HAL open science

Nuage de Points et Modélisation 3D

Jean-Emmanuel Deschaud

To cite this version:

Jean-Emmanuel Deschaud. Nuage de Points et Modélisation 3D. Vision par ordinateur et reconnaissance de formes [cs.CV]. Sorbonne Université, 2020. tel-03916523

HAL Id: tel-03916523

https://hal.science/tel-03916523v1

Submitted on 30 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mémoire pour l'obtention de l'Habilitation à Diriger des Recherches

Sorbonne Université

UFR d'Ingénierie

Spécialité : Computer Science

Présentée par

Jean-Emmanuel Deschaud

Nuage de Points et Modélisation 3D

Soutenue le 17 Décembre 2020

Devant le jury composé de :

M. Paul CHECCHIN	PR, Université Clermont Auvergne	Rapporteur
M. David FILLIAT	PR, ENSTA Paris	Rapporteur
M. Pierre GRUSSENMEYER	PR, INSA de Strasbourg	Rapporteur
M. François GOULETTE	PR, MINES ParisTech	Examinateur
Mme Beatriz MARCOTEGUI	PR, MINES ParisTech	Examinatrice
M. Pascal MORIN	PR, Sorbonne Université	Examinateur
M. Nicolas PAPARODITIS	DR, Université Gustave Eiffel	Examinateur
M. Laurent TRASSOUDAINE	PR, Université Clermont Auvergne	Examinateur

Table des matières

I Synthèse scientifique **1**

1 Introduction **3**

2 Synthèse des travaux **7**
2.1 Systèmes mobiles d'acquisition de données 3D 8
2.1.1 Premiers prototypes de système mobile de cartographie : LARA3D version 2006 puis LARA3D version 2010 8
2.1.2 Améliorations de la calibration d'un système mobile de cartographie basé LiDAR 13
2.1.3 Nouveau prototype L3D2 et création des jeux de données Paris-rue-Madame et Paris-Lille-3D 18
2.1.4 Drone LiDAR 22
2.1.5 SLAM LiDAR 24
2.2 Analyse de nuages de points 30
2.2.1 Détection de primitives géométriques 30
2.2.2 Segmentation sémantique 31
2.3 Rendu de nuages de points, reconstruction 3D et simulation 48
2.3.1 Rendu de nuages de points 48
2.3.2 Reconstruction 3D 51
2.3.3 Simulation 58

3 Perspectives de recherche **69**
3.1 Systèmes mobiles d'acquisition de données 3D 69
3.2 Recalage, classification et sémantisation 71
3.3 Rendu de nuages de points, reconstruction 3D et simulation 72

II Parcours scientifique **75**

4 Parcours professionnel **77**

5 Encadrement et enseignement **79**
5.0.1 Encadrement 79
5.0.2 Enseignement 81

6 Diffusion et liste des publications **83**
6.1 Diffusion 83
6.1.1 Création de la start-up Terra3D 83
6.2 Projets de recherche 84
6.3 Liste des publications 85

6.3.1 Articles de revues avec comité de lecture 85
6.3.2 Articles de conférence avec comité de lecture 85

Bibliographie **89**

Première partie

Synthèse scientifique

Chapitre 1

Introduction

La donnée "Nuage de points" est un type de donnée produit par des capteurs ou des procédés de plus en plus répandus et performants (LiDARs fixes, mobiles, caméras RGB-D, photogrammétrie...). Ces données se trouvent généralement sous la forme d'une liste de points avec les coordonnées géométriques x, y, z et parfois d'autres attributs (réflectance, couleur...). De la même façon que pour les images, de nombreux traitements sont nécessaires pour analyser et transformer ces données.

Ces nuages de points sont une représentation géométrique de scènes ou d'objets réels. Un enjeu de recherche qui m'a toujours fortement motivé dans mes travaux (avec de nombreuses applications potentielles) est la capacité à transformer ces données pour modéliser le réel en étant le plus réaliste et fidèle possible (par exemple la simulation d'une ville comme Paris depuis le sol dans un casque de réalité virtuelle).

Ce mémoire d'HDR présente un panorama des travaux de recherche que j'ai menés et encadrés depuis ma thèse sur les nuages de points. Ces travaux se situent à l'intersection de plusieurs communautés de recherche : la robotique, la photogrammétrie/lasergrammétrie, la communauté *Computer Vision* et enfin la communauté de l'informatique graphique. La possibilité de rencontrer et d'interagir avec ces différentes communautés a toujours été pour moi très enrichissant.

Lors de ma thèse de 2007 à 2010, j'ai principalement travaillé sur des données 3D acquises par des systèmes mobiles de cartographie équipés d'un capteur LiDAR. Ces systèmes innovants permettent d'acquérir de grands volumes de données de scènes extérieures ce qui a poussé les recherches sur des traitements de plus en automatisés : segmentation sémantique, reconstruction de surfaces... La Figure 1.1 montre un nuage de points 3D acquis par un système mobile basé LiDAR.

Depuis, j'ai continué à travailler sur les systèmes mobiles d'acquisition de données 3D tout en diversifiant les traitements de nuages de points. On peut regrouper mes recherches en 3 axes principaux :

— Systèmes mobiles d'acquisition de données 3D
— Analyse de nuages de points
— Rendu de nuages de points, reconstruction 3D et simulation

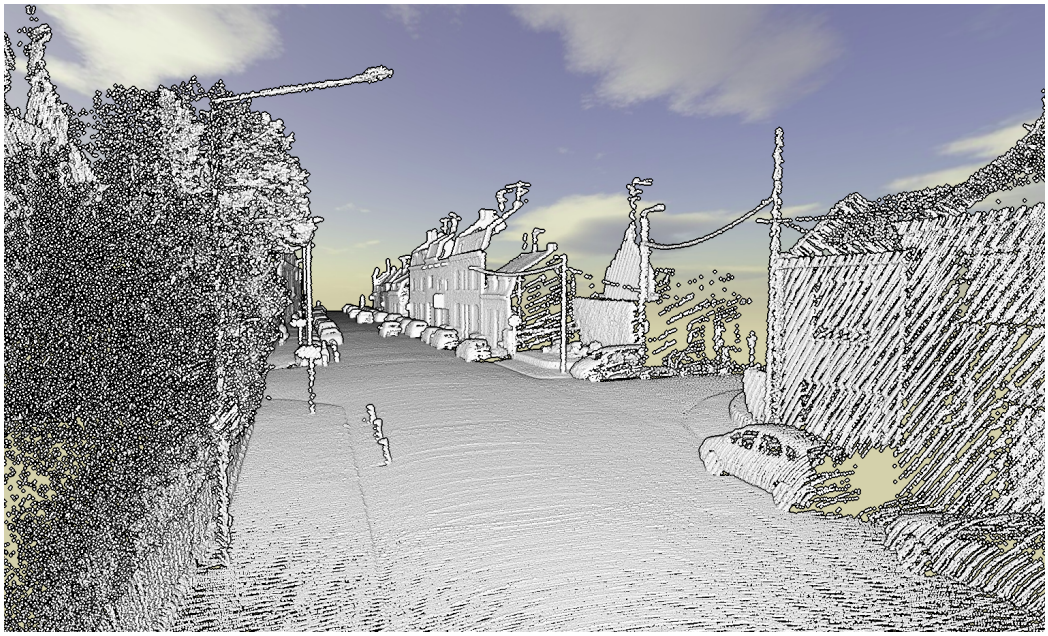

FIGURE 1.1 – Exemple d'un nuage de points produit par le système mobile basé LiDAR de MINES ParisTech. Acquisition d'une rue de Lille Métropole.

Pour le premier axe, nous avons travaillé avec un doctorant sur une méthode permettant de retrouver les paramètres de calibration extrinsèque et intrinsèque d'un LiDAR sur un système mobile de cartographie à partir seulement du nuage de points consolidé et sans mire de calibration. J'ai aussi exploré la piste d'un SLAM (Simultaneous Localization And Mapping) basé LiDAR permettant de produire une donnée 3D très précise.

Pour le deuxième axe, deux doctorants que j'ai encadrés ont étudié les algorithmes de segmentation sémantique de nuages de points, c'est à dire la capacité à donner une classe à chaque point du nuage. Il s'agit d'une étape importante pour de nombreuses applications. Lors des derniers travaux publiés en 2019, nous avons proposé une convolution par points permettant d'appliquer une architecture de type U-Net pour de la segmentation sémantique directement sur le nuage de points (sans représentation intermédiaire de type images ou voxels).

Pour le troisième axe, nous avons développé plusieurs applications exploitant les nuages de points de systèmes mobiles. La première application va, à partir d'un nuage de points, reconstruire un modèle 3D (sous forme de maillage surfacique) de l'environnement routier pour estimer (avec des techniques de lancer de rayon) les distances de visibilité sur route. Dans une deuxième application, nous avons extrait le sol puis, à l'aide d'un algorithme de reconstruction de surface très rapide, nous avons pu produire des surfaces précises de la route sous forme de maillages,

exploitables dans des moteurs de rendu pour de la simulation de véhicule de conduite. Nous verrons aussi comment nous avons modélisé l'environnement pour permettre une simulation réaliste d'un capteur LiDAR à partir de données LiDAR. Enfin, nous avons imaginé comment utiliser directement un nuage de points pour permettre la simulation d'écoulement de fluides autour de la géométrie représentée par la donnée 3D.

Je vais développer ces 3 axes dans le Chapitre 2. Ces travaux ont été menés au sein de l'équipe NPM3D (Nuage de Points et Modélisation 3D) du Centre de Robotique de MINES ParisTech avec le Professeur François Goulette (le directeur de thèse de toutes les thèses que j'ai encadrées) qui a créé l'activité "Cartographie mobile" avant mon arrivée. Une collaboration forte a aussi eu lieu sur certains travaux avec la Professeure Beatriz Marcotegui du Centre de Morphologie Mathématique de MINES ParisTech.

Lors des travaux de recherche présentés dans ce manuscrit, j'ai aussi tenté dans la mesure du possible d'appliquer les résultats à une grande variété de nuage de points provenant de systèmes et de contextes variés avec quelques exemples dans la Figure 1.2.

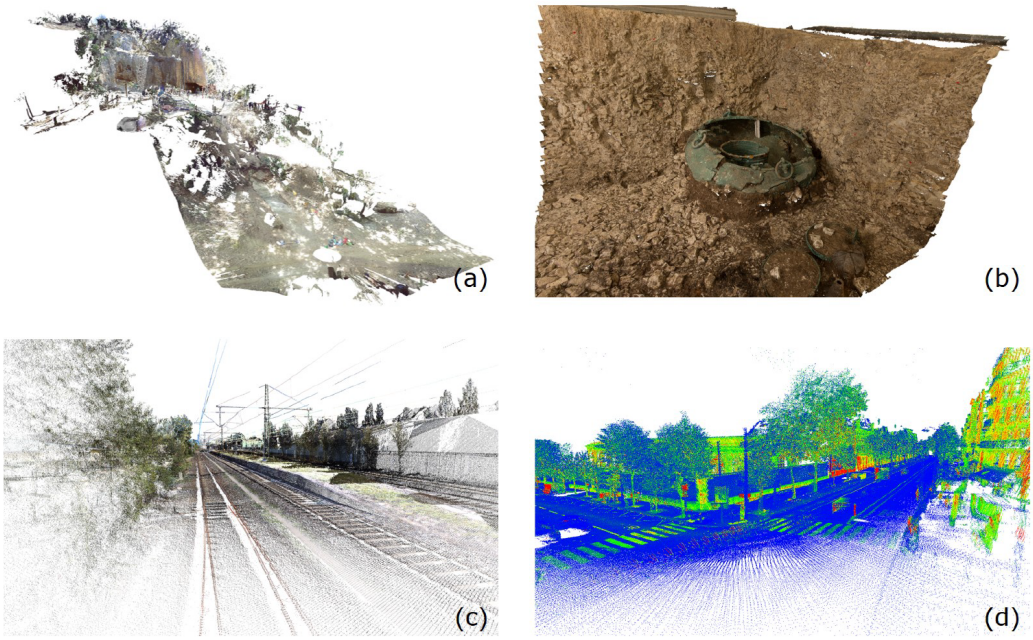

FIGURE 1.2 – Exemples de nuages de points traités dans les travaux présentés dans ce manuscrit. (a) Nuage de points coloré de la façade d'une tombe étrusque produit par plusieurs stations d'un LiDAR fixe FARO X130. (b) Nuage de points coloré obtenu par photogrammétrie d'un chaudron retrouvé dans une tombe celtique. (c) Nuage de points coloré d'une voie de chemin de fer par scan mobile avec un capteur LiDAR mono-fibre RIEGL. (d) Nuage de points avec information de réflectance d'une rue à Paris par scan mobile avec un capteur LiDAR multi-fibres Velodyne HDL32.

Chapitre 2

Synthèse des travaux

Contents

2.1 Systèmes mobiles d'acquisition de données 3D 8

2.1.1 Premiers prototypes de système mobile de cartographie : LARA3D version 2006 puis LARA3D version 2010 8

2.1.2 Améliorations de la calibration d'un système mobile de cartographie basé LiDAR 13

2.1.3 Nouveau prototype L3D2 et création des jeux de données Paris-rue-Madame et Paris-Lille-3D 18

2.1.4 Drone LiDAR 22

2.1.5 SLAM LiDAR 24

2.2 Analyse de nuages de points 30

2.2.1 Détection de primitives géométriques 30

2.2.2 Segmentation sémantique 31

2.3 Rendu de nuages de points, reconstruction 3D et simulation 48

2.3.1 Rendu de nuages de points 48

2.3.2 Reconstruction 3D 51

2.3.3 Simulation 58

Nous allons décrire dans ce chapitre une synthèse des travaux que j'ai menés ces dix dernières années. Nous pouvons les regrouper en 3 grandes catégories qui suivent le processus de génération de données qui va de l'acquisition à la création du modèle final.

Dans la première section, nous verrons les méthodes d'acquisition, principalement autour des systèmes mobiles de cartographie basé LiDAR, les approches pour calibrer les capteurs et enfin des travaux récents sur de l'odométrie LiDAR pour la cartographie et le véhicule autonome.

Dans la deuxième section, nous verrons comment analyser les données sous forme de nuage de points 3D : le segmenter en différents sous-nuages avec une cohérence géométrique ou lui donner une information sémantique de classes à chaque point ou groupe de points.

Enfin, dans la dernière section, nous traiterons des méthodes qui transforment le nuage de points en un modèle. Ce modèle va bien sûr très fortement dépendre de

l'application. Cela va des modèles surfaciques de route pour faire de la simulation précise et fidèle dans le domaine automobile jusqu'aux modèles implicites pour de la simulation d'écoulements de fluides.

2.1 Systèmes mobiles d'acquisition de données 3D

Une partie de mes travaux de recherche porte sur l'acquisition de données 3D. Il s'agit plus particulièrement de prototypes d'acquisitions mobiles terrestres basé LiDAR (véhicule et drone). Ces systèmes permettent de produire de grandes quantités de données 3D augmentant le champ des applications mais aussi les besoins sur la capacité de traiter ces grands volumes de données de façon de plus en plus automatisées.

2.1.1 Premiers prototypes de système mobile de cartographie : LARA3D version 2006 puis LARA3D version 2010

Le Centre de Robotique de MINES ParisTech a été précurseur dans les systèmes mobiles de cartographie basé LiDAR, avec des premiers travaux dès 2003. Le principe consistait à détourner les capteurs LiDARs monocouche utilisés à l'horizontale à l'avant des véhicules pour les mettre à l'arrière à la verticale de façon à scanner l'environnement (voir Figure 2.1). Seuls quelques laboratoires dans le monde travaillaient sur le même principe comme [Fruh & Zakhor 2001] de l'Université de Berkeley ou [Zhao & Shibasaki 2003] de l'Université de Tokyo. La capacité de ces véhicules terrestres à scanner de grands environnements était déjà très prometteuse par rapport à des scanners fixes. La Figure 2.1 présente la version 2006 de LARA3D, véhicule prototype de cartographie mobile basée LiDAR.

Mais le fait d'avoir un véhicule mobile rend la tâche de création de la donnée 3D complexe.

La première difficulté consiste à localiser précisément le véhicule (avec ses 6 degrés de liberté). La solution consiste à utiliser de nombreux capteurs (généralement coûteux) et à fusionner leurs données. On utilise principalement une centrale inertielle (ou IMU) pour mesurer les accélérations et les vitesses angulaires du véhicule (et en déduire par intégration le déplacement du véhicule) et un GPS (qui permet de connaître la position du véhicule en coordonnées géographiques). La centrale inertielle est bruitée et biaisée : elle va donc donner une position qui va dériver avec le temps alors que le GPS donne une position absolue mais dépendant fortement de la qualité de réception des signaux satellites (très dégradé en environnement urbain).

La deuxième difficulté vient de la calibration des capteurs (calcul de la transformation rigide entre les repères des différents capteurs) : premièrement entre les

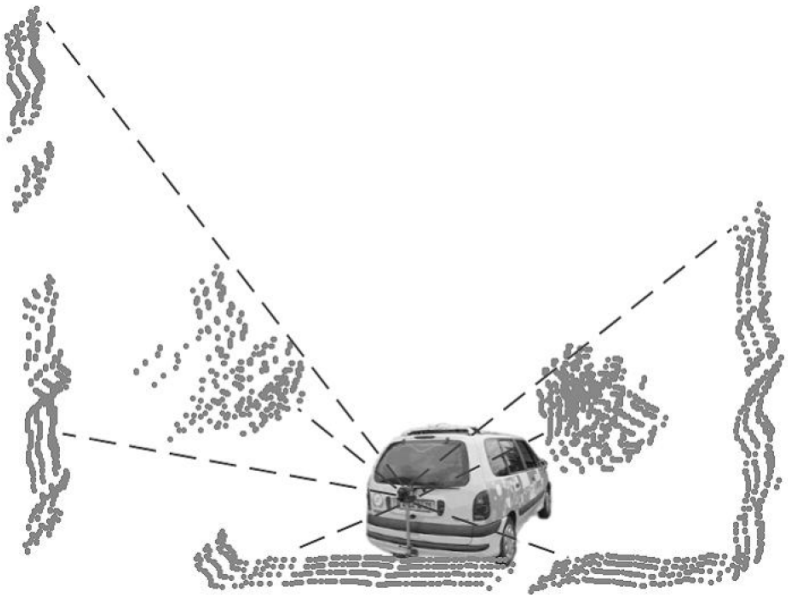

FIGURE 2.1 – Système mobile de cartographie LARA3D version 2006 de MINES ParisTech.

capteurs de navigation (GPS <-> IMU) mais aussi entre les capteurs de perception (LiDARs, caméras) et l'IMU (qui sert généralement de repère commun pour tous les capteurs). Nous verrons des travaux que nous avons menés de 2013 à 2016 pour améliorer cette calibration.

La Figure 2.2 montre un des premiers résultats (nuage de points de la Rue Soufflot à Paris) produit par le véhicule prototype LARA3D de 2006. LARA3D était alors équipé d'un GPS Trimble AG132 (utilisant le positionnement différentiel EGNOS), une IMU Crossbow VG600, un LiDAR IBEO fonctionnant à 10Hz (avec 1080 mesures de distances par rotation) et enfin une caméra Fish-Eye Allied Vision Marlin. On peut voir sur l'image que la densité du nuage de points était déjà intéressante mais la donnée comporte de nombreux défauts visibles (par exemple des zigzags verticaux au niveau des balcons qui vient d'une mauvaise localisation du véhicule).

L'article [Brun *et al.* 2007] de 2006 (correspondant à des travaux de mon stage aux MINES avec le doctorant Xavier Brun) montre des premiers résultats de reconstruction 3D puis de texturation à partir des images de la caméra (voir Figure 2.3). L'article [Deschaud *et al.* 2010] explique plus en détails le processus de colorisation et de texturation.

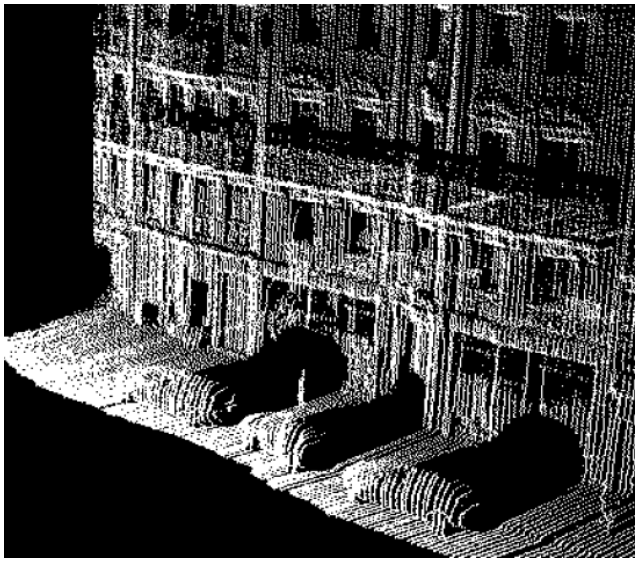

FIGURE 2.2 – Nuage de Points de la Rue Soufflot à Paris produit avec le véhicule prototype LARA3D de 2006.

FIGURE 2.3 – Modèle 3D texturé de la Rue Saint-Jacques à Paris produit avec les données du véhicule prototype LARA3D de 2006 (laser + image).

De 2007 à 2010, j'ai travaillé pendant ma thèse à MINES ParisTech sur l'amélioration du processus d'acquisition et de traitement des nuages de points pour produire des modèles 3D plus précis et plus denses. Les différentes améliorations portaient sur de nouveaux capteurs de perception (LiDARs SICK LMS 221, caméras Canon EOS 5D), un meilleur positionnement (fusion GPS-IMU-odométrie des roues), une meilleure calibration LiDAR/caméra et enfin de nouveaux traitements (calcul de normales, reconstruction de surface et texturation de modèles 3D). La Figure 2.4 montre le système résultant et la Figure 2.5 montre le fonctionnement

du système.

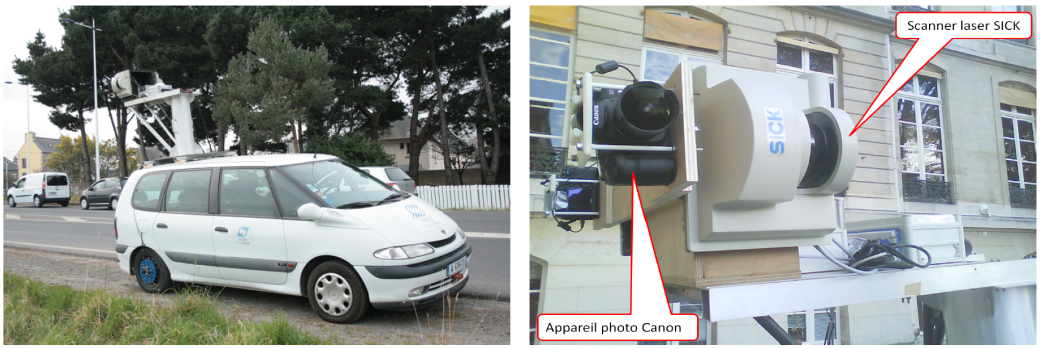

FIGURE 2.4 – Système mobile de cartographie LARA3D version 2010 de MINES ParisTech.

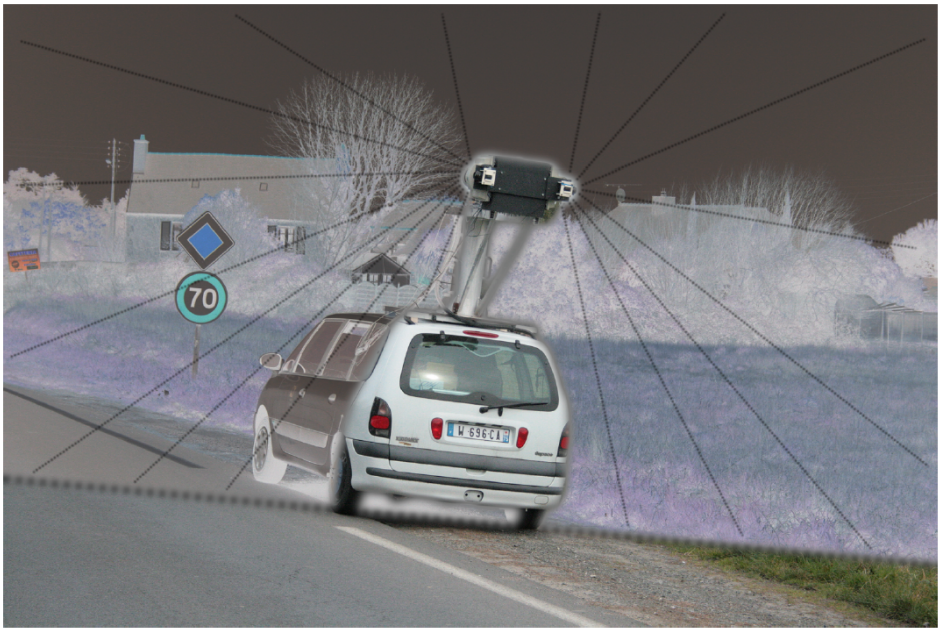

FIGURE 2.5 – Numérisation par coupe transversale avec le système prototype LARA3D version 2010.

La Figure 2.6 montre un exemple de nuage de points et de photos pris avec ce système LARA3D version 2010.

Pour modéliser des scènes d'environnements urbains et routiers, nous avons choisi ce qui nous semble être le meilleur compromis entre la généricité, la fidélité et la compacité des modèles 3D avec une modélisation hybride sous la forme d'un

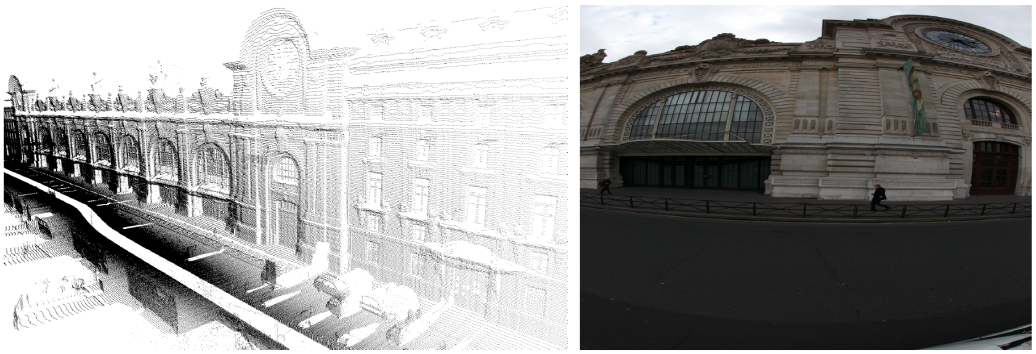

FIGURE 2.6 – A gauche, nuage de points de la façade du Musée d'Orsay acquis avec le prototype LARA3D version 2010. A droite, une image de l'appareil photo Canon EOS 5D prise pendant l'acquisition.

maillage et de primitives planes. La Figure 2.7 montre le modèle 3D texturé de la façade du Musée d'Orsay à Paris. Lors de l'étape de texturation, chaque sommet d'un triangle est projeté dans l'image lui offrant la meilleure vue, c'est-à-dire maximisant l'aire du triangle projeté.

FIGURE 2.7 – Modèle 3D texturé de la façade du Musée d'Orsay à Paris produit à partir du véhicule prototype LARA3D de 2010.

2.1.2 Améliorations de la calibration d'un système mobile de cartographie basé LiDAR

Une grande source d'erreur sur les données provenant des systèmes mobiles est la calibration des capteurs. Il s'agit d'un problème complexe qui est souvent résolu en pratique par des mesures topographiques. Cette opération est longue et nécessite d'être répétée à chaque modification de la configuration des capteurs.

L'image de la Figure 2.8 montre les différents repères : repère de navigation (IMU), repère LiDAR et repère monde impliqués dans la création de données provenant d'un système mobile de cartographie.

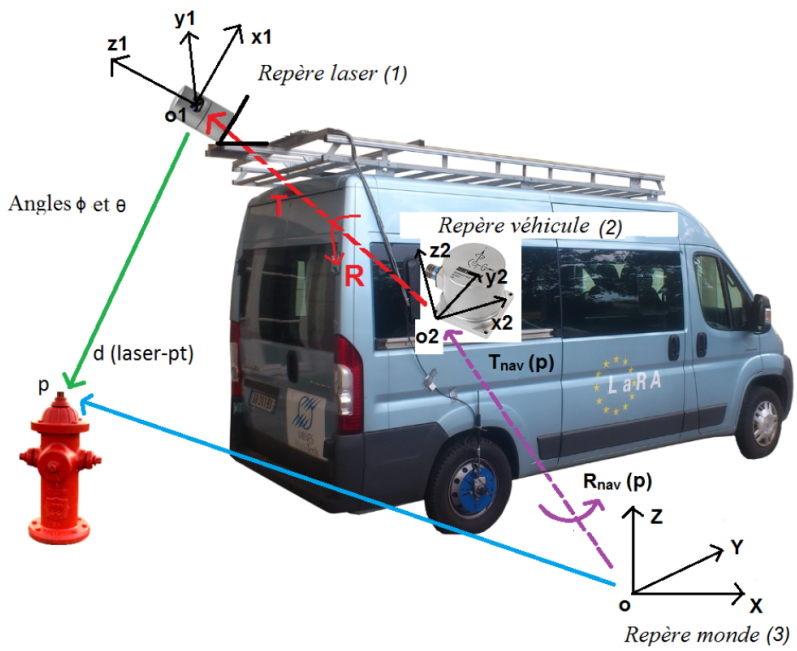

FIGURE 2.8 – Repères de navigation (IMU), LiDAR et monde pour la création de données 3D à partir d'un système mobile de cartographie.

Nous avons pu remarquer avec LARA3D que la calibration entre l'IMU et le LiDAR est particulièrement importante (R et t dans la Figure 2.8). Il existe des procédures semi-automatiques à partir de mires de calibration comme dans [Huang *et al.* 2013] mais cela reste aussi fastidieux.

Plus récemment, des approches entièrement automatiques ont vu le jour. Le principe est de définir une énergie représentant la "qualité" du nuage de points et de minimiser cette énergie suivant les paramètres de calibration. Cette "qualité" est mesurée localement par la planarité des surfaces représentées par les points. Cette hypothèse est généralement vérifiée dans le cas d'acquisition de données en

environnement urbain avec par exemple le sol ou des façades d'immeubles (qui sont localement planaires). La deuxième hypothèse de cette approche est d'avoir un nuage de points dense (pour avoir suffisamment de données pour une estimation correcte du plan local). L'arrivée des LiDARS multi-couches (ou multi-fibres) comme le Velodyne HDL64 à partir de 2006 a permis de produire des données vérifiant cette deuxième hypothèse (HDL64 = 64 fibres, HDL32 = 32 fibres laser). Une des premières approche suivant cette idée est celle de [Levinson & Thrun 2014]. Ils utilisent la métrique *point-to-plane* et trouvent les paramètres de calibration en minimisant la projection de chaque point du nuage sur son plan local.

Dans le cadre d'un projet de recherche appelé TerraMobilita, nous avons amélioré cette méthode avec les travaux de thèse de Houssem Nouira publiés dans [Nouira *et al.* 2015], [Nouira *et al.* 2016] et [Nouira *et al.* 2017]. Les principales contributions par rapport à la méthode de [Levinson & Thrun 2014] sont :

- Optimisation conjointe des paramètres intrinsèques et extrinsèques du LiDAR
- Ajout d'un poids de planarité dans l'énergie
- Optimisation par Gauss-Newton (alors que [Levinson & Thrun 2014] effectuait une recherche par grille)

Nous avons ainsi une énergie du nuage de point définie par :

$$E = \sum_{i=1}^{B} \sum_{j=i-N, j\neq i}^{i+N} \sum_{k} w_{i,j,k} (n_{i,k} \cdot (p_{i,k} - m_{j,k}))^2$$

$$p_{i,k} = R_{nav}(t_{p_{i,k}})(\boldsymbol{R}\, p'_{i,k} + \boldsymbol{t}) + t_{nav}(t_{p_{i,k}})$$

$$m_{j,k} = R_{nav}(t_{m_{j,k}})(\boldsymbol{R}\, m'_{j,k} + \boldsymbol{t}) + t_{nav}(t_{m_{j,k}})$$

avec :

— B le nombre de fibres du LiDAR ($B = 32$ pour le Velodyne HDL32)
— i va itérer sur toutes les fibres du LiDAR
— N le nombre de fibres voisines qui vont être prises en compte (les fibres sont numérotées par ordre croissant verticalement) (nous prendrons $N = 2$)
— j va itérer sur les $2N$ fibres voisines
— k va itérer sur un sous-échantillonnage des points de la fibre i
— $p_{i,k}$ et $m_{j,k}$ sont respectivement, le $k$ième point de la fibre i dans le repère monde (et son plus proche voisin de la fibre j dans le repère monde)
— $p'_{i,k}$ et $m'_{j,k}$ correspondent aux points $p_{i,k}$ et $m_{i,k}$ dans le repère LiDAR
— $R_{nav}(t)$ et $t_{nav}(t)$ sont respectivement les matrices de rotation et les vecteurs de translation pour passer du repère de navigation (IMU) au repère monde. Ces transformations sont fournies par le post-traitement des données de navigation (IMU et GPS) et dépendent du temps (elles sont donc différentes pour chaque point $p_{i,k}$ en fonction du *timestamp* $t_{p_{i,k}}$ du point)
— $w_{i,j,k}$ est un poids qui varie de 0 à 1 et qui va dépendre de la distance entre les deux points $p_{i,k}$ et $m_{i,k}$ mais aussi de la planarité du voisinage de $p_{i,k}$

— $n_{i,k}$ correspond à la normale au point $p_{i,k}$ (calculée par voisinage en prétraitement)
— $\boldsymbol{R}$ et $\boldsymbol{t}$ sont la matrice de rotation et le vecteur de translation pour passer du repère LiDAR au repère de navigation (IMU). Ce sont les paramètres extrinsèques de calibration que nous cherchons à obtenir. La matrice R est paramétrée par 3 angles d'Euler $\boldsymbol{R} = R_z(\boldsymbol{\gamma})R_y(\boldsymbol{\beta})R_x(\boldsymbol{\alpha})$

En plus de chercher la calibration entre le LiDAR et l'IMU (paramètres extrinsèques), il est possible d'utiliser cette même formulation pour trouver aussi les paramètres intrinsèques de calibration du LiDAR.

En effet, en s'inspirant du modèle du capteur Velodyne HDL64 décrit par [Glennie & Lichti 2010] et du modèle capteur Velodyne HDL32 décrit par [Chan & Lichti 2013], nous avons défini notre propre modèle du Velodyne HDL32 et écrivons le point $p'_{i,k}$ sous la forme :

$$p'_{i,k} = \begin{pmatrix} (\rho_{i,k} + \boldsymbol{\delta\rho_i})\cos(\theta_{i,k} + \boldsymbol{\delta\theta_i})\cos(\phi_i + \boldsymbol{\delta\phi_i}) \\ -(\rho_{i,k} + \boldsymbol{\delta\rho_i})\sin(\theta_{i,k} + \boldsymbol{\delta\theta_i})\cos(\phi_i + \boldsymbol{\delta\phi_i}) \\ (\rho_{i,k} + \boldsymbol{\delta\rho_i})\sin(\phi_i + \boldsymbol{\delta\phi_i}) + \boldsymbol{h_i} \end{pmatrix}$$

avec :
— $p'_{i,k}$ le $k$ième point de la fibre i dans le repère du capteur LiDAR
— $\rho_{i,k}$ la distance entre l'origine du capteur LiDAR et le point $p'_{i,k}$
— $\theta_{i,k}$ l'angle horizontal mesuré par l'encodeur pour le point $p'_{i,k}$ lors de la rotation du LiDAR
— ϕ_i l'angle vertical de la fibre i (qui ne dépend pas de k)
— $\boldsymbol{\delta\rho_i}$, $\boldsymbol{\delta\theta_i}$ et $\boldsymbol{\delta\phi_i}$ sont les paramètres intrinsèques pour chaque fibre i avec respectivement les corrections sur les distances, les angles horizontaux et verticaux
— $\boldsymbol{h_i}$ est un paramètre intrinsèque représentant une correction suivant z pour chaque fibre i dans le repère capteur

Une des fibres est prise comme fibre de référence (en l'occurrence la fibre 15 étant la fibre horizontale) et n'a pas de paramètre intrinsèque. Cela donne un total de $31 * 4 = 124$ paramètres intrinsèques. La Figure 2.9 montre la répartition verticale des fibres laser du Velodyne HDL32.

Les travaux de [Demantké *et al.* 2011] permettent d'associer une dimensionnalité à un voisinage d'un point dans un nuage de points. On calcule pour cela trois termes a_{1D}, a_{2D} et a_{3D} en fonction des points du voisinage (voisinage sphérique de rayon fixe dans notre cas). Le terme le plus grand $d^* = \text{argmax}_i(a_{iD})$ va définir la dimensionnalité du voisinage. L'image de la Figure 2.10 montre le calcul de la dimensionnalité sur un nuage acquis par système mobile LiDAR. On voit que a_{1D} est le plus grand pour les voisinages linéaires (câbles par exemple), a_{2D} est le plus grand pour les voisinages planaires (façades d'un immeuble...) et enfin a_{3D} est le

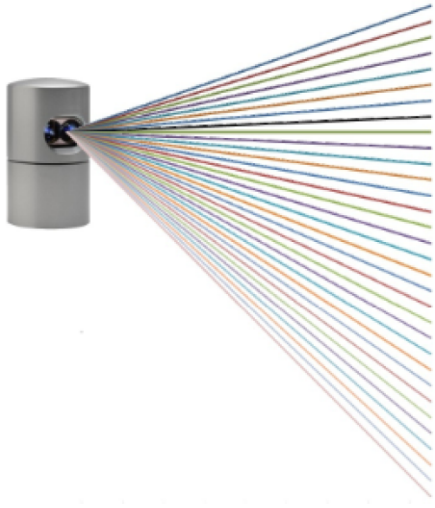

FIGURE 2.9 – Répartition des fibres laser du Velodyne HDL32.

plus grand pour les voisinages volumiques (la végétation par exemple).

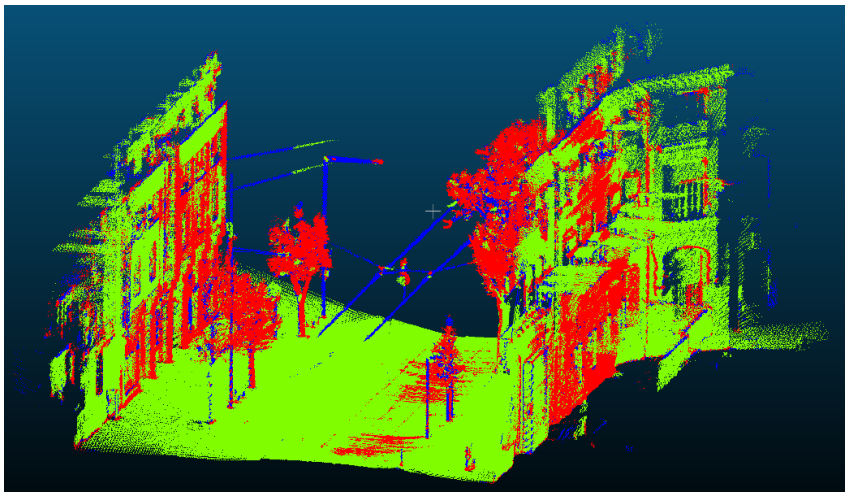

FIGURE 2.10 – Nuage de points 3D d'une scène urbaine coloré par dimensionnalité (en bleu les points de dimension 1, en vert les points de dimension 2 et enfin en rouge les points de dimension 3).

Nous avons utilisé cette dimensionnalité dans notre méthode de calibration pour donner plus de poids aux points qui ont un voisinage planaire (score a_{2D} élevé) :

$$w_{i,j,k} = \begin{cases} \max(a_{2D}(p_{i,k}), a_{2D}(m_{j,k})) & \text{si} \quad ||p_{i,k} - m_{j,k}|| < d_{\max} \\ 0 & \text{sinon.} \end{cases}$$

avec $d_{\max}$ un paramètre de la méthode sur la distance maximale entre deux points du nuage qui serviront à la calibration (nous prendrons $d_{\max} = 0.20m$).

La minimisation de l'énergie E est un problème de moindre carré non linéaire et peut donc se résoudre par Gauss-Newton. La méthode Gauss-Newton est une méthode itérative qui demande de résoudre à chaque étape un système linéaire de la forme $J_E^T J_E \delta x = J_E^T b$ avec J_E la matrice jacobienne associée à E et b le vecteur associé à l'erreur. Nous effectuons la résolution de tous les paramètres (extrinsèques et intrinsèques) de façon conjointe. L'optimisation complète s'effectue en 35 min pour un nuage d'environ 10 millions de points.

Pour valider notre approche, nous avons créé un jeu de données synthétique de nuage de points (en simulant un LiDAR Velodyne HDL32 en mouvement dans une scène géométrique simple composée de plans). Avec un nuage de points simulé, nous pouvons ajouter une erreur et étudier le comportement de notre méthode comparée à celle de [Levinson & Thrun 2014]. La Figure 2.11 présente les résultats sur un nuage de points synthétique.

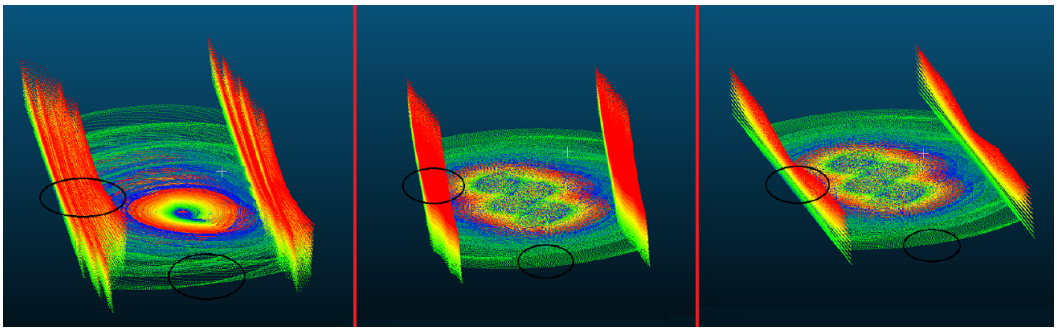

FIGURE 2.11 – A gauche, nuage de points simulé avec des erreurs ajoutées dans les paramètres extrinsèques de calibration. Au milieu, nuage de points obtenu après calibration par la méthode de [Levinson & Thrun 2014]. A droite, nuage de points obtenu après calibration par notre méthode.

La Figure 2.12 montre le résultat de la calibration sur un nuage de points réel obtenu par le système mobile de MINES ParisTech. On peut clairement voir l'amélioration dans la qualité de nuage de points (façades, voiture...).

La Figure 2.13 montre l'intérêt d'effectuer une optimisation conjointe des paramètres intrinsèques et extrinsèques avec des tests effectués sur le jeu de données synthétique. De plus, nos tests nous ont montré que nous n'avons pas besoin d'avoir une initialisation précise des paramètres pour que notre optimisation converge.

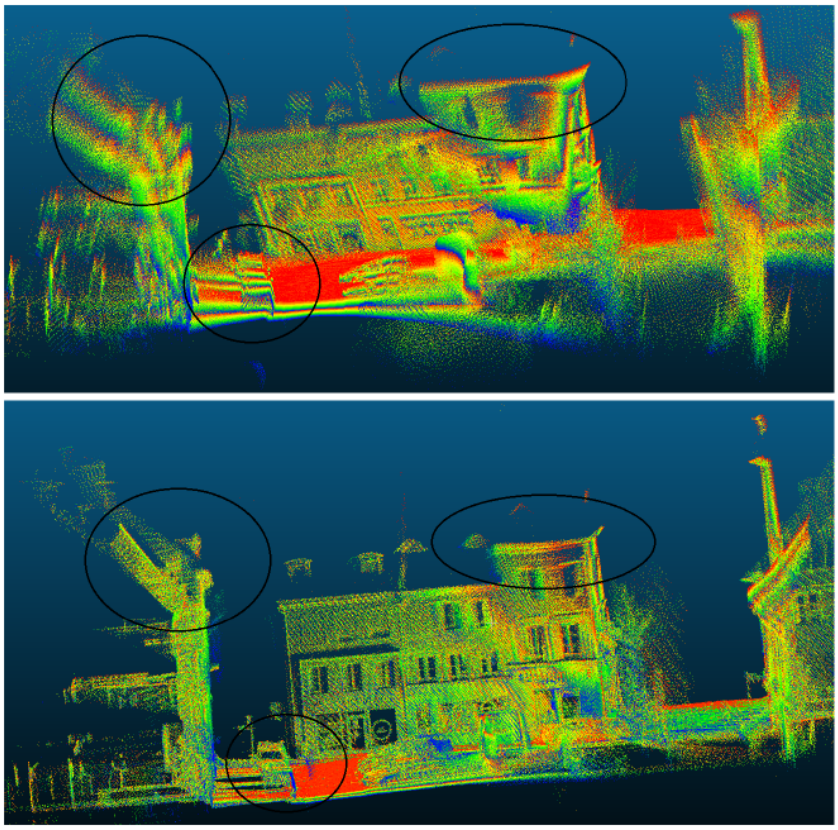

FIGURE 2.12 – Au-dessus, nuage de points sans calibration. En dessous, nuage de points après calibration par notre méthode automatique.

2.1.3 Nouveau prototype L3D2 et création des jeux de données Paris-rue-Madame et Paris-Lille-3D

En 2010, le véhicule prototype LARA3D a été remplacé par un nouveau véhicule prototype appelé L3D2. Depuis 2012 (mon retour à MINES ParisTech), j'ai participé à l'amélioration de ce système mobile de cartographie basé LiDAR. Il s'agit d'un investissement important au niveau ingénierie pour obtenir un système opérationnel.

Les données récoltées à partir de ce nouveau système ont permis en 2014 la publication d'un premier jeu de données 3D appelé "Paris-rue-Madame". Ce jeu de données a été publié dans [Serna *et al.* 2014] et a aidé la communauté pour comparer des méthodes de segmentation et de classification de nuages de points. Il est toujours disponible à l'url suivant : `http://www.cmm.mines-paristech.fr/~serna/rueMadameDataset.html`.

En 2018, nous sommes allés plus loin dans la création d'un jeu de données beaucoup plus volumineux appelé "Paris-Lille-3D". Ce jeu de données est com-

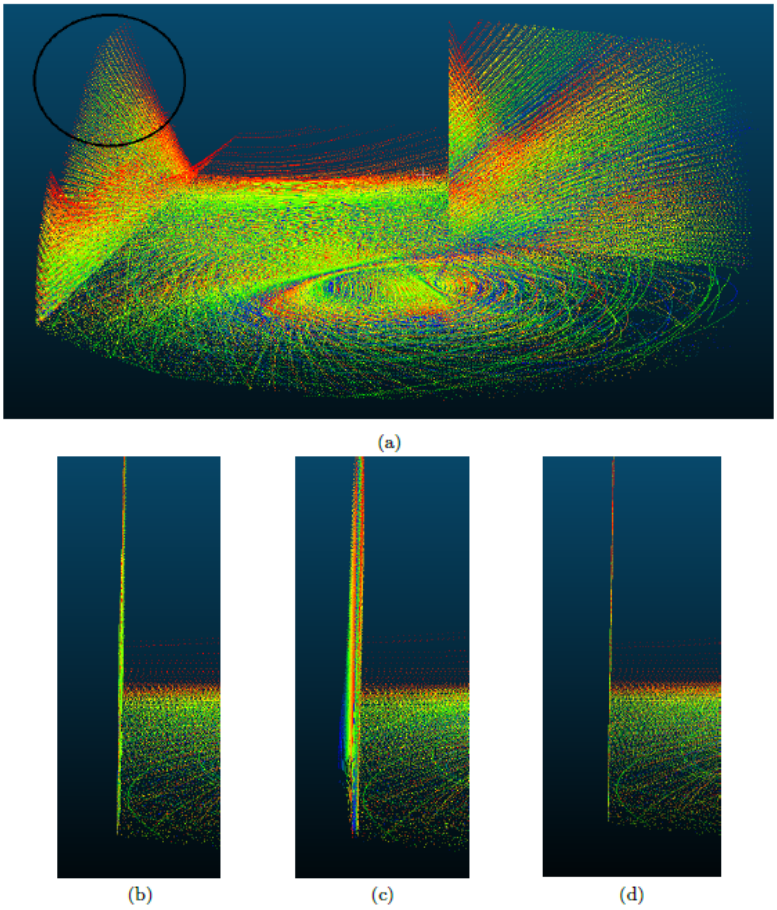

FIGURE 2.13 – (a) Nuage de points synthétique. (b) Vue sur le côté du nuage avec optimisation des paramètres extrinsèques puis optimisation des paramètres intrinsèques de calibration (c) Optimisation intrinsèque puis extrinsèque (d) Optimisation conjointe des paramètres de calibration

posé de plusieurs nuages de points de Paris et Lille avec un total de plus de 140 millions de points labélisés à la main en 50 classes (sol, voitures, barrières, poteaux, piétons, panneaux de signalisation...). Ce jeu de donnée est suffisamment volumineux et de qualité pour aider les travaux de recherches sur la classification et la segmentation sémantique de nuages de points (que nous verrons en section 2.2).

L3D2 était équipé en 2018 (voir Figure 2.14) d'une centrale inertielle LANDINS d'iXblue, d'un GPS Novatel FlexPak 6 et d'un LiDAR Velodyne HDL32 (positionné à l'arrière pour faire de la cartographie). La localisation du véhicule est basée sur le post-traitement des données GPS et IMU avec le logiciel Novatel Inertial Explorer.

Le jeu de données Paris-Lille-3D publié dans [Roynard *et al.* 2018c] est composé

FIGURE 2.14 – Prototype L3D2 version 2018 qui a été utilisé pour produire le jeu de données Paris-Lille-3D

de 3 nuages ("Lille1.ply", "Lille2.ply" et "Paris.ply"). La Figure 2.15 montre une partie d'un des nuages de points du jeu de données avec la couleur représentant les différentes classes labélisées entièrement à la main. Le Tableau 2.1 montre une comparaison du jeu de données Paris-Lille-3D avec d'autres jeux de données du monde académique.

FIGURE 2.15 – Une partie d'un des nuages de points du jeu de données Paris-Lille-3D. La couleur représente les classes labélisées à la main (une couleur par classe).

Le jeu de données produit a été rendu publique sur le site NPM3D : `https://npm3d.fr/paris-lille-3d`. Ce site web propose un *benchmark* en ligne de segmentation sémantique de nuages de points (sur un nombre plus restreint de 9 classes). Pour cela, nous avons produit 3 autres nuages de points provenant de deux autres villes Dijon et Ajaccio (voir Figure 2.16) avec une labélisation faite à la

Nom	Type de capteur	Surface couverte (longueur)	Nombre de points	Nombre de classes
S3DIS	caméra RGB-D	$5944m^2$ (-)	695,9M	13
ScanNet	caméra RGB-D	$34456m^2$ (-)	5581M	20
Semantic3D	LiDAR fixe	$40980m^2$ (-)	1660M	8
Oakland	MLS mono-fibre	$2694m^2$ ($1510m$)	1,61M	5
IQumulus	MLS mono-fibre	$2517m^2$ ($210m$)	12M	22
Paris-rue-Madame	**MLS multi-fibres**	**$2492m^2$ ($160m$)**	**20M**	**17**
Paris-Lille-3D	**MLS multi-fibres**	**$42714m^2$ ($1940m$)**	**143.1M**	**50**

Tableau 2.1 – Comparaison des jeux de données "Paris-rue-Madame" et "Paris-Lille-3D" avec d'autres jeux de données "Nuage de Points" du monde académique. Les deux premiers jeux de données sont des scènes d'intérieur, les autres des scènes d'extérieur. MLS = *Mobile Laser Scanning*

main qui n'a pas été publiée. Les chercheurs peuvent soumettre leurs résultats sur ce jeu de test et la métrique mIoU (pour *mean Intersection over Union* ou indice de Jaccard moyenné par classe) sera automatiquement calculée par le serveur en ligne.

Section	Surface couverte	Longueur	Nombre de points
Ajaccio2	$2917m^2$	120m	10M
Ajaccio57	$2281m^2$	100m	10M
Dijon9	$2531m^2$	100m	10M
Total	$7729m^2$	320m	30M

Figure 2.16 – Nuages de points tests du *benchmark* en ligne NPM3D.

On peut voir sur la Figure 2.17 le tableau en 2020 des méthodes publiées sur le

site web NPM3D.

Show 10 entries Search:

	Name	Av. IoU	Ground	Building	Pole	Bollard	Trash can	Barrier	Pedestrian	Car	Natural
1	LightConvPoint	82.7	99.6	98.1	77.2	91.1	64.7	66.5	58.1	95.6	93.9
	LightConvPoint: convolution for points -- https://arxiv.org/abs/2004.04462										
2	KP-FCNN	82.0	99.5	94.0	71.3	83.1	78.7	47.7	78.2	94.4	91.4
	H. Thomas, J.E. Deschaud, B. Marcotegui, F. Goulette, Y. Le Gall. Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods. In The IEEE International Conference on Computer Vision (ICCV), 2019.										
3	MS-RRFSegNet	79.2	98.6	98.0	79.7	74.3	75.1	57.9	55.9	82.0	91.4
4	RandLANet	78.5	99.5	97.0	71.0	86.7	50.5	65.5	49.1	95.3	91.7
5	ConvPoint	75.9	99.5	95.1	71.6	88.7	46.7	52.9	53.5	89.4	85.4
	Generalizing discrete convolutions for unstructured point clouds http://arxiv.org/abs/1904.02375										
6	SIConv	70.1	99.4	95.0	65.4	71.5	22.8	43.2	48.9	95.2	89.3
7	HDGCN	68.3	99.4	93.0	67.7	75.7	25.7	44.7	37.1	81.9	89.6
	Hierarchical Depthwise Graph Convolutional Neural Network for 3D Semantic Segmentation of Point Clouds. ICRA2019										
8	MS3_DVS	66.9	99.0	94.8	52.4	38.1	36.0	49.3	52.6	91.3	88.6
	Classification of Point Cloud for Road Scene Understanding with Multiscale Voxel Deep Network. 10th workshop on Planning, Perception and Navigation for Intelligent Vehicules PPNIV'2018										
9	EdConvE	66.7	99.3	92.1	37.7	60.7	78.9	35.6	28.2	83.2	84.5
10	RF_MSSF	56.3	99.3	88.6	47.8	67.3	2.3	27.1	20.6	74.8	78.8
	H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette, L. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In 2018 International Conference on 3D Vision (3DV), 2018 .										

Showing 1 to 10 of 11 entries Previous 1 2 Next

FIGURE 2.17 – Tableau en ligne sur le site web NPM3D du *benchmark* proposé pour le jeu de données Paris-Lille-3D : `https://npm3d.fr/paris-lille-3d`.

2.1.4 Drone LiDAR

Lors des travaux de thèse de Xavier Roynard, nous avons mis au point un drone équipé d'une plateforme permettant l'acquisition de données LiDAR et images (voir Figure 2.18). Il s'agit d'un drone DJI Matrice 600 (pouvant porter une nacelle de plusieurs kg) équipé d'une plateforme avec carte de calcul Nvdia Jetson TX1 (équipée d'un GPU), d'un LiDAR Velodyne VLP16 ou HDL32 (suivant les expérimentations) et d'une caméra FLIR.

L'objectif premier de ce drone est de produire des données de cartographie LiDAR par drone (qui seront de plus en plus utilisés à l'avenir dans certains métiers comme la surveillance de lignes haute tension). Cette plateforme nous permettra aussi de travailler à plus long terme sur les problématiques de drone autonome

(localisation, perception de l'environnement, cartographie temps réel...).

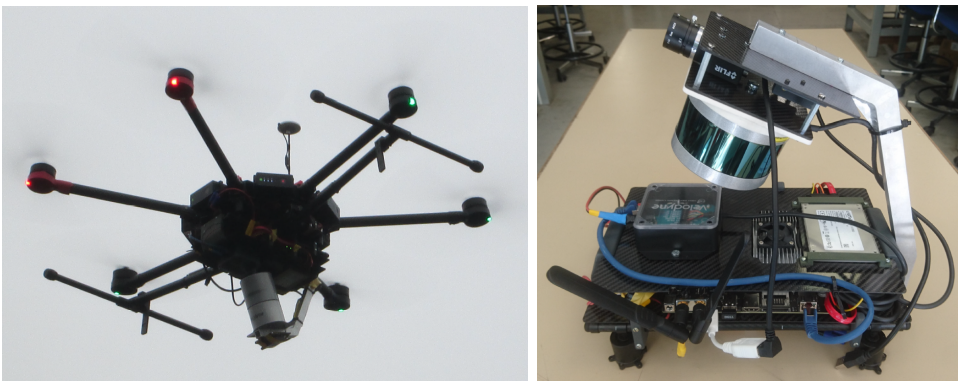

FIGURE 2.18 – A gauche, drone équipé d'un LiDAR Velodyne HDL32 en vol. A droite, plateforme du drone équipée d'un Velodyne VLP16 et d'une caméra FLIR.

La législation française n'autorisant pas le survol de zones habités, il a été difficile de mettre en place un scénario expérimental. La Figure 2.19 présente les résultats d'une acquisition du drone équipé du Velodyne HDL32 au-dessus d'une ferme familiale près d'Angers. Le nuage de points a été produit uniquement à partir des données LiDAR et d'une méthode SLAM LiDAR qui est décrite ci-dessous.

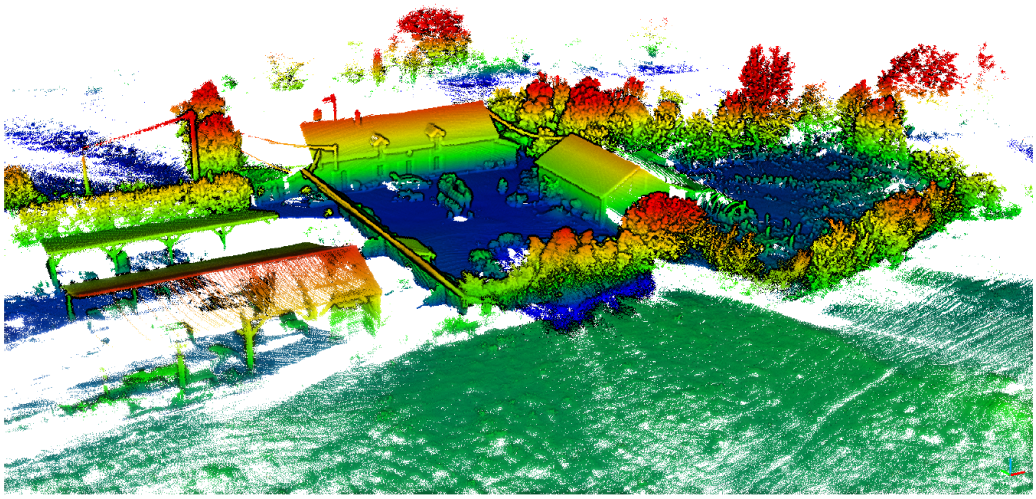

FIGURE 2.19 – Nuage de points d'une ferme produite à partir de la plateforme drone LiDAR.

2.1.5 SLAM LiDAR

Nous avons décrit jusqu'ici plusieurs systèmes mobiles de cartographie (véhicule et drone) que nous avons développés et améliorés lors de différents travaux de recherche. Nous avons particulièrement insisté sur les travaux de recherche pour améliorer la calibration des capteurs. Nous revenons ici à la partie localisation. Nous utilisons traditionnellement une méthode de couplage IMU-GPS (à travers un logiciel de post-traitement Novatel Inertial Explorer). Le problème qui découle du fait d'utiliser des capteurs spécifiques pour la localisation du système est que la donnée 3D produite comporte toujours des erreurs comme décrite dans [Barber *et al.* 2008] (problèmes de GPS en canyon urbain, calibration et dérive de l'IMU, calibration IMU<->LiDAR...).

Pour créer la donnée 3D, nous sommes donc partis sur une autre piste qui consiste à utiliser uniquement les données LiDAR. Un premier avantage est qu'il n'y a plus de problème de calibration (le capteur de localisation est le même que le capteur de perception). C'est en 2014 que la méthode LOAM [Zhang & Singh 2014] (pour *LiDAR Odometry And Mapping*) a pour la première fois montré des capacités intéressantes de localisation et cartographie avec 6 degrés de liberté et cela seulement à partir d'un capteur LiDAR multi-couches. C'est du SLAM LiDAR (SLAM pour *Simultaneous Localization And Mapping*). Pour cela, ils détectent des *features* sur la dernière *frame* LiDAR (*frame* = données correspondant à une rotation de 360 degrés du LiDAR). Ces *features* correspondent à des coins ou des zones planaires. Ils font ensuite le recalage de ces *features* avec une carte de tous les *features* précédemment vus. Dans la même idée, KinectFusion [Newcombe *et al.* 2011] publié en 2011 est un SLAM basé sur des capteurs RGB-D. Ils développent l'idée de faire du recalage *frame-to-model* au lieu de faire du *frame-to-frame* et cela en temps réel sur GPU. Le *model* est une carte de voxels représentant la surface sous forme de TSDF (*Truncated Signed Distance Function* introduit par [Curless & Levoy 1996]). Nous sommes partis de cette idée pour imaginer un SLAM LiDAR *frame-to-model* publié dans [Deschaud 2018] (cette activité de recherche a été explorée et étudiée par moi seul).

Notre méthode n'utilise ni GPS, ni IMU, ni caméra mais uniquement les données LiDAR, ce LiDAR devant être un LiDAR multi-couches. Tout d'abord, nous créons le nuage de points correspondant à une rotation du LiDAR (en tenant compte du déplacement du véhicule pendant une rotation complète du LiDAR). Ensuite, on sélectionne des points échantillons dans ce nuage de points qui vont servir au recalage avec la carte. La carte est aussi un nuage de points, il s'agit de l'ensemble des n dernières *frames* rassemblées dans un repère commun. Le recalage du nuage de points de la *frame* courante s'opère avec la carte nuage de points en utilisant une métrique proche de ICP *Point-to-plane* appelée IMLS (pour *Implicit Moving Least Squares*) d'où le nom de la méthode IMLS-SLAM.

ICP (*Iterative Closest Point*) de [Besl & McKay 1992] est une méthode classique de recalage nuage à nuage. Il existe de nombreuses variantes de cette méthode, certaines décrites dans [Rusinkiewicz & Levoy 2001]. La plus connue est la variante *Point-to-plane*. Pour recaler deux nuages de points P et Q, on essaye de trouver la transformation (c'est-à-dire la rotation R et la translation t) qui minimise l'erreur :

$$E = \sum_{i=1}^{k} (n_i \cdot (\boldsymbol{R} p_i + \boldsymbol{t} - q_i))^2$$

avec :

- k le nombre de points échantillons pris dans le nuage P
- p_i le point i du nuage P
- q_i le point le plus proche de p_i dans le nuage Q
- n_i la normale au point q_i dans le nuage Q
- $\boldsymbol{R}$ et $\boldsymbol{t}$ sont la transformation que l'on cherche entre le nuage P et le nuage Q

Il n'y a pas de solution analytique à ce problème de minimisation. On opère donc par méthode de Gauss-Newton qui consiste à linéariser la matrice de rotation R et on arrive à la résolution d'un système linéaire sous la forme $Cx = b$.

La méthode classique pour obtenir un ICP rapide est de faire un échantillonnage en sélectionnant des points au hasard (car le calcul de voisinage est trop lent pour se faire sur tout le nuage de points). Pour que le recalage soit plus précis entre nos deux nuages de points (entre la *frame* et la carte), l'idée est d'utiliser des points soigneusement choisis plutôt que des points au hasard. Inspiré par les travaux de [Gelfand *et al.* 2003], nous choisissons les échantillons qui permettent de minimiser le conditionnement de la matrice C. De plus, nous avons repris l'idée développée pendant les travaux de calibration sur l'importance des zones planaires dans les nuages de points. Or les zone planaires permettent d'améliorer le recalage. Pour cela, on calcule la planarité des voisinages de chaque point avec a_{2D} et on ajoute ce facteur dans la sélection des échantillons. La Figure 2.20 montre le résultat d'un échantillonnage sur un nuage de points d'une *frame*.

Nous avons testé ce SLAM LiDAR sur le *benchmark* publique d'odométrie de KITTI (jeu de données connu dans le domaine du véhicule autonome). Il s'agit d'un jeu de données d'évaluation d'odométrie LiDAR et caméra avec 22 séquences en tout dont 11 séquences avec le GPS comme vérité terrain. Le LiDAR est un Velodyne HDL64 posé à la verticale. Nous étions en 2018 2e meilleur SLAM LiDAR sur le serveur en ligne KITTI : `http://www.cvlibs.net/datasets/kitti/eval_odometry.php`. Le Tableau 2.2 montre les résultats sur les données d'entraînement comparé à la méthode état de l'art LOAM [Zhang & Singh 2014]. La Figure 2.21 montre le nuage de points produit avec IMLS-SLAM sur la séquence 0 du jeu de données KITTI.

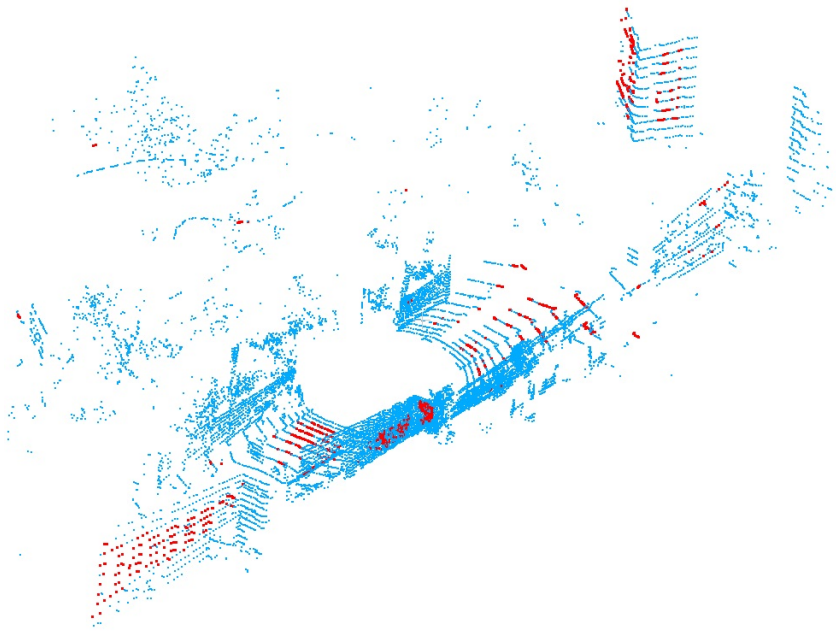

FIGURE 2.20 – Notre stratégie d'échantillonnage d'un nuage de points d'une *frame*. Les points en rouge sont les points sélectionnés pour le recalage. On peut voir que les points sur les zones planaires mais aussi ceux éloignés du capteur ont été choisis (permettant de mieux estimer les rotations).

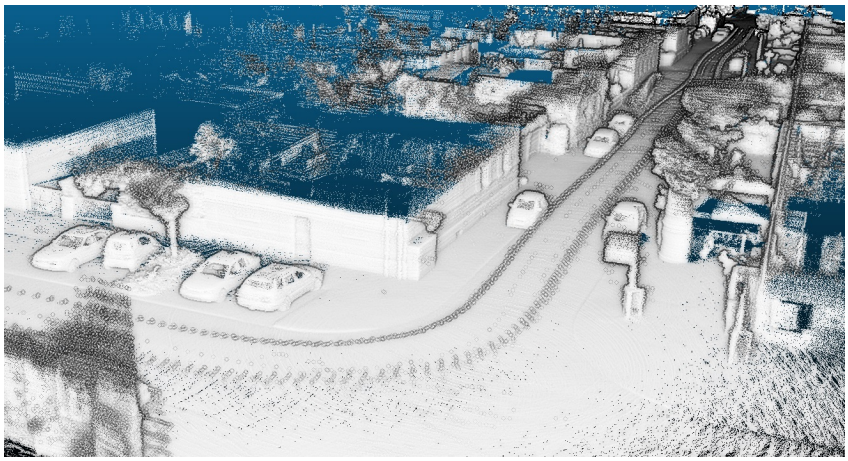

FIGURE 2.21 – Nuage de points généré à partir de la séquence 0 du jeu de données KITTI avec notre SLAM LiDAR.

La Figure 2.22 montre le test du SLAM sur des données du LiDAR Velodyne HDL32 posé à la verticale sur notre prototype L3D2. Nous avons fait deux tours du

Séquence	Environnement	LOAM [Zhang & Singh 2014]	Notre SLAM
0	Urbain	0.78%	**0.50%**
1	Autoroute	1.43%	**0.82%**
2	Urbain+Rural	0.92%	**0.53%**
3	Rural	0.86%	**0.68%**
4	Rural	0.71%	**0.33%**
5	Urbain	0.57%	**0.32%**
6	Urbain	0.65%	**0.33%**
7	Urbain	0.63%	**0.33%**
8	Urbain+Rural	1.12%	**0.80%**
9	Urbain+Rural	0.77%	**0.55%**
10	Urbain+Rural	0.79%	**0.53%**

Tableau 2.2 – Résultats de notre SLAM comparé à LOAM (résultats de LOAM sont pris de [Zhang & Singh 2014]) sur les 11 premières séquences du jeu de données KITTI. La métrique utilisée est la dérive en translation de KITTI.

jardin du Luxembourg dans Paris. On peut voir la qualité de la donnée produite avec la grille du jardin du Luxembourg visible. Il est impossible d'avoir ce niveau de détail en utilisant notre méthode classique de localisation basée sur la fusion des données IMU et GPS dans Paris.

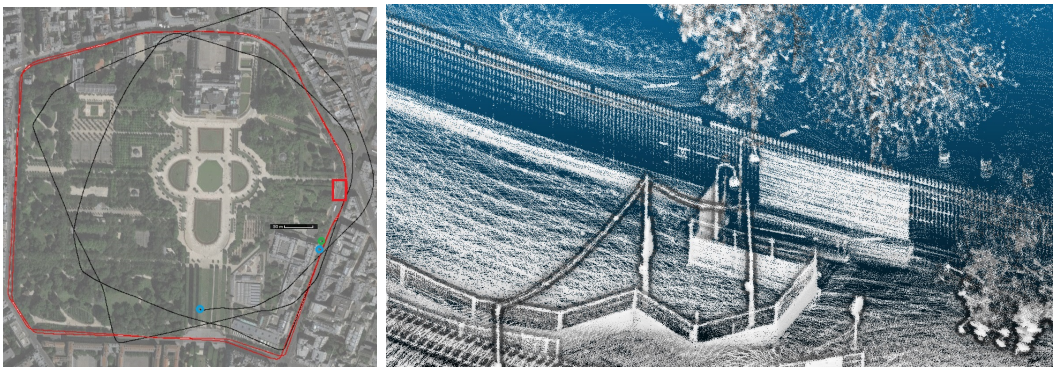

Figure 2.22 – A gauche, la trajectoire de IMLS-SLAM en rouge comparée avec une trajectoire en noire produite par un ICP classique *Point-to-plane*. A droite, un zoom sur une petite portion du nuage de points produit avec IMLS-SLAM. On peut voir la qualité de la donnée produite avec la grille du jardin du Luxembourg visible.

On peut voir sur l'image de la Figure 2.23 les résultats du SLAM avec un positionnement du LIDAR différent de celle des données KITTI et du test précédent autour du jardin du Luxembourg. Le LiDAR est à l'arrière du véhicule et incliné de

45 degrés (en position *mapping*). Cela devrait être une situation plus difficile pour l'algorithme car ainsi il y a moins de recouvrement entre deux *frames* successives. Pourtant, on peut voir la qualité du nuage de points produit par le SLAM sans aucun autre capteur que le LiDAR.

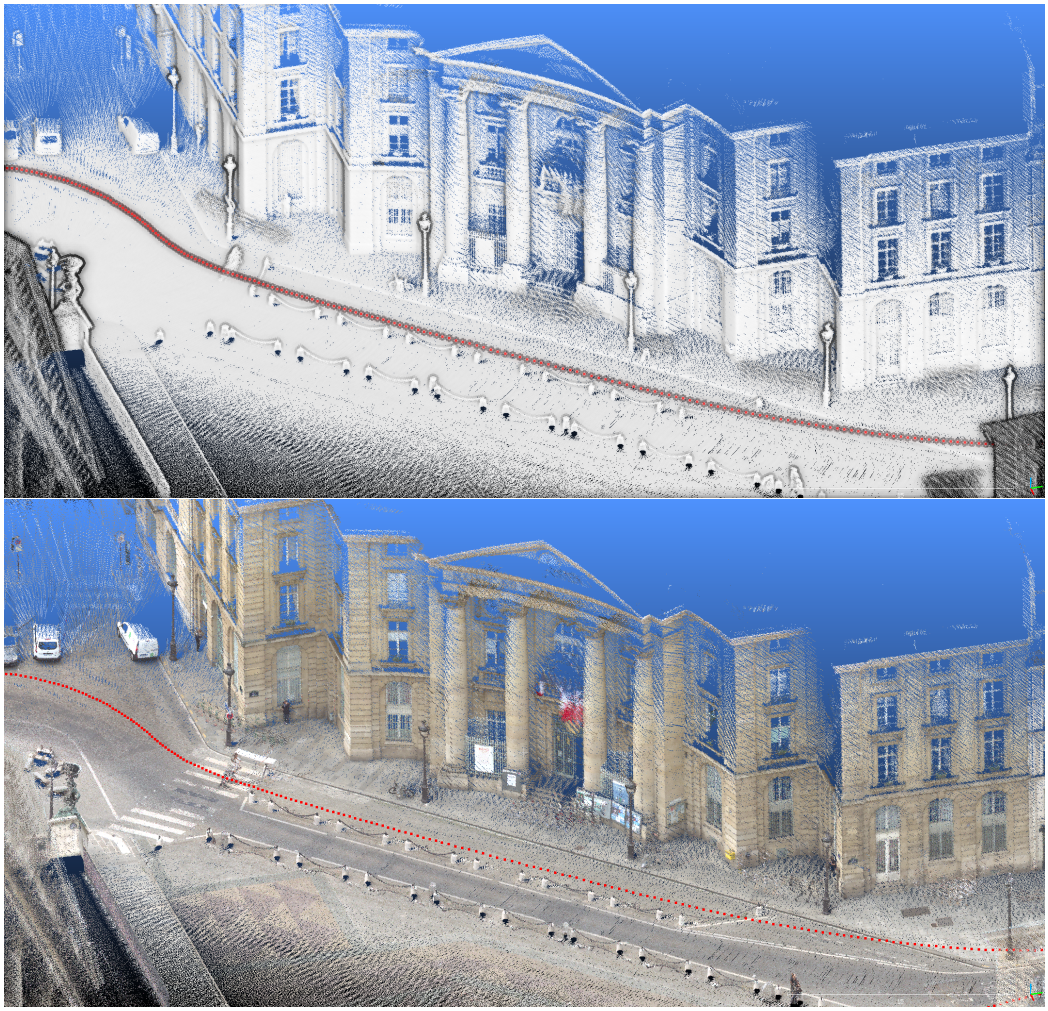

FIGURE 2.23 – Au-dessus, nuage de points produit par IMLS-SLAM avec le LiDAR positionné à l'arrière en mode *mapping*. Acquisition de la rue Soufflot et du tour du Panthéon avec vue sur la Mairie du 5ième (trajectoire du système mobile en rouge). En-dessous, même acquisition avec l'ajout de la couleur au nuage de points provenant d'une caméra Ladybug5 (6 caméras dans un seul bloc pour un champ de vue à 360°)

Nous utilisons maintenant la méthode SLAM LiDAR pour produire les données nuage de points avec le prototype L3D2. La qualité des nuages de points est bien supérieure à ce que nous obtenions avec notre post-traitement des données

GPS et IMU. Sachant que cette méthode n'utilise que le LiDAR, les données ne sont pas géoréférencées. Nous travaillons tout d'abord sur une méthode de fermeture de boucle avec détection de recouvrement dans les données LiDAR (en post-traitement) puis à intégrer les trajectoires produites par la fusion GPS et IMU, le tout par optimisation de graphe comme g2o dans [Kümmerle *et al.* 2011]. Cela nous permettra de garder la qualité des nuages de points grâce au SLAM LiDAR tout en produisant des données géoréférencées.

2.2 Analyse de nuages de points

Nous avons vu dans la section 2.1 les moyens expérimentaux permettant la création de grands volumes de données sous forme de nuages de points. Une partie importante de mes travaux de recherche consiste à améliorer les méthodes d'analyse de ces données, étape souvent nécessaire avant la partie modélisation (section 2.3). Par analyse, on entend ici toute méthode permettant de comprendre les données sans les transformer. Cela peut être de la détection de primitives, de la segmentation, de la classification d'objets, de la segmentation sémantique. Une contrainte forte dans les traitements que nous développons est la capacité de mise à l'échelle. Les données produites par les systèmes mobiles (véhicule, drone, caméras RGB-D...) sont toujours plus volumineux. C'est pourquoi nous gardons à l'esprit les temps de traitement et les complexités des algorithmes développés.

J'ai principalement focalisé mes travaux d'analyse sur la détection de primitives géométriques (et plus particulièrement les zones planaires dans les nuages de points) et la segmentation sémantique (la capacité à donner une information sémantique à chaque point ou groupe de points). Dans la mesure du possible, nous avons tenté de développer des méthodes suffisamment génériques pour s'adapter à une grande variété de nuages de points (scènes intérieurs et extérieurs, données LiDAR ou RDB-D...).

2.2.1 Détection de primitives géométriques

Pour traiter de grands volumes de données, il est nécessaire de la segmenter en plusieurs parties. Une méthode simple est de détecter et segmenter le nuage de points en primitives géométrique. Une primitive géométrique récurrente dans les nuages de points produits par les systèmes mobiles est le plan, très présent dans les scènes urbaines (sol, façades de bâtiments...) et les scènes d'intérieur (murs, tables...). La détection de ces plans permet par exemple ensuite de modéliser différemment les parties planaires et non planaires.

Les 3 grandes familles de segmentation en images ou 3D sont la croissance de région, la transformée de Hough [Hough 1962] et la méthode RANSAC (*RANdom SAmple Consensus*) de [Fischler & Bolles 1981]. Dans [Deschaud & Goulette 2010a], nous avons effectué une analyse comparative de RANSAC et de la croissance de région appliqués aux nuages de points pour faire de la détection de plans. Ces deux méthodes ont des avantages et des inconvénients. La première, RANSAC, est très rapide pour détecter des plans de grande taille mais va demander de nombreuses itérations pour trouver les petits plans alors que la croissance de région est rapide pour détecter les petits plans et demandent beaucoup de temps pour croître et segmenter les grandes zones planaires du nuage de points.

Nous avons voulu accélérer la méthode de croissance de région. Au lieu de prendre les k plus proches voisins à chaque itération, on découpe l'ensemble du nuage en voxels et la croissance se fait voxel par voxel, en ajoutant à chaque étape l'ensemble des points de chaque voxel ce qui rend la croissance beaucoup plus rapide. La méthode a été publiée dans [Deschaud & Goulette 2010a]. De plus, dans ces travaux, nous avons précisé trois critères d'appartenance d'un point à un plan pour affiner la segmentation : la distance de chaque point au plan, l'angle entre la normale d'un point et la normale du point d'origine de la croissance et l'angle entre les normales des points et la normale du plan. La Figure 2.24 montre un résultat de notre méthode de croissance par voxels sur un nuage de points d'une église acquise par un scanner fixe LEICA Cyrax. La méthode est très rapide avec un temps de traitement d'environ 1 ms par point.

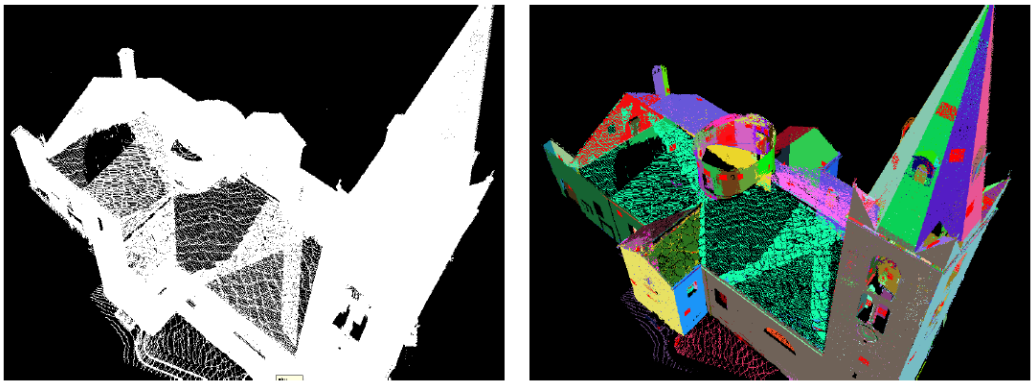

FIGURE 2.24 – A gauche, nuage de points d'une acquisition fixe par un scanner LEICA Cyrax. A droite, ensemble de plans détectés par notre méthode de croissance par voxels (en rouge, les points à l'origine des croissances par voxels).

2.2.2 Segmentation sémantique

Un traitement important pour les nuages de points est la segmentation sémantique. Il s'agit de donner une information sémantique de classe à chaque point ou à un groupe de points (de la même façon qu'en image où l'on cherche une information de classe par pixel). Les classes vont bien sûr dépendre du type de données en entrée et de l'application. J'ai encadré deux thèses sur ce sujet avec Xavier Roynard de 2015 à 2019 puis Hugues Thomas de 2016 à 2019. Nous verrons dans cette partie les différentes méthodes que nous avons développées au cours de ces années avec deux grandes familles : tout d'abord les méthodes non *Deep*, c'est à dire par apprentissage classique avec des descripteurs calculés à la main puis l'arrivée des méthodes *Deep* pour les nuages de points, c'est à dire par apprentissage profond.

Tous ces travaux ont été rendus possibles grâce à la création de nombreux jeux de données publiques dans le domaine du nuage de points et notamment Paris-rue-Madame [Serna *et al.* 2014] et Paris-Lille-3D [Roynard *et al.* 2018b] que nous avons présentés en section 2.1. Mais on peut citer aussi ScanNet (scans de pièces en intérieur), S3DIS (scans d'intérieur de bâtiments), Semantic3D (scans LiDARs fixes de scènes extérieures), ModelNet40 (modèles CAO transformés en nuages de points), ShapeNetPart (nuages de points d'objets segmentés en parties)... L'augmentation des jeux de données disponibles a réellement permis de faire de nombreux progrès et notamment pour les méthodes *Deep* qui demandent de gros volumes de données pour l'apprentissage.

2.2.2.1 Segmentation sémantique non *Deep*

Il y a principalement deux approches différentes pour traiter le problème de la segmentation sémantique. La première consiste à d'abord segmenter la scène en groupe de points et de classifier chaque groupe. Dans la deuxième approche, chaque point est considéré de façon individuel et on lui attribue une classe (les points pouvant ensuite être groupés en objets si nécessaire).

En 2014, une des méthodes les plus performantes était celle de [Serna *et al.* 2014]. Elle était basée sur de la segmentation avec une chaîne de traitements qui suivait celle de [Golovinskiy *et al.* 2009] de 2009. La méthode se décompose en 3 étapes : a) Extraction du sol, b) Segmentation des objets, c) Classification des objets par apprentissage supervisé. Les travaux de [Serna *et al.* 2014] appliquent ce schéma en utilisant des opérateurs morphologiques pour l'étape de segmentation et un SVM (*Support Vector Machine*) pour l'étape de classification.

En 2016, nous avons proposé une nouvelle méthode publiée dans [Roynard *et al.* 2016] qui suit aussi les étapes de [Golovinskiy *et al.* 2009]. Pour l'extraction de sol, nous avons appliqué une méthode de croissance de région en projetant le nuage de points sur une grille de voxels 2D suivant l'axe z. Pour la segmentation des objets, nous avons aussi effectué une croissance de région mais cette fois-ci dans une grille de voxels 3D. Enfin, pour la classification, nous avions voulu comparer les nombreux descripteurs 3D de la littérature. Nous avons ainsi choisi les plus classiques comme les descripteurs géométriques (inspiré de [Serna *et al.* 2014]), CVFH [Aldoma *et al.* 2011] pour *Clustered Viewpoint Feature Histograms* (histogrammes d'angles de normales), ESF [Wohlkinger & Vincze 2011] pour *Ensembles of Shape Functions* (histogrammes de distance entre des paires ou triplets de points) et GRSD [Marton *et al.* 2010] pour *Global Radius-based Shape Descriptors* (histogrammes des paramètres des sphères approximant localement la surface).

En tout, nous avions pour chaque objet 3D un descripteur total de longueur 990 avec 21 pour les caractéristiques géométriques, 308 pour CVFH, 640 pour ESF

et 21 pour GRSD.

Pour le descripteur géométrique, nous avons choisi :

— le nombre de points de l'objet à classifier
— $z_{max} - z_{min}$
— σ_z l'écart-type suivant z
— $x_{max} - x_{min}$ dans le repère local de l'objet
— $y_{max} - y_{min}$ dans le repère local de l'objet
— la plus petite valeur propre λ_0 de la matrice de covariance des points projetés dans la plan (x, y)
— la plus grande valeur propre λ_1 de la matrice de covariance des points projetés dans la plan (x, y)
— $\frac{\lambda_1}{\lambda_0}$
— l'aire de l'enveloppe convexe 2D des points projetés dans le plan (x, y)
— l'aire de l'enveloppe convexe 3D
— le volume de l'enveloppe convexe 3D
— un histogramme en 10 cases des points suivant z

Nous avions choisi un *Random Forest* comme algorithme de classification car il présente de nombreux avantages par rapport aux SVM.

Les expérimentations avaient été faites sur une partie du jeu de données Paris-rue-Madame [Serna *et al.* 2014] et une partie du jeu de données Paris-Lille-3D [Roynard *et al.* 2018b] (qui n'existait pas encore). La Figure 2.25 montre le résultat des étapes d'extraction du sol et de segmentation des objets sur une partie du nuage de Lille.

La Figure 2.26 montre un exemple de nuage de points 3D d'un objet après l'étape de segmentation (ici une voiture). On peut voir que la segmentation n'est pas idéale avec la disparition du bas de la voiture dans l'étape d'extraction du sol.

Le Tableau 2.3 montre les résultats de notre méthode comparés à ceux de [Serna *et al.* 2014] sur le jeu de données Paris-rue-Madame. On peut y voir une amélioration du F1 score (moyenne harmonique de la précision et du rappel).

Méthode	**Précision**	**Rappel**	**F1 score**
[Serna *et al.* 2014]	**100.0%**	94.6%	97.2%
Notre méthode	96.97%	**99.53%**	**98.23%**

Tableau 2.3 – Résultat de notre méthode de segmentation puis classification sur le jeu de données Paris-rue-Madame.

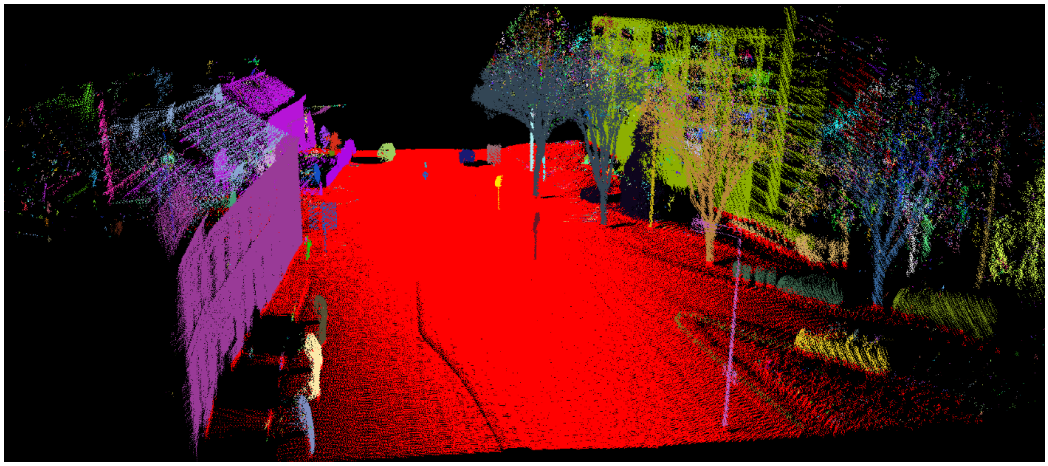

FIGURE 2.25 – Exemple d'extraction du sol (en rouge) et de segmentation des objets (autres couleurs) sur une partie du jeu de données Paris-Lille-3D.

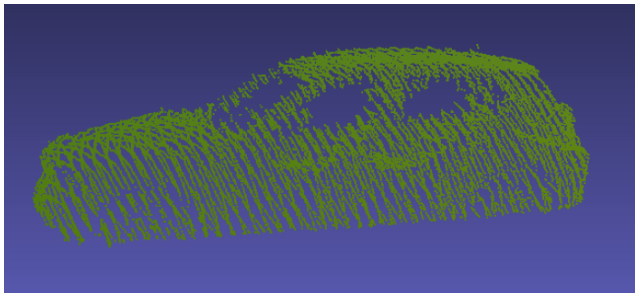

FIGURE 2.26 – Nuage de points d'un objet segmenté sur lequel nous calculons les descripteurs géométriques, CVFH, ESF et GRSD pour classifier l'objet en tant que voiture.

Le Tableau 2.4 montre une comparaison des résultats de classification en utilisant les différents descripteurs seuls ou une combinaison de tous les descripteurs. Ce qui est très intéressant, c'est de voir que les descripteurs géométrique seuls (avec seulement 21 variables) permettent d'obtenir un bon F1 score pour la classe "voiture" (classe qui nous intéressait particulièrement pour notre application d'études de stationnement dans les villes) de 97.58% alors qu'avec tous les descripteurs (990 variables), on arrive à un score à peine plus élevé de 98.23%. De plus, les descripteurs géométriques sont très rapides à calculer (avec un temps moyen de 0.9 ms par point) alors que le descripteur de taille 990 met 74.5 ms par point à être calculé (dû au besoin de calculer les normales pour CVFH et GRSD). Ces conclusions nous ont poussé pour la suite à des travaux utilisant uniquement les descripteurs géométriques plutôt que de se baser sur des descripteurs 3D complexes.

L'inconvénient de la stratégie en 3 étapes avec extraction du sol puis segmenta-

Descripteurs	Nombre de variables	OOB score	F1 score pour la classe "voiture"
Géométrique	21	98.30%	97.58%
CVFH	308	93.71%	91.42%
ESF	640	94.29%	91.76%
GRSD	21	94.38%	91.84%
Geom+CVFH+GRSD+ESF	990	**98.80%**	**98.23%**

Tableau 2.4 – Comparaison des résultats de classification en testant différentes combinaisons de descripteurs (OOB = *Out-of-bag*).

tion des objets et classification de chacun des objets, c'est l'étape de segmentation qui peut poser de nombreux problèmes (car elle n'utilise aucune information sémantique). En 2016, les travaux de [Hackel *et al.* 2016] ont montré qu'il est possible d'obtenir de très bons résultats en faisant de la classification par point en prenant des voisinages multi-échelles et en accélérant les temps de calcul par sous-échantillonnage du nuage à chaque échelle. Mais cette méthode utilise un voisinage k-nn (pour k *nearest neighbors*, les k plus proches voisins) qui n'a pas réellement de sens géométrique. Nous avons donc adapté et amélioré ces travaux dans [Thomas *et al.* 2018] en utilisant un voisinage sphérique et en choisissant avec attention les descripteurs géométriques.

La Figure 2.27 montre la différence entre l'utilisation d'un voisinage k-nn (prendre les k plus proches points) et d'un voisinage sphérique (prendre tous les points dans une sphère de rayon r). On peut remarquer que cela a une influence importante au niveau géométrique (particulièrement dans les zones où la densité varie fortement). Le problème du voisinage sphérique, c'est qu'il est possible d'avoir des voisinages avec trop peu de points (échelle trop petite ou densité trop faible) ou des voisinages avec trop de points (échelle trop grande ou densité des points trop forte). Une solution est d'effectuer un sous-échantillonnage du nuage à chaque échelle mais en utilisant une grille de voxels dont la taille va varier avec l'échelle. De cette façon, on évite de tomber dans les deux écueils précédents.

Le Tableau 2.5 montre les descripteurs utilisés dans notre méthode multi-échelles. On calcule pour cela à chaque échelle la matrice de covariance M des points de chaque voisinage $M = \sum_{p_j \in S(p_i)} (p_j - \bar{p_i}) \cdot (p_j - \bar{p_i})^T$ avec $S(p_i)$ le voisinage sphérique du point p_i et $\bar{p_i}$ le barycentre des points du voisinage de p_i. λ_i et e_i sont les valeurs et vecteurs propres calculés à partir de la matrice M avec $\lambda_1 \geq \lambda_2 \geq \lambda_3$. On calcule 8 échelles (on multiplie par 2 le rayon du voisinage sphérique à chaque échelle) ce qui donne un descripteur final de taille 144 par point. Pour l'apprentissage supervisé, nous avons de nouveau utilisé un *Random Forest*.

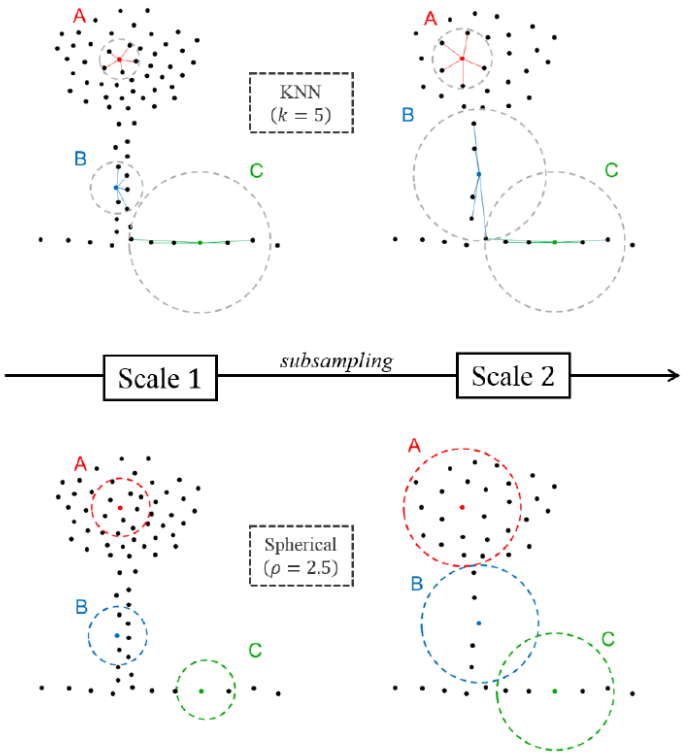

FIGURE 2.27 – Comportement des voisinages multi-échelles suivant l'utilisation d'un voisinage k-nn ou sphérique.

Le Tableau 2.6 montre les résultats de cette méthode multi-échelles, appelée RF_MSSF (pour *Random Forest with Multi-Scale Spherical Features*) sur le jeu de données Semantic3D [Hackel *et al.* 2017] (nuages de points acquis par LiDAR statique). En 2018, elle était devant toutes les méthodes de segmentation sémantique non *Deep* et devant les premières méthodes basées *Deep* (que nous verrons ci-dessous) sauf SPGraph [Landrieu & Simonovsky 2018]. La Figure 2.28 montre des images de nuages de points classifiés avec la méthode multi-échelles comparées à la vérité-terrain sur Semantic3D.

2.2.2.2 Segmentation sémantique *Deep*

De nombreuses tâches pour les nuages de points sont maintenant résolues à partir de réseaux convolutifs profonds comme l'estimation de normales [Boulch & Marlet 2016], l'estimation de flux de scènes [Liu *et al.* 2019], le recalage de nuages [Lu *et al.* 2019], l'extraction de primitives [Li *et al.* 2018], la reconstruction de surfaces [Mescheder *et al.* 2018], la détection d'objets en

Descripteurs	Définitions
Somme des valeurs propres	$\sum \lambda_i$
Omnivariance	$(\prod \lambda_i)^{\frac{1}{3}}$
Entropie	$-\sum \lambda_i \ln(\lambda_i)$
Linéarité	$(\lambda_1 - \lambda_2)/\lambda_1$
Planarité	$(\lambda_2 - \lambda_3)/\lambda_1$
Sphéricité	λ_3/λ_1
Courbature	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$
Verticalité (x2)	$\lvert\frac{\pi}{2} - angle(e_i, e_z)\rvert_{i\in(0,2)}$
Moments absolus (x6)	$\frac{1}{N}\lvert\sum((p-p_0)\cdot e_i)^k\rvert_{i\in(0,1,2)}^{k\in(1,2)}$
Moments verticaux (x2)	$\frac{1}{N}\lvert\sum((p-p_0)\cdot e_z)^k\rvert^{k\in(1,2)}$
Nombre de points	N

Tableau 2.5 – Descripteurs utilisés dans la méthode multi-échelles de segmentation sémantique par point (calculés à chaque échelle).

Méthode	mIoU
TMLC-MSR [Hackel *et al.* 2016]	54.2%
DePr3SS [Lawin *et al.* 2017]	58.5%
SnapNet [Boulch *et al.* 2017]	59.1%
SEGCloud [Tchapmi *et al.* 2017]	61.3%
SPGraph [Landrieu & Simonovsky 2018]	**73.2%**
RF_MSSF (notre méthode)	62.7%

Tableau 2.6 – mIoU (pour *mean Intersection over Union* ou indice de Jaccard moyenné par classe) sur le jeu de données Semantic3D (nuages de points acquis par LiDAR statique) avec la méthode multi-échelles RF_MSSF.

3D [Qi *et al.* 2019]...

Depuis 2015, il y a eu de nombreuses tentatives pour adapter les méthodes *Deep* à la problématique de segmentation sémantique de nuages de points. C'est l'année 2017 qui a été un tournant pour la 3D, marquée par le Challenge ShapeNet Core55 [Yi *et al.* 2017] impliquant 8 équipes de recherche pour de la récupération de modèles 3D à partir d'une base de données à grande échelle de modèles 3D. Cette année a vu apparaître de nouvelles méthodes *Deep* avec de nouvelles représentations des données 3D en entrée.

On peut séparer les méthodes *Deep* en 4 familles.

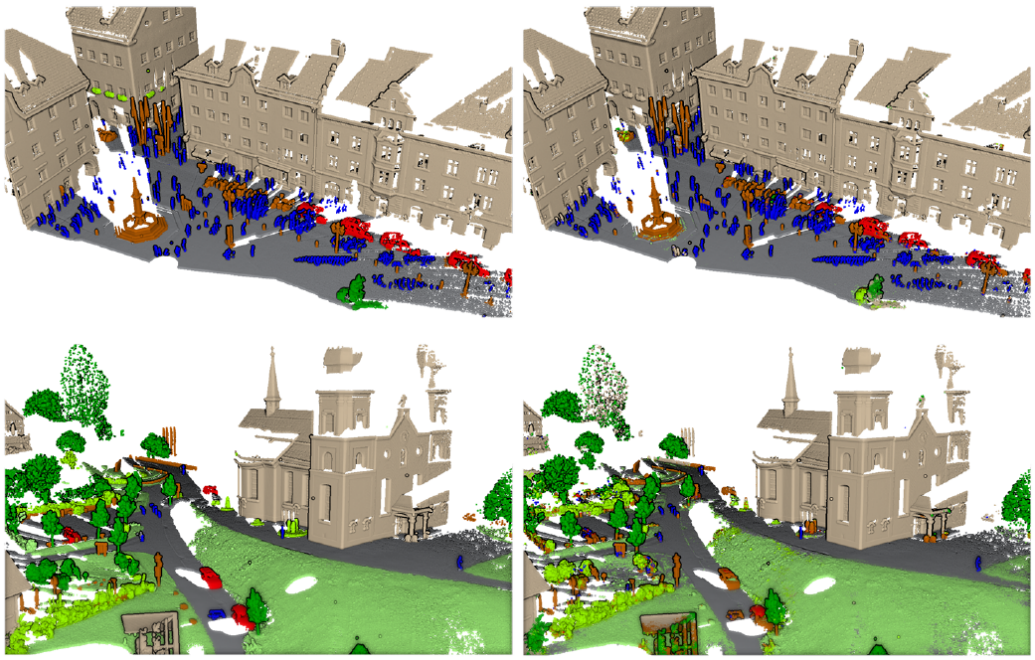

FIGURE 2.28 – Exemples de scènes classifiées avec la méthode multi-échelles sur Semantic3D avec à gauche la vérité-terrain et à droite la prédiction.

Projection sur des images L'approche la plus directe et la plus intuitive pour appliquer des architectures *Deep* à de la segmentation sémantique de nuages de points est d'utiliser des réseaux de neurones classiques sur des images obtenues à partir des nuages de points. Pour cela, on peut effectuer tout type de projection (perspective, parallèle, panoramique...) du nuage de points sur une caméra virtuelle (positionnée aléatoirement, en vue de dessus, placée verticalement...) en conservant par pixel des données de profondeur, d'accumulation, RGB (pour des nuages de points colorés)...

Cette méthode peut être améliorée en prenant plusieurs vues de la même scène ou du même objet et en fusionnant les résultats comme SnapNet dans [Boulch *et al.* 2017]. Il est à noter que ce type de méthode permet d'utiliser des réseaux pré-entraînés sur d'autres jeux de données (nombreux en image).

Convolution dans une grille de voxels Le premier réseau *Deep* construit pour classifier des nuages de points 3D est VoxNet de [Maturana & Scherer 2015]. La donnée nuage de points en entrée était transformée en une grille d'occupation (0 ou 1) et un réseau de neurones convolutifs 3D (CNN pour *Convolutional Neural Network*) permettait de prédire la classe de l'objet. En 2015, il n'était pas encore capable de rivaliser avec les méthodes non *Deep*. Depuis, de nombreuses

variantes sont apparues comme OctNet de [Riegler *et al.* 2017] qui utilise une structure hybride grille-octree ou comme VoxelNet de [Zhou & Tuzel 2018] qui augmente l'information contenue dans chaque voxel en utilisant un réseau de type PointNet [Qi *et al.* 2017b] (que nous verrons dans la partie Convolution sur nuages de points).

Convolution sur des graphes Il est possible de transformer un nuage de points en un graphe en utilisant les informations de voisinages (par exemple les k-nn voisins) comme arêtes du graphe. Une méthode appelée SPGraph [Landrieu & Simonovsky 2018] qui a eu de très bons résultats sur Semantic3D effectue d'abord une segmentation pour regrouper les points en cluster de points (appelés superpoints) et construit un graphe de superpoints (avec moins d'arêtes qu'en utilisant tous les points). Une convolution de graphe est alors appliquée pour classifier chaque superpoint. Les dernières méthodes basées graphe comme DGCNN de [Wang *et al.* 2019] définissent une convolution basée arêtes (EdgeConv) qui s'applique aux différentes couches du réseau avec un graphe calculé dynamiquement.

Convolution sur nuages de points Certaines méthodes sont capables de fonctionner directement sur les nuages de points et donc sur les données brutes. Une des premières méthode de ce type est PointNet [Qi *et al.* 2017b] qui a obtenu des très bons résultats sur le jeu de données ModelNet pour de la classification d'objets. PointNet est basé sur le fait qu'un nuage de points vérifie certaines propriétés de symétrie et utilise des opérateurs respectant ces symétries (comme les MLP pour *multilayer perceptron* et le *global pooling*). Mais cette architecture était limitée à de petits nuages de points de 1024 points. PointNet++ de [Qi *et al.* 2017a] a amélioré cela en extrayant des voisinages, appliquant PointNet à ces clusters et agrégeant l'information de façon hiérarchique comme dans les réseaux CNN.

Nous avons travaillé sur deux méthodes *Deep* de segmentation sémantique : la première basée sur une convolution dans une grille multi-échelles de voxels appelée MS3_DeepVoxScene et publiée dans [Roynard *et al.* 2018a] et la deuxième, une convolution sur nuages de points, appelée KPConv et publiée dans [Thomas *et al.* 2019]. Nous allons présenter ces deux méthodes.

MS3_DeepVoxScene La méthode proposée par le doctorant Xavier Roynard s'appelle MS3_DeepVoxScene pour *Multi-Scale Deep Voxel Scene*. Pour classifier un point p_i du nuage de points, on construit à partir de son voisinage trois grilles de voxels d'échelles différentes puis un réseau profond différent à chaque échelle va inférer pour chaque grille un descripteur de taille 1024. On utilise à chaque échelle

une grille cubique de taille 32^3 et une taille de discrétisation identique dans toutes les directions. On ne travaille qu'avec des grilles d'occupation binaires, chaque voxel contenant soit un 0 si le voxel est vide, soit un 1 s'il est occupé par au moins un point. Les 3 descripteurs inférés par les 3 réseaux 3D convolutifs sont concaténés et enfin un réseau MLP classique à deux couches va prédire la classe du point p_i. La Figure 2.29 présente un schéma de cette méthode d'apprentissage profond par grille de voxels multi-échelles. La figure 2.30 présente plus en détails l'architecture du réseau utilisé.

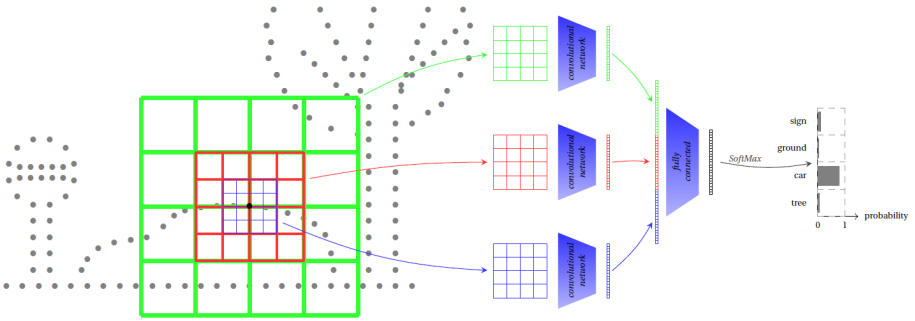

FIGURE 2.29 – Présentation du principe de la méthode MS3_DeepVoxScene d'apprentissage profond par grille de voxels multi-échelles (schématisé en 2D pour la simplicité).

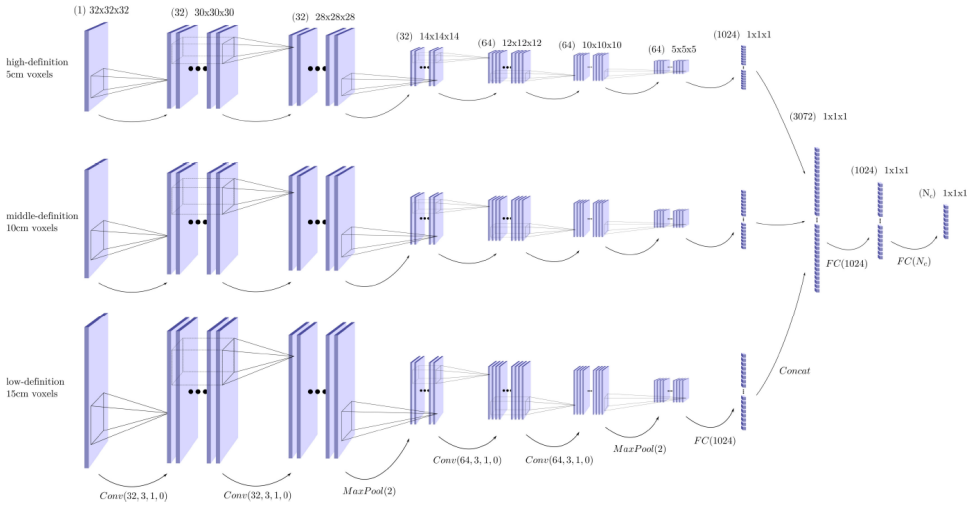

FIGURE 2.30 – L'architecture du réseau MS3_DeepVoxScene.

Le Tableau 2.7 montre les résultats quantitatifs sur le jeu de données Seman-

tic3D. L'intérêt d'utiliser un réseau profond par rapport à des descripteurs calculés à la main est confirmé ici avec des résultats meilleurs (65.3% mIoU) que notre méthode précédente RF_MSSF non *Deep* avec 62.7% mIoU (qui utilise pourtant beaucoup plus d'échelles, 8 exactement, au lieu de 3 échelles pour le réseau profond). De plus, on voit l'amélioration apportée par l'utilisation de plusieurs échelles au lieu d'une seule avec 8.2 IoU de plus entre le réseau profond avec une seule échelle et le réseau avec 3 échelles (tous les autres paramètres étant identiques).

Méthode	**mIoU**
TMLC-MSR [Hackel *et al.* 2016]	54.2%
DeePr3SS [Lawin *et al.* 2017]	58.5%
SnapNet [Boulch *et al.* 2017]	59.1%
SEGCloud [Tchapmi *et al.* 2017]	61.3%
RF_MSSF (notre méthode)	62.7%
SPGraph [Landrieu & Simonovsky 2018]	**73.2%**
MS1_DeepVoxScene (notre méthode)	57.1%
MS3_DeepVoxScene (notre méthode)	65.3%

Tableau 2.7 – Résultats (Février 2018) du réseau MS3_DeepVoxScene (avec 3 échelles de voisinage : voxels de 5 cm, 10 cm et 15 cm) et MS1_DeepVoxScene (avec une seule échelle : voxels de 10 cm) sur le jeu de données Semantic3D.

La Figure 2.31 montre les résultats qualitatifs sur le jeu de données Paris-Lille-3D.

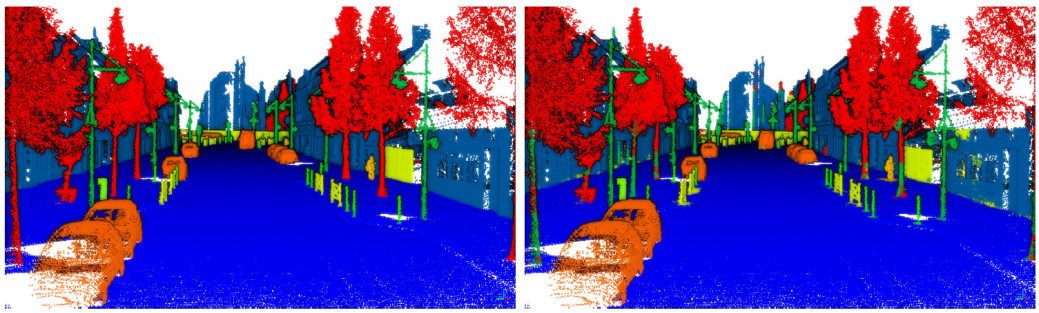

Figure 2.31 – A gauche, la vérité terrain de la segmentation sémantique du jeu de données Paris-Lille-3D. A droite, prédiction par le réseau MS3_DeepVoxScene (avec 3 échelles de voisinage : voxels de 5 cm, 10 cm et 15 cm).

Malgré une amélioration des résultats avec une IoU moyenne de 65.3% pour le jeu de données Semantic3D, l'inconvénient de cette méthode MS3_DeepVoxScene est de passer par une représentation intermédiaire sous forme de grille d'occupation

de voxels et d'être limitée à 3 échelles de voisinage. Nous allons voir la méthode KPConv capable de faire des convolutions directement sur le nuage de points.

KPConv KPConv est une méthode de convolution qui se base directement sur le nuage de points. L'idée est de faire porter les couches d'activation du réseau par le nuage de points et à chaque couche, d'effectuer un sous-échantillonnage pour diminuer le nombre d'activations mais augmenter la taille des descripteurs. La convolution est basée sur un nuage de point que l'on appelle noyau (*Kernel*) et qui va porter les poids (paramètres du réseau) sur les points (d'où KP pour *Kernel Point*). On opère donc une convolution entre un nuage de points (le noyau) et un autre nuage de points (le support). On utilise une fonction de corrélation pour affecter les poids du premier nuage au second. On définit donc la convolution comme :

$$g(p_i) = \sum_{\substack{j \\ ||p_j - p_i|| \leq r}} \sum_{k<K} h(p_j - p_i, \tilde{x_k}) W_k$$

avec $\{\tilde{x_k}, k < K\}$ les points du noyau et W_k leurs poids associés.
et avec la fonction de corrélation h :

$$h(y_j, \tilde{x_k}) = \max(0, 1 - \frac{||y_j - \tilde{x_k}||}{\sigma})$$

où σ est un paramètre qui va jouer sur la distance d'influence des points du noyau.

La Figure 2.32 montre une comparaison entre une convolution image et KPConv.

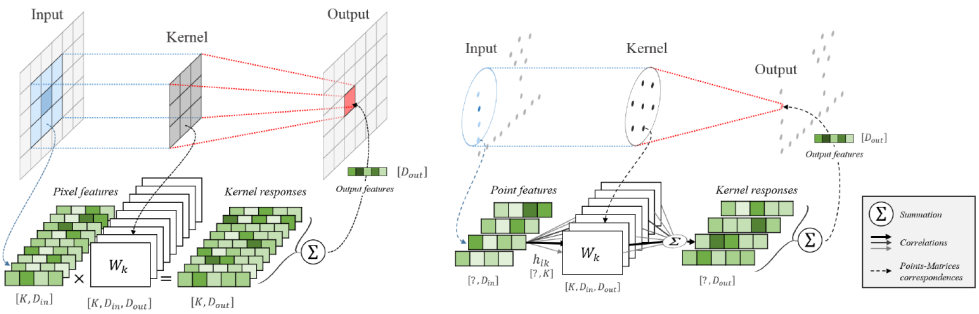

FIGURE 2.32 – Comparaison entre une convolution image (à gauche) et KPConv en 2D pour simplifier (à droite).

A chaque couche du réseau, un sous-échantillonnage (basé sur une grille de voxels comme dans la méthode multi-échelles RF_MSSF) permet de diminuer le nombre de points support (en doublant la taille des voxels à chaque couche). Pour cela, les points supports sont choisis comme les barycentres des voxels non vides.

On choisit le descripteur de ce barycentre par maximum des descripteurs de tous les points du voxel.

Nous avons testé cette architecture pour trois tâches : de la classification de nuage de points (jeu de données ModelNet40), de la segmentation (jeu de données ShapeNetPart) et de la segmentation sémantique (jeux de données ScanNet, S3DIS, Semantic3D et Paris-Lille-3D). Pour de la classification de nuages de points, nous avons utilisé un réseau de convolution à 5 couches KP-CNN (munie de la convolution 3D décrite plus haut) avec des blocs de type ResNet, de la *batch normalization* et *leaky ReLU* comme fonction d'activation. La dernière couche est une couche MLP avec un *softmax* comme en classification d'images. Pour de la segmentation ou segmentation sémantique, nous avons une architecture de type U-Net avec un encodeur identique à KP-CNN et pour le décodeur, un sur-échantillonnage par plus proche voisin et des *skip links* pour passer les descripteurs des couches intermédiaires de l'encodeur au décodeur. La Figure 2.33 présente les réseaux KP-CNN et KP-FCNN.

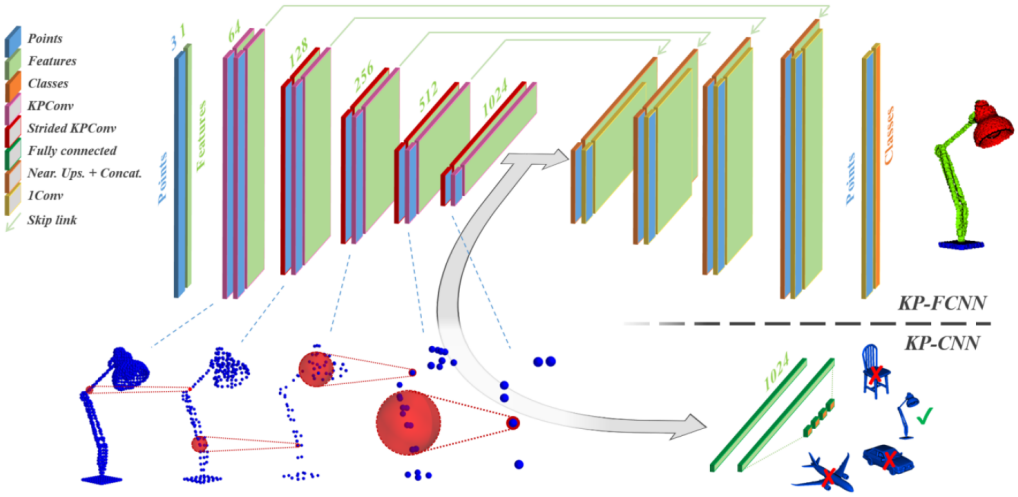

FIGURE 2.33 – Illustration des réseaux KP-FCNN (au-dessus, pour la segmentation sémantique) et KP-CNN (en dessous, pour la classification).

Le Tableau 2.8 présente les résultats de KP-CNN sur le jeu de données ModelNet40 et KP-FCNN sur le jeu de données ShapeNetPart. On peut voir que KPConv obtient de meilleurs résultats que toutes les autres méthodes *Deep*.

Le Tableau 2.9 présente les résultats de segmentation sémantique de KP-FCNN sur 4 jeux de données différents : ScanNet (scènes d'intérieurs avec des capteurs RGB-D), Semantic3D (scènes extérieures avec des scanners fixes), S3DIS (scènes d'intérieurs) et Paris-Lille-3D (scène d'extérieur à partir de systèmes mobiles

	ModelNet40	ShapeNetPart
Méthode	**OA**	**mIoU**
SPLATNet	-	83.7%
SGPN	-	82.8%
3DmFV-Net	91.6%	81.0%
SynSpecCNN	-	82.0%
RSNet	-	81.4%
SpecGCN	91.5%	-
PointNet++	90.7%	81.9%
SO-Net	90.9%	81.0%
PCNN by Ext	92.3%	81.8%
SpiderCNN	90.5%	82.4%
MCConv	90.9%	-
FlexConv	90.2%	84.7%
PointCNN	92.2%	84.6%
DGCNN	92.2%	85.0%
SubSparseCNN	-	83.3%
KPConv (notre méthode)	**92.9%**	**85.0%**

Tableau 2.8 – Résultats (Octobre 2019) de classification de nuages points et de segmentation en parties. Les métriques sont OA (*Overall Accuracy*) pour ModelNet40 et mIoU (IoU moyenné par classe) pour ShapeNetPart.

LiDAR). KP-FCNN ne peut pas prendre en entrée tous les points de ces nuages de points. Nous allons donc sélectionner une partie (une sphère de taille fixe de rayon $2m$ pour ScanNet et S3DIS et de rayon $3m$ pour Semantic3D et Paris-Lille-3D) et le réseau va prédire une classe pour chaque point inclus dans cette sphère. Cette étape est répétée de sorte à ce qu'au final chaque point du nuage soit vu plusieurs fois et la classe finale sera la classe prépondérante. Au final, KPConv obtient les meilleurs résultats sur l'ensemble de ces jeux de données (sauf ScanNet) parmi toutes les méthodes de l'état de l'art. KPConv est aussi une des premières méthodes à montrer des résultats sur autant de jeux de données différents. Cela montre l'intérêt de notre convolution sur nuage de points. Le Tableau 2.9 montre aussi le grand nombre de méthodes apparues ces dernières années pour faire de la segmentation sémantique sur nuages de points.

La Figure 2.34 montre des images de résultats sur deux nuages de points de scènes extérieures (Semantic3D) et intérieures (S3DIS). La figure 2.35 montre les résultats de la segmentation sémantique avec KP-FCNN sur le jeu de données Paris-Lille-3D. KPConv a depuis été amélioré et des résultats plus récents sont visibles sur le *benchmark* en ligne NPM3D où l'algorithme arrive à un score de 82.0% en mIoU. Du à l'inégalité de répartition des points dans les différentes

Méthode	ScanNet	Semantic3D	S3DIS	PL3D
TMLC-MSR	-	54.2%	-	-
DeePr3SS	-	58.5%	-	-
SnapNet	-	59.1%	-	-
SEGCloud	-	61.3%	48.9%	-
RF_MSSF (notre méthode)	-	62.7%	49.8%	56.3%
MS3_DVS (notre méthode)	-	65.3%	-	66.9%
PointNet	-	-	41.1%	-
PointNet++	33.9%	-	-	-
SPLATNet	39.3%	-	-	-
Eff3DConv	-	-	51.8%	-
TangentConv	43.8%	-	52.6%	-
MSDVN	-	65.3 %	54.7%	66.9%
RSNet	-	-	56.5%	–
FCPN	44.7%	-	-	-
PointCNN	45.8%	-	57.3%	-
PCNN	49.8%	-	-	-
SPGraph	-	73.2%	58.0%	-
ParamConv	-	-	58.3%	-
SubSparseCNN	**72.5%**	-	-	-
KPConv (notre méthode)	68.6%	**74.6%**	**65.4%**	**72.3%**

Tableau 2.9 – Résultats (Octobre 2019) de segmentation sémantique de nuages points. La métrique est mIoU (IoU moyenné par classe). Les scores pour ScanNet, Semantic3D et Paris-Lille-3D (PL3D) sont pris de leur *benchmark* en ligne. Le score pour S3DIS correspond au score sur le nuage Area-5.

classes, cela représente un OA (*Overall Accuracy*) de 95.3% des points qui sont correctement classifiés. Ce chiffre montre que la segmentation sémantique *Deep* de nuage de points arrive à obtenir de très bons résultats.

En conclusion, les méthodes de segmentation sémantique ont énormément évolué ces 5 dernières années, principalement avec l'arrivée de l'apprentissage profond sur les nuages de points qui est maintenant largement devant les méthodes classiques d'apprentissage supervisé avec des descripteurs.

La capacité à opérer une convolution directement sur les nuages de points a permis par la suite à résoudre d'autres problèmes que la segmentation sémantique. KPConv a été utilisé par d'autres chercheurs pour faire le recalage de nuages de points par calcul de descripteurs (D3Feat [Bai *et al.* 2020]) et l'amélioration de la rétroprojection en 3D de la sémantique prédite sur des images 2D (KPRNet [Kochanov *et al.* 2020]).

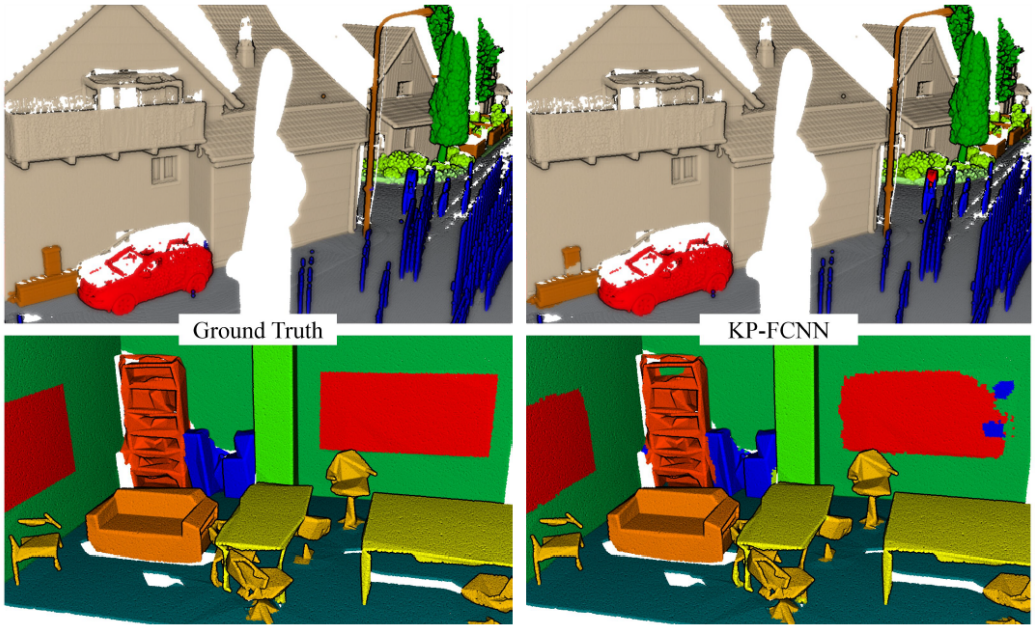

FIGURE 2.34 – Résultats visuels sur des scènes extérieures et intérieures. Au-dessus, le jeu de données Semantic3D. Au-dessous le jeu de données S3DIS. A gauche, la vérité terrain et à droite la prédiction par KP-FCNN.

FIGURE 2.35 – Segmentation sémantique du jeu de données Paris-Lille-3D avec KPConv. Au-dessus, la vérité terrain. Au milieu, la prédiction par KP-FCNN. En dessous, les erreurs de prédiction en rouge.

2.3 Rendu de nuages de points, reconstruction 3D et simulation

Nous avons vu dans les deux sections précédentes comment acquérir des nuages de points à grande échelle puis comment leurs donner une information sémantique. Dans cette partie, nous allons tout d'abord voir une méthode innovante pour visualiser un nuage de points afin d'obtenir un rendu permettant au mieux de représenter la géométrie.

Ensuite, nous verrons plusieurs applications où à chaque fois, nous transformons le nuage de points en un modèle. Ce modèle va dépendre bien sûr de l'application.

Pour les deux premières applications, nous faisons de la reconstruction de surface : nous représentons la géométrie sous forme de maillage triangulaire. Dans un cas, il s'agit de reconstruire l'environnement routier pour être capable de calculer des distances de visibilité pour la sécurité routière. Dans l'autre cas d'usage, nous construisons un modèle 3D précis de la route pour l'utiliser en environnement virtuel dans les simulateurs de conduite.

Pour la troisième application, nous transformons le nuage de points en un modèle hybride surfacique et volumique. L'objectif est de permettre de simuler de façon réaliste un capteur LiDAR à partir de données LiDAR.

Enfin, pour la dernière application, la géométrie et plus exactement la distance au nuage de points est représentée par une fonction implicite. Cette représentation permet d'adapter un maillage volumique pour la simulation d'écoulements de fluides autour de la géométrie du nuage de points.

2.3.1 Rendu de nuages de points

Avant de regarder comment passer d'un nuage de points à un modèle, nous allons voir une branche peu étudiée qui est le rendu direct de nuage de points. En effet, un nuage de points est une donnée volumineuse, éparse et chaque point n'ayant pas de de volume, il est difficile d'utiliser directement ces données pour tenter d'obtenir un rendu photo-réaliste de l'objet ou de la scène scannée. On passe ainsi généralement par une représentation intermédiaire sous forme de maillage nécessitant une étape de reconstruction de surface avec des méthodes connue comme Poisson de [Kazhdan & Hoppe 2013] ou RIMLS de [Öztireli *et al.* 2009]. Mais ces méthodes sont lentes et ne fonctionnent pas toujours sur des surfaces complexes ou lorsque les données sont très bruitées et incomplètes.

Une autre approche moins coûteuse que la reconstruction de surface est de faire du *splatting* [Botsch *et al.* 2005], c'est à dire remplacer chaque point par un *splat*

(un disque avec une information de normale) mais cela nécessite tout de même de lourds calculs en prétraitement (avant le rendu) simplement pour faire le calcul des normales. Cela devient même fastidieux lorsque les données sont très volumineuses ou impossible lorsque l'on souhaite visualiser les données *on the fly* (c'est à dire au moment de l'acquisition).

La méthode de rendu de nuages de points la plus basique et la plus utilisée consiste à projeter chaque point sur un pixel dans l'espace projectif de la caméra. Mais cette méthode (qui nécessite une information de couleur par point) produit aussi des artefacts comme on peut le voir sur la Figure 2.36.

FIGURE 2.36 – Principaux artefacts observables en rendu simple de nuages de points. (1) Problème de remplissage : proche du point de vue, la densité du nuage de points projeté sur l'image est trop faible. (2) Problème d'occlusion : on peut voir à travers le modèle. (3) Problème d'échantillonnage : *Aliasing*.

Pour remédier à cela, nous avons proposé une nouvelle méthode de rendu *screen-space* publiée dans [Bouchiba *et al.* 2018]. Il s'agit d'un rendu en espace image où le nuage points est d'abord projeté sur la caméra virtuelle puis les opérations de calculs d'occlusion et de remplissage sont approximées et se font dans l'image. Les travaux les plus proches sont ceux de [Pintus *et al.* 2011] qui font aussi du rendu en espace image. Toutefois, ils utilisent un voisinage fixe pour calculer les occlusions et le remplissage des pixels. Un petit voisinage limite la qualité du rendu. Mais en augmentant la taille de ce voisinage, le temps de calcul par image ne permet plus d'avoir du rendu temps réel. Pour améliorer cela, nous avons mis en place des opérateurs pyramidaux fonctionnant sur différentes résolutions de l'image permettant d'avoir des voisinages adaptatifs et une grande fréquence de rendu. La Figure 2.37 permet de comparer le calcul de normales sur une image de profondeur

avec à gauche un calcul sur l'image de profondeur de niveau 0 qui donne un résultat bruité et à droite le calcul des normales en utilisant plusieurs niveaux de résolution (pyramide d'images de profondeur) permettant d'obtenir des normales plus lisses et avec un très faible surcoût en temps de calcul.

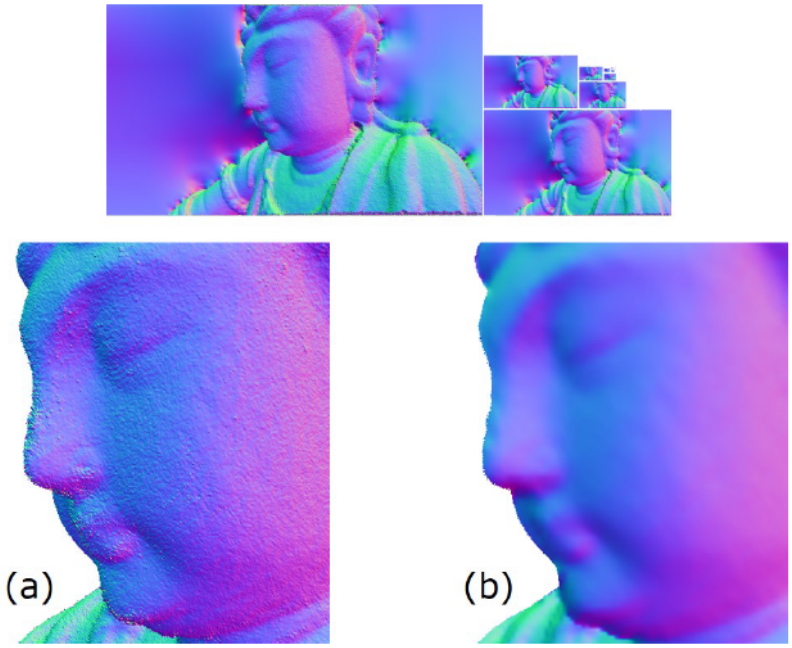

FIGURE 2.37 – Nuage de points Water-moon avec 6 millions de points obtenu par photogrammétrie. Au-dessus, pyramide d'images de normales calculées à partir d'une pyramide d'images de profondeur et l'algorithme *pull-push* de [Kraus 2009]. (a) Rendu utilisant les normales du niveau 0 donnant une surface bruitée. (b) Rendu utilisant les normales à différents niveaux de la pyramide d'images de normales.

La Figure 2.38 présente la chaîne de traitement complète pour le rendu temps réel d'un nuage de points. La première étape consiste à projeter le nuage de points sur la caméra virtuelle avec un point par pixel. Ensuite, nous construisons une pyramide d'images de profondeur (8 niveaux nous ont semblé suffisants). L'image de profondeur du 4ième niveau nous donne une information dense de profondeur nous permettant d'estimer le rayon de voisinage que nous allons utiliser ensuite (différent pour chaque pixel). La seconde étape consiste à enlever les points projetés non visibles depuis la caméra. Enfin, l'algorithme *pull-push* de [Kraus 2009] permet de construire une pyramide d'images de profondeur denses. Les normales sont calculées à partir des différents niveaux de la pyramide d'images de profondeur denses. La dernière partie est l'ombrage avec une technique classique de *deferred shading* permettant un rendu temps réel avec de nombreuses lumières.

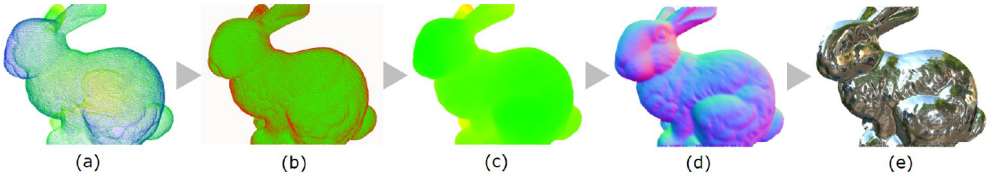

FIGURE 2.38 – Chaîne de traitement pour le rendu temps réel d'un nuage de points. (a) Projection du nuage sur la caméra virtuelle avec un point par pixel. (b) Rejet des points non visibles. (c) Remplissage par voisinage adaptatif. (d) Calcul des normales. (e) Ombrage

Les Figures 2.39 et 2.40 permettent de voir les bons résultats de rendus obtenus par notre méthode *screen-space* sur des nuages de points acquis par scanner fixe ou scanner mobile. Le Tableau 2.10 montre que l'étape de suppression des points non visibles (l'étape la plus coûteuse en temps) est en moyenne 10 fois plus rapide que celle de [Pintus *et al.* 2011]. Au final, nos temps de traitements sont constants pour différents jeux de données et restent équivalents à la passe géométrique (permettant des rendus à plus de 120 FPS, rendant possible l'exploitation de ces résultats dans les casques de réalité virtuelle). La vidéo suivante permet de visualiser les différentes étapes de la méthode de rendu : `https://www.youtube.com/watch?v=Z-N6Y6gW8Qo`.

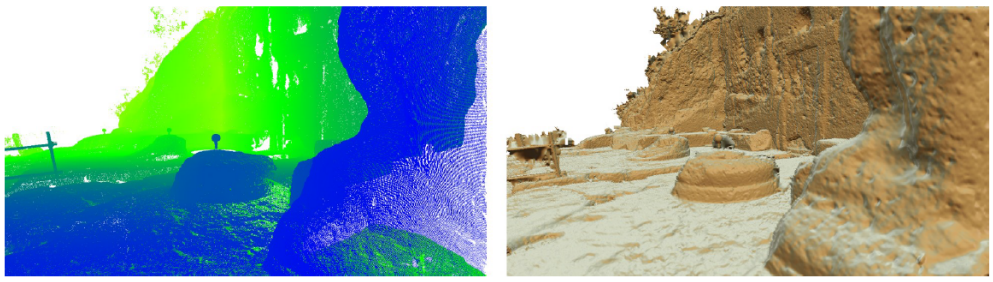

FIGURE 2.39 – A gauche, nuage de 73 millions de points de la façade d'une tombe étrusque acquis par scanner fixe FARO X130 (Tombe Scalina). A droite, rendu temps réel à 120 FPS grâce à notre méthode *screen-space* par opérateurs pyramidaux.

2.3.2 Reconstruction 3D

Le rendu en espace-image est un outil intéressant pour visualiser de gros volumes de données sous formes de nuages de points. Mais cela ne donne qu'une géométrie approximée de la scène. Pour les deux applications présentées ci-dessous, ayant besoin d'une géométrie plus exacte et plus complète, nous allons transformer le nuage de points en un modèle de type maillage surfacique.

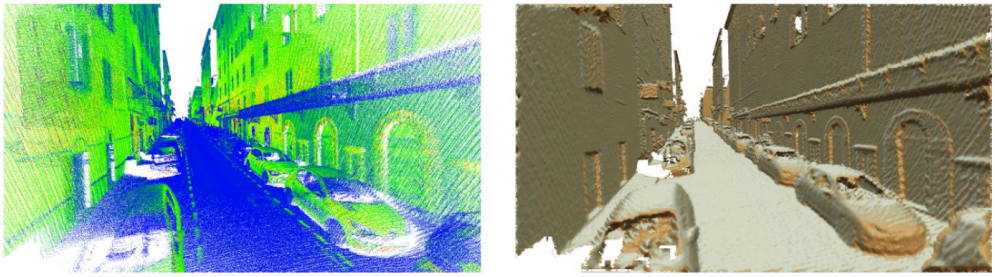

FIGURE 2.40 – A gauche, nuage de points d'une rue de la ville d'Ajaccio acquis par notre système mobile LiDAR L3D2 (2.7 milliard de points). A droite, rendu temps réel avec notre méthode *screen-space*.

Nuage de points	Passe géométrique	HPR [Pintus *et al.* 2011]	Notre HPR
Scalina	1.4 ms	58 ms	**2.7 ms**
Water-moon	2.0 ms	21 ms	**2.1 ms**
Ajaccio	11.4 ms	21.6 ms	**2.3 ms**

TABLEAU 2.10 – Comparaison du temps de calcul moyen pour la suppression des points non visibles (HPR pour *Hidden Point Removal*) entre la méthode de [Pintus *et al.* 2011] et notre méthode de rendu. Comme référence, la première colonne donne le temps de calcul de la passe géométrique, c'est à dire la projection des points 3D.

2.3.2.1 Modélisation 3D de l'environnement pour le calcul de visibilité des routes

La distance de visibilité sur route est une donnée importante pour la sécurité routière et joue un rôle dans le choix des vitesses limites des véhicules. Or il existe peu de méthodes permettant de connaître précisément cette distance le long d'une route. La difficulté venant aussi du fait qu'elle peut varier en fonction des saisons et des travaux sur la route ou aux abords de la route. Une méthode expérimentale appelée VISULINE est de faire rouler deux véhicules (équipés de GPS) roulant à une distance fixe l'un de l'autre. Le long du parcours étudié, le deuxième véhicule note à une certaine fréquence (par exemple toutes les 5s) si le premier véhicule est visible ou non. Pour connaître la distance de visibilité maximale, ces deux véhicules vont devoir rouler plusieurs fois sur le même parcours (avec des distances inter-véhicules croissants). Cette méthodologie reste fastidieuse.

Dans le cadre d'un projet de recherche (DIVAS), nous avons travaillé sur une méthode innovante : le principe est d'utiliser les données 3D de l'environnement acquis par les plateformes mobiles basées LiDAR comme LARA3D. Tout d'abord,

à partir du nuage de points produit par le système, nous enlevons les objets mobiles présents sur la route et appliquons un algorithme de reconstruction de surface. Nous avons utilisé pour cela une variante de la méthode BPA (*Ball Pivoting Algorithm*) de [Bernardini *et al.* 1999] qui est une méthode très rapide. A partir du maillage de l'environnement, nous pouvons effectuer un calcul de visibilité dans le modèle 3D en simulant un véhicule (modélisé par un parallélépipède de 1.5x4x1.3m). Le modèle 3D est projeté sur la caméra virtuelle (située sur la trajectoire d'origine à une distance fixe en arrière). Le véhicule est considéré comme visible si plus de 5% de sa surface est visible (par rastérisation). Ce même calcul se fait le long de la trajectoire (dans notre cas tous les 1m) avec différentes distances entre la caméra et le véhicule 3D.

Un logiciel spécifique (Figure 2.41) appelé "Qt-Ballad" a été développé par l'IFSTTAR pour effectuer les calculs, visualiser le modèle 3D et se déplacer le long de la trajectoire d'acquisition. La distance de visibilité disponible ainsi évaluée peut être comparée avec la distance de visibilité requise pour la sécurité routière, calculée à partir des caractéristiques de la trajectoire (courbure).

Nous avons mené une expérimentation sur une route de 1.6km en France (la RD786) présentant des caractéristiques potentiellement intéressantes en termes de visibilité géométrique et d'accidentologie (publiée dans [Tarel *et al.* 2012]). Nous avons obtenu une estimation de la distance de visibilité avec la méthode VISULINE avec 9 passages de deux véhicules (50m, 65m, 85m, 105m, 130m, 160m, 200m, 250m, 280m) et l'avons comparé avec notre méthode basée sur la modélisation 3D avec un seul passage de notre système d'acquisition mobile basé LiDAR. La Figure 2.42 montre les résultats pour deux vitesses de passage du véhicule d'acquisition à 45 km/h et 70km/h. On peut voir que la modélisation 3D permet d'obtenir des résultats très proches de la méthodologie VISULINE mais le gain de temps est notable puisqu'il faut 9 passages dans la méthode des véhicules suiveurs et 1 seul passage avec la méthode de modélisation 3D.

2.3.2.2 Reconstruction précise de la route pour les simulateurs de conduite

Les simulateurs de conduite (notamment pour le véhicule autonome) deviennent de plus en plus utilisés dans toutes les étapes de construction d'un véhicule (design, tests d'ADAS, tests d'ergonomie...). Ces simulateurs demandent ainsi d'avoir des environnements de plus en plus précis et réalistes. Dans le cadre du projet SIMVIR, nous avons travaillé sur la reconstruction précise des routes (à partir de données LiDAR mobiles) pour les utiliser dans les simulateurs de conduite.

Les travaux ont été mené par Daniela Craciun en Post-Doc au Centre de Robotique et ont amené aux publications dans [Craciun *et al.* 2014a],

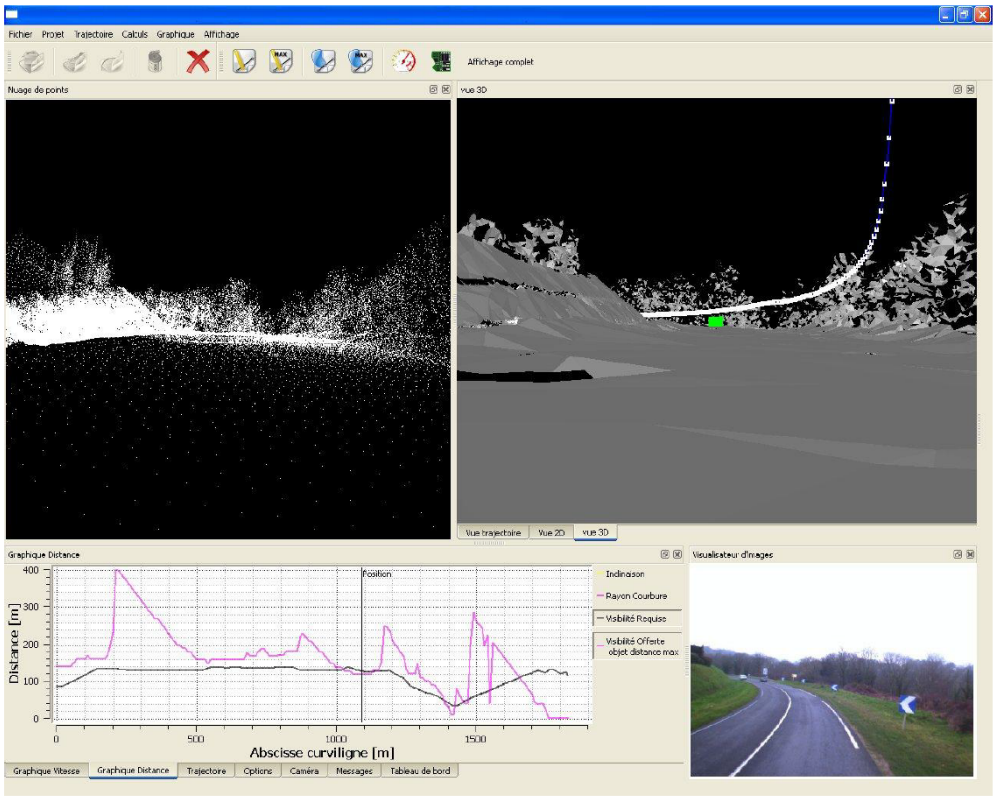

FIGURE 2.41 – Interface du logiciel de calcul de distance de visibilité sur routes développé par l'IFSTTAR. En haut à gauche, le nuage de points d'origine acquis par la plateforme mobile basée LiDAR. En haut à droite, le modèle 3D construit automatiquement à partir du nuage de points (maillage triangulaire). La trajectoire de la plateforme d'acquisition est formée des points blancs. Le bloc vert représente un véhicule simulé pour tester la visibilité sur route. En bas à droite, une image de la route étudiée. En bas à gauche, en magenta la distance de visibilité calculée par le modèle 3D et en noir la visibilité requise pour la sécurité routière. La droite verticale montre une zone de la route où la distance de visibilité réelle n'est pas suffisante.

[Craciun *et al.* 2014b] et [Craciun *et al.* 2017]. Nous avons proposé une méthode qui permet de reconstruite de façon entièrement automatique un maillage de la route en minimisant la taille du maillage créé, en conservant le maximum de détails sur la surface de la route et en préservant les bords saillants (comme les bords de trottoir).

La Figure 2.43 présente le schéma global de la méthode de reconstruction. Tout d'abord, la donnée 3D est découpée en sections linéaires (avec N millions de points), chaque section peut être traitée de façon indépendante pour créer un maillage puis au final, une procédure de fermeture va permettre de recoller les maillages de deux

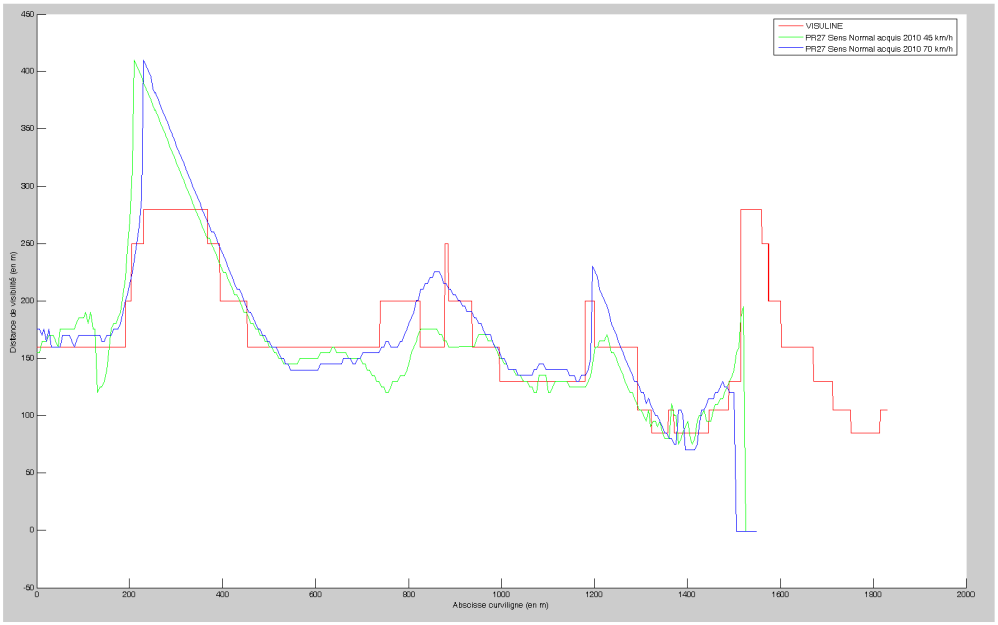

FIGURE 2.42 – Comparaison sur la même section de route de 1.6km de la méthode expérimentale VISULINE (2 véhicules suiveurs avec différentes distances inter-véhicules) avec notre méthode de calcul de distance de visibilité par modélisation 3D pour deux vitesses du système d'acquisition mobile LiDAR.

sections consécutives.

Dans chaque section, on effectue une segmentation du sol avec une méthode proche de [Serna *et al.* 2014] (la segmentation sémantique va aussi permettre d'enlever les objets mobiles sur la route). Les points 3D sont ensuite projetés sur un plan par moindre carré et une triangulation de Delaunay 2D est opérée [Shewchuk 1996]. La connectivité calculée est conservée sur les points 3D permettant d'obtenir un maillage 3D. La surface est régularisée en utilisant la procédure *Sinc Windowed smoothing* de [Taubin *et al.* 1996] (voir Figure 2.44) puis une décimation est appliquée en utilisant une implémentation dans VTK de [Schroeder *et al.* 1992] permettant de réduire le nombre de triangles de 90% (voir Figure 2.45).

La Figure 2.46 montre le résultat final de la reconstruction d'une route dans Paris, la rue Cassette. On peut voir la qualité visuelle de la reconstruction avec en encadré le ralentisseur de type dos d'âne mais aussi les bords saillants des trottoirs (en comparant avec une image prise de Google Street View). Pour le temps de calcul, cela demande au total environ 20s sur CPU pour une section de 50m du jeu de données rue Cassette comme celui de la Figure 2.46 (50m correspond à environ 1 million de points). Enfin, pour valider la précision du maillage final par rapport aux caractéristiques de la route, nous avons effectué sur le terrain des mesures (largeur

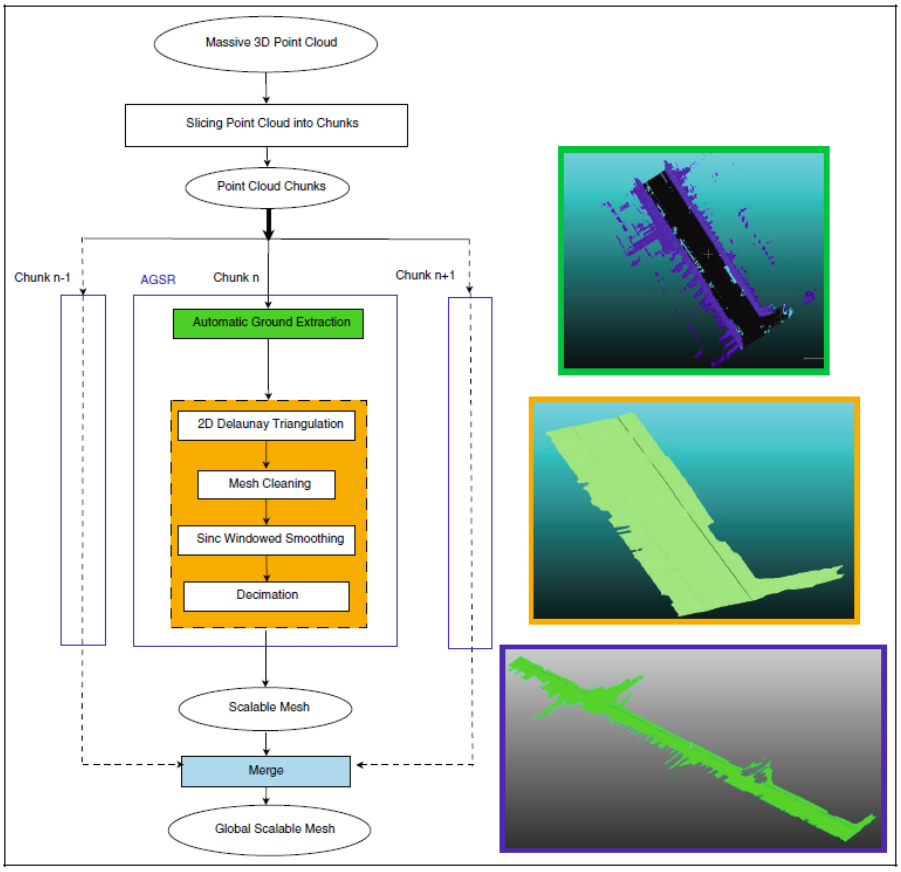

FIGURE 2.43 – Architecture globale de la méthode de reconstruction de routes avec encadré en couleur les résultats correspondants à 3 étapes importantes du processus : en vert, la segmentation du sol ; en orange, le maillage débruité et décimé ; en bleu, les maillages assemblés de deux sections consécutives.

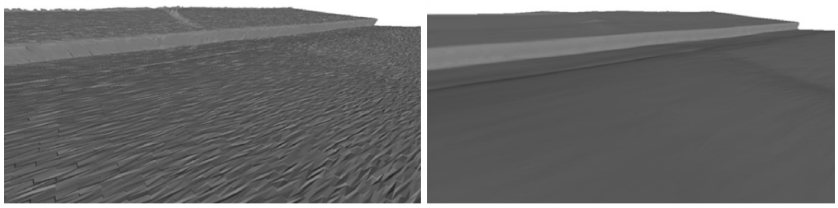

FIGURE 2.44 – Résultat de la procédure *Sinc Windowed smoothing* sur le jeu de données rue Cassette. A gauche, on peut voir le bruit sur le maillage provenant des données mobiles LiDAR. A droite, le maillage débruité.

de la route, hauteur du trottoir) en certains points précis et les avons comparés avec les mêmes mesures sur le maillage. Nous avons trouvé une erreur sur les distances

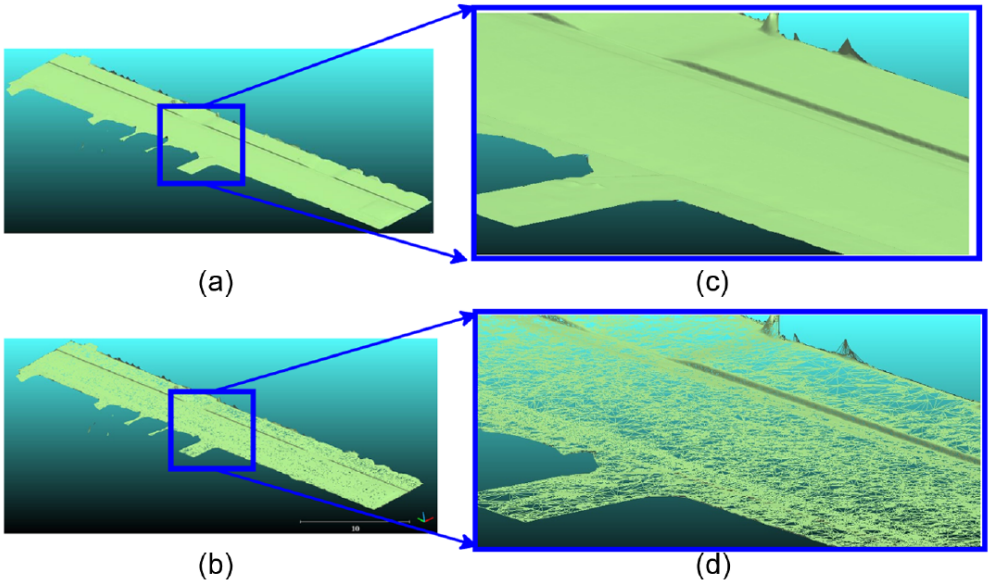

FIGURE 2.45 – Résultats de la décimation. (a) Maillage débruité avec 2.54 millions de triangles. (c) Zoom sur la zone bleue du maillage débruité. (b) Maillage décimé d'un facteur de 90% avec 254 000 triangles. (d) Zoom sur la zone bleue du maillage débruité et décimé.

mesurées inférieure à 1.5 cm (inférieur à l'écart-type du bruit du LiDAR utilisé). Cela montre que l'acquisition mais aussi tout notre traitement permet de construire un modèle très précis de la route (avec seulement 200k triangles pour 50m de route).

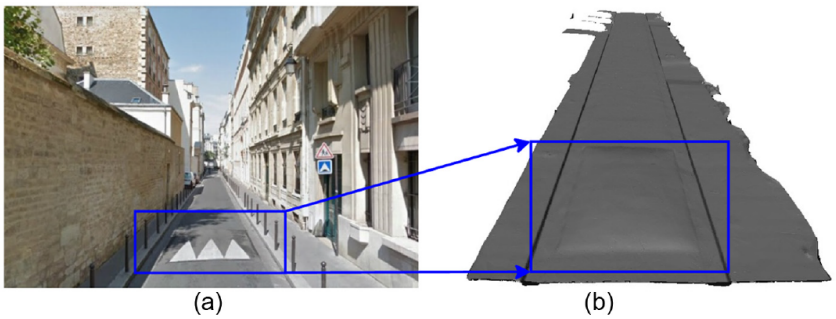

FIGURE 2.46 – Résultat final de la reconstruction de la route sur le jeu de données "rue Cassette" à Paris. (a) Image prise de Google Street View de la rue. (b) Maillage avec 203 000 triangles construit à partir d'un million de points pour une longueur de 51m.

La Figure 2.47 montre la comparaison de notre méthode de reconstruction avec deux autres méthodes classiques de reconstruction de surfaces à partir de nuages de

points 3D. La première est la reconstruction de Poisson [Kazhdan & Hoppe 2013] qui dans notre cas va lisser la surface et perdre les détails comme les bords saillants des trottoirs. La deuxième méthode est une méthode par propagation [Marton *et al.* 2009] implémentée dans PCL. Elle conserve mieux les détails mais la surface finale est très bruitée. Même en appliquant notre *Sinc Windowed smoothing*, cela ne permet pas de retrouver une surface lisse. De plus, notre méthode est 5 fois plus rapide que Poisson et 16 fois plus rapide que la méthode de propagation. Ainsi, notre méthode de reconstruction de la route est suffisamment rapide pour être appliquée sur de grands volumes de données mobiles : sur 100 km de linéaire acquis par système mobile LiDAR, la reconstruction prendrait 10h de calcul à partir d'un nuage de 6 milliards de points.

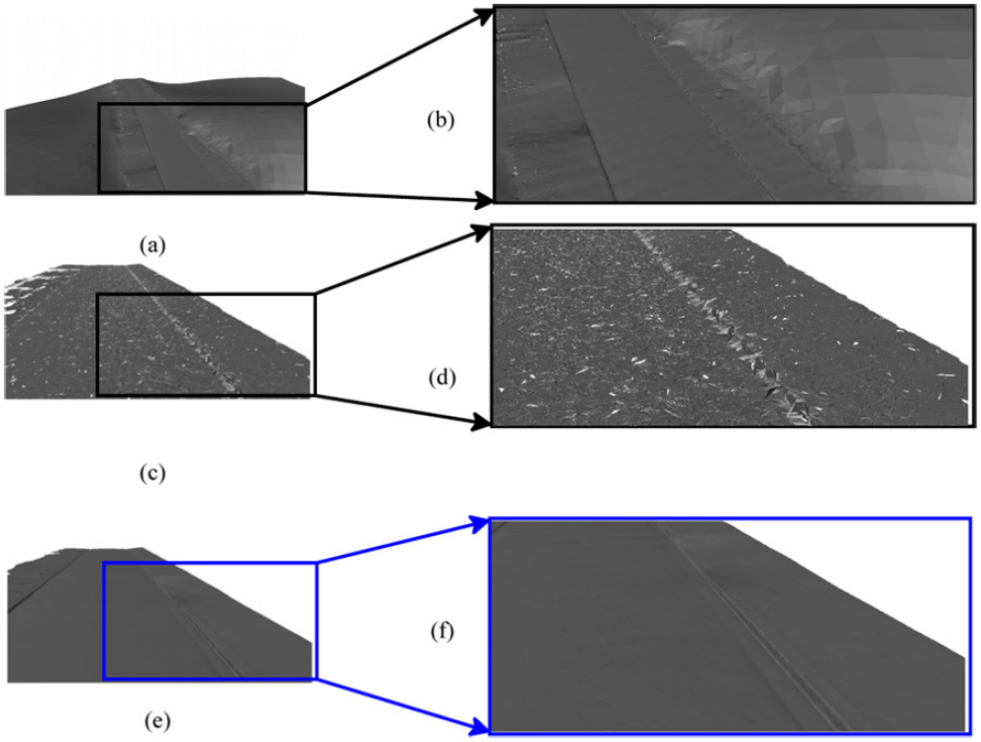

FIGURE 2.47 – Comparaison de différentes méthodes de reconstruction sur la même zone. (a)+(b) Méthode Poisson [Kazhdan & Hoppe 2013]. (c)+(d) Méthode de propagation [Marton *et al.* 2009] (e)+(f) Notre méthode de reconstruction.

2.3.3 Simulation

Nous allons voir maintenant deux autres applications : la simulation de capteurs LiDARs à partir de données LiDARs et la simulation d'écoulements de fluides à partir de données nuages de points d'objets ou de scènes réelles. Dans les deux cas,

nous allons transformer le nuage en un modèle spécifique à chaque application.

2.3.3.1 Simulation d'un LiDAR à partir de données LiDAR

La capacité à simuler de façon réaliste les LIDARs est importante pour tester des solutions nouvelles de véhicules autonomes équipés de ces capteurs. La plupart des simulateurs de LiDARS utilisent des environnements virtuels avec des modèles surfaciques de type maillage et génèrent des distances par lancer de rayon dans le modèle en ajoutant un bruit gaussien. Cela fonctionne bien pour des environnements de type planaire mais représente très mal le comportement des LiDARS dans la végétation ou d'autre types de surfaces plus particulières (comme un grillage par exemple).

Dans nos travaux publiés dans [Deschaud *et al.* 2012] et [Browning *et al.* 2012] lors de mon Post-Doc à Carnegie Mellon University, nous avons modélisé des données acquises par LiDAR en un modèle volumique et stochastique permettant de représenter mieux la nature spécifique de l'interaction du LiDAR avec telle ou telle zone. Il s'agit d'une modélisation basée sur de la donnée 3D : un nuage de points acquis par un système mobile LiDAR d'un environnement que l'on cherche à simuler. Nous construisons au final un modèle hybride surfacique et volumique. On calcule pour cela en chaque point du nuage un score de sphéricité basé sur les valeurs propres de la matrice de covariance du voisinage du point (voir section 2.2 pour plus de détails sur le calcul de la sphéricité). On lisse ensuite la valeur de sphéricité avec un filtre bilatéral. Quand la sphéricité est supérieure à $\frac{1}{3}$, on considère que le point appartient au modèle volumique sinon il appartient au modèle surfacique. Pour le modèle surfacique, nous appliquons une méthode classique de reconstruction de surface, RIMLS de [Öztireli *et al.* 2009] pour construire un maillage triangulaire. Pour le modèle volumique, on effectue un regroupement par k-means des points sous forme d'ellipsoïdes. La Figure 2.48 montre le modèle hybride produit à partir d'un nuage de points.

La simulation du LiDAR se fait par lancer de rayon dans la scène en calculant l'intersection du rayon avec les triangles et les ellipsoïdes rencontrés. Pour chaque instance de modèle (triangles et ellipsoïdes), on ajoute un score de perméabilité. Ce score est calculé lors de la construction du modèle avec le nombre de rayons ayant traversé le modèle sur le nombre total de rayons (ceux ayant traversé et ceux ayant touché le modèle). Lors de la simulation, un simple test contre ce score de perméabilité permet de savoir si on considère que le rayon simulé doit continuer à traverser la scène ou non. La différence entre le modèle surfacique et le modèle volumique se situe lors du calcul de la distance retournée par le lancer de rayon. On ajoute un bruit gaussien dans le cas surfacique alors que dans le cas volumique, l'intersection du rayon avec l'ellipsoïde est une gaussienne qui déterminera la distance à retourner. Le bruit est ainsi encoder dans le modèle dans

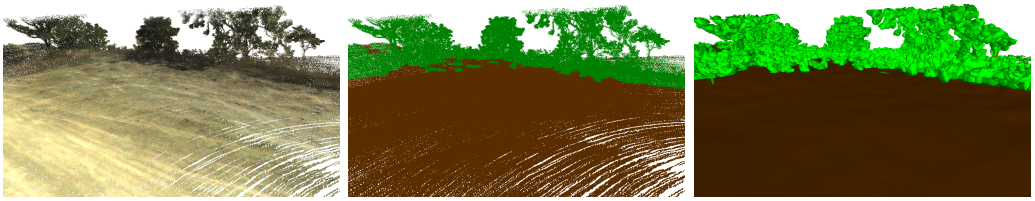

FIGURE 2.48 – Exemple de modèle hybride surfacique et volumique produit à partir d'un nuage de points. A gauche, nuage de points coloré provenant d'une acquisition par système mobile LiDAR + caméra. Au milieu, classification basée sur le score de sphéricité (marron = surfacique, vert = volumique). A droite, modèle hybride rendu avec des triangles pour les zones surfaciques et des ellipsoïdes pour les zones volumiques.

le cas volumique et va varier suivant les zones et le type de terrain. La Figure 2.49 présente le principe général en 2D de notre méthode avec la modélisation à partir de données nuage de points et la simulation d'un LiDAR. Même si le modèle volumique pourrait être utilisé pour les zones surfaciques, il a besoin de plus de paramètres que le modèle surfacique et a aussi tendance à surestimer le bruit dans les zones surfaciques. C'est pourquoi nous construisons un modèle hybride surfacique et volumique.

FIGURE 2.49 – Représentation schématique en 2D de notre modélisation et simulation. A gauche, environnement que l'on souhaite simuler avec de la végétation, un grillage et les façades d'un bâtiment. Au milieu, un nuage de points acquis avec un système de type mobile LiDAR. A droite, le modèle hybride surfacique et volumique ainsi que la simulation d'un LiDAR par lancer de rayons dans ce modèle.

La Figure 2.50 présente un exemple de l'intérêt de notre modélisation. On peut y voir un arbuste qui a été scanné en statique. Pour deux zones très proches, on peut voir que la réponse du LiDAR est très différente avec à gauche un histogramme de distances mesurées de forme gaussienne (sûrement une branche) et à droite un histogramme beaucoup plus allongé (de la végétation peu dense). Nous avons pu reproduire ces deux comportements avec notre modélisation volumique. Pour cela, nous avons simulé 150 000 lancers de rayons dans notre modèle volu-

mique puis construit l'histogramme des distances (courbe en bleu sur la Figure 2.50).

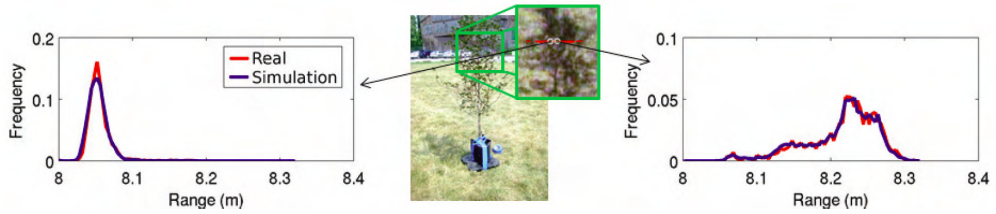

FIGURE 2.50 – Histogrammes de distances d'un LiDAR sur deux zones proches d'un arbuste. Environ 1500 distances mesurées ont permis de générer les courbes rouges. En bleu, les histogrammes générés par simulation à partir de notre modélisation volumique avec 150 000 lancers de rayons simulés.

Pour tester la validité de notre méthode de modélisation pour la simulation de LiDAR, nous avons fait des expérimentations avec plusieurs passages d'un système mobile LiDAR dans la même zone. Le premier passage sert à la construction du modèle et les autres passages vont servir de vérité pour la simulation d'un LiDAR (en utilisant la même trajectoire). On compare ainsi le nuage de points produit par la simulation avec le nuage de points acquis lors des passages différents du passage utilisé pour construire le modèle. La Figure 2.51 présente un résultat qualitatif avec au-dessus le nuage vérité terrain (ce que l'on cherche à simuler), au milieu le nuage produit avec un modèle surfacique (maillage + bruit gaussien) et en dessous le nuage produit avec notre modèle hybride surfacique et volumique. On peut observer que la végétation est bien modélisée avec notre méthode. Nous avons aussi mesuré sur plusieurs expérimentations la distance nuage à nuage (entre le nuage vérité terrain et le nuage simulé) et passons d'une erreur moyenne de 10.09 cm avec un modèle surfacique avec bruit gaussien à une erreur moyenne de 4.38 cm avec notre modèle hybride.

2.3.3.2 Simulation d'écoulement de fluides à partir de nuage de points

Les simulations numériques de phénomènes physiques comme l'aérodynamisme, l'acoustique, la propagation d'ondes et l'écoulement de fluides demandent en entrée des modèles 3D de plus en plus précis de l'objet ou de la scène étudiée. Ces modèles sont habituellement réalisés à la main par CAO (Conception Assistée par Ordinateur) mais le processus reste généralement long et complexe.

Dans le cadre de la thèse d'Hassan Bouchiba et en collaboration avec l'Institut de Calcul Intensif de l'Ecole Centrale de Nantes (spécialisé dans la simulation d'écoulements de fluides en volume immersif), nous avons travaillé sur la possibilité d'utiliser directement des nuages de points en entrée de leur simulateur.

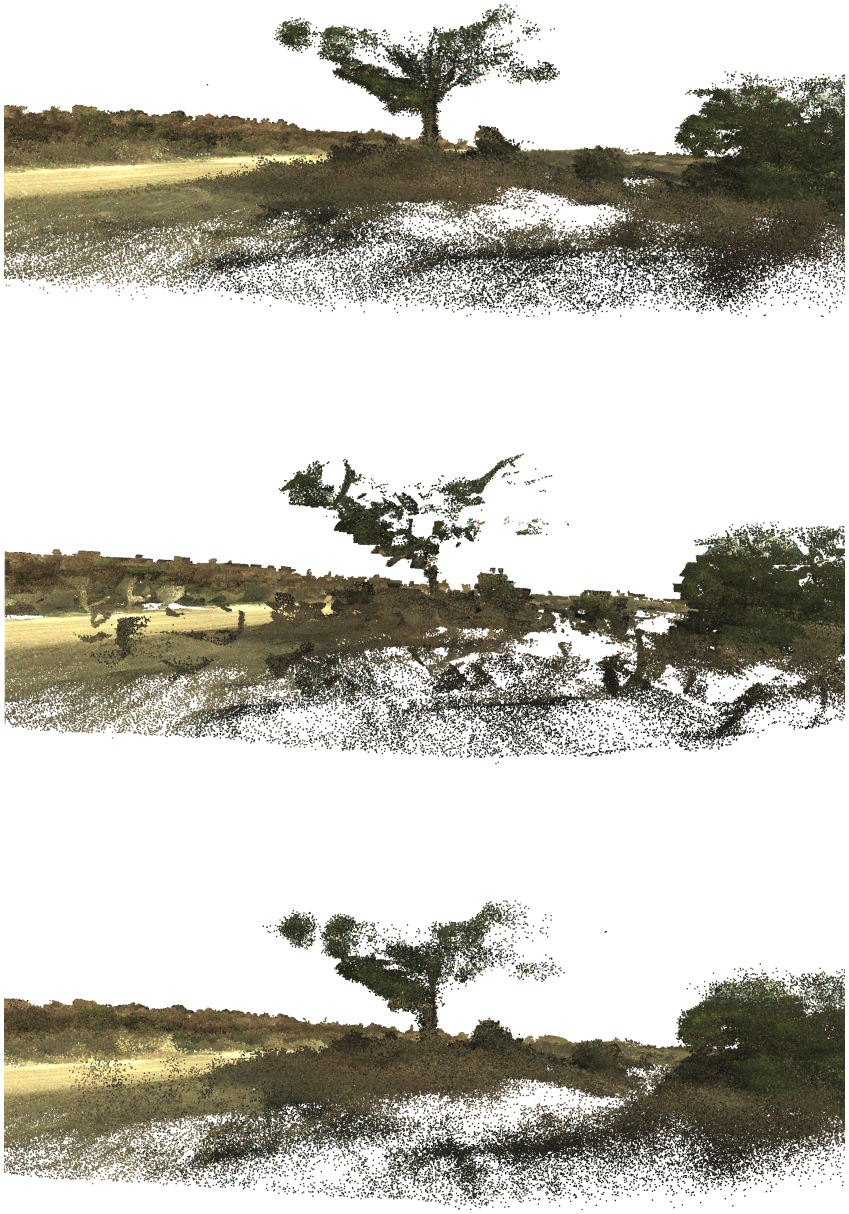

FIGURE 2.51 – Comparaison de simulations d'un LiDAR dans un environnement réel avec de la végétation. Au-dessus, nuage de points vérité terrain (données 3D correspondant à la trajectoire que l'on simule). Au milieu, nuage produit par un modèle surfacique avec bruit gaussien. En dessous, nuage produit par notre modèle hybride surfacique et volumique.

La Figure 2.52 montre les différentes méthodes pour passer d'un objet réel à une simulation d'écoulement de fluides. Notre méthode publiée dans [Bouchiba *et al.* 2020] n'utilise pas de maillage surfacique comme intermédiaire et va directement construire un maillage volumique dans tout le domaine (c'est pourquoi on appelle cela une simulation en volume immersif). Le maillage volumique anisotrope est ajusté autour de l'écoulement et de la surface de l'objet grâce à une métrique d'erreur calculée a posteriori. Pour représenter la distance signée de tout point de l'espace à la surface, nous avons défini une fonction scalaire EIMLS (*Extended Implicit Moving Least Squares*). La simulation d'écoulements de fluides s'opère ensuite en utilisant le maillage comme discrétisation de l'espace avec un solveur en éléments finis des équations de Navier-Stokes en régime incompressible. A notre connaissance, il n'existe pas de travaux similaires dans la littérature.

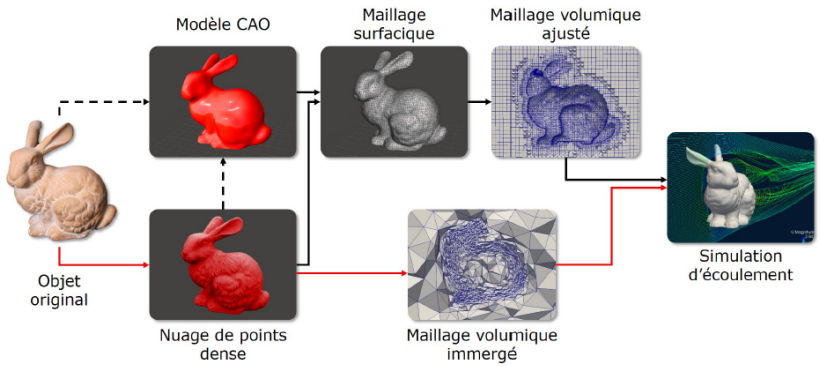

FIGURE 2.52 – Différentes méthodes pour simuler des écoulements autour d'objets réels. Les flèches noires en pointillées représentent des étapes manuelles. Les flèches noires pleines représentent des méthodes automatiques ou semi-automatiques. Les flèches rouges présentent notre méthode. Dans notre cas, le nuage de points est directement utilisé pour le calcul du maillage volumique qui sert à la simulation.

La Figure 2.53 montre pour un jeu de données 2D les différentes étapes de la méthode. Le nuage de points 2D a été obtenu en effectuant une coupe du nuage de points 3D *Bunny* de Stanford.

Pour s'adapter à la géométrie de l'objet, le maillage anisotrope calcule en chaque nœud la distance signée à la surface. Notre contribution consistait à définir une fonction permettant en tout point de l'espace (en l'occurrence en tout nœud du maillage volumique) de calculer une distance signée à la surface de l'objet.

Dans le domaine de la reconstruction de surface à partir de nuages de points, IMLS pour *Implicit Moving Least Squares* de [Kolluri 2008] est une méthode clas-

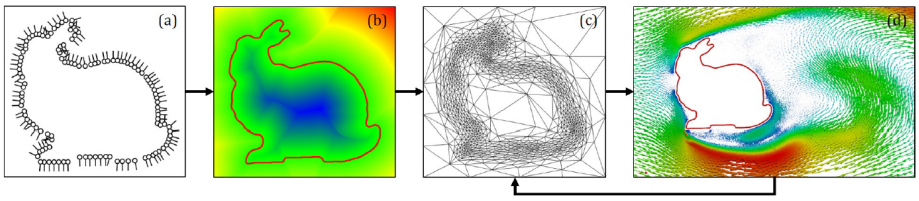

FIGURE 2.53 – Aperçu en 2D de notre méthode. (a) Nuage de points avec des normales orientées. (b) Fonction scalaire implicite calculée par EIMLS. L'iso-zéro est représentée en rouge. (c) Maillage anisotrope adapté autour de la surface du nuage de points. (d) Calcul de l'écoulement par la méthode des éléments finis en volume immergé puis mise à jour du maillage à chaque pas de temps.

sique où la surface est représentée par l'iso-zéro d'une fonction scalaire $f_{IMLS}(x)$:

$$f_{IMLS}(x) = \frac{\sum_{i \in N} w_i(x)(x - p_i) \cdot n_i}{\sum_{i \in N} w_i(x)}$$

avec le terme $(x - p_i) \cdot n_i$ la distance signée au plan représenté par le point p_i avec sa normale n_i et w_i le poids associé au point p_i.

Avec la formulation IMLS classique, on choisit un poids :

$$w_i(x) = \exp(-\frac{||p_i - x||^2}{2h^2})$$

Cette fonction $f_{IMLS}(x)$ permet aussi d'obtenir en tout point x une approximation de la distance signée à la surface représentée par le nuage de points. Mais comme le montre la Figure 2.54, cette formulation pose des problèmes lorsque l'on veut calculer la distance signée à un point éloigné de la surface (dû à des problème numériques dans le calcul du poids $w_i(x)$). Nous avons donc proposé EIMLS (pour *Extended Implicit Moving Least Squares*) en utilisant un h variable $h^E(x) = \max(\frac{||p_c(x)-x||}{l_\gamma}, h)$ avec $p_c(x)$ le point du nuage le plus proche de x et l_γ un paramètre dépendant de la précision numérique.

La Figure 2.55 présente dans un cas 2D comment le maillage volumique anisotrope va s'adapter à la géométrie de la surface (grâce à notre formulation EIMLS) mais aussi au fluide (le maillage est plus fin là où la vitesse du fluide est plus grande). Pour un nombre de nœuds constant, la simulation sera ainsi beaucoup plus précise qu'en utilisant un maillage non adaptatif à la géométrie et à la vitesse du fluide.

La Figure 2.56 présente un résultat sur le nuage de points 3D du *Bunny*. On peut observer les recirculations du fluide près des oreilles. La simulation a été produite avec un maillage de 1 million de nœuds répartis sur 100 cœurs CPU

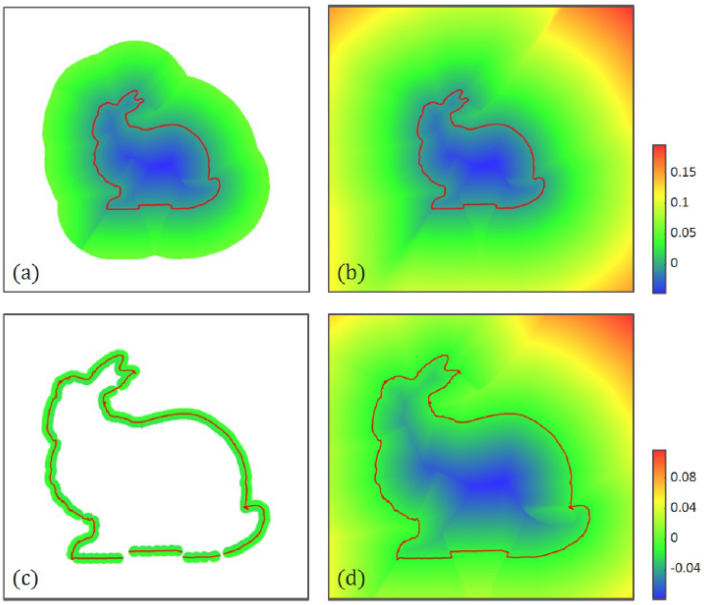

FIGURE 2.54 – Comparaison entre IMLS (à gauche) et EIMLS (à droite). La ligne du haut est obtenue avec $h = 0.0015$ et la ligne du bas avec $h = 0.0001$. IMLS n'est pas défini sur tout le domaine (parties blanches) alors que EIMLS l'est dans les deux cas.

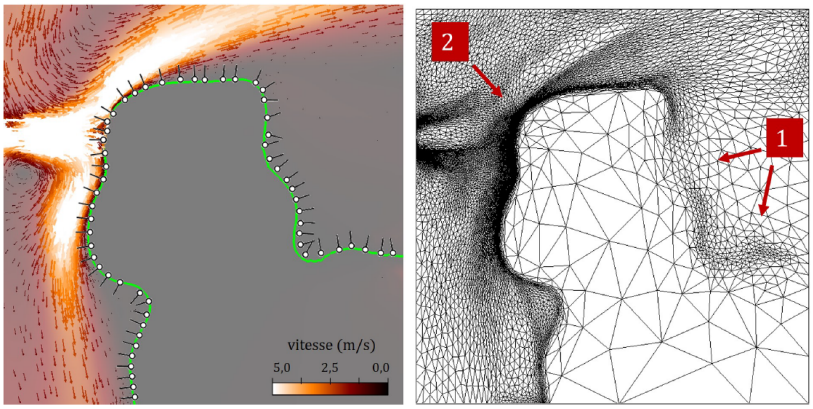

FIGURE 2.55 – A gauche, écoulement calculé par éléments finis à partir d'un nuage de points 2D. L'air est soufflé à partir d'une petite entrée sur la gauche du domaine à 5 m/s. A droite, le maillage correspondant qui s'adapte à la géométrie de l'objet mais aussi à la vitesse du fluide.

et a demandé en tout 3 heures de calculs. Tous les calculs ont été réalisés avec le code ICI-lib développé à l'Institut de Calcul Intensif de l'école Centrale de Nantes.

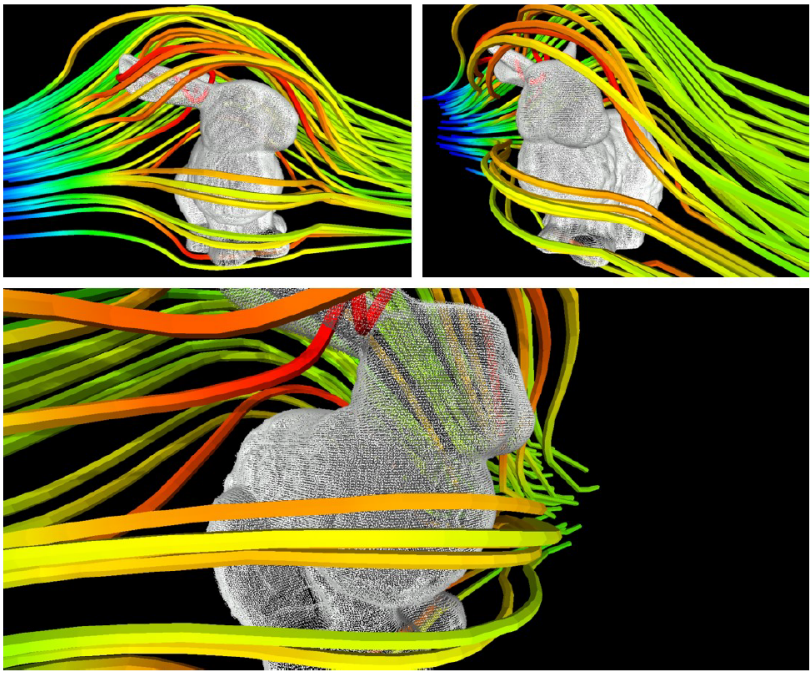

FIGURE 2.56 – Simulation d'écoulement de fluides à partir du nuage de points 3D du *Bunny*.

La Figure 2.57 montre l'exemple d'une simulation d'une ventilation dans une salle acquise par scanner fixe. L'air est soufflé par un trou carré de 20cm de côté au niveau du plafond. La salle semble partitionner le fluide d'air en deux. Ce type de simulation pourrait permettre d'évaluer plusieurs solutions d'aménagement d'une pièce ou d'un ensemble de pièces à partir d'une numérisation 3D d'un bâtiment existant.

La Figure 2.58 est un exemple de simulation de déplacement d'air dans une rue à partir d'un nuage de points acquis par système mobile LiDAR. La simulation a été réalisée sur un maillage comportant 33,000 nœuds : de l'air est soufflé d'un côté de la rue, dans l'axe de la route. Les résultats montrent que la rue induit un pincement de l'écoulement qui a pour effet d'accélérer l'air au centre de la rue. Cet effet d'"entonnoir" est bien connu dans les villes. Simuler l'écoulement de l'air à l'échelle d'une rue ou d'un quartier pourrait être utilisé par exemple, pour étudier l'effet de la pollution sur les bâtiments.

En conclusion, nous avons vu des applications très différentes de l'utilisation de nuages de points de scènes ou d'objets réels. La modélisation de ces données 3D (sous forme de maillage surfacique, modèle volumique ou fonction implicite...) est

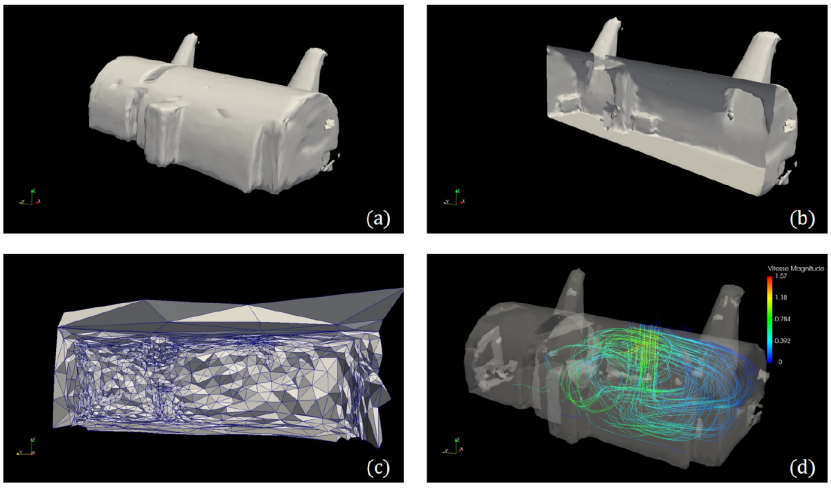

FIGURE 2.57 – Simulation d'écoulement de fluides à partir d'un nuage de points d'une salle de MINES ParisTech acquis par scanner fixe FARO X130. (a) Iso-zéro de la surface extraite par la fonction EIMLS à partir du maillage adaptatif. (b) Coupe de l'iso-zéro pour voir l'intérieur de la salle. (c) Coupe du maillage volumique anisotrope. (d) Simulation d'une ventilation venant du plafond dans le nuage de points.

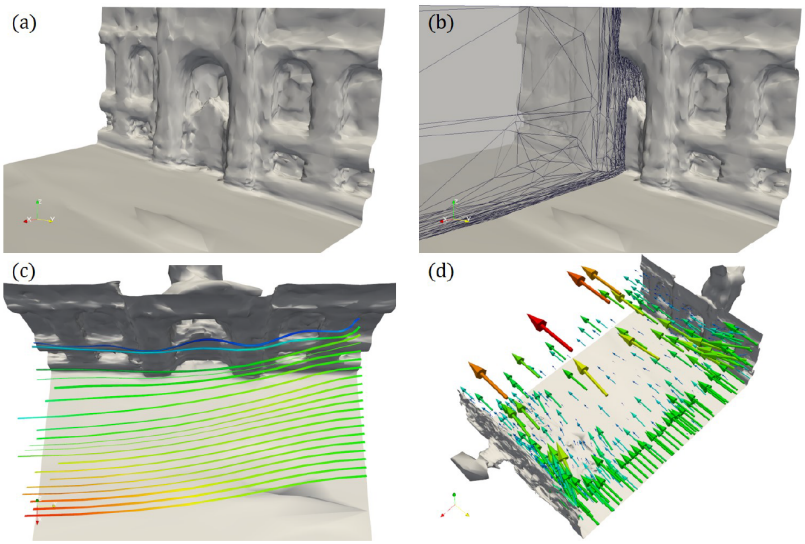

FIGURE 2.58 – Simulation d'écoulement de fluides à partir d'un nuage de points d'une rue à Paris scannée par système mobile LiDAR. (a) Iso-zéro de la surface à partir du maillage adaptatif. (b) Coupe du maillage volumique anisotrope. (c+d) Lignes de courant et vitesse du fluide.

une partie importante de mes travaux en cohérence avec les parties précédentes (création de données 3D et analyse). On peut voir un fil conducteur dans tous ces travaux : la capacité de modéliser et/ou simuler le réel le plus fidèlement possible.

Chapitre 3

Perspectives de recherche

Contents

3.1 Systèmes mobiles d'acquisition de données 3D 69
3.2 Recalage, classification et sémantisation 71
3.3 Rendu de nuages de points, reconstruction 3D et simulation 72

Je vais développer dans cette partie les perspectives pour mes futurs travaux de recherche. Les thématiques que je souhaite développer sont dans la lignée de mes travaux précédents. Certains des travaux présentés ici ont d'ailleurs déjà été commencé par des doctorants que j'encadre.

3.1 Systèmes mobiles d'acquisition de données 3D

Les systèmes d'acquisitions mobiles LiDAR + caméra sont amenés à se développer et ainsi à apporter de nouvelles applications que nous n'imaginons pas encore. Je souhaite donc continuer à travailler sur ces systèmes et à améliorer l'exploitation des données produites.

Dans le cadre d'un projet de recherche FUI REPLICA, nous sommes en train de tester de nouveaux capteurs de type caméras hyperspectrales ajoutées aux systèmes mobiles pour faciliter la segmentation sémantique des données et la classification des matériaux rencontrés dans l'environnement urbain ou autoroutier. La Figure 3.1 présente L3D2 avec deux caméras hyperspectrales ajoutées au système LiDAR Velodyne + caméra Ladybug5. De premières données ont été acquises et seront prochainement publiées.

Je souhaite aussi continuer à explorer les méthodes d'odométrie basée LiDAR ou hybride LiDAR/caméra. Nous venons de démarrer une nouvelle thèse CIFRE basée sur l'idée d'aller au-delà des méthodes classiques géométriques en utilisant des réseaux profonds pour l'inférence de poses. L'intérêt est de pouvoir apprendre des poses à partir d'un grand volume de données de façon non supervisé. Les travaux récents comme DeepLO [Cho *et al.* 2019] vont dans cette direction et semblent prometteurs.

FIGURE 3.1 – Système d'acquisition L3D2 (LIDAR Velodyne HDL32 + caméra Ladybug5) avec de nouvelles caméras hyperspectrales Photonfocus MV1 (VIS 16 bandes + NIR 25 bandes).

Enfin, j'aimerais aussi traiter le sujet de la localisation précise temps réel de véhicules à partir de capteurs LiDARs en utilisant une carte HDMap dense de type nuage de points. Les cartes HDMap peuvent d'ailleurs être produites à partir de nuages de points de système mobiles (une application importante de la segmentation sémantique, voir Figure 3.2).

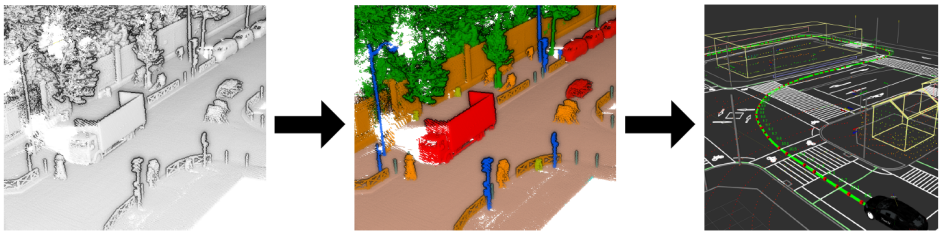

FIGURE 3.2 – Une application importante de la segmentation sémantique de nuages de points : la construction de HDMap pour le véhicule autonome.

3.2 Recalage, classification et sémantisation

Le recalage consiste à trouver la transformation entre deux nuages de points ayant généralement un recouvrement partiel. Il s'agit d'un sujet difficile, surtout lorsque les deux nuages de points n'ont pas de première estimation de la transformation et que le recouvrement est faible. Dans le cadre d'une thèse en lien avec le laboratoire d'archéologie de l'ENS, nous étudions le recalage de motifs celtiques (principalement sur des pièces de monnaie).

La Figure 3.3 présente les données d'entrée, qui sont des nuages de points de pièces de monnaie scannées par un SmartScan à lumière structurée (précision de mesure inférieure à $50\mu m$). L'objectif est d'être capable de classer ces pièces suivant leur coin d'origine (outil servant à frapper la pièce). Les archéologues le font généralement à la main en comparant les pièces deux à deux (impossible à faire pour des trésors contenant plusieurs milliers de pièces). Pour automatiser la tâche, nous avons imaginé une chaîne de traitement avec d'abord le recalage des motifs puis une classification des coins par comparaison des histogrammes de distances pièce à pièce. La tâche la plus complexe reste celle du recalage entre deux pièces car les motifs varient peu entre deux coins et varie quand même au sein d'un même coin (érosion avec le temps de la pièce). La Figure 3.4 présente un premier résultat de classification d'une centaine de pièces suivant le coin utilisé pour leur fabrication.

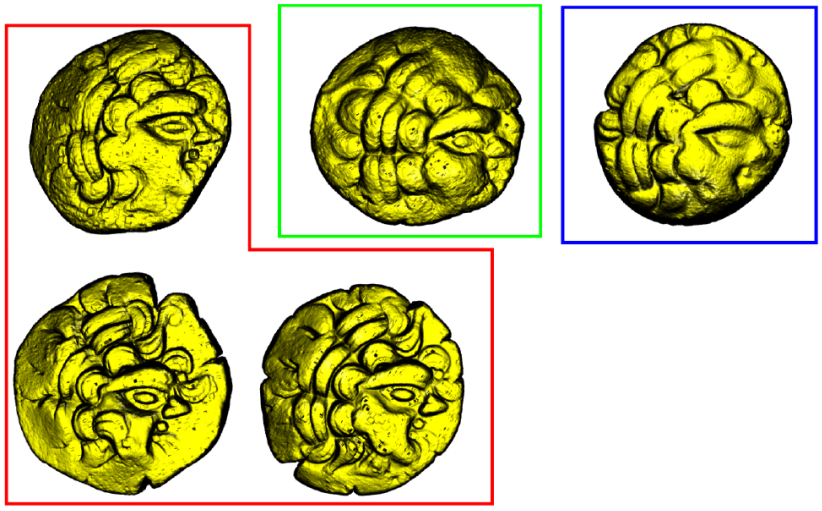

FIGURE 3.3 – Exemples de nuages de points de pièces de monnaie celtique. Les trois pièces dans le cadre en rouge ont été frappées par le même coin (outil utilisé pour leur fabrication). Les deux autres pièces ont été frappées chacune par un coin différent. On peut voir la grande similarité de motif pour toutes les pièces et donc la difficulté à recaler puis classifier correctement ce type de données.

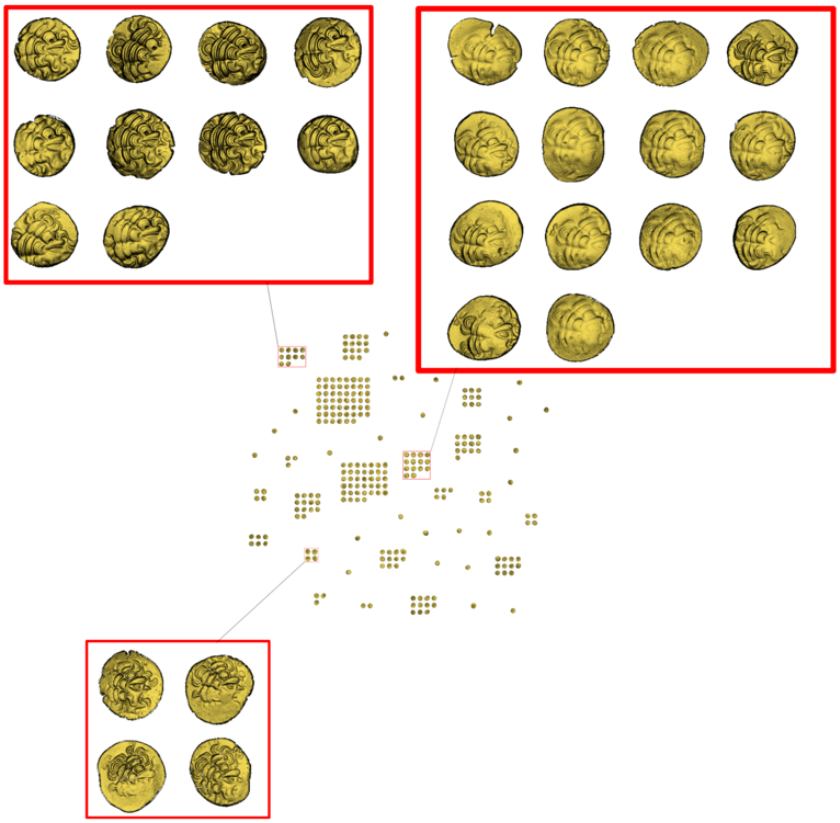

FIGURE 3.4 – Premiers résultats de classification de pièces de monnaies celtiques suivant leur coin.

La segmentation sémantique est toujours un sujet d'actualité pour les données de type nuages de points. C'est un sujet important pour la compréhension de la scène en conduite autonome (principalement les objets mobiles). Les méthodes *Deep* vues dans le chapitre précédent commencent à bien fonctionnent sur des données denses mais des progrès restent à faire sur des données éparses comme lors du traitement d'une seule *frame* LiDAR. J'aimerais ainsi améliorer les architectures sur nuages de points pour être plus facilement exploitables en temps réel. Au-delà de la segmentation sémantique, il reste de nombreux progrès à faire sur la capacité à bien détecter les objets dans une *frame* LiDAR et les suivre dans le temps.

3.3 Rendu de nuages de points, reconstruction 3D et simulation

J'aimerais continuer à travailler sur les méthodes de rendu photo-réaliste à partir de nuages de points. Les données produites par des systèmes mobiles de cartographie pourraient être utilisées directement comme environnement virtuel

dans des simulateurs (au lieu de passer par une étape de reconstruction comme vu au chapitre précédent). Cela demande à trouver comment faire du lancer de rayon dans des nuages de points (qui se fait traditionnellement dans des modèles de type maillage). Nous avons lancé une thèse sur ce sujet. La Figure 3.5 présente des premiers résultats de rendu par lancer de rayons avec une technique dite de *splatting* sur un nuage de points acquis par scanner fixe (façade MINES ParisTech).

FIGURE 3.5 – A gauche, rendu par lancer de rayons sur un nuage de points de la façade de MINES ParisTech acquis pas un scanner fixe FARO (environ 400 000 *splats*). En haut à droite, zoom sur un rendu en utilisant seulement les points. En bas à droite, rendu par *splatting* à partir du nuage de points.

En conclusion, je souhaite continuer mes recherches sur les nuages de points d'environnements réels ou d'objets à travers différentes applications. On peut voir pour les prochaines années une application transversale aux différents axes de recherche que je souhaite développer : il s'agit du traitement de données de type LiDAR pour le véhicule autonome (pour les tâches de localisation et de perception mais aussi de simulation). Cela reste un sujet scientifique complexe mais avec un enjeu important ouvrant de nombreux champs de recherche.

Deuxième partie

Parcours scientifique

Chapitre 4

Parcours professionnel

Parcours professionnel depuis la soutenance de thèse :

- 2010 : Doctorat en Informatique temps-réel, robotique et automatique au Centre de Robotique de MINES ParisTech. Sujet de thèse “Traitements de nuages de points denses et modélisation 3D d’environnements par système mobile LiDAR/Caméra”
- 2010-2011 : Post-Doctorat au laboratoire NREC à Carnegie Mellon University aux Etats-Unis sur le sujet “Modélisation et simulation de scanners lasers pour véhicules terrestres autonomes”
- 2012-2015 : Post-Doc puis Tenure Track au centre de Robotique de MINES ParisTech en Vision 3D / Robotique Mobile.
- Depuis 2015 : Chargé de Recherche au centre de Robotique à MINES ParisTech en Vision 3D / Robotique mobile.

J’ai découvert les technologies de cartographie mobile et la production des nuages de points 3D lors de mon stage de fin d’étude à MINES ParisTech en 2006. J’ai été fasciné par cette technologie capable de “scanner” l’environnement extérieur avec une grande précision. Cela m’a donné l’envie de poursuivre par une thèse sur le même sujet qui s’est terminée en 2010. Cette thèse m’a permis de découvrir un grand nombre de thématiques autour de la donnée 3D. On retrouve d’ailleurs dans mon manuscrit de thèse (en ébauche) les différents axes explorés ces dix dernières années et décrits dans la première partie de ce manuscrit (acquisition de données, segmentation, reconstruction, rendu...). Mon grand intérêt pour ce domaine vient aussi de la multidisciplinarité avec un mélange de mathématiques appliquées, d’algorithmie, d’informatique graphique, de programmation et d’ingénierie.

Cette expérience de recherche m’a conforté dans l’idée de poursuivre une carrière dans le domaine de la recherche et l’enseignement supérieur. J’ai tout d’abord poursuivi par un Post-Doc aux Etats-Unis dans l’un des plus grands départements de Robotique au monde, le Robotics Institute à l’Université Carnegie Mellon. J’ai travaillé là-bas sur un projet de simulation réaliste de capteurs LiDAR basé données.

Je suis revenu en France en 2012 pour un autre Post-Doc à MINES ParisTech et j’ai postulé à un poste de Chargé de Recherche au Centre de Robotique des

MINES où j'avais fait ma thèse.

Depuis 2015, je suis Chargé de Recherche au Centre de Robotique de MINES ParisTech. Je codirige avec le Professeur François Goulette l'équipe NPM3D sur les thématiques de Vision 3D / Robotique Mobile décrites dans la première partie de ce manuscrit.

Chapitre 5

Encadrement et enseignement

Contents

5.0.1 Encadrement 79
5.0.2 Enseignement 81

5.0.1 Encadrement

Au cours de mes dix années de recherche, j'ai encadré 4 thèses, 3 Post-doctorants, 2 ingénieurs de recherche et 11 stagiaires.

Voici un petit résumé des 4 sujets de thèse donc j'ai été encadrant et qui ont été soutenus :

- 2013-2016 : Houssem Nouira (financement projet FUI TerraMobilita), **Affinement de relevés laser mobiles issus de LiDARs multi-couches** Les Systèmes Mobiles de Cartographie basés LIDAR permettent d'obtenir des cartes 3D de l'environnement, qui sont géo-référencées grâce à d'autres capteurs embarqués sur le véhicule : GPS, centrale inertielle, ou encore odomètre sont de tels capteurs qui permettent de localiser le véhicule mobile pendant la campagne d'acquisition. Toutefois, ces cartes manquent de précisions et un affinage des cartes est essentiel dans de nombreux cas d'applications où une précision fine est requise sur les cartes 3D, comme pour des applications de classifications par exemple. Lors de la création de cartes 3D géoréférencées, les données sont tout d'abord acquises par le capteur LIDAR et référencées dans le repère cartésien du laser à l'aide d'un calibrage intrinsèque du capteur d'acquisition. Ensuite, un calibrage extrinsèque du capteur permet de caractériser la transformation entre le capteur et le véhicule, et permet de référencer les données dans le repère « body », lié au véhicule d'acquisition. Enfin, avec la trajectoire du véhicule obtenue en fusionnant les données issues des GPS, centrale inertielle et odomètre, il est possible de géoréférencer les données lasers. Cette thèse a eu pour objectif d'affiner les relevés laser issus d'acquisitions effectuées à l'aide d'un véhicule mobile de cartographie, en optimisant plusieurs paramètres différents qui entrent en compte dans le géoréférencement des données. Nous nous sommes intéressés à l'affinement des nuages de points par optimisation des paramètres de calibrage extrinsèque dans un premier temps, puis par optimisation des

paramètres de calibrage intrinsèque, et enfin par optimisation des paramètres de translations liés à la trajectoire du véhicule mobile.

- 2014-2018 : Hassan Bouchiba (financement contrat doctoral MINES ParisTech), **Contributions en traitements basés points pour le rendu et la simulation en mécanique des fluides**
  Nous nous sommes intéressés dans cette thèse à l'utilisation de nuages de points comme unique représentation explicite de surface. La première contribution proposée était une nouvelle méthode de rendu de nuages de points bruts et massifs par opérateurs pyramidaux en espace image. Cette nouvelle méthode s'applique aussi bien à des nuages de points d'objets scannés, que de scènes complexes. La succession d'opérateurs en espace image permet alors de reconstruire en temps réel une surface et d'en estimer des normales, ce qui permet par la suite d'en obtenir un rendu par ombrage. De plus, l'utilisation d'opérateurs pyramidaux en espace image permet d'atteindre des fréquences d'affichage plus élevées d'un ordre de grandeur que l'état de l'art. La deuxième contribution présentée était une nouvelle méthode de simulation numérique en mécanique des fluides en volumes immergés par reconstruction implicite étendue. La méthode proposée se base sur une nouvelle définition de surface implicite par moindres carrés glissants étendue à partir d'un nuage de points. Cette surface est alors utilisée pour définir les conditions aux limites d'un solveur Navier-Stokes par éléments finis en volumes immergés, qui est utilisé pour simuler un écoulement fluide autour de l'objet représenté par le nuage de points. Le solveur est interfacé à un mailleur adaptatif anisotrope qui permet de capturer simultanément la géométrie du nuage de points et l'écoulement à chaque pas de temps de la simulation.

- 2015-2019 : Xavier Roynard (financement contrat doctoral MINES ParisTech), **Sémantisation à la volée de nuages de points 3D acquis par systèmes embarqués**
  Cette thèse se trouvait à la confluence de deux mondes en pleine explosion : la voiture autonome et l'intelligence artificielle (particulièrement l'apprentissage profond). Le premier tirant profit du deuxième, les véhicules autonomes utilisent de plus en plus de méthodes d'apprentissage profond pour analyser les données produites par ses différents capteurs (dont les LiDARs) et pour prendre des décisions. Alors que les méthodes d'apprentissage profond ont révolutionné l'analyse des images (en classification et segmentation par exemple), elles ne produisent pas des résultats aussi spectaculaires sur les nuages de points 3D. Une première raison est que les jeux de données de nuages de points 3D annotés sont rares et de qualité moyenne. Nous avons donc présenté dans cette thèse un nouveau jeu de données réalisé par acquisition mobile pour produire suffisamment de données et annoté à la main pour assurer une bonne qualité de segmentation. De plus ces jeux de

données sont par nature déséquilibrés en nombre d'échantillon par classe et contiennent beaucoup d'échantillons redondants, nous avons donc proposé une méthode d'échantillonnage adaptée à ces jeux de données. Un autre problème rencontré quand on essaye de classifier un point à partir de son voisinage sous forme de grille de voxels est le compromis entre un pas de discrétisation fin (pour décrire précisément la surface) et une grille de taille élevée (pour aller chercher du contexte). Nous avons ainsi proposé également des méthodes de réseaux tirant profit de voisinages multi-échelles. Ces méthodes atteignaient l'état de l'art des méthodes de classification par points sur plusieurs benchmarks publiques. Enfin pour respecter les contraintes imposées par les systèmes embarqués (traitement en temps réel et peu de puissance de calcul), nous avons testé une méthode qui permet de n'appliquer les couches convolutives que là où il y a de l'information à traiter.

- 2016-2019 : Hugues Thomas (financement CIFRE Terra3D), **Apprentissage de nouvelles représentations pour la sémantisation de nuages de points 3D**
  Dans cette thèse, nous avons proposé une nouvelle définition de voisinages sphériques multi-échelles, une comparaison avec les k plus proches voisins multi-échelles, une nouvelle stratégie d'apprentissage actif, la segmentation sémantique des nuages de points à grande échelle, et une étude de l'influence de la densité dans les représentations multi-échelles. En se basant sur ces contributions, nous avons introduit le « Kernel Point Convolution » (KP-Conv), qui utilise des voisinages sphériques et un noyau défini par des points. Ces points jouent le même rôle que les pixels du noyau des convolutions en image. Nos réseaux convolutifs ont surpassé les approches de segmentation sémantique de l'état de l'art dans presque toutes les situations.

5.0.2 Enseignement

En tant qu'enseignant-chercheur à MINES ParisTech, j'ai été impliqué dans différents cours et charges d'enseignement. Ci-dessous, voici la liste des cours les plus représentatifs :

- 2012-2017 : Co-Responsable puis Responsable de l'enseignement MIG ("Métiers Ingénieur Généraliste") pour les élèves ingénieurs de MINES ParisTech
- 2014-2018 : Implication dans le cours "Vision et Perception pour la Robotique" au Master 2 SAR de l'Université Paris 6 (Sorbonne Université)
- 2018-2020 : Responsable du cours Analyse Complexe, tronc commun 2A du cycle ingénieur civil de MINES ParisTech
- Depuis 2013 : Co-Responsable du cours "Nuages de points 3D et modélisation" du Master MVA de l'ENS Paris-Saclay et du Master IASD de l'Université PSL
- Depuis 2017 : Cours "Perception 3D" à ParisTech Shanghai Jiao Tong

- Depuis 2020 : Co-Responsable du cours Optimisation Convexe, tronc commun 1A du cycle ingénieur civil de MINES ParisTech

L'un des enseignements les plus innovants et intéressants que j'ai encadré a été le MIG ("Métiers Ingénieur Généraliste") donné aux élèves 1A du cycle ingénieur civil des MINES. Il s'agit d'un enseignement avec un petit groupe d'élèves (environ 10-12) qui sont pris en charge à temps plein pendant 3 semaines sur une thématique proche du métier d'ingénieur et/ou de chercheur. Le temps était divisé en 3 parties avec une semaine de cours/conférence par des experts, quelques jours de visites (entreprises, usines, laboratoires), une semaine de mini-projet encadrés et enfin quelques jours de rédaction en commun d'un rapport et d'une présentation faite devant un jury externe. Chaque année, je montais le cours avec une thématique nouvelle : par exemple en 2015 "la cartographie 3D appliquée aux véhicules autonomes" et en 2016 la thématique "la technologie 3D au service de l'archéologie et de la sauvegarde du patrimoine".

Chapitre 6

Diffusion et liste des publications

Contents

6.1 Diffusion **83**

6.1.1 Création de la start-up Terra3D 83

6.2 Projets de recherche **84**

6.3 Liste des publications **85**

6.3.1 Articles de revues avec comité de lecture 85

6.3.2 Articles de conférence avec comité de lecture 85

6.1 Diffusion

Les travaux de recherche que j'ai encadré à MINES ParisTech ont donné lieu à de nombreuses communications nationales, internationales et des publications dans des conférences et des revues avec comité de lecture (liste ci-dessous).

Depuis 2019, nous organisons chaque année un séminaire "Analyse et représentations de nuages de points 3D" à MINES ParisTech pour la communauté scientifique et industrielle. Ce séminaire a attiré plus de 100 participants en 2020.

Enfin, les travaux menés ont permis la création d'une start-up en 2015 qui s'est depuis bien développé et compte en 2020 sept salariés.

6.1.1 Création de la start-up Terra3D

J'ai été en Octobre 2015 l'un des 5 fondateurs de la start-up Terra3D, spécialisée dans la vente de logiciels pour l'analyse automatique de nuages de points.
Voici le site Internet de la start-up Terra3D : `http://www.terra3d.fr/`
Depuis 2015, je suis conseiller scientifique de cette start-up qui permet de valoriser les travaux de recherche de l'équipe NPM3D (voir Figure 6.1).

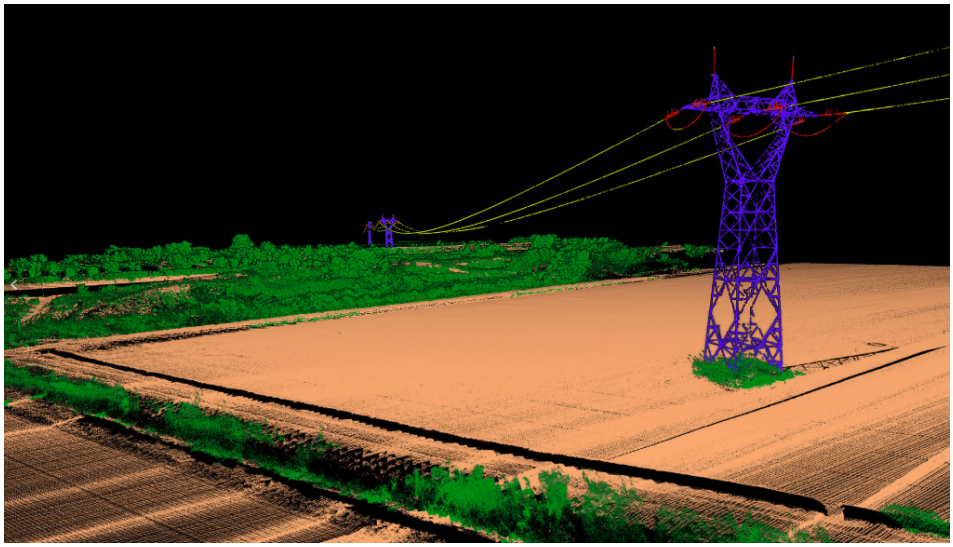

FIGURE 6.1 – Produit Powerlines de la start-up Terra3D

6.2 Projets de recherche

J'ai été impliqué dans plusieurs projets de recherche permettant le financement des travaux présentés dans ce manuscrit :

— FUI TerraMobilita (2012-2015) : mise au point de nouveaux processus automatisés de création et de mise à jour de cartes 3D de voirie urbaine et développement de nouveaux services et applications pour les collectivités.
— Carnot Maillage (2013) : projet qui vise à combiner deux domaines en plein essor : l'imagerie 3D et la simulation numérique, en se concentrant sur un verrou : le maillage. L'objectif est de faire le pont entre les données issues des capteurs 3D et les représentations numériques avec pour ambition d'ouvrir la voie de la modélisation numérique à partir du réel.
— FUI SIMVIR (2013-2015) : enrichir les outils de production d'environnement virtuel pour les simulateurs de conduite, intégrer et optimiser des outils permettant de produire un modèle numérique 3D à grande échelle (plusieurs dizaine de kilomètres urbains) et de précision centimétrique pour être fidèle à la réalité.
— PSL Celtes3D (2016-2018) : projet sur le savoir-faire des celtes à l'épreuve des nouvelles techniques d'imagerie numérique.
— FUI REPLICA (2018-2022) : compléter les plateformes de simulation existantes avec le développement de nouvelles briques afin de franchir le gap attendu en test du véhicule autonome.
— PSL ARCHEO3D (2018-2021) : reconnaissance automatique de motifs sculptés sur des surfaces courbes 3D et son application à l'art celtique.

6.3 Liste des publications

6.3.1 Articles de revues avec comité de lecture

— [Bouchiba *et al.* 2020] Hassan Bouchiba, Simon Santoso, Jean-Emmanuel Deschaud, Luisa Rocha-Da-Silva, François Goulette et Thierry Coupez. *Computational Fluid Dynamics on 3D Point Set Surfaces.* Journal of Computational Physics: X, page 100069, 2020

— [Roynard *et al.* 2018c] Xavier Roynard, Jean-Emmanuel Deschaud et François Goulette. *Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification.* The International Journal of Robotics Research, vol. 37, no. 6, pages 545–557, 2018

— [Craciun *et al.* 2017] Daniela Craciun, Jean-Emmanuel Deschaud et François Goulette. *Automatic Ground Surface Reconstruction from mobile laser systems for driving simulation engines.* Journal of Simulation, vol. 93, no. 3, pages 201–211, 2017

— [Nouira *et al.* 2017] Houssem Nouira, Jean-Emmanuel Deschaud et François Goulette. *Point cloud refinement with self-calibration of a mobile multibeam lidar sensor.* The Photogrammetric Record, vol. 32, no. 159, pages 291–316, 2017

— [Barczyk *et al.* 2015] Martin Barczyk, Silvère Bonnabel, Jean-Emmanuel Deschaud et François Goulette. *Invariant EKF Design for Scan Matching-Aided Localization.* IEEE Transactions on Control Systems Technology, vol. 23, no. 6, pages 2440–2448, 2015

— [Browning *et al.* 2012] Brett Browning, Jean-Emmanuel Deschaud, David Prasser et Peter Rander. *3D Mapping for high-fidelity unmanned ground vehicle lidar simulation.* The International Journal of Robotics Research, vol. 31, no. 12, pages 1349–1376, 2012

— [Deschaud *et al.* 2010] Jean-Emmanuel Deschaud, Xavier Brun et François Goulette. *Colorisation et texturation temps réel d'environnements urbains par système mobile avec scanner laser et caméra fish-eye.* Revue Française de Photogrammétrie et de Télédétection, Septembre 2010

6.3.2 Articles de conférence avec comité de lecture

— [Horache *et al.* 2020] Sofiane Horache, François Goulette, Jean-Emmanuel Deschaud, Thierry Lejars et Katherine Gruel. *Automatic Clustering of Celtic Coins based on 3D Point Cloud Pattern Analysis.* ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. V-2-2020, pages 973–980, 2020

— [Thomas *et al.* 2019] Hugues Thomas, Charles Ruizhongtai Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette et Leonidas Guibas. *KPConv: Flexible and Deformable Convolution for Point Clouds.* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 6410–6419, 2019

— [Duque-Arias *et al.* 2019] David Duque-Arias, Santiago Velasco-Forero, François Goulette, Jean-Emmanuel Deschaud et Beatriz Marcotegui. *A graph-based color lines model for image analysis.* International Conference on Image Analysis and Processing, Lecture Notes in Computer Science, Trento, Italy, Septembre 2019

— [Roynard *et al.* 2018a] Xavier Roynard, Jean-Emmanuel Deschaud et François Goulette. *Classification of Point Cloud for Road Scene Understanding with Multiscale Voxel Deep Network.* 10th workshop on Planning, Perceptionand Navigation for Intelligent Vehicules PPNIV'2018, Madrid, Spain, Octobre 2018

— [Roynard *et al.* 2018b] Xavier Roynard, Jean-Emmanuel Deschaud et François Goulette. *Paris-Lille-3D: A Point Cloud Dataset for Urban Scene Segmentation and Classification.* CVPR Workshop on Real-World Challenges and New Benchmarks for Deep Learning in Robotic Vision, Salt Lake City, United States, Juin 2018

— [Deschaud 2018] Jean-Emmanuel Deschaud. *IMLS-SLAM: Scan-to-Model Matching Based on 3D Data.* 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 2480–2485, 2018

— [Bouchiba *et al.* 2018] Hassan Bouchiba, Jean-Emmanuel Deschaud et François Goulette. *Raw Point Cloud Deferred Shading through Screen Space Pyramidal Operators.* Proceedings of the 39th Annual European Association for Computer Graphics Conference: Short Papers, EG, page 25–28, Goslar, DEU, 2018. Eurographics Association

— [Thomas *et al.* 2018] Hugues Thomas, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette et Yann Le Gall. *Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods.* 2018 International Conference on 3D Vision (3DV), Verone, Italy, Septembre 2018. 3DV2018

— [Roynard *et al.* 2016] Xavier Roynard, Jean-Emmanuel Deschaud et François Goulette. *Fast and Robust Segmentation and Classification for Change Detection in Urban Point Clouds.* ISPRS 2016 - XXIII ISPRS Congress, Prague, Czech Republic, Juillet 2016. International Society for Photogrammetry and Remote Sensing

— [Nouira *et al.* 2016] Houssem Nouira, Jean-Emmanuel Deschaud et François Goulette. *Point Cloud Refinement with a Target-free Intrinsic Calibration of a Mobile Multi-beam Lidar System.* ISPRS congress 2016 International Society for Photogrammetry and Remote Sensing, Prague, Czech Republic, Juillet 2016

— [Nouira *et al.* 2015] Houssem Nouira, Jean-Emmanuel Deschaud et François Goulette. *Target-free Extrinsic Calibration of a Mobile Multi-beam Lidar System.* Laserscanning, Geospatial week 2015, La grande Motte, France, Septembre 2015

— [Craciun *et al.* 2014b] Daniela Craciun, Andrés Serna, Jean-Emmanuel Deschaud, Beatriz Marcotegui et François Goulette. *Scalable and Detail-Preserving Ground Surface Reconstruction from Large 3D Point Clouds Ac-*

quired by Mobile Mapping Systems. PCV (Photogrammetric Computer Vision), volume 3, pages 73 – 80, Zurich, Switzerland, Septembre 2014

— [Serna *et al.* 2014] Andrés Serna, Beatriz Marcotegui, François Goulette et Jean-Emmanuel Deschaud. *Paris-rue-Madame database: a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods.* 4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014, Angers, France, Mars 2014

— [Craciun *et al.* 2014a] Daniela Craciun, Jean-Emmanuel Deschaud et François Goulette. *Scalable And Detail-Preserving Ground Surface Reconstruction From Mobile Laser Systems For Driving Simulators Engines.* Driving Simulation Conference, Paris, France, Septembre 2014

— [Barczyk *et al.* 2014] Martin Barczyk, Silvère Bonnabel, Jean-Emmanuel Deschaud et François Goulette. *Experimental implementation of an Invariant Extended Kalman Filter-based scan matching SLAM.* 2014 American Control Conference, pages 4121–4126, 2014

— [Deschaud *et al.* 2012] Jean-Emmanuel Deschaud, David Prasser, M. Freddie Dias, Brett Browning et Peter Rander. *Automatic data driven vegetation modeling for lidar simulation.* 2012 IEEE International Conference on Robotics and Automation, pages 5030–5036, 2012

— [Tarel *et al.* 2012] Jean-Philippe Tarel, Pierre Charbonnier, François Goulette et Jean-Emmanuel Deschaud. *3D Road Environment Modeling Applied to Visibility Mapping: An Experimental Comparison.* 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, pages 19–26, 2012

— [Bonnabel *et al.* 2011] Silvère Bonnabel, Jean-Emmanuel Deschaud et Erwan Salaün. *A simple nonlinear filter for low-cost ground vehicle localization system.* 2011 50th IEEE Conference on Decision and Control and European Control Conference, pages 3270–3275, 2011

— [Deschaud & Goulette 2010b] Jean-Emmanuel Deschaud et François Goulette. *Point Cloud Non Local Denoising using Local Surface Descriptor Similarity.* PCV (Photogrammetric Computer Vision), Paris, France, Septembre 2010

— [Deschaud & Goulette 2010a] Jean-Emmanuel Deschaud et François Goulette. *A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing.* 3DPVT, Paris, France, Mai 2010

— [Brun *et al.* 2007] Xavier Brun, Jean-Emmanuel Deschaud et François Goulette. *On-the-way Mobile Mapping using Laser Range Scanner and Fisheye Camera.* MMT (Mobile Mapping Technology), Padua, Italy, Mai 2007

Bibliographie

[Aldoma *et al.* 2011] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu et G. Bradski. *CAD-model recognition and 6DOF pose estimation using 3D cues.* 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pages 585–592, 2011. (Cité en page 32.)

[Bai *et al.* 2020] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan et Chiew-Lan Tai. *D3Feat : Joint Learning of Dense Detection and Description of 3D Local Features.* arXiv e-prints, page arXiv :2003.03164, Mars 2020. (Cité en page 45.)

[Barber *et al.* 2008] David Barber, Jon Mills et Sarah Smith-Voysey. *Geometric validation of a ground-based mobile laser scanning system.* ISPRS Journal of Photogrammetry and Remote Sensing, vol. 63, no. 1, pages 128 – 141, 2008. Theme Issue : Terrestrial Laser Scanning. (Cité en page 24.)

[Barczyk *et al.* 2014] Martin Barczyk, Silvère Bonnabel, Jean-Emmanuel Deschaud et François Goulette. *Experimental implementation of an Invariant Extended Kalman Filter-based scan matching SLAM.* 2014 American Control Conference, pages 4121–4126, 2014. (Cité en page 87.)

[Barczyk *et al.* 2015] Martin Barczyk, Silvère Bonnabel, Jean-Emmanuel Deschaud et François Goulette. *Invariant EKF Design for Scan Matching-Aided Localization.* IEEE Transactions on Control Systems Technology, vol. 23, no. 6, pages 2440–2448, 2015. (Cité en page 85.)

[Bernardini *et al.* 1999] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva et G. Taubin. *The ball-pivoting algorithm for surface reconstruction.* IEEE Transactions on Visualization and Computer Graphics, vol. 5, no. 4, pages 349–359, 1999. (Cité en page 53.)

[Besl & McKay 1992] P. J. Besl et N. D. McKay. *A method for registration of 3-D shapes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pages 239–256, 1992. (Cité en page 25.)

[Bonnabel *et al.* 2011] Silvère Bonnabel, Jean-Emmanuel Deschaud et Erwan Salaün. *A simple nonlinear filter for low-cost ground vehicle localization system.* 2011 50th IEEE Conference on Decision and Control and European Control Conference, pages 3270–3275, 2011. (Cité en page 87.)

[Botsch *et al.* 2005] M. Botsch, A. Hornung, M. Zwicker et L. Kobbelt. *High-quality surface splatting on today's GPUs.* Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005., pages 17–141, 2005. (Cité en page 48.)

[Bouchiba *et al.* 2018] Hassan Bouchiba, Jean-Emmanuel Deschaud et François Goulette. *Raw Point Cloud Deferred Shading through Screen Space Pyramidal Operators.* Proceedings of the 39th Annual European Association for

Computer Graphics Conference : Short Papers, EG, page 25–28, Goslar, DEU, 2018. Eurographics Association. (Cité en pages 49 et 86.)

[Bouchiba *et al.* 2020] Hassan Bouchiba, Simon Santoso, Jean-Emmanuel Deschaud, Luisa Rocha-Da-Silva, François Goulette et Thierry Coupez. *Computational Fluid Dynamics on 3D Point Set Surfaces.* Journal of Computational Physics : X, page 100069, 2020. (Cité en pages 63 et 85.)

[Boulch & Marlet 2016] Alexandre Boulch et Renaud Marlet. *Deep Learning for Robust Normal Estimation in Unstructured Point Clouds.* Computer Graphics Forum, vol. 35, no. 5, pages 281–290, 2016. (Cité en page 36.)

[Boulch *et al.* 2017] A. Boulch, B. Le Saux et N. Audebert. *Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks.* Proceedings of the Workshop on 3D Object Retrieval, 3Dor17, page 17–24, Goslar, DEU, 2017. Eurographics Association. (Cité en pages 37, 38 et 41.)

[Browning *et al.* 2012] Brett Browning, Jean-Emmanuel Deschaud, David Prasser et Peter Rander. *3D Mapping for high-fidelity unmanned ground vehicle lidar simulation.* The International Journal of Robotics Research, vol. 31, no. 12, pages 1349–1376, 2012. (Cité en pages 59 et 85.)

[Brun *et al.* 2007] Xavier Brun, Jean-Emmanuel Deschaud et François Goulette. *On-the-way Mobile Mapping using Laser Range Scanner and Fisheye Camera.* MMT (Mobile Mapping Technology), Padua, Italy, Mai 2007. (Cité en pages 9 et 87.)

[Chan & Lichti 2013] Ting Chan et Derek Lichti. *Feature-based Self-Calibration of Velodyne HDL-32E LiDAR for Terrestrial Mobile Mapping Applications.* The 8th International Symposium on Mobile Mapping Technology, 05 2013. (Cité en page 15.)

[Cho *et al.* 2019] Younggun Cho, Giseop Kim et Ayoung Kim. *DeepLO : Geometry-Aware Deep LiDAR Odometry.* CoRR, vol. abs/1902.10562, 2019. (Cité en page 69.)

[Craciun *et al.* 2014a] Daniela Craciun, Jean-Emmanuel Deschaud et François Goulette. *Scalable And Detail-Preserving Ground Surface Reconstruction From Mobile Laser Systems For Driving Simulators Engines.* Driving Simulation Conference, Paris, France, Septembre 2014. (Cité en pages 53 et 87.)

[Craciun *et al.* 2014b] Daniela Craciun, Andrés Serna, Jean-Emmanuel Deschaud, Beatriz Marcotegui et François Goulette. *Scalable and Detail-Preserving Ground Surface Reconstruction from Large 3D Point Clouds Acquired by Mobile Mapping Systems.* PCV (Photogrammetric Computer Vision), volume 3, pages 73 – 80, Zurich, Switzerland, Septembre 2014. (Cité en pages 54 et 86.)

[Craciun *et al.* 2017] Daniela Craciun, Jean-Emmanuel Deschaud et François Goulette. *Automatic Ground Surface Reconstruction from mobile laser systems for driving simulation engines.* Journal of Simulation, vol. 93, no. 3, pages 201–211, 2017. (Cité en pages 54 et 85.)

[Curless & Levoy 1996] Brian Curless et Marc Levoy. *A Volumetric Method for Building Complex Models from Range Images.* Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH96, page 303–312, New York, NY, USA, 1996. Association for Computing Machinery. (Cité en page 24.)

[Demantké *et al.* 2011] Jérôme Demantké, Clément Mallet, Nicolas David et Bruno Vallet. *Dimensionality based scale selection in 3D lidar point clouds.* Proceedings of the ISPRS Workshop Laser Scanning, vol. 38, pages 97–102, 01 2011. (Cité en page 15.)

[Deschaud & Goulette 2010a] Jean-Emmanuel Deschaud et François Goulette. *A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing.* 3DPVT, Paris, France, Mai 2010. (Cité en pages 30, 31 et 87.)

[Deschaud & Goulette 2010b] Jean-Emmanuel Deschaud et François Goulette. *Point Cloud Non Local Denoising using Local Surface Descriptor Similarity.* PCV (Photogrammetric Computer Vision), Paris, France, Septembre 2010. (Cité en page 87.)

[Deschaud *et al.* 2010] Jean-Emmanuel Deschaud, Xavier Brun et François Goulette. *Colorisation et texturation temps réel d'environnements urbains par système mobile avec scanner laser et caméra fish-eye.* Revue Française de Photogrammétrie et de Télédétection, Septembre 2010. (Cité en pages 9 et 85.)

[Deschaud *et al.* 2012] Jean-Emmanuel Deschaud, David Prasser, M. Freddie Dias, Brett Browning et Peter Rander. *Automatic data driven vegetation modeling for lidar simulation.* 2012 IEEE International Conference on Robotics and Automation, pages 5030–5036, 2012. (Cité en pages 59 et 87.)

[Deschaud 2018] Jean-Emmanuel Deschaud. *IMLS-SLAM : Scan-to-Model Matching Based on 3D Data.* 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 2480–2485, 2018. (Cité en pages 24 et 86.)

[Duque-Arias *et al.* 2019] David Duque-Arias, Santiago Velasco-Forero, François Goulette, Jean-Emmanuel Deschaud et Beatriz Marcotegui. *A graph-based color lines model for image analysis.* International Conference on Image Analysis and Processing, Lecture Notes in Computer Science, Trento, Italy, Septembre 2019. (Cité en page 86.)

[Fischler & Bolles 1981] Martin A. Fischler et Robert C. Bolles. *Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.* Commun. ACM, vol. 24, no. 6, page 381–395, Juin 1981. (Cité en page 30.)

[Fruh & Zakhor 2001] C. Fruh et A. Zakhor. *3D model generation for cities using aerial photographs and ground level laser scans.* Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, volume 2, pages II–II, 2001. (Cité en page 8.)

[Gelfand *et al.* 2003] N. Gelfand, L. Ikemoto, S. Rusinkiewicz et M. Levoy. *Geometrically stable sampling for the ICP algorithm.* Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings., pages 260–267, 2003. (Cité en page 25.)

[Glennie & Lichti 2010] Craig Glennie et Derek Lichti. *Static Calibration and Analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning.* Remote Sensing, vol. 2, 06 2010. (Cité en page 15.)

[Golovinskiy *et al.* 2009] A. Golovinskiy, V. G. Kim et T. Funkhouser. *Shape-based recognition of 3D point clouds in urban environments.* 2009 IEEE 12th International Conference on Computer Vision, pages 2154–2161, 2009. (Cité en page 32.)

[Hackel *et al.* 2016] Timo Hackel, Jan Wegner et Konrad Schindler. *Fast semantic segmentation of 3d point clouds with strongly varying density.* ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. III-3, pages 177–184, 06 2016. (Cité en pages 35, 37 et 41.)

[Hackel *et al.* 2017] Timo Hackel, N. Savinov, L. Ladicky, Jan D. Wegner, K. Schindler et M. Pollefeys. *SEMANTIC3D.NET : A new large-scale point cloud classification benchmark.* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, volume IV-1-W1, pages 91–98, 2017. (Cité en page 36.)

[Horache *et al.* 2020] Sofiane Horache, François Goulette, Jean-Emmanuel Deschaud, Thierry Lejars et Katherine Gruel. *Automatic Clustering of Celtic Coins based on 3D Point Cloud Pattern Analysis.* ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. V-2-2020, pages 973–980, 2020. (Cité en page 85.)

[Hough 1962] Paul VC Hough. *Method and means for recognizing complex patterns*, Décembre 18 1962. US Patent 3,069,654. (Cité en page 30.)

[Huang *et al.* 2013] P. Huang, W. Hong, H. Chien et C. Chen. *Extrinsic calibration of a multi-beam LiDAR system with improved intrinsic laser parameters using v-shaped planes and infrared images.* IVMSP 2013, pages 1–4, 2013. (Cité en page 13.)

[Kazhdan & Hoppe 2013] Michael Kazhdan et Hugues Hoppe. *Screened Poisson Surface Reconstruction.* ACM Trans. Graph., vol. 32, no. 3, Juillet 2013. (Cité en pages 48 et 58.)

[Kümmerle *et al.* 2011] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige et W. Burgard. *G2o : A general framework for graph optimization.* 2011 IEEE International Conference on Robotics and Automation, pages 3607–3613, 2011. (Cité en page 29.)

[Kochanov *et al.* 2020] Deyvid Kochanov, Fatemeh Karimi Nejadasl et Olaf Booij. *KPRNet : Improving projection-based LiDAR semantic segmentation.* arXiv e-prints, page arXiv :2007.12668, Juillet 2020. (Cité en page 45.)

[Kolluri 2008] Ravikrishna Kolluri. *Provably Good Moving Least Squares.* ACM Trans. Algorithms, vol. 4, no. 2, Mai 2008. (Cité en page 63.)

[Kraus 2009] Martin Kraus. *The Pull-Push Algorithm Revisited.* Proceedings GRAPP 2009, 2009. (Cité en page 50.)

[Landrieu & Simonovsky 2018] L. Landrieu et M. Simonovsky. *Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs.* 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4558–4567, 2018. (Cité en pages 36, 37, 39 et 41.)

[Lawin *et al.* 2017] Felix Järemo Lawin, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, Fahad Shahbaz Khan et Michael Felsberg. *Deep Projective 3D Semantic Segmentation.* Computer Analysis of Images and Patterns, pages 95–107, Cham, 2017. Springer International Publishing. (Cité en pages 37 et 41.)

[Levinson & Thrun 2014] Jesse Levinson et Sebastian Thrun. Unsupervised calibration for multi-beam lasers, pages 179–193. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. (Cité en pages 14 et 17.)

[Li *et al.* 2018] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi et Leonidas Guibas. *Supervised Fitting of Geometric Primitives to 3D Point Clouds*, 2018. (Cité en page 36.)

[Liu *et al.* 2019] X. Liu, C. R. Qi et L. J. Guibas. *FlowNet3D : Learning Scene Flow in 3D Point Clouds.* 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 529–537, 2019. (Cité en page 36.)

[Lu *et al.* 2019] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan et Shiyu Song. *DeepICP : An End-to-End Deep Neural Network for 3D Point Cloud Registration.* arXiv e-prints, page arXiv :1905.04153, Mai 2019. (Cité en page 36.)

[Marton *et al.* 2009] Z. C. Marton, R. B. Rusu et M. Beetz. *On fast surface reconstruction methods for large and noisy point clouds.* 2009 IEEE International Conference on Robotics and Automation, pages 3218–3223, 2009. (Cité en page 58.)

[Marton *et al.* 2010] Z. Marton, D. Pangercic, R. B. Rusu, A. Holzbach et M. Beetz. *Hierarchical object geometric categorization and appearance classification for mobile manipulation.* 2010 10th IEEE-RAS International Conference on Humanoid Robots, pages 365–370, 2010. (Cité en page 32.)

[Maturana & Scherer 2015] D. Maturana et S. Scherer. *VoxNet : A 3D Convolutional Neural Network for real-time object recognition.* 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 922–928, 2015. (Cité en page 38.)

[Mescheder *et al.* 2018] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin et Andreas Geiger. *Occupancy Networks : Learning 3D Reconstruction in Function Space.* CoRR, vol. abs/1812.03828, 2018. (Cité en page 36.)

[Newcombe *et al.* 2011] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges et A. Fitzgibbon. *KinectFusion : Real-time dense surface mapping and tracking.* 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pages 127–136, 2011. (Cité en page 24.)

[Nouira *et al.* 2015] Houssem Nouira, Jean-Emmanuel Deschaud et François Goulette. *Target-free Extrinsic Calibration of a Mobile Multi-beam Lidar System.* Laserscanning, Geospatial week 2015, La grande Motte, France, Septembre 2015. (Cité en pages 14 et 86.)

[Nouira *et al.* 2016] Houssem Nouira, Jean-Emmanuel Deschaud et François Goulette. *Point Cloud Refinement with a Target-free Intrinsic Calibration of a Mobile Multi-beam Lidar System.* ISPRS congress 2016 International Society for Photogrammetry and Remote Sensing, Prague, Czech Republic, Juillet 2016. (Cité en pages 14 et 86.)

[Nouira *et al.* 2017] Houssem Nouira, Jean-Emmanuel Deschaud et François Goulette. *Point cloud refinement with self-calibration of a mobile multibeam lidar sensor.* The Photogrammetric Record, vol. 32, no. 159, pages 291–316, 2017. (Cité en pages 14 et 85.)

[Pintus *et al.* 2011] Ruggero Pintus, Enrico Gobbetti et Marco Agus. *Real-time Rendering of Massive Unstructured Raw Point Clouds using Screen-space Operators.* The 12th International Symposium on Virtual Reality, Archeology and Cultural Heritage VAST, pages 105–112, 2011. (Cité en pages 49, 51 et 52.)

[Qi *et al.* 2017a] Charles R. Qi, Li Yi, Hao Su et Leonidas J. Guibas. *PointNet++ : Deep Hierarchical Feature Learning on Point Sets in a Metric Space.* Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS17, page 5105–5114, 2017. (Cité en page 39.)

[Qi *et al.* 2017b] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo et Leonidas J. Guibas. *PointNet : Deep Learning on Point Sets for 3D Classification and Segmentation.* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 77–85, 2017. (Cité en page 39.)

[Qi *et al.* 2019] Charles R. Qi, Or Litany, Kaiming He et Leonidas J. Guibas. *Deep Hough Voting for 3D Object Detection in Point Clouds.* CoRR, vol. abs/1904.09664, 2019. (Cité en page 37.)

[Riegler *et al.* 2017] G. Riegler, A. O. Ulusoy et A. Geiger. *OctNet : Learning Deep 3D Representations at High Resolutions.* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6620–6629, 2017. (Cité en page 39.)

[Roynard *et al.* 2016] Xavier Roynard, Jean-Emmanuel Deschaud et François Goulette. *Fast and Robust Segmentation and Classification for Change Detection in Urban Point Clouds.* ISPRS 2016 - XXIII ISPRS Congress, Prague,

Czech Republic, Juillet 2016. International Society for Photogrammetry and Remote Sensing. (Cité en pages 32 et 86.)

[Roynard *et al.* 2018a] Xavier Roynard, Jean-Emmanuel Deschaud et François Goulette. *Classification of Point Cloud for Road Scene Understanding with Multiscale Voxel Deep Network.* 10th workshop on Planning, Perceptionand Navigation for Intelligent Vehicules PPNIV'2018, Madrid, Spain, Octobre 2018. (Cité en pages 39 et 86.)

[Roynard *et al.* 2018b] Xavier Roynard, Jean-Emmanuel Deschaud et François Goulette. *Paris-Lille-3D : A Point Cloud Dataset for Urban Scene Segmentation and Classification.* CVPR Workshop on Real-World Challenges and New Benchmarks for Deep Learning in Robotic Vision, Salt Lake City, United States, Juin 2018. (Cité en pages 32, 33 et 86.)

[Roynard *et al.* 2018c] Xavier Roynard, Jean-Emmanuel Deschaud et François Goulette. *Paris-Lille-3D : A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification.* The International Journal of Robotics Research, vol. 37, no. 6, pages 545–557, 2018. (Cité en pages 19 et 85.)

[Rusinkiewicz & Levoy 2001] S. Rusinkiewicz et M. Levoy. *Efficient variants of the ICP algorithm.* Proceedings Third International Conference on 3-D Digital Imaging and Modeling, pages 145–152, 2001. (Cité en page 25.)

[Schroeder *et al.* 1992] William J. Schroeder, Jonathan A. Zarge et William E. Lorensen. *Decimation of Triangle Meshes.* Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH92, page 65–70, New York, NY, USA, 1992. Association for Computing Machinery. (Cité en page 55.)

[Serna *et al.* 2014] Andrés Serna, Beatriz Marcotegui, François Goulette et Jean-Emmanuel Deschaud. *Paris-rue-Madame database : a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods.* 4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014, Angers, France, Mars 2014. (Cité en pages 18, 32, 33, 55 et 87.)

[Shewchuk 1996] Jonathan Richard Shewchuk. *Triangle : Engineering a 2D quality mesh generator and Delaunay triangulator.* Applied Computational Geometry Towards Geometric Engineering, pages 203–222, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg. (Cité en page 55.)

[Tarel *et al.* 2012] Jean-Philippe Tarel, Pierre Charbonnier, François Goulette et Jean-Emmanuel Deschaud. *3D Road Environment Modeling Applied to Visibility Mapping : An Experimental Comparison.* 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, pages 19–26, 2012. (Cité en pages 53 et 87.)

[Taubin *et al.* 1996] Gabriel Taubin, Tong Zhang et Gene Golub. *Optimal surface smoothing as filter design.* Computer Vision — ECCV '96, pages 283–292, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg. (Cité en page 55.)

[Tchapmi *et al.* 2017] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak et Silvio Savarese. *SEGCloud : Semantic Segmentation of 3D Point Clouds.* CoRR, vol. abs/1710.07563, 2017. (Cité en pages 37 et 41.)

[Thomas *et al.* 2018] Hugues Thomas, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette et Yann Le Gall. *Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods.* 2018 International Conference on 3D Vision (3DV), Verone, Italy, Septembre 2018. 3DV2018. (Cité en pages 35 et 86.)

[Thomas *et al.* 2019] Hugues Thomas, Charles Ruizhongtai Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette et Leonidas Guibas. *KPConv : Flexible and Deformable Convolution for Point Clouds.* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 6410–6419, 2019. (Cité en pages 39 et 85.)

[Wang *et al.* 2019] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein et Justin M. Solomon. *Dynamic Graph CNN for Learning on Point Clouds.* ACM Trans. Graph., vol. 38, no. 5, Octobre 2019. (Cité en page 39.)

[Wohlkinger & Vincze 2011] W. Wohlkinger et M. Vincze. *Ensemble of shape functions for 3D object classification.* 2011 IEEE International Conference on Robotics and Biomimetics, pages 2987–2992, 2011. (Cité en page 32.)

[Yi *et al.* 2017] Li Yi, Lin Shao, Manolis Savva, Haibin Huang, Yang Zhou, Qirui Wang, Benjamin Graham, Martin Engelcke, Roman Klokov, Victor Lempitsky, Yuan Gan, Pengyu Wang, Kun Liu, Fenggen Yu, Panpan Shui, Bingyang Hu, Yan Zhang, Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Minki Jeong, Jaehoon Choi, Changick Kim, Angom Geetchandra, Narasimha Murthy, Bhargava Ramu, Bharadwaj Manda, M Ramanathan, Gautam Kumar, P Preetham, Siddharth Srivastava, Swati Bhugra, Brejesh Lall, Christian Haene, Shubham Tulsiani, Jitendra Malik, Jared Lafer, Ramsey Jones, Siyuan Li, Jie Lu, Shi Jin, Jingyi Yu, Qixing Huang, Evangelos Kalogerakis, Silvio Savarese, Pat Hanrahan, Thomas Funkhouser, Hao Su et Leonidas Guibas. *Large-Scale 3D Shape Reconstruction and Segmentation from ShapeNet Core55.* arXiv e-prints, page arXiv :1710.06104, Octobre 2017. (Cité en page 37.)

[Zhang & Singh 2014] Ji Zhang et Sanjiv Singh. *LOAM : Lidar Odometry and Mapping in Real-time.* Robotics : Science and Systems, 2014. (Cité en pages 24, 25 et 27.)

[Zhao & Shibasaki 2003] Huijing Zhao et Ryosuke Shibasaki. *A vehicle-borne urban 3-D acquisition system using single-row laser range scanners.* IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication

of the IEEE Systems, Man, and Cybernetics Society, vol. 33 4, pages 658–66, 2003. (Cité en page 8.)

[Zhou & Tuzel 2018] Y. Zhou et O. Tuzel. *VoxelNet : End-to-End Learning for Point Cloud Based 3D Object Detection.* 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4490–4499, 2018. (Cité en page 39.)

[Öztireli *et al.* 2009] A. C. Öztireli, G. Guennebaud et M. Gross. *Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression.* Computer Graphics Forum, vol. 28, no. 2, pages 493–501, 2009. (Cité en pages 48 et 59.)

Nuage de Points et Modélisation 3D

Résumé : Ce mémoire d'HDR présente un panorama des travaux de recherche que j'ai menés et encadrés au sein de l'équipe Nuage de Points et Modélisation 3D du Centre de Robotique de MINES ParisTech.
Ces travaux portent tout d'abord sur les systèmes innovants de cartographie mobile basés LiDAR permettant la création de grands volumes de données 3D représentants l'environnement routier ou urbain traversés. A partir de ces données, des traitements automatisés sont nécessaires comme la détection de primitives géométriques, la segmentation sémantique, la reconstruction de surfaces pour arriver à une modélisation du réel. Je détaille dans ce manuscrit les contributions apportées à ces différentes problématiques avec des solutions les plus génériques possibles aux données d'entrée. Plusieurs applications concrètes ont été imaginées à partir de nuage de points d'environnements réels : le calcul de distance de visibilité sur route, la création d'environnements virtuels pour les simulateurs de conduite, la simulation de capteurs LiDARs à partir de données LiDAR et enfin la simulation d'écoulements de fluides autour de géométrie sous forme de nuage de points.

Mots clés : Nuage de points 3D, LiDAR, SLAM, segmentation, reconstruction

Point cloud and 3D Modeling

Abstract : This HDR report presents a panorama of the research work that I carried out and supervised within the Point Cloud and 3D Modeling team of the Robotics Center of MINES ParisTech.
This work relates first of all to innovative LiDAR-based mobile mapping systems allowing the creation of large volumes of 3D data representing the road or urban environment crossed. From this data, automated processing is necessary such as the detection of geometric primitives, semantic segmentation, surface reconstruction to arrive at a model of reality. I detail in this manuscript the contributions made to these different issues with the most generic solutions possible to the input data. Several concrete applications have been imagined from point clouds of real environments : the calculation of visibility distance on the road, the creation of virtual environments for driving simulators, the simulation of LiDARs sensors from LiDAR data and finally the simulation of fluid flows around geometry in the form of point clouds.

Keywords : 3D Point cloud, LiDAR, SLAM, segmentation, reconstruction