forme de DGNSS. Certaines solutions GNSS peuvent se connecter au WAAS et donc utiliser les stations de référence du WAAS pour recevoir les messages de correction. Contrairement au GNSS classique, le système RTK utilise la phase de l'onde porteuse du signal du satellite en plus du contenu du message. L'estimation de la phase de la porteuse est faite à la fois sur la base de référence et le module mobile. Les distances récepteurs-satellites sont calculées en multipliant la longueur d'onde par le nombre de fois qu'il y a eu un cycle, et ensuite est additionné la différence de phase.

Or, seule la différence de phase peut être directement mesurée par les récepteurs. Le nombre de cycles est, quant à lui, déterminé indirectement en réalisant un processus appelé *Ambiguity resolution*. Notez que cette résolution d'ambiguïté est faite pour un satellite en particulier. Advenant de la perte de visibilité de ce satellite et l'arrivée en visibilité d'un nouveau satellite, le système doit résoudre à nouveau cette ambiguïté. Ainsi, la performance d'un système RTK pourra être affectée par des variations rapides de visibilité de satellites. Lorsqu'un système a une excellente confiance en cette estimation, les estimations de positions sont dites *fixed*. Par opposition, si cette estimation d'ambiguïté est moins fiable, elle portera le non de *float*<sup>1</sup>.

Comme la longueur d'onde de la porteuse est de 19 cm, la mesure de la phase permet au système d'obtenir des informations de position en deçà du décimètre, nettement mieux que la précision du système WAAS.

La base de référence envoie à tous les récepteurs mobiles des messages de corrections contenant des informations tirées des différents signaux. Dans ces messages, il peut y avoir les informations suivantes :

1. des observations GNSS (signaux satellites observés) ;
2. les coordonnées de la station fixe ;
3. les mesures de distance venant des signaux ;
4. la description de l'antenne de la station de référence ; et
5. l'éphéméride de satellites observés.

Ces récepteurs mobiles exploitent ces informations pour faire les corrections nécessaires sur leur position. Un système RTK est une solution très avantageuse, car elle permet d'aller chercher une précision qui était auparavant réservée au récepteur haut de gamme, mais à un prix maintenant nettement réduit [7].

---

1. Ce terme fait référence au calcul en points flottants. Dans ce mode *float*, la valeur normalement entière du nombre de cycles est remplacée par une estimation en points flottants.

### 1.4.3 Correction RTK en différé via post-traitement

Il est possible d'utiliser les données venant de la station de base et du module mobile pour faire une correction, en différé, c'est-à-dire après la prise de données. Ceci est notamment pratique lorsqu'il n'existe pas de canal de communication en temps réel entre la station de base et les récepteurs mobiles. Dans un scénario industriel, ceci permettrait de faire un suivi hors-ligne (*offline*) des équipements forestiers. Une démonstration de cette capacité de traitement en différé est présentée dans la sous-section 4.1.1.

Un exemple de format de données utilisé pour représenter l'information GNSS est le protocole National Marine Electronics Association (NMEA). Ce protocole tient son nom de l'organisation qui l'a créé. Ce protocole contient les diverses informations qui sont utilisées pour voir une trajectoire ou pour faire la correction. Un exemple de phrase est :

```
$GNGGA,182237.80,4646.7954377,N,07116.4206719,  
W,1,07,0.0,91.765,M,-28.206,M,0.0,*51
```

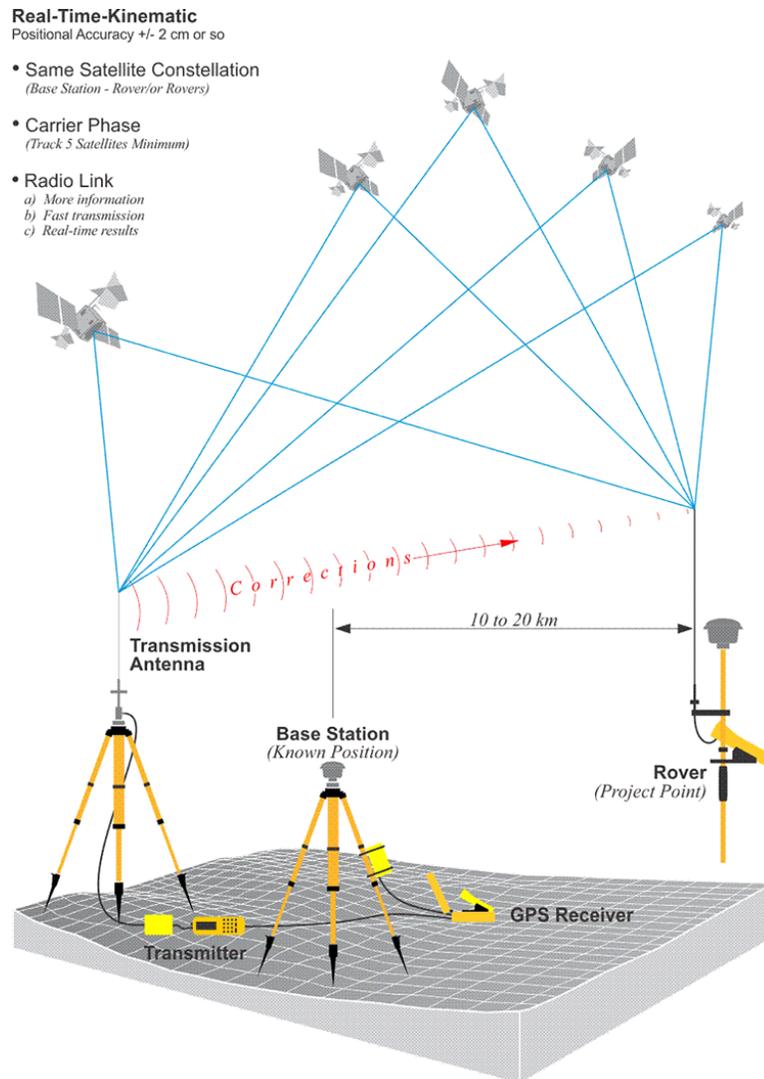


FIGURE 1.3 – Illustration de la méthode RTK, tiré de [8]. On peut y voir la station de référence envoyer un message de correction au module mobile, ici appelé Rover, via un signal radio. Notez que le nombre minimal de satellite pour lequel le suivi de la phase est nécessaire est de cinq, soit un de plus que pour les systèmes WAAS. La précision de  $\pm 2$  cm s’explique en partie par la courte longueur d’onde de la porteuse (19 cm).

En regardant le manuel SIRF TECHNOLOGY [9], nous pouvons y voir que le code GGA indique une position fixe déterminée lors de la trajectoire. La valeur 182237.80 est le temps en Temps Universel Coordonné (UTC). La valeur 4646.7954377 est la latitude, avec N pour nord. La longitude est donnée par 07116.4206719, avec W pour ouest. La valeur 1 détermine le type de position. Dans cet exemple, il s’agit d’un positionnement GNSS sans correction venant d’une station de référence. Cette valeur

aurait été de 2 si elle était venue d'un positionnement DGNS. Ensuite 07 indique le nombre de satellites utilisés pour l'estimation de positionnement. La phrase continue avec l'altitude, des unités et l'identifiant de station de référence. Il y a des phrases pour le statut des signaux (MSS), pour l'information sur l'azimut (position dans le ciel) des satellites (GSV), pour la vitesse par rapport au sol (VTG), etc. Ces phrases montrent les données qui sont envoyées à la station mobile par la station de référence pour faire les corrections et celle que l'utilisateur d'un système GNSS peut extraire pour faire une visualisation. Une série de ces messages est montrée à la Figure 1.4.

```

$GNCGA,182237.80,4646.7954377,N,07116.4206719,W,1,07,0.0,91.765,M,-28.206,M,0.0,*51
$GPGSA,A,3,13,15,29,30,51,,,,,,,,,17.7,16.9,5.2*30
$SGLGSA,A,3,86,,,,,,,,,0.0,0.0,0.0*20
$GAGSA,A,3,09,,,,,,,,,0.0,0.0,0.0*2A
$GPGSV,2,1,05,13,67,080,35,15,64,206,44,29,30,220,43,30,18,047,41,1*6E
$SGLGSV,2,2,05,51,26,225,38,,,,,,,,,1*5F
$SGLGSV,1,1,01,86,58,210,41,,,,,,,,,1*4C
$GAGSV,1,1,01,09,69,243,41,,,,,,,,,7*44
$GNGST,182237.80,0.031,,,39.910,42.534,17.808*50
$GNVTG,,T,,M,0.68,N,1.27,K,A*37

```

FIGURE 1.4 – Exemple de données en format NMEA.

Par contre, pour faire la correction en temps réel sur le module mobile, un autre protocole est utilisé. Le protocole utilisé pour la correction en temps réel est le RTCM3 [10]. Ce protocole compresse les données nécessaires pour faire la correction, permettant d'avoir des données plus légères au transfert. Ce protocole contient toutes les informations énumérées dans la section sur le RTK (sous-section 1.4.2).

#### 1.4.4 Manufacturier de produits RTK

Dans les années 1990, lorsque les récepteurs GNSS ont commencé à réduire de taille, ceux-ci se vendaient autour de 10 000 \$, en fonction de la qualité [7]. Avec l'arrivée des *software-defined radio* et des circuits programmables Field-Programmable Gate Array (FPGA), le coût de développement et de fabrication des appareils RTK a grandement chuté. Ainsi, de nombreux manufacturiers offrent maintenant des solutions abordables pour faire de la localisation avec des systèmes RTK.

Plusieurs manufacturiers offrent des solutions complètes. Ces solutions comprennent

Manufacturier	Nom du modèle	Prix
Emlid	REACH RS+ et REACH M+	\$1 000
Trimble	Trimble R2 RTK Rover	\$13 859
Swift Navig.	Piksi Multi Evaluation Kit	\$1 995
Hemisphere GNSS	S320 GNSS RTK Base And Rover	\$8 890
NovAtel	OEM7700 (Rover)	<\$1 000

TABLE 1.1 – Liste de différentes solutions RTK commerciales.

généralement un récepteur pour agir en tant que station de référence et un autre qui agit comme récepteur mobile.. Une liste de manufacturiers avec des exemples de modèle, prix et lien est disponible dans le Tableau 1.1. Pour notre étude, nous avons choisi la solution de la compagnie *Emlid*. Ce manufacturier offre, au prix d'environ 1 000 USD, une solution complète RTK, illustrée dans la Figure 1.5. Celle-ci comprend notamment une station de référence (*Reach RS*) certifiée IP67 [11] avec environ 5 Go de capacité de stockage non volatil pour les données brutes, pouvant être utilisée par exemple pour stocker les données nécessaires au traitement en différé (voir sous-section 1.4.3). La certification IP67 est une classification du degré de protection contre les poussières et les liquides. Un module mobile (*Reach M+*) contient toutes les fonctionnalités logicielles de la base de référence, mais sans la certification IP67 et avec une capacité de stockage non volatil limité à 2 Go.



FIGURE 1.5 – Solution complète RTK. Dans la photo, on peut voir les deux principaux éléments. Le plus gros boîtier est le *Reach RS*, qui est la station de référence. Le plus petit boîtier est le *Reach M+*, qui est le module mobile.

# Chapitre 2

## Cartographie 3D via lidar

Ce chapitre se concentre sur les données lidar, la théorie derrière la génération de cartes de nuage de points 3D et toute l'information nécessaire pour accomplir l'objectif réalisé dans le chapitre 5. Ces cartes sont très importantes pour comprendre la géométrie de l'environnement, qui pourrait influencer la réception de signaux des capteurs GNSS. Nous parlerons aussi de l'outil pour faire la réalisation des cartes et les difficultés rencontrées dans les divers environnements tout en montrant des résultats.

Une fois les données GNSS et lidar récupérées, la prochaine étape était de faire une carte 3D pour avoir une idée de l'environnement. Lors de la construction d'une carte 3D, il y a aussi une analyse préliminaire pour générer toute l'information nécessaire à l'analyse. Cette analyse inclut plusieurs filtres qui vont permettre le recalage de nuage de points et de faire l'analyse de géométrie nécessaire pour le chapitre 5. Ces informations incluent les normales de surfaces et la distance moyenne des  $k$  voisins les plus proches.

### 2.1 Données

Les capteurs lidar produisent des données nommées *nuages de points* en effectuant des balayages de leur environnement. Les nuages de points sont un ensemble de points 3D ayant comme origine le capteur. Ils sont produits à un intervalle de temps qui est propre au capteur, en projetant un laser qui va se refléter sur les objets autour du

capteur. Cette réflexion permet d'estimer la distance entre le capteur et l'objet qui a réfléchi le laser. En sachant la distance et l'orientation du laser envoyé, il est donc possible de situer un point dans un espace 3D. En faisant cela pour plusieurs lasers de la même source, nous pouvons assembler un nuage de points qui nous donne une idée de la géométrie de notre environnement. La Figure 2.1 montre un exemple de données lidar qui est utilisé pour l'analyse de l'environnement. Cette structure de données va être utilisée pour faire la carte et nous pouvons extraire les informations géométriques nécessaires pour estimer l'effet de l'environnement sur les signaux satellites.

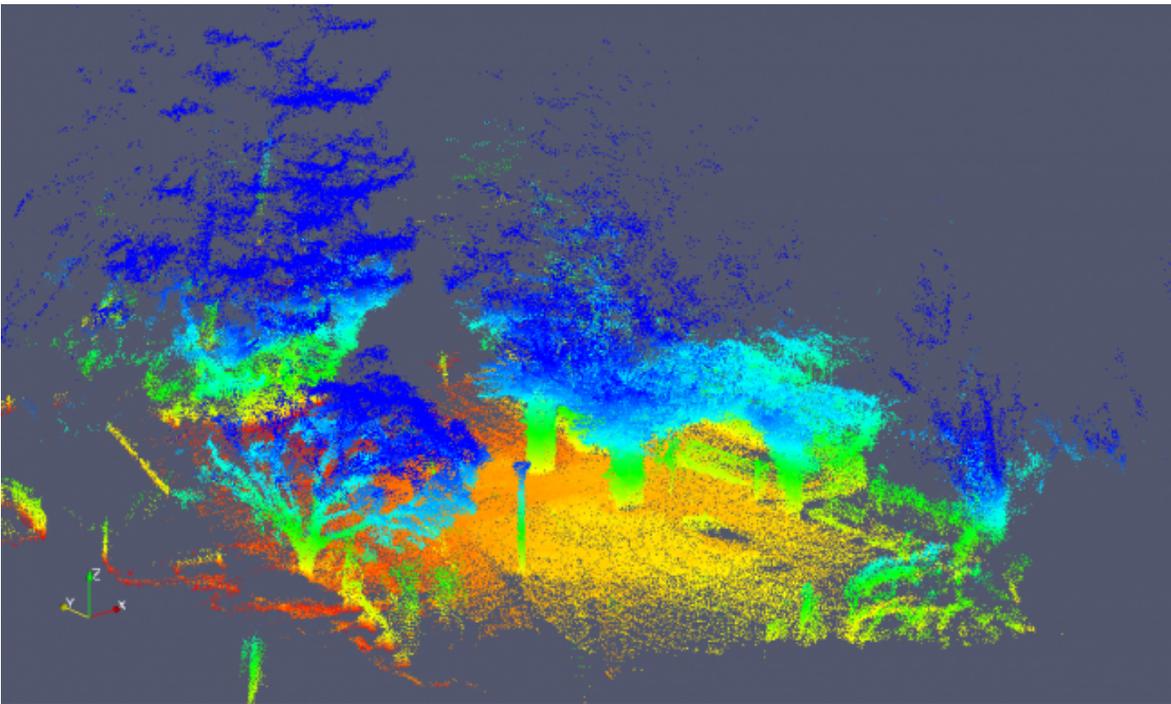


FIGURE 2.1 – Exemple de données lidar avec une scène venant d'un capteur mis dans un contexte extérieur hivernal [12]. Dans la photo, on peut voir une carte obtenue en passant un lidar dans un sentier entouré d'arbres.

## 2.2 Théorie

Pour faire la construction de la carte, un algorithme nommé Iterative Closest Point (ICP) a été utilisé. Le but principal de cet algorithme est de trouver une rotation et translation entre deux nuages de points qui vont permettre de caler l'un des nuages sur l'autre. Plusieurs variantes ont été implémentées au fil des années [13]. Premièrement,

il y a le *point-to-point* qui regroupe les points ensemble, en suivant une méthode pré-déterminée. Ces méthodes comprennent : prendre les points les plus proches, prendre les points avec les normales les plus similaires, etc. Par contre, lors de cette maîtrise la variante utilisée a été celle du *point-to-plane* [14] [15]. La différence avec le *point-to-point* est que l'erreur vient de la distance de chaque point avec une tangente venant d'un groupe de point dans l'autre nuage. Un exemple de cette variante est illustré dans la Figure 2.2. On peut y voir que les points de la source sont jumelés à des tangentes venant des points de la destination. Le *point-to-plane* a été démontré comme ayant un meilleur taux de succès pour la convergence sur une solution [16].

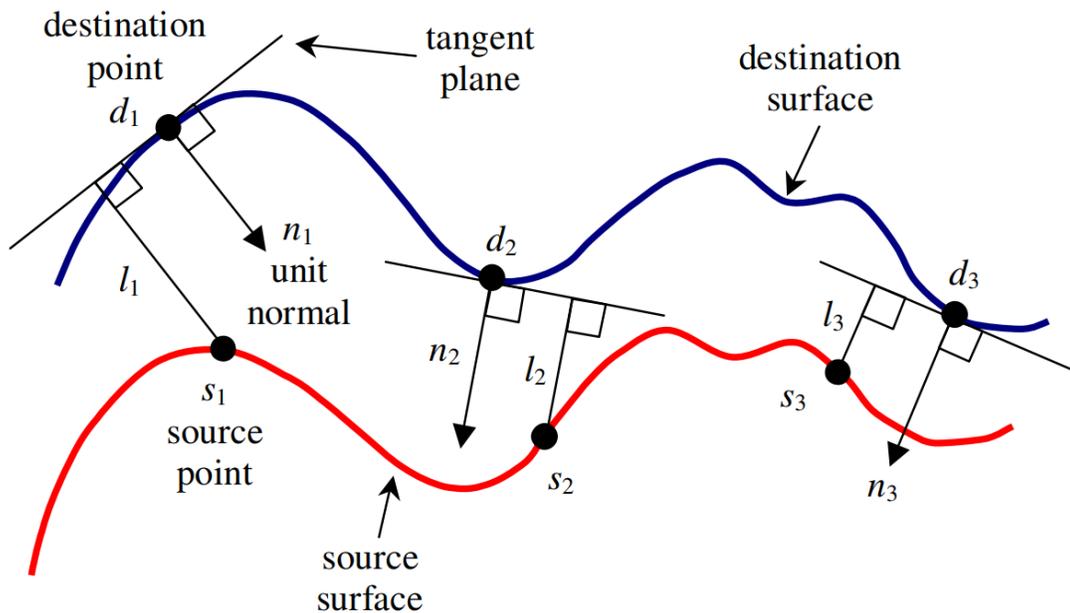


FIGURE 2.2 – Exemple d'erreur entre deux surfaces pour la variante de ICP du *point-to-plane*. [17]

## 2.3 Outil de génération

L'outil principal utilisé pour la réalisation des cartes est un algorithme de cartographie développé dans le laboratoire *ASL* de l'*ETH* à Zurich. Cet algorithme utilise *libpoint-matcher*, une librairie d'ICP, et *libnabo*, une librairie permettant de trouver les  $k$  plus proches voisins. Ces deux librairies ont aussi été développées à l'*ASL ETH*. L'accumu-

lation des nuages de points, assemblés par l'algorithme, forme une carte de l'environnement. Cet algorithme de cartographie est un outil supporté dans Robot Operating System (ROS), alors divers outils supplémentaires sont fournis pour le compléter. Par exemple, il est possible d'utiliser un rosbag pour alimenter les données à l'algorithme. Un rosbag est un outil utilisé pour faire l'enregistrement de données avec une étampe de temps. Cette étampe nous permet donc ensuite de faire dérouler les données comme si elles étaient réelles.

L'algorithme de cartographie prend deux nuages de points et fait les étapes suivantes. Premièrement, il applique des filtres sur les nuages de points, individuellement. Comme l'algorithme utilise la minimisation d'erreur *point-to-plane*, certains filtres sont nécessaires. Les filtres utilisés seront expliqués en plus grand détail dans les sections 2.3.1, 2.3.2 et 2.3.3. Deuxièmement, l'algorithme de cartographie va utiliser une méthode de regroupement des points lidar. Dans notre cas, nous avons utilisé la méthode de regroupement de bases qui était celle des  $k$  plus proches voisins. Troisièmement, l'algorithme applique un algorithme pour la réduction d'erreur. Comme mentionné plus tôt, l'algorithme pour la réduction d'erreur utilisé est celui qui réduit l'erreur *point-to-plane* en appliquant une translation et rotation sur le nuage de points en entrée. L'algorithme de cartographie va donc répéter les étapes deux et trois jusqu'à ce que l'erreur soit assez petite, que le nombre d'itérations soit écoulé ou toute autre condition mise par l'utilisateur. Dans la Figure 2.3, on peut voir un exemple de la suite des étapes.

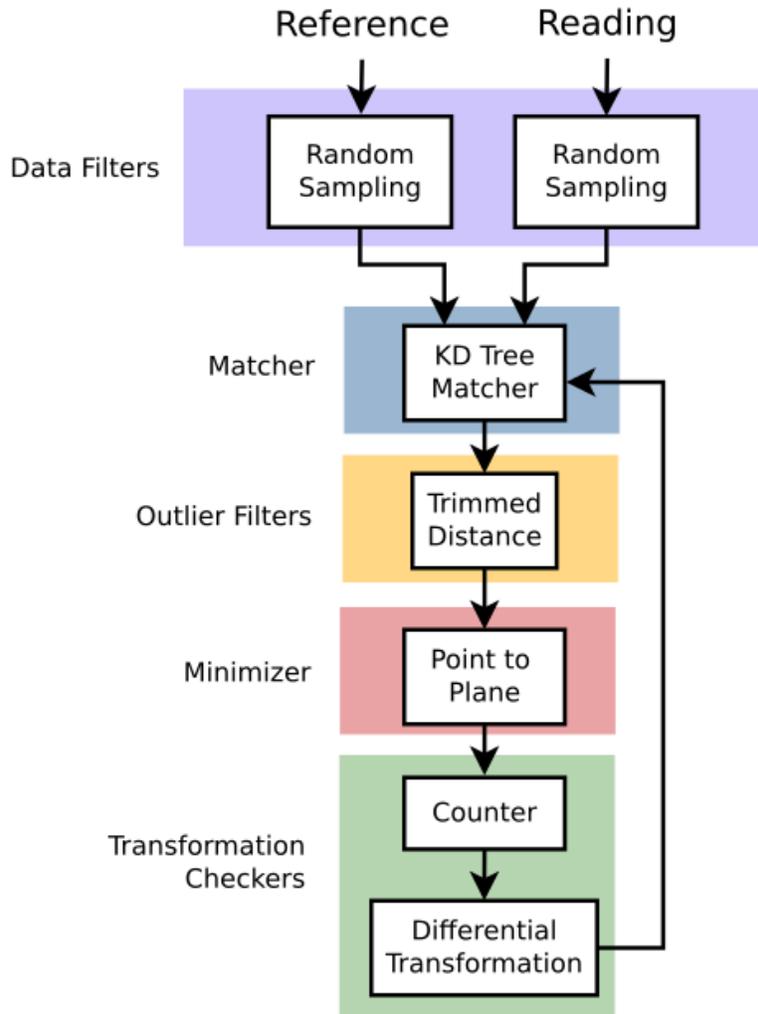


FIGURE 2.3 – Étapes de l’algorithme de cartographie [18]. La première étape est de filtrer les nuages de points pour réduire le temps de calcul. Ensuite les points des nuages sont jumelés avec les points les plus près de l’autre nuage. Un filtre de données aberrantes est appliqué. Les nuages sont ensuite "glissés" sur les plans pour essayer de trouver la meilleure transformation. Ce glissement se fait plusieurs fois. Cette opération s’arrête après un certain nombre d’itérations. Il y a aussi un vérificateur de transformation pour déterminer si un mouvement est meilleur qu’un autre.

### 2.3.1 Octree Grid Filter

Nos analyses sur la géométrie et l’obstruction de l’environnement se produisent autant dans un contexte urbain que dans une forêt dense. En raison de sa structure, une forêt génère beaucoup plus de points qu’une surface relativement lisse de bâtiment. C’est pourquoi nous devons utiliser un filtre qui va permettre aux deux environnements

d'avoir une densité de points similaire. Ce filtre utilise un octree [19] pour faire la représentation spatiale du nuage de points. Un octree est une structure de données qui permet de représenter les points en arbre à huit branches où les points sont sur les branches. Un noeud dans un arbre d'un octree est un cube d'une dimension donnée par la grosseur du nuage de points. Le cube est ensuite divisé en huit autres cubes qui seront ses branches, et cela, jusqu'à ce que les cubes atteignent une grosseur minimale. Cette représentation est ensuite utilisée pour faire un sous-échantillonnage de chaque branche de points. Ce filtre présente plusieurs méthodes pour choisir les points à décimer. Dans notre cas, le filtre prend tout simplement le premier point qu'il trouve dans la branche. La grosseur minimale des boîtes conservées pour la génération de cartes est de 0.3 mètre.

### 2.3.2 Surface Normal Filter

Comme l'algorithme de cartographie utilise la normale de surface des points pour déterminer l'erreur entre les nuages de points et que nous utilisons cette information pour faire l'analyse des nuages de points, nous avons donc utilisé un filtre pour calculer ces normales de surface. Ce filtre utilise les points voisins pour déterminer la normale de surface d'un point. Ce filtre permet ensuite de conserver l'information sur la carte. De plus, il nous permet de décider le nombre de voisins à considérer dans le calcul. Dans notre cas, le nombre de points est de 15 points. Ce chiffre a été déterminé empiriquement, en faisant des observations sur les nuages de points sortant du filtre. On voulait s'assurer d'avoir assez de points pour avoir la densité des branches de la forêt, mais pas trop pour qu'un plan devienne une sphère. La Figure 2.4 montre la sortie de ce filtre, avec des flèches sur les points indiquant la direction de la normale de surface.

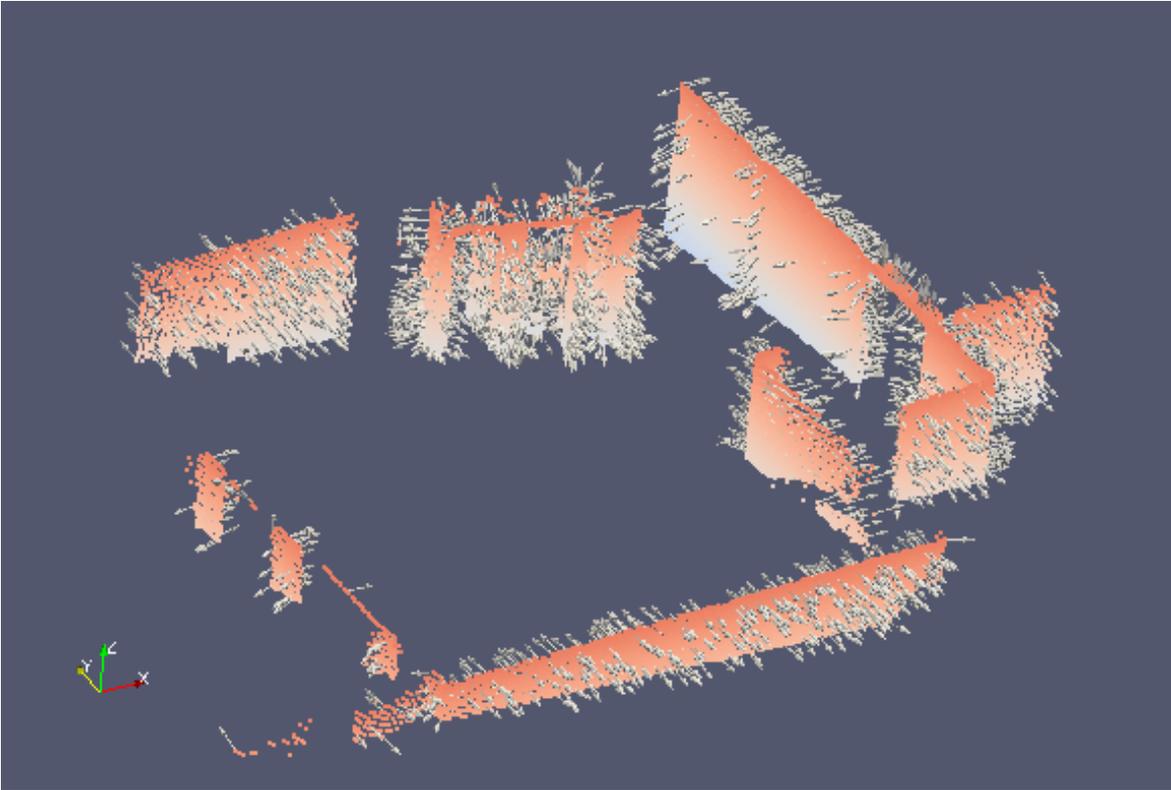


FIGURE 2.4 – Normale de surfaces des points venant du filtre en considérant les 20 plus proches voisins [18].

### 2.3.3 Observation Direction Filter

Comme le dernier filtre ne donne pas d'orientation aux normales de surface, il est difficile de les utiliser pour analyser la géométrie des points. Heureusement, il y a un filtre qui accomplit précisément cette tâche. Ce filtre change les directions des normales, afin que ceux-ci soient orientés dans la même direction par rapport au capteur. Par exemple, dans notre cas, nous avons déterminé que les normales de surface devraient être orientées vers l'intérieur du balayage, donc orienté vers le capteur. L'importance de cette réorientation est pour l'extraction du sol dans le papier du chapitre 5. Comme il est important de savoir quels points qui ont une normale de surface qui pointe vers le ciel. Sinon certains points du sol pourraient avoir une normale qui pointe vers le centre de la Terre, et cela ne serait automatiquement pas considéré comme le sol. La Figure 2.5 montre la sortie de ce filtre. Comme on peut le voir, les flèches pointent vers l'intérieur du balayage, au lieu de directions aléatoires, comme à la Figure 2.4.

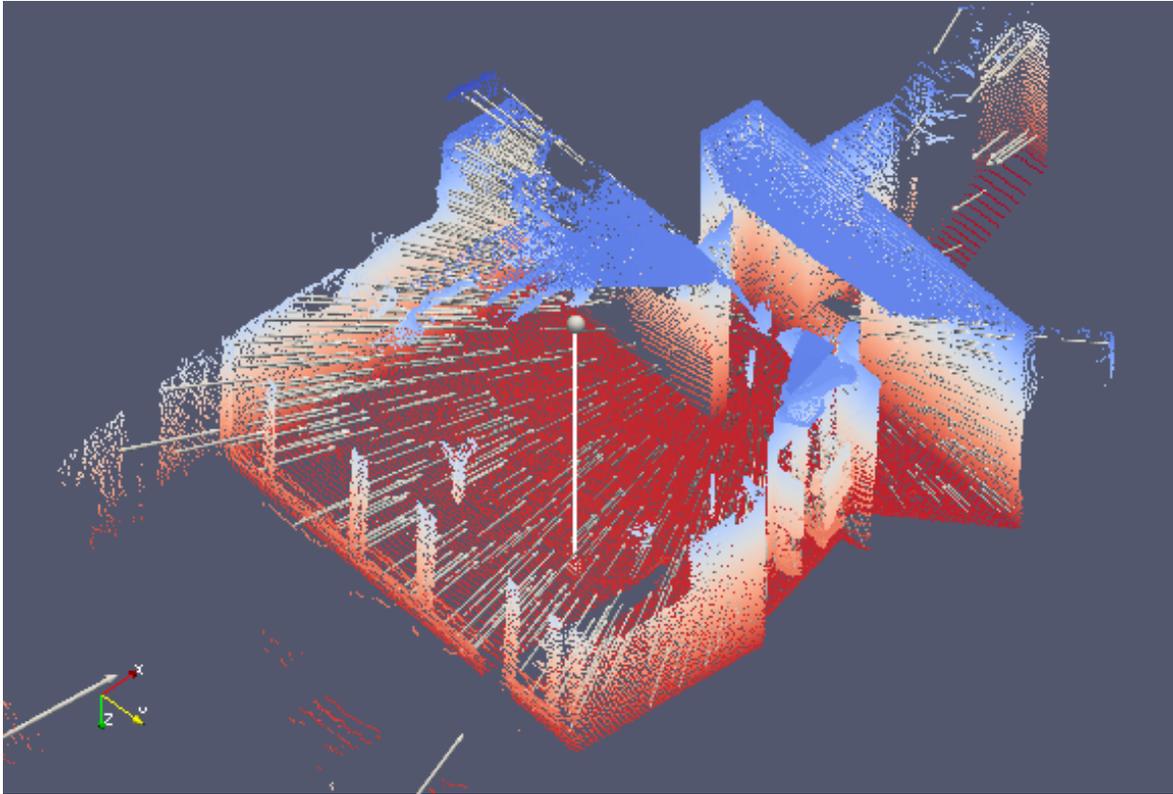


FIGURE 2.5 – Réorientation des normales de surface [18]. On peut voir que les flèches sont maintenant toutes orientées vers un point central, correspondant à la position du capteur. On peut voir la différence avec la Figure 2.4.

# Chapitre 3

## Plate-forme de saisie de données

Comme la récolte de données est un élément important des approches *Industries 4.0* et de la robotique mobile en générale, il est important d’avoir une plate-forme informatique permettant de faire une saisie automatisée des informations pertinentes. Cette plate-forme est pour accomplir les objectifs de ce mémoire sur les applications, la récolte de données et ensuite faire l’analyse des données des receveurs GNSS.

Dans notre cas où l’environnement de l’industrie forestière est extérieur et est sujet à des températures relativement extrêmes selon les saisons, il faut que cette plate-forme informatique soit robuste. Elle doit aussi idéalement être facile à installer sur tout véhicule. Cette plate-forme doit être mise sur un véhicule aussi sans nécessairement devoir se connecter au système électronique interne du véhicule, donc elle doit être indépendante. Par exemple, lors des expérimentations du chapitre 5, nous avons dû installer cette plate-forme sur une motoneige. La plate-forme finale peut être vue dans la section 3.4. Comme les données peuvent être collectées dans des régions éloignées, cette plate-forme gagne à être *cloud-enabled*. En effet, une fois les données téléchargées sur un serveur par la plate-forme via un réseau sans fil, un utilisateur distant pourrait se connecter à ce même cloud pour prendre les données dont il a besoin pour faire son analyse.

En utilisant ROS, un outil logiciel répandu dans la communauté robotique, il est possible de faire la gestion de toutes sortes de types de capteurs, allant de GPS, de Lidars

mobiles, jusqu'à l'odométrie du véhicule. Ensuite, l'enregistrement se fait grâce à un outil standard dénommé *rosbag* [20]. Cet outil permet de faire l'enregistrement des données qui sont gérées par ROS et ensuite de les visualiser comme si nous recevions les données en temps réel. Pour assurer le transfert possible des données sans connexion, nous avons ajouté un script de gestion de données qui se sert de deux clés USB amovibles, permettant le transfert manuel des données. Ces clés peuvent être retirées et ajoutées à tout moment. Comme cette plate-forme ne sera pas toujours directement accessible pour faire des modifications dans ses fonctions, une solution de connexions à distance à la plate-forme a été choisie. Cette fonction de contrôle à distance nous permet de faire des modifications à tout moment lorsque la plate-forme est allumée, même lors de son enregistrement de données. Pour confirmer le fonctionnement de toutes ces fonctionnalités, des tests d'endurances ont été faits. La première expérience effectuée a été de faire fonctionner la plate-forme dans un véhicule en mouvement. À cet effet, nous avons installé la plate-forme dans une voiture et l'avons conduit au travers du campus de l'Université Laval. Nous avons donc été capables de voir les données sur un serveur, dans l'une des clés USB et nous avons pu tester la connexion à distance à la plate-forme. Deuxièmement, nous avons fait un test d'endurance sur plusieurs semaines, pour valider le bon fonctionnement des scripts de gestion. Cette expérience était en partie conçue pour s'assurer qu'il n'avait pas un débordement de mémoire dans un serveur, les clés USB ou la mémoire locale.

### 3.1 Caractéristique matérielle de l'ordinateur

Plusieurs caractéristiques physiques de la plate-forme étaient nécessaires pour aller faire la récolte de données dans des environnements pouvant être considérés comme difficile :

1. température d'opération à -20 degrés Celsius pour le fonctionnement en hiver ;
2. démarrage avec le véhicule ;
3. délai modifiable (minimum de 5 minutes) à la fermeture pour que la plate-forme reste allumée après la fermeture du véhicule ;

4. CPU avec référence de performance de maximum 3000 selon <https://www.cpubenchmark.net/>;
5. au moins un Solid-state drive (SSD) de 120 GB facilement remplaçable ;
6. capacité sans-fil, Wi-Fi et Bluetooth ;
7. minimum de 4 ports USB pour la connexion à plusieurs capteurs ; et
8. minimum de 2 Prises General-purpose input/output (GPIO), pour des entrées/sorties discrètes.

## 3.2 Solution logicielle

Pour faire la bonne récolte de données, plusieurs solutions logicielles ont dû être utilisées pour satisfaire les diverses fonctions de l'ordinateur :

1. connexion à distance : TeamViewer, connexion ssh ;
2. mémoire sur un serveur : Google Cloud ou autre serveur ;
3. gestion de données locales et sur le cloud : Python ; et
4. enregistrement des données : ROS et rosbag.

Bien que l'ordinateur ait été complété, nous n'avons pas été en mesure de le déployer à temps sur un site comme une cour à bois. Cependant, nous avons pu le mettre en marche pour la récolte de données pour l'article présenté au chapitre 5. Tous les aspects logiciels ont été utilisés lors de la récolte des chapitres 1 et 4 pour assurer l'intégrité et l'efficacité de la récolte de données.

## 3.3 Capteurs

Plusieurs capteurs ont été ajoutés sur la plate-forme finale, énumérés ici-bas :

1. GNSS RTK ;
2. Lidar ; et

### 3. Centrale inertielle.

Ces capteurs ont été déterminés comme étant nécessaires, afin de permettre l'analyse de données GNSS et pour reconstruire l'environnement autour du récepteur. La reconstruction de l'environnement est expliquée dans le chapitre 2 et cette reconstruction de l'environnement est nécessaire pour comprendre la géométrie de l'environnement pour finalement estimer l'influence sur les récepteurs GNSS. Cette estimation est réalisée dans le chapitre 5. Dans le chapitre 5, il y aura une explication et des résultats des diverses méthodes utilisées pour faire cette analyse.

## 3.4 Plate-forme finale

Pour accomplir les expérimentations nécessaires à la réalisation du papier au chapitre 5, nous avons implémenté une plate-forme finale qui peut être vue à la Figure 3.1 et Figure 3.2. Dans la Figure 3.1, on peut voir la structure extérieure de la plate-forme. Les bâtons de métal servent à surélever les capteurs principaux. En effet, le récepteur GNSS doit préférablement être situé le plus haut possible, afin qu'il ne soit pas obstrué par quoi que ce soit. Ensuite, le Lidar est mis en angle pour la capture de la cime des arbres, pour assurer d'avoir le plus d'information sur l'obstruction à laquelle le récepteur GNSS sera exposé. Les éléments restants sont isolés de l'extérieur grâce à un boîtier *Pelican*. Ces boîtiers sont certifiés comme étant imperméables et résistants à la poussière. L'intérieur de ce boîtier est montré dans la Figure 3.2. L'alimentation électrique, qui consiste en une batterie et un convertisseur, est connectée aux divers éléments de la plate-forme, incluant l'ordinateur. Il y a une centrale inertielle située en arrière de la mousse supérieure. Une centrale inertielle est un capteur qui permet de mesurer l'accélération et la vitesse angulaire de la plate-forme. Cette information est très utile pour estimer le mouvement et donc la position de la plate-forme dans l'environnement. La position venant de cette centrale inertielle est finalement très utile pour la reconstruction de l'environnement, qui est expliqué dans le prochain chapitre.



FIGURE 3.1 – Plate-forme finale : structure extérieure. Le récepteur GNSS est positionné le plus haut pour réduire les interférences. Le lidar est mis en angle pour permettre de voir les obstacles pour le récepteur.



FIGURE 3.2 – Plate-forme finale : structure intérieure. La batterie rouge occupe une bonne partie de la boîte. On peut voir l'électronique et l'ordinateur à droite.

# Chapitre 4

## Résultats pour la validation du matériel

### 4.1 Résultats sur les données GNSS

Pour faire une première série de collecte de données RTK, un robot *Husky* de la compagnie *Clearpath* (voir Figure 4.1) [21] a été utilisé, avec la solution DGNSS de *Emlid*. Sur la photo, on peut voir le module GNSS dont l'antenne est située environ 60 cm au-dessus du reste du robot. La station de référence est placée sur un trépied, environ 90 cm au-dessus du sol. Elle est aussi placée, lors des expérimentations, à un endroit qui a une bonne ouverture au ciel. On peut voir la station de référence sur son trépied à la Figure 4.2



FIGURE 4.1 – Robot *Husky* utilisé pour la collecte de données.



FIGURE 4.2 – Station de référence utilisée pour la collecte de données.

#### 4.1.1 Expérimentation sur la reconstruction en différé de la trajectoire d'un véhicule

Tel que décrit à la sous-section 1.4.3, il peut arriver des situations pour lesquelles il n'est pas possible d'échanger en temps réel les messages de corrections entre la station de base et celle mobile. Par exemple, dans le chapitre 5, il y a plusieurs trajectoires où l'échange de messages de correction n'était pas possible. Cependant, la technologie DGNSS permet quand même d'intégrer ces informations *a posteriori*, à condition que ces informations (que nous appellerons *données brutes*) aient été enregistrées par les

deux stations. Nous avons exploré cette capacité du système à reconstruire des trajectoires précises en différé. Notamment, cette reconstruction nous a permis de tester les divers outils disponibles pour faire ce post-traitement. Celle-ci est faite grâce aux données brutes récoltées pendant un certain temps. Ces données sont en premier lieu sous le format UBX. UBX est un format de données similaire à NMEA pour ce qui est du contenu. Par contre, ce protocole n'est pas directement interprétable. Une fois que deux fichiers UBX ont été générés par la prise de données, il est possible d'utiliser les applications données par la librairie RTKLIB. La première application est RTKCONV qui nous permet de faire la conversion des données UBX en données d'observation et de navigation. Les fichiers d'observation contiennent les observations de satellites que le receveur a vues pendant la récolte de données. Le fichier de navigation contient les informations de chacun des satellites (éphéméride, paramètres de temps et la correction sur l'horloge) perçus pendant la récolte de données. Ensuite, avec l'application RTKPOST, le fichier d'observation de la station de référence et du module mobile et le fichier de navigation du module mobile, il est possible de faire le post-traitement. Cette étape de post-traitement va donc générer un fichier de position (.pos). Ce fichier contient toutes les positions du receveur corrigées par le post-traitement. Pour faire la comparaison de trajectoire avec et sans correction, un autre enregistrement de données de position a été fait sur le module mobile. Finalement, l'application RTKPLOT permet de faire la visualisation des données.

La Figure 4.3 montre la correction d'une trajectoire en post-traitement en utilisant des données brutes.

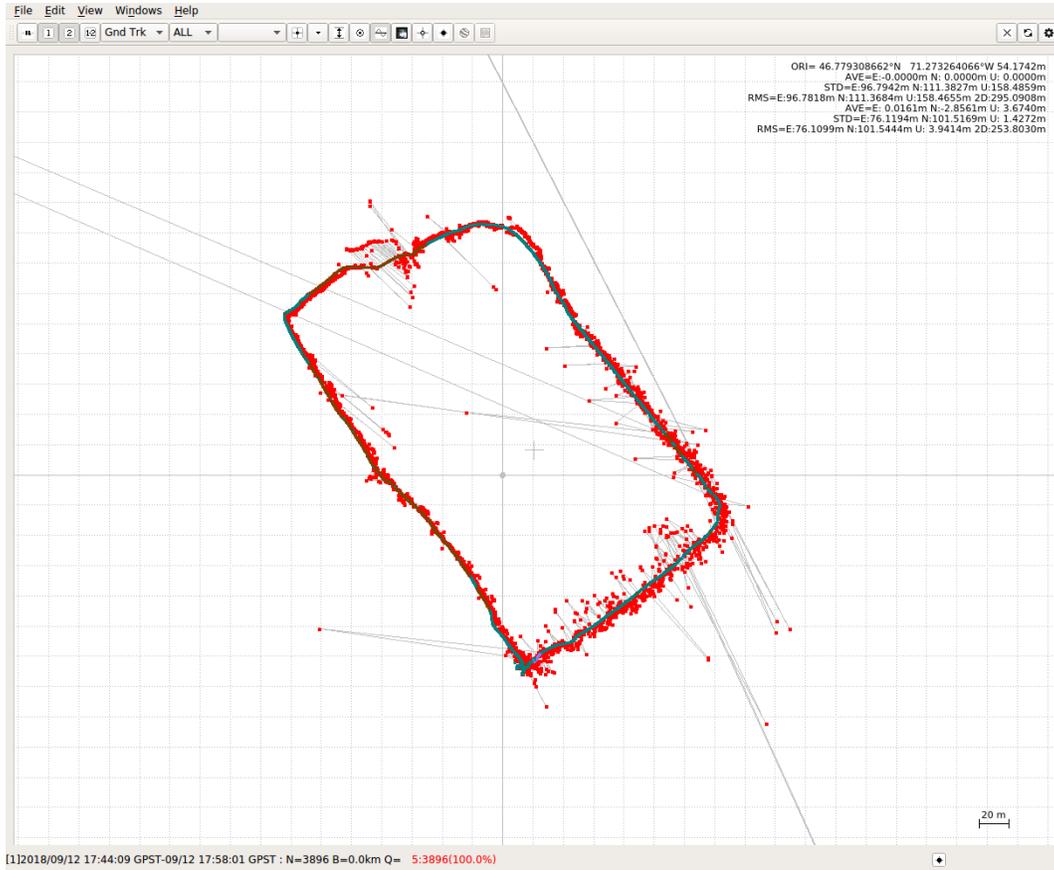


FIGURE 4.3 – Résultat de post-traitement sur des données brutes. Les points rouges sont les solutions GNSS non corrigées, les points bleus sont les points corrigés et les lignes grises sont la connexion entre les points.

#### 4.1.2 Cour à bois

Nous avons l'opportunité d'aller dans une cour à bois pour faire une prise de données avec ce véhicule Husky. Cette prise de données a servi à valider la performance de la solution DGNSS avec et sans la correction venant d'une base de référence, pour ce type d'environnement important pour l'industrie du bois. La Figure 4.4 montre une trajectoire simple avec et sans correction. Dans cette trajectoire, nous pouvons voir les endroits avec des erreurs de GNSS qui augmentent à certains endroits sur la trajectoire. Notamment, il y a augmentation d'erreurs lorsque le robot se rapproche du bâtiment en bas à droite de la figure. Cette augmentation peut aussi être observée dans les Figure 4.5 et Figure 4.6. Ces figures montrent la distance entre les points, en orange pour l'odomètre du robot et en bleu pour le récepteur GNSS du robot. Il serait

attendu que l'odomètre ait un comportement similaire au GNSS et cela est observé dans la trajectoire corrigée. Par contre, dans la trajectoire absente de correction, le GNSS a une erreur si grande qu'elle est relativement incompréhensible. Il y a aussi une augmentation de distance entre les pas au début de la trajectoire non corrigée. Nous attribuons ceci à l'arrivée du robot en proximité du bâtiment dans la Figure 4.4. Ceci est dû à l'obstruction faite par le bâtiment et possiblement l'effet *multipath* dû à la surface métallique du bâtiment. On peut voir aussi que la correction permet de très bien réduire cette erreur.



FIGURE 4.4 – Résultat de trajectoire avec (en bleu) et sans (en rouge) correction. Notez les très grands écarts de position pour la trajectoire non corrigée.

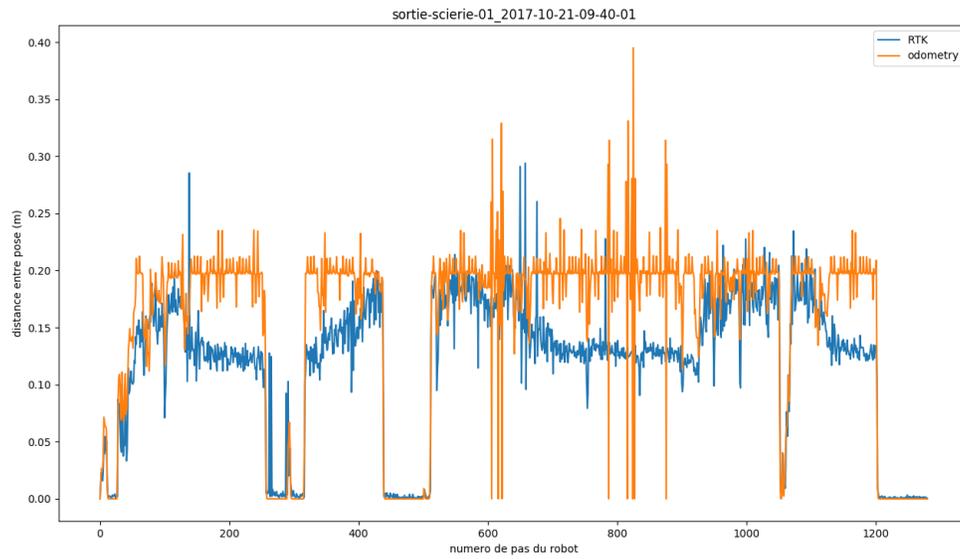


FIGURE 4.5 – Comparaison odomètre et GNSS avec correction lors de la trajectoire. On peut voir que la correction permet d’obtenir des distances similaires à ceux de l’odométrie.

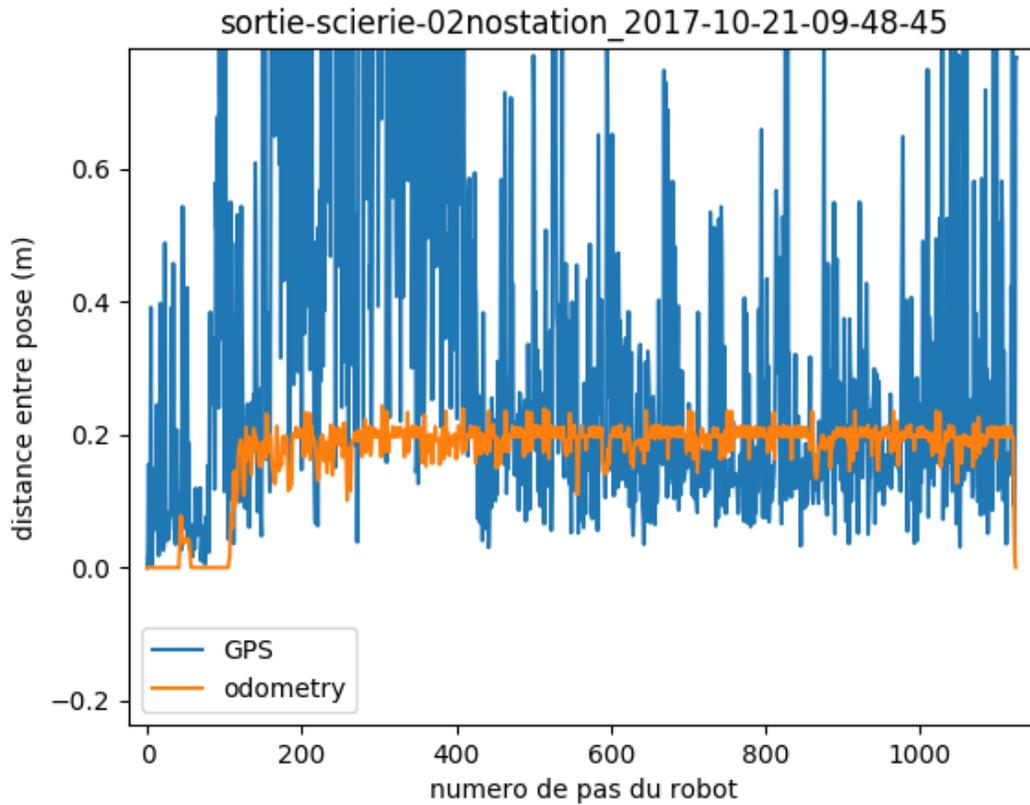


FIGURE 4.6 – Comparaison odomètre et GNSS sans correction lors de la trajectoire. Notez les très grands écarts de position pour la trajectoire non corrigée.

Nous avons fait une autre trajectoire afin de voir l’impact des occlusions partielles sur la performance. Pour cette trajectoire, l’occlusion provenait d’un abri en métal communément appelé *T-shed*, illustré à la Figure 4.7. Ces structures sont relativement communes dans les cours à bois. Du fait de leur couverture métallique, elles sont susceptibles d’interférer avec les signaux GNSS. Comme nous pouvons le constater à la Figure 4.8, cette trajectoire montre que l’obstruction partielle génère une source d’erreurs significatives pour le GNSS sans correction. Tout comme la trajectoire précédente, la même observation peut être faite en comparant l’odomètre du robot et le récepteur GNSS. Par contre, avec la correction, la trajectoire RTK a pu garder une bonne précision et robustesse, ne démontrant pas de grands sauts aléatoires.



FIGURE 4.7 – Exemple de T-shed

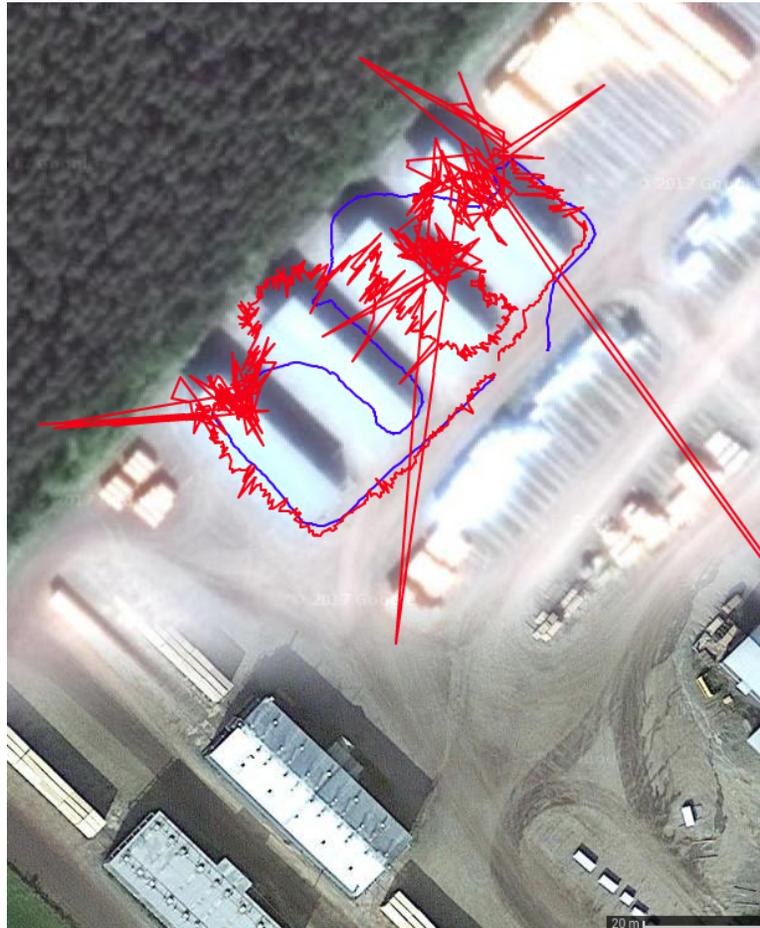


FIGURE 4.8 – Résultat de trajectoire obstruée avec et sans correction. Bleu : avec correction.  
Rouge : sans correction

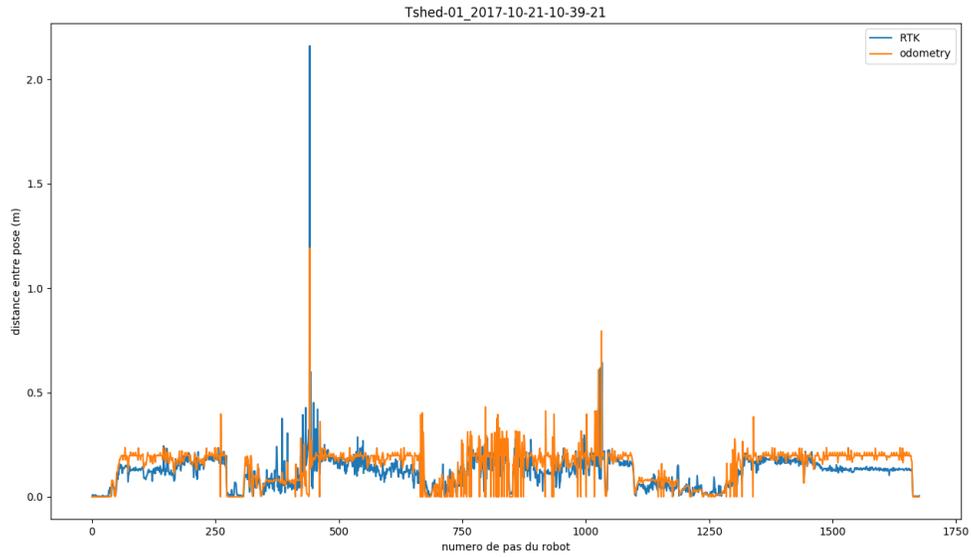


FIGURE 4.9 – Comparaison odomètre et GNSS avec correction lors de la trajectoire

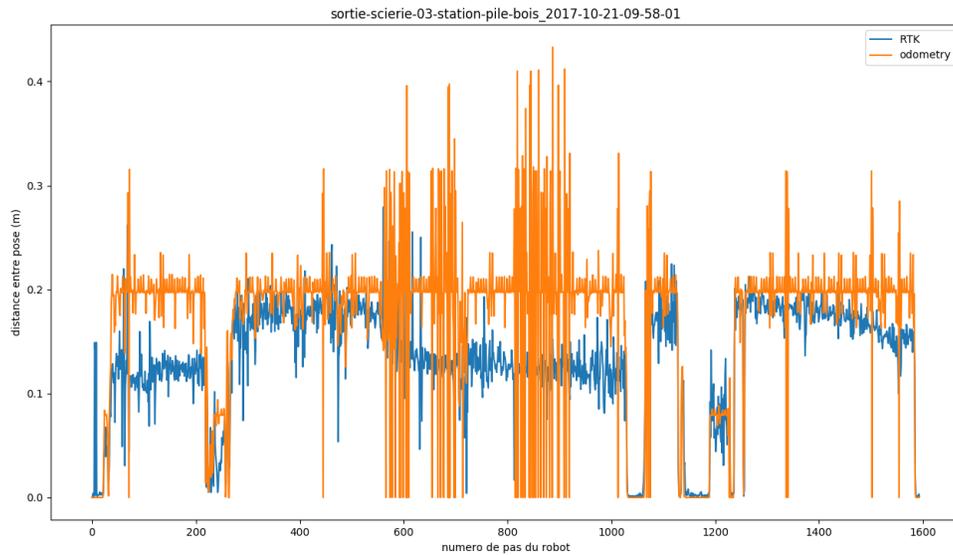


FIGURE 4.10 – Comparaison odomètre et GNSS sans correction lors de la trajectoire. Notez les très grands écarts de position pour la trajectoire non corrigée.

Dans cette trajectoire, nous pouvons observer quelque chose de similaire à l'autre trajectoire dans la cour à bois. L'obstruction faite par les *T-sheds* et les empilements de

bois provoquent de nombreuses erreurs. Encore une fois, la correction via RTK a permis d'avoir une trajectoire très précise.

### 4.1.3 Forêt

Il est possible qu'un véhicule soit appelé à se déplacer dans des zones ayant une bonne densité d'arbres, alors il est important de voir la performance de la correction dans un tel environnement. Pour cette récolte de données, nous avons utilisé encore une fois le robot *Husky* (illustré à la Figure 4.1), étant donné sa grande mobilité. Une vitesse constante de 1 mètre par seconde a été choisie pour faire les déplacements. La trajectoire a été faite dans un boisé de l'Université Laval, à des temps différents de l'année, soit avant, vers la fin octobre, et après, vers la fin novembre, que les feuilles soient tombées des arbres. Pour assurer une bonne répétition de la trajectoire lors de la conduite manuelle du robot, des arbres ont été marqués avec un ruban. Cet environnement a aussi été utilisé pour l'article scientifique dans le chapitre 5, par contre pendant le mois de mars.

Dans la Figure 4.11 suivante, on peut voir l'impact de la forêt sur la localisation via GNSS et RTK. On peut notamment constater que la correction a un effet similaire sur les erreurs provoquées par les obstructions de la cour à bois. Dans la Figure 4.11, les points mauves sont les positions sans correction. Les points verts et bleu pâle sont des positions avec correction. Les points verts offrent un positionnement de meilleure qualité que les bleus pâles [22]. Les points verts (Q1) sont des positionnements dits *fixed* et les bleus sont *float* (Q2) [23]. Un positionnement *fixed*, donc Q1, est lorsque les ambiguïtés ont pu être fixées sur leurs valeurs entières correctes. Un positionnement *float*, donc Q2, est lorsque les ambiguïtés ont une valeurs réelles estimées. Les points rouges (Q5) sont des points qui n'ont pas pu être corrigés ou il n'y avait pas assez de satellites pour faire un positionnement. On peut voir dans le bas de la figure le pourcentage de chacune des catégories de points. Pour la Figure 4.11, 13.8% sont verts, 85.5% sont bleus et 0.7% sont rouges.

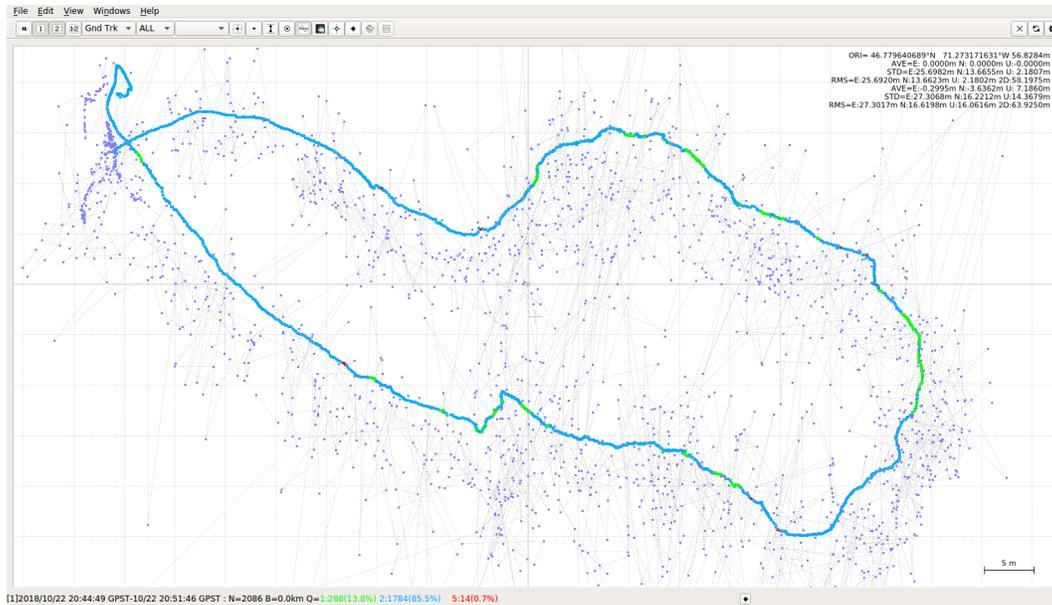


FIGURE 4.11 – Résultat de post-traitement sur des données brutes dans un boisé de l'Université Laval, avant chute des feuilles.

Comme la densité du feuillage peut influencer les performances de localisation des systèmes GNSS, nous avons refait cette trajectoire lorsque les feuilles sont tombées et donc ne participent plus à l'obstruction. Cette influence est aussi observée dans l'article du chapitre 5. Le résultat de cette trajectoire peut être vu dans la Figure 4.12. Si on regarde les pourcentages des catégories de points de positionnement, on peut voir qu'il y a plus de points verts (18.3%) que la trajectoire avec les feuilles (13.8%). Il y a, par contre, un plus grand pourcentage de points rouges, avec une augmentation de 0.7% à 4.7%. On peut voir aussi un décalage entre les trajectoires corrigées et non corrigées. Nous attribuons cela à la géométrie des satellites utilisés par le module. Si les satellites sont généralement dans une direction latérale du module, on pourra observer un décalage parallèle au positionnement des satellites. Dans notre cas, la limite du boisé était vers le sud. On peut donc voir un décalage parallèle à l'axe nord-sud.

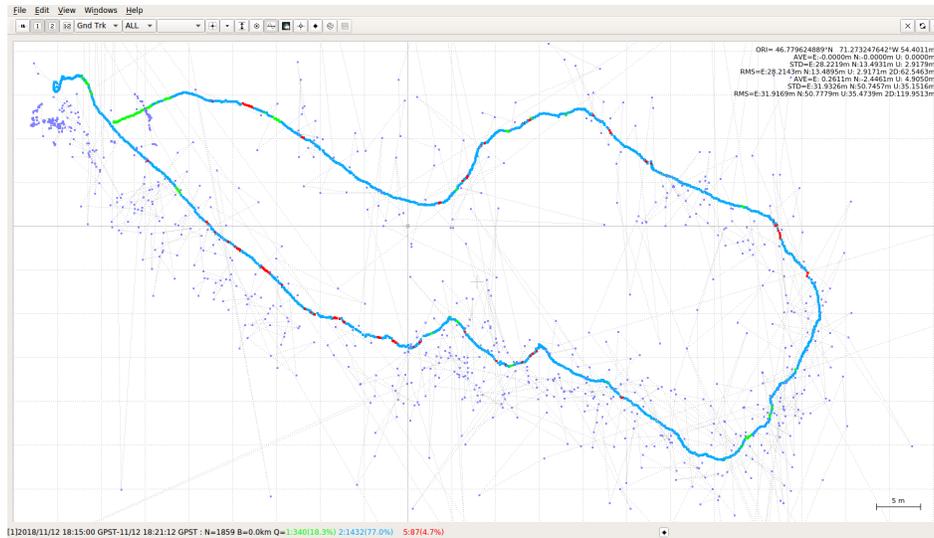


FIGURE 4.12 – Résultat de post-traitement sur des données brutes dans un boisé de l’Université Laval, après chute des feuilles.

Afin de mieux visualiser l’impact du feuillage sur la correction, nous avons calculé des histogrammes de vitesse, basée sur les positions retournées par le système de localisation. Comme la vitesse du robot était généralement constante, un système de localisation précis devrait retourner une distribution centrée autour de la vitesse réelle ( $1\text{ m/s}$ ), alors qu’un système fortement bruité verra cette vitesse varier fortement. Comme les positionnements GNSS et RTK sont horodatés, il est possible de calculer la vitesse avec la distance et la différence de temps entre deux positionnements qui s’enchaînent. La Figure 4.13 montre que la vitesse entre les positions varie considérablement sans la correction. En effet, dans les histogrammes non corrigés du haut, on peut remarquer qu’une grande portion des vitesses estimée par la différence des positions est supérieure à  $5\text{ m/s}$ , bien au-delà de la vitesse réelle du robot. Cette distribution peu réaliste des vitesses s’explique par le fait qu’une solution GNSS a une précision de l’ordre d’une dizaine de mètres. Ainsi, les mesures des positionnements ont été grandement affectées par les sources d’erreurs, produisant ces vitesses irréalistes. Pour les histogrammes de vitesse pour les positions corrigées via RTK, on peut voir que la majorité des vitesses sont entre  $0$  et  $1\text{ m/s}$ . Ceci a encore du sens, étant donné qu’une solution RTK a une précision de l’ordre du décimètre et que le véhicule se déplaçait à une vitesse inférieure ou égale à  $1\text{ m/s}$ . Cependant, il est difficile d’interpréter l’impact du feuillage sur la

qualité de la correction via RTK. En effet, la médiane et l'écart-type des vitesses ne changent pas beaucoup entre les deux histogrammes du bas, et peuvent être facilement attribués à des différences de positionnement de la constellation des satellites, ou aux variations naturelles de vitesse induites par l'opérateur du véhicule.

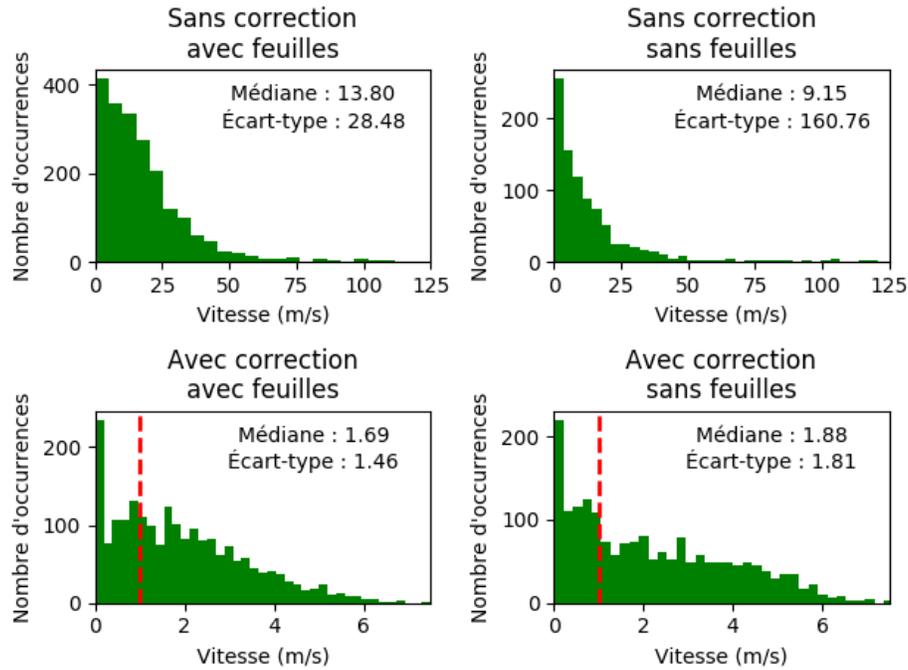


FIGURE 4.13 – Histogramme de distance des diverses trajectoires.

En regardant le nombre de satellites par catégories de positionnement, on peut voir que ce nombre influence beaucoup la qualité du positionnement. La Figure 4.14 et la Figure 4.15 montrent que la plupart des positions Q5 sont dues à un manque de satellite pour le module mobile. Cela peut être attribué au haut niveau d'obstruction venant de l'environnement, comme un boisé. Ce phénomène est quelque chose qui a été étudié en détail, et même prédit dans le chapitre 5.

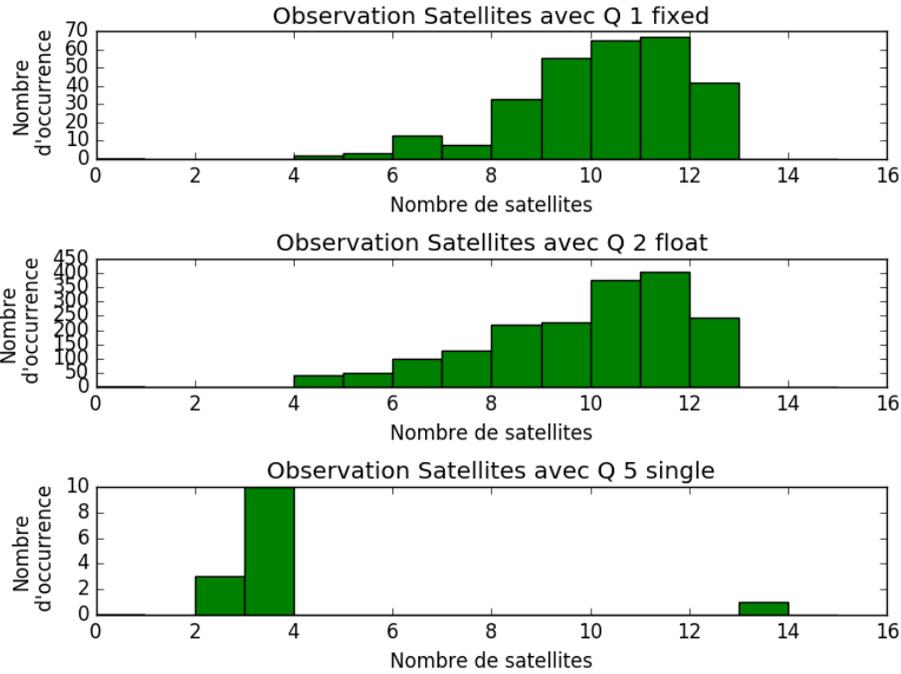


FIGURE 4.14 – Histogramme de nombre de satellites de la première trajectoire (Figure 4.11), avant la chute des feuilles, pour chacune des catégories de positionnement.

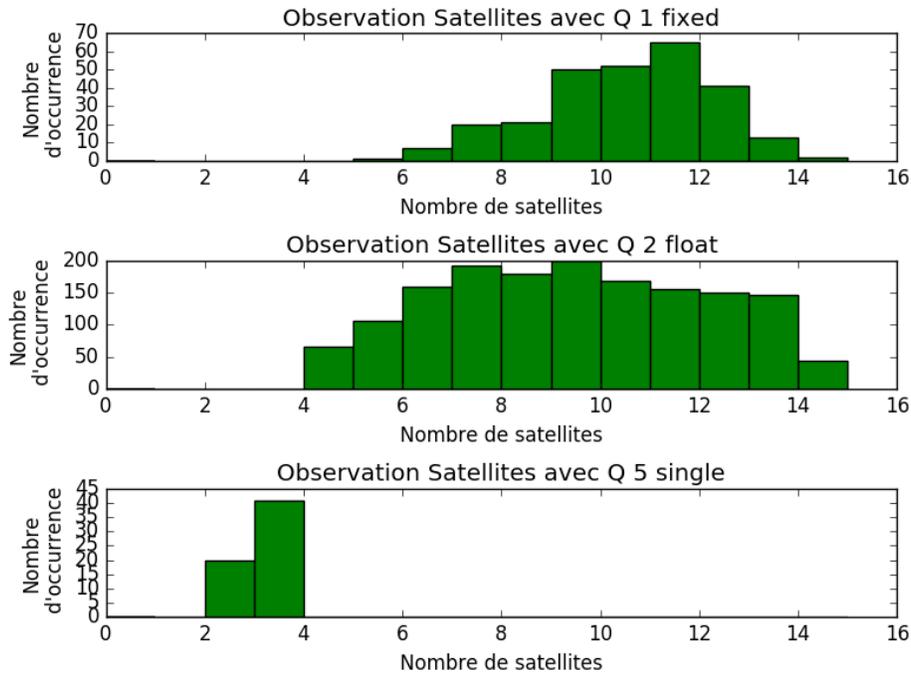


FIGURE 4.15 – Histogrammes du nombre de satellites de la deuxième trajectoire (Figure 4.12), après la chute des feuilles, pour chacune des catégories de positionnement.

Ceci montre un grand contraste de l’histogramme de satellites (Figure 4.16) de la trajectoire représentée dans la Figure 4.3, qui a été effectuée dans un environnement beaucoup moins obstrué. On peut voir avec la Figure 4.3, que la correction est beaucoup plus facile pour le module mobile.

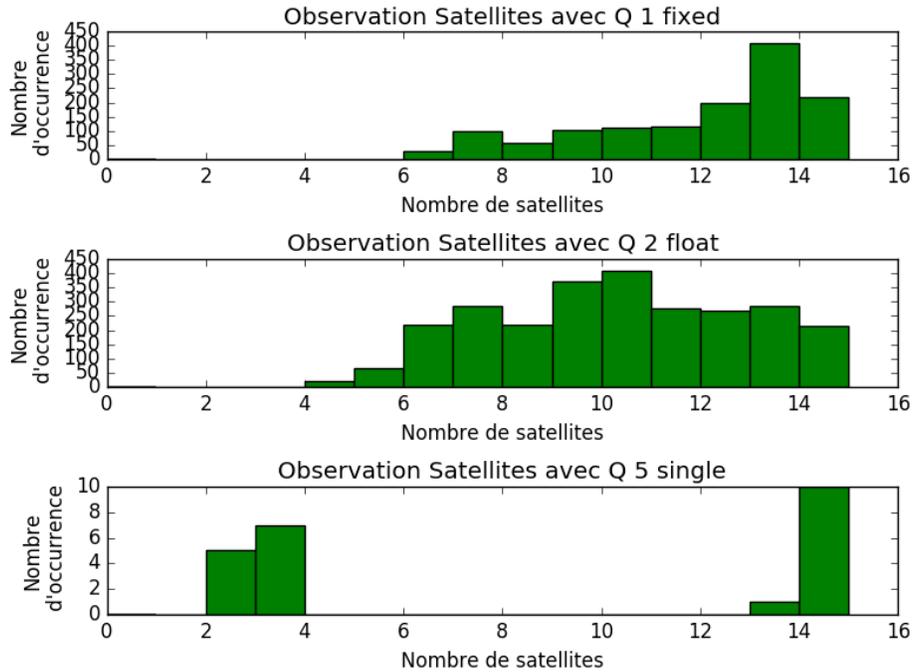


FIGURE 4.16 – Histogrammes du nombre de satellites de la trajectoire Figure 4.3, pour chacune des catégories de solution.

## 4.2 Résultats sur la génération de cartes

Pour les expérimentations utilisées lors de ce mémoire, plusieurs cartes ont été générées dans des environnements divers. Les prochaines figures montrent les environnements qui auront été utilisés dans la réalisation de l’article dans le chapitre 5. Les points rouges représentent les positions au sol que la plate-forme pourrait être située dans cet environnement c.-à-d. les points avec une normale de surface orientée vers le ciel. Les Figure 4.17 et Figure 4.20 ont été les plus faciles à générer, étant donné que les murs avec plusieurs coins droits et des fenêtres offraient des points de repère faciles pour l’algorithme de cartographie. Le reste des figures (Figure 4.18 et Figure 4.19) ont

été plus difficiles, étant donné que les arbres et les quelques murs sans caractéristiques proéminentes ne permettaient pas de bien faire le calage entre les nuages de points.

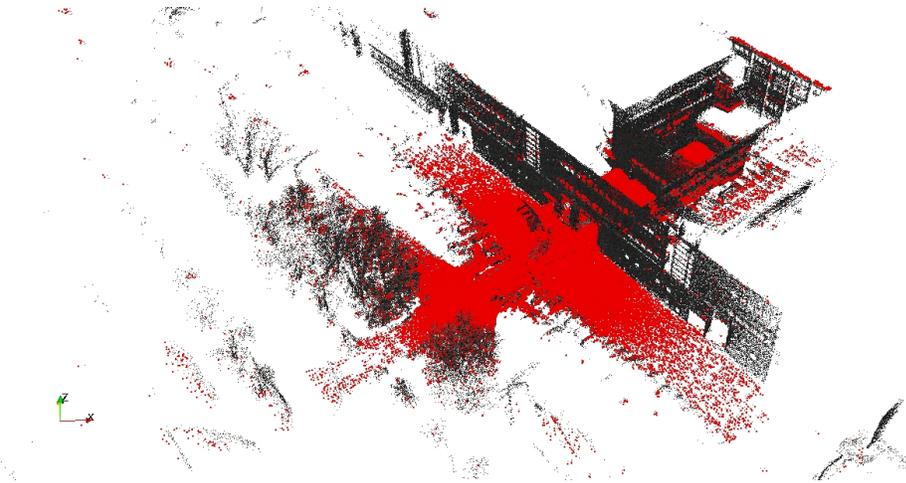


FIGURE 4.17 – Carte 3D du stationnement du Pouliot.

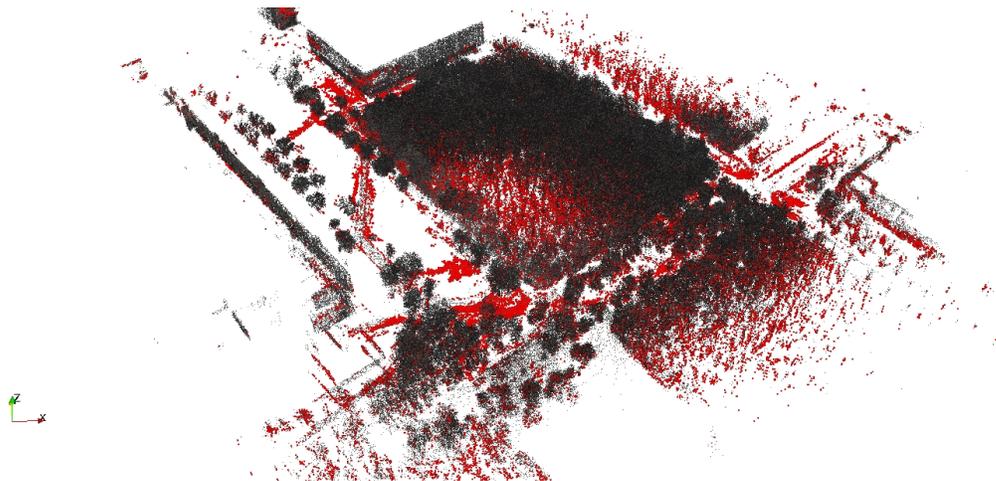


FIGURE 4.18 – Carte 3D du grand axe de l'université Laval.

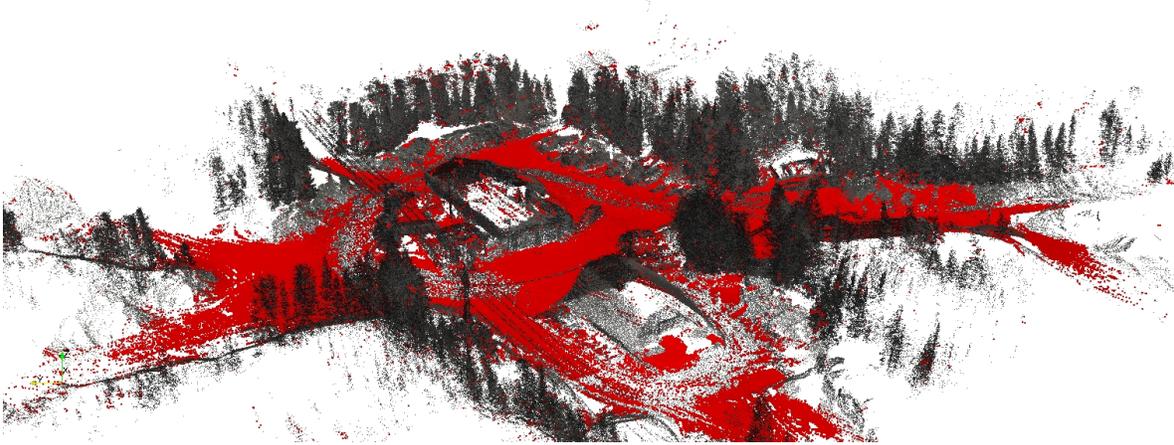


FIGURE 4.19 – Carte du garage de la forêt Montmorency.



FIGURE 4.20 – Carte 3D du pavillon Paul Comtois.

### 4.3 Conclusion

Dans ce chapitre, nous avons exploré les deux principales technologies utilisées pour le contexte de suivi de machinerie lourde. La première consiste en la possibilité d'utiliser les technologies de localisation via satellite. Plus particulièrement, la technologie du RTK, qui devient de plus en plus compétitive, permet de localiser un véhicule en milieu extérieur, avec une bonne précision. Notamment, nous avons constaté que, comparativement à un système GNSS standard sans correction (souvent dénommé GPS), les systèmes RTK sont à la fois beaucoup plus robustes et précis. Des tests dans une cour

à bois, ainsi que dans une forêt avec et sans feuillage, ont confirmé le haut potentiel des systèmes RTK pour l'industrie forestière. Finalement, ces tests ont permis de voir l'effet de l'obstruction partielle du ciel sur la réception du signal. Comme cette obstruction vient de l'environnement autour du récepteur, il est important pour l'analyse de données d'avoir une idée de la densité et de la géométrie de l'environnement. Nous représentons et générons l'environnement comme des nuages de points. Cette représentation et génération de l'environnement, la deuxième technologie, ont été expliquées en plus grand détail dans le chapitre 2. Cette obstruction partielle est justement le problème principal, donnant la motivation de l'article de la conférence Field and Service Robotics (FSR) présenté dans le chapitre 5.

# Chapitre 5

## Article FSR 2019

Ce chapitre contient l'article qui a été rédigé pour la conférence *Field and Service Robotics 2019*. Cette conférence est à sa 12e édition et a eu lieu à Tokyo, Japon. Cet article contient une explication plus spécifique sur la visibilité de satellites dans des environnements obstrués. Il y a aussi une revue de littérature sur les avancements faits sur la modélisation de cette visibilité. Ensuite, une explication détaillée de la méthode, des résultats obtenus lors des expériences et finalement d'une conclusion.

### 5.1 Résumé

Pour aider les futurs agents mobiles à planifier leurs déplacements dans des environnements difficiles, un modèle prédictif a été conçu pour déterminer les zones favorables au positionnement GNSS. Le modèle est capable de prédire le nombre de satellites viables pour un récepteur GNSS, sur la base d'une carte de nuages de points 3D et d'une constellation de satellites. Les effets d'occlusion et d'absorption de l'environnement sont pris en compte. Une plate-forme mobile robuste a été conçue pour collecter des données afin de générer les cartes de nuages de points. La plate-forme a été déployée pendant l'hiver canadien, connu pour sa grande quantité de neige et ses températures extrêmement basses. Les environnements de test incluent une forêt boréale très dense et un campus universitaire avec de grands bâtiments. Les résultats de l'expérience indiquent que le modèle fonctionne bien dans les environnements structurés et non structurés.

## 5.2 Abstract

To help future mobile agents plan their movement in harsh environments, a predictive model has been designed to determine what areas would be favorable for GNSS positioning. The model is able to predict the number of viable satellites for a GNSS receiver, based on a 3D point cloud map and a satellite constellation. Both occlusion and absorption effects of the environment are considered. A rugged mobile platform was designed to collect data in order to generate the point cloud maps. It was deployed during the Canadian winter known for large amounts of snow and extremely low temperatures. The test environments include a highly dense boreal forest and a university campus with tall buildings. The experiment results indicate that the model performs well in both structured and unstructured environments.

## 5.3 Introduction

Demand for precise localization of heavy machinery is on the rise in many industries, such as agriculture, mining, and even highway trucks driving. This demand is partially driven by a strong push for *Industry 4.0*, which relies on automation, real-time planning and tracking of the supply chain. It is also driven by an increased need for the safety of workers. For instance, logging is the single most dangerous profession in the U.S.A. [24], and could thus benefit from such an increase in vehicle automation. Due to the large variety of outdoor environments in which heavy machinery operates (cities, open spaces, forests), one must be able to ensure that localization will be robust. To estimate its position, an autonomous vehicle often combines measurements from different sources, such as 3D laser scanners (lidar), an Inertial Measurement Unit (IMU), cameras and GNSS solutions. In this paper, we focus strictly on the case where the GNSS modality is used online, along with a 3D map generated beforehand from lidar data.

GNSS consists of two parts : ground receivers, and constellations of satellites transmitting precise, atomic-clock-based time signals. Each of these satellites continuously transmits a radio signal which encodes time measurements given by its on-board ato-

mic clock. The GNSS receiver can estimate its distance (called a *pseudo-range*) from a satellite, based on the difference between its internal clock and the received satellite clock. With signals from at least four different satellites, it is possible to uniquely determine the position of a receiver, by solving a least-squares linear problem. The main factors affecting the quality of a GNSS are the number of visible satellites, and the precision of individual pseudo-range measurements. To increase precision and robustness, modern GNSS receivers now rely on multiple satellite constellations (e.g., GPS from the United States, Russian GLONASS, Galileo from the European Union and China's Beidou). In ideal conditions, the radio signal from the satellite to the receiver would travel without interference. In reality, this signal is affected by the Earth's atmosphere. It also possibly reaches the receiver via multiple paths, due to reflections from objects on the Earth's surface. To cope with these problems, GNSS solutions can be augmented by a correction system, becoming a DGNSS.

Although the state-of-the-art DGNSS receivers can achieve sub-centimeter accuracy in nominal conditions, deployments in increasingly harsher environments such as forests bring new challenges. Contrary to buildings, trees do not reflect the signals, and therefore the multipath error (called *urban-canyon effect*) is less prominent (see Figure 5.1). Instead, trees tend to gradually absorb the signals coming from satellites [25]. Importantly, this attenuation of signals may render some satellites unusable by the receiver [26]. In such conditions, GNSS receivers may not be as precise [27], even with the correction systems [28]. There is therefore a strong need to be able to predict the impact of the environment on these signals, in order to assert the ability to locate.

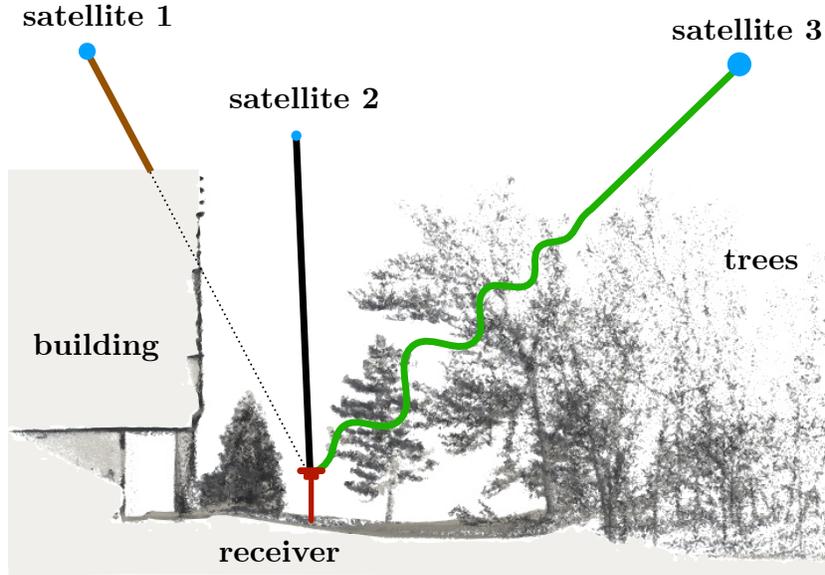


FIGURE 5.1 – Side view of a complex environment with a building on the left and trees on the right. A GNSS receiver is in the middle, trying to capture satellite information. Signals from GNSS interact differently with the environment, depending on the structure involved.

One way to quantify the quality of a satellite signal is through its Signal to Noise Ratio (SNR). Research so far has focused mainly on means to either estimate the effect of the environment on the SNR for individual satellites, or to determine the effect on the positioning as a whole. In this paper, we propose a model that can predict the visibility of a GNSS constellation in complex environments such as a highly dense forest with high absorption and urban canyons with high occlusion by buildings. To achieve this prediction, our model takes into account the geometry of both the satellite constellation and the environment. In particular, our approach is able to leverage local shape and density of 3D point clouds to determine which mechanism (occlusion vs. absorption) is more likely, for a given satellite.

## 5.4 Related work

Several ways of estimating the impact of the environment on disrupting the satellite signal of GNSS have been proposed in the literature. One related line of work investigates the impact of non-absorbing obstacles such as buildings on GNSS signals. TAYLOR et al.

[29] explicitly modeled the environment in 3D for the purpose of determining the availability of GNSS signals through line-of-sight. Geographic Information System (GIS) techniques (i.e., a combination of photogrammetry and lidar-based elevation mapping) were used to create polygonal models of buildings. Similarly, the work of ZHANG et al. [30] shows how to use a 3D model of an urban canyon (provided by the *Google Earth* application) to determine the pseudo-range error caused by the multipath signal reception. The same 3D model was also used to determine the distribution of the positioning error in a given area. The knowledge of the pseudo-range error in urban canyons allows correction of the GNSS positioning as presented in [31]. Their approach achieves reduction of the positioning error from 9.2 to 5.2 m, using 3D models based on publicly available GIS data. In [32], reflected satellite signals are interpreted as new satellites which add new information to the problem. This approach was tested at a wind tunnel intake whose front wall acted as a perfect reflector due to a safety wire mesh.

These methods [29]–[32] assume that a precise 3D model of the environment, typically constructed from GIS data, is available beforehand. Moreover, they target exclusively urban canyons, which is too restrictive for some industries, such as forestry. Another line of work studied the impact of *signal attenuation* from absorption, with a special attention to trees. For instance, WRIGHT et al. [33] has shown that one can use the absorption coefficient index and the elevation map of a forest to estimate the effect of the vegetation on SNR signals. An extension of this work employed sky-oriented photos as a means of estimating canopy closure in pine forests WRIGHT et al. [34]. This method only estimates the signal loss, but cannot determine how many satellites would be viable. Therefore, it is not well suited for satellite visibility prediction. Similarly, HOLDEN et al. [35] established a relationship between the fragmentation of sky pictures and the accuracy of the positioning. However, this method focused on tree canopy using stationary analysis without modeling the effect of structures such as buildings. It also only used photos to determine obstacles. In our work, we extend this approach by explicitly modeling the effect of structures that exhibit a masking effect, from their uniform flat surfaces.

While some methods estimate the pseudo-range error from an a priori 3D model of the environment, others exploit 3D point clouds to determine whether or not a given satellite signal is reachable by the GNSS receiver [36]. The line-of-sight between the satellites and the receiver is then simply determined by using a ray tracing algorithms. By identifying these satellites, the authors were capable of estimating off-line a more realistic confidence on the positioning. This method, however, exhibits problems with large trees since they do not necessarily block the signals entirely, but rather may absorb some of them. Therefore, simply removing a satellite because of tree occlusions makes their model underestimate the positioning confidence. CHEN et al. [37] proposed to use a 3D point cloud to generate a binary mask of the sky to remove occluded satellites. A similar method [38] has been used in an urban environment, using infrared images taken at night, to estimate satellite masking from buildings. Again, if a satellite lies behind *any* obstacle (building or tree), it is entirely masked. This binary approach is not well suited for the highly dense forest environment. Indeed, most of the satellites would be deemed as masked, given the fact that the majority of the sky is occluded by the canopy. Our approach is similar to the last two, but also take into consideration the type of structures affecting the occluded satellites. More precisely, we take into account that trees do not block the signal completely, but rather decrease their SNR.

## 5.5 Theory

Our goal is to predict how the environment influences the ability of a GNSS receiver to locate itself. More precisely, we look at modeling the environment’s interaction with the signal of each satellite, from the ground receiver perspective. To this effect, we leverage GNSS constellation configuration information and a representation of the environment, namely a large 3D point cloud map of an area of interest, to predict an effective number of satellites available for localization.

### 5.5.1 Processing GNSS information

Ground receivers produce information, in the form of *sentences*, under the format of NMEA. We use three specific sentences to understand the effect of the environment on the receiver : 1. **RMC** for the estimated position of the receiver, 2. **GST** for statistics on the positioning of the receiver, and 3. **GSV** for information on individual satellite. From the **GST** sentence, one can extract a positioning metric called Dilution Of Precision (DOP). This metric is a unitless indicator of the confidence the receiver has in a positioning output. For example, a DOP of 1 is considered as ideal, but one higher than 20 signifies poor positioning accuracy. With the **GSV** sentence, one obtains a satellite's position in the sky (elevation and azimuth) as well as its SNR. A good SNR would be around 45 dB, while a poor SNR would be around 30 dB. This information allows an understanding of what is happening to the signals, particularly when measuring how different environments are affecting them. To make sure we obtain a good positioning, the mobile receiver is corrected by a reference station. The correction comes from using RTK, a reference station measures errors, and that station, then transmits corrections to the mobile receiver. We obtain the satellite constellation from the reference station used for the RTK correction.

In our model, we represent a satellite constellation as a Gaussian mixture model, with one Gaussian per satellite. Its mean represents the position of a satellite, while its variance  $\sigma$  captures the effect of the GNSS acting as a radio wave source. We further process this representation as a discretized hemispherical projection of the sky  $\mathcal{S}$ , with each cell  $s_{ij}$  having a resolution of  $e \times l$  (i.e., division on azimuth and elevation). Given that Gaussians are used, this *sky maps* has the property that summing all cells  $s_{ij} \in \mathcal{S}$  gives the total number of satellites  $v$  that are visible at this time and place. An example of such a hemispherical projection  $\mathcal{S}$ , and the variation of visible satellites along a trajectory can be seen in the Figure 5.2.

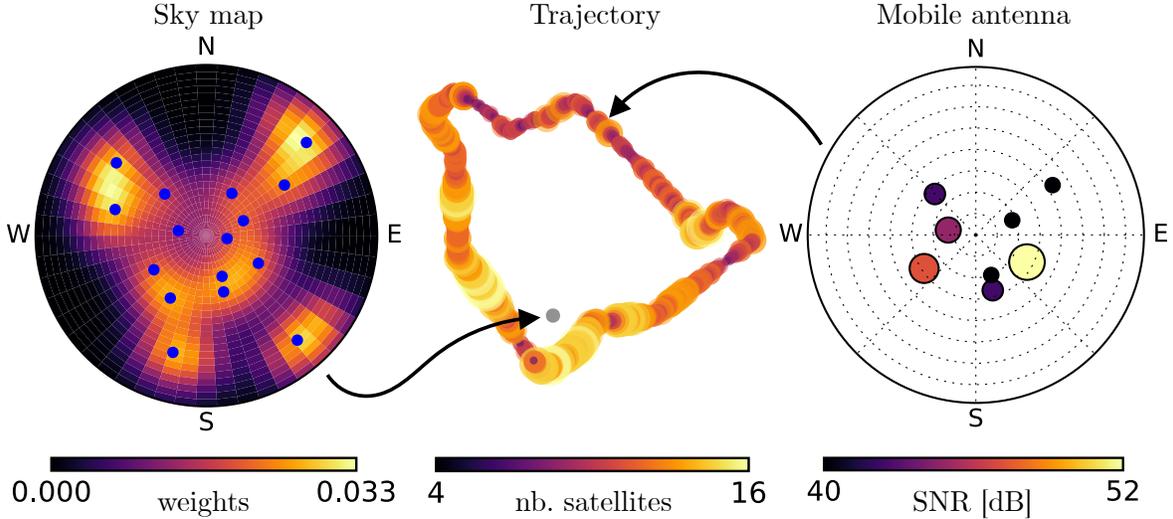


FIGURE 5.2 – Different uncertainty information accessible through NMEA sentences. *Left* : individual satellite positions from a reference antenna installed in an open area and used as reference to build the sky map  $\mathcal{S}$ . *Middle* : trajectory of a moving receiver passing from open sky to a forest, size and color coded with the number of visible satellites at each position. The large variation of the latter is the justification for our approach. *Right* : individual satellite position with larger dot representing higher SNR for a single position of the moving receiver.

### 5.5.2 Processing 3D Points

The second source of information comes from a dense 3D representation of the environment in the form of a map  $\mathbf{M}$ , generated offline. One issue with mobile lidars is that they produce an uneven distribution of points when registered together to form such a 3D map. To mitigate this, we used an octree to produce a uniform distribution of points by keeping a single point per bounding box of  $d_{\text{box}}$  wide. Since the environment can either absorb (e.g., forest) or block (e.g., buildings) signals our model has to treat these sources of obstruction differently. We rely on the shape of the distribution of neighboring points to determine if a given obstacle is most susceptible to cause absorption or occlusion. More precisely, for each point, we compute the covariance of its  $k_{\text{nn}}$  nearest neighbors. If the distance between a tested point and the mean  $\mu_{\text{nn}}$  of its surrounding points is greater than  $d_{\text{nn}}$ , this point is on a corner or at the periphery of the map. We consider that for these points, the covariance computation is invalid, and they are thus removed. For the remaining valid covariances, we then sort their eigenvalues  $\lambda_i$ , with  $\lambda_1 < \lambda_2 < \lambda_3$ . By analyzing their relative ratios, we defined local features that

try to capture the level of ‘unstructureness’  $u \in [0, 1]$ , ‘structureness’  $s \in [0, 1]$ , and the spherical level  $\delta \in [-1, 1]$  of each point. These features are defined as :

$$u = \frac{\lambda_1}{\lambda_3}, \quad s = \left( \frac{\lambda_2}{\lambda_3} \right) \left( \frac{\lambda_2 - \lambda_1}{\sqrt{(\lambda_2^2 + \lambda_1^2)}} \right), \quad \text{and} \quad \delta = u - s. \quad (5.1)$$

We can observe from Equation 5.1 that, as  $\lambda_1$  approaches  $\lambda_3$ , the value of  $u$  will be close to one. This will be the case, for spherical distributions, typical of objects present in unstructured environments. A geometrical interpretation is less straightforward for  $s$ . When looking at the first ratio for  $s$ , as  $\lambda_2$  is approaching  $\lambda_3$ , the covariance tends to assume a circular shape, while the second ratio will tends one if  $\lambda_1$  is a lot smaller than  $\lambda_2$  (i.e., a flat shape). If the local geometry is close to a line, then both  $u$  and  $s$  will be near zero. By subtracting these two values, we can represent the spherical level on one axis  $\delta$  with values close to minus one being planes, around zero being edges, and close to one being a diffuse sphere. Figure 5.4 (*left*) shows an example of the spherical level  $\delta$  in an environment with buildings and trees.

### 5.5.3 Predicting the number of satellites

For a given point on the ground and its normal vector  $\mathbf{n}$ , we compute the estimated number  $\hat{v}$  of visible satellites. To do so, we developed a hemispherical and discretized reduction model  $\mathbf{P}$  of the sky perceived by a receiver at this pose, aligned with the normal  $\mathbf{n}$ . It is built with the same resolution as our satellite map  $\mathbf{S}$ . For each cell  $p_{ij} \in \mathbf{P}$ , two statistics are extracted from the 3D points of the map  $\mathbf{M}$  falling into its angular bounding box. The first one,  $\delta_{\text{med}}$ , is the median value of the  $\delta$ ’s computed with Equation 5.1, for all the points within this cell. This variable  $\delta_{\text{med}}$  tries to capture the type of interaction (occluding vs. absorbing) a satellite signal would experience when passing through this cell and towards the receiver. The second statistic is the number  $m$  of points in this cell. This number is independent of the lidar scanning density, due to the voxel filtering described in sous-section 5.5.2. We propose to model the signal reduction  $p(\cdot) \in [0, 1]$  as a function that transitions smoothly from occlusion

to absorption :

$$p(\delta_{\text{med}}, m) = \underbrace{\left(1 + \exp(-\alpha(\delta_{\text{med}} - \beta))\right)^{-1}}_{\text{from occlusion to absorption}} \underbrace{\exp(-\gamma m)}_{\text{absorption model}} . \quad (5.2)$$

The parameter  $\alpha$  controls the slope of the transition,  $\beta \in [-1, 1]$  the position of the transition, and  $\gamma$  is the absorbance. For cells containing mostly occluding obstacles (such as in structured environments),  $\delta_{\text{med}}$  is generally around -1 and the first term (sigmoid function) will be near 0. This captures the fact that when going through highly structured obstacles, the GNSS signal is fully absorbed, i.e.  $p \rightarrow 0$ . For diffused obstacles,  $\delta_{\text{med}} \approx 1$ , and the first term will be close to 1. Consequently, the signal is reduced following an exponential decay function, based on the number of points in a cell  $m$  (i.e.,  $p$  would be approximately equal to  $\exp(-\gamma m)$ , the absorption model). To let the signal to go through when there are very few points, a binary mask  $\mathbf{B}$  is applied. Each cell  $b_{ij} \in \mathbf{B}$  equals 1 if the number of points  $m$  is lower than  $m_{\text{occ}}$ , and 0 otherwise. By comparison, the solution proposed by [36] is equivalent to masking a cell if the number of points  $m$  is larger than 0. Figure 5.3 gives a visual intuition of how  $p(\cdot)$  is computed for two different types of environments. Finally, we multiply our satellite representation  $\mathbf{S}$  with the maximum of our reduction model  $\mathbf{P}$  and our binary mask  $\mathbf{B}$  and sum over all cells to predict the number of satellites viewed  $\hat{v}$  following

$$\hat{v} = \sum_i \sum_j s_{ij} \max(p_{ij}, b_{ij}), \quad \text{with} \quad \hat{v} \leq v. \quad (5.3)$$

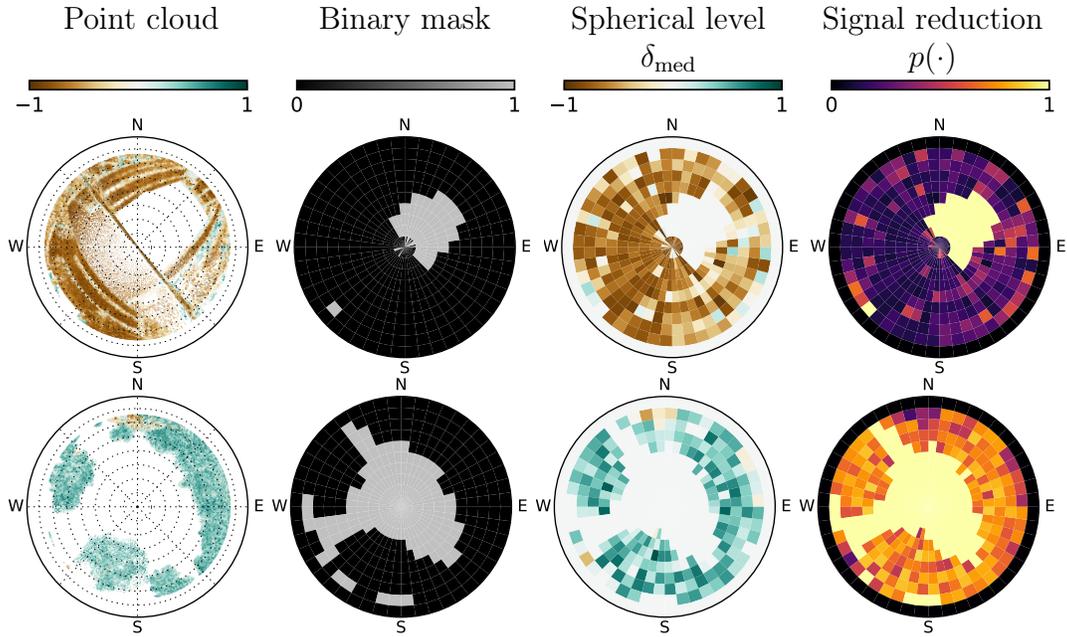


FIGURE 5.3 – Example of signal reduction. *Top row* : a structured environment with a building covering part of the sky. *Bottom row* : an unstructured environment from a path in a forest. In the point cloud, the points have been colored with their spherical level value  $\delta$  to show the difference between structured and unstructured environments. Those values are reflected in the spherical level histogram and have an effect on the signal reduction.

### 5.5.4 Extrapolation to Any Position on the Ground

As a last processing step, we want to predict the number of satellites for all potential places in a map  $\mathbf{M}$  where a virtual receiver could stand. Figure 5.4 (*right*) shows an example of the segmentation of the potential positions (i.e., the ground). For each point, the spherical level  $\delta$  and its eigenvector associated to  $\lambda_1$  (i.e., the surface normal vector  $\mathbf{n}$ ) is used to segment the ground following the constraints  $\delta < \delta_{\text{ground}}$  and  $\arccos(n_z) < \epsilon$ . Because the ground is not always flat, we use the surface normal  $\mathbf{n}$  to position the virtual receiver at the proper angle and height. This step is important to properly control the field of view of the virtual receiver in situations where a robot is heavily tilted and the antenna would point toward obstacles.

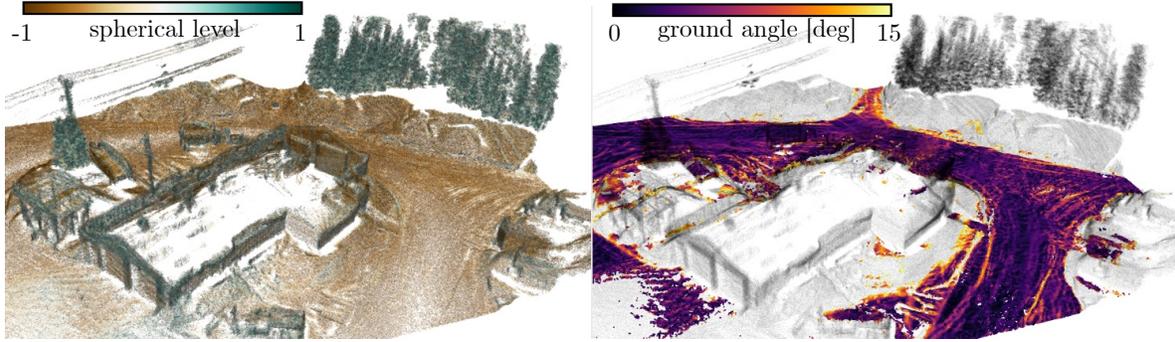


FIGURE 5.4 – Different information extracted from the geometry of a point cloud for the absorption model and the ground segmentation. *Left* : side view of a map with color representing the level of structure. *Right* : from the points strongly defined as structured, a subset of points with surface normal vector pointing upward is labeled as the ground.

## 5.6 Experimental Setup and Methodology

To test and validate our prediction model, we designed a mobile data collection platform composed of an IMU, a 3D lidar, along with a mobile and a fixed GNSS antenna. The IMU, an *Xsens Mti-30*, was used for the map  $\mathbf{M}$  generation and the state estimation. The lidar, a *Robosense RS-LiDAR-16*, was mounted with a tilt angle of  $27^\circ$  to ensure that we can capture vertical structures as high as possible. This was necessary to properly generate a view of the environment above the antenna via the map  $\mathbf{M}$ . Our algorithm is independent of the mapping and localization sensors used, as long as accurate and dense 3D maps can be produced from them. Both GNSS antennas are model *Emlid Reach RS+*, and coupled together to provide an RTK solution. We used the default factory settings for which the receiver units ignored satellites beneath an elevation of  $15^\circ$  above the horizon or having an SNR lower than 35 dB. The mobile receiver was placed one meter above the platform to prevent obstructions from the platform itself. As for the static receiver, it was mounted on a tripod and placed at a location having an open view of the sky. All the mobile sensors were connected to a computer inside a *Pelican* case. This waterproof case protected the electronics, while serving as a mechanical mount to hold the sensors. This rugged and compact platform, thus allowed for field deployments in harsh winter conditions. It also allowed for various means of transport, as displayed in Figure 5.5, depending on terrain complexity.



FIGURE 5.5 – *Left* : Rack mounted on a small robot for easy terrain (e.g., parking lot, sidewalks). *Middle* : Rack pulled by a snowmobile for maps on hardened snow. *Right* : As a last resort, the rack could be pulled by humans, which has proven to be useful on narrow/dense paths in the forest.

Five data gathering campaigns, as shown in Figure 5.6, were conducted on the Laval University campus and its remote research forest, the Montmorency Forest. All environments involved in the experiments were covered by a thick layer of snow and targeted variable numbers of satellites, from zero to 20. Some of the trajectories, namely **Parking** and **Courtyard**, are within well-structured environments. Both of these trajectories passed under covered porches to reach interior courtyards, creating situations where few satellites are visible. Another trajectory, **Mixed**, explored an unstructured environment. It started in an open field with sparse large trees on each side of a path and eventually traversed a deciduous forest, which is sparse during the winter since all the leaves have fallen. Finally, the last trajectory **Garage-1** was in the Montmorency Forest circling small storage buildings. The surrounding area was a coniferous forest, which remains very dense during winter. We adjusted our parameters based on **Garage-1**, **Mixed**, **Parking** and **Courtyard**. We used **Garage-2** as our validation dataset by returning 4

weeks later for another trajectory in the same area so that the satellite constellations would be different. Finally, to support a better reproducibility of our results, we list all of the parameters used during our experiments in Tableau 5.1. All maps and trajectory depictions follow the East North Up (ENU) geographical coordinate system.

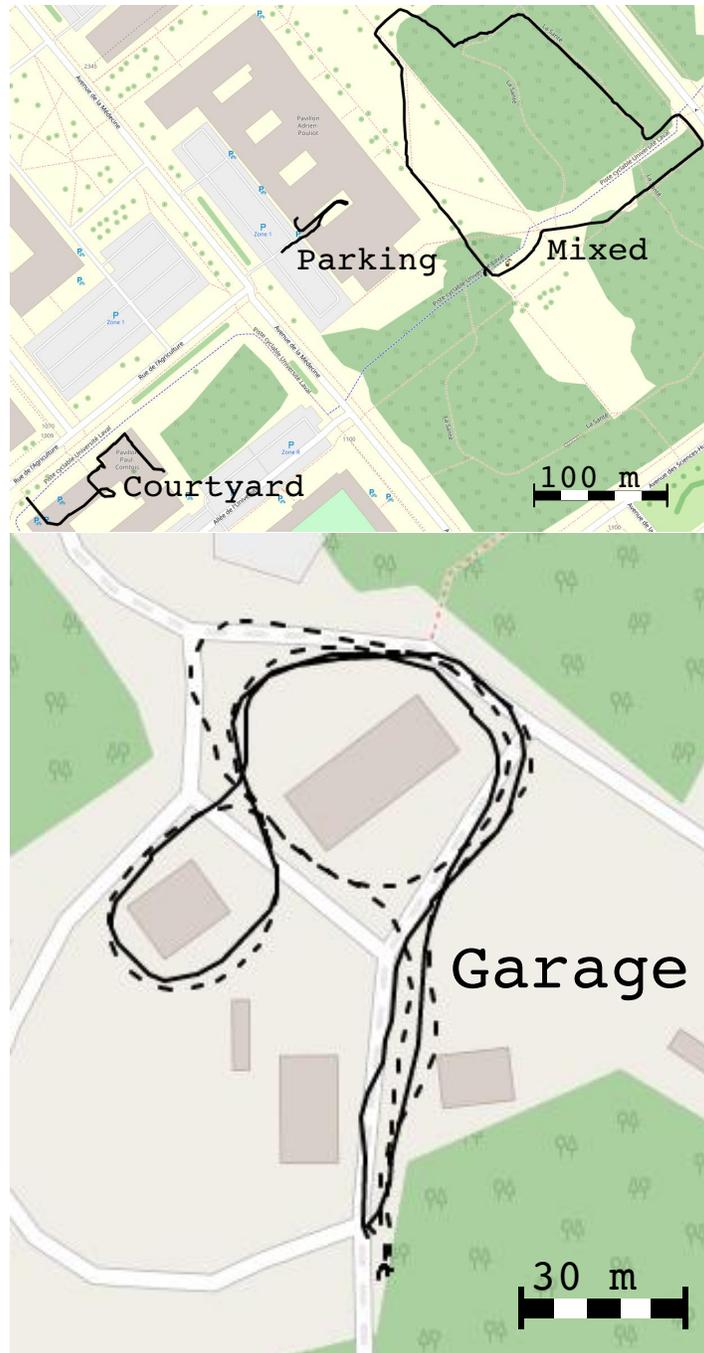


FIGURE 5.6 – Overview of all trajectories used during our experiments, all of them in Quebec, Canada. *Left* : three trajectories on the campus of Laval University. *Right* : two trajectories in the Montmorency Forest. The dashed line being the validation trajectory *Garage-2*. ©OpenStreetMap contributors.

TABLE 5.1 – List of parameters used for our results.

<i>Description</i>	<i>Parameter</i>	<i>Experimental value</i>
Variance on satellite angular positions	$\sigma$	12.5°
Octree bounding box size	$d_{\text{box}}$	0.1 m
Number of nearest neighbors	$k_{\text{nn}}$	50
Maximum offset from the mean nearest neighbors	$d_{\text{nn}}$	0.25 m
Maximum structured value for the ground	$\delta_{\text{ground}}$	-0.6
Maximum angle for ground classification	$\epsilon$	10°
Resolution of the angular histogram	$e \times l$	7.5° × 9°
Minimum number of points for occupancy reduction model	$m_{\text{occ}}$	5
	$\alpha, \beta, \gamma$	4, 0.25, 10 <sup>-10</sup>

## 5.7 Results

First, we evaluated the capacity for our reduction model  $p(\cdot)$  to adapt from structured to unstructured environments, by looking at specific trajectories and comparing it with the current literature. Then, we investigated the stability of our model through four datasets. Finally, we validated our extrapolation of the number of satellites on all points from the ground with a separate dataset.

### 5.7.1 Adaptability to the Environment Geometry

As a baseline, we used a binary removal of the satellites as in [36]. This baseline was chosen because it also employs point clouds for satellite occlusion prediction. By comparing the **Mixed**, **Parking** and the **Garage-1** for the Montmorency Forest, we determined that our model performed best in dense environments. From Figure 5.7 (left), we can clearly see that for areas covered by trees (the green region), our model can predict the number of visible satellites, while the forest caused too much obstruction for the binary removal to be effective. This is why an absorption model is appropriate for highly dense forest environment. The blue zones in (right), corresponding to coverage by buildings, demonstrate that our approach can also cope with this kind of occlusion.

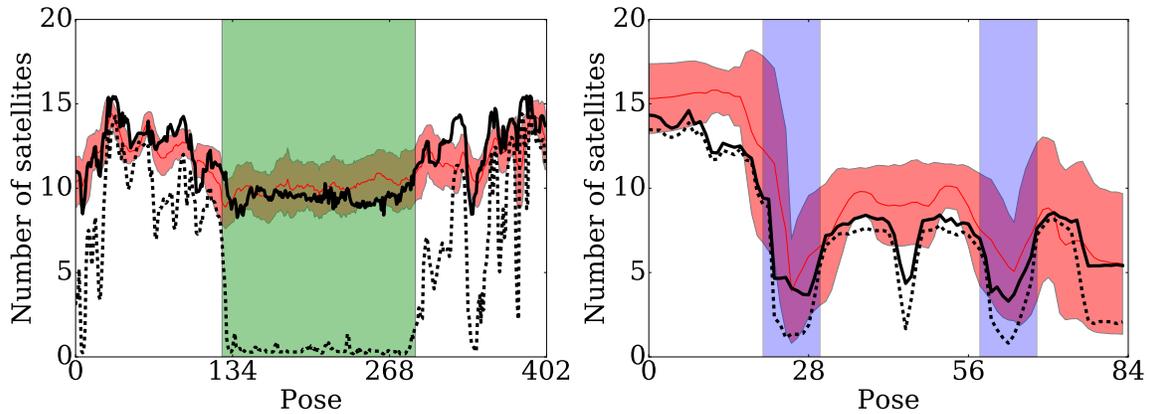


FIGURE 5.7 – Number of satellites actually perceived (red line) compared to our model (black line) and the model of [36] (dashed black line). The red shaded area corresponds to the standard deviation from an averaging window of 5 m *Left* : trajectory from **Mixed** passing through highly dense forest that covers the sky. *Right* : trajectory from **Parking** passing through structured environment that blocks signals. At Pose 40, the platform is right next to a tall building. The model sees the wall as an obstruction, but since the platform is in a reflective environment, it still receives a good amount a signal.

### 5.7.2 Stability Through Different Environments

To further illustrate the ability of our model in Equation 5.2 to cope with a wide variety of environments, we show the relationship between the actual and predicted number  $\hat{v}$  of visible satellites. The predictions are for the four trajectories **Garage-1**, **Mixed**, **Parking** and **Courtyard** that were used to identify the parameters  $\alpha$ ,  $\beta$  and  $\gamma$ . The plot in Figure 5.8 shows that our model is indeed able to predict the number of visible satellites with a reasonable error, for all four trajectories. Moreover, it is able to cope with a large range of satellite visibility (0-20).

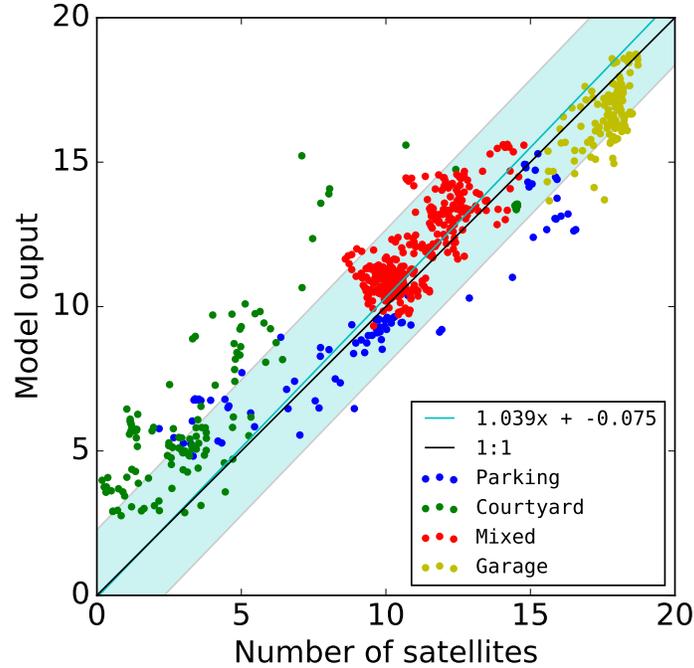


FIGURE 5.8 – Relation between satellite visibility and the number of satellites seen. Each of the trajectories has been weighted by the number of points and from that we fitted a polynomial equation. This equation (cyan line) is compared with a 1 :1 black line. The cyan area is the mean squared error from the fitting. Since the number of satellites has an influence on the DOP [39], the model can help determine the confidence in the positioning output. This is also why we also keep the points with less than 4 satellites, so that the model can determine those states also.

### 5.7.3 Validation of the Predicted Coverage

Since *Garage-2* is the same environment, the performance of our model should be similar. The Figure 5.9 shows the predicted number  $\hat{v}$  of visible satellite, along the trajectory. Just like the previous ones, our model tends to underestimate the number of visible satellites. Again, the model of [36] performs worse than ours, with a much more significant underestimation.

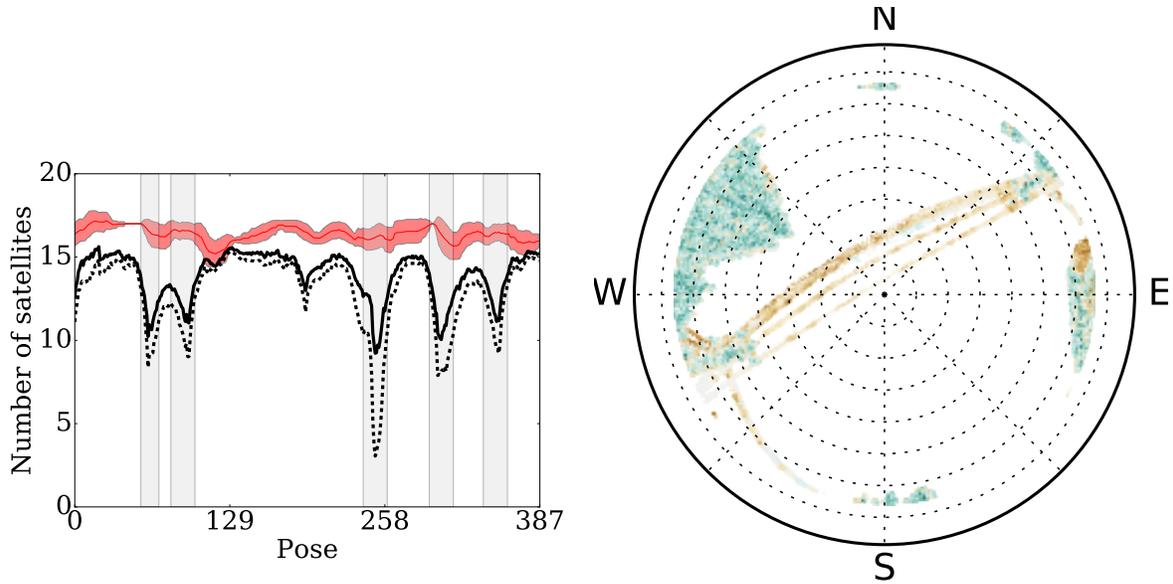


FIGURE 5.9 – *Left* : Validation trajectory performance with the areas with power lines in gray. *Right* : Point cloud with a view of the power lines that are identified as structured points.

Given our model, a query time and a 3D map, we can now predict ahead of time what would be the positions for a GNSS receiver. Such a *visibility map*  $M_v$  is depicted in Figure 5.10. By changing the orientation of the satellites in the sky, we can see the effect it has on the map  $M_v$ . We can see that in certain areas, such as near buildings or trees, the number of expected visible satellites is reduced. This visibility map  $M_v$  could be given to a path planning algorithm that minimizes the probability of getting lost [40].

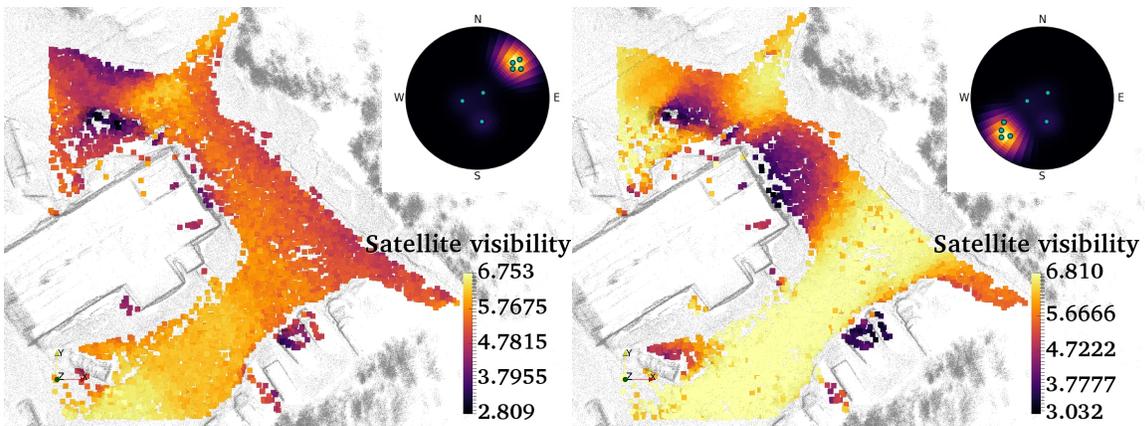


FIGURE 5.10 – Predicted coverage on the ground plane. *Left* :  $M_v$  with satellite signals from around the north-east. *Right* :  $M_v$  with satellite signals from around the south-west.

### 5.7.4 Histogram resolution on occupancy

One encountered difficulty is that a planar surface may not produce as many points as a tree would. Because of this, the histogram bins contain fewer values to work with and so the precision decreases as shown in Figure 5.11. If the resolution is too high, some parts of the map may be classified as unoccupied in the binary mask  $\mathbf{B}$  (e.g., a wall of a building which is not dense enough). The signal is then incorrectly allowed through by the model.

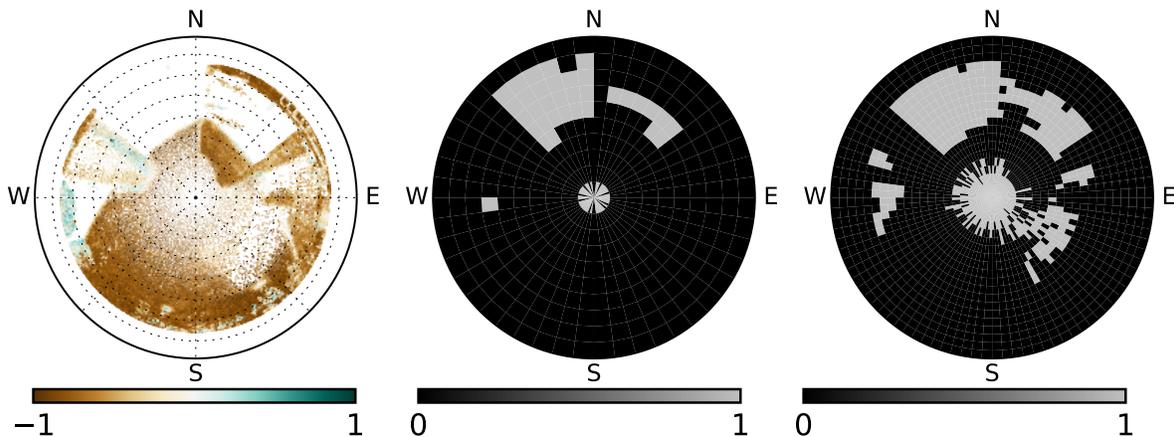


FIGURE 5.11 – Effect of the resolution on the binary mask  $\mathbf{B}$ . *Left* : point cloud of a roof and a pillar. *Middle* : Occupancy histogram with a resolution of  $7.5^\circ \times 9^\circ$ . *Right* : Occupancy histogram with a resolution of  $3.75^\circ \times 4.5^\circ$ .

A possible solution to this problem is reducing the size  $d_{\text{box}}$  of the bounding box of the octree filter or using ray tracing handling octree directly. This adjustment would increase the number of points avoiding the original cause of the problem.

## 5.8 Conclusion

In this paper, we proposed a new model for estimating the number of visible GNSS satellites, using a 3D point cloud map and future satellite constellation information. Our model takes into account both absorbing (e.g., trees) and occluding (e.g., building) structures, by only relying on the distribution and density of the 3D map points. We also designed a flexible data gathering platform that is able to function in harsh Canadian

winters. To estimate and validate parameters of the model, we moved our platform along several trajectories in diverse environments that featured dense forests and high buildings. Contrary to a pure masking approach based on the line-of-sight, our model was able to successfully predict the number of visible satellites in all these environments. The prediction can thus be useful for trajectory planning algorithms which rely on localization precision—the number of viable satellites influences the DOP. Future work includes explicit DOP prediction based on the geometry of the remaining satellites and their individual predicted SNR.

## Acknowledgements

We thank everyone at the Montmorency Forest for their help and their equipment. This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through grants CRDPJ 511843-17 BRITE (bus rapid transit system), CRDPJ 527642-18 SNOW (Self-driving Navigation Optimized for Winter) and FORAC. Finally, we also thank Maxime Vaidis and Charles Villeneuve for their help with the snowmobile.

# Conclusion

Comme expliqué dans le chapitre 1, les récepteurs DGNSS pourraient être une solution potentielle de localisation d'équipements lourds pour l'industrie forestière.

Étant donné que la géométrie de l'environnement a un impact important sur le type de localisation traitée dans ce mémoire, nous avons expliqué la génération des cartes de nuages de points dans le chapitre 2. Cette génération nous a permis de faire ressortir diverses données géométriques sur l'environnement qui sont importantes pour l'analyse finale.

Pour faire la récolte de données pouvant démontrer l'effet de l'environnement sur la localisation, nous avons conçu une plate-forme informatique, décrite dans le chapitre 3. Nous avons aussi complété cette plate-forme de plusieurs capteurs et d'une structure protectrice, afin qu'elle soit indépendante du véhicule sur lequel elle est installée.

La section 4.1, utilisant les données collectées par la plate-forme, nous aura ainsi démontré ce qu'il arrivait lorsque les récepteurs GNSS et DGNSS se retrouvent dans des endroits avec beaucoup d'obstruction. De plus, la section 4.2 montre que les résultats venant de la génération de carte expliquée dans le chapitre 2.

Comme ceci présente un futur problème pour l'industrie, nous avons donc développé une méthode, expliquée dans le chapitre 5, qui nous permet de mesurer la performance des DGNSS dans les endroits présentant beaucoup d'obstruction, tels un canyon urbain ou une forêt dense. Cette méthode, qui peut être utilisée comme heuristique, pourrait être utilisée pour la planification de trajectoire dans des environnements à haute obstruction. Nous croyons qu'elle pourrait aider à répandre l'utilisation de la technologie DGNSS.

# Bibliographie

- [1] H. SUN, D. SLAUGHTER, M. P. RUIZ, C. GLIEVER, S. UPADHYAYA et R. SMITH, “RTK GPS mapping of transplanted row crops”, *Computers and Electronics in Agriculture*, t. 71, n° 1, p. 32–37, 2010.
- [2] A. I. of AERONAUTICS et ASTRONAUTICS, *History of the GPS Program*, [https://www.aiaa.org/uploadedFiles/About\\_AIAA/Press\\_Room/Videos/IAF-60th-Anniv-GPS-Nomination.pdf](https://www.aiaa.org/uploadedFiles/About_AIAA/Press_Room/Videos/IAF-60th-Anniv-GPS-Nomination.pdf), 2011.
- [3] P. C. of EARTH et M. SCIENCES, *The Navigation Solution*, <https://www.e-education.psu.edu/geog862/node/1724>.
- [4] —, *GPS Error Sources*, <https://www.e-education.psu.edu/geog160/node/1924>.
- [5] F. A. ADMINISTRATION, *WIDE-AREA AUGMENTATION SYSTEM PERFORMANCE ANALYSIS REPORT*, <http://www.nstb.tc.faa.gov/REPORTS/waaspan17.pdf>, 2006.
- [6] —, *Current WAAS Vertical Navigation Service Snapshot Display*, [http://www.nstb.tc.faa.gov/RT\\_VerticalProtectionLevel.htm](http://www.nstb.tc.faa.gov/RT_VerticalProtectionLevel.htm), 2018.
- [7] R. ODOLINSKI, *Innovation : Low-cost single-frequency positioning approach*, <http://gpsworld.com/innovation-low-cost-single-frequency-positioning-approach-with-gps-bds-rtk/>, 2017.
- [8] P. C. of EARTH et M. SCIENCES, *Relative Positioning*, <https://www.e-education.psu.edu/geog862/node/1725>.

- [9] I. SIRF TECHNOLOGY, *NMEA Reference Manual*, <https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual1.pdf>, 2005.
- [10] GEO++, *Real-Time GNSS Data Transmission Standard RTCM 3.0*, [http://www.geopp.com/pdf/gppigs06\\_rtc\\_m\\_f.pdf](http://www.geopp.com/pdf/gppigs06_rtc_m_f.pdf), 2006.
- [11] L. RESOURCE SUPPLY, *IP67, What Does That Mean ?*, <http://www.resourcesupplyllc.com/PDFs/WhatDoesIP67Mean.pdf>.
- [12] F. POMERLEAU, M. LIU, F. COLAS et R. SIEGWART, “Challenging data sets for point cloud registration algorithms”, *Int. J. Robotics Res.*, t. 31, n° 14, p. 1705–1711, déc. 2012.
- [13] F. POMERLEAU, F. COLAS, R. SIEGWART et S. MAGNENAT, “Comparing ICP variants on real-world data sets”, *Autonomous Robots*, avr. 2013.
- [14] K.-L. LOW, “Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration”, jan. 2004.
- [15] A. SEGAL, D. HÄHNEL et S. THRUN, “Generalized-ICP”, *Proc. of Robotics : Science and Systems*, juin 2009.
- [16] S. RUSINKIEWICZ et M. LEVOY, “Efficient variants of the ICP algorithm”, p. 145–152, 2001.
- [17] K.-L. LOW, “Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration”, jan. 2004.
- [18] F. POMERLEAU, *libpointmatcher Tutorials*, <https://github.com/ethz-asl/libpointmatcher/blob/master/doc/index.md>.
- [19] J. ELSEBERG, D. BORRMANN et A. NÜCHTER, “Efficient processing of large 3D point clouds”, p. 1–7, 2011.
- [20] J. B. TIM FIELD Jeremy Leibs, *rosvag*, <http://wiki.ros.org/rosvag>.
- [21] CLEARPATH, *HUSKY*, <https://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>.
- [22] T. TAKASU, *RTKLIB ver. 2.4.2 Manual*, [http://www.rtklib.com/prog/manual\\_2.4.2.pdf](http://www.rtklib.com/prog/manual_2.4.2.pdf).

- [23] D. C. L. KEVIN W. HALL Peter Gagliardi, “GPS accuracy part 2 : RTK float versus RTK fixed”, *CREWES Research Report*, t. 22, 2010.
- [24] *U.S. Bureau of Statistics*, 2016.
- [25] H. LEE, K. C. SLATTON, B. E. ROTH et W. P. CROPPER, “Prediction of forest canopy light interception using three-dimensional airborne lidar data”, *Int. J. Remote Sens.*, t. 30, n° 1, p. 189–207, 2009.
- [26] W. WRIGHT, B. WILKINSON et W. CROPPER, “Development of a GPS forest signal absorption coefficient index”, *Forests*, t. 9, p. 226, avr. 2018.
- [27] Y. MORALES et T. TSUBOUCHI, “GPS moving performance on open sky and forested paths”, *IEEE IROS*, p. 3180–3185, 2007.
- [28] P. ATINÇ, “Accuracy analysis of GPS positioning near the forest environment”, *Croat. J. For. Eng.*, t. 29, p. 189–199, déc. 2008.
- [29] G. TAYLOR, J. LI, D. KIDNER, C. BRUNSDON et M. WARE, “Modelling and prediction of GPS availability with digital photogrammetry and lidar”, *Int. J. Geogr. Inf. Sci.*, t. 21, n° 1, p. 1–20, 2007.
- [30] G. ZHANG et L. HSU, “A new path planning algorithm using a GNSS localization error map for UAVs in an urban area”, *J. Intell. Robotic Syst.*, t. 95, n° 1, p. 219–235, 2019.
- [31] S. MIURA, L. HSU, F. CHEN et S. KAMIJO, “GPS error correction with pseudo-range evaluation using three-dimensional maps”, *IEEE ITS*, t. 16, p. 3104–3115, 2015.
- [32] Y. NG et G. XINGXIN GAO, “Direct position estimation utilizing non-line-of-sight (NLOS) GPS signals”, *ION GNSS+*, p. 1279–1284, nov. 2016.
- [33] W. WRIGHT, P.-W. LIU, K. SLATTON, R. SHRESTHA et al., “Predicting L-band microwave attenuation through Forest Canopy using directional structuring elements and airborne lidar”, *IGARSS*, t. 3, p. 688–691, 2008.

- [34] W. C. WRIGHT, B. E. WILKINSON et W. P. CROPPER, “Estimating signal loss in pine forests using hemispherical sky oriented photos”, *ECOL INFORM*, t. 38, p. 82–88, 2017.
- [35] N. HOLDEN, A. MARTIN, P. OWENDE et S. WARD, “A method for relating GPS performance to forest canopy”, *Int. J. of Forest Engineering*, t. 12, n° 2, 2001.
- [36] D. MAIER et A. KLEINER, “Improved GPS sensor model for mobile robots in urban terrain”, *ICRA*, p. 4385–4390, 2010.
- [37] Y. CHEN, L. ZHU, J. TANG, L. PEI et al., “Feasibility study of using mobile laser scanning point cloud data for GNSS line of sight analysis”, en, *MOB INF SYST*, t. 2017, p. 11, 2017.
- [38] T. SUZUKI, M. KITAMURA, Y. AMANO et T. HASHIZUME, “High-accuracy GPS and GLONASS positioning by multipath mitigation using omnidirectional infrared camera”, p. 311–316, 2011.
- [39] R. B. LANGLEY, “Dilution of precision”, *GPS world*, p. 52–59, mai 1999.
- [40] R. HE, S. PRENTICE et N. ROY, “Planning in information space for a quadrotor helicopter in a GPS-denied environment”, in *ICRA*, 2008, p. 1814–1820.