

THÈSE - THESIS

présentée à - defended in

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par - by

PIERRE BIASUTTI

POUR OBTENIR LE GRADE DE - TO GET THE DEGREE OF

DOCTEUR - DOCTOR

INFORMATIQUE - COMPUTER SCIENCE

2D Image Processing Applied to 3D LiDAR Point Clouds

Date de soutenance - defense date : 4 Octobre 2019

Devant la commission d'examen composée de - With a review board composed of :

Raphaëlle	CHAINED	Pr.	Université de Lyon	Rapportrice
Christian	HEIPKE	Pr.	Université de Hannover	Rapporteur
Mathieu	BRÉDIF	CR	Institut Géographique National	Encadrant
Pascal	DESBARATS	Pr.	Université de Bordeaux	Examinateur
Gabriele	FACCILOLO	Pr.	École Normale Supérieure, Paris-Saclay	Président du Jury
Jean-François	AUJOL	Pr.	Université de Bordeaux	Co-directeur de thèse
Aurélien	BUGEAU	MCF	Université de Bordeaux	Directrice de thèse
Jennifer	SIMEON	-	Geosat	Invitée

Titre

Traitement d'image 2D appliqué à des nuages de points LiDAR 3D

Résumé

L'intérêt toujours grandissant pour les données cartographiques fiables, notamment en milieu urbain, a motivé le développement de systèmes de cartographie mobiles terrestres. Ces systèmes sont conçus pour l'acquisition de données de très haute précision, telles que des nuages de points LiDAR 3D et des images optiques. La multitude de données, ainsi que leur diversité, rendent complexe le traitement des données issues de ce type de systèmes. Cette thèse se place dans le contexte du traitement de l'image appliqué au nuages de points LiDAR 3D issus de ce type de système.

Premièrement, nous nous intéressons à des images issues de la projection de nuages de points LiDAR dans des grilles de pixels 2D régulières. Ces projections créent généralement des images éparses, dans lesquelles l'information de certains pixels n'est pas connue. Nous proposons alors différentes méthodes pour des applications telles que la génération d'orthoimages haute résolution, l'imagerie RGB-D et l'estimation de la visibilité des points d'un nuage.

De plus, nous proposons d'exploiter la topologie d'acquisition des capteurs LiDAR pour produire des images de faible résolution: les range-images. Ces images offrent une représentation efficace et canonique du nuage de points, tout en étant directement accessibles à partir du nuage de points. Nous montrons comment ces images peuvent être utilisées pour simplifier, voire améliorer, des méthodes pour le recalage multi-modal, la segmentation, la désoccultation et la détection 3D.

Mots-clés

LiDAR, système de cartographie mobile, traitement d'image, orthoimage, visibilité, range-image, recalage, segmentation, désoccultation, détection 3D

Title

2D Image Processing Applied to 3D LiDAR Points Clouds

Abstract

The ever growing demand for reliable mapping data, especially in urban environments, has motivated the development of *close-range* Mobile Mapping Systems (MMS). These systems acquire high precision data, and in particular 3D LiDAR point clouds and optical images. The large amount of data, along with their diversity, make MMS data processing a very complex task. This thesis lies in the context of 2D image processing applied to 3D LiDAR point clouds acquired with MMS.

First, we focus on the projection of the LiDAR point clouds onto 2D pixel grids to create images. Such projections are often sparse because some pixels do not carry any information. We use these projections for different applications such as high resolution orthoimage generation, RGB-D imaging and visibility estimation in point clouds.

Moreover, we exploit the topology of LiDAR sensors in order to create low resolution images, named range-images. These images offer an efficient and canonical representation of the point cloud, while being directly accessible from the point cloud. We show how range-images can be used to simplify, and sometimes outperform, methods for multi-modal registration, segmentation, desocclusion and 3D detection.

Keywords

LiDAR, mobile mapping system, image processing, orthoimage, visibility, range-image, registration, segmentation, disocclusion, 3D detection

Remerciements

Je souhaite tout d’abord adresser ces remerciements à Raphaëlle Chainé et Christian Heipke pour avoir accepté de rapporter ma thèse, et pour leurs remarques avisées sur le manuscrit. Je remercie également Garbiele Facciolo et Pascal Desbarats pour avoir accepté de participer au jury de la soutenance de ma thèse.

J’ai eu la chance de mener ces travaux de thèse en collaboration avec le Laboratoire Bordelais de Recherche en Informatique, l’Institut de Mathématiques Bordelais et l’Institut National de l’Information Géographique et Forestière. Pendant 3 ans, j’ai été encadré avec attention par Aurélie Bugeau (LaBRI), Jean-François Aujol (IMB) et Mathieu Brédif (IGN). Grâce à eux, ces trois années ont été à la fois une incroyable expérience professionnelle ainsi qu’une formidable expérience humaine. Leur soutien indéfectible, leur disponibilité et leurs précieux conseils m’ont sans aucun doute permis de mener mes travaux à bien. L’exigence dont ils ont su faire preuve avec mesure, ainsi que leur passion pour la recherche, m’ont appris à sans cesse remettre en question mes travaux afin de les améliorer. Je souhaite donc leur exprimer ma plus grande gratitude pour leur considération à mon égard, pour m’avoir transmis ce goût pour la recherche de la meilleure des manières, et pour avoir fait preuve d’autant d’investissement à l’égard de mes travaux.

Je remercie aussi mes collègues du LaBRI, de l’IMB et de l’IGN avec qui j’ai eu l’occasion d’échanger, et avec lesquels je suis devenu amis pour certains (Giorgia, Michaël, Christelle, Kilian, Vincent, Florian, Michaël, Rohan, Philippe et tous les autres). Je remercie en particulier Rémi pour ses précieux conseils à la fin de la thèse, pour m’avoir aidé à préparer la soutenance dans son SPA, et pour ces deux conférences riches en émotions dont on ne retiendra qu’un mot : pédalo.

J’aimerais aussi adresser un remerciement spécial à mon ami Antoine, pour son soutien inaltérable depuis le début de nos études, il y a maintenant 8 ans. Ce manuscrit marque la fin de ces longues années d’études ensemble, en France ou à l’étranger, durant lesquels nous avons pu échanger sans cesse sur des sujets plus ou moins scientifiques. Je tiens aussi à le remercier, ainsi que Joanna, pour avoir eu l’idée brillante du Dé Caféiné, qui m’a permis de garder un semblant de vie sociale durant les derniers mois de la thèse.

Pour m’avoir soutenu pendant ces 3 ans, malgré mes indisponibilités chroniques, je remercie profondément tous mes amis : Véronique, Vincent, Alizé, Robin, Eva, François, Mathieu, Marine, Sophie, Melvin, Mélina, Marie, Thomas, Marine, Fabien, Martin. Merci de m’avoir toujours soutenu et d’avoir toujours été là dans les moments forts.

Enfin, je souhaite adresser un ultime remerciement à mes parents, à Maria et Julien, à Mamie et bien entendu à Guillemette pour leur soutien immuable, pour m’avoir supporté au quotidien, pour s’être intéressé à mes travaux et pour avoir toujours fait en sorte que cette expérience se passe le mieux possible.

À Mamina ...

Summary

Résumé en Français	3
General introduction	12
I Image processing on sparse projection of 3D LiDAR point clouds	20
1 Orthoimage generation from onground LiDAR acquisition	24
1.1 Introduction	25
1.2 Framework description	28
1.3 Projection of LiDAR point cloud	29
1.4 Diffusion of sparse images	32
1.5 Inpainting of occlusions	35
1.6 Results	39
1.7 Conclusion and future work	48
2 Dense depth map from sparse projection	49
2.1 Introduction	50
2.2 Model	53
2.3 Experimental results	57
2.4 Conclusion	61
3 Visibility estimation of a point cloud from a given point of view	62
3.1 Introduction	63
3.2 Related works	64
3.3 Visibility estimation method	66
3.4 Visibility estimation dataset for LiDAR point clouds	68
3.5 Experiments & Results	70
3.6 Conclusion	77
Part conclusion	78
II Image processing on 3D LiDAR point clouds in sensor	

topology	81
4 Dense 2D representation of a 3D LiDAR point cloud	86
4.1 Problem statement	87
4.2 Range-images derived from the sensor topology	87
4.3 Interest and applications	90
5 Point cloud to image registration	92
5.1 Introduction	93
5.2 Mutli-modal alignment	94
5.3 Methodology	97
5.4 Experiments and results	102
5.5 Conclusion	105
6 Object segmentation	106
6.1 Introduction	107
6.2 Point cloud segmentation	107
6.3 Proposed region segmentation method	110
6.4 Proposed semantic segmentation method	114
6.5 Experiments	117
6.6 Conclusion	119
7 Object removal	121
7.1 Problem statement	122
7.2 Object removal methods	122
7.3 Range-image disocclusion technique	124
7.4 Results & Analysis	127
7.5 Conclusion	133
8 Object detection	134
8.1 Introduction	135
8.2 2D detection architecture for 3D detection	136
8.3 Methodology	137
8.4 Results	141
8.5 Conclusion	143
Part conclusion	144
General conclusion and perspectives	149
1 General conclusion	149
2 Further works	150

A	Primal-dual algorithm for solving Equation (2.7)	154
1	Discrete setting and definitions	154
2	A primal-dual algorithm	156
B	Supplementary experiments of Chapter 2	161
1	Parameters of the algorithm and model choices	161
2	Results with urban data	162
3	Performance on visibility estimation	168
	Bibliography	171
	Table of contents	185

Acronyms list

The following table lists all acronyms that are used hereafter in the document.

CNN	Convolutional Neural Network
DSM	Digital Surface Model
DTM	Digital Terrain Model
GSD	Ground Sampling Distance
HPR	Hidden Point Removal
IoU	Intersection over Union
K-NN	K-Nearest Neighbors
LiDAR	Light Detection and Ranging
MAE	Mean Absolute Error
MMS	Mobile Mapping Systems
MSE	Mean Squared Error
NMSE	Normalized MSE
PnP	Perspective-n-Point
PSNR	Peak Signal to Noise Ratio
RGB-D	RGB Color + Depth image
SSD	Sum of Square Differences
SSIM	Structural SIMilarity
TIN	Triangular Irregular Networks
TV	Total Variation

Résumé en Français

Introduction

L'intensité historique et technologique du 20^{ème} siècle a été à l'origine de nombreuses améliorations des techniques de cartographies. Ces innovations ont principalement été motivées par un besoin toujours plus grandissant de données cartographiques précises pouvant répondre aux nouvelles attentes du secteur public, industriel et militaire, et qui ne pouvait pas être satisfaites par les techniques de cartographie manuelle pratiquées à cette époque. Le développement de nouveaux dispositifs d'acquisition pour la photographie, l'inférométrie et la mesure radar ainsi que la démocratisation de nouveaux moyens de transports ferroviaires, aériens et routiers, ont rendu possible la conception de systèmes de cartographie dit "mobiles". Grâce à la multitude de capteurs embarqués, ainsi que la possibilité de se mouvoir, ces systèmes de cartographie mobiles ont permis de pallier les limitations des techniques traditionnelles de cartographie, tant en terme de vitesse d'acquisition qu'en terme de diversité des données récoltées.

Ainsi, des systèmes aéroportés ont pu être utilisés pour photographier tout type territoire depuis le ciel. Les images acquises pouvaient ensuite être combinées pour produire des cartes texturées pouvant couvrir de très grandes superficies pour une durée d'acquisition très courte. Ces systèmes ont rapidement remplacé les techniques traditionnelles basées sur des relevés manuels pour de nombreuses applications, ces derniers étant imprécis et nécessitant des temps d'acquisition conséquents. De plus, les systèmes aéroportés ont ouvert la voie à de nouvelles applications comme l'étude du développement de l'urbanisation. De manière analogue, le développement de satellites d'observation permettant une acquisition plus distante des territoires a aussi rendu possible l'étude des comportements atmosphériques pour la météorologie. Les systèmes d'acquisition mobiles ont donc systématiquement accéléré des processus de cartographie fastidieux tout en offrant de nouvelles applications aux mesures effectuées.

La popularisation récente des smartphones, la prolifération d'applications mobiles exploitant la géolocalisation et la vision, ainsi que l'engouement de ces dernières années pour la conduite autonome, sont à l'origine de nouveaux besoins en terme de données cartographiques. De plus, la densification des populations dans les zones urbaines des pays développés pose de nombreux problèmes pour lesquels une information cartographique précise et fiable est nécessaire afin d'y apporter des solutions viables. Le développement de dispositifs de cartographie mobiles embarqués sur des véhicules terrestres a permis de répondre à ces besoins en offrant des descriptions plus précises des scènes urbaines par rapport aux systèmes plus anciens (aériens, satellites), grâce à une proximité accrue entre le système de cartographie et l'environnement ciblé.

Systèmes de cartographie mobiles terrestres

Les systèmes de cartographie mobiles terrestres ont été développés afin d'effectuer des acquisitions précises depuis le sol. Pour beaucoup d'applications, ces acquisitions doivent pouvoir être géo-référencées avec une précision centimétrique. De ce fait, ces systèmes disposent généralement d'un ensemble de capteurs pour la géolocalisation précise, tels qu'un GPS et une centrale inertielle. Ces capteurs permettent d'accéder à tout moment à la position et l'orientation du véhicule.

Afin d'obtenir la meilleure description de l'environnement urbain possible, les systèmes de cartographie mobiles sont équipés d'une suite de capteurs visuels. Plus précisément, ils disposent généralement de plusieurs caméras optiques de résolutions excédant un mégapixel. Ces caméras sont disposées de sorte à offrir une couverture panoramique complète autour du véhicule d'acquisition. Certains systèmes possèdent aussi plusieurs caméras orientées dans la même direction afin de pouvoir exploiter la redondance de l'information et/ou la vision stéréoscopique. Enfin, certains systèmes disposent aussi de capteurs multi-spectraux, notamment infrarouge, pour permettre des acquisitions dans des milieux faiblement éclairés.

Les systèmes de cartographie mobiles terrestres sont aussi conçus pour faire l'acquisition de l'information géo-spatiale en plus des données visuelles. Ainsi, des capteurs télémétriques, principalement LiDAR, sont aussi embarqués. Les capteurs LiDAR permettent de mesurer la distance entre le capteur et sa cible en mesurant la durée séparant l'émission d'un rayon laser et la réception de sa réflexion sur l'objet ciblé. Souvent, l'intensité de la réflexion est aussi mesurée. Cette information, appelée réflectance, peut servir à mesurer l'indice de réfraction de l'objet visé, permettant ainsi d'avoir une information de texture en plus de la mesure de distance. Les capteurs LiDAR modernes peuvent pivoter de sorte à permettre l'acquisition panoramique de plusieurs milliers de points par secondes, avec une précision millimétrique. L'information de distance ainsi que l'orientation du capteur permettent de déduire la coordonnées 3D de la mesure dans le repère du capteur, produisant ainsi un nuage de point 3D d'une grande précision.

La multitude de données, ainsi que leur diversité, rendent complexe le traitement des données issues de ce type de systèmes. Premièrement, le caractère hétérogène des données acquises par les différents capteurs empêchent une fusion multimodale directe. En effet, bien que ces données représentent toutes le même environnement, un nuage de points 3D et une image optique sont deux modalités exprimées dans des domaines très différents : l'image optique consiste en une matrice 2D à 3 canaux (RVB) de plusieurs milliers de pixels, alors que le capteur LiDAR produit un ensemble de points 3D. De plus, l'information géométrique offerte par le nuage de points LiDAR n'est pas organisée. De ce fait, il est difficile d'accéder au voisinage d'un point. Le traitement des données de ce type de systèmes représente donc un enjeu

majeur dans le contexte actuel.

On se concentre ici sur les données issues de systèmes de cartographie mobiles terrestres, et en particulier sur les nuages de points LiDAR. De ce fait, ne sont considérés que des jeux de données pour lesquels les trajectoires, les nuages de points LiDAR et les images optiques sont disponibles pour chaque scène. De plus, comme ces systèmes de cartographies sont des outils de pointe, nous considérons qu’une bonne approximation de la calibration de chaque système est connue a priori. Les travaux présentés dans cette thèse se concentrent sur trois jeux de données satisfaisant toutes ces caractéristiques: les données Stereopolis-II ([Paparoditis et al., 2012](#)), le jeu de données KITTI ([Geiger et al., 2012](#)) et le jeu de données Oxford RobotCar ([Maddern et al., 2017](#)).

Applications

Cette thèse se place dans le contexte du traitement de l’image appliqué aux nuages de points LiDAR pour différentes applications, détaillées ci-après. Pour cela, les nuages de points sont représentés sous la forme d’images obtenues selon deux procédés distincts. Dans la première partie de ce travail, on s’intéresse à des applications exploitant des nuages de points projetés dans des grilles de pixels régulières. Dans la seconde partie, la structure d’acquisition des capteurs LiDAR est exploitée pour produire des images.

Orthophotographie de haute résolution

Les orthophotographies sont des images aériennes prises d’un point de vue orthogonal au sol. Ces images sont le plus souvent acquises par des dispositifs aéroportés. Cependant, la distance entre le système d’acquisition et la zone ciblée limite grandement la résolution spatiale des acquisitions, typiquement 1m^2 par pixel, et justifie de la nécessité d’une acquisition rapprochée. On propose dans ([Biasutti et al.](#),



Figure 1: Comparaison entre une (a) orthophotographie aérienne classique de résolution $0.5\text{m}^2/\text{pixel}$ et (b) notre résultat terrestre de résolution $1\text{cm}^2/\text{pixel}$.

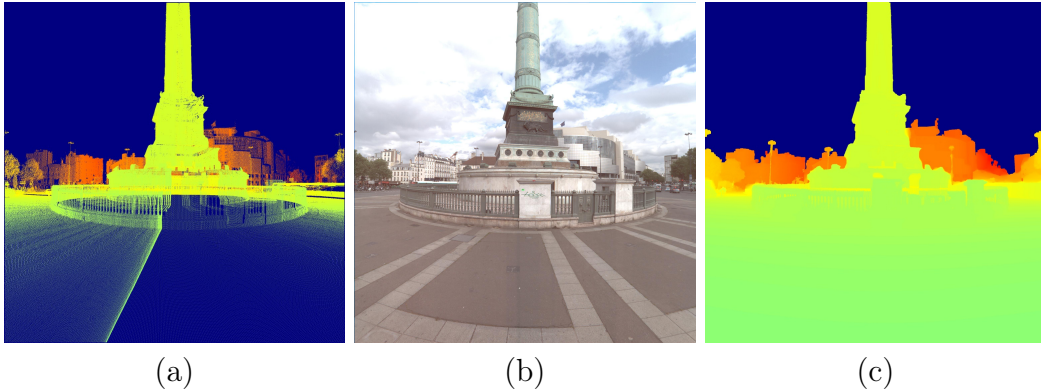


Figure 2: Exemple d’image RGB-D obtenue par notre méthode. (a) profondeur du nuage de points LiDAR projetée dans le domaine de l’image (b). (c) correspond à la profondeur densifiée.

2019a) une chaîne de traitement complète pour la production d’orthoimages de très haute résolution par projection du nuage de points LiDAR. Cette chaîne de traitement permet d’atteindre une résolution de 1cm^2 par pixel, bien supérieure à celles d’orthoimages issues de systèmes aériens, comme illustré Figure 1. De plus, notre méthode permet une reconstruction plus fine des détails de la scène comparée aux autres méthodes de l’état de l’art.

Imagerie RGB-D

Les images RGB-D (couleur et profondeur) sont particulièrement utiles dans de nombreuses applications liées à la vision. Ces images sont le plus souvent obtenues par la combinaison d’une caméra optique et d’un capteur de profondeur (souvent de résolution limitée), ou par stéréovision (dont les mesures ont une précision très limitée au-delà de quelques mètres). Pour pallier les limitations en terme de résolution et de précision des méthodes classiques, on propose une méthode variationnelle pour la production d’image RGB-D à partir d’une image optique et d’un nuage de points LiDAR projeté dans le domaine de cette image. Les résultats quantitatifs et qualitatifs de cette méthode, décrite dans (Bevilacqua et al., 2017), montrent que celle-ci améliore les méthodes existantes. Un exemple de résultat est proposé Figure 2.

Estimation de visibilité

La projection d’un nuage de points LiDAR dans le domaine d’une image optique produit le plus souvent une image éparse (*e.g.* une image dans laquelle l’information portée par certains pixels n’est pas connue, comme montré Figure 2(a)). Cette image éparse donne lieu à de nombreuses ambiguïtés. En effet, le nuage de points et



Figure 3: Colorisation d'un nuage de points sans estimation de visibilité (a) et avec estimation de visibilité (b).

l'image optique n'étant pas acquis depuis le même point de vue, certains éléments visibles dans une modalité ne le sont pas dans l'autre. On propose dans ([Biasutti et al., 2019d](#)) une méthode pour l'estimation de visibilité des points d'un nuage étant donné un point de vue. Contrairement aux autres méthodes de l'état de l'art, cette méthode ne requière pas que le nuage soit échantillonné de manière homogène. Elle est donc particulièrement adaptée à la donnée LiDAR. Nous proposons aussi un jeu de données manuellement annoté qui permet d'évaluer les performances de ces méthodes. L'analyse quantitative démontre que notre méthode donne de meilleures performances que l'état de l'art sur un nuage de points LiDAR, ce qui est confirmé visuellement Figure 3.

Nuage de points et topologie du capteur

Le traitement d'un nuage de points LiDAR est complexe du fait de l'absence de corrélation entre chaque point. La projection du nuage de points dans une grille de pixel 2D, ou la représentation sous la forme de voxels 3D, permettent de corréler

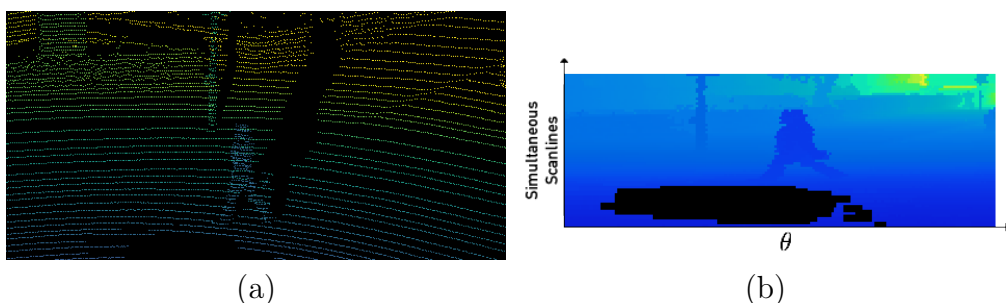


Figure 4: Nuage de points LiDAR (a) et sa représentation en topologie capteur (b).



Figure 5: Exemple d’alignement optique / LiDAR. (a) montre le décalage initial, (b) montre le résultat de l’alignement. Les tracés verts mettent en avant les contour du nuage de points projeté que l’on cherche à aligner.

spatialement les points, mais produisent des image éparses de très grandes dimensions. Les capteurs LiDAR modernes opèrent de manière structurée. Chaque point est acquis suivant un motif régulier. La structure d’acquisition, appelée topologie du capteur, peut ainsi être utilisée pour en dériver une image dense: la **range-image**. Cette range-image donne ainsi une représentation canonique du nuage de points dans laquelle chaque pixel est associé à un point suivant le motif d’acquisition du capteur. Ce principe est illustré Figure 4, et résumé dans (Biasutti et al., 2018). La range-image est utilisée dans les applications présentées ci-après.

Alignement optique et LIDAR

La fusion multimodale entre image optique et nuage de points LiDAR nécessite le plus souvent d’exprimer une modalité dans le domaine de l’autre; typiquement, en projetant le nuage de points dans le domaine de l’image grâce à la calibration du système. Bien que les systèmes de cartographie mobiles soient des systèmes très haut de gamme, la calibration de ces systèmes peut se dégrader au fur et à mesure d’une campagne d’acquisition. Il est donc souvent nécessaire d’aligner les deux modalités en post-production. A cette fin, nous proposons une chaine de traitement complète permettant d’aligner le rendu 3D d’un nuage de points avec une image optique grâce à une méthode variationnelle. Notre modèle, décrit dans (Biasutti et al., 2019c) et illustré Figure 5, permet de dépasser les limitations des précédentes méthodes.

Segmentation géométrique

La segmentation géométrique d’un nuage de point consiste à regrouper les points entre eux selon des critères géométriques. Cette tâche repose le plus souvent sur une

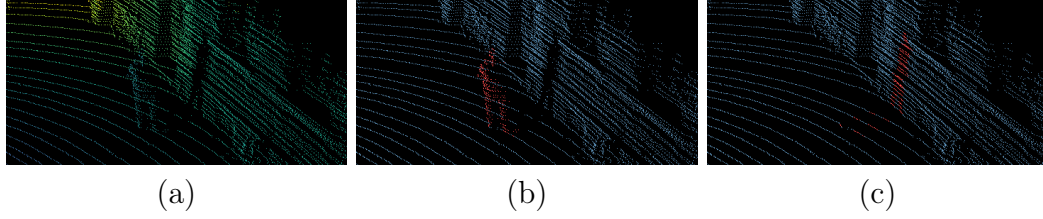


Figure 6: Exemple complet du processus de désoccultation d’un nuage de points LiDAR. (a) le nuage de points initial (coloré par la distance des points au capteur), (b) le nuage segmenté, (c) la reconstruction de la scène après retrait de l’objet segmenté.

estimation préalable du voisinage de chaque point, qui peut être fastidieuse lorsque le nombre de points est très grand ($> 10^6$ points). De plus, cette tâche nécessite parfois de connaître à priori le nombre de groupes à former, ce qui peut s’avérer impraticable dans des cas réels. Dans (Biasutti et al., 2018), nous introduisons un nouveau modèle de segmentation géométrique de nuage de points basé sur une segmentation d’histogramme de profondeur en topologie capteur. Tout en étant agnostique du nombre d’objets à segmenter, cette méthode permet une segmentation très fine de nuages de points de très grandes tailles, comme montré Figure 6(b). De plus, pour certains capteurs, cette segmentation peut être faite directement au fur et à mesure de l’acquisition.

Segmentation sémantique

Contrairement à la segmentation géométrique, la segmentation sémantique cherche à grouper les points en fonction du type d’objet auquel ils appartiennent. Depuis plusieurs années, les méthodes basées sur l’apprentissage profond atteignent des résultats acceptables. Néanmoins, la nature de la donnée LiDAR implique souvent d’utiliser des réseaux de neurones gourmands en mémoire et lents à l’exécution. D’autre part, cette tâche est au centre de beaucoup de travaux dans la communauté du traitement d’image pour la segmentation sémantique d’images. On propose ici une nouvelle méthode de segmentation sémantique de nuage de points directement

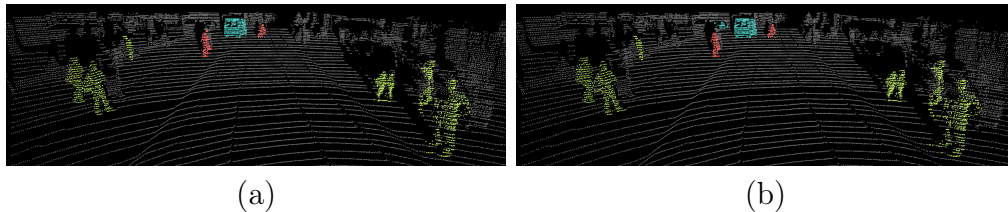


Figure 7: Résultat de la segmentation sémantique d’un nuage de points LiDAR (a), et la vérité terrain correspondante (b).

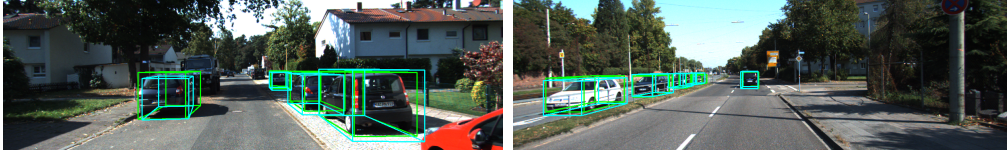


Figure 8: Résultats de la méthode de détection 3D.

adaptée d’une méthode de segmentation d’image basée sur des réseaux de neurones convolutifs. Cette méthode est décrite dans (Biasutti et al., 2019e). Nous montrons par cette approche comment la range-image peut faire le pont entre le traitement de données 3D et le traitement d’image. Les résultats, illustrés Figure 7, montrent qu’une architecture simple donne des résultats équivalents aux méthodes de l’état de l’art.

Désoccultation et reconstruction

Dans le but d’obtenir des fonds de carte 3D, il est souvent nécessaire de retirer des objets non-permanents de la scène (voitures, vélos, piétons). La segmentation permet de sélectionner ces objets. Néanmoins, le retrait des points associés à ces objets laisse des trous dans l’acquisition, ce qui rend celle-ci plus difficile à interpréter. Il est donc nécessaire de reconstruire la portion de nuage de points manquante. Dans (Biasutti et al., 2018), nous présentons une méthode pour la désoccultation de nuage de points. Cette méthode s’appuie sur une méthode d’inpainting la reconstruction plausible du nuage de points, tout en réduisant la dimension du problème à un problème d’estimation de profondeur. Les résultats montrent l’efficacité de la méthode, comme on peut le voir Figure 6(c).

Détection et localisation d’objets

Les véhicule autonomes requièrent généralement des systèmes de perceptions avancés. Plus particulièrement, la capacité de détecter et de localiser les objets environnants est cruciale pour ce genre de systèmes. On propose une chaine de traitement dans Biasutti et al. (2019b) pour la détection et la localisation d’objets en 3D à partir de réseaux de neurones convolutifs. Cette méthode, basée sur les range-images, permet d’effectuer la détection à très haute fréquence, ce qui est particulièrement adapté au contexte des systèmes embarqués. Deux exemples de résultats sont proposés sur la Figure 8.

General introduction

The historical and technological intensity of the 20th century have led to a significant improvement of mapping technologies. This improvement was largely motivated by the constantly increasing need for better mapping material that could suit the new requirements brought by the public, the industrial and the military domains. Such requirements could not be met by manual measurements in the sense that these techniques could only provide few geospatial information at the cost of very long acquisition campaigns, and they were therefore replaced by novel means of acquisition for many applications. In particular, the development of new remote sensing tools, such as optical cameras, radars or interferometers, in conjunction with the democratization of cars, trains and planes, have lead to the development of the first MMS. These systems typically allow the acquisition of geospatial data from a mobile vehicle using different sensors. Due to the variety of sensors mounted on such systems, as well as the ability to move, MMS have pushed back the limitations of traditional mapping techniques both in terms of acquisition speed and in terms of diversity of the acquired data.

For example, aerial photography has been extensively used to provide accurate textured acquisitions of lands and cities by using airborne optical cameras. The resulting images could then be manually combined to create textured maps that could cover large areas in reasonable times. Therefore, this technique quickly overcame the limitations of traditional methods – namely manual surveying – that often required several months, if not years, to cover a similar area, without being able to access specific terrains and without being able to acquire any details of the landscape. Such MMS could therefore be used to produce maps as well as for novel applications such as urban surveying. A similar case can be found for more distant, where the development of satellites did not only offer the ability to produce maps of wider areas from a farther point of view, but could also be used to gather information for weather forecasting.

Recently, the popularization of smartphones and the proliferation of mobile applications that take advantage of geolocalization and vision, as well as the ongrowing will of building autonomous systems such as autonomous cars, have motivated the development of MMS that could acquire data with a much finer precision (Vallet, 2016). To that extent, a new generation of MMS built on terrestrial vehicles that could navigate on roads and in an urban environment (*e.g.* cars, trucks) have been proposed. Such systems are also refered to as *close-range* MMS in constrast with flying systems that operate far from the target area. The proximity of the acquisition system with the observed environment enables the acquisition of much finer details, which can be exploited in numerous applications as detailed hereafter.

Close-range MMS

Close-range Mobile Mapping Systems were developed for accurate ground-based acquisition. These systems are often required to provide a centimetric georeferencing precision that cannot be reached using only GPS. To that end, they are equipped with GPS along with Inertial Navigation Systems (INS). The INS integrates measures from a system of sensors (odometer, inertial measurement unit (IMU), compass) to estimate the trajectory of the vehicle along 6 degrees of freedom – 3D translation, pitch, roll and yaw – at a very high frequency in order to compensate the low frequency of GPS measurements. Therefore, the accurate position and orientation of the acquisition system is available at any time with a guaranteed precision. Sometimes, a wheel odometer which estimates the distance traveled by the MMS by measuring the number of wheel turns is also added to the georeferencing components in order to increase the georeferencing accuracy. The combination of all of these sensors enables to reach a subcentimetric precision of the georeferencing.

The aim of *close-range* MMS is to acquire a complete information of the scanned environment. Therefore, most of modern *close-range* MMS are equipped with a suite of visual sensors, mostly optical cameras with a resolution exceeding one megapixel. These cameras are often arranged such that they cover the whole surroundings of the vehicle they are mounted on. Several cameras can also be oriented in the same direction to provide either redundant information or stereoscopic vision. Finally, some systems embed multi-spectral sensors, such as infrared sensors, to be able to operate without being sensitive to the scene illumination.

As these systems are also meant to acquire geospatial data, the visual sensors are used along with range-measurement sensors, namely LiDAR (Light Detection And Ranging) sensors. A LiDAR sensor is a time-of-flight sensor that measures its distance to a target. It operates by illuminating the target with pulsed laser light and by measuring return times of the reflected pulses to assess the distance. Recent sensors are able to rotate around an axis while acquiring hundred thousands of points each second, providing a representation of higher dimension (typically 2D or 3D). They are used to produce 3D point clouds with millimetric precision, which contributes in the global accuracy of *close-range* MMS. In the majority of LiDAR devices, the intensities of the reflected pulses are also measured. This measures how much of the laser light is back-scattered by the hit surface. Far from measuring the whole 4D Bidirectional Reflectance Distribution Function (BRDF) over all possible incoming and outgoing angles at the hit 3D point (and not considering any normal vector estimate), this single sample of the BRDF is still valuable. This measure produces a texture-like information, later referred to as *intensity*. To compensate for systematic biases due to the sensor and to the distance of the target, it is common practice to derive from this *intensity* a so-called *reflectance*, which is the albedo of a perfectly diffuse (Lambertian) front-facing target that would yield the same inten-

sity value when placed at the measured distance.

This thesis lies in the context of processing LiDAR data coming from *close-range* MMS. We only consider data in which trajectories, LiDAR point clouds and optical images are all available for each scenes. Moreover, as MMS are extremely high-end systems, we consider that a good approximation of the calibration settings is known a-priori. The works presented in this document rely on the data provided by:

- Stereopolis-II acquisitions ([Paparoditis et al., 2012](#))
- the KITTI vision suite ([Geiger et al., 2012](#))
- the Oxford RobotCar dataset ([Maddern et al., 2017](#))

whose basic specifications are summed up Figure 9. These systems were chosen as they all meet the requirements mentionned above, while proposing different combinations of sensor resolutions. Moreover, they provide benchmarks for certain deep-learning applications (*e.g.* 3D detection or semantic segmentation), which permits to compare our results to state-of-the-art methods. Hereafter in the thesis, the term MMS will relate to *close-range* MMS if not mentioned otherwise.

Processing MMS data is a very complex task for different reasons. First, the heterogeneous character of the data acquired by the different sensors prevents a direct fusion. Although they are representing complementary aspects of the same environment, a 3D point cloud and an optical image are two data that are expressed in two different domains. Indeed, the optical camera produces 2D images with 3 channels (RGB) of thousands, if not millions, of pixels whereas the LiDAR produces sets of millions of 3D points. Moreover, a point cloud only represents geometrical measurements of independant 3D points. Thus, accessing the neighborhood of a point, which is needed by many methods, is non-trivial and it often requires expensive preprocessings. On the other hand, processing 2D images provides spatial correlation that can be directly exploited in many ways (*e.g.* gradient computation or feature extraction).

Applications

Many applications benefit from MMS data, such as road and urban inventory, itinerary computation, mapping of road marks, road surface modeling and quality measurements, accessibility checking for soft mobilities, 3D city modeling, image based localization, object detection and segmentation for autonomous mobilities and high resolution orthoimaging. These applications use the multimodal aspect of the acquisition that MMS provide, and/or they rely on the precise georeferencing of the acquired data.







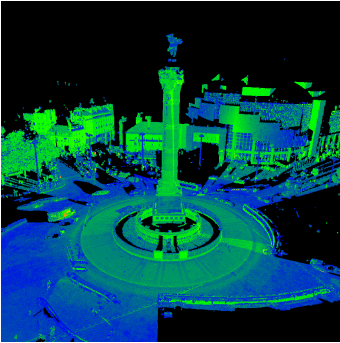
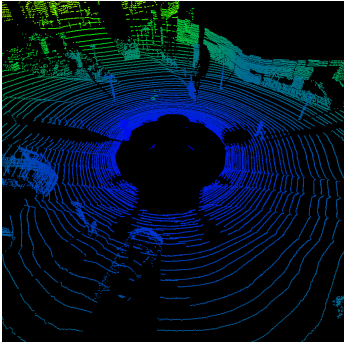
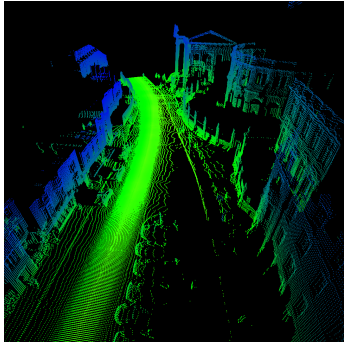
Acquisition systems		
Stereopolis-II (Paparoditis et al., 2012)	KITTI (Geiger et al., 2012)	Oxford RobotCar (Maddern et al., 2017)
		
Optical images		
		
$2048 \times 2048\text{px}$	$1242 \times 375\text{px}$	$1280 \times 960\text{px}$
LiDAR point clouds		
		
300000pts/s	1000000pts/s	27000pts/s

Figure 9: Overview of the 3 used MMS and associated data. The first row displays the acquisition systems, the second row shows a typical optical image that is acquired by each MMS as well as the corresponding resolution. The last row shows georeferenced LiDAR point clouds scene in 3D with the associated acquisition rate for each LiDAR sensor. Note that the colors of the point clouds are arbitrary set to improve understandability.

For example, the orthoimage production is the task of building a map from aerial images in which the perspective is corrected such that the final map corresponds to a vertical projection of the area on an horizontal plane. These products are usually obtained using airborne cameras or satellite imaging. However, due to the distance between the sensor and the ground, ground structures are likely to be occluded by overhanging objects such as bridges, canopies or tall buildings. Moreover, the resolution of the resulting maps is often limited to 5cm per pixel. Although such resolutions are sufficient to produce road maps, new applications require maps with finer resolution. Indeed, novel european legislations require that buried structures such as gaz pipes have to be registered on maps with a precision of less than 10 cm. As *close-range* MMS operate at ground level, they provide georeferenced LiDAR point clouds which are often of centimetric precision. These point clouds can therefore be used to produce orthoimages of higher resolution while respecting new regulations.

3D city modeling is another important application of MMS data as it can be used to conduct many types of surveys as well as being used in order to compute realistic itineraries for various kinds of mobilities. The task of city modeling relies on acquiring as many geometric information as possible of the urban environment, especially using LiDAR sensors. By exploiting the georeferencing information, all LiDAR point clouds are combined together into a single model. The aim is to gather precise 3D information of the permanent structures of the urban environment, such as facades, poles, traffic lights, signs or pavements. These information can be used to automatically perform surveys of the urban environment by using segmentation approaches. Unfortunately, *close-range* MMS have to operate in road traffic in spaces that are often crowded, as closing specific urban areas during the acquisition would be both expensive and unpracticable. Therefore, non-permanent entities such as cars, pedestrians or cyclists are also acquired by the LiDAR sensor. The 3D model does not only contain non-permanent entities, which is already an issue, but these entities prevent the sensor from acquiring the permanent structure that is located behind. To that end, being able to distinguish these entities and being able to remove them seamlessly from the scene is a major stake of MMS data processing for modeling applications.

Apart from solely exploiting the LiDAR data, the multimodal aspect of MMS acquisition also offers great applications. For example, city models can be colorized by projecting the color information of optical images on the LiDAR points. The colored model provides a material that is easier to understand for operators and it can also be used for augmented reality applications. However, the projection of the optical image colors on the LiDAR points strongly relies on the quality of the calibration. In realistic scenarios, the calibration settings are not perfect as MMS are subject to physical constraints (vibrations, shocks) during the acquisition that might alter the initial settings. This leads to a misalignment of the two modalities

which can be confusing in visualization tasks. The interest for automatic alignment between LiDAR data and optical data is therefore a crucial issue for multimodal fusion applications.

Finally, the development of autonomous systems, especially autonomous vehicles that are able to drive in cities, requires to create robust perception systems. The detection of objects in optical images is a now well known subject that has been at the center of the image processing community for the past decades. However, images only provide spatial information in their own 2D domain. Complex perception systems such as autonomous vehicles require much more complex perception systems that are able to efficiently localize and classify objects in their surroundings. This task, usually referred to as 3D detection, can largely take advantage of LiDAR data acquired by MMS as it provides precise geospatial information of the environment around the vehicle.

In this document, we propose to study how image processing methods can be applied to MMS data processing for various applications:

- orthoimage generation.
- LiDAR to image alignment.
- visibility estimation of a point cloud given a viewpoint.
- point cloud semantic segmentation.
- point cloud disocclusion.
- 3D object detection.

Content of the thesis

This thesis focuses on processing MMS data, especially LiDAR data, by operating on 2D pixel representations of the point clouds either by projecting it, or by exploiting the structure that is inherent to the sensor in order to generate a 2D image of the point cloud. Therefore, this document is divided in two parts.

The first part deals with the problem of processing 3D LiDAR point clouds that are being projected in 2D frames. These frames can either be the ground, considered as an horizontal plane, in the case of orthoimage generation (Chapter 1), or in the domain of an optical image for RGB-D imaging (Chapters 2 and 3).

In the second part of this thesis, we show in Chapter 4 how the intrinsic structure of the acquisition – named sensor topology – can be used to represent the point cloud as a 2D image that overcomes the limitations of a 2D projection (namely sparse projections and overlapping information). In particular, we show how it can be used for LiDAR point cloud to optical image alignment by proposing a

complete framework in Chapter 5. We also show how the sensor topology can be used for semantic segmentation using deep-learning techniques in Chapter 6 and for disocclusion using a PDE-based method in Chapter 7. Finally, we study the problem of 3D object detection by proposing a complete framework that takes advantage of convolutional neural networks in Chapter 8.

Part I

Image processing on sparse projection of 3D LiDAR point clouds

Summary

With the unceasing need of very precise acquisitions of the urban environment, either brought by new government regulations or by modern applications such as autonomous driving and 3D cartography, many traditional acquisition methods have shown their limitations. These limitations arise from different factors. For aerial imagery – and despite the presence of very high resolution imaging sensors – the distance between the sensors and the targeted urban scene limits the final resolution of the acquisition to several centimeters per pixel in the best cases. This type of acquisition is also affected by overhanging structures (*e.g.* buildings, tunnels, canopies) that might hide areas of the scene. For color and depth acquisitions, also known as RGB-D acquisitions, the available systems are often composed of a depth sensor of smaller resolutions than the associated optical sensors. This is often due to cost constraints, and it implies that we cannot access depth acquisitions at the same resolution as the one from optical images.

To overcome the limitations of classical acquisition systems, we investigate the use of MMS data, especially 3D LiDAR point clouds and optical images, to produce very high resolution products in the urban environment. Indeed, the projection of LiDAR point clouds in 2D-pixel grids or in image domains allows to use 2D image processing methods while preserving the precision of the measurements. It results in products that are easy to manipulate and understand especially for non-expert users.

A complete pipeline is first presented for the generation of very high resolution 2D orthoimages. This pipeline operates by projecting ground points on a 2D pixel grid, producing a sparse orthoimage. A densification step is then done and holes are inpainted to finally obtain a dense orthoimage at subcentimetric resolution.

We also investigate the generation of RGB-D data by projecting the LiDAR point cloud into the domain of an optical image. Again, this projection creates a sparse image, with visual ambiguities as the points and the optical image are not necessarily acquired from the same location. We propose a multi-modal variational model to densify this projection and to reduce visual ambiguities.

Finally, we explore the problem of visibility estimation of a point cloud given a point of view as it is a key issue for many tasks, such as multi-modal fusion or interactive visualization.

Content

- Chapter 1 presents the problem of the generation of high resolution orthoimages and the related works. It also presents a complete pipeline that enables subcentimetric orthoimage generation from MMS data.
- Chapter 2 presents the problem of high resolution RGB-D imaging and the related state-of-the-art. It then proposes to create high resolution RGB-D

images by fusing optical images and projected LiDAR data in a variational model. This fusion raises the problem of visibility estimation in a point cloud.

- Chapter 3 proposes to further investigate the problem of visibility estimation in a LiDAR point cloud given a point of view. Limitations of the state-of-the-art methods are shown in the case of point clouds with variable densities. A solution is proposed to leverage the variation of density that is inherent to the LiDAR point cloud in an urban environment.

Publications related to this part

Peer reviewed international journals (2)

- [1] *Diffusion and inpainting of reflectance and height LiDAR orthoimages*
Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Computer Vision and Image Understanding (CVIU), 2019, 179 (1), pp. 31–40
- [2] *Joint inpainting of depth and reflectance with visibility estimation*
Marco Bevilacqua, Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Int. Journal of Photogrammetry and Remote Sensing (IJPRS), 2017, 125 (1), pp. 16–32

Peer reviewed international conference proceedings (1)

- [3] *Visibility estimation in point clouds with variable density*
Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Int. Conference on Computer Vision Theory and App. (VISAPP), 2019, pp. 27–35

Peer reviewed national conference proceedings (2)

- [4] *Estimation de visibilité dans un nuage de point LiDAR*
Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Conférence Française de Photogrammétrie et de Télédétection (CFPT), 2018
- [5] *Diffusion anisotrope et inpainting d’orthophotographies LiDAR mobiles*
Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Reconnaissance des Formes, Image, Apprentissage (RFIA), 2016

Chapter 1

Orthoimage generation from onground LiDAR acquisition

Table of contents

1.1	Introduction	25
1.1.1	DSM generation from LiDAR data	26
1.1.2	Orthophotography from LiDAR data	27
1.2	Framework description	28
1.3	Projection of LiDAR point cloud	29
1.3.1	Filtering ground points	30
1.3.2	Sparse projections	31
1.3.3	Parameters	31
1.3.4	Dependency to the sensor	32
1.4	Diffusion of sparse images	32
1.4.1	Choice of the approach and requirements	32
1.4.2	Proposed algorithm	33
1.4.3	Comparison with other diffusion techniques	34
1.4.4	Parameters	35
1.5	Inpainting of occlusions	35
1.5.1	Occlusion hole detection	36
1.5.2	Exemplar-based inpainting	36
1.5.3	Modification to the original algorithm	37
1.5.4	Parameters	39
1.6	Results	39
1.6.1	Parameters	39
1.6.2	Qualitative analysis	40
1.6.3	Quantitative analysis	41
1.6.4	Computational speed	44
1.7	Conclusion and future work	48

1.1 Introduction

Orthophotographies and Digital Surface Models (DSM), defined respectively as the color and ground height orthoimages (*i.e.* raster maps defined on a regular horizontal grid), are ubiquitous products in modern cartography. They are widely used in many application fields such as remote sensing, geographical information and earth observation, mapping and environmental studies. Such orthoimages are traditionally computed from an aerial perspective (satellites, planes and more recently unmanned aerial vehicles (UAVs)). Although aerial imagery techniques provide a very well known and common approach to the problem of orthoimage generation, they may be limited in terms of accuracy and resolution and they certainly suffer from occlusions caused by the natural and urban environment such as trees, tunnels, overhangs or tall buildings (Fig. 1.1.a).

These limitations prevent orthoimages generated by above-ground datasets to be used for a whole new set of applications that rely on a precise mapping of the ground and which cannot suffer from such large occlusions. These applications include, mostly in an urban context, accessibility assessment for soft mobilities (disabled, wheelchairs and strollers) and itinerary computations (Serna and Marcotegui, 2013), precise mapping of road marks (Hervieu et al., 2015), road limits or curbs (Hervieu and Soheilian, 2013b), road inventory (Pu et al., 2011), road surface modelling and quality measurements (Hervieu and Soheilian, 2013a), mobile mapping registrations on aerial images (Tournaire et al., 2006) or image based localization using ground landmarks (Qu et al., 2015). Moreover, recent legislations in European countries call for a subdecimetric accuracy mapping of underground networks (water pipes, gaz pipes, internet wires and phone wires) as the lack of accurate data has lead to accidents and delays in many public works. Very high resolution orthoimaging and DSMs generation with limited occlusions could help in meeting the requirements of these legislations as it would provide sub-centimetric accuracy mapping of the ground.

To maximize orthoimage resolution and to minimize occlusions, we propose to

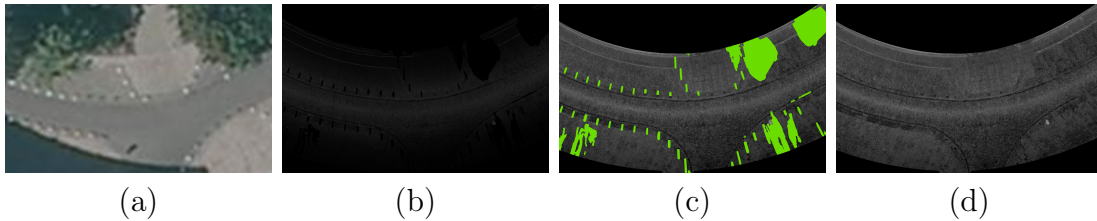


Figure 1.1: (a) Aerial orthoimage. (b) Rasterized LiDAR point cloud (reflectance attribute). (c) Interpolated LiDAR reflectance with estimated occlusion mask in green. (d) Proposed LiDAR orthoimage with inpainted reflectance. The aerial orthoimage has a Ground Sampling Distance (GSD) of $0.5\text{m}^2/\text{pixel}$ whereas the proposed result has a GSD of $1\text{cm}^2/\text{pixel}$.

leverage ground-level LiDAR scans acquired by MMS or from fixed stations. The proximity of the acquisition ensures a high resolution as well as a diminution of occluded areas. As in (Vallet and Papelard, 2015), we propose to derive a gray image from the reflectance attribute of the LiDAR samples (which measures backscattered energy) instead of relying on optical imagery (which would introduce difficulties in dynamic environments and require precise co-registration) to produce sub-centimetric orthoimages.

1.1.1 DSM generation from LiDAR data

The projection of a ground-level point cloud at centrimetric resolutions creates a sparse image due to its inhomogeneous sampling density (Figure 1.1 (b)). This problem is strongly related to DSM generation from LiDAR data, especially airborne LiDAR data which has been widely studied over the past decades as presented in (Chen et al., 2017b).

DSM generation DSMs are mostly represented either by Triangular Irregular Networks (TINs) (Zhang and Lin, 2013; Guan et al., 2014; Chen et al., 2016) or by raster images (Kraus and Pfeifer, 2001; Wack and Wimmer, 2002; Shan and Toth, 2008; Chen et al., 2012). A TIN is a mesh that represents a continuous surface entirely using triangles. In the case of airborne LiDAR TINs, the vertices of each triangle directly correspond to the LiDAR measurements. DSMs built from raster images consist in the projection of the LiDAR measurements on an horizontal grid, producing a sparse image in which some pixels do not contain any information. In both cases, the main challenges of DSM generation still remains twofold to create a higher level of representation. First the ground points have to be filtered to isolate ground information from the rest of the acquisition. Second, interpolation needs to be done to connect each LiDAR measurements.

Ground point filtering Ground point filtering aims at isolating points that belong to the ground (*e.g.* earth surface in case of airborne DSMs) from points that belong to elevated structures (such as buildings, trees, cars, fences or poles). This is especially useful for DSM generation as the final product aims at modeling only ground information of the area. For airborne LiDAR data, ground point filtering is done by defining slope operators in order to follow the ground surface (Zakšek and Pfeifer, 2006; Shao and Chen, 2008; Hu et al., 2015). These operators are used to estimate the relative slope of each LiDAR measurement given a set of neighbors. If the estimated slope is too steep, the surface is considered to locally correspond to an elevated structure that does not belong to the ground. However, these methods are developed in order to extract the ground on large scale, for terrain with high relief variation. In the case of urban scenarios, these methods fail to distinguish correctly small objects (such as bikes or pedestrians) from the ground.

Interpolation DSM interpolation approaches depend on the final product representation. For TINs, a common approach for the interpolation is by using Delaunay triangulation. Delaunay Triangulation connects a set of unorganized points with triangles such that the circumcircle of each triangle does not contain any other points. This property ensures to reduce the number of elongated triangles (*e.g.* triangles that have two very acute angles leading to a thin, elongated shape) and it produces a more homogeneous mesh. TINs interpolation is also done by plane fitting, as proposed in (Bitenc et al., 2011). In this case, the K-nearest neighbors (KNN) are computed for each LiDAR point. After that, a plane is fitted to each point given its set of K-neighboring points to produce a planar local representation of the surface. However, these approaches are not relevant to our problem as we aim at generating orthoimages as well as raster DSMs.

On the other hand, the interpolation of raster images DSMs has already been the object of several works. It has been proposed in (Kraus and Pfeifer, 2001) where the authors propose to interpolate the raster image by a coarse to fine approach. This approach uses raster image DSMs from low scale to final scale in order to estimate the interpolation. In Shan and Toth (2008), the authors propose to use moving least squares to perform the interpolation directly at final scale. The generation of DSMs does not require to preserve textures as the surface model is textureless. However in our context, we aim at generating orthoimages from the reflectance as well as DSMs from the height. Therefore, the preservation of texture is a key point of our problem, which requires to use other approaches for the raster image interpolation.

1.1.2 Orthophotography from LiDAR data

Although the problem of DSM generation from airborne LiDAR presents similarities with our problem, ground filtering cannot be done in the same way because of the fine scale of the objects in the scene. Moreover in our case, interpolation requires to preserve textures to produce orthophotographies. The method proposed in (Vallet and Papelard, 2015) offers a full pipeline for the production of both orthophotography and DSMs from MMS data. First, ground points are isolated by performing hard thresholding on the elevation of each point in the scene in order to produce a raw ground point estimation. The remaining points are then refined by considering the vertical cylindrical neighborhood of each point, in which only points that are close to the lowest point in terms of elevation are kept. The ground points are then projected on a 2D-grid in a similar way, as it is presented in Section 1.3. The pixels of the 2D-grid are either filled by the reflectance value or by the elevation value of the point that is projected there. Finally, the authors propose to use the Poisson interpolation (Pérez et al., 2003) to deal with the high variations of density in the raster images derived from MMS. They advocate that this method can be used for the interpolation of any modality of the point cloud while preserving texture information. This pipeline offers an efficient method for the production of orthophotography and DSMs from MMS data. However, the resulting orthopho-

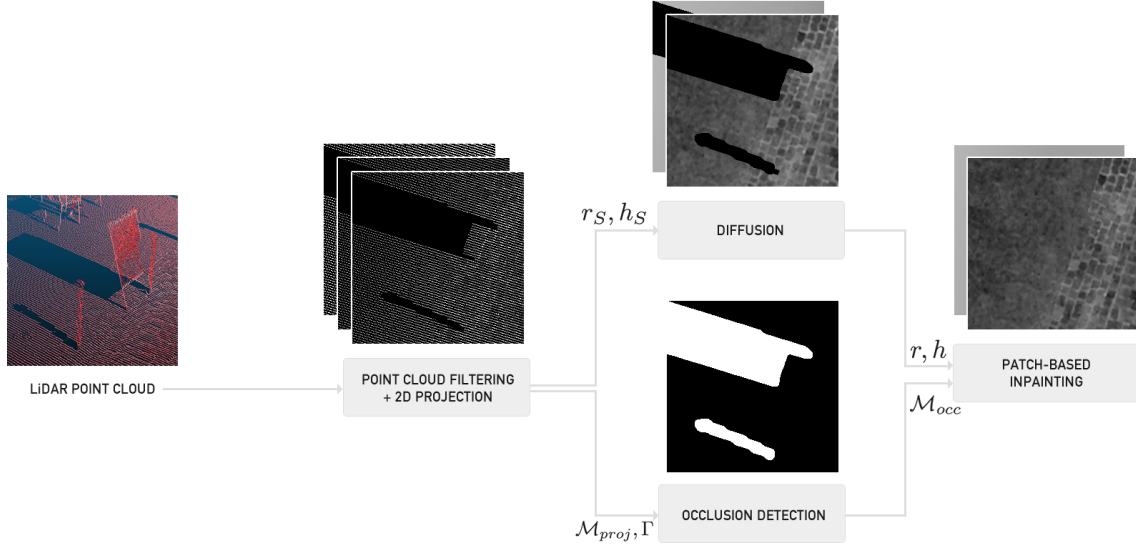


Figure 1.2: Full orthoimage production pipeline from MLS. Framed rectangles represent processing steps, arrows show exchanged data. h_0 and u_0 are the projections of each point height and reflectance respectively onto a horizontal grid. \mathcal{M}_{proj} is a binary mask of pixels where at least one point was projected.

tographies often lack of details and they show oversmoothed textures as explained in the experiments Section 1.6.

1.2 Framework description

Orthoimage generation from LiDAR scans acquired at ground level has been scarcely studied in the past. Nevertheless, the relation between LiDAR reflectance and optical acquisition has already been used for different applications such as depth map generation from point cloud (Bevilacqua et al., 2017), which shows the correlation between both modalities, and motivates the use of reflectance as alternative texture information of an orthoimage. This chapter introduces an efficient and fully automatic pipeline to reconstruct an orthoimage from a LiDAR point cloud. The proposed framework is summed up in Figure 1.2.

From the point cloud, we need to extract the ground points (*e.g.* points that do not belong to a mobile object or to an object that is lying on the ground). This is done by computing an envelop Γ (see section 1.3). The reflectance and height values of these ground points are then projected in two 2D-images respectively: r_s and h_s . This projection is done by removing the z (height) coordinate and rounding the coordinates to the chosen resolution. We also build a mask \mathcal{M}_{proj} of the pixels where at least one point was projected.

At this point, the projections u_0 and h_0 are often sparse as they do not cover

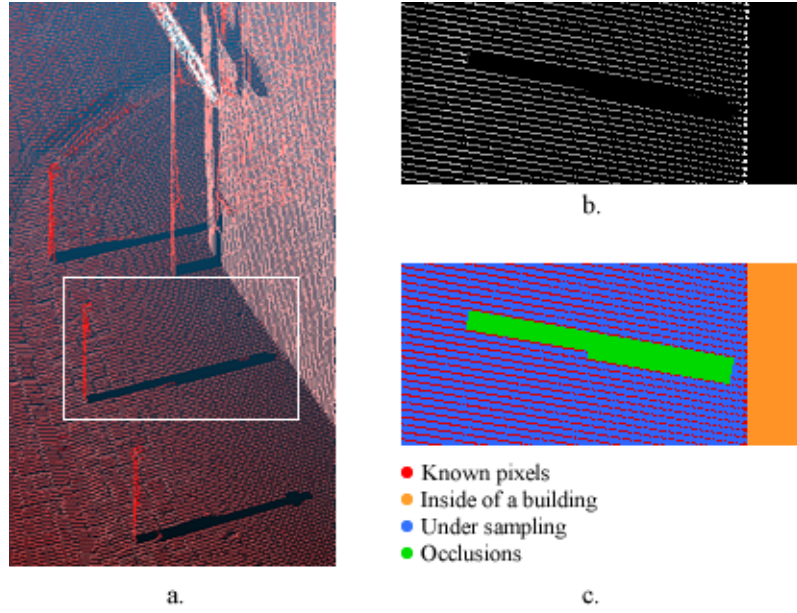


Figure 1.3: Highlighting of the different types of holes in the sparse projection. (a) is the original point cloud colored with the reflectance, (b) is the sparse projection of the white rectangle in (a) after extracting ground points, (c) is the sparse projection labelled with the different kinds of holes.

all the pixels of the images. Figure 1.3 presents an example of the different kinds of missing pixels that result from the projection. Some parts of the projection correspond to the inside of a building (Figure 1.3 (c) in orange), under sampling holes appear in between lines of acquisition (Figure 1.3 (c) in blue) and an occlusion is caused by a pole blocking the laser beams (Figure 1.3(c) in green). In order to reconstruct the missing information of the orthoimage, we first perform diffusion on both r_S and h_S by coupling reflectance and height in an anisotropic diffusion algorithm in order to remove holes due to undersampling. The resulting images are respectively called r and h . After this step, there are still some large holes remaining. Their locations are defined by the occlusion mask \mathcal{M}_{occ} , which is retrieved through mathematical morphology. Finally, we reconstruct occlusion holes using an exemplar-based inpainting method that uses both reflectance and height information, as well as an assumption about the alignment between structures to inpaint.

1.3 Projection of LiDAR point cloud

The projection of a point cloud onto a 2D pixel grid is a typical discretization problem. It mostly requires to define a mapping between the point cloud metric frame and the 2D-pixel grid. However, in the case of Digital Terrain Model, it is also

needed to filter out off-ground points (trees, urban structures, cars). We introduce a novel approach for ground point filtering in section 1.3.1 and we explain how the projections are done in section 1.3.2. More details about the parametrization of the projection can be found in 1.3.3.

1.3.1 Filtering ground points

The definition of ground-points in a point cloud can be tedious as we have to filter groups of point that represent relatively planar structures and which do not belong to any other objects than the ground itself. Ground filtering is a typical DSM generation problem (Meng et al., 2010). Traditional aerial DSMs generally model wide scenes of several square kilometers. In order to correctly include details of urban scenes (pavements, steps or any lightly elevated structure that belongs to the ground), it is necessary to model the ground at a finest scale. In urban scenario, plane fitting is often used as primary ground segmentation. Although it allows a fast and simple estimation of ground points, considering horizontal planes relative to the acquisition system can be ambiguous. Indeed, modern MLSs tend to be accurate enough to acquire ceilings through windows, creating false positives. Vertical planes are also relevant (pavements, stairs), but not in every cases (trucks, billboards). This problem has been investigated by considering it as a classification problem (Rottensteiner and Briese, 2002) or by performing advanced structural analysis (Kraus and Pfeifer, 2001; Brédif et al., 2015). However, these solutions have shown their limitations when the scene presents high diversity of objects. In particular, they lack of precision when aiming at estimating the boundaries of the ground in urban scenes because other objects (cars, ceilings) are often considered as the ground as they share common structural properties.

We propose a novel approach for ground point filtering based on the way the acquisition is done. We aim to filter out hovering object or any point that is above another one. As the points are acquired with a certain uncertainty, we cannot directly compare points coordinates as the likelihood of two points having the exact same (x, y) coordinates is negligible. We first create an empty envelop of the size of the projection where each pixel has an infinite value. This envelop will help defining the boundaries of the 2D region that represents the ground while ensuring that all points that fall into the envelop trully are ground points. We then consider segments made by each point and its relative LiDAR sensor location. Each segment is discretized in the envelop using the Bresenham line algorithm (Bresenham, 1965). As the beam is perfectly straight, we can estimate the height of the segment at any position of the segment. Each pixel is then filled with the lowest height value of segments that cross it. Note that in our case, only points below the sensor are considered. This reduces the amount of data to process while ensuring that none of the ground points are discarded. However, this is only suitable for MLS in urban scenarios. Figure 1.4 shows a slice of the maximal envelop Γ computed on a set of beams that overlaps. We can see that for every overlapping beams, only the portion

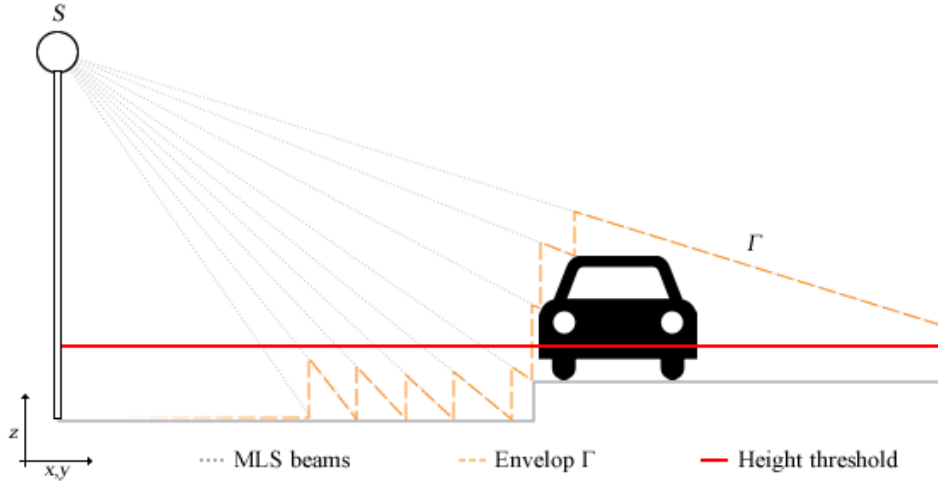


Figure 1.4: Slice of an envelop Γ obtained by evaluating several aligned beams coming from the sensor S until they hit an object of the scene. The red line is the final threshold applied to the envelop to exclude too high points.

of the lowest one is kept in the envelop. Finally, we filter the point cloud by taking only points that are under the envelop and the threshold, with an epsilon margin.

1.3.2 Sparse projections

Using the filtered point cloud, we want to produce two sparse images corresponding to the reflectances and the heights in the sensor frame: u_0 and h_0 defined on the mask \mathcal{M}_{proj} . The values for each pixel in u_0 is the mean of the reflectances of every points that is being projected in it. The values in h_0 are the same using the height in the sensor frame. Finally, a mask image \mathcal{M}_{proj} is produced where pixels are valued 1 where at least one point was projected and 0 elsewhere. Note that at high resolution (1px per square centimeter), the use of the mean is relevant on our data as only few points (less than 5) project in each pixel. However, if the amount of points that projects in each pixel increases a lot (when working at lower resolution for example), one can consider using the median instead of the mean to remove outliers. Note that the computational cost of the median is higher than the cost of the mean. Thus, its use will significantly increase the running time of the projection step.

1.3.3 Parameters

The choice of the mapping between real coordinates and pixels mostly depends on the density of the point cloud. In our case, with an acquisition done using a RIEGL LMS-Q120i which produces 300 000 points per second, an acceptable resolution was $1\text{px} = 1\text{cm}^2$. The height threshold is arbitrary but in the case of an urban scenario,

it should be kept under the height of the acquisition vehicle. More details about the parameters are provided in Section 1.6.1.

1.3.4 Dependency to the sensor

It is important to point out that the type of missing data are directly related to the chosen resolution as well as the type of sensor. The holes due to the acquisition sampling are less likely to appear when choosing a lower resolution. Moreover, the missing values in between acquisition lines are specific to the sensor mentioned above. They are quite homogeneous and create a regular pattern. With a panoramic sensor such as the one used in (Geiger et al., 2013), the missing pixels will appear in a random pattern, but will create a more dense image for the same resolution, which makes our pipeline still suitable for this type of data.

1.4 Diffusion of sparse images

The two images obtained in the previous section are sparse in the sense that they do not cover every pixels of the DSM. Therefore, we need to interpolate the images in order to get a dense representation of them. The goal is to fill in gaps between relatively close pixels that are due to the acquisition undersampling. In this section, we first explain what are the requirements that the filling method needs to meet. Then we introduce a modification to existing methods in order to enhance the results. Finally we show a comparison of different methods to validate our proposed modification.

1.4.1 Choice of the approach and requirements

A typical approach for filling small holes by interpolation is to use diffusion algorithms. Several diffusion techniques exist such as the total variation (Chambolle and Pock, 2011), the generalized total variation (Bredies et al., 2010), structure tensor diffusion (Weickert, 1998; Bertalmio et al., 2000) or partial differential equation diffusion (Aubert and Kornprobst, 2006) and extended to multi-modal data (Zhuang and Bioucas-Dias, 2018).

Here, we focus on iterative methods which are more flexible. A basic diffusion algorithm is the so called Gaussian diffusion which is an isotropic technique that consists in updating the image with its own Laplacian (Koenderink, 1984). However in the case of an urban scenario, an anisotropic diffusion is more relevant as very high gradients appear at the edge of different structures (roads, pavements, stairs) and it needs to be preserved.

The Perona-Malik algorithm (Perona and Malik, 1990) is a well known algorithm for anisotropic diffusion. It is partially inspired by the Gaussian diffusion and it is

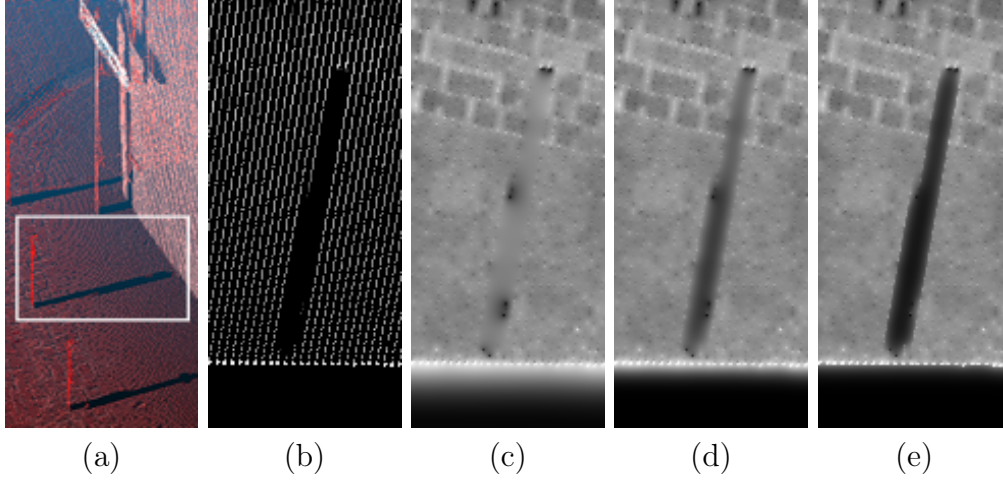


Figure 1.5: Comparison of different diffusion techniques for filling stripe holes. (a) is the point cloud, (b) its projection (rotated for clarity purpose), (c) is the Gaussian diffusion result, (d) is the Perona-Malik algorithm result and (e) is the result of our proposed modification. We can see that our modification provides a better conservation of big holes while filling perfectly the stripe holes.

defined as follows:

$$\begin{cases} \frac{\partial I}{\partial t} - \text{div}(c(|\nabla I|)\nabla I) = 0 & \text{in } \Omega \times (0, t) \\ \frac{\partial I}{\partial N} = 0 & \text{in } \partial\Omega \times (0, T) \\ I(0, x) = I_0(x) & \text{in } \Omega \end{cases} \quad (1.1)$$

where $I_0 \in \Omega$ is the input image, div is the divergence operator, ∇ is the gradient operator, N is the normal vector to the boundary of Ω and c is an increasing function. A common choice for c is the weighting function $c(|\nabla I|) = \frac{1}{\sqrt{1+(|\nabla I|/\alpha)^2}}$, α being a weighting factor that quantifies how much the gradient information needs to be considered. This technique ensures the preservation of edges while ensuring smooth transitions between sampled scan lines. Nevertheless, this technique only takes into account the gradients of a single channel. In our context, the diffusion needs to be blocked in case of a high gradient in the reflectance image as well as in the case of a high gradient in the height image that could correspond to the junction between the road and a pavement, or steps of stairs. Therefore, we need to modify equation (1.1) in order to take both channels into account.

1.4.2 Proposed algorithm

We propose here a modification to the Perona-Malik equation (1.1) by coupling heights and reflectances as follows, using previously introduced notations:

$$\begin{cases} \frac{\partial r}{\partial t} - \operatorname{div}(f(|\nabla r|, |\nabla h|)\nabla r) = 0 & \text{in } \Omega \times (0, t) \\ \frac{\partial h}{\partial t} - \operatorname{div}(f(|\nabla r|, |\nabla h|)\nabla h) = 0 & \text{in } \Omega \times (0, t) \\ \frac{\partial r}{\partial N} = 0 & \text{in } \partial\Omega \times (0, T) \\ \frac{\partial h}{\partial N} = 0 & \text{in } \partial\Omega \times (0, T) \\ r(0, x) = r_S(x) & \text{in } \Omega \\ h(0, x) = h_S(x) & \text{in } \Omega \end{cases} \quad (1.2)$$

where we recall that r_S is the reflectance image and h_S is the height image. We introduce the new weighting function f that emerges from the one used in equation (1.1) as follows:

$$f(|\nabla r|, |\nabla h|) = \frac{1}{\sqrt{1 + \frac{|\nabla r|^2}{\alpha^2} + \frac{|\nabla h|^2}{\beta^2}}} \quad (1.3)$$

having α, β as weighting constants quantifying how gradients of reflectance and height need to be considered. The choice of coupling both reflectance and height information into the same model is motivated by the fact that reflectance and height gradients are not always at the same locations and therefore, are complementary. Note that coupling various modalities in a model has already been proposed in (Auclair-Fortier and Ziou, 2006) for multi-spectral images, however in that case authors present a model specifically designed for merging multiple images representing the same object at different wavelengths. Using our method, we can now take into account gradients coming from both r_S and h_S .

1.4.3 Comparison with other diffusion techniques

In this section, we propose an evaluation of the performances of our model against Gaussian diffusion and closest neighbors diffusion. Projecting a point cloud acquired at very low speed provides a dense image locally. Therefore, we can define a ground truth using this region of the projection. We define a set of 20 masks of same dimension as the ground truth and we randomly set 80% of the pixels to 1. For each method and each mask, we recover pixels of the ground truth where the mask is valued 1, using the rest of the image. Note that the percentage of missing pixel (here, 80%) is defined as the average missing pixels ratio of our dataset. Finally, we compute the average of classical similarity metrics (MSSIM and MPSNR, which are respectively the mean of the SSIMs and the mean of the PSNRs) for each methods on the reconstructed images compared to the ground truth. The results are summed up in Table 1.1 in which we can see that our method outperforms the two other diffusion methods. Figure 1.6 presents one set of results. The Gaussian model as well as our model better succeed in recovering the aspect of the original image. Our method outperforms the Gaussian diffusion by recovering sharper edges.

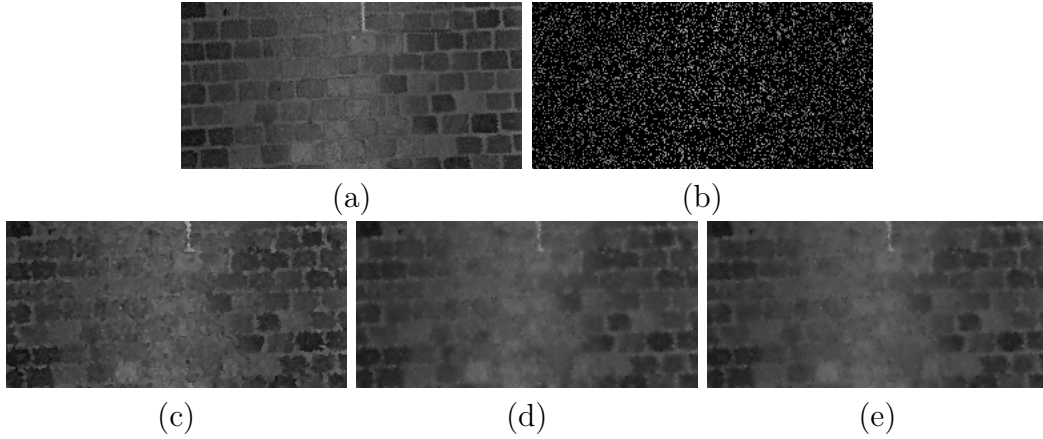


Figure 1.6: Result of the different diffusion models on degraded ground truth. (a) original image, (b) original image with 80% of pixels removed, (c) Closest Neighbors result, (d) Gaussian diffusion result and (e) Our result. We can see that the Gaussian diffusion and our model better recover the aspect of the image. Our method succeeds in a finer edge recovery.

Table 1.1: Evaluation of different diffusion algorithms

Metric	Closest Neighbors	Gaussian	Proposed Model
MSSIM	0.8056	0.8550	0.8591
MPSNR (dB)	33.21	34.59	35.08

1.4.4 Parameters

In practice, the proposed diffusion technique was implemented by solving the PDE system with a first order explicit Euler scheme with respect to the time variable. The number of iterations has to be chosen in order to fill in stripe holes. It depends on the chosen resolution as very sparse images will require more iterations to fully fill the image. Moreover, a good speed-up can be obtained by using the result of the closest neighbors diffusion of both u_0 and h_0 as the initialization for the proposed model as it drastically lowers the number of required iterations. The weighting term for the reflectances α should be higher than the one for the height β in order to completely block the diffusion in case of large height variation while connecting close pixels. Practical details will be given in section 1.6.1. Note that only unknown pixels in \mathcal{M}_{proj} should be updated to prevent oversmoothing the final images.

1.5 Inpainting of occlusions

After the projection, some holes are not only caused by some undersampling but also by the beam being blocked by an object (cars, poles, lights, pedestrians or

bikes) before reaching the ground. This leads to a ground projection with a lot of information at the edge turned toward the sensor, but nothing when going further. As occlusion holes are wider than stripe holes, the diffusion algorithm proposed above is not suitable in order to reach a visually satisfying result. In this section, we first see how occlusion holes are detected in the image. We then present the problem of texture synthesis in our case and we give a first solution. Finally, we introduce an improvement to this solution based on assumptions made on the urban scenario.

1.5.1 Occlusion hole detection

The occlusion detection consists in defining which holes are caused by the sampling rate and which holes are caused by a blocking of the laser beams. This can be done by applying mathematical morphology on the projection mask \mathcal{M}_{proj} before diffusion where each known pixel is valued 1 and all other pixels are valued 0. At this point, everything with the 0 value is considered as occlusion holes.

Having \mathcal{M}_{proj} , a simple morphological operation known as closing (Serra, 1982) is enough to detect occlusions and build the occlusion mask \mathcal{M}_{occ} . The closing consists in applying a dilation of a certain radius to the mask and then to apply an erosion of the same radius. This leads to a closing of small 0-labelled areas surrounded by ones. Choosing wisely the radius of the closing ensures that undersampling holes are eliminated while preserving the shape and the position of the occlusion holes.

Unfortunately, the resulting mask does not consider the boundaries of the scene, and it tends to extend further. We recall that when projecting the point cloud (Section 1.3), a Γ envelop is computed in order to define the boundaries of the scene. Thus, we consider the intersection of the computed mask and the Γ envelop to prevent the mask from expending outside of the ground region, typically inside of buildings or in regions too far from the sensor (Figure 1.3).

1.5.2 Exemplar-based inpainting

Among the variety of different inpainting algorithms, exemplar-based algorithms are known for being more effective and more reliable in filling large areas (with large internal radius). Exemplar-based inpainting consists in trying to find the best candidate in the known region of the image for the patch centered on a pixel lying on the border of the hole. Once found, the candidate is used to fill the unknown part of the image by copying the color in its central pixel (Efros and Leung, 1999) or the full patch (Criminisi et al., 2004). The operation is repeated until the hole is fully closed. More recent approaches, such as (Daribo and Pesquet-Popescu, 2010) or (Wang et al., 2008) reconstruct the texture using both color information and depth information. However these algorithms require different acquisitions of the same view, which is not applicable in our case as we aim at performing the reconstruction on a single acquisition pass.

The urban scenario presents a huge variety of structures (roads, pavements,

stairs, gutters) as well as many different textures (roads, cobbles, floor tiles). Thus, we decided to base our work on the Criminisi et al. (Criminisi et al., 2004) algorithm that was designed for the good preservation of the structures in the reconstruction. More complex approaches exist that rely on the work presented in Criminisi et al. (2004) such as (Buysens et al., 2015b) and (Lorenzi et al., 2011) however it would have been less intuitive to adapt them to our context. In (Criminisi et al., 2004), authors put forward the idea that the order in which areas are reconstructed have a high impact in the final result. They introduce a priority term that takes into account the strength and the direction of the image’s gradient at the border of the unfilled area. A patch that contains a strong gradient in the direction orthogonal to the border of the region to reconstruct is evaluated before more uniform patches.

1.5.3 Modification to the original algorithm

Coupling reflectances and heights The algorithm presented in (Criminisi et al., 2004) offers a very good technique for region filling. However, it can fail when the area to fill is very large. Therefore, we introduce a modification to the algorithm by taking the height information into account as a guide for the reconstruction. The idea is to use the height information to restrain the selection of best candidate patches to the areas of similar height by computing the Sum of Squared Differences of the candidate patch in both the reflectance and the height images. The Sum of Squared Differences (SSD) is defined as follows:

$$\text{SSD}(P_1, P_2) = \sum_{i,j \in \Omega} (P_1(i, j) - P_2(i, j))^2 \quad (1.4)$$

having P_1, P_2 the two 2D-patches that are compared and Ω the domain of definition of the image. In our modification, and for each candidate, a score is attributed by combining both channels as follows:

$$S_p(P_t, P_c) = \text{SSD}(P_t^R, P_c^R) + \eta \times \text{SSD}(P_t^H, P_c^H) \quad (1.5)$$

where P_t is the target patch to be filled and P_c is a candidate patch. P_c can be any patch in the image that has no pixel that belongs to an occlusion hole. However, for speed-up purpose, we can limit the selection of P_c to be in a certain radius around P_t . η is a regularization parameter and the superscripts R, H denote that the patch is taken in the reflectance image or the height map respectively.

The impact of the use of the height map in the synthesis is very noticeable in Figure 1.7. The structure of the road is well preserved using the proposed modification compared to the original algorithm in which artifacts appear after some iterations. These artifacts mislead the reconstruction and the result is visually incoherent.

Taking advantage of urban environment Although the current modification of the algorithm provides a very good solution for filling occlusion holes, the reconstruction can fail sometimes when the hole is very large. This happens for holes

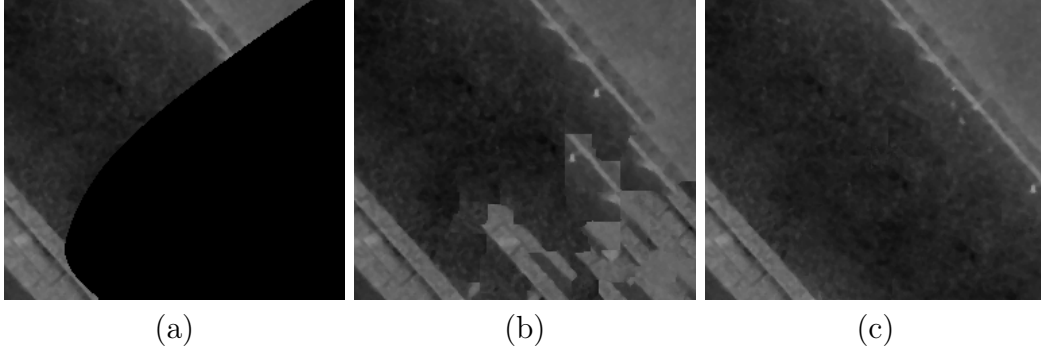


Figure 1.7: Comparison of (Criminisi et al., 2004) and our proposed modification on the junction between the road and a pavement. (a) is the original unfilled image where the dark region is being reconstructed using exemplar-based inpainting, (b) the result from Criminisi et al. (2004), (c) our proposed optimization. The result is clearly better in (c) as the reconstruction conserves the structures of the image without creating new artifacts such as the one appearing on the left of (b).

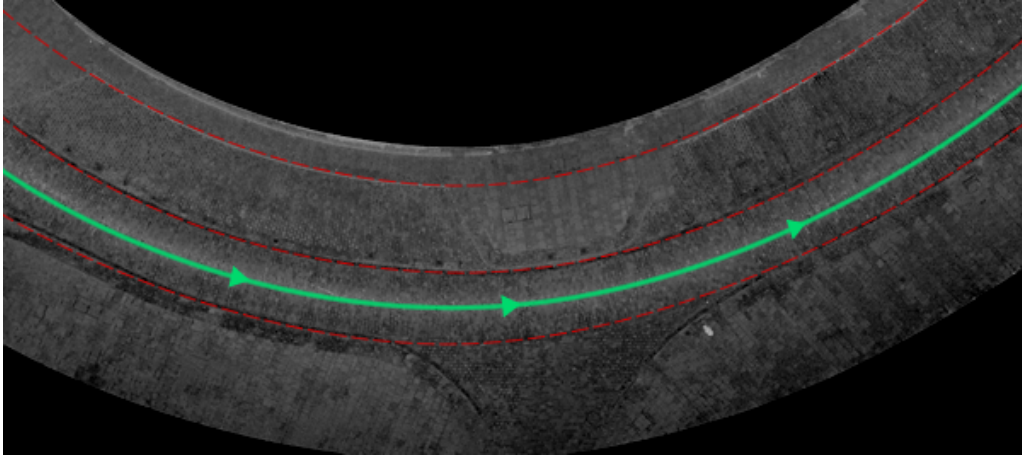


Figure 1.8: Illustration of the assumption that the urban environment evolves in a similar way than the path of the sensor. The straight green line shows the path of the sensor. Each dashed line represents areas of same distance to the sensor.

that are caused by cars or trucks where the area to reconstruct is significantly larger than regular holes (10^6 pixels at a $1\text{px} = 1\text{cm}^2$ resolution for a standard car and the portion of pavement behind it) and it can become a common issue. Indeed, at the center of the holes the nearest known information is too far away and the error accumulated along the iterations is likely to fail the reconstruction. To improve the results in the concerned areas, we advocate that the structure of a urban environment is very likely to evolve in a similar way to the vehicle path as illustrated in Figure 1.8. Therefore, we can constrain the selection of candidates to patches that are at a similar distance to the sensor than the current patch. The range attribute of the LiDAR image provides this information for each point.

We define the new score equation as follows, using previously introduced notations:

$$S_f(P_t, P_c) = \left[1 + \left(\frac{|\text{dist}(P_t) - \text{dist}(P_c)|}{\gamma} \right)^2 \right] \times S_p(P_t, P_c) \quad (1.6)$$

having $\text{dist}(P)$ the distance between the sensor and the center of the patch P and γ a regularization parameter that constrains the selection of patch to a range interval around the current range. The range can be accessed everywhere in the image by precomputing a signed distance map of the area to the path of the vehicle (*e.g.* where the range is the lowest).

Large patches and artifacts When the reconstruction is done at a very high resolution, large patches (10^3px) are likely to be required in order to correctly represent the structural elements of the image. This might lead to abrupt junctions between reconstructed patches. Therefore, we propose to enhance the copy of the patch by performing the seam carving using graphcuts presented in [Rubinstein et al. \(2008\)](#). The goal is to compute the optimal cut between P_t and P_c where they overlap to obtain a seamless result.

1.5.4 Parameters

η should be kept under 1 to ensure the visual coherence of the reconstruction. Parameter γ depends on the size of the occlusion. When $\gamma = 1$, the regularization is very strong and the selection of the candidate patch is constrained on a narrow band of same distance to the sensor point. When the value of the parameter is highly increased ($\gamma > 10^4$), no regularization operates and the algorithm behaves as if the range was not taken into account. Therefore, one can alternate between these two values for γ depending on the internal radius of the occlusion (see next section).

1.6 Results

We conclude this chapter by presenting different results obtained using the proposed framework. We first present a general set of parameters for an automatic reconstruction of a set of orthoimages. We then demonstrate the efficiency of the solution by showing various results and comparison to existing methods. After that, we validate the quality of the framework using numerical criterions. Finally, some details about the computation time are drawn.

1.6.1 Parameters

In the same way as other pipelines, this one comes with a set of parameters that was used for producing every images displayed in this chapter.

Projection The objective of this study was to provide very high quality orthoimages. Therefore, all reconstructions were done with a resolution of $1\text{px} = 1\text{cm}^2$. A threshold of 60cm from the road level was used to filter out points after the computation of the envelop.

Diffusion For the diffusion step, we found the best balance of results by setting $\alpha = 5, \beta = 0.7$ with 3 iterations and by first interpolating u_0 and h_0 using the nearest neighbor algorithm.

Mask extraction In this step, a closing radius of 6px was enough to fill stripe holes while leaving occlusions intact.

Inpainting At $1\text{px} = 1\text{cm}^2$, the chosen patch size was 43x43px to fit the smallest structuring element (cobble). In all our experiment, $\eta = 0.2$ ended up being a very good choice. Finally, we set the value of γ to 0.3 or 10^6 , the choice being made by automatically checking whether the internal radius of the evaluated occlusion was higher than 50cm or not.

1.6.2 Qualitative analysis

A quick glance at the difference between traditional aerial orthophotography and MLS orthoimage using our framework is given in Figure 1.9. The resolution provided by a typical aerial camera is between 5cm^2 and 10cm^2 per pixel (while advanced acquisition systems have reached up to 3cm^2 per pixel), where our reconstruction is done at 1cm^2 per pixel. Fine textures and very precise details are noticeable in the reconstruction whereas only main structures can be seen in the aerial orthophotography. Moreover, the aerial orthophotography presents various occlusions such as trees that do not appear in our result.

In Figure 1.10, we show a visual comparison between the proposed framework and the method introduced in (Vallet and Papelard, 2015). We can see that both algorithms perform about the same for stripe holes, but our solution gives more satisfying results for large occlusions. The texture is better reconstructed using our method. This will be later discussed in Section 1.6.3.

More reconstruction results are displayed in Figure 1.11. Each step of the pipeline is illustrated. We can see on Figure 1.11 top that the framework performs a very good reconstruction on fine details such as cobbles. In Figure 1.11 bottom, 25% ($\sim 5.10^5\text{px}$) of the area is occluded, mostly due to the presence of cars and poles. However, our framework delivers a plausible reconstruction of the scene, leading to a result that is much more understandable than without any further processing than projection. Finally, Figure 1.12 shows an extreme scenario where the use of the range is relevant as the structure of the scene follows the same path as the road. The environment is fully reconstructed (16%, $\sim 10^6\text{px}$) while preserving the structure of the road.



Figure 1.9: Comparison between aerial orthophotography with a standard resolution (10cm^2 per pixel) (top) and MLS orthoimage using our model at 1cm^2 per pixel (bottom). Traditional orthophotography provides limited resolution and suffers from occlusions brought by the coverage of trees and other structures whereas our model provides unobstructed, high resolution orthoimages. The aerial image comes from Geoportal.

In Figure 1.13, the framework is applied on data provided by the Semantic3D dataset (Hackel et al., 2017). This dataset is acquired using a static LiDAR sensor. There, we can see that the area under the sensor as well as occlusion on the ground are successfully recovered while preserving the fine cobble texture.

The purpose of this pipeline is to generate both reflectance and height orthoimages. In Figure 1.14, we show how the two outputs can be combined in order to obtain a 3D model of the road. Figure 1.14 (a) and 1.14 (b) are the reflectance and height images of the area that is being modelled in Figure 1.14 (c). The 3D model respects the topography of the scene with the junction of the road and a pavement.

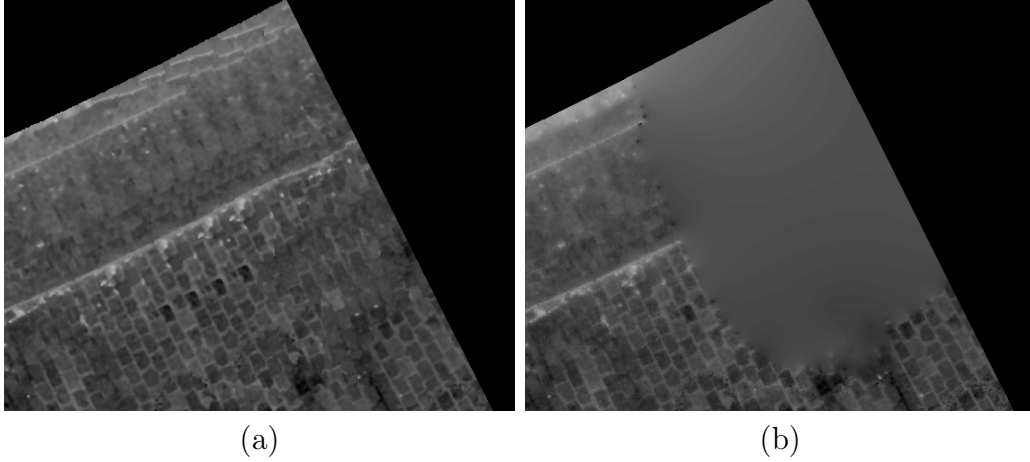


Figure 1.10: Comparison between our proposed framework (a) and the one introduced in (Vallet and Papelard, 2015) (b). Texture is better preserved using our framework.

Table 1.2: Numerical comparison between reconstructions

Image	Artificial occlusion		Real occlusion	
	<i>STD</i>	<i>Hist. dist.</i>	<i>STD</i>	<i>Hist. dist.</i>
Ground truth	4.51	-	4.79	-
Proposed framework	4.56	0.14	4.29	0.19
Vallet and Papelard (2015)	1.87	0.78	2.05	0.80

1.6.3 Quantitative analysis

Apart from the visual results, we also provide a numerical comparison between the proposed framework and the one of Vallet and Papelard (2015). Measuring similarities between two images is a tough task as the plethora of different metrics are all designed for a single aspect of the image (color variation, gradient similarity and correlation). In the case of texture synthesis, the similarity cannot be directly compared as the goal is not to obtain exactly the same result, but to obtain visual coherence in the reconstruction. Thus, we advocate that the measure of the standard deviation and the distance between histograms, also known as Wasserstein metric in Rabin et al. (2011), provides simple and efficient metric for evaluating the quality of our results.

Table 1.2 sums up the comparison of the inpainting step on two examples: an image where the hole has been manually removed and an image where the ground truth is available as the vehicle did a second pass in which the occlusion disappear. For each example, we compute the standard deviation of the region reconstructed by exemplar-based inpainting. We also compute the distance between the normalized

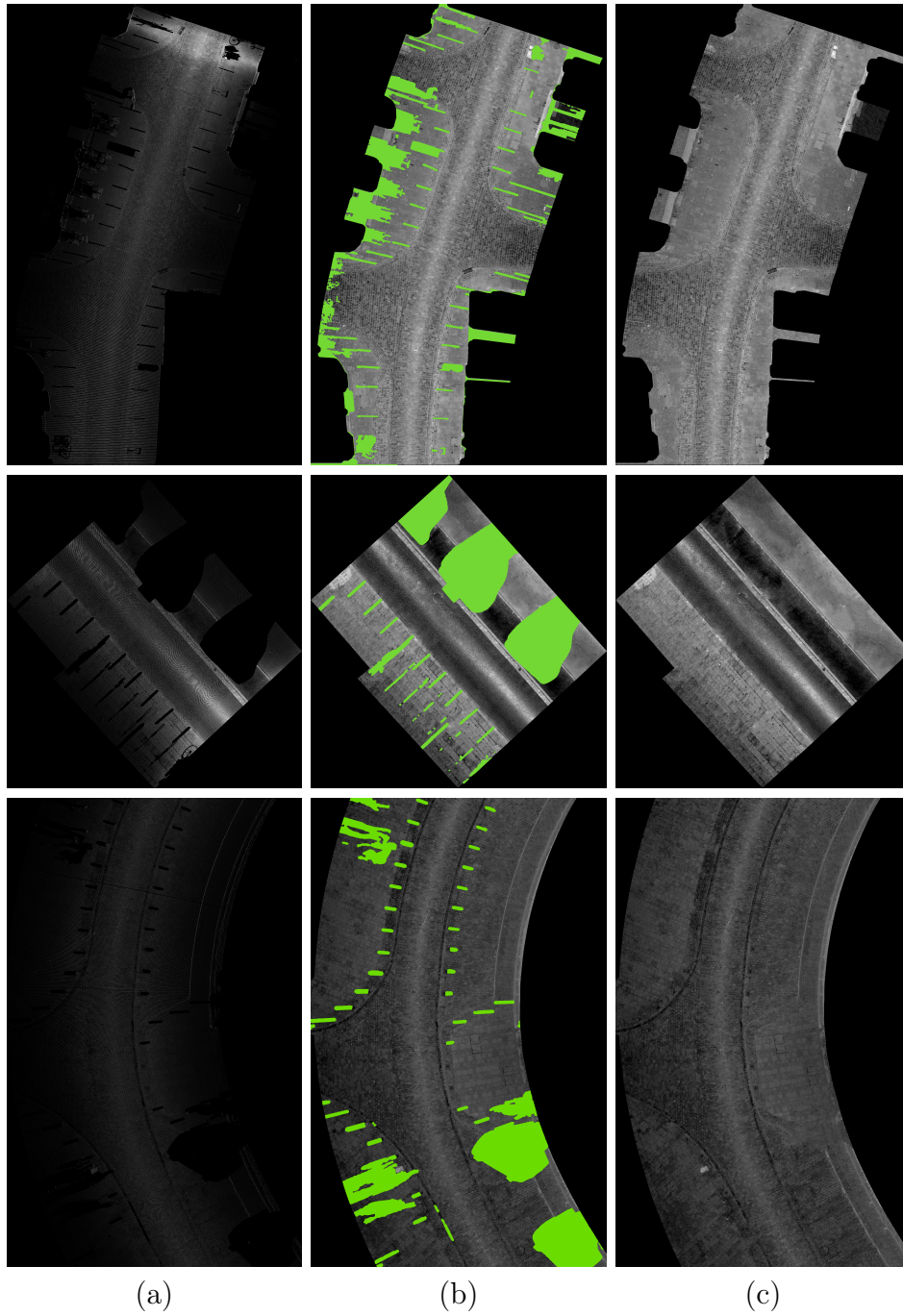


Figure 1.11: Various results on different urban scenes. (a) shows the original point clouds projected on an horizontal grid (sparse). (b) are the results after stripe holes were filled. Areas that present large occlusions are highlighted in green. (c) the final results of our method. In both results, the orthoimage is successfully reconstructed while improving the understandability of the scene.

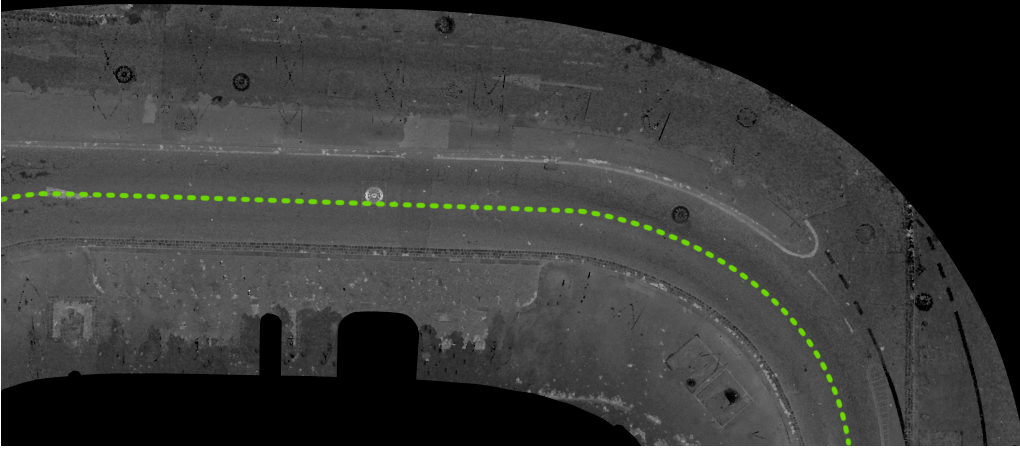


Figure 1.12: Example of scene that follows the vehicle path. In this case, the use of the range information is very relevant. The green dashed line denotes the vehicle path.

histograms of the ground truth and each output. For both examples, our method provides a standard deviation that is very close to the ground truth resulting in visually similar textures.

As the proposed framework also reconstructs the height map of the acquired area, we provide a numerical analysis of this aspect. The choice of the metric in that case is quite easier as the height map is more homogeneous than the reflectance image, especially in a urban scenario as can be seen in Figure 1.15. Therefore, the Normalized Mean Square Error is enough to estimate how good the reconstruction is. We found out that in general the mean square error was below 1cm. This validates the proposed framework for the reconstruction of height map.

1.6.4 Computational speed

The performances of the framework in terms of computational speed are mostly affected by the amount of occlusions and the resolution at which the reconstruction is being made. As the framework is composed of several steps, we present the computation time of each step as well as the total time of processing. All the results are given using MATLAB 2015a on a single thread with an Intel Core i5 CPU at 3.40GHz.

The speed of computation is summed up in Table 1.3. The evaluation is done for the reconstruction of the same point set at different resolutions. The choice of resolution and the amount of stripe holes do not affect much the computation time in proportion. However, the inpainting of large occlusions drastically increases the time of computation in the case of very high resolution. The computation speed of this step might be largely improved by using approaches derivated from Patch-Match (Barnes et al., 2009). Moreover, the framework can be run in parallel as each

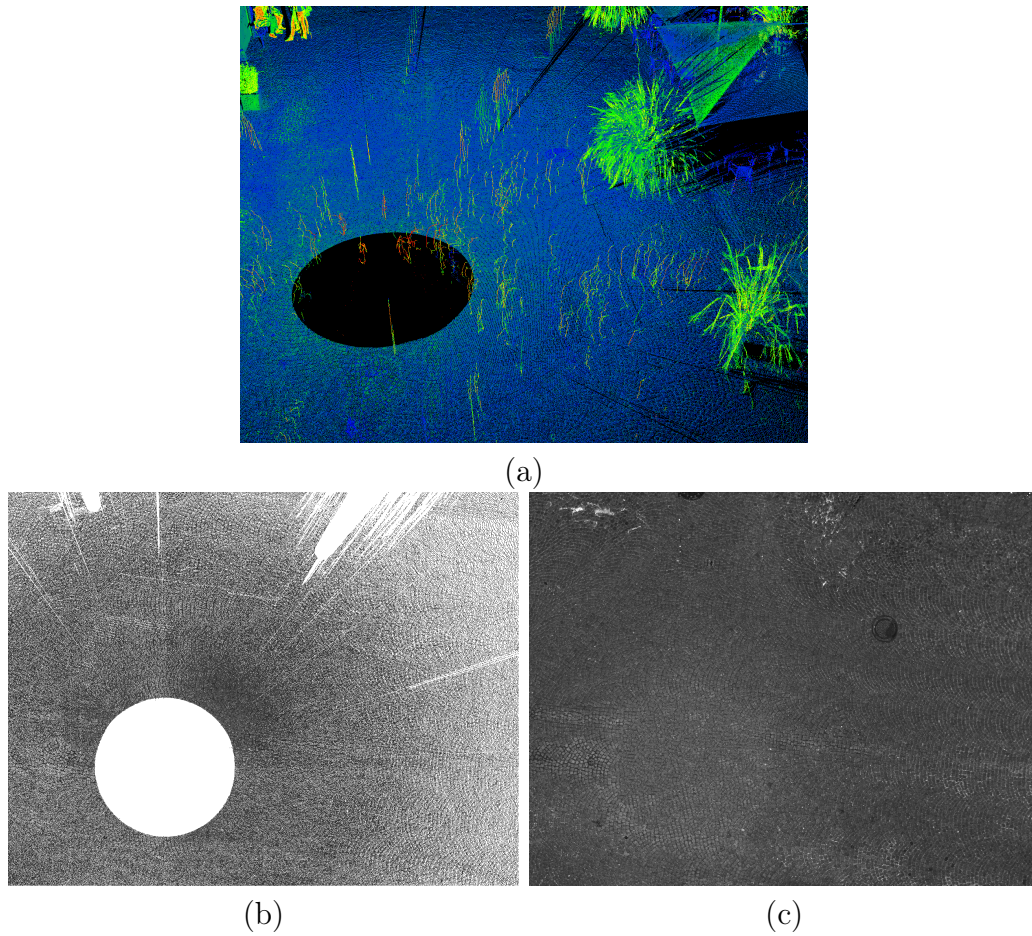


Figure 1.13: Example of reconstruction on the Semantic 3D dataset. (a) is the area from which the orthoimage is acquired, (b) is the projection of the ground points on an horizontal grid, (c) is the final result. The final result provides a plausible estimation of the area under the acquisition sensor.

step is independent of the next ones.

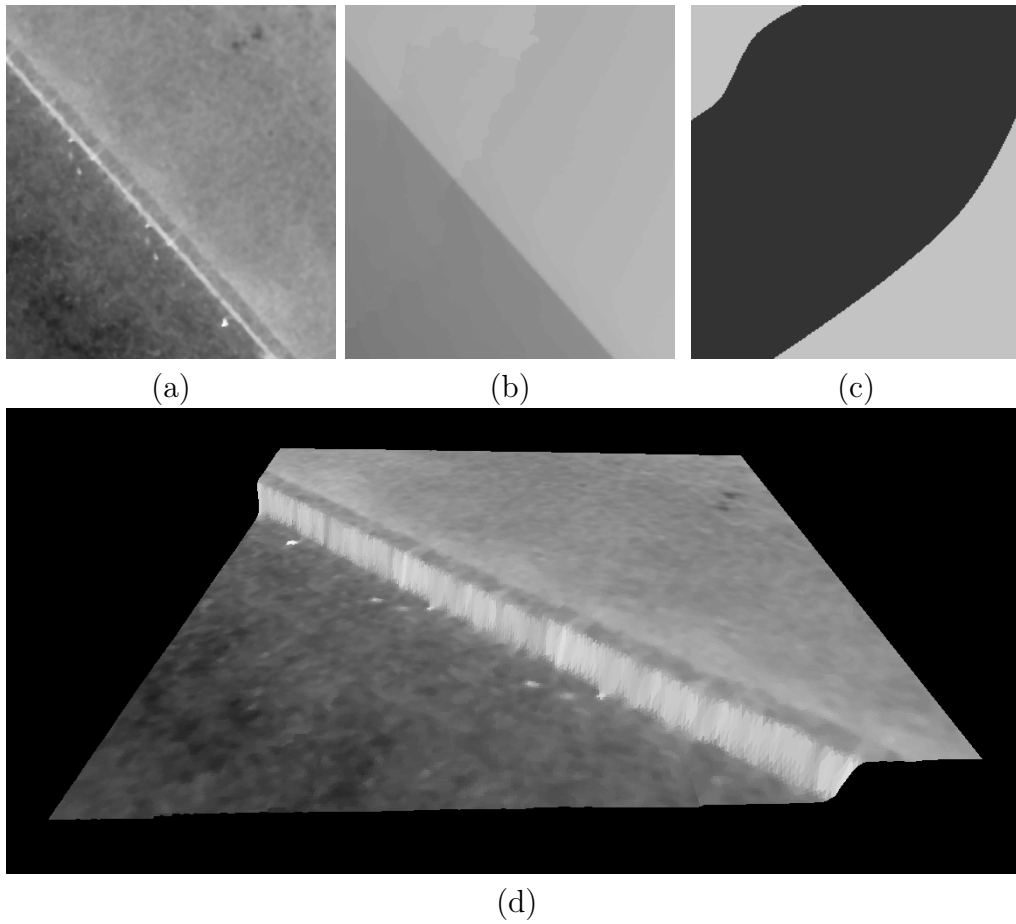


Figure 1.14: 3D model of the ground of a part of an orthoimage. (a) is the reflectance image used as texture for the 3D model, (b) is the height image used as the height coordinate of the 3D model, (c) is the mask where the darkest region was reconstructed using exemplar-based inpainting, (d) is the 3D model obtained using both reflectance and height orthoimages.

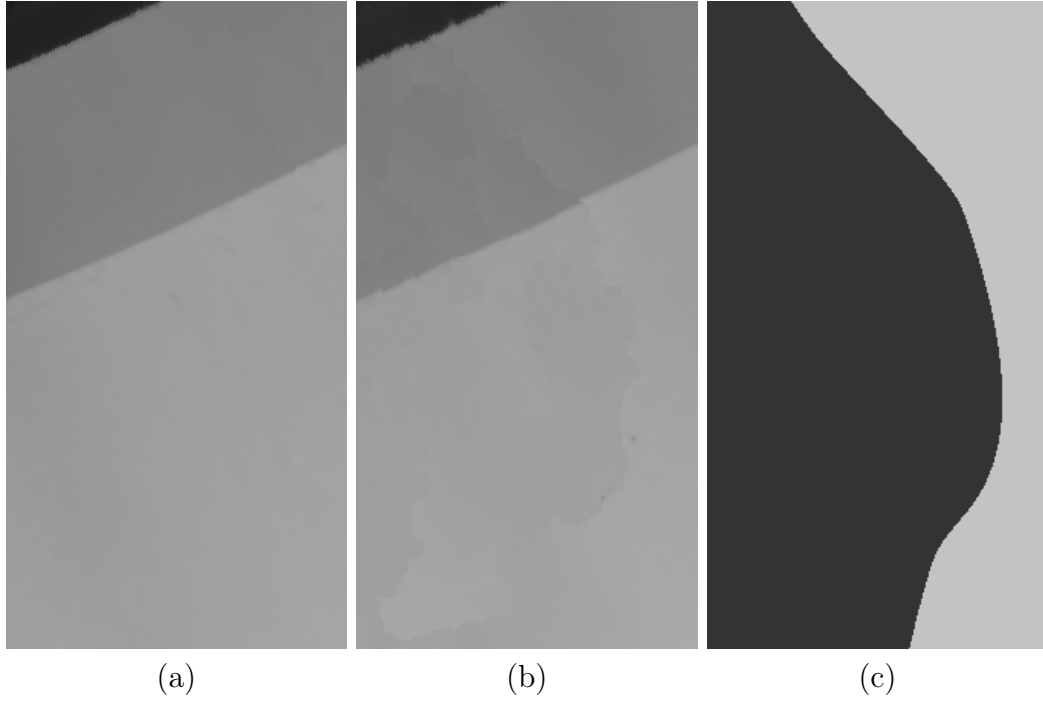


Figure 1.15: Comparison between height images with and without occlusion on the junction between a road and a pavement. (a) is the original height map, (b) is the reconstruction of an occlusion in the same area. The occlusion corresponds to the darkest region of (c). The mean square error of the reconstruction (b) compared to (a) on the occlusion region is 2mm.

Table 1.3: Comparison of computation speed compared to the resolution

Image size	600x550px	2400x2200px
Image resolution	1px = 4cm ²	1px = 1cm ²
Percentage of stripe holes	13%	61%
Percentage of occlusion holes	22%	25%
2D Projection	2.13s	3.78s
Diffusion	1.54s	3.27s
Mask extraction	0.18s	0.91s
Exemplar-based inpainting	23.81s	6.31min
Total	27.66s	6min38s

1.7 Conclusion and future work

In this chapter, we have proposed a complete framework to reconstruct high quality ground orthoimages from a point cloud acquired with LiDAR. This framework consists of several steps, which make use of classical modern imaging techniques. By taking into account the multi-modal nature of the data, we propose several modifications of these methods, leading to significantly better results. The framework is designed to work automatically with a set of parameters that ensures satisfying results on a large variety of input data as demonstrated by the results. Our approach performs at least as well as previous techniques. In case of large occlusions or complex textures, it drastically outperforms earlier works in terms of visual quality. Moreover, robustness towards edge and structure conservation in both reflectance and height domain has been demonstrated. This work has been the subject of two publications: [Biasutti et al. \(2016\)](#) and [Biasutti et al. \(2019a\)](#).

We have seen that diffusion and inpainting algorithms can be used in order to densify the sparse projection of a LiDAR point cloud in order to produce high resolution dense images. In the next chapter, we show how this type of process can be improved by using an extra modality as the support to the densification. Namely, we propose a coupled inpainting method that uses an optical image as the support of the densification of the projection of a LiDAR point cloud.

Chapter 2

Dense depth map from sparse projection

Table of contents

2.1	Introduction	50
2.1.1	Addressed problem and related works	51
2.2	Model	53
2.2.1	Visibility-weighted data-fidelity terms	54
2.2.2	Removal cost	55
2.2.3	Coupled Total Variation	56
2.3	Experimental results	57
2.3.1	Quantitative evaluation with a benchmark data set	59
2.4	Conclusion	61

2.1 Introduction

In the previous chapter, we have proposed to create dense orthoimages from orthogonal projections of LiDAR point clouds on horizontal grids by extracting ground points and performing joint diffusion on both reflectance and elevation without any extra material. Theoretically, the same kind of approach could be used on any type of projection of the point cloud on a 2D pixel grid. In particular, it could be used on a projection of the point cloud in the image domain of an optical image provided by the same MMS as the LiDAR data to create a higher level of representation. However, the case of orthogonal projection of a point cloud on an horizontal grid, as well as the extraction of ground points, relies on many priors that cannot all be assumed when changing the context of application.

On the one hand, orthogonal projection on an horizontal plane allows to compare overlapping points to filter out the wanted slice of information (the ground in the case of the previous chapter). On the other hand, ground points extraction relies on the prior that the ground is locally continuous and relatively planar. As a result the remaining points are all representing the same surface, without any ambiguity.

Recently, RGB-D imaging (*i.e.* images with color and depth channels) have met a lot of success in various applications such as depth-image rendering (Zinger and Do, 2010; Schmeing and Jiang, 2011), gesture recognition (Ren et al., 2013) or augmented reality applied to traffic simulation (Brédif, 2013). These RGB-D images are mainly produced by combining optical images with non-visual sensors such as Time-of-Flight cameras (Kolb et al., 2010) that acquire co-registered depth and color images, or Kinect cameras (Zhang, 2012) that use structural light to retrieve depth information. MMS also allow to build sparse RGB-D images by projecting a LiDAR point cloud in the domain of an optical image, as shown Figure 2.1 (a).

The perspective brought by the projection of the point cloud in the image domain often introduces ambiguities as certain points that were visible from the sensors position are not visible from the optical image point of view, as illustrated Figure on 2.1 (b). Moreover, when considering the projection of the point cloud in an image looking forward for example, many discontinuities arise as moving objects are acquired at different temporalities and do not appear at the same locations in the projection and in the image. Therefore, it is required to reconsider the densification of such projections in order to obtain plausible results. Finally, in the previous chapter, the proposed method only relies on the LiDAR point cloud as no extra structural knowledge was available at this resolution. When projecting the LiDAR point cloud in an optical image domain, it is reasonable to assume that the image can provide supplementary structural information to improve the reconstruction.

The following work has been done together with Marco Bevilacqua during his post-doctorate. I was in charge of the quantitative analysis of this work, which is detailed in Section 2.3.

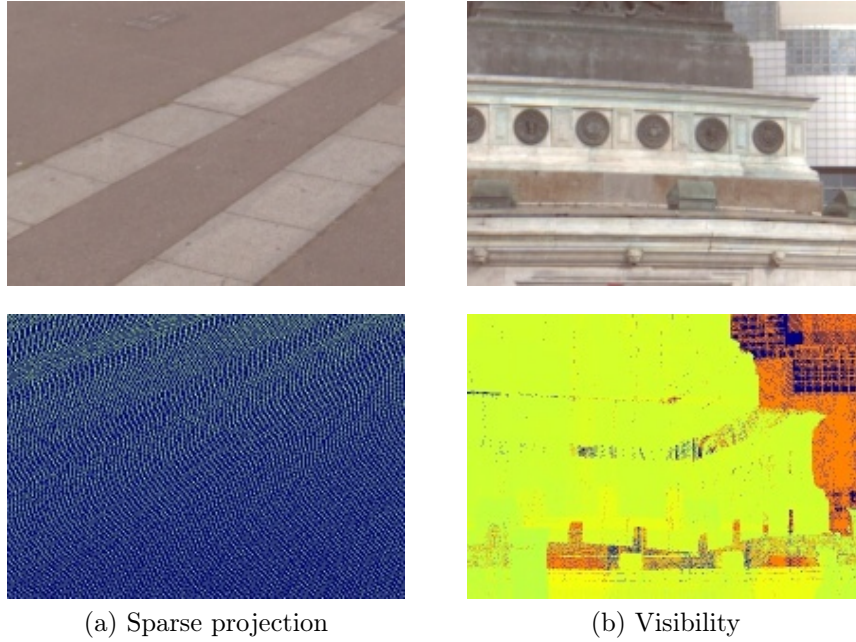


Figure 2.1: Examples of parts from a resulting input depth image (bottom row), with the corresponding parts from the reference color image (top row), showing the issues mentioned in Section 2.1.1: sparse projection and visibility.

2.1.1 Addressed problem and related works

The problem of creating dense RGB-D images from a LiDAR projection in an image domain is strongly related to the problem of depth densification, also known as depth upsampling. This problem arises from the fact that most modern ToF sensors acquire depth information at a lower resolution than the associated optical image, due to practical constraints and cost limitations. Therefore, it is often required to upsample the depth image to the resolution of the associated optical image. Although it can intuitively be done by directly interpolating the depth image to the wanted resolution, this often leads to oversmoothed results. To improve the accuracy of the upsampling, more recent methods use the optical image as a guide for the upsampling.

Multilateral filtering A first type of approaches known as multilateral filtering ([Chan et al., 2008](#); [Yang et al., 2013](#); [Garcia et al., 2010](#)) aims at smoothing the depth image with respect to the optical image edges. This enables the preservation of edges of the depth image. Similar approaches are proposed in ([Park et al., 2011](#)) and ([Huhle et al., 2010](#)), but using Non-Local Means. Although these methods offers better results than basic interpolation, they tend to make small details disappear in the upsampled depth image.

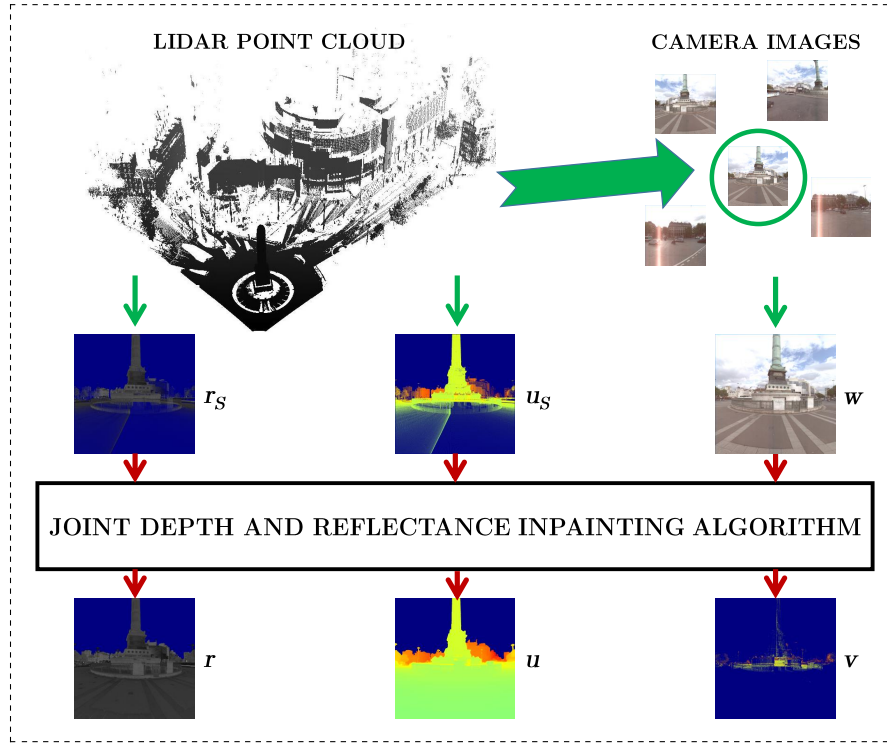


Figure 2.2: General scheme of the proposed approach. The final outputs of the algorithm are the in-painted reflectance and depth images, r and d respectively, and a binary visibility image v . To represent v , we show the original depth values that finally get $v \simeq 0$.

Variational approaches More recently, variational approaches were proposed to better preserve small details as well as strong edges. In (Harrison and Newman, 2010), a method is proposed to assign pixels of the optical image with a depth value, using both image colors and ToF measurement. The problem is posed as an optimization of a cost function encapsulating a spatially varying smoothness cost and measurement compatibility. In the same spirit, the authors of (Ferstl et al., 2013b) present an optimization-based depth upsampling method, which uses an Anisotropic Total Generalized Variation (ATGV) term to regularize the solution while exploiting the optical image information. Another recent algorithm for the upsampling of sparse depth data is presented in (Schneider et al., 2016). The key idea here is to exploit additional object boundary cues (via structured edge detection and semantic scene labelling) together with usual intensity cues in a unique optimization framework. These methods lead to more accurate results than multilateral filtering approaches and they succeed in good preservation of small details in the upsampled depth image.

Although the problem of depth upsampling has similarities with the problem that we aim at addressing, none of the presented methods consider the problem of visibility mentioned above. Indeed, as the depth sensor is located close to the optical camera,

there is no need to exclude some depth measurement in the upsampling process.

In this chapter, we present a novel approach for the generation of RGB-D images using LiDAR point cloud projection with visibility estimation. Figure 2.2 depicts the scheme of the proposed approach. Given an MMS data set consisting of a LiDAR point cloud and a set of camera images, we choose among the latter a reference color image (I), and we obtain input depth (ζ_S) and reflectance (r_S) images by re-projecting the LiDAR points according to the image geometry. The two LiDAR-originated images are sparse images with irregular sampling and need to be inpainted. We propose to do that jointly and simultaneously estimate the visibility of the input points, within a variational optimization framework. There are three outputs to the algorithm: the inpainted depth and reflectance (ζ and r , respectively), and a binary image expressing the visibility at each point (v).

2.2 Model

Let $\Omega \subseteq \mathbb{R}^2$ be the “full” image support, and $\Omega_S \subseteq \Omega$ the sparse image support where the input images are defined (*i.e.* there is at least one LiDAR point ending up there after projection). Given an input depth image $\zeta_S : \Omega_S \rightarrow \mathbb{R}$, an input reflectance image $r_S : \Omega_S \rightarrow \mathbb{R}$, and the luminance component of their corresponding color image $I : \Omega \rightarrow \mathbb{R}$ (defined in the complete domain), the goal is to fully inpaint the depth and reflectance input images to obtain $\zeta : \Omega \rightarrow \mathbb{R}$ and $r : \Omega \rightarrow \mathbb{R}$, and concurrently estimate a visibility attribute $v : \Omega_S \rightarrow \mathbb{R}$. For each input pixel, v indicates whether it is visible from the image view point and should thus be taken into account in the inpainting process. Figure 2.3 reports an example of three possible input images - depth (ζ_S), reflectance (r_S) and camera images - and their respective gradient images.

We model our joint inpainting problem as an optimization problem with three variables, ζ , r , and v , to be estimated. Lower and upper bounds for the values of ζ and r are considered in the expression. The visibility attribute v takes values in $[0, 1]$, where $v = 0$ stands for “hidden” and $v = 1$ means that the point is visible from the considered image view point. The model considered consists of four terms:

$$\min_{\substack{\zeta \in [\zeta_m, \zeta_M] \\ r \in [r_m, r_M] \\ v \in [0, 1]}} F(\zeta, v | \zeta_S) + G(r, v | r_S) + H(v | \zeta_S, r_S) + R(\zeta, r | I) . \quad (2.1)$$

$F(\zeta, v | \zeta_S)$ and $G(r, v | r_S)$ are two data-fidelity terms, for depth and reflectance respectively. In both of them the visibility attribute v intervenes. $H(v | \zeta_S, r_S)$ is a term depending exclusively on v , which represents the total cost of classifying input pixels as non-visible. Finally, $R(\zeta, r | I)$ is a regularization term that penalizes the total variation of ζ and r , by also taking into account the color image w . In next sections we will detail all the terms composing (2.1).

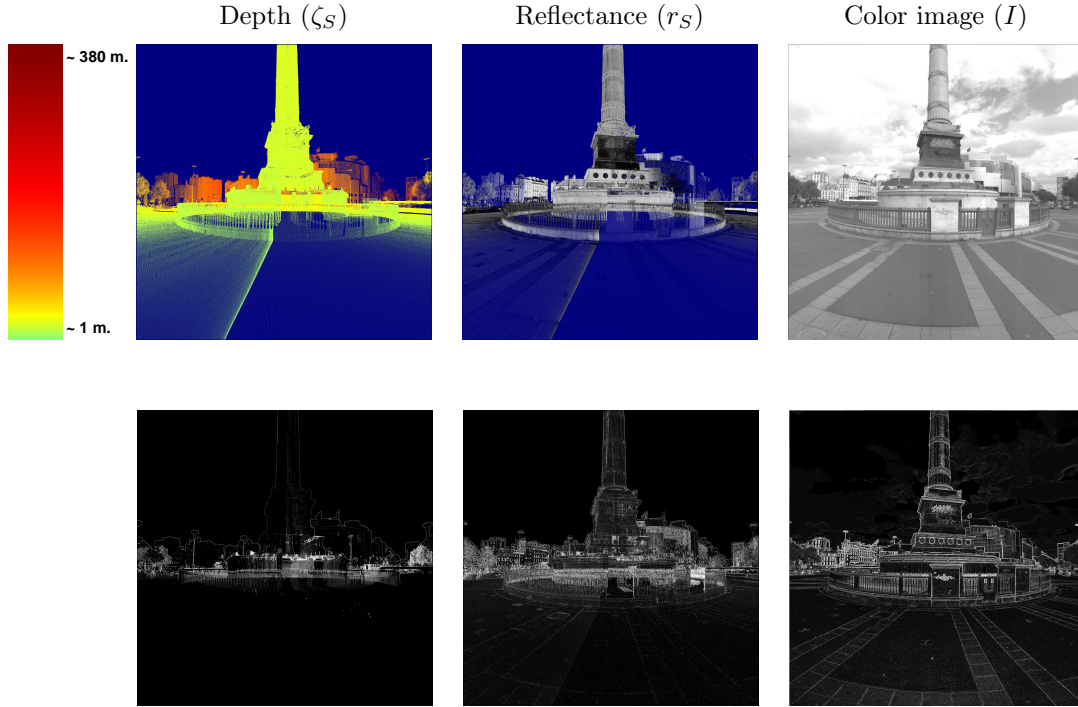


Figure 2.3: Example of input depth, reflectance and color images (top row), and their respective gradient images (bottom row). Beside the input depth image, the color map used to encode depth values is reported. Gradients of depth and reflectance are computed on the interpolated versions of the input sparse images, initially obtained by nearest neighbor interpolation.

2.2.1 Visibility-weighted data-fidelity terms

The data-fitting terms in (2.1) are meant to enforce fidelity to the original values of depth and reflectance, ζ_S and r_S respectively. Deviations from the original values are more penalized if the points are considered “trustful”; conversely, for erroneous original measures (*e.g.* referring to hidden points) larger deviations are allowed. Therefore we use the visibility attribute v to weight the data terms. For the reflectance data-fidelity term $G(r, v|r_S)$ we have the following expression:

$$G(r, v|r_S) = \eta_2 \int_{\Omega_S} v |r - r_S| dx_1 dx_2, \quad (2.2)$$

where η_2 is a coefficient weighting the term within the model, and dx_1 and dx_2 express the differential lengths in the two image directions. Note that in (2.2) a ℓ_1 -norm error is used. The ℓ_1 norm is considered in substitution of the classical ℓ_2 measure of the error for its effectiveness in implicitly removing impulse noise with strong outliers (Nikolova, 2004) and its better contrast preservation (Chan and Esedoglu, 2005). As said, weighting by v relaxes the dependence on the input data for those points classified as hidden.

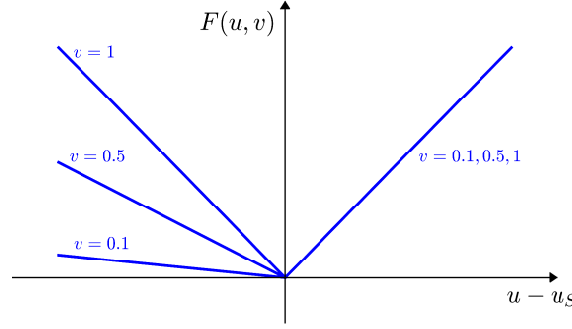


Figure 2.4: Depth data-fidelity cost $F(\zeta, v|\zeta_S)$ as a function of $\zeta - \zeta_S$ for different values of v ($\eta_1 = 1$ for simplicity). For over-estimated depths ($\zeta - \zeta_S > 0$) the cost is independent of v , whereas for $\zeta - \zeta_S < 0$ we have different lines as v varies.

The depth data-fidelity term, weighted by the coefficient η_1 , is further divided into two terms, as follows:

$$\begin{aligned} F(\zeta, v|\zeta_S) &= \eta_1 \left(\int_{\Omega_S} \max(0, \zeta - \zeta_S) dx_1 dx_2 + \int_{\Omega_S} v(\max(0, \zeta_S - \zeta)) dx_1 dx_2 \right) \\ &= F_1(\zeta|\zeta_S) + F_2(\zeta, v|\zeta_S). \end{aligned} \quad (2.3)$$

The basic idea behind this separation is to treat differently over- and under-estimated depths. Points for which the estimated depth is greater than the original value ($\zeta > \zeta_S$) most likely correspond to correct input measures, where the over-estimation would be due to the surrounding presence of larger erroneous depths. The expression $\max(0, \zeta - \zeta_S)$ is meant to select this kind of points (over-estimated depths). As they are considered reliable, an unweighted data-fitting term, $F_1(\zeta|\zeta_S)$, is imposed. It is easy to see that for these points the visibility attribute v tends to converge to 1, *i.e.* they are the best candidates for being classified as visible points. Conversely, the hidden points to remove are sought among depth values which undergo under-estimation ($\zeta < \zeta_S$). These points are taken into account in the second term $F_2(\zeta, v|\zeta_S)$, where the ℓ_1 error is weighted by the visibility attribute. Ideally, a fraction of them, the most “problematic” ones, will be classified as hidden ($v = 0$) and thus not considered in the data fitting cost. Figure 2.4 shows graphically the depth data-fidelity cost as a function of $\zeta - \zeta_S$. Depending on the value of the visibility attribute v , the ℓ_1 -type error $|\zeta - \zeta_S|$ is relaxed for negative depth deviations ($u < u_S$).

2.2.2 Removal cost

The second term of the model (2.1) is meant to penalize the total number of hidden points.

$$H(v|\zeta_S, r_S) = \int_{\Omega_S} \alpha(\zeta_S, r_S)(1 - v) dx_1 dx_2. \quad (2.4)$$

The cost of a single pixel exclusion is proportional to $1 - v$, *i.e.* we have the highest cost for an input pixel when it is totally excluded in the data-fitting cost ($v = 0$). We individually weight each removal cost, in order to give different importance to each decision visible/hidden. Individual weighting is given by a coefficient dependent on the original depth and reflectance values, $\alpha(\zeta_S, r_S)$. We generally choose $\alpha = k_1\zeta_S + k_2r_S$. The linear dependence of α on the depth and the reflectance “balances” the three terms of (2.1) depending on v , such that k_1 and k_2 appear to be constants.

2.2.3 Coupled Total Variation

Depth upsampling/inpainting methods that exploit corresponding camera images often relate image edges to depth edges. This has shown to improve the quality of the reconstructed depth images.

To couple two images in a total variation framework, we adopt the *coupled* total variation (coupled TV) of (Pierre et al., 2015):

$$\text{TV}_\lambda(a, b) = \int_{\Omega} \sqrt{(\partial_{x_1}a)^2 + (\partial_{x_2}a)^2 + \lambda^2(\partial_{x_1}b)^2 + \lambda^2(\partial_{x_2}b)^2} dx_1 dx_2 . \quad (2.5)$$

where λ is a coupling parameter. When $\lambda \neq 0$ the minimization of TV_λ encourages the gradient “jumps” to occur at the same locations in a and b . The coupled TV is then a way to align the edges of an image with those of a given one.

In our problem we have three types of images: a color image I , a depth image ζ , and a reflectance image r . Figure 2.3 reports in the bottom row an example of gradient magnitudes related to three images. The gradients of the input depth and reflectance images have been computed after initial interpolation of the latter. As we can clearly see from the image, the color image gradient particularly matches the reflectance one, while being rather dissimilar to the depth gradient. In turn, the reflectance gradient shares some patterns, yet less prominently, with the depth one (*e.g.* the area at the base of the column, where multiple layers mix and produce a similar effect in the two gradient images). We therefore propose to match the three gradients two by two: depth with reflectance, and the same reflectance with the fixed color image. By using the previous definition of coupled TV (2.5), we express the regularization term as follows:

$$R(\zeta, r|I) = \text{TV}_{\lambda_1}(\zeta, r) + \text{TV}_{\lambda_2}(r, I) . \quad (2.6)$$

After detailing all the terms, our model (2.1) can therefore be rewritten as follows, the four terms being still distinct:

$$\begin{aligned} \min_{\substack{\zeta \in [\zeta_m, \zeta_M] \\ r \in [r_m, r_M] \\ v \in [0, 1]}} & \underbrace{\eta_1 \left(\int_{\Omega_S} \max(0, \zeta - \zeta_S) + \int_{\Omega_S} v(\max(0, \zeta_S - \zeta)) \right)}_{F: \text{Data-fidelity for Depth}} + \underbrace{\eta_2 \int_{\Omega_S} v|r - r_S|}_{G: \text{Data-fidelity for Reflectance}} \\ & + \underbrace{\int_{\Omega_S} \alpha(\zeta_S, r_S) (1 - v)}_{H: \text{Removal cost}} + \underbrace{\text{TV}_{\lambda_1}(\zeta, r) + \text{TV}_{\lambda_2}(r, I)}_{R: \text{TV regularization}} . \end{aligned} \quad (2.7)$$

This model is solved with a primal-dual algorithm. More details can be found in Appendix A.

2.3 Experimental results

The method is evaluated with a dataset acquired by the Stereopolis-II (Paparoditis et al., 2012) composed of LiDAR measures and camera-originated images. With this data set, we provide a qualitative evaluation of our algorithm in comparison with other methods, by showing the reconstructed depth and reflectance images, and we assess the quality of the visibility estimation task, which is a crucial characteristic of our algorithm. Moreover, we also provide a quantitative analysis on the KTTI dataset (Geiger et al., 2012). Before showing results and comparisons, in Section 1 we motivate some critical choices in terms of model and algorithmic parameters.

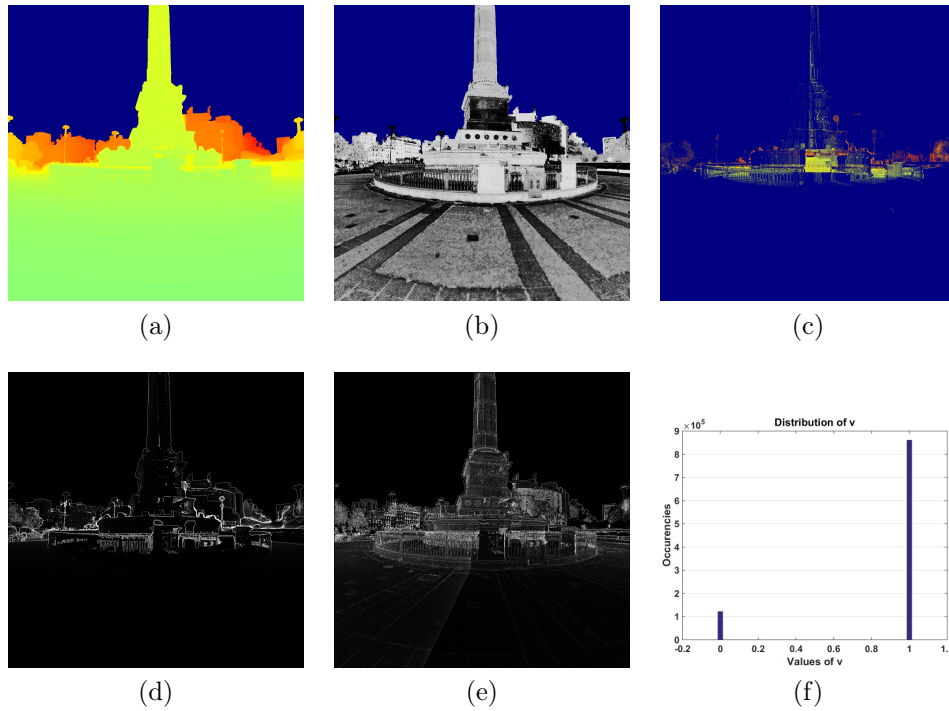


Figure 2.5: Output of the proposed algorithm: (a) Inpainted depth, (b) Inpainted reflectance, (c) Removed points ($v = 0$), (d) Final depth gradient, (e) Final reflectance gradient, (f) Final histogram of v .

If we observe the input sparse depth image of Figure 2.3, we see that the major problems come from the fact that depth values referring to the building behind the column appear mixed with foreground depths. With our algorithm we are able to resolve these conflicts, as we can see in the inpainted depth image (Figure 2.5a). Part

of the input pixels have in fact been removed, *i.e.* classified as non-visible ($v = 0$). Figure 2.5c reports the locations of such points in the original depth image. From the histogram of the values of v (Figure 2.5f) it is evident that the algorithm produces a bi-partition of the points according to their visibility attribute. Figure 2.5 shows also the inpainted reflectance and the final depth and reflectance gradients. By comparing the latter to the original gradients (Figure 2.3), we can observe that they end up incorporating elements of the color image gradient, while removing erroneous edges. Moreover, in Figure 2.6, we show the result of the proposed method. There, we can see how precise the resulting depth and reflectance images are. In particular, we can see that the visual ambiguities have completely disappear as no artifact are present in the final image. Further details about parameters as well as more visual results are available in Appendix B.

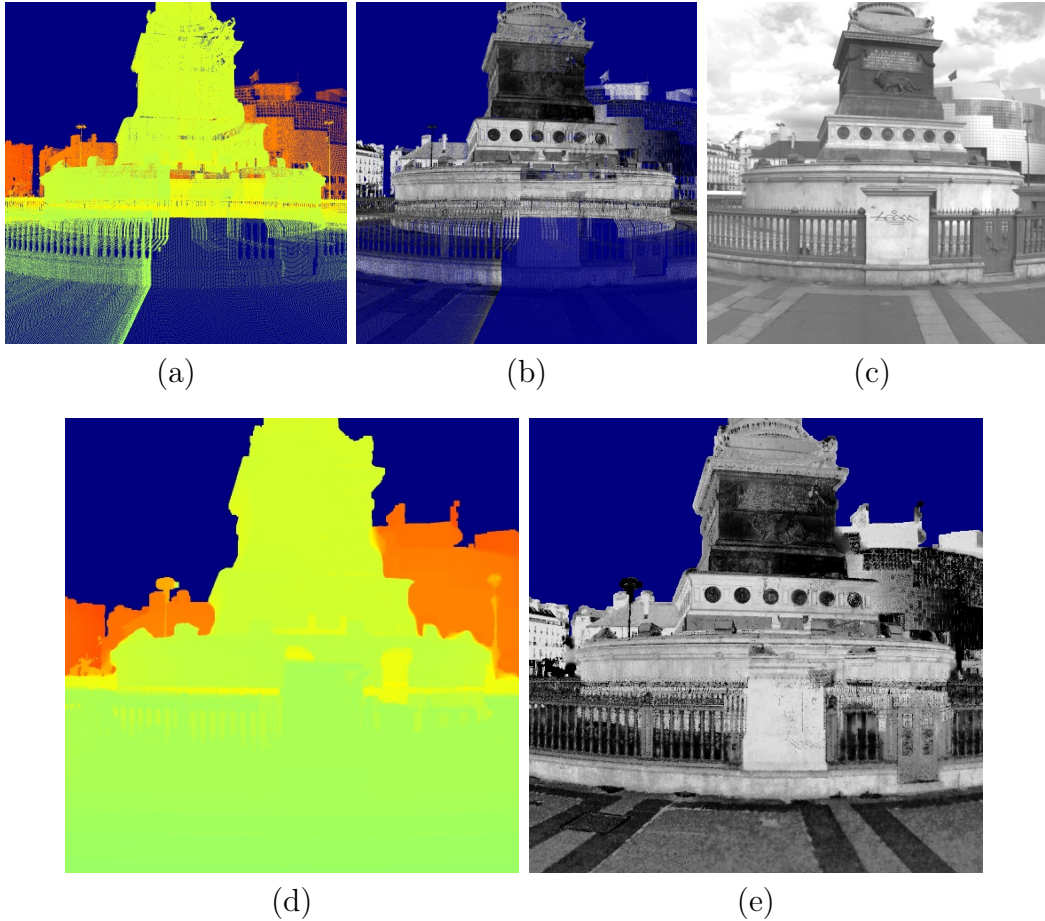


Figure 2.6: Example of result produced by the method. (a) input depth, (b) input reflectance, (c) input image, (d) reconstructed depth output and (e) reconstructed reflectance output. We can see that the results are visually convincing.

2.3.1 Quantitative evaluation with a benchmark data set

As mentioned above, my main contribution to this work was to carry the quantitative analysis of this method. To that end, I used the dataset provided by the KITTI Vision Benchmark Suite (Geiger et al., 2013). The LiDAR measures are generally used as ground truth for algorithm evaluations. In (Menze and Geiger, 2015) a novel dataset is presented for stereo benchmarking, which considers also moving objects. By making a special processing on the latter and manually removing erroneous points due to occlusions, ground truth disparity maps are obtained. These maps appear “cleaner” and denser than the input depth images that can be obtained with the raw LiDAR data, and they can therefore be used to evaluate algorithm estimating disparity. To exploit this possibility, as described in (Schneider et al., 2016, Sec. 4.3), we use the ground truth maps of this stereo benchmark data set to have a quantitative evaluation of our depth+reflectance inpainting algorithm. As done by the authors of (Schneider et al., 2016), we identify 82 frames (provided ground truth disparity maps) for which we can find correspondences in the raw data set, *i.e.* a corresponding color image and related LiDAR point cloud. We then use the raw data LiDAR to compute an input depth (*e.g.* Figure 2.7a) and we use the provided ground truth map to compute a Mean Absolute Error (MAE):

$$\text{MAE}(u_1, u_2) = \frac{1}{N} \sum_{i,j \in \Omega} |u_1(i, j) - u_2(i, j)| \quad (2.8)$$

having u_1, u_2 are images defined on Ω with N pixels where each pixel intensity represents the depth value. The ground truth maps, although denser than the input maps, are sparse, *i.e.* they are not defined for all pixels (only about 19% of the pixels have values). Thus, the MAE is computed only for those pixels which are defined in the respective ground truth map.

We computed the MAE for all 82 frames of the found correspondences, for our method and the ATGV-based algorithm of (Ferstl et al., 2013b). We also compare with a two-step approach, where AGTV-based inpainting is preceded by a hidden point removal (HPR) operation, performed with the algorithm of (Katz et al., 2007). The resulting average MAEs, which are measured as the average pixel displacement between two disparity maps, are reported in Table 2.1.

	ATGV	HPR+ATGV	Proposed
Average MAE (<i>px.</i>)	2.13	2.07	1.99

Table 2.1: Average Mean Absolute Error (MAE), *i.e.* average pixel displacement between ground truth and reconstructed disparity maps, obtained by averaging the results of 82 frames of the 2015 KITTI stereo benchmark data set.

When creating the ground truth maps, the authors of the KITTI benchmark data set have removed objects presenting particular issues in terms of visibility. Other

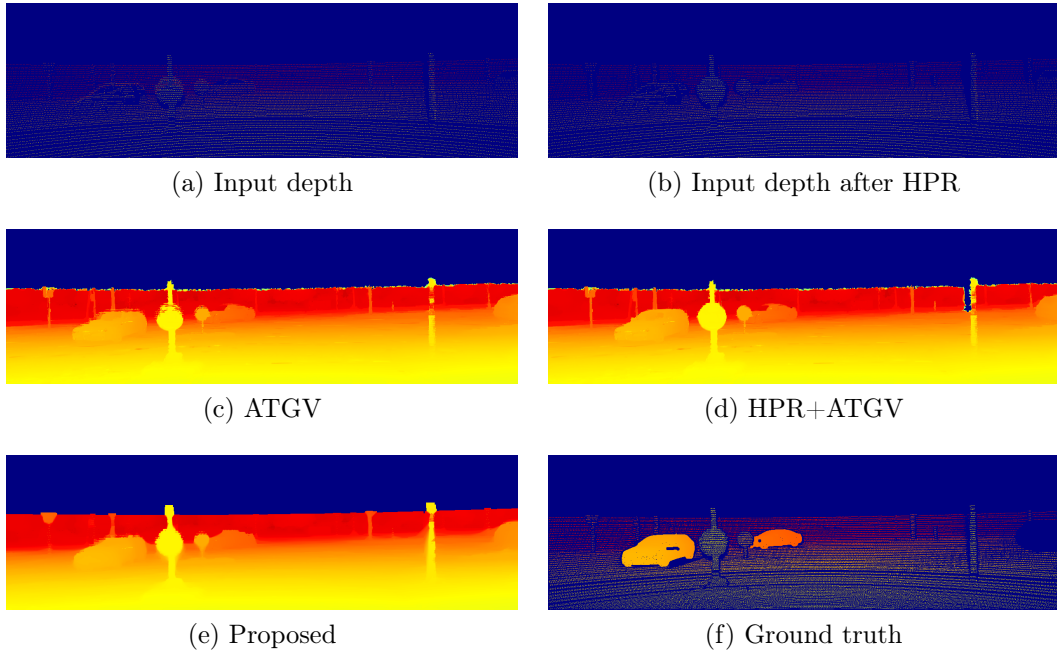


Figure 2.7: Case example from the 2015 KITTI stereo benchmark data set. For each input depth map (a), we have a ground truth disparity map available yet sparse (f), for which it is possible to compute an error by only considering the pixels where it is defined. By applying a hidden point removal (HPR) algorithm to the input depth data it is possible to create a new input map where background hidden pixels have been removed (b). Results for different depth inpainting strategies are reported (c, d, e).

objects are instead manually handled (they are removed from the scene and re-inserted after fitting a CAD model). Thus, the ground truth maps basically consist of the latter and fixed parts of the scene (*e.g.* streets and walls) that do not yield any ambiguity. Due to this relative “simplicity” of the data set, the performance in terms of average MAE are rather similar among the three methods (ATGV, HPR+ATGV, and proposed method), with our method obtaining a slightly lower error. Nevertheless, we can observe that the ATGV method of (Ferstl et al., 2013b) produces more artifacts (see, for example, the reconstructed pole on the left in Figure 2.7c, in comparison to Figure 2.7e). Most of these artifacts can be removed by performing a preliminary HPR step (see, in Figure 2.7b, an example of input depth map cleaned out of ambiguous pixel). The combination of a HPR step and the ATGV-based depth upsampling algorithm of (Ferstl et al., 2013b) yields inpainted depth maps with a visual quality comparable to the one of our approach. However, as stated in Section 3, with our approach we keep the advantage of having an all-in-one procedure performing jointly inpainting and “soft” visibility estimation (without the need of setting a per-image global threshold as requested by the algorithm of (Katz et al., 2007)). We also expect for our method a greater improvement of the MAE metric and the visual outcome on more complex scenes.

2.4 Conclusion

In this chapter, we have presented a strategy to jointly densify depth and reflectance images with the guidance of a co-registered color image, and by simultaneously estimating a visibility attribute for each pixel. The proposed approach is particularly suited for LiDAR and optical data acquired by MMS. By projecting the 3D LiDAR points in the domain of a chosen image, we obtain depth and reflectance images, which suffer of practical issues due to the big diversity of the LiDAR and optical sensor acquisitions. By estimating visibility, we aim at solving one of these issues (*i.e.* the appearance (in depth and reflectance) of parts of objects that are non-visible from the image view point, but captured by the LIDAR sensor). Those points are meant to be detected by our algorithm and thus discarded in the densification process. The proposed approach consists in a variational optimization problem, where three variables (depth, reflectance, and visibility) are simultaneously estimated. The superiority of the proposed method compared to the state-of-the-art proves that the visibility estimation is a necessary step and it also indicates that the joint exploitation of depth and reflectance is a key aspect for the success of the algorithm. The mutual benefit comes from the fact that depth is particularly important for the visibility estimation task; in turn, reflectance is crucial in restoring the correct edges, via coupling with the color image. This work has been the subject of the following publication: [Bevilacqua et al. \(2017\)](#).

However, such heavy variational model uses a lot of computational time. Moreover, some applications require to project the point cloud in point of views that were not acquired with optical cameras. In this case, the use of the proposed method for visibility estimation is not possible. Thus, developing a faster visibility estimation method that relies only on the point cloud is crucial. This is the subject of the next chapter.

Chapter 3

Visibility estimation of a point cloud from a given point of view

Table of contents

3.1	Introduction	63
3.2	Related works	64
3.3	Visibility estimation method	66
3.4	Visibility estimation dataset for LiDAR point clouds	68
3.4.1	Overview of the dataset	69
3.5	Experiments & Results	70
3.5.1	Evaluation on the Visibility Estimation Dataset	70
3.5.2	Evaluation on constant density point cloud	74
3.5.3	Example of application to data fusion	77
3.6	Conclusion	77

3.1 Introduction

The estimation of the visibility of a point cloud consists in assigning a label to each point of the scene: visible if the point lies on an object that is directly visible from a given viewpoint, non-visible otherwise (Fig. 3.1). This task is a typical step for various applications in computer graphics such as in surface reconstruction (Zach et al., 2007; Shalom et al., 2010; Berger et al., 2017) in which estimating and removing points that are not visible from a given point of view improves the interpolation and the approximation of the surface to recover. In point cloud rendering and visualization (Pintus et al., 2011; Bouchiba et al., 2017), the estimation of the visibility enables better rendering performances as well as an improvement of the scene understanding.

In the previous chapter, we have shown how the estimation of visibility can be used along with optical image in a variational model to produce a better densification of the projection of a LiDAR point cloud. However, such a model is slow as the proposed energy function contains many terms and it requires several thousands of iterations to converge. Moreover, some of the applications mentioned above require to estimate the visibility in novel points of view, where no optical image is available. In this case, the use of the method proposed in Chapter 2 is not suitable. On the other hand, the existing methods (Zach et al., 2007; Shalom et al., 2010; Berger et al., 2017; Pintus et al., 2011; Bouchiba et al., 2017) strongly rely on strict sampling assumptions (Berger et al., 2017) (*e.g.* on point clouds with constant density in terms of number of points per cubic meters).

The multi-modal aspect of MMS data, especially LiDAR and optical, may be leveraged to improve detection, classification and prediction techniques in urban environments (Benenson et al., 2014; Eigen et al., 2014). Therefore, the fusion and the registration of LiDAR and optical data became critical as it is a pre-requisite to increase performances of classification/prediction algorithms. Most of the recent related works strongly rely on good visibility estimates (Mastin et al., 2009; Guislain et al., 2017).

On the one hand, the majority of actual LiDAR/optical registration techniques that use visibility rely on estimation techniques that were built for point clouds with strict sampling assumptions - meaning that the density of points has to be the same everywhere in the point cloud - that are not met by the LiDAR data on which they operate. On the other hand, point cloud rendering and surface reconstruction methods presented above are not designed to perform on point clouds with variable density. However, the quality of the visibility estimation is a crucial preprocessing step for multi-modal fusion applications as it drastically lowers the ambiguities from one modality to another.

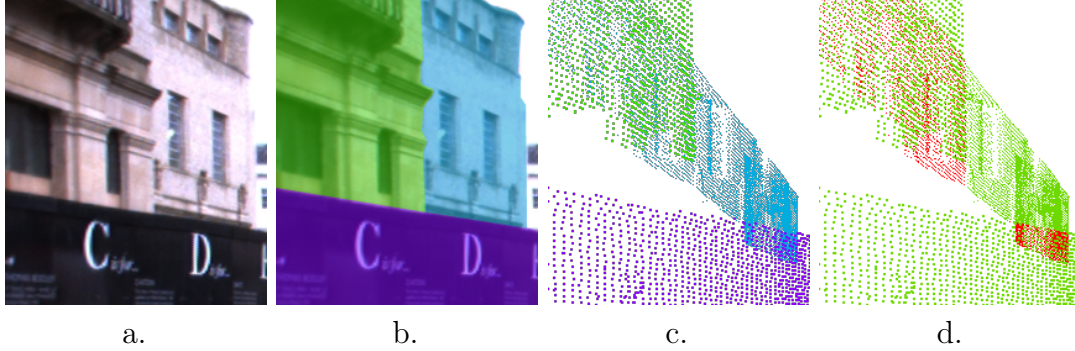


Figure 3.1: Illustration of the visibility problem. (a) an optical image corresponding to a view point, (b) the 3 main structures of the scene, (c) the projection of the acquired point cloud seen from the same view point with same colors as in (b), (d) the visibility errors brought by the projection (red points should not be visible).

3.2 Related works

There have been many contributions to the state-of-the-art techniques for the estimation of the visibility of a point cloud given a certain viewpoint. In this section, we review the methods that are most relevant to the stated problem.

Surface reconstruction based methods One intuitive way to compute the visibility of a point cloud is to first reconstruct the surface. Indeed, the projection of the surface as a depth map may be used to estimate which points are not visible by simply comparing the depth of each projected point to the observed depth of the surface at the same location. If both measures are similar, the point is visible, otherwise if the points is farther, it is considered as hidden. As some surface reconstruction methods do not require prior knowledge of the visibility, they can be used for visibility estimation. To that end, *Surface smoothness* approaches (Lipman et al., 2007; Xiong et al., 2014) approximate the surface by locally defining operators that weigh surrounding points in order to estimate the local surface. This constrains the reconstructed surface to fit the point cloud as close as possible while ensuring a certain level of smoothness and preserving sharp features. To deal with large amounts of missing data, *Volume smoothness* techniques (Tagliasacchi et al., 2011; Huang et al., 2013) exploit the prior of smooth variation of the volume of the reconstructed surface. Unfortunately, these methods are based on strong prior of uniform sampling of the point cloud, which is not suitable for MMS LiDAR point clouds. *Primitive based* methods (Schnabel et al., 2009; Lafarge and Alliez, 2013) aim at fitting geometric shapes (*i.e.* planes, spheres, cylinders, boxes, etc.) in order to reconstruct the scene. However, the complex shapes that can be met in real world scene often jeopardize the results of such methods. Finally, *Global regularity* approaches (Li et al., 2011a,b; Monszpart et al., 2015) take advantage of the repeatability of certain

parts of the scene. These methods have shown great strength for the reconstruction of individual regular shapes such as facades or roads but underperform on realistic complete scenes. Although each technique provides satisfying results on specific scenarios, surface reconstruction is a difficult problem, which often requires additional information, such as normals, sufficiently dense input and uniform sampling.

Convex hull based methods Some methods estimate the visibility based on the local geometry of the 3D point cloud. Based on the raw point cloud (*i.e.* only 3D positions), (Katz et al., 2007) propose an approach for estimating which part of the point cloud is not self-occluded given a certain viewpoint. This method performs better on closed shapes. First, a spherical inversion is performed on the point cloud. The goal being to inverse which side of the object is facing the observer. After that, the convex hull of the inverted point cloud augmented by the viewpoint position is computed. The convex hull of a set of points is the smallest convex set that contains all the points, as illustrated in Figure 3.2 (left). Then, points that are lying on the convex hull are considered visible whereas the rest of the point cloud is considered non-visible. This principle is shown in Figure 3.2 (right). The acceptance of concave features is tuned by the sphere radius, which is a global parameter so that this method strongly relies on a uniform sampling of the point cloud. Later, this method was improved to handle small changes in the sampling corresponding to noisy acquisitions in (Mehra et al., 2010). Here, the authors propose to introduce a threshold that allows to also consider points that are close to the convex-hull. However, this method still relies on constant density in the point cloud. Moreover, the computational cost of the convex hull (Barber et al., 1996) can rapidly increase depending on the wanted concavity. Finally, (Katz et al., 2007) and (Mehra et al., 2010) are both designed to perform on point clouds that represent closed shapes, acquired from all directions, which is not realistic in urban scenarios where MMS are not able to scan all surfaces.

Likelihood based methods Different methods aim at estimating the likelihood of a point to be visible, given a point of view, by considering its neighborhood. The most common methods rely on the estimation of visibility cones in screen-space (*e.g.* in the domain of a 2D projection) (Shalom et al., 2010), and more recently (Pintus et al., 2011). For each point, a visibility cone is estimated by considering its neighborhood. The aspect of the cone is directly related to the visibility. A point that belongs to a wide cone is more likely to be visible than a point that belongs to a narrow cone. If the line that fits the point of view and the point intersects the cone, and if the cone is wide enough compared to a given threshold, the point is considered visible. This principle is illustrated on Figure 3.3. However, the opening threshold used to consider whether a point is visible or not strongly depends on the point cloud, and can be hard to set.

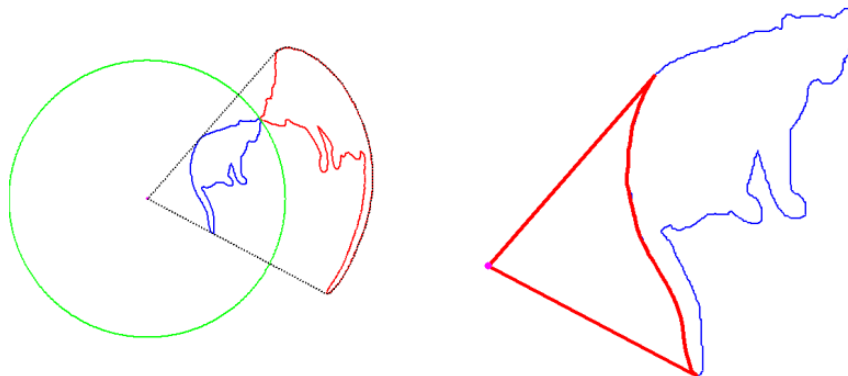


Figure 3.2: Left: convex hull (dotted line) computed on the set of points composed of a spherically inverted model (red) and the center of the point of view (magenta). The inverted model is obtained by inverting the original model (blue) with respect to the green circle. Right: back projection of the convex-hull. Here, points of the model (blue) that lie on the projected convex-hull (red) are considered visible.

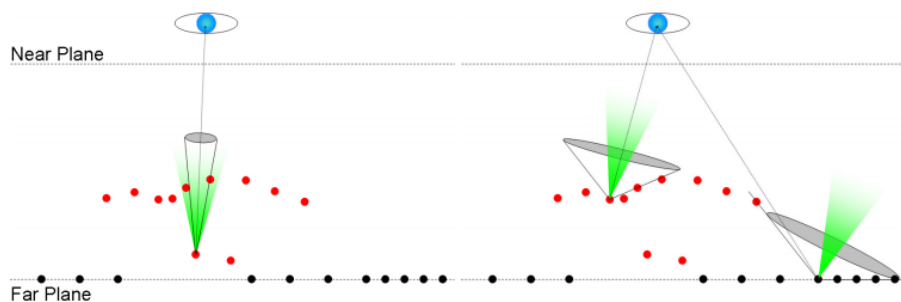


Figure 3.3: Left: example of a cone that is smaller than the opening threshold (in green). The point is considered non-visible. Right: example of visible points. The cones are wider than the opening threshold, and the axis formed by the point of view and each point intersect the cones.

In the next sections, we propose an automatic screen-space method for estimating the visibility of points in a point cloud given a viewpoint. This method makes no assumptions on the sampling or the density of the point cloud and can therefore be performed on any point cloud.

3.3 Visibility estimation method

The first contribution of this chapter is a method for estimating visibility in a 3D point cloud that is robust to high sampling variations.

As illustrated in Figure 3.4, points from two objects located at different distances from the given viewpoint overlap in the image plane once projected. In this context, a point is visible only if it lies on the closest object and is occluded otherwise. From this observation, we propose an algorithm that considers the neighborhood of a point in screen-space in order to estimate whether this point lies on the closest object or not. The algorithm consists in 4-steps detailed hereafter.

Projection to screen-space Let \mathcal{P} be a 3D point cloud, and Φ a viewpoint such that any 3D point

$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathcal{P}$ can be projected as a point $p = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathcal{P}_\Phi$ in the image plane of the

viewpoint Φ . The relation between P and p is illustrated on Figure 3.4. We also define d_p as the depth of the point p . It corresponds to the 3D Euclidean distance between P and the center of the viewpoint as illustrated in Figure 3.4(a).

Neighbors computation We define $\mathcal{N}(p)$ as the set of the N nearest neighbor points of p in the image plane as explained in Figure 3.5(b). The $\mathcal{N}(p)$ set can be computed using any K-NN algorithm with a Euclidean distance. The use of the K-NN algorithm defined in Friedman et al. (1977) ensures logarithmic computation time while being parallelizable.

Visibility estimation For each point, we want to determine if it lies on the object in its neighborhood that is the closest to the viewpoint. If so, we can consider it as visible. To that end, we compare its position to the closest and the farthest point of its neighborhood. We define the visibility of each point as follows:

$$\alpha_p = e^{-\frac{(d_p - d_p^{min})^2}{(d_p^{max} - d_p^{min})^2}} \quad (3.1)$$

where

$$d_p^{min} = \min_{q \in \mathcal{N}(p)} d_q, d_p^{max} = \max_{q \in \mathcal{N}(p)} d_q$$

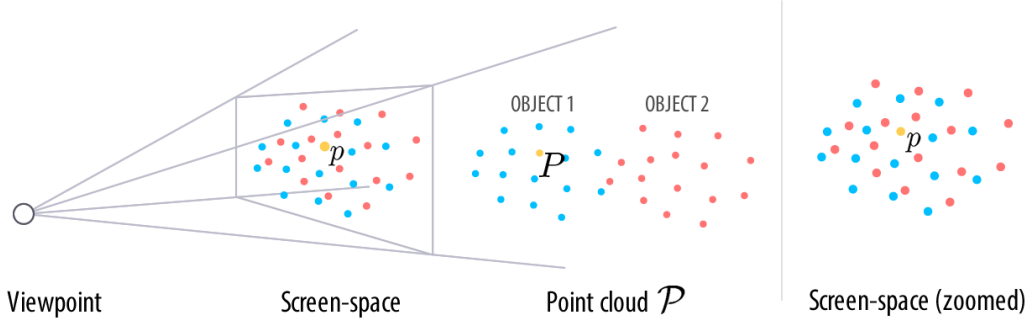


Figure 3.4: Illustration of notations in 3D and in the screen-space. Blue points corresponds to the points lying on the closest object while red points lies on the farthest object. Although the blue points and red points are well separated in 3D, they overlap in screen-space.

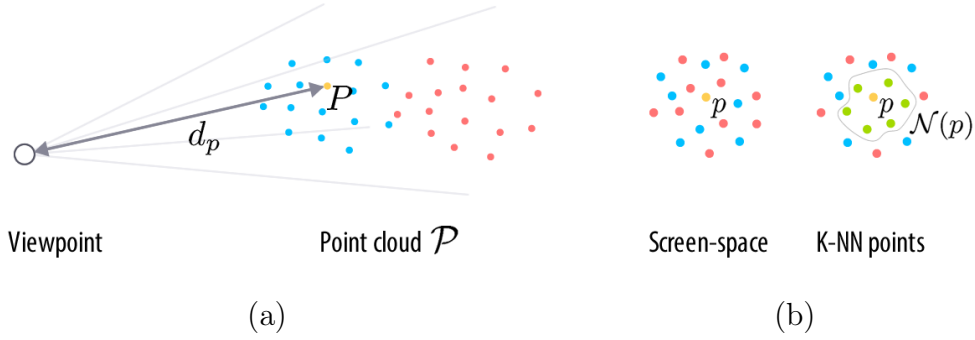


Figure 3.5: Illustration of the depth and of the closest points. (a) d_p corresponds to the depth between P and the center of the viewpoint, (b) shows an example of the N closest points in screen-space ($N = 6$).

The visibility estimation of each point $p \in \mathcal{P}_\Phi$ is now given by $\alpha_p \in [0, 1]$, where $\alpha_p = 0$ means that p is occluded and $\alpha_p = 1$ means that p is surely visible.

Binarization The visibility of a point cloud being a binary notion, we propose the following binarization of α_p :

$$\hat{\alpha}_p = \begin{cases} 1 & \text{if } \alpha_p \geq \bar{\alpha} \\ 0 & \text{otherwise.} \end{cases} \quad (3.2)$$

with $\bar{\alpha} = \frac{1}{\text{Card}(\mathcal{P}_\Phi)} \sum_{p \in \mathcal{P}_\Phi} \alpha_p$ the mean of the estimated visibilities. Note that various $\bar{\alpha}$ values have been tested such as $\bar{\alpha} = 0.5$ or the median value of the estimated visibilities as discussed in Section 3.5. However, in our experiments on LiDAR data, the mean value remains the best threshold. When point clouds have constant densities, $\bar{\alpha} = 0.99$ appears to be more adequate.

3.4 Visibility estimation dataset for LiDAR point clouds

The evaluation of visibility estimation techniques has mostly been done either by visual analysis or by comparison to degraded synthetic models. In (Shalom et al., 2010; Katz et al., 2007), visual results are displayed to show the qualitative performances of each algorithm. In (Mehra et al., 2010), small degradations on synthetic model are applied in order to build groundtruths. Although these methods of evaluation provide convincing results, they do not provide complete and objective quantitative measures on real data. Real data, such as LiDAR, differ from synthetic data in two aspects. The first difference is that the point cloud density is highly variable on real data depending on the distance to the sensor, while constant on synthetic data. The second one is that real urban data only acquire partial representations of each object of the scene as the sensor does not see objects from every possible viewpoints. On the other hand, the synthetic data presented in the related works (Shalom et al., 2010; Katz et al., 2007; Mehra et al., 2010) are always complete 3D objects, which can be seen from any viewpoint. To our knowledge, we proposed the first annotated dataset on real urban LiDAR data, which makes the second contribution of this chapter.

3.4.1 Overview of the dataset

We propose a manually annotated dataset containing over a 1 million points with the label 1 or 0 depending on if the points are visible or not. This dataset has been obtained by manually labeling 3 point clouds acquired by the RobotCar system (Maddern et al., 2017) at different locations, in urban environment. Two of these point clouds are acquired several meters from one another in order to test the stability of visibility estimation methods. The third point cloud corresponds to another location and covers a much wider area which enables testing the limit of methods in case of large distances ($> 100\text{m}$).

Annotations were done manually by comparing the projections of the point clouds to the optical images acquired at the same viewpoints. Figure 3.6 presents an overview of the produced dataset. The first row illustrates each scene as acquired from the optical sensor at each viewpoint. The second row shows the projections of each point cloud in the image domain (with the calibration matrices provided by the Robotcar dataset), where occluded points are highlighted in red. Finally, the third row shows a 3D visualization of each point cloud, with same color code than above. It illustrates the amount of points to be processed as well as the size of the scenes. The statistics of the dataset are summed up in Table 3.1. The dataset proposes different levels of visible / occluded points, as well as different size of the scene.

This dataset is publicly available online¹. The archive contains 3 text files in the

¹<http://www.labri.fr/perso/pbiasutt/Visibility/>

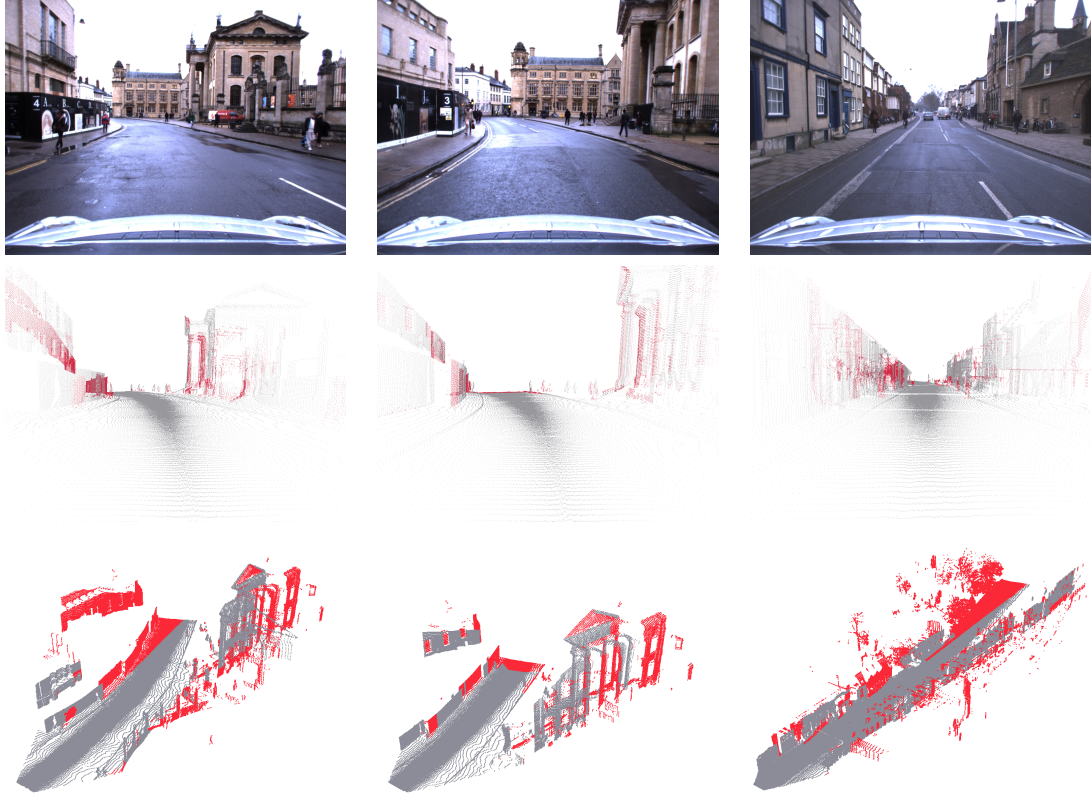


Figure 3.6: Overview of the proposed dataset. First row: optical image corresponding to each viewpoint. Second row: point cloud once projected in the image domain, where red pixels correspond to occluded points. Third row: 3D visualization of each point cloud, with the same color code than in the second row.

.xyz format that correspond to each point cloud. In each file, a line corresponds to $[x, y, z, x_\Phi, y_\Phi, label]$ where x, y, z are the 3D coordinates of the point, x_Φ, y_Φ are the 2D coordinates of the point when projected into Φ and *label* is the visibility label (0 for occluded points, 1 for visible points). To ensure good understanding of each of the 3 scenes, we also provide the optical RGB image of size 1280×960 px associated to each viewpoint.

Table 3.1: Content of the visibility estimation dataset

	Points	Visibility	Farthest point	Size
Scene #1	337384	55.5%	75.8m	20.6Mb
Scene #2	247682	57.0%	54.3m	15.1Mb
Scene #3	463531	65.9%	179.2m	28.3Mb
Total	1048597	59.46%	-	64Mb

3.5 Experiments & Results

In order to evaluate the performances of our visibility estimation method, we first perform a full numerical and visual comparison between our method and other state-of-the-art methods on 1) the proposed visibility dataset, 2) a point cloud with constant density. Next, we show application of our method to data fusion by performing point cloud colorization from RGB images. All the algorithms are run on Matlab 2018a with a 3.5Ghz CPU.

3.5.1 Evaluation on the Visibility Estimation Dataset

Using our new annotated dataset, we propose an evaluation with two state-of-the-art methods, and with our proposed model, against a groundtruth. For each method, we set all the parameters to their optimal values (*e.g.* the parameters that give best results against the groundtruth). In our case, we set $N = 75$ and we detail results for different $\bar{\alpha}$ values. We measure the efficiency of each method by computing the following metric:

$$S(\mathcal{P}) = \frac{1}{\text{Card}(\mathcal{P})} \sum_{P \in \mathcal{P}} \alpha_p \times \text{GT}_p \quad (3.3)$$

where GT_p corresponds to the annotation of the point P (0 or 1, occluded or visible respectively). This metric aims at capturing the percentage of correctly labeled points provided by each method. The results of this evaluation are displayed in Table 3.2.

Table 3.2 demonstrates that our algorithm outperforms each compared methods for the 3 scenes. The best scores are obtained by setting the threshold $\bar{\alpha}$ equal to the mean of visibility estimations for our method. This observation is explainable. Indeed, as mentioned above, LiDAR acquisitions only capture pieces of the scene. Thus, objects are represented by one of their face only which makes them well separated from one another. In this sense, when an object overlaps another in screen-space, the mean of the visibility estimations usually represents the visibility of a point that would be in between those two objects. If a point has a visibility estimation above this threshold, it is likely to fall on the closest object, otherwise,

Table 3.2: Comparison of the scores of two state-of-the-art and our visibility estimation methods on our visibility estimation dataset

Threshold	HPR	Cone	Ours	Ours	Ours
	Katz et al. (2007) optimal	Pintus et al. (2011) optimal	$\bar{\alpha} = 0.5$	α_p median	α_p mean
POV #1	74.09%	68.76%	90.15%	86.35%	90.96%
POV #2	69.09%	61.68%	86.95%	86.78%	88.39%
POV #3	81.55%	75.58%	82.21%	76.35%	83.75%
Average	74.91%	68.67%	86.43%	83.16%	87.70%
Total time	7.82s	1.53s	0.91s	1.03s	0.91s

it is occluded. Therefore, the mean value can be used when working on LiDAR point clouds because of the way objects are separated from one another, making the method fully automatic in this context.

We also demonstrate that our method operates faster than any other tested method with the ability of treating the whole dataset in less than a second. Moreover, the code is run on a single CPU. Among the 4 steps of the algorithm, the computation of the K-NN is the most time consuming (about 86% of the total running time). Therefore, one can expect much faster running times by operating on GPU with parallel implementation of the K-NN algorithm.

The problem of visibility estimation is a classification problem with two classes: visible and occluded points. Therefore, we enrich our evaluation by computing typical classification metrics for each method and we display them in Table 3.3. For each metric, the best scores are obtained using our method. In particular, our method with $\bar{\alpha} = 0.5$ maximizes the true-positives and minimizes the false-negatives. On the opposite, our method with $\bar{\alpha}$ as the median of the estimations maximizes the false-positives and true-positives. Once again, using our method with $\bar{\alpha}$ as the mean of the estimations provides a good tradeoff between true-positives/true-negatives and false-positives/false-negatives. The methods of Katz et al. (2007) and Pintus et al. (2011) tend to over-estimate the visibility of each point, resulting in many occluded points being labeled as visible. This is expressed by the very high percentage of false-positives. We computed accuracy and the F1-score of each method against the ground truth. For both criterions, our method with $\bar{\alpha}$ as the mean of the estimation achieves, once again, the best results.

For the task of data-fusion, it is often preferable to discard the maximum of occluded points (Bevilacqua et al., 2017). Therefore, the number of false-positives has to be kept as low as possible. In this sense, our method provides very satisfactory results, especially using the $\bar{\alpha}$ as the mean of estimations when working on LiDAR data.

We conclude this evaluation on LiDAR data by a visual analysis of the results of the different methods. Figure 3.7 shows the results of the visibility estimations visualized in 3D. For each result, the dark cone in the bottom left corner represents

3. Visibility estimation of a point cloud from a given point of view

Table 3.3: Comparison of the different methods for point cloud visibility classification

	HPR Katz et al. (2007)	Cone Pintus et al. (2011)	Ours $\bar{\alpha} = 0.5$	Ours α_p median	Ours α_p mean
Threshold	optimal	optimal			
True-positive	89.54%	85.16%	95.45%	78.31%	88.23%
False-positive	18.84%	17.78%	10.78%	3.66%	5.15%
False-negative	6.26%	8.61%	2.79%	13.18%	7.14%
True-negative	54.47%	56.24%	72.45%	90.80%	86.93%
Accuracy	85.16%	84.27%	92.52%	90.94%	93.44%
F1-score	87.71%	86.59%	93.37%	90.29%	93.49%

the viewpoint. Figure 3.7(a) shows the point cloud colorized with the depth toward the viewpoint (cold colors for close points, hot colors for far points). Figure 3.7(b) shows the annotated groundtruth for this scene, where red points are points that are visible from the viewpoint and dark points are supposed to be occluded. Figure 3.7(c) and 3.7(d) are the results of HPR (Katz et al., 2007) and our method (with $\bar{\alpha}$ set as the mean of estimations) respectively. We can see that HPR estimates too many points as visible points, especially on the closest points. On the opposite, our method succeeds in discarding occluded points, and provides a result that is very close to the groundtruth.

We also illustrate these results as seen from the associated viewpoint in Figure 3.8. For better understanding purpose, Figure 3.8(a) shows an image acquired from the same viewpoint. In Figure 3.8(b), we only display visible points of the groundtruth. Figure 3.8(c) and 3.8(d) shows the results of HPR and our method respectively. We can see once again that HPR labels too many occluded points as visible and it fails to distinguish foreground from background objects. This is mostly because this scene presents very high variations of density. In particular, the center of the road concentrates a very high density of points as the sensor is close from the road. Therefore, the convex-hull has to be relaxed enough to fit this region of the point cloud, which leads to visual aberrations on regions with lower density. Our method succeeds to obtain better results in this scene, which demonstrates its robustness against high density variations.

3.5.2 Evaluation on constant density point cloud

In previous section, we demonstrated that our method performs better than other methods for point clouds with high density variations. In this section, we aim at showing that our method remains competitive on constant density point clouds. The Stanford Bunny model is a point cloud (from the Stanford University CG Laboratory) that was created by merging 10 depth acquisitions of a real object and equalizing the density of the fused point cloud. The final point cloud is composed of 31655 points. As each depth acquisition only acquires points that are visible from a

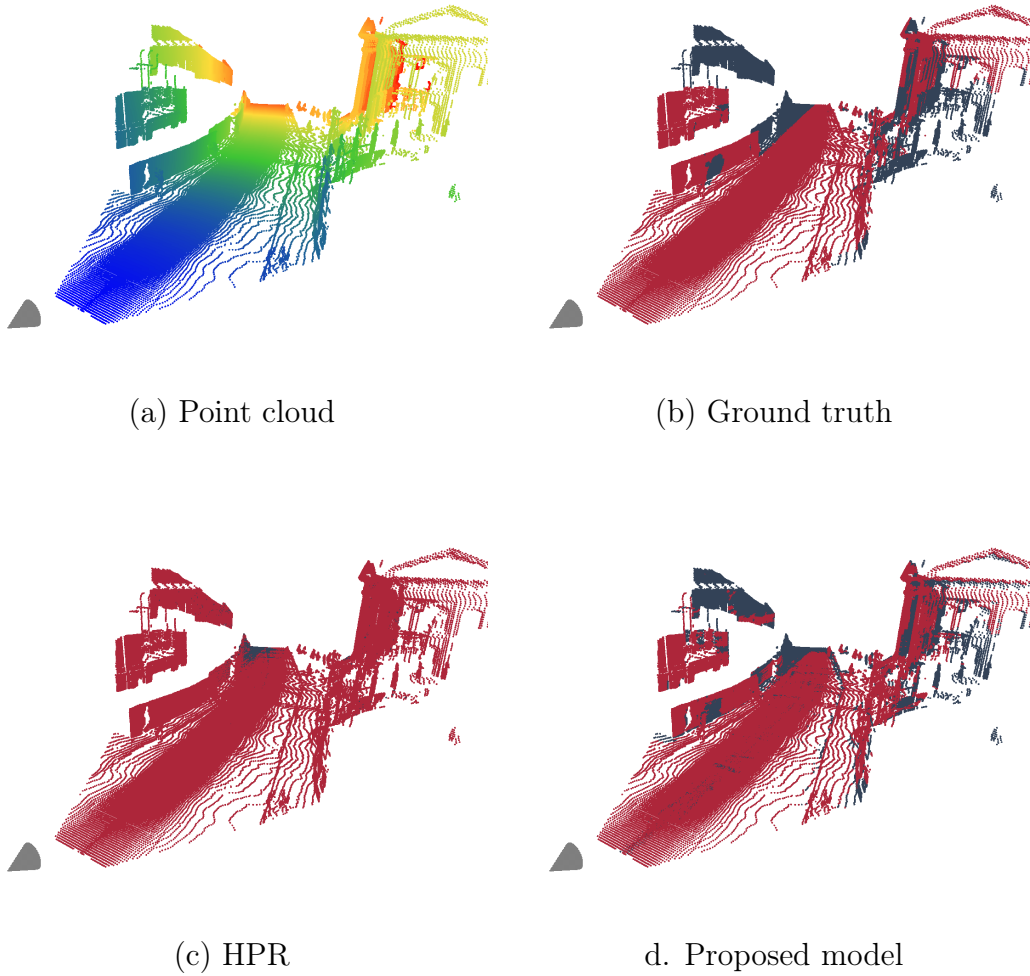


Figure 3.7: Results of visibility estimation on the first scene of our visibility estimation dataset. (a) the point cloud where the heat of the color is proportional to the depth, (b) is the annotated point cloud (red: visible, grey: non-visible), (c) HPR result and (d) our result. The result brought by HPR estimates too many visible points, whereas our method provides a result that is very close to the groundtruth.

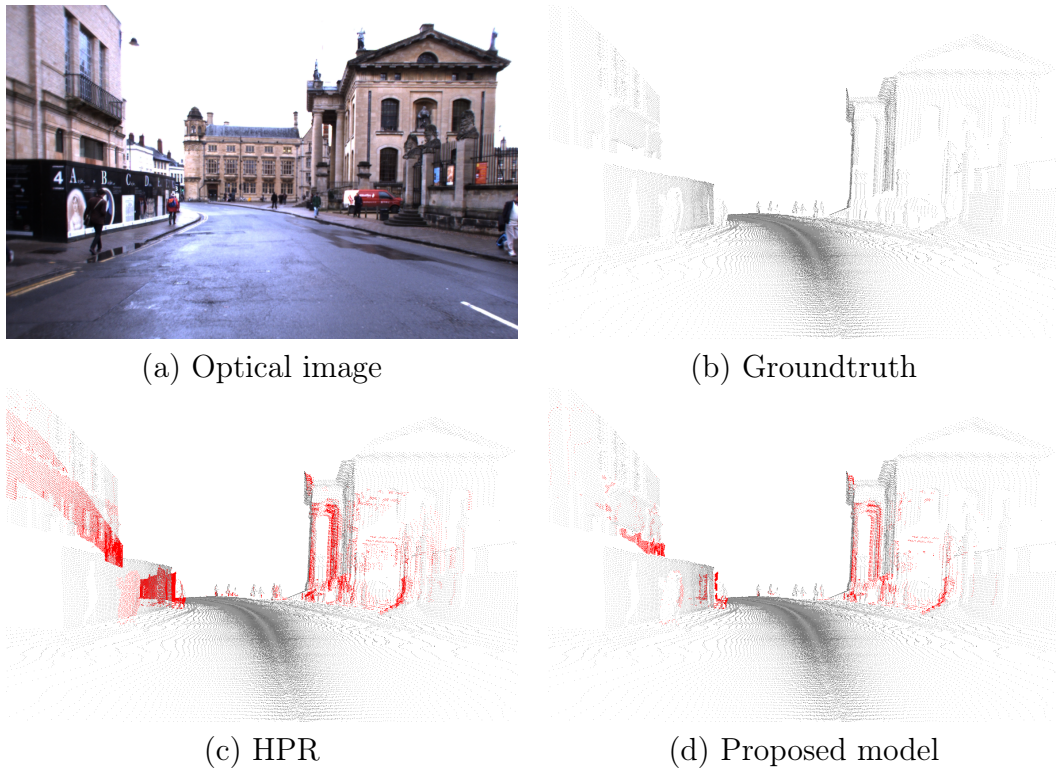


Figure 3.8: Results of the visibility estimations on the first scene of the dataset in screen-space. (a) the optical image associated with the point of view, (b) visible points with respect to our annotation, (c) HPR result and (d) ours. Red points in (c) and (d) correspond to misestimated points.

Table 3.4: Comparison of the scores of the different methods on constant density point cloud

Threshold	HPR optimal	Cone optimal	Ours $\bar{\alpha} = 0.99$	Ours α_p mean
Score (Eq. (3.3))	96.57%	93.75%	95.23%	93.02%
True-positive	95.17%	88.63%	94.44%	98.07%
False-positive	1.15%	0.88%	2.14%	6.07%
False-negative	2.28%	5.37%	2.63%	0.91%
True-negative	97.82%	98.33%	95.95%	88.50%
Accuracy	98.25%	96.76%	97.56%	96.40%
F1-score	98.23%	96.59%	97.54%	96.57%

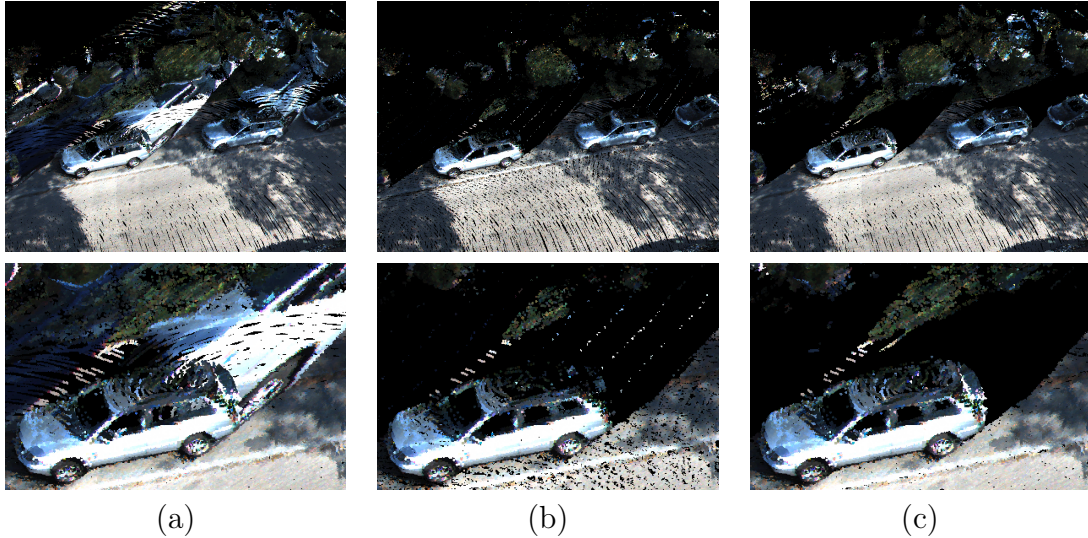


Figure 3.9: Comparison of the colorization of point clouds using RGB images. (a) colorization without any visibility information, (b) colorization with HPR [Katz et al. \(2007\)](#) for the visibility estimation, (c) colorization with our visibility estimation method. The result provided by our method presents no artifacts on occluded areas, especially behind cars compared to the two other results.

single viewpoint, we created a groundtruth by comparing the final point cloud to the points that were acquired at a certain viewpoint. Criterion (3.3) and classification metrics have been computed for our method and state-of-the-art methods. Results are displayed in Table 3.4.

Here, the point cloud is of constant density and it represents a very smooth object as illustrated in Figure 3.10. This is a scenario that is perfectly adequate for the HPR algorithm, which shortly outperforms the two other methods. Our method is outperformed only by about 1 percent but it still remains very efficient on these types

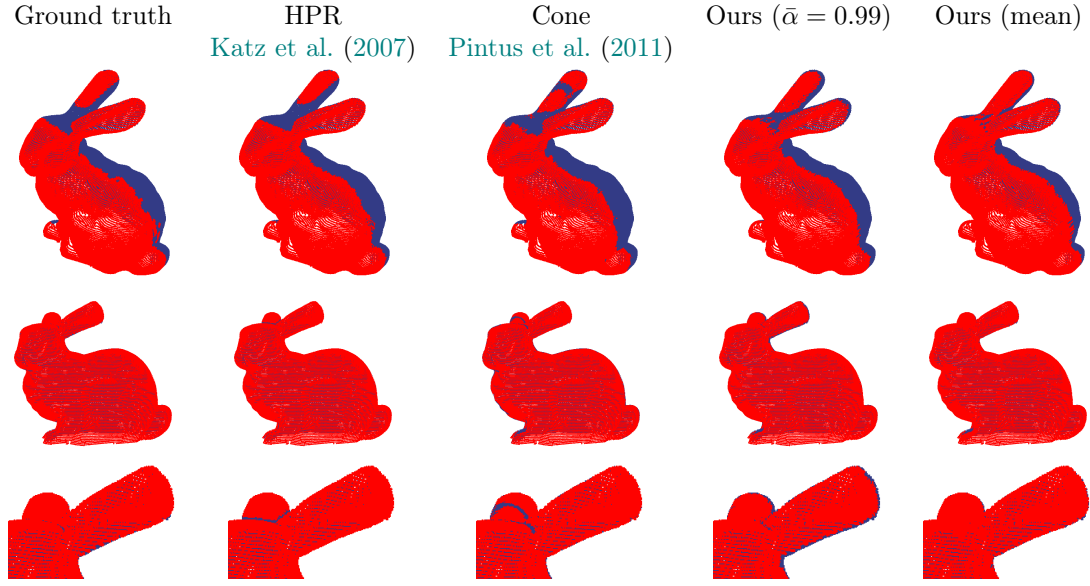


Figure 3.10: Visual comparison of the visibility estimation from different methods on a point cloud with constant and high density. Each column corresponds to one method. Rows are respectively: the results in 3D, the results in 2D (seen from the viewpoint), and a zoom of the 2D result focused on the ear region. The 3 methods succeed very well in estimating the visibility. Our method misses some points that are tangent to the viewpoint, as it can be seen on the last row, but still succeeds to correctly estimate the visibility of the remaining points.

of data. Compared to the two other methods, our method fails on tangential points that are located at the boundaries of the projection of the object as it is presented on the last row of Figure 3.10. This is mostly because on tangential points, the neighborhood covers only a small area, thus the difference between foreground and background is thus hard to set. These artifacts are limited when using the mean of estimation as visibility threshold, but it increases false-positives. Table 3.4 also illustrates the classification metrics. We can see that all tested methods reach very good levels of accuracy and F1-score. Our method succeeds better when $\bar{\alpha} = 0.99$ than when using the mean value. Indeed, for complete objects, there is no separation between foreground object and background object as was the case for LiDAR point clouds. Only points with high likelihood should be kept to improve results, which justifies $\bar{\alpha} = 0.99$.

Finally, Table 3.4 assesses that all methods limit the appearance of false-positives while ensuring to gather as many visible points as possible. To that end, HPR and our method succeed the best true-positive/false-positive ratio, which is ideal for data-fusion purposes, as discussed in next section.

3.5.3 Example of application to data fusion

To conclude our experiments, we show the interest of our visibility estimation for the task of data fusion. Using the KITTI dataset, we aim at colorizing a 3D LiDAR point cloud acquired in a street using only RGB images. Each point is projected in the image domain of the closest image (*e.g.* the image that was acquired at the closest position from the point). The point then takes the color of the pixel it projects onto only if it is considered visible. Figure 3.9 presents the result of the colorization on a point cloud composed of 3289533 points, and it is colorized using 40 RGB images. Figure 3.9(a) shows the colorization result where all points are considered visible. We can see that artifacts appear as the colors do not match the objects. This is particularly noticeable behind cars where the ground points take the color of the car. Figure 3.9(b) displays the colorization result where the visibility is estimated using HPR. There, some artifacts appear behind cars as the convex-hull goes through the glasses of the car. Moreover, this method discards many visible points on the ground and behind cars compared to our method (about 17% less points are colorized). Figure 3.9(c) presents the colorization result using our visibility estimation method. The artifacts behind cars have completely disappeared, while keeping most of the visible points. Finally, due to the number of points, visibility estimation using HPR for each viewpoint takes an average of 10.9 seconds whereas our method processes the point cloud at each viewpoint in about 1.2 seconds.

3.6 Conclusion

In this chapter, we have proposed a novel method for visibility estimation in a point cloud. Compared to other methods from the literature, this method is very robust to high variations of density. By considering the closest neighbors of each point in screen-space, we defined a criterion in order to automatically determine the visibility of each point. We have also proposed a new annotated dataset for testing the efficiency of point cloud visibility algorithms on real LiDAR urban data. This dataset is composed of over a million of manually annotated points. Finally we have compared our method to the state-of-the-art. We have validated that our method significantly outperforms existing methods on real urban data. Although our method was specifically designed for the estimation of visibility on point cloud with various density (such as LiDAR point clouds), we have also demonstrated that it still remains competitive on point clouds with constant density. This work was published in (Biasutti et al., 2019d).

Conclusion of the first part

This part of the thesis details how image processing techniques can be used to produce high resolution products using MMS data. To that end, we have proposed to consider a sparse projection of the point cloud onto a 2D pixel grid instead of directly operating on the raw point cloud.

Orthoimages A first method was proposed for the generation of subcentimetric orthoimages from LiDAR point cloud. The proposed method offers to project ground points onto a 2D-pixel grid, producing a sparse orthoimage. Empty pixels that come from the undersampling of the acquisition are then filled using a PDE-based method whereas large holes that correspond to non permanent objects are filled with a patch-based inpainting method. This framework succeeds in the generation of high-resolution orthoimages that can be used to satisfy novel European regulations.

This method strongly relies on the estimation of ground points, which assumes that the ground is flat in the area close to the acquisition vehicle. This is relevant as the computation of a large area is often decomposed in the computation of smaller areas, called tiles. However, if a very large area is considered, such assumption might be erroneous. Thus, more complex ground extraction method could be used to increase the robustness of the model, such as the deep-learning approach presented in (Velas et al., 2018). Moreover, filling large occlusions in a very large orthoimage might drastically increase the computational time. Thus, a good speed-up could be obtained by using the method proposed in (Barnes et al., 2009) with the same metric as presented in Section 1.5.

RGB-D images The task of RGB-D image generation from MMS data was also investigated. To that end, we proposed a novel variational approach to densify the sparse projection of a LiDAR point cloud in an optical image domain. The proposed energy functional is able to successfully densify the projection while taking visibility ambiguities into account.

However, because the energy functional is composed of many terms, its optimization requires many iterations. We believe that the visibility term could be replaced by a visibility estimation preprocessing step based on the method presented in Chapter 3. Another track of improvement could be about taking moving into account during the diffusion process. Indeed, because of the way LiDAR sensors and optical image perform acquisition, there might be a time delay between the acquisition of the optical image and the acquisition of the LiDAR points. This delay might decorrelate both modalities, leading to artifacts in the output of the method.

Visibility We have concluded this part by presenting a method to estimate visible points of a point cloud with variable density in screen-space. First, the point cloud is projected into an image domain. After that, the 2D K-NN are computed for each projected point. Then, we proposed a criterion that estimates whether a point is visible or not depending on its neighbors in screen-space. The proposed method outperforms existing methods on point clouds with variable density while remaining competitive on synthetic homogeneous point clouds.

The evaluation of this method could be extended to photogrammetric point clouds (*i.e.* points clouds that are constructed by the analysis of many). Indeed, when building these point clouds, the color of the images is projected onto each point. Therefore, the visibility of such point clouds can be directly estimated by comparing the color carried by each point to the value of the pixel it projects into given a viewpoint. Moreover, we would like to investigate other application of the proposed method, as discussed in Chapter 8.5.

Part II

Image processing on 3D LiDAR
point clouds in sensor topology

Summary

In the first part, we have shown how image processing methods can be used to process LiDAR point clouds in specific applications by working on projections of the point clouds. However, each proposed model systematically had to deal with the sparsity of the projections, often requiring an expensive densification step before any other treatment. This densification sometimes also requires a coupling with another modality, mostly optical images. In this case, we assume that both the other modality and the projection are perfectly aligned. Although this is a valid assumption if the calibration of the system is precisely known, it is often not the case. Accurate MMS calibration is a very complicated task as the calibration is exposed to external conditions during the acquisition, which might alter the original settings.

To overcome the problem of projection densification, we investigate how the acquisition pattern of common LiDAR sensors can be used to produce a new type of 2D image that represents the point cloud. Indeed, modern LiDAR sensors follow strict and regular sampling patterns that bring structure to the acquisition. This structure can thus be used to automatically derive a 2D image from a 3D point cloud. Such a 2D image, also named range-image, can be used in many applications while simplifying the formulation of each problem.

To that extent, a full framework for 3D LiDAR point cloud to optical image alignment is proposed. This framework uses a range-image to instantly reconstruct the mesh of the point cloud. Then, a rendering of the mesh in the optical image domain is aligned with the optical image itself using a variational approach.

We also offer to investigate point cloud segmentation using a range-image. First, a region segmentation method is proposed. It is based on a-contrario histogram segmentation that enables online segmentation of massive point clouds. After that, we propose a semantic segmentation approach that uses a convolutional neural network on range-images. This method is more specifically designed for autonomous systems.

The segmentation of an object in the point cloud can be used as a mask in the range-image. We propose a method for efficiently removing these objects from the point cloud. The proposed method takes advantage of the extensive literature of image inpainting to propose an efficient variational approach that operates on range-image to remove objects from the point cloud.

Finally, we conclude this part by showing how this representation can also be used to perform 3D detection on a LiDAR scene by extending an existing model originally introduced for 2D detection on RGB images.

Content

- Chapter 4 introduces the range-image.
- Chapter 5 presents the problem of multi-modal alignment and proposes a novel

framework for LiDAR to optical image alignment. The framework first intends to reconstruct the mesh of the point cloud using the range-image. Then, a rendering of the mesh in the image domain is aligned with the image using a variational approach.

- Chapter 6 details the problem of both point cloud segmentation and point cloud semantic segmentation. A method for each problem that takes advantage of range-images is then presented.
- Chapter 7 investigates the problem of object removal in 3D point clouds and depth reconstruction. A method is proposed that benefits from the 2D image inpainting literature to operate on range-images.
- Chapter 8 demonstrates how a 2D object detection method can be adapted to perform 3D detection in a point cloud by using range-images.

Publications related to this part

Peer reviewed international journal (1)

- [1] *Range-image: incorporating sensor topology for LiDAR point clouds processing*
Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Photogrammetric Engineering & Remote Sensing (PERS), 2018, 84 (6), pp. 367–375

Peer reviewed international conference proceedings (1)

- [2] *Disocclusion of 3D LiDAR point clouds using range-images*
Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Annals of Photogrammetry and Remote Sensing (ISPRS), 2017, pp. 75-82

Peer reviewed national conference proceedings (1)

- [3] *Désocclusion de nuage de points LiDAR en topologie capteur*
Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Groupe de Recherche en Traitement du Signal et de l'Image (GRETSI), 2017
- [4] *Détection et localisation d'objets 3D par apprentissage profond en topologie capteur*
Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Groupe de Recherche en Traitement du Signal et de l'Image (GRETSI), 2019

Under review (2)

- [5] *Fast image and LiDAR alignment based on 3D rendering in sensor topology*
Pierre Biasutti, Jean-François Aujol, Mathieu Brédif and Aurélie Bugeau
Submitted to Pattern Recognition Letters, 2019
- [6] *RIU-Net: Embarrassingly simple semantic segmentation of 3D LiDAR point cloud*
Pierre Biasutti, Aurélie Bugeau, Jean-François Aujol and Mathieu Brédif
ArXiv preprint, 2019

Chapter 4

Dense 2D representation of a 3D LiDAR point cloud

Table of contents

4.1	Problem statement	87
4.2	Range-images derived from the sensor topology	87
4.2.1	Sensor topology	88
4.2.2	From sensor topology to range-image	88
4.3	Interest and applications	90

4.1 Problem statement

In the first part of this document, we have studied how 2D projections of 3D LiDAR point clouds can be used to create high resolution products, while simplifying the processings.

However, the projection of a point cloud on a high resolution pixel grid produces a sparse image, in which many pixels do not contain any information. Indeed, the 2D projection of the point cloud does not correspond to a bijection from each 3D point to a pixel. Since many image processing methods implicitly assume that the input image is dense, it is frequently needed to densify such projections such as in Chapter 1 and 2 or to retrieve neighbors of each isolated pixels, as in Chapter 3. These preprocessing steps are often not trivial and they often have an impact on the computational time.

In this chapter, we offer to investigate another way to represent 3D LiDAR point clouds by 2D maps that are intrinsically dense. These representations can be directly extracted from most recent LiDAR sensors, at almost no computational cost. Because this data is dense, it can be used to overcome most of the limitations brought by the projection of the 3D point cloud on a 2D pixel grid as illustrated in the next chapters of this part.

4.2 Range-images derived from the sensor topology

We aim at demonstrating that a simplified model of the point cloud can be directly derived from it using the intrinsic topology of the sensing pattern during acquisition. This section introduces the sensor topology and how it can be exploited on various kinds of sensors.

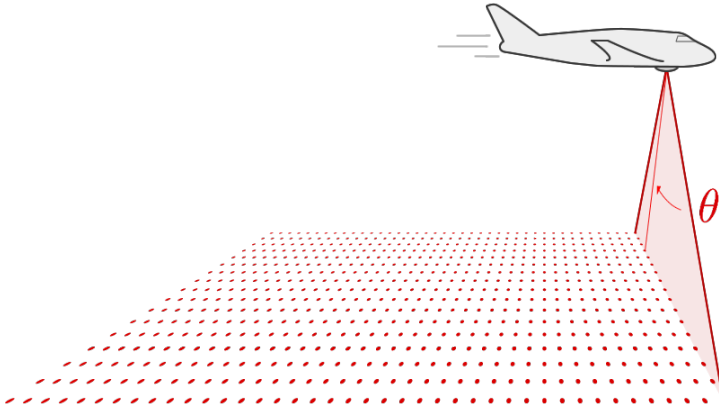


Figure 4.1: Example of the intrinsic topology of a 2D LiDAR sensor built on a plane

4.2.1 Sensor topology

Most modern LiDAR sensors offer an intrinsic 2D topology that can be accessed in raw acquisitions. However, this feature started to be considered in the literature only recently:

- for surface reconstruction in (Guinard and Vallet, 2018).
- for semantic segmentation in (Landrieu and Boussaha, 2019), (Wu et al., 2018), (Bichen et al., 2018) and (Yuan et al., 2018).
- as extra input data for 3D detection in (Chen et al., 2017a).
- for ground extraction in (Velas et al., 2018).
- for graph computation and pointcloud compression (Bletterer, 2018).

However, this property of the LiDAR sensors is often only partially presented. Namely, LiDAR points may obviously be ordered along scanlines, yielding the first dimension of the sensor topology, linking each LiDAR pulse to the immediately preceding and succeeding pulses within the same scanline. For most LiDAR devices, one can also order the consecutive scanlines. It amounts to considering a second dimension of the sensor topology across the scanlines as it can be seen in Figure 4.1.

4.2.2 From sensor topology to range-image

The sensor topology often varies with the type of LiDAR sensor that is being used. 2D LiDAR sensors (*i.e.*, featuring a single simultaneous scanline acquisition) such as the one used in (Paparoditis et al., 2012) generally send an almost constant number H of pulses per scanline (or per turn for 360 degree 2D LiDARs) where each pulse was emitted at a certain θ angle value. Therefore, any measurement of the sensor might be organized in an image of size $W \times H$, where W is the number of consecutive scanlines and thus a temporal dimension. This is illustrated in Figure 4.2 in which one can see how the 2D image is spanned by the sensor topology. In this thesis, such images are only built using the range measurement as pixel intensity, later referred to as range-images. Note that these range-images differ from typical range-images (Kinect, RGB-D) as the origin of acquisition is not the same for each pixel and the 3D directions of pixels are not regularly spaced along the image, but warped by the orientation changes of the sensor trajectory.

3D LiDAR sensors are based on multiple simultaneous scanline acquisitions (*e.g.* $H = 64$ fibers) such as in the MMS proposed in (Geiger et al., 2013). Again, each scanline contains the same number of points and each scanline may be stacked horizontally to form the same type of structure, as illustrated in Figure 4.3. Each point is defined by two angles and a depth, (θ, ϕ, d) respectively, with steps of $(\Delta\theta, \Delta\phi)$ between two consecutive positions. Each point p of the LiDAR point cloud can be mapped to the coordinates (x, y) with $x = \lfloor \frac{\theta}{\Delta\theta} \rfloor, y = \lfloor \frac{\phi}{\Delta\phi} \rfloor$ of a range-image. Note

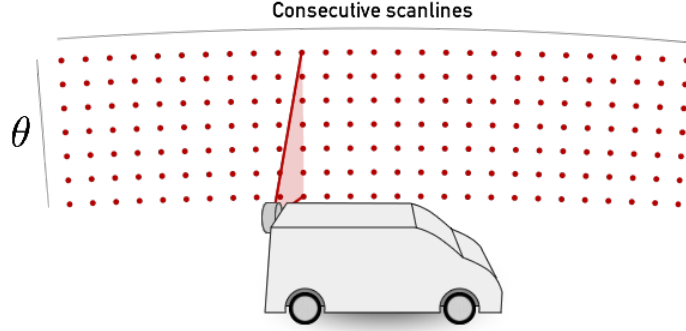


Figure 4.2: Example of 2D LiDAR sensor and the related topology

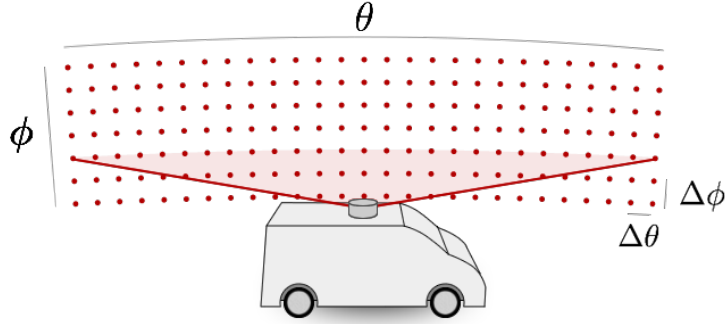


Figure 4.3: Example of 3D LiDAR sensor and the related topology

that Figures 4.2 and 4.3 are simplified for better understanding, but that realistic cases can be more chaotic as discussed later in this section.

Whereas LiDAR pulses are emitted somewhat regularly, many pulses yield no range measurements due, for instance, to reflective surfaces, absorption or absence of target objects (*e.g.* in the sky direction) or an ignored measurement whenever the measure is too uncertain. Therefore the sensor topology is only a relevant approximation for emitted pulses but not for echo returns, such that the range-image is sparse with undefined values where the sensor measured no echoes (or when further processing was performed on the acquisition, leading to the removal of points having a too incertain measurement). This is illustrated in Figure 4.4.b in which pulses with no echoes appear in dark. Note that considering multi-echo datasets as a multilayer depth image is beyond the scope of this thesis, which only considers first returns.

This 2D sensor topology encodes an implicit neighborhood between LiDAR measurement pulses. Whereas the implicit topology of pixels in optical images is supported by a regular geometry of rays (shared origin and regular grid of directions if geometric distortion is neglected), the proposed 2D sensor topology for LiDAR

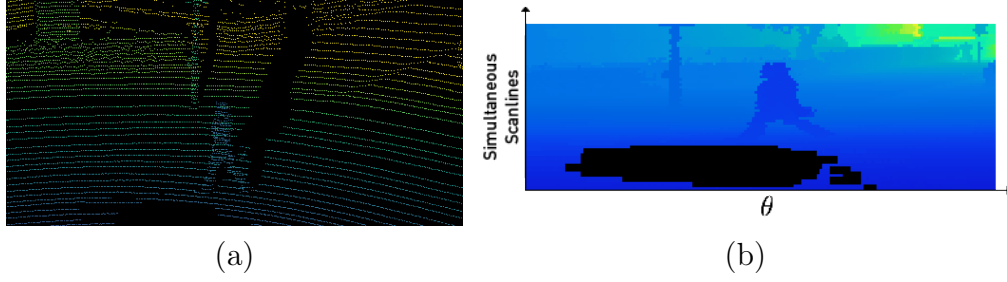


Figure 4.4: Example of a point cloud from the KITTI database (Geiger et al., 2013) (a) turned into a range-image (b). Note that the dark area in (b) corresponds to pulses with no returns.

point clouds is supported by the trajectory-warped geometry of 3D rays. However, it readily provides, with minimal effort, an approximation of the immediate 3D point neighborhoods, especially if the sensor moves or turns slowly compared to its sensing rate. We argue however that this approximation is sufficient for most purposes, as it has the additional advantage of providing pulse neighborhoods that are reasonably local both in terms of space and time. It is thus robust to misregistrations, and very efficient to handle (constant time access to neighbors). Moreover, as LiDAR sensor designs evolve to higher sampling rates within and/or across scanlines, the sensor topology will better approximate spatio-temporal neighborhoods, even in the case of mobile acquisitions.

We argue that most raw LiDAR datasets contain all the information (scanline ordering, pulses with no echo, number of points per turn...) to enable the access to a well-defined implicit sensor topology. However it sometimes occurs that the dataset received further processings (points were reordered or filtered, or pulses with no return were discarded) or that the sensor did not acquire neighboring points consecutively. Therefore, the sensor topology may then only be approximated using auxiliary point attributes (time, θ , fiber id...) and guesses about acquisition settings (*e.g.* guessing approximate Δtime or $\Delta\theta$ values between successive pulse emissions). Using this information, one can recreate the range-image by stacking points even if some points were discarded. Defining a grid-like topology is a good approximation if the number of pulses per scanline/per turn is close to an integer constant with relatively stable rotation offsets between pulses.

4.3 Interest and applications

The use of a range-image as a simplified representation of a point cloud directly brings spatial structure to the point cloud. Therefore, retrieving neighbors of a point, which was formerly done using advanced data structures (Muja and Lowe, 2014) or by computing geometrical neighbors in projection (Biasutti et al., 2019d), is now a trivial operation and is given without any ambiguities. Range-images have

also proved to be a very efficient data structure for simplified 2D representations of point clouds. Indeed, a typical 2D projection of a point cloud produces a sparse image in which most of the pixels are filled with no information. Moreover, to prevent two different pixels from falling in the same pixel when being projected, the dimension of the image is required to be very large. This is often a limitation for many computer vision and deep-learning methods. On the other hand, a range-image is a canonical representation of a point cloud in the sense that it only requires as many pixels as there are points in the point cloud to represent all the information. The theory and the interest of the range-image has been the object of a publication ([Biasutti et al., 2018](#)).

In the next chapters, we show that considering the range-image that corresponds to a point cloud supported by its implicit sensor topology, rather than the point cloud itself, enables the adaptation of many existing image processing approaches to LiDAR point cloud processing (*e.g.*: segmentation, semantic segmentation and disocclusion in Chapters [6](#) and [7](#)) or mutli-modal processing (*e.g.*: registration and detection in Chapters [5](#) and [8](#)), without any preprocessing step.

Chapter 5

Point cloud to image registration

Table of contents

5.1	Introduction	93
5.2	Mutli-modal alignment	94
5.2.1	Mutli-modal image registration	94
5.2.2	LiDAR to optical registration	95
5.3	Methodology	97
5.3.1	Fast mesh reconstruction in sensor topology	97
5.3.2	Depth to optical image alignment	99
5.4	Experiments and results	102
5.4.1	Quantitative analysis	102
5.4.2	Qualitative analysis	104
5.5	Conclusion	105

5.1 Introduction

As mentioned in previous chapters, Mobile Mapping Systems can be used on wide acquisition campaigns led in cities, on roads, on highways, resulting in the production of very large – multi-modal – datasets, thanks to sensors that acquire different aspects of the scene. However, due to the complexity of such acquisition systems, the calibration from one sensor to the other is often flawed. This can be caused by the instability of the sensors throughout a mobile acquisition, where the calibration slowly deteriorates while the system is being operated. Therefore, the different modalities are slightly misaligned which can compromise further processing requiring multi-modal data fusion.

For example, point cloud colorization can be achieved by projecting the color information of the optical image on the LiDAR point cloud. However, a slight misalignment can result in colors being projected on points that do not belong to the correct object, which is particularly visible on object’s silhouettes. Multi-modal object detection also requires a good alignment. Indeed, such methods usually feed both optical images and LiDAR point clouds to neural networks, or more generally to classification methods, in order to estimate the location of each object in the scene. Misalignment between both modalities might confuse the network as both data end up indicating different locations, resulting in bad performances.

Although it is possible to interactively reduce this misalignment by visually inspecting both data in the same domain, it is often practically infeasible as the datasets are typically composed of thousands of examples. The automatic alignment of LiDAR data to optical image is therefore a crucial issue.

The problem of LiDAR to image alignment raises several issues. First of all, direct comparison between the two modalities can only be done if they share common attributes (colors or reflectances). However, in many systems, each sensor solely acquires a specific aspect of the scene. For example, LiDAR sensors acquire the geometry of the scene whereas optical cameras acquire the visual information. Moreover, optical sensors and LiDAR sensors are located at different positions on the MMS, and they often operate differently. For example, optical cameras instantly acquire a single point of view, whereas 2D LiDAR sensors require the MMS to move in order to acquire the geometry of the scene. This implies that the different sensors do not acquire the scene from the same point of view, resulting in visual ambiguities. The correlation between both modalities is therefore irrelevant for some parts of each data as they have not been observed by the other sensor.

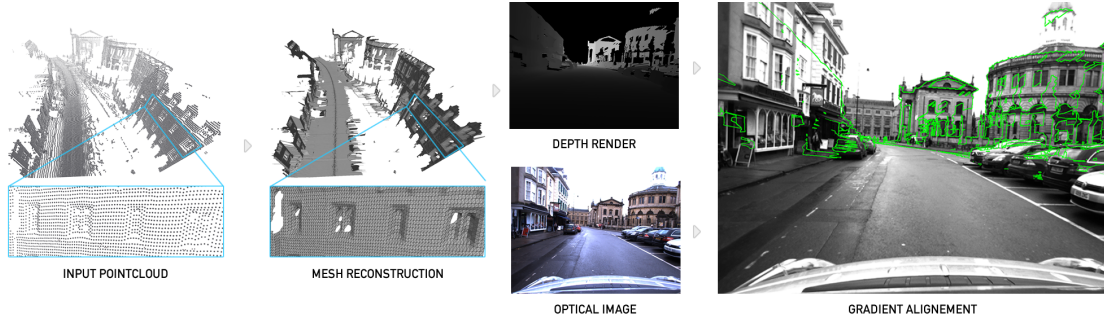


Figure 5.1: Scheme of the proposed framework.

5.2 Mutli-modal alignment

Multi-modal registration has been a subject of interest over the past decades. In this section, previous works on multi-modal registration as well as previous works on LiDAR to optical image are introduced.

5.2.1 Mutli-modal image registration

In computer vision, registration methods often consist in the detection and the matching of corresponding features from two different modalities. Feature points are extracted using common methods (SIFT (Lowe, 2004) or SURF (Bay et al., 2006)), or more specific adaptations (Mikolajczyk and Schmid, 2005; Rublee et al., 2011). These features are then matched using the RANSAC algorithm (Fischler and Bolles, 1981) to estimate the optimal transformation, as it can be seen in many biomedical imaging works (Allaire et al., 2008; Paganelli et al., 2012; Toews et al., 2013). However, these methods rely on strong similarities between each modalities which can be limited in a mutli-modal context. This problem can also be solved using variational approaches. In this case, the optimal alignment can be defined as the maximum of a given metric, typically Mutual Information (Viola and Wells III, 1997) or Cross-correlation (Roshni and Revathy, 2008), which aim at finding correlations between two distributions of intensities. These methods perform well as long as there exists a bijection between both modalities (*e.g.* between CT and MR images) which is not the case between 3D points and optical images. Another approach presented in (Sutour et al., 2015) aligns the gradients of both modalities, thus being agnostic to any correlation between the modalities. To that end, the authors assume that different modalities share some common strong gradients. The problem is formulated with the following energy function:

$$C(T) = \int_{\Omega} |\nabla u_1(T_{t_x, t_y, z}(X)) \cdot \nabla u_2(X)| dX \quad (5.1)$$

where u_1, u_2 are the two modalities and $T_{t_x, t_y, z}$ is the 2D homogeneous transformation matrix that should best align u_1 over u_2 . The proposed functional is not convex so

that both modalities are assumed to be intialized close to the optimal alignment. Although this method provides an effective solution to the problem of multi-modal fine alignment, it only estimates translation and scaling without rotation. Moreover, it implies that gradients can be computed on both modalities, which is not trivial when dealing with sparse projections of 3D points.

5.2.2 LiDAR to optical registration

The problem of LiDAR to optical registration can be divided into three main kinds of approaches: *2D feature-based*, *3D-based* and *statistical methods*.

2D feature-based methods aim at establishing correspondences between feature points of the optical image and the point cloud projected in the optical image domain.

In (Moussa et al., 2012), the authors propose a method that uses ASIFT features (Morel and Yu, 2009) to match a colorized point cloud with an optical image. Aberrant correspondences are then filtered out using RANSAC (Fischler and Bolles, 1981). The final 3D pose is estimated by solving a Perspective-n-Point problem (Lepetit et al., 2009) in which the 2D coordinates of feature points in the optical image is associated with the 3D locations of the corresponding feature points in the point cloud.

(González et al., 2009) propose a method for estimating the location of an optical image relatively to a 3D colorized point cloud of the same scene. The image is first enhanced to increase its contrasts. Then, the projection of the point cloud is manually resized in order to fit the optical image as well as possible. After that, correspondences are estimated by averaging cross-correlation and least square metrics. Finally, the 3D pose is retrieved using RANSAC. This method assumes that the original image and the point cloud are acquired at very close location otherwise the distortion brought by the resizing method would affect the correspondence finding step.

Although 2D feature-based methods provided straight forward ways to estimate the optimal alignment between optical image and point cloud, they typically rely on shared information between the two modalities. This can be a major drawback on light acquisition system where the LiDAR sensor only acquires 3D related data.

3D-based methods offer to align the 3D LiDAR point cloud with the 3D reconstruction of a set of optical images.

(Corsini et al., 2013) propose a two-step method for 3D-based point cloud to image alignment. First, a 3D sparse point cloud is reconstructed from a set of input optical images by using Structure From Motion (SFM) algorithm. The SFM algorithm is designed to find 2D correspondences in images of an input set of images

and to regress the 3D pose of each image as well as the 3D position of each feature point, producing a sparse point cloud. After that, the 4-points congruent set (Aiger et al., 2008) algorithm is used to align the sparse 3D point cloud with the 3D LiDAR point cloud. Later, (Abayowa et al., 2015) propose a similar method for aligning a 3D LiDAR point cloud with a set of aerial optical images. A dense 3D point cloud model is built from the set of optical images using the dense 3D reconstruction method described by (Furukawa and Ponce, 2010). Then, the pose of the dense point cloud is recovered by using Iterative Closest Point (ICP) (Besl and McKay, 1992) algorithm in order to minimize the distance error between the dense point cloud and the LiDAR point cloud. Although these methods achieve high quality results, they require a set of input optical images instead of a single image. Moreover, 3D registration methods are largely sensitive to missing data that often appear in real urban LiDAR data.

Statistical methods for point cloud to image registration try to define metrics that can be used to measure similarities between the two input modalities. Most of the time, the metric is computed in the 2D image domain. The work described in (Miled et al., 2016) proposes to align the sparse projection of a LiDAR point cloud with an optical image by comparing both modalities using Mutual Information (MI). This metric is used to find the dependency between the colors carried by the optical images and the reflectances brought by the LiDAR point cloud. The pose between the image and the point cloud is computed using a variational model that maximizes the MI metric between the two modalities. This method achieves very convincing results. However it strongly relies on the quality of the reflectances acquired by the LiDAR sensor. In practical use, only very few high quality LiDAR sensors can reach such levels of accuracy. Most common sensors acquire reflectance with high level of noise. Moreover, the reflectance is only relevant in certain scenarios and it cannot be used on wet surfaces or highly reflective surfaces for example. To overcome the problem of using reflectance, a method for the registration of a raw LiDAR point cloud with a single image is proposed in (Castorena et al., 2016). There, the authors propose to align the edges of the interpolated projection of a LiDAR point cloud with the edges of an optical image. However, the interpolation of the projection is only relevant in the case that the LiDAR sensor and the optical image share a close point of view. Otherwise a lot of ambiguities can arise from the LiDAR projection in the image domain which often leads to large errors in the calibration estimation. Later, (Guislain et al., 2017) proposed a method that aims at aligning only visible points of the LiDAR point cloud with the optical image. To do so, they first estimate the visible points given the optical image point of view using (Rubinstein et al., 2008), which was introduced in Chapter 3. The remaining points are used to produce a dense image of reflectances by performing bilinear inpainting. This dense reflectance image is aligned with the optical image using a metric that is less sensitive to missing data than Mutual Information. In the case when the reflectance is not available, they offer to compute the same metric on a dense normal map of the

visible points. This method achieves very good results when the visibility estimation performs well. This is the case when each different objects of the 3D scene are well separated. However, in the case of urban scenes, the amount of missing data as well as the heterogeneity of the shapes and object is very challenging for visibility estimation methods as shown in (Biasutti et al., 2019d). Therefore, the quality of the results on real urban data often lacks of accuracy.

In this chapter, we propose a novel method for LiDAR point cloud and optical image alignment that uses the topology of the LiDAR sensor to generate a dense image without any visibility ambiguities. This dense image is later aligned with the optical image using a variational model. An overview of the proposed approach is shown in Figure 5.1.

5.3 Methodology

In this section, we present each step of the proposed framework for point cloud to image registration. The proposed framework is highlighted in Figure 5.1: first, an image is created by rendering the triangulation based on the sensor topology of the point cloud. Then, this rendering is aligned with the optical image using a variational approach to align the gradients of both modalities.

5.3.1 Fast mesh reconstruction in sensor topology

The first step of the proposed framework consists in the reconstruction of the mesh of the point cloud. The problem of mesh reconstruction consists in linking points of a point cloud with triangles in order to approximate the surface of the objects in the scene. Surface reconstruction is traditionally done by smoothness approaches (Lipman et al., 2007; Xiong et al., 2014), primitive approximation (Schnabel et al., 2009; Lafarge and Alliez, 2013) or global regularity approaches (Li et al., 2011a,b; Monszpart et al., 2015). However, these methods are often computationally expensive. Moreover, they often require strong assumptions on the homogeneity of the point cloud, which is not suitable in the case of LiDAR acquisitions. To overcome these problems, we propose a very fast approach for mesh reconstruction that exploits sensor topology to instantly create a raw mesh from the point cloud. Note that more precise meshes can be reconstructed using the analogue method proposed in Guinard and Vallet (2018) but with a substantive impact on the computational time. However this work focuses on the efficiency and the performance of the final alignment between LiDAR point cloud and optical image. Thus, the use of the method proposed in Guinard and Vallet (2018) is out of the scope of this work, although it would be interesting to test.

The range-image representation (introduced Chapter 4) of the point cloud enables direct neighborhood computation: the set of neighbors of a given point can be

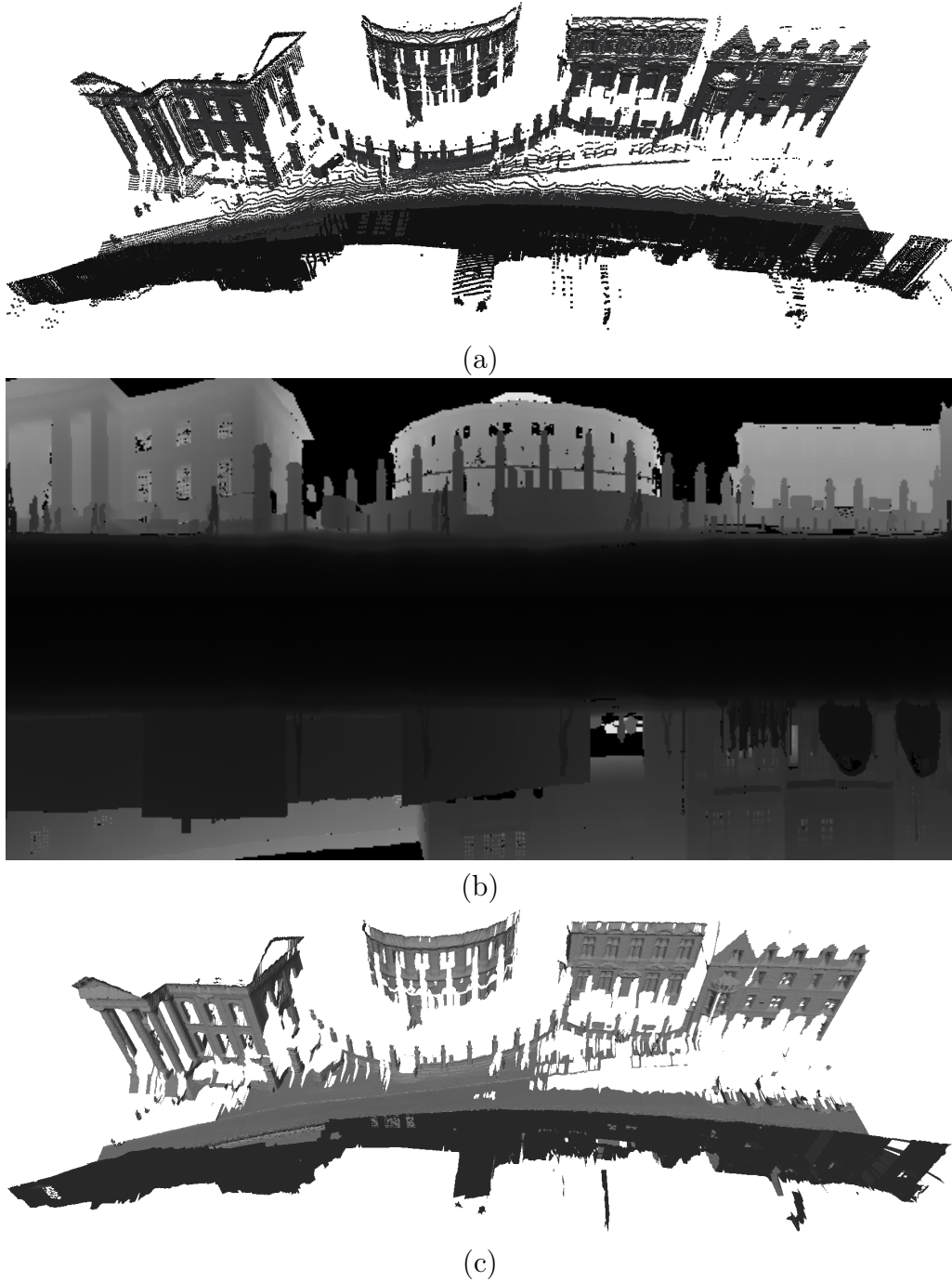


Figure 5.2: Mesh reconstruction scheme. (a) is the input point cloud, (b) the point cloud as seen in sensor topology and (c) the reconstructed mesh.

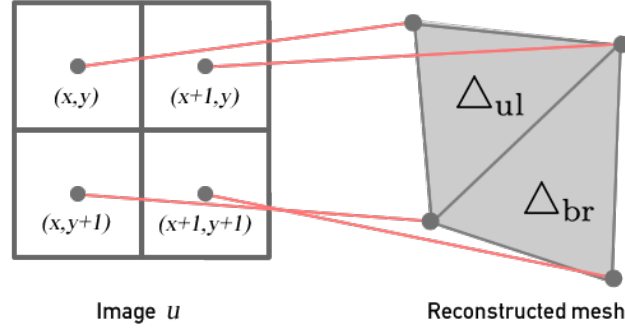


Figure 5.3: Triangle construction from image in sensor topology.

directly retrieved by checking the adjacent pixels of its projection in u . An example of the point cloud and its associated range-image are shown Figure 5.2.

For each pixel (x, y) of the range-image u , 2 triangles Δ_{ul}, Δ_{br} are created as follows:

$$\begin{aligned}\Delta_{ul} &= \{u(x, y), u(x + 1, y), u(x, y + 1)\} \\ \Delta_{br} &= \{u(x + 1, y), u(x + 1, y + 1), u(x, y + 1)\}\end{aligned}$$

This principle is illustrated on Figure 5.3. After that, triangles are filtered out by discarding the ones that have at least one edge that is longer than a certain threshold t , typically $t = 1.0\text{m}$. This step prevents separate objects from being connected together which enhances the overall quality of the mesh. An example of reconstructed mesh is showed in Figure 5.2(c). Finally, the mesh is being rendered from the optical camera location, with the same intrinsic parameters. This produces a dense image I_{mesh} of the point cloud. As the mesh is not textured, I_{mesh} is filled by the values of the *z-buffer* of the rendering (*i.e.* the depth of each pixel). Figure 5.4 displays an example of a sparse projection of the point cloud (b) in the image domain of (a) compared to texture-less rendering (c) and depth rendering (d). We can see that the renderings are largely denser than the sparse projection, resulting in the appearance of strong depth gradients.

5.3.2 Depth to optical image alignment

As mentioned in Section 5.2, the alignment between a LiDAR point cloud \mathcal{P} and an optical image I is non-trivial as both modalities do not share any common attribute. The mesh rendering I_{mesh} provides strong depth gradients in the image domain. These gradients correspond to object contours which can also be met in the optical image. Although strong depth gradients can occur without appearing in the optical image, and vice-versa, it is reasonable to assume that most depth gradients also appear in the optical image in real data. Therefore, aligning \mathcal{P} and I in the domain of I can be simplified as the alignment between the gradients of I_{mesh} and I . However, this assertion is only true if the initialization of the alignment between

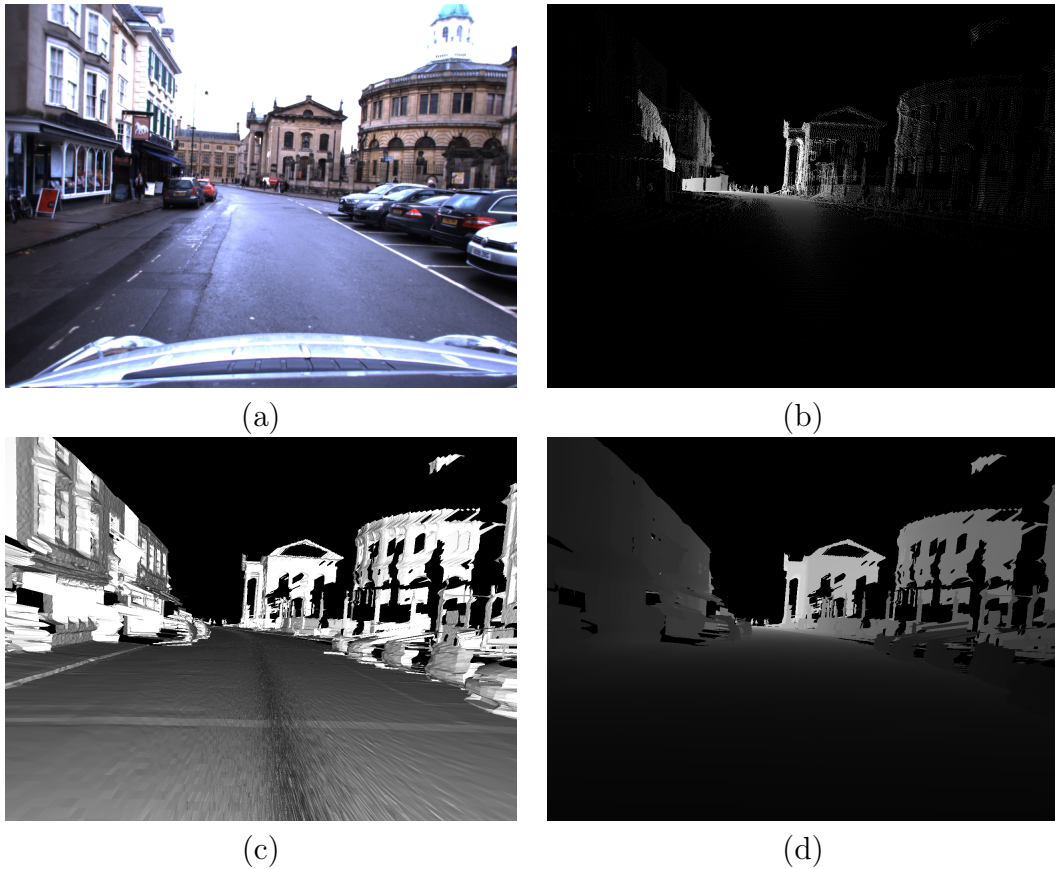


Figure 5.4: Rendering of mesh at optical image location. (a) is the optical image, (b) is the point cloud projected in the optical image domain without mesh reconstruction, (c) is a texture-less rendering of the mesh with flat shading that uses the normals of the triangles and (d) is the depth rendering of the mesh reconstructed from the point cloud.

I_{mesh} and I is relatively close. Indeed, the perspective induced by the 3D rendering introduces deformations that are proportional to the depth of the scene. Thus, if the initialization is too far from the optimal alignment, the alignment between the gradients of I_{mesh} and I is not possible.

The method described in [Sutour et al. \(2015\)](#) offers to align gradients of two modalities expressed in the same image domain (Section 5.2.1). To that extent, they define a variational model in which gradient alignment between images u_1 and u_2 is done by maximizing the criterion presented Equation (5.1) for a 2D affine transform with 3 degrees of freedom: vertical and horizontal translation t_x, t_y as well as zooming z :

$$T_{t_x, t_y, z}(X) = \begin{pmatrix} 1+z & 0 & t_x \\ 0 & 1+z & t_y \\ 0 & 0 & 1 \end{pmatrix} X$$

where Ω is the domain of definition of I . In the case of LiDAR point cloud to optical image alignment, rotation should also be considered in the transform as we cannot assume that the rotation between both sensors is always null. Therefore, we propose to extend the model presented in [Sutour et al. \(2015\)](#) in order to estimate rotation as well as translation and zooming.

We define $\bar{T}_{z, t_x, t_y, \theta}$ the 4 degrees of freedom (t_x, t_y translation, z zoom and θ rotation) transformation matrix such that:

$$\bar{T}_{t_x, t_y, z, \theta} = T_{t_x, t_y, z} \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

with $s = 1 + z$ to simplify notations. Similarly to [Sutour et al. \(2015\)](#), the gradients of I_{mesh} and I are aligned by maximizing the following criterion:

$$C(\bar{T}) = \int_{\Omega} |\nabla I_{\text{mesh}}(\bar{T}_{t_x, t_y, z, \theta}(X)) \cdot \nabla I(X)| dX.$$

Using this formulation, an explicit optimization scheme is built to maximize the proposed criterion at each iteration n , by performing a gradient ascent on each parameters of the transformation $\bar{T}_{z, t_x, t_y, \theta}$:

$$\begin{cases} t_x^{n+1} = t_x^n + \lambda_1 \frac{\partial C}{\partial t_x}(\bar{T}_{t_x, t_y, z, \theta}) \\ t_y^{n+1} = t_y^n + \lambda_2 \frac{\partial C}{\partial t_y}(\bar{T}_{t_x, t_y, z, \theta}) \\ z^{n+1} = z^n + \lambda_3 \frac{\partial C}{\partial z}(\bar{T}_{t_x, t_y, z, \theta}) \\ \theta^{n+1} = \theta^n + \lambda_4 \frac{\partial C}{\partial \theta}(\bar{T}_{t_x, t_y, z, \theta}) \end{cases}$$

Table 5.1: MAE of each method compared to the manually aligned data for each parameter on 50 randomly generated transformations.

Method	Mean Absolute Error			
	t_x	t_y	z	θ
Mutual Information	16.3	11.9	0.05	0.46
Sutour et al. (2015) (baseline)	2.91	6.76	0.006	0.57
baseline + rotation	2.96	6.29	0.004	0.04
baseline + rotation + refined	1.93	3.31	0.005	0.03

where the partial derivatives of $C(\bar{T}_{t_x, t_y, z, \theta})$ are defined as follows for each iteration:

$$\begin{aligned}
\frac{\partial C}{\partial t_x}(\bar{T}_{t_x, t_y, z, \theta}) &= \int_{\Omega} \sigma \nabla^2 \bar{I}_{\text{mesh}}(X) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \nabla I(X) dX, \\
\frac{\partial C}{\partial t_y}(\bar{T}_{t_x, t_y, z, \theta}) &= \int_{\Omega} \sigma \nabla^2 \bar{I}_{\text{mesh}}(X) \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \nabla I(X) dX, \\
\frac{\partial C}{\partial z}(\bar{T}_{t_x, t_y, z, \theta}) &= \int_{\Omega} \sigma \nabla^2 \bar{I}_{\text{mesh}}(X) \begin{pmatrix} x \cos \theta + y \sin \theta \\ -x \sin \theta + y \cos \theta \end{pmatrix} \cdot \nabla I(X) dX, \\
\frac{\partial C}{\partial \theta}(\bar{T}_{t_x, t_y, z, \theta}) &= \int_{\Omega} \sigma \nabla^2 \bar{I}_{\text{mesh}}(X) \begin{pmatrix} -x \cdot s \sin \theta - y \cdot s \cos \theta \\ x \cdot s \cos \theta - y \cdot s \sin \theta \end{pmatrix} \cdot \nabla I(X) dX
\end{aligned}$$

having $\bar{I}_{\text{mesh}}(X) = I_{\text{mesh}}(\bar{T}_{t_x, t_y, z, \theta}(X))$ and $\sigma = \text{sign}(\nabla I_{\text{mesh}}(X) \cdot \nabla I(X))$. The functional we aim at optimizing is not convex. Therefore, it is highly subject to local maxima. However we consider that the alignment we seek to perform only concerns data provided by calibrated MMS. Therefore, the provided alignment of the LiDAR point cloud and the optical image is assumed to be close to the optimal alignment, as discussed here after in Section 5.4.1.

For the gradient ascent scheme, we set $\lambda_1 = \lambda_2 = 10^{-3}$ to be larger than $\lambda_3 = \lambda_4 = 10^{-5}$ as the translation expressed in pixel is likely to be larger than the rotation or the zooming factor. We set the maximum number of iterations to 200. However, most of our experiments have shown that the method converges in less than 30 iterations on the data presented in Section 5.4.

Finally, we propose to improve the gradient ascent scheme by refining the search steps at each iteration. The search step λ_x^n at iteration n is then defined as follows:

$$\lambda_x^n = \begin{cases} \lambda_x^{n-1} & \text{if } C^n(\bar{T}) > \rho C^{n-1}(\bar{T}) \\ \lambda_x^{n-1}/2 & \text{otherwise} \end{cases} \quad (5.2)$$

with $C^n(\bar{T})$ the energy at iteration n , $\rho = 0.99$. This improvement prevents the algorithm from being directly stuck in a local maxima, and it provides better results in practice as demonstrated in Section 5.4.1.

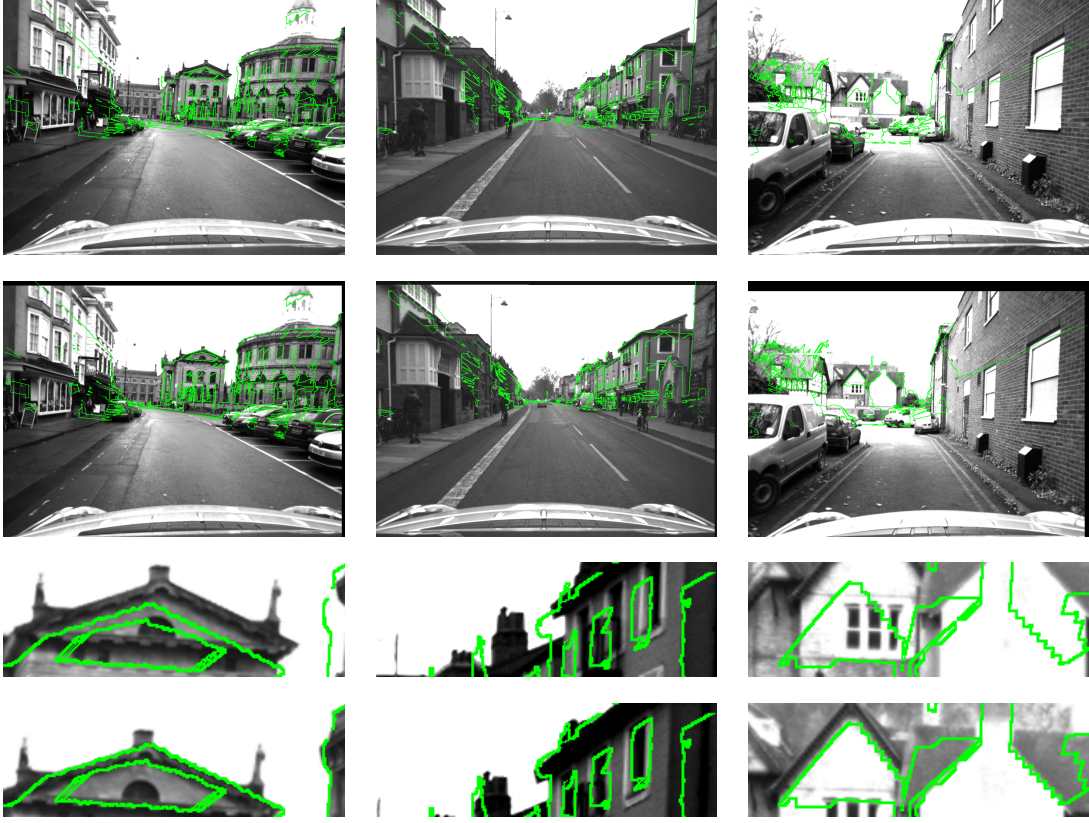


Figure 5.5: Example of alignments produced by our method. The green lines correspond to the strong gradients of the depth rendering. The first row shows the original alignment between the optical images and the mesh gradients. The second row shows the alignment produced by our method. A closeup look at details of original alignment and our results is showed on the last two rows respectively.

5.4 Experiments and results

We conclude this chapter by presenting different results obtained using the proposed framework. The proposed pipeline is evaluated on the RobotCar dataset ([Maddern et al., 2017](#)) which provides images of resolution 1280×960 px as well as point clouds composed of millions of points. We demonstrate the efficiency of the proposed method through a quantitative and qualitative analysis.

5.4.1 Quantitative analysis

The calibration of the RobotCar dataset does not provide a perfect alignment between LiDAR point clouds and optical images. We propose to manually align mesh renderings with optical images to create ground truths. We found out that the original data alignment compared to the ground truth alignment presents a Mean

Absolute Error (MAE) of about 19px for translation, 0.9 degree for rotation and 0.01 for zooming. We propose to apply comparable transformations onto manually aligned renders to generate evaluation data. The transformations are generated by randomly and uniformly shifting the renders between -20 and 20 pixels on both x and y axis, rotating the renders between -1 and 1 degree and zooming by a uniform random factor between 0.95 and 1.05 .

We compare our method with and without the refinement of the search steps (Equation (5.2)) to the method proposed in [Sutour et al. \(2015\)](#), as this method presents the baseline of gradient alignment without the estimation of the rotation. Moreover, we also compare our method to an exhaustive search of the maximum of the Mutual Information ([Viola and Wells III, 1997](#)) as done in recent multi-modal alignment methods, such as [Miled et al. \(2016\)](#). We compute the MAE between each estimated parameter (t_x, t_y, z, θ) and the ground truth. The results of this experiment are summarized in Table 5.1. We can see that our method achieves very fine alignment of LiDAR point cloud and optical image. The method with refinement of search steps provides finer results than each other method. The use of the functional defined in [Sutour et al. \(2015\)](#) as well as the extension presented in this chapter outperforms the exhaustive search with Mutual Information metric.

Moreover, we can see that extending the original functional by adding the regression of rotation improves the results not only in the estimation of the rotation, but also in the overall alignment. This is due to the fact that limiting the transformations to translation and scaling prevents the algorithm from finding the optimal alignment. Therefore, the baseline algorithm finds another local maxima which does not align well both modalities. This shows the importance of predicting the rotation as well as the baseline parameters of the transformation. Finally, the refinement of the search steps prevents the variational model from being stuck in local maxima, which makes it more robust to largely shifted initialization while keeping the same computational cost compared to the method presented in [Sutour et al. \(2015\)](#).

5.4.2 Qualitative analysis

We conclude our experiments with a qualitative analysis. Figure 5.5 presents the results of LiDAR point cloud to optical image alignment using our method. The first row shows the original alignment, the second row shows the results of the alignments using our method, with closeup looks at the original alignments and our results on the last two rows respectively. On each image, the strong gradients of the depth renderings are represented by green lines on the optical images.

The results presented in Figure 5.5 highlight that our model succeeds in aligning gradients of both modalities, producing a very good 2D registration between LiDAR point clouds and optical images. From initialization with shifted alignments (shown in the first row), our method produces results where both modalities are seamlessly aligned (second row). In particular, the last row of Figure 5.5 shows some areas where the variational model perfectly matches the renders and the optical images

on structures that display strong gradients such as roof lines or windows. Moreover, the method only requires to match a small amount of gradients in order to correctly align both modalities. This property makes it more robust to outliers as some gradients of the depth rendering do not correspond to any gradient in the optical image, and vice-versa, as discussed previously in Section 5.3.2. Finally, our method is able to produce good alignment even when initialized with large shifts between both modalities. This is specially visible in the last column where we can see that in the original alignment, the optical image is shifted from the gradients of the depth render. Despite this initialization, our method succeeds in producing a very fine alignment of the two modalities as it can be seen on the lowest line.

5.5 Conclusion

In this chapter, a novel framework for LiDAR point clouds to optical images alignment has been proposed. The first step of this framework offers to reconstruct the mesh from the point cloud by exploiting the topology of the sensor. After that, the mesh is rendered with the same pose as the optical image. Finally, the gradients of the rendering and the optical image are aligned using an adapted variational approach, and a method is proposed to refine the search step during the optimization. The qualitative and quantitative results demonstrate that the framework succeeds in very fine alignment between both modalities. The proposed method has been the object of a publication ([Biasutti et al., 2019c](#)), currently under review.

Although this registration method achieves very good results, it relies on the assumption that details (mostly objects) that appear in the LiDAR point cloud also appear in the optical image. However, considering the different temporalities between optical and LiDAR acquisition, this requirement cannot always be met in dynamic scenes (vehicles, pedestrians). In this case, segmenting and removing new objects from the point cloud might be needed to improve the coherence between the modalities. This is the subject of the next two chapters: Chapter 6 and Chapter 7.

Chapter 6

Object segmentation

Table of contents

6.1	Introduction	107
6.2	Point cloud segmentation	107
6.2.1	Region segmentation	107
6.2.2	Semantic segmentation	108
6.2.3	Tradeoff between region and semantic segmentation	110
6.3	Proposed region segmentation method	110
6.3.1	Methodology	110
6.3.2	Results & Analysis	112
6.4	Proposed semantic segmentation method	114
6.4.1	Input of the network	114
6.4.2	Architecture	114
6.4.3	Loss function	117
6.4.4	Training	117
6.5	Experiments	117
6.6	Conclusion	119

6.1 Introduction

MMS tend to acquire mobile objects that are not persistent to the scene. This often happens in urban environments with objects such as cars, pedestrians, traffic cones, etc. As LiDAR sensors cannot penetrate opaque objects, these mobile objects cast shadows behind them where no point has been acquired (Figure 6.1, top). As a result, merging optical data with the point cloud can be ambiguous as the point cloud might represent objects that are not present in the optical image. Therefore, the ability to segment such objects in the 3D LiDAR scene, as illustrated in the bottom Figure 6.1, is crucial for numerous applications, such as the registration method presented in Chapter 5.

To that extent, we argue that exploiting the sensor topology brings spatial structure into the point cloud that can be used for segmentation. This chapter introduces two methods for point cloud segmentation based on range-images: one for region segmentation and the second for semantic segmentation.

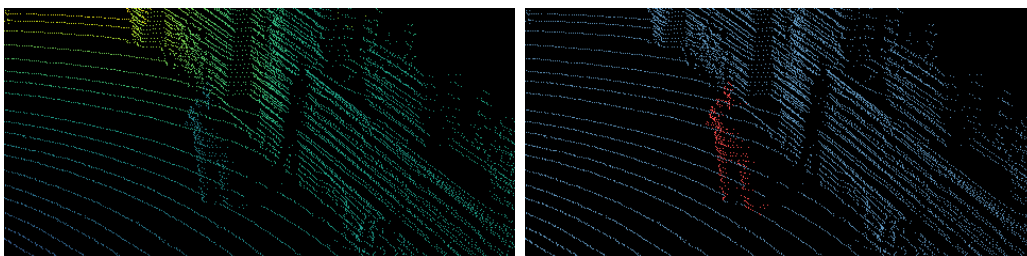


Figure 6.1: Result of the segmentation of a pedestrian in a point cloud using range-images. (left) original point cloud, (right) segmentation using range-image. The pedestrian is correctly segmented.

6.2 Point cloud segmentation

The problem of point cloud segmentation has been extensively addressed in the past years. It is often separated in two group of methods: region segmentation methods which aim at clustering the point clouds into regions, and semantic segmentation methods which target to label each point according to the nature of the object to which they belong.

6.2.1 Region segmentation

The segmentation of point clouds in regions is usually done either using geometry-based approaches, which directly operate on the 3D point cloud in order to aggregate points of a same region, or by turning the point cloud into a simpler representation beforehand.

Geometry-based segmentation The first well-known method in this category is region-growing where the point cloud is segmented into various geometric shapes based on the neighboring area of each point (Huang and Menq, 2001). Later, techniques that aim at fitting primitives (cones, spheres, planes, cubes ...) in the point cloud using RANSAC (Schnabel et al., 2007) have been proposed. Others look for smooth surfaces (Rabbani et al., 2006). Although these methods do not need any prior about the number of objects, they often suffer from over-segmenting the scene resulting in objects segmented in several parts.

Simplified model for segmentation MMS LiDAR point clouds typically represent massive amounts of unorganized data that are difficult to handle. Different segmentation approaches based on a simplified representation of the point cloud have been proposed. (Papon et al., 2013) propose a method in which the point cloud is first turned into a set of voxels which are then merged using a variant of the SLIC algorithm for super-pixels in 2D images (Achanta et al., 2012). This representation leads to a fast segmentation but it might fail when the scale of the objects in the scene is too different. (Gehring et al., 2017) propose to extract moving objects from MLS data by using a probabilistic volumetric representation of the MLS data in order to cluster points between mobile objects and static objects. However this technique can only be used with 3D sensors. Another simplified model of the point cloud is presented by (Zhu et al., 2010). The authors take advantage of the range-image (Chapter 4) to segment it before performing classification. The segmentation is done through a graph-based method as the notion of neighborhood is easily computable on a 2D image. Although the provided segmentation algorithm is fast, it suffers from the same issues as geometry-based algorithms such as over-segmentation or incoherent segmentation. Finally, an approach for urban objects segmentation using elevation images is proposed in (Serna and Marcotegui, 2014). There, the point cloud is simplified by projecting its statistics onto a horizontal grid. Advanced morphological operators are then applied on the horizontal grid and objects are segmented using a watershed approach. Although this method provides good results, the overall precision of the segmentation is limited by the resolution of the projection grid and it leads to the occurrence of artifacts at object borders.

6.2.2 Semantic segmentation

The problem of point cloud semantic segmentation has only been studied recently, contrary to image semantic segmentation which had been a very popular computer vision issue over the past decade. Because of our interest for range-images, as well as the plethora of related works for 2D images, we propose to introduce both image and point cloud semantic segmentation.

Semantic segmentation for images Semantic segmentation of images has been the subject of many works in the past years. Recently, deep-learning methods have

largely outperformed existing methods. The method presented in (Long et al., 2015) was the first to propose an accurate end-to-end network for semantic segmentation. This method is based on an encoder in which each scale is used to compute the final segmentation. Only a few month later, the U-net architecture (Ronneberger et al., 2015) (later generalized in (Badrinarayanan et al., 2017)) has been proposed for the semantic segmentation of medical images. This method is an encoder-decoder that is able to reach very fine precision in the segmentation. These two methods have largely influenced recent works such as DeeplabV3+ (Chen et al., 2018b) that uses dilated convolutional layers and spatial pyramid pooling modules in an encoder-decoder structure to improve the quality of the prediction. Other approaches explore multi-scale architectures to produce and fuse segmentations performed at different scales (Lin et al., 2017a; Zhao et al., 2018). Most of these methods are able to produce very accurate results, on various types of images (medical, outdoor, indoor). The review presented in (Briot et al., 2018) of CNNs methods for semantic segmentation provides a deep analysis of some recent techniques. This work demonstrates that a combination of various components would most likely improve segmentation results on wider classes of objects.

Semantic segmentation for point clouds The first approaches for point cloud semantic segmentation were done using heavy pipelines, composed of many successive steps such as: ground removal, point cloud clustering, feature extraction as presented in (Himmelsbach et al., 2008; Feng et al., 2014).

However, these methods often require many parameters and they are therefore hard to tune. Recently, (Landrieu and Boussaha, 2019) offers to extract features of the point cloud using a deep-learning approach. Then, the segmentation is done using a variational regularization. Another approach called PointNet presented in (Qi et al., 2017) proposes to directly input the raw 3D LiDAR point cloud to a network composed of a succession of fully-connected layers to classify or segment the point cloud. However, due to the heavy structure of this architecture, it is only suitable for small point clouds. Moreover, processing 3D data often increases the computational time due to the dimension of the data (number of points, number of voxels), and the absence of spatial correlation. To overcome these limitations, the methods presented in (Li, 2017) and it (Zhou and Tuzel, 2018) propose to represent the point cloud as a voxel-grid which can be used as the input of 3D CNN. These methods achieve satisfying results for 3D detection. However, semantic segmentation would require a voxel-grid of very high resolution, which would increase the computational cost as well as the memory usage.

Recently, Wu et al. proposed SqueezeSeg (Wu et al., 2018), a novel approach for the semantic segmentation of a LiDAR point cloud represented as a spherical range-image (Chapter 4). This representation allows to perform the segmentation by using simple 2D convolutions, which lowers the computational cost while keeping good accuracy. The architecture is derived from the SqueezeNet image segmentation method (Iandola et al., 2016). The intermediate layers are "fire layers", *i.e.* layers

made of one squeeze module and one expansion module. Later on, the same authors improved this method in (Bichen et al., 2018) by adding a context aggregation module and by considering focal loss and batch normalization to improve the quality of the segmentation. A similar range-image approach was proposed in (Yuan et al., 2018), where a Atrous Spatial Pyramid Pooling (Chen et al., 2018a) and squeeze reweighting layer (Hu et al., 2018) are added. These range-image methods succeed in real-time computation. However, their results often lack of accuracy which limits their usage in real scenarios.

6.2.3 Tradeoff between region and semantic segmentation

Regarding LiDAR point clouds acquired using MMS, the goal of segmentation is often to be able to cluster objects in the scene. To that end, semantic segmentation methods generally offer attracting results. However, as mentionned above, accurate methods rely on neural networks which are often limited when the data to process contains too many samples (*e.g.* a point cloud with millions of points). Therefore, the need for region segmentation methods that are able to efficiently process large point clouds still remains an open issue. Therefore, in the next Sections of this chapter, we propose very fast region segmentation method based on histograms of depth in range-images as well as a CNN based semantic segmentation method. Note that considering instance segmentation is beyond the scope of this thesis. Therefore, not individualizing each separate object is not considered as an error.

6.3 Proposed region segmentation method

In this section, we propose a simple yet efficient region segmentation technique based on range histograms.

6.3.1 Methodology

For the sake of simplicity, we assume that the ground is relatively flat and we remove ground points, which are identified by plane fitting. Note that these points could have also been identified using the method proposed in Chapter 1. Instead of segmenting the whole range-image u directly, we first split this image into S sub-windows $u_s, s = 1 \dots S$ of size $W_s \times H$ along the horizontal axis to prevent each sub-window from representing several objects at the same range. For each u_s , a depth histogram \mathcal{H}_s of B bins is built. This histogram is automatically segmented into C_s classes using the a-contrario technique presented in (Delon et al., 2007). This technique presents the advantage of segmenting a 1D-histogram without any prior assumption, *e.g.* the underlying density function or the number of objects. Moreover, it aims at segmenting the histogram following an accurate definition of an admissible segmentation, preventing over- and under-segmentation. An example of a segmented histogram is given in Figure 6.2.

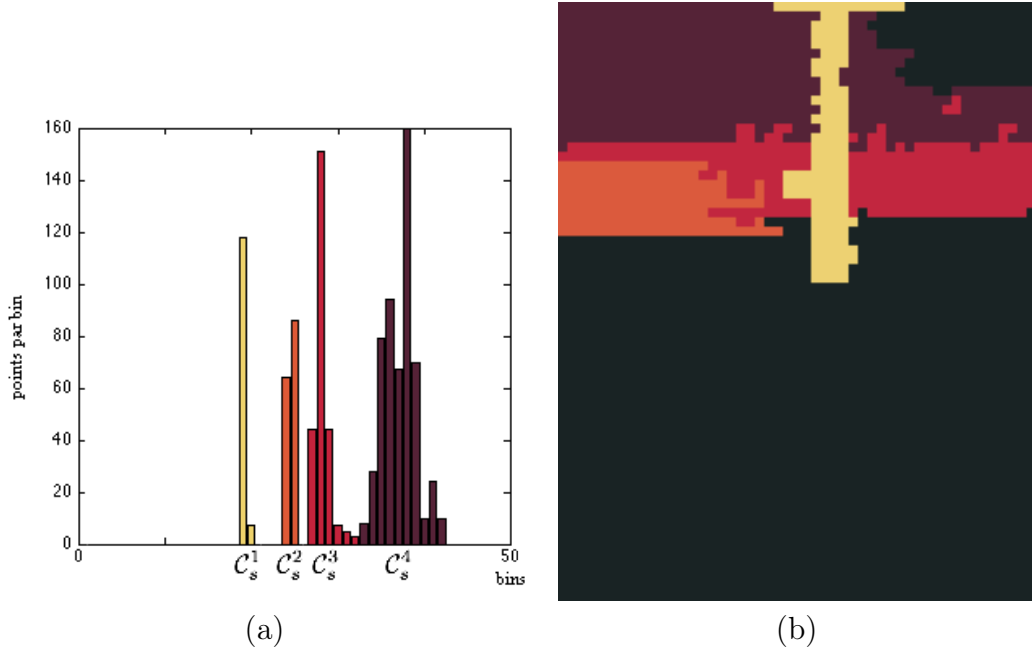


Figure 6.2: Result of the histogram segmentation using the approach of (Delon et al., 2007). (a) segmented histogram (bins of 50cm), (b) result in the range-image using the same colors. We can see how well the segmentation follows the different modes of the histogram.

Once the histograms of successive sub-images have been segmented, we merge together the corresponding classes by checking the distance between each of their centroids in order to obtain the final segmentation labels. Let us define the centroid C_s^i of the i^{th} class C_s^i in the histogram \mathcal{H}_s of the sub-image u_s as follows:

$$C_s^i = \frac{\sum_{b \in C_s^i} b \times \mathcal{H}_s(b)}{\sum_{b \in C_s^i} \mathcal{H}_s(b)} \quad (6.1)$$

where b are all bins belonging to class C_s^i . The distance between two classes C_s^i and C_r^j of two consecutive windows r and s can be defined as follows:

$$d(C_s^i, C_r^j) = |C_s^i - C_r^j| \quad (6.2)$$

Finally, we can set a threshold such that if $d(C_s^i, C_r^j) \leq \tau$, classes C_s^i and C_r^j should be merged (*e.g.* they now share the same label). If two classes of the same window are eligible to be merged with the class of an other window, then only the one with lower depth should be merged. Results of this segmentation procedure can be found in the next subsection. The choice of W_s , B and τ mostly depends on the type of data that is being treated (sparse or dense). For sparse point clouds (few thousand points per turn), B has to remain small (*e.g.* 50) whereas for dense point clouds

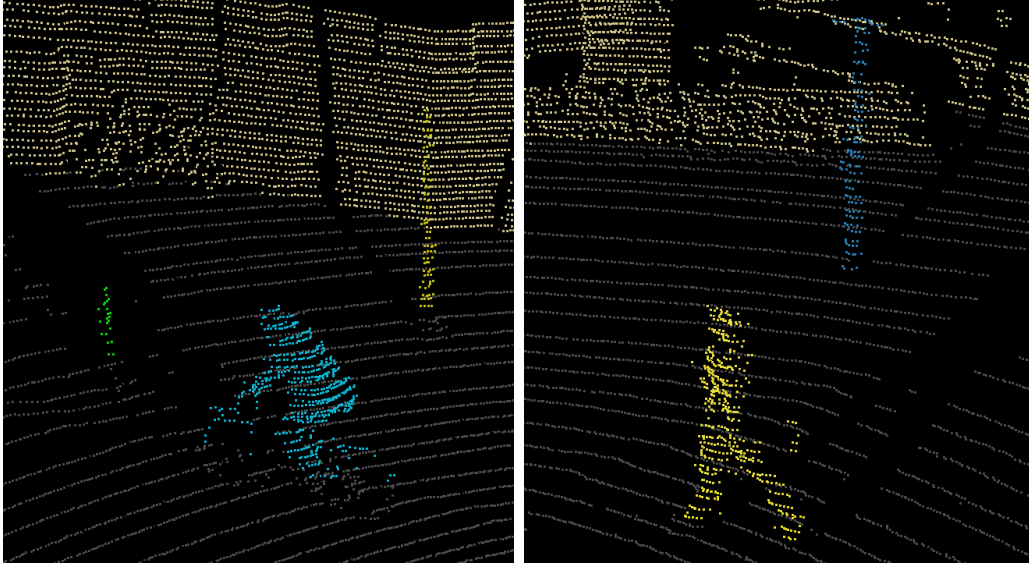


Figure 6.3: Example of point cloud segmentation using our model on various scenes. We can note how each label strictly corresponds to a single object (pedestrian, poles, walls).

(> 10^5 points per turn), this value can be increased (*e.g.* 200). In practice, we found out that good segmentations may be obtained on various kinds of data by setting $W_s = 0.5 \times B$ and $\tau = 0.2 \times B$. Note that the windows are not required to be overlapping in most cases, but for very sparse point clouds, an overlap of 10% is enough to achieve good segmentation. For example in our experiments on the KITTI dataset (Geiger et al., 2013), for range-images of size $2215 \times 64\text{px}$, $W_s = 50$, $B = 100$, $\tau = 20$ with no overlap.

6.3.2 Results & Analysis

Figure 6.3 shows two examples of segmentations obtained using our method on different point clouds from the KITTI dataset (Geiger et al., 2013). Each object, of different scale, is correctly distinguished from all others as an individual entity. Moreover, both results appear to be visually plausible.

Apart from the visual inspection, we also performed a quantitative analysis on the IQmulus dataset (Vallet et al., 2015). The IQmulus dataset consists of a manually annotated point cloud of 12 million points acquired with the Stereopolis-II MMS, in which points are clustered into several classes corresponding to typical urban entities (cars, walls, pedestrian, etc.). Our aim is to compare the quality of our segmentation on several objects to the ground truth provided by this dataset. First, the point cloud is segmented using our technique, using 100px wide windows with a 10px overlap and a threshold for merging set to 50. After that, we manually select labels that correspond to the wanted object (hereafter: cars). We then compare the result of the segmentation to the ground truth in the same area, and compute the

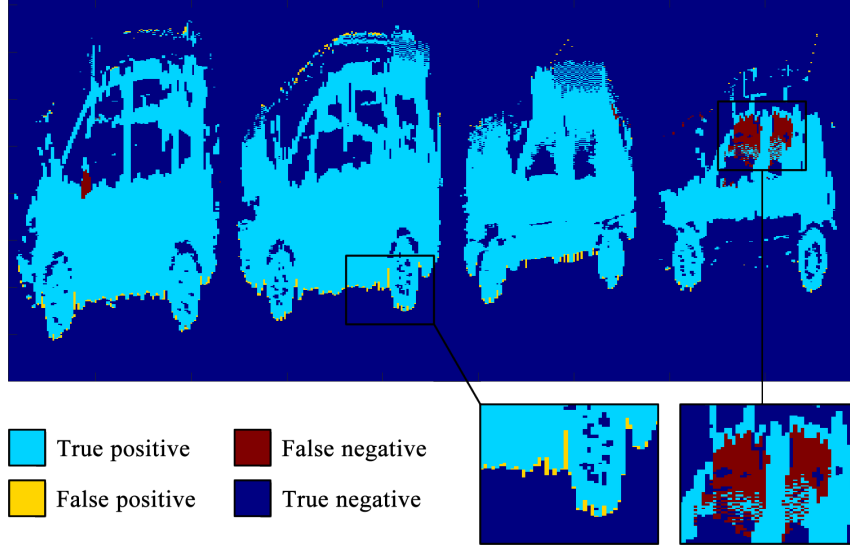


Figure 6.4: Quantitative analysis of the segmentation of cars. Our segmentation result only slightly differs from the ground truth in areas close to the ground or for points that were largely deviated such as points through windows.

Jaccard distance (Intersection over Union) between our result and the ground truth. Figure 6.4 presents the result of such a comparison. The overall distance shows that the segmentation matches 97.09% of the ground truth, for a total of 59021 points, which is very acceptable for such a number of points. Although the result is very satisfying, our result differs in some ways from the ground truth. Indeed, in the first zoom of Figure 6.4, our model better succeeds in catching the points of the cars that are close to the ground (we remind here that the ground truth on IQmulus was manually labelled and thus subject to errors). In the second zoomed-in part, points belonging to the windows of the car were not correctly retrieved using our model. This is because the measure in areas where the beam was highly deviated (*e.g.* beams that were not reflected in the same direction as the one they were emitted along) is not reliable as the range estimation is not realistic. Therefore our model fails in areas where the estimated 3D point is not close to the actual 3D surface. Note that a similar case appears for the review mirror (Figure 6.4, on the left) which is made of specular material that leads to bad measurements.

In some extreme cases, the segmentation is not able to separate objects that are too close from the sensor point of view. Figure 6.5.a shows a result of the segmentation in a scene where two cars are segmented with the same label (symbolised by the same color). In order to better distinguish the different objets, one can simply compute the connected components of the points regarding their 3D neighborhood (that can be computed using K-NN for example). Figure 6.5.b shows the result of such post-processing on the same two cars. We can notice how both cars are distinguished from one other.

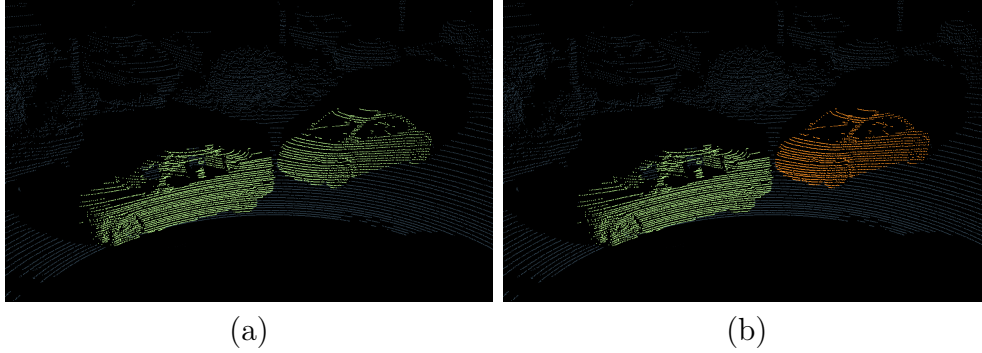


Figure 6.5: Result of the segmentation of a point cloud where two objects end up with the same label (a), and the labeling after considering the connected components (b).

6.4 Proposed semantic segmentation method

In this section, we present RIU-Net, our adaptation of the U-net architecture (Ronneberger et al., 2015) for the semantic segmentation of LiDAR point clouds, as illustrated in Figure 6.6. The method consists in feeding a U-net architecture with 2-channel images encoding range and elevation.

6.4.1 Input of the network

As mentioned above, processing raw LiDAR point clouds is computationally expensive. Indeed, these 3D point clouds are stored as unorganized lists of (x, y, z) Cartesian coordinates. Processing such data, or turning them into voxels involve heavy memory costs. To overcome such limitations, we propose to use a range-image named u with two channels: the depth towards the sensor and the elevation. In perfect conditions, the resulting image is completely dense, without any missing data. However, due to the nature of the acquisition, some measurements are considered invalid by the sensor and they lead to empty pixels, as discussed in Chapter 4. We propose to identify such pixels using a binary mask m equal to 0 for empty pixels and to 1 otherwise.

6.4.2 Architecture

The U-net architecture (Ronneberger et al., 2015) is an encoder-decoder. As illustrated in Figure 6.7, the first half consists in the repeated application of two 3×3 convolutions followed by a rectified linear unit (ReLU) and a 2×2 max-pooling layer that downsamples the input by a factor 2. Each time a downsampling is done, the number of features is doubled. The second half of the network consists in up-sampling blocks where the input is upsampled using 2×2 up-convolutions. Then, concatenation is done between the upsampled feature map and the corresponding feature map of the first half. This allows the network to capture global details while

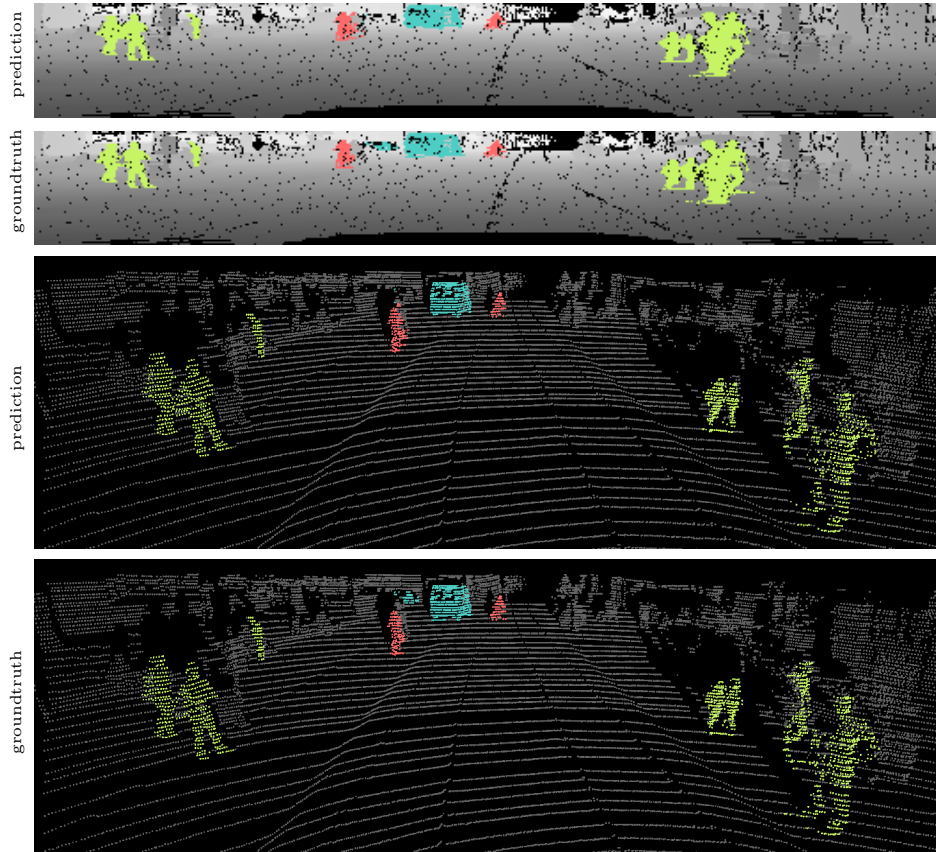


Figure 6.6: Result of the range-image semantic segmentation produced by the proposed method. The first two results show the prediction of the proposed model and the groundtruth respectively, seen in the sensor topology. The last two results show the same prediction and groundtruth in 3D.

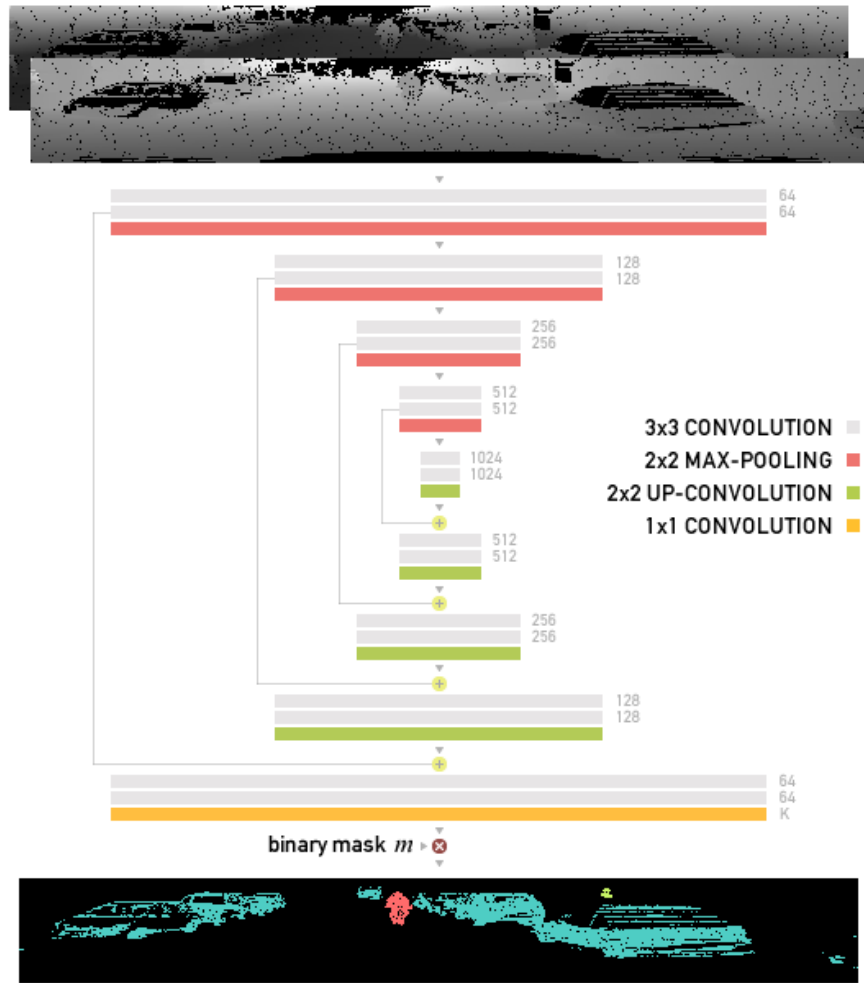


Figure 6.7: RIU-Net: U-Net architecture adapted to point cloud semantic segmentation with the depth and elevation channels input (top) and the output segmented image (bottom).

keeping fine details. After that, two 3×3 convolutions are applied followed by a ReLU. This block is repeated until the output of the network matches the dimension of the input. Finally, the last layer consists in a 1×1 convolution that outputs as many features as the wanted number of possible labels K 1-hot encoded.

6.4.3 Loss function

The loss function of the semantic segmentation network is defined as the cross-entropy of the softmax of the output of the network. The softmax is defined pixel-wise for each label k as follows:

$$p_k(x) = \frac{\exp(a_k(x))}{\sum_{k'=0}^K \exp(a_{k'}(x))}$$

where $a_k(x)$ is the activation for feature k at pixel position x . Defining $l(x)$ as the groundtruth label of the x pixel, we compute the cross-correlation as follows:

$$E = \sum_{x \in \Omega} \mathbb{1}_{\{m(x) > 0\}} w(x) \log(p_{l(x)}(x))$$

where Ω is the domain of definition of u , $m(x) > 0$ are the valid pixels and $w(x)$ is a weighting function introduced to give more importance to pixels that are close to a separation between two labels, as defined in (Ronneberger et al., 2015).

6.4.4 Training

We train the network with the Adam stochastic gradient optimizer (Kingma and Ba, 2014) and a learning rate set to 0.001. We also use batch normalization with a momentum of 0.99 to ensure good convergence of the model. Finally, the batch size is set to 8 and the training is stopped after 10 epochs.

6.5 Experiments

To test RIU-Net, we follow the experimental setup of the SqueezeSeg approach (Wu et al., 2018) for both training and evaluation. Indeed, they provide range-images with segmentation labels exported from the 3D object detection challenge of the KITTI dataset (Geiger et al., 2012). They also provide the training / validation split that they used for their experiments, which contains 8057 samples for training and 2791 for validation.

Figure 6.6 shows a segmentation result of RIU-Net and the groundtruth both on the range-image (top) and in 3D (bottom). The segmentation in 3D is obtained by labelling the raw point cloud according to the result on the range-image. More

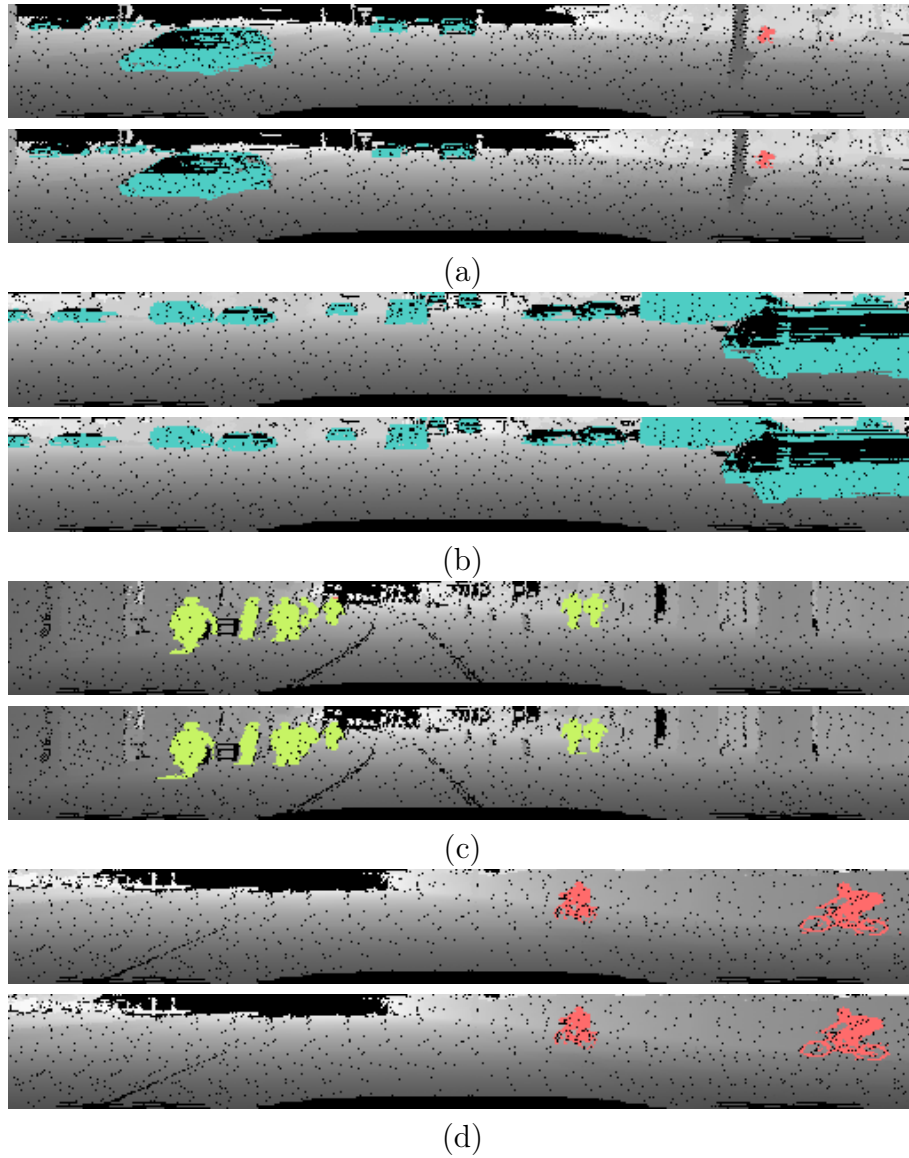


Figure 6.8: Results of the semantic segmentation of the proposed method (top) and groundtruth (bottom). Labels are associated to colors as follows: blue for cars, red for cyclists and lime for pedestrians.

results are shown in Figure 6.8. They all highlight how visually similar the results obtained with RIU-Net and the groundtruth are.

Similarly to (Wu et al., 2018) and (Bichen et al., 2018), we use the Intersection-over-Union (IoU) metric to evaluate RIU-Net and we compare it with the state-of-the-art:

$$\text{IoU}_l = \frac{|\rho_l \cap G_l|}{|\rho_l \cup G_l|}$$

where ρ_l and G_l denote the predicted and groundtruth sets of points that belongs to label l respectively.

Table 6.1 presents the results obtained for the segmentation of cars, cyclists and pedestrians, with SqueezeSeg (Wu et al., 2018), SqueezeSegv2 (Bichen et al., 2018) and PointSeg (Yuan et al., 2018) compared to RIU-Net. The scores of state-of-the-art methods are taken from the corresponding papers, using the same training conditions. We can see that the results of RIU-Net are comparable to the one of the state-of-the-art, despite the architecture being very simple. Our method achieves better average IoU scores compared to the PointSeg and the SqueezeSeg architectures. Moreover, it outperforms all the compared methods for cyclists. We believe that the increased number of parameters of our model compensate with the fact that the architecture had not been specifically designed for LiDAR point cloud segmentation in sensor topology, contrary to the other methods. The method SqueezeSegV2 is built on the SqueezeSeg architecture, while proposing several modifications (content aggregation modules and fire layers) driven by the specificity of the data. Therefore, it is reasonable to think that comparable results could be achieved with our model by applying the same modifications, however our goal is to keep the architecture simple and as generic as possible. Visual results of our method against the ground truth are displayed in Figure 6.8. Finally, we advocate that the proposed model can operate with a frame-rate of 90 frames per second on a single GPU, which is comparable, if not faster, to state-of-the-art methods and is largely over the standard requirements of real-time applications.

Table 6.1: Comparison (IoUs, %) of our approach with the state-of-the-art for the semantic segmentation of the KITTI dataset.

	Cars	Pedestrians	Cyclists	Average
PointSeg (Yuan et al., 2018)	67.4	19.2	32.7	39.8
SqueezeSeg (Wu et al., 2018)	64.6	21.8	25.1	37.2
SqueezeSegv2 (Bichen et al., 2018)	73.2	27.8	33.6	44.9
RIU-Net	62.5	22.5	36.8	40.6

6.6 Conclusion

In this chapter, we have presented two methods for 3D point cloud region segmentation and semantic segmentation that both take advantage of the range-image representation.

For region segmentation, the range-image is segmented using an histogram segmentation method. After that, the produced clusters are merged by comparing their centroids in consecutive windows. This produces a very fine segmentation of the point cloud and leads to very good qualitative and quantitative results. Moreover, the segmentation of the point cloud can be done online any time a new window is acquired, leading to great speed improvement, constant memory requirements and the possibility of online processing during the acquisition. This method has been presented in (Biasutti et al., 2017a), (Biasutti et al., 2017b) and (Biasutti et al., 2018).

For the semantic segmentation, the range-image is used as the input to a very common image semantic segmentation architecture. This permits to achieve scores that are comparable with the state of the art, while keeping the architecture very simple and while demonstrating that range-images are a valid bridge between image processing and 3D point cloud processing. This has been the object of a preprint (Biasutti et al., 2019e), and is still under investigation.

Among the many applications of a segmented point cloud, either in regions or with semantic information, disocclusion represents a crucial stake for the production of accurate 3D maps. This is the object of the next chapter.

Chapter 7

Object removal

Table of contents

7.1	Problem statement	122
7.2	Object removal methods	122
7.2.1	Image object removal	122
7.2.2	Point cloud object removal	124
7.3	Range-image disocclusion technique	124
7.4	Results & Analysis	127
7.4.1	Sparse point cloud	127
7.4.2	Dense point cloud	129
7.4.3	Quantitative analysis	130
7.4.4	Overlapping objects	132
7.5	Conclusion	133

7.1 Problem statement

Acquisition campaigns are typically done in environments that display car traffic, cyclists and pedestrians. This leads to the acquisition of non-persistent objects as discussed in the previous chapter. In a LiDAR acquisition, this implies that the laser beam is being blocked by these objects which prevents the acquisition of the structures that are situated behind. Therefore, the final point cloud contains holes – also named shadows – that correspond to parts of structures that have not been acquired by the sensor as they were obstructed by another object situated closer to the sensor. These shadows are largely visible when the point cloud is not viewed from the original acquisition point of view, as illustrated in Figure 7.3 (a). As a result, these defaults might end up being distracting and confusing for visualization.

To prevent such artifacts from appearing in the final acquisition, one might consider multiple acquisitions of the same area with temporal spacing or adding more sensors with different orientations. This would theoretically increase the completeness of the final acquisition as non-permanent objects are likely to have moved between two passages. However, despite increasing the chance of having a more complete acquisition, it does not guarantee that all the wanted areas of the scene will be acquired, but it does drastically increase the cost and the duration of the acquisition campaign.

As some applications rather focus on the interpretability of the point cloud rather than the precision of the measurements, it might be interesting to be able to automatically reconstruct parts of the point clouds that could not be acquired during the campaign. Such methods would need to provide plausible reconstructions (*i.e.* reconstruction that are visually pleasing, while helping to understand the scene), and to respect the topology of the scene as accurately as possible.

7.2 Object removal methods

The problem of object removal consists in replacing some object in a scene by a plausible approximation of what would have been there if the object to remove was not present in the acquisition.

7.2.1 Image object removal

The problem of object removal in images – also referred to as inpainting – is a well known theme of the image processing community. It can be formulated as filling the pixels that correspond to the object to remove. In many cases, interpolation methods (*e.g.* linear, bicubic) do not provide satisfying results as they tend to oversmooth the reconstructed areas and they tend to be unable to recreate details. Thus, this problem has been intensively investigated over the past decades, and

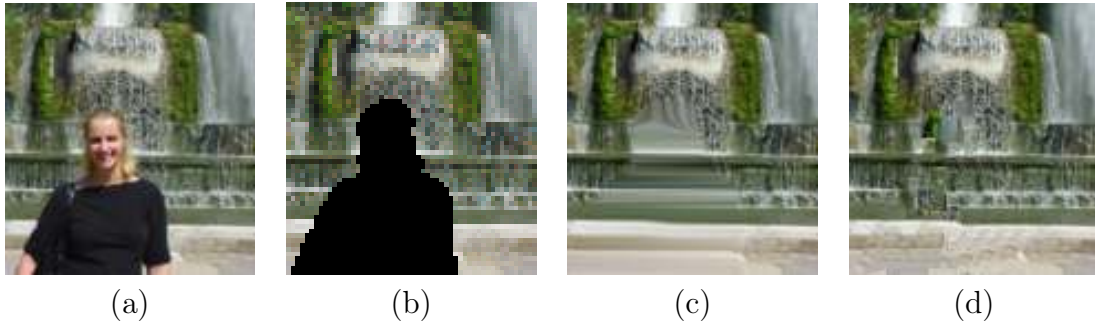


Figure 7.1: Inpainting of the image (a) in the masked area in black (b) using a geometric method (c) (Tschumperlé, 2006) and using a patch-based method (Criminisi et al., 2004).

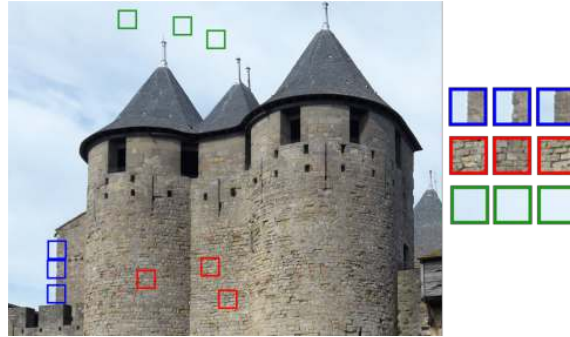


Figure 7.2: Illustration of the self-similarity principle. This image contains many repetition of local information.

many methods were proposed. These methods can be divided into two main groups: geometric approaches and patch-based approaches, as illustrated in Figure 7.1.

Geometric approaches Geometric approaches were naturally proposed to compensate the limitations of interpolation approaches. Indeed, they allow to constrain the reconstruction to respect some properties. In Weickert (1998) and Bertalmio et al. (2000), the authors propose different PDE-based methods that preserve edges in the reconstruction while Tschumperlé (2006) adopted a model that preserves curvatures in the reconstruction, as shown Figure 7.1 (c). These methods were then improved by using the Total Variation (TV) as proposed in Bredies et al. (2010) and Chambolle and Pock (2011). As mentioned in Chapter 2, they have been also extended to RGB-D images by taking advantage of the bi-modality of the data in (Ferstl et al., 2013a) and Bevilacqua et al. (2017). Although recent approaches achieve very satisfying results, they often rely on energy functions where the constraints are formulated with respect to the color information of the image, which is often not available in a LiDAR point cloud.

Patch-based approaches Patch-based methods were also proposed to overcome the limitations of interpolation methods. In that case, we consider that the information to reconstruct appear elsewhere in the image. This principle, called self-similarity, is illustrated in Figure 7.2. These methods were briefly mentioned in Chapter 1, where we proposed to extend the patch-based method suggested in Criminisi et al. (2004). This method was also extended in Lorenzi et al. (2011) and Buysens et al. (2015b) which have proven their strengths for image inpainting. They have been extended for RGB-D images in Buysens et al. (2015a) and for dense colored LiDAR point clouds with explicit grid topology in Doria and Radke (2012). However, patch-based methods extensively rely on texture information as well as the fact that similar objects appear with the same aspect everywhere in the image. Because of the way the sensor operates, similar objects can appear very differently in the range-image. Moreover, the range-image typically lacks of texture. Thus patch-based methods cannot be efficiently used on range-images.

7.2.2 Point cloud object removal

Object removal has only been scarcely investigated for 3D point clouds (Sharf et al., 2004; Park et al., 2005; Becker et al., 2009). However, these methods often rely on strong sampling assumptions, especially homogeneity, which is often not suitable for LiDAR point clouds, as discussed in Chapter 3.

In the next Section, we propose a novel approach for removing object in LiDAR point clouds in sensor topology.

7.3 Range-image disocclusion technique

The segmentation techniques introduced in Chapter 6 provide labels of objects which can be used to create masks for object removal, either by manual selection for the histogram-based segmentation method, or by selecting which type of object to remove on the semantic segmentation. By considering u the range-image representation of the point cloud rather than the point cloud itself, the problem of disocclusion can be reduced to the estimation of a set of 1D ranges instead of a set of 3D points, where each range is associated with the ray direction of the pulse. The Gaussian diffusion algorithm provides a very simple algorithm for the disocclusion of objects in 2D images by solving partial differential equations. This technique is defined as follows:

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u = 0 & \text{in } (0, T) \times \Omega \\ u(t = 0, x, y) = u(x, y) & \text{in } \Omega \end{cases} \quad (7.1)$$

having u defined on Ω , t being a time range and Δ the Laplacian operator. As the diffusion is performed in every direction, the result of this algorithm is often very smooth. Therefore, the result in 3D lacks of coherence as shown in Figure 7.3.b.

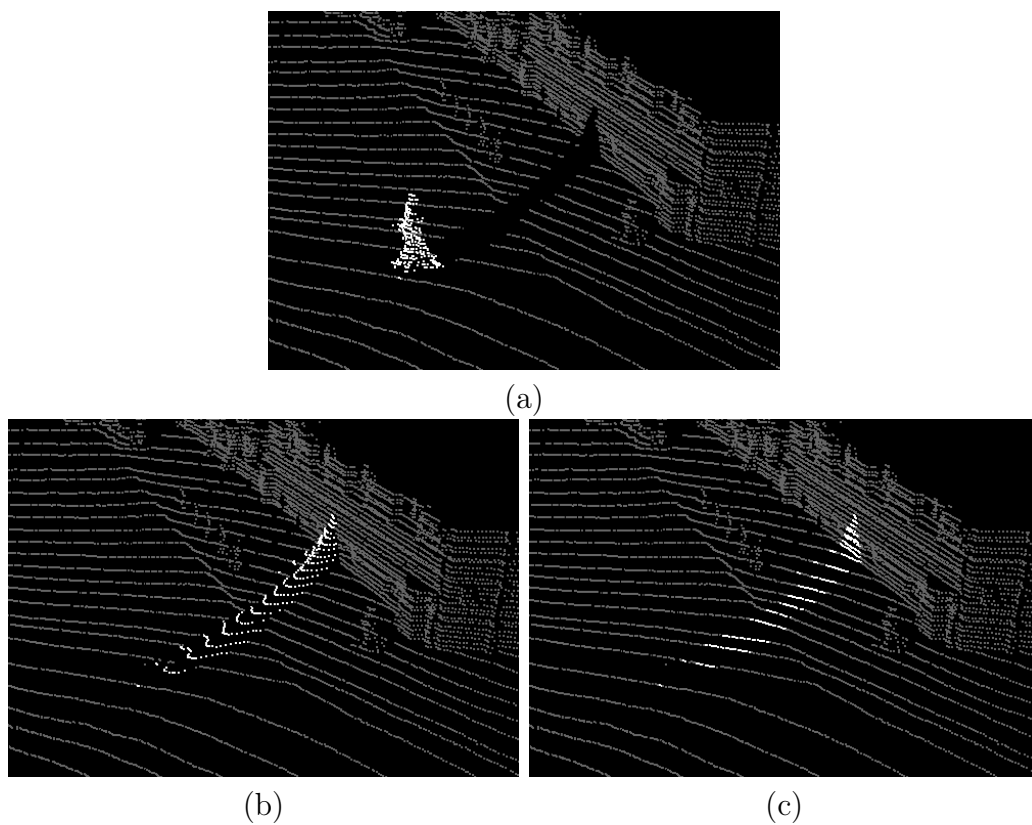


Figure 7.3: Comparison between disocclusion algorithms. (a) is the original point cloud (white points belong to the object to be disoccluded), (b) the result after Gaussian diffusion and (c) the result with our proposed algorithm (1500 iterations). Note that the Gaussian diffusion oversmooths the background of the object whereas our proposed model respects the coherence of the scene.

In this work, we argue that the structures that require disocclusion are likely to evolve smoothly along the x_W and y_W axes of the real world as defined in Figure 7.4.a. Therefore, we set η for each pixel to be a unitary vector orthogonal to the projection of z_W in the u range-image (Figure 7.4.b). This vector defines the direction in which the diffusion should be done to respect this prior. Note that most MLS systems provide georeferenced coordinates of each point that can be used to define η . For example, using a 2D LiDAR sensor that is orthogonal to the path of the vehicle, one can define η as the projection of the pitch angle of the acquisition vehicle.

We aim at extending the level lines of u along η . We assume the η to be a constant vector field. This can be expressed as $\langle \nabla u, \eta \rangle = 0$. Therefore, we define the energy $F(u) = \frac{1}{2} \langle \nabla u, \eta \rangle^2$. The disocclusion is then computed as a solution of the minimization problem $\inf_u F(u)$. As $\langle \nabla F(u), du \rangle = \lim_{\alpha \rightarrow 0} \frac{F(u + \alpha du) - F(u)}{\alpha}$, using the Green formula, the gradient of the energy function is given by $\nabla F(u) = -\langle (\nabla^2 u) \eta, \eta \rangle = -u_{\eta\eta}$, where $u_{\eta\eta}$ stands for the second order derivative of u with respect to η and $\nabla^2 u$ for the Hessian matrix. The minimization of F can be done by gradient descent. If we cast it into a continuous framework, we end up with the following equation to solve our disocclusion problem:

$$\begin{cases} \frac{\partial u}{\partial t} - u_{\eta\eta} = 0 & \text{in } (0, T) \times \Omega \\ u(t = 0, x, y) = u(x, y) & \text{in } \Omega \end{cases} \quad (7.2)$$

using the notations introduced earlier. We recall that the Laplacian $\Delta u = u_{\eta\eta} + u_{\eta^T \eta^T}$, where η^T stands for a unitary vector orthogonal to η . Thus, Equation (7.2) can be seen as an adaptation of the Gaussian diffusion equation (7.1) with respect to the diffusion prior in the direction η . Figure 7.3 shows a comparison between the original Gaussian diffusion algorithm and our modification. The Gaussian diffusion leads to an over-smoothing of the scene, creating an aberrant surface, whereas our modification provides a result that is more plausible.

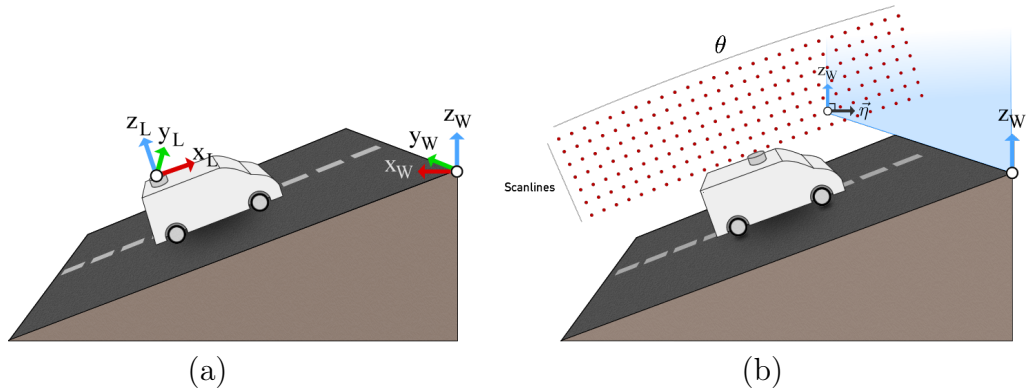


Figure 7.4: (a) is the definition of the different frames between the LiDAR sensor (x_L , y_L , z_L) and the real world (x_W , y_W , z_W), (b) is the definition and the visualization of η .

The equation proposed in (7.2) can be solved iteratively. The number of iterations simply depends on the size of the area that needs to be filled in.

7.4 Results & Analysis

In this part, the results of the disocclusion of their background are detailed.

7.4.1 Sparse point cloud

A first result is shown in Figure 7.5. This result is obtained for a sparse point cloud ($\approx 10^5$ pts) of the KITTI database. A pedestrian is segmented out of the scene using our proposed region segmentation technique (Section 6.3) and a manual selection of the corresponding label. This is used as a mask for the disocclusion of its background using our modified variational technique for disocclusion. Figure 7.5.a shows the original range-image. In Figure 7.5.b, the dark region corresponds to the result of the segmentation step for the pedestrian. For practical purpose, a very small dilation is applied to the mask (radius of 2px in sensor topology) to ensure that no outlier points (near the occluder’s silhouette with low accuracy or on the occluder itself) bias the reconstruction. Finally, Figure 7.5.c shows the range image after reconstruction. We can see that the disocclusion performs very well as the pedestrian has completely disappeared and the result is visually plausible in the range-image. Notice how the implicit sensor topology of the range-image has allowed here to use a standard 2D image processing technique from mathematical morphology to filter mislabelled and inaccurate points near silhouettes.

In this scene, η has a direction that is very close to the x axis of the range-image and the 3D point cloud is acquired using a 3D LiDAR sensor. Therefore, the coherence of the reconstruction can be checked by looking how the acquisition lines are connected. Figure 7.6 shows the reconstruction of the same scene in three dimensions. This reconstruction simply consists in the projection of the depth of each pixel along the axis formed by each corresponding point and the sensor origin. We can see that the acquisition lines are properly retrieved after removing the pedestrian. This result was generated in 4.9 seconds using Matlab on a 2.7GHz processor. Note that a similar analysis can be done on the results presented in Figure 7.7.

7.4.2 Dense point cloud

In this work, we aim at presenting a model that performs well on both sparse and dense data. Figure 7.8 shows a result of the disocclusion of a car in a dense point cloud. This point cloud was acquired using the Stereopolis-II system (Paparoditis et al., 2012) and it contains over 4.9 million points. In Figure 7.8.a, the original point cloud is displayed with the color based on the reflectance of the points for a better understanding of the scene. Figure 7.8.b highlights the segmentation of the car using our model (Section 6.3), dilated to prevent aberrant points. Finally, Figure 7.8.c depicts the result of the disocclusion of the car using our method. The car is perfectly removed from the scene. It is replaced by the ground that could not have been measured during the acquisition. Although the reconstruction is satisfying,

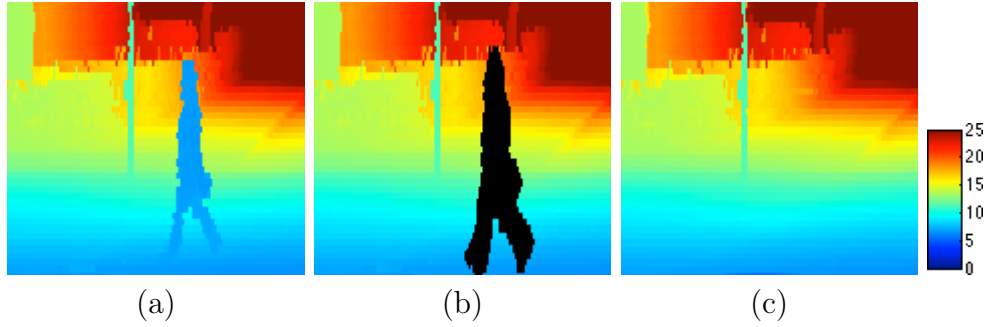


Figure 7.5: Result of disocclusion on a pedestrian on the KITTI database (Geiger et al., 2013). (a) is the original range image, (b) the segmented pedestrian (dark), (c) the final disocclusion. Depth scale is given in meters. After disocclusion, the pedestrian completely disappears from the image, and its background is reconstructed accordingly to the rest of the scene.

Table 7.1: Comparison of the average MAE (Mean Absolute Error) on the reconstruction of occluded areas.

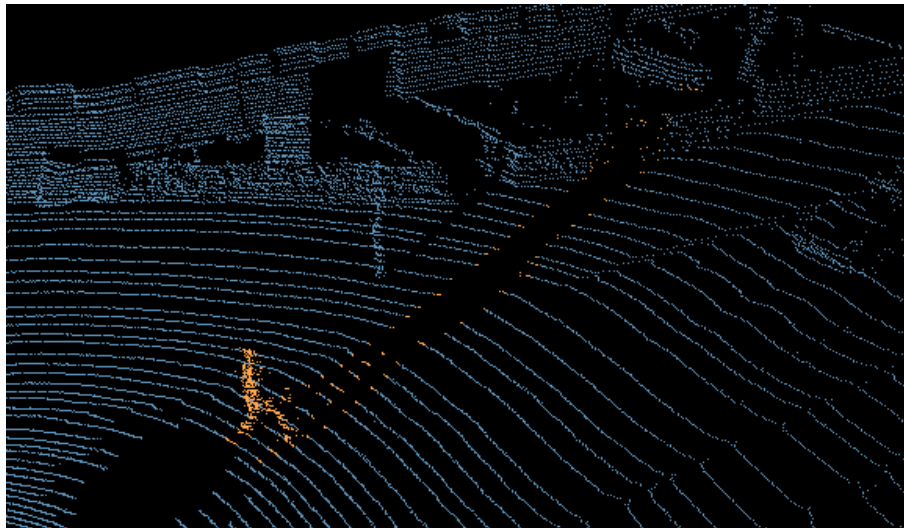
	Gaussian	Proposed model
Average MAE (meters)	0.591	0.0279
Standard deviation of MAEs	0.143	0.0232

some gaps are left in the point cloud. Indeed, in the data used for this example, pulses returned with large deviation values were discarded. Therefore, the windows and the roof of the car are not present in the point cloud before and after the reconstruction as no data is available. We could have added these no-return pulses in the inpainting mask as well to reconstruct these holes.

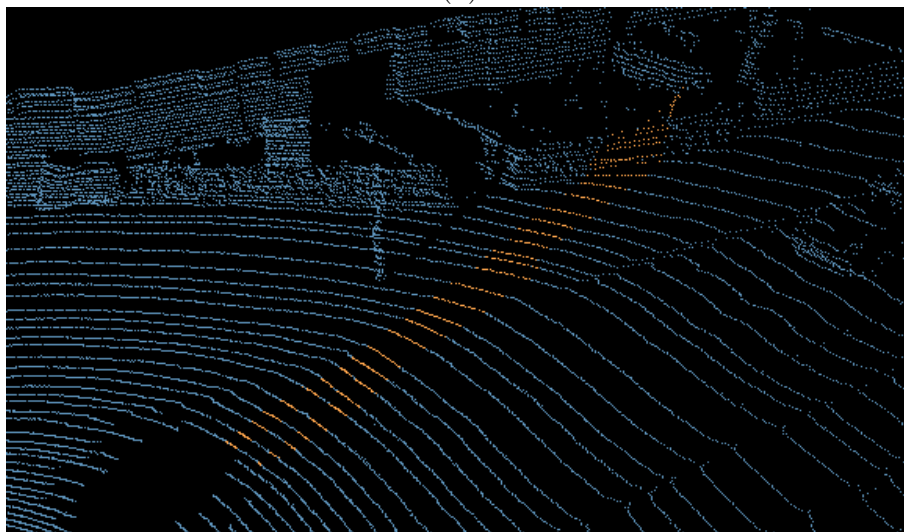
7.4.3 Quantitative analysis

To conclude this section, we perform a quantitative analysis of our disocclusion model on the KITTI dataset. The experiment consists in removing areas of various point clouds in order to reconstruct them using our model. The original point clouds can serve as ground truth. Note that areas are removed while taking care that no objects are present in those locations. Indeed, this test aims at showing how the disocclusion step behaves when reconstructing backgrounds of objects. The size of the removed areas corresponds to an approximation of a pedestrian’s size at 8 meters from the sensor in the range-image (20×20 px).

The test was done on 20 point clouds in which an area was manually removed and then reconstructed. After that, we computed the MAE between the ground truth and the reconstruction (where the occlusion was simulated) using both Gaussian disocclusion and our model. Table 7.1 sums up the result of our experiment. We can note that our method provides a great improvement compared to the Gaussian



(a)



(b)

Figure 7.6: 3D representation of the disocclusion of the pedestrian presented in Figure 7.5. (a) is the original mask highlighted in 3D, (b) is the final reconstruction.

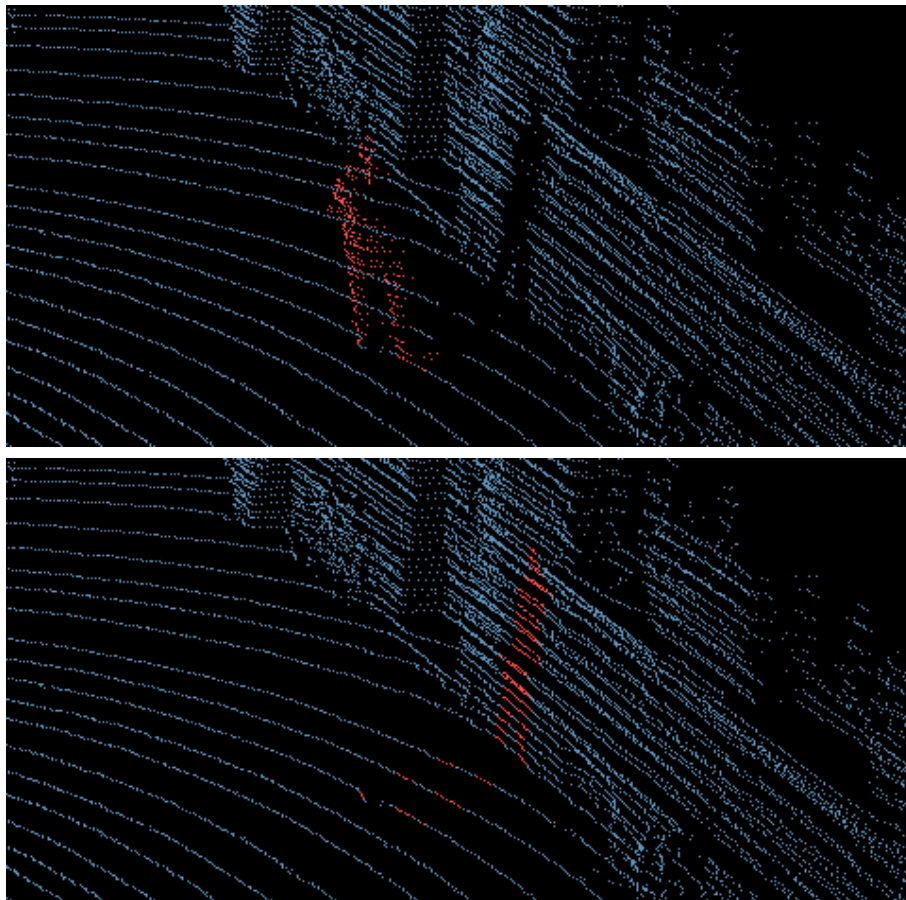


Figure 7.7: Result of the disocclusion of a pedestrian in a point cloud using range-images. (top) segmented point cloud, (bottom) disocclusion result.

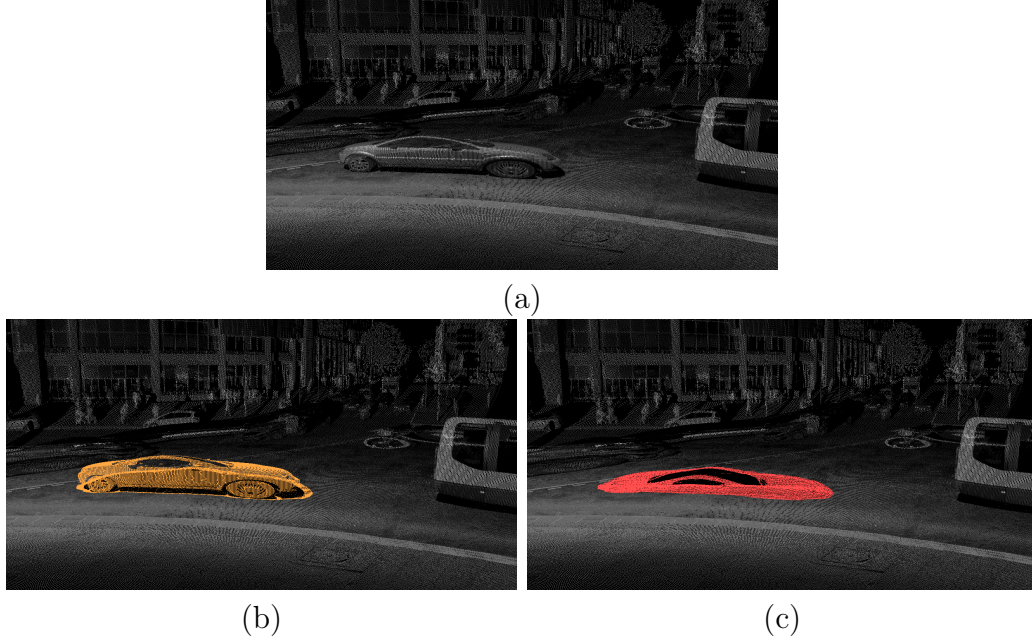


Figure 7.8: Result of the disocclusion on a car in a dense point cloud. (a) is the original point cloud colorized with the reflectance, (b) is the segmentation of the car highlighted in orange, (c) is the result of the disocclusion. The car is entirely removed and the road is correctly reconstructed.

disocclusion, with an average MAE lower than 3cm. These results are obtained on scenes where objects are located from 12 to 25 meters away from the sensor. The result obtained using our method is very close to the sensor accuracy as mentioned by the manufacturer ($\simeq 2cm$).

Figure 7.9 shows an example of disocclusion following this protocole. The result of our proposed model is visually very plausible whereas the Gaussian diffusion ends up oversmoothing the reconstructed range-image which increases the MAE.

7.4.4 Overlapping objects

Although the proposed disocclusion method performs well in realistic scenarios as demonstrated above, in some specific contexts, the reconstruction quality can be debatable. Indeed, when two small objects (pedestrians, poles, cars, etc.) overlap in front of the 3D sensor (*e.g.* one object is in front of the other), the disocclusion of the closest object may not fully recover the farthest object. Figure 7.10.a shows an example of such a scenario where the goal is to remove the cyclist (highlighted in green). In this case, a pole (Figure 7.10.a, in orange) is situated between the cyclist and the background. Figure 7.10.b presents the disocclusion of the cyclist. The background is reconstructed in a plausible way, however, details of the occluded part of the pole are not recovered.

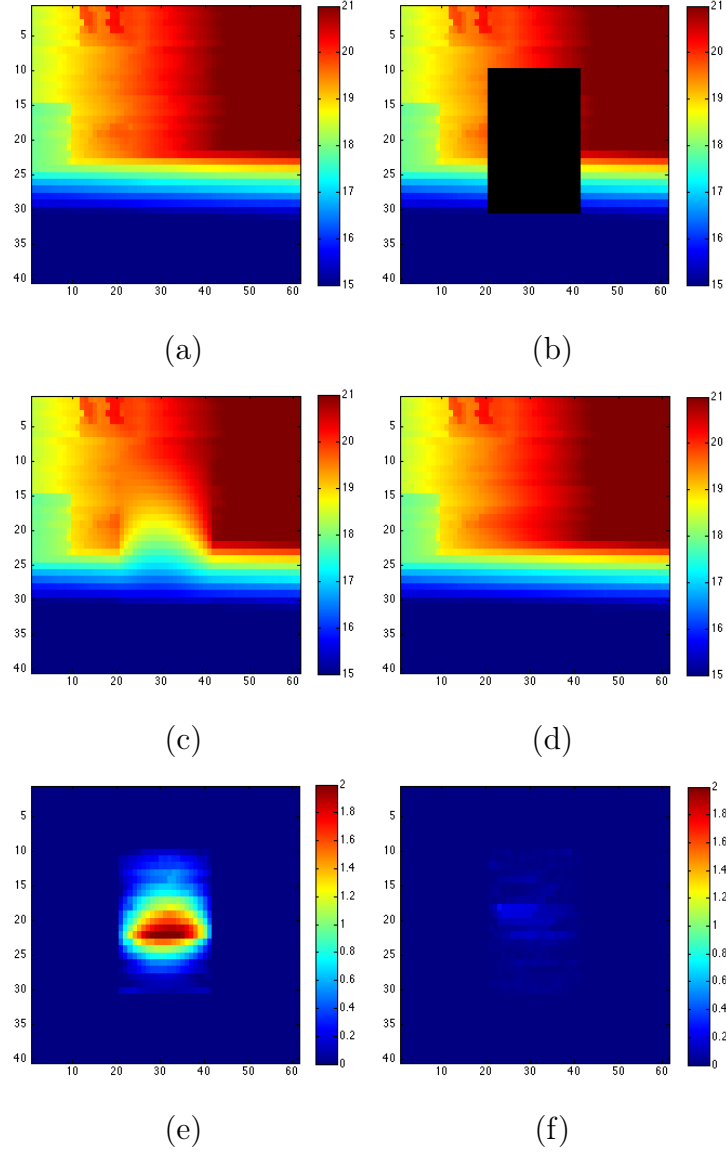


Figure 7.9: Example of results obtained for the quantitative experiment. (a) is the original point cloud (ground truth), (b) the artificial occlusion in dark, (c) the disocclusion result with the Gaussian diffusion, (d) the disocclusion using our method, (e) the Absolute Difference of the ground truth against the Gaussian diffusion, (f) the Absolute Difference of the ground truth against our method. Scales are given in meters.

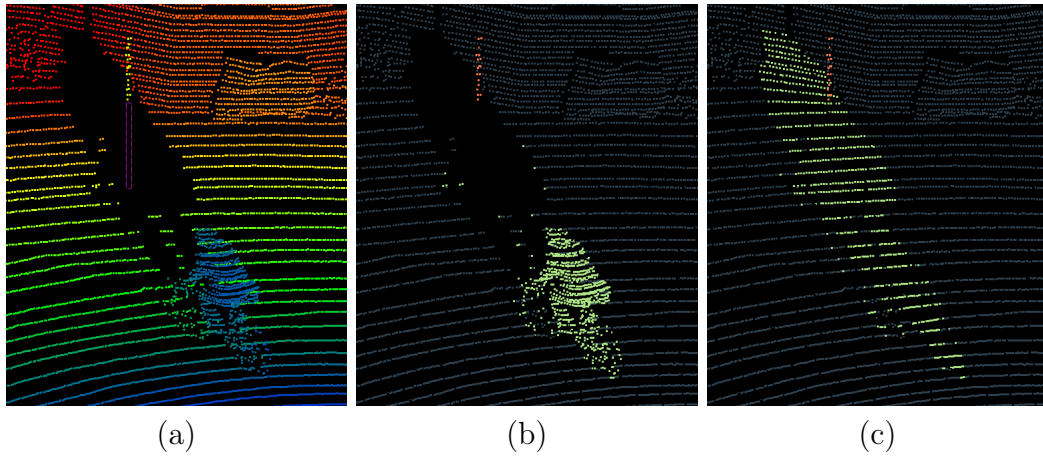


Figure 7.10: Example of a scene where two objects overlap in the acquisition. (a) is the original point cloud colored with depth towards sensor with the missing part of a pole highlighted with dashed pink contour, (b) shows the two objects that overlap: a pole (highlighted in orange) and a cyclist (highlighted in green), (c) shows the disocclusion of the cyclist. Although the background is reconstructed in a plausible way, details of the occluded part of the pole are missing.

7.5 Conclusion

In this section, we have proposed a novel approach to the problem of object removal in 3D LiDAR point clouds. Considering the range-image derived from the sensor topology has enabled a simplified formulation of this problem from having to determine an unknown number of 3D points to estimating only the 1D range in the ray directions of a fixed set of range-image pixels. Beyond simplifying drastically the search space, it also provides directly a reasonable sampling pattern for the reconstructed point set. Moreover, we have also proposed an improvement of a classical imaging technique that takes the nature of the point cloud into account (horizontality prior on the 3D embedding), leading to better results. We have validated the object removal method by visual inspection as well as quantitative analysis against ground truth and we have proved its effectiveness in terms of accuracy. This method has been presented in the following works: (Biasutti et al., 2017a), (Biasutti et al., 2017b) and (Biasutti et al., 2018).

Chapter 8

Object detection

Table of contents

8.1	Introduction	135
8.2	2D detection architecture for 3D detection	136
8.3	Methodology	137
8.3.1	3D detection and localization	137
8.3.2	2D detection on optical image	140
8.3.3	Projection and fusion of the predictions	141
8.4	Results	141
8.4.1	Qualitative analysis	141
8.4.2	Quantitative analysis	141
8.5	Conclusion	143

8.1 Introduction

We conclude this part of the thesis by exploring another application of MMS data. With the growing interest for autonomous driving, building onboard perception systems has become a major stake of the computer vision community. In particular, 3D object detection and localization is a crucial step to enable autonomous systems to sense their environment.

Over the past decade, 2D object detection on optical images have known great improvements (Ren et al., 2015; Lin et al., 2017b; Liu et al., 2016; Redmon and Farhadi, 2017; Lin et al., 2017c). On the other hand, 3D detection systems fail to achieve comparable performances in terms of accuracy or computational time.

3D object detection applied to LiDAR point clouds have recently been the subject of many papers thanks to the ongrowing use of deep-learning. The majority of proposed methods are based on discrete representations of the point cloud. These discrete representations either correspond to a vertical projection of the points on an horizontal pixel grid (Luo et al., 2018; Yan et al., 2018), sometimes coupled with extra modalities such as optical images (Chen et al., 2017a; Ku et al., 2018), or they correpond to 3D voxel grids (Zhou and Tuzel, 2018). This problem has also been adresssed by operating the 3D detection on subsets of the input point clouds. These subsets are typically extracted by projecting 2D optical detections in the point cloud to recover regions of interest. Points that fall in the regions of interest are then used as inputs to a neural networks (Qi et al., 2018; Shi et al., 2018). Although most of these methods achieve reasonable 3D detection scores and good 3D localizations, they often require towering computational power in order to treat the whole point cloud with enough precision - several millions of voxels are needed to represent a 3D LiDAR point cloud of the KITTI dataset (Geiger et al., 2012) at a 0.1 cubic meter resolution per voxel.

In the next section, we propose to investigate how the range-image can be used to perform 3D detection and localization while decreasing the computational cost. To that end, we present a novel – and lightweight – pipeline that exploits the range-image representation of the input point cloud to enable the use of 2D convolutional neural network by adapting an existing architecture for 2D detection. The 3D predictions are then refined by automatically merging it with 2D optical detections to avoid ambiguities as it will be discussed hereafter. This way, the proposed model illustrated in Figure 8.1 is able to perform 3D object detection and 3D object localization in real-time, making it very suitable for low power onboard systems.

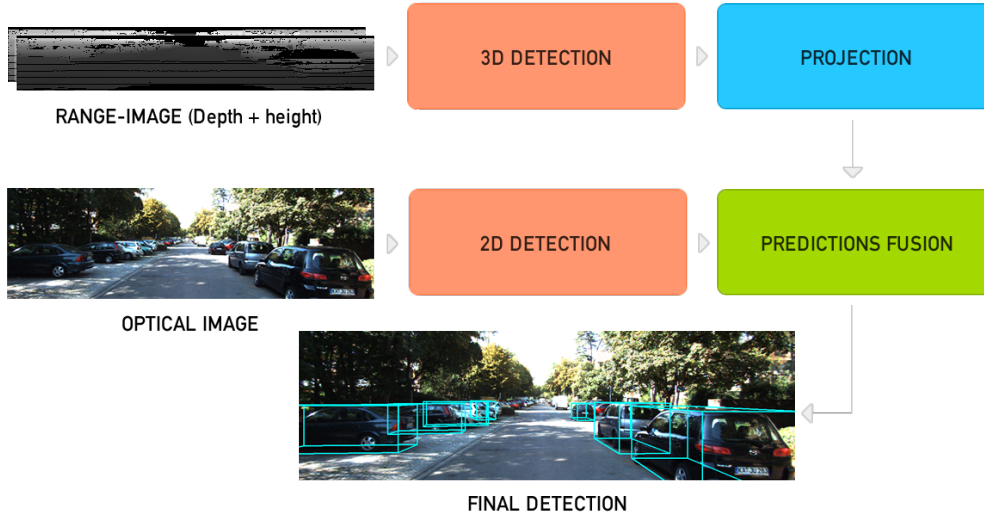


Figure 8.1: Proposed pipeline.

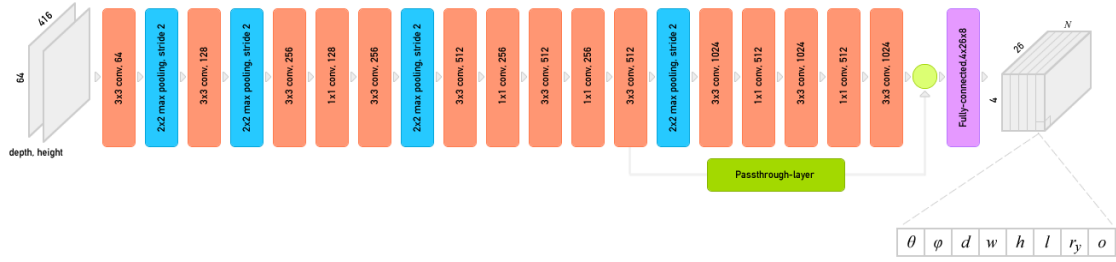


Figure 8.2: 3D detection and localization architecture.

8.2 2D detection architecture for 3D detection

As mentionned above, the task of 3D detection has already been studied by proposing computationally expensive architectures. Indeed, using the point cloud as the input, either raw or under voxel representation, implies that the network operates on high dimensional data that are often very greedy in terms of memory requirements. Therefore, the use of range-images for this task is an intuitive way to lower the memory usage while bringing the problem back to a better known paradigm: 2D Convolutional Neural Networks (CNN).

The task of 2D detection on RGB images is a well-known problem of computer vision that extensively makes use of CNNs. It has been the subject of a very large amount of contributions over the past years, largely encouraged by deep-learning improvements. In (Girshick, 2015), the authors propose a first architecture that aims at classifying windows of a RGB image. A sliding window is fed to a CNN that predicts both class and coordinates of the object in the window. However, the use of a sliding window is computationally expensive as the network has to test a lot of windows, with various dimensions, to efficiently perform the detection.

Region Proposal Networks To overcome the problem of using a sliding window, (Ren et al., 2015) offers to add a first stage to the network that shares the same features as the fast-RCNN stage, but it aims at predicting windows in which an object might appear. This allows to perform the 2D detection in real-time on low resolution images. This type of first stage network are often referred to as Region Proposal Networks. In (Lin et al., 2017b), the authors propose to extend the method presented in (Ren et al., 2015) by successfully using an encoder-decoder architecture to improve the precision of the results. Finally, in (Lin et al., 2017c), the authors investigate the case of learning hard classes that are less represented than others in the dataset by proposing a new exponential loss. Although these methods achieve very high scores on 2D detection challenges (Everingham et al., 2010), they can be hard to train, especially as they require to balance positive and negative samples.

Single Shot Detectors Instead of proposing a preliminary stage that proposes regions of interest, (Redmon et al., 2016) presents a single stage network that directly estimates if an object is present or not in a set of candidate bounding boxes. The image is fed to an encoder which outputs a feature map of lower scale. Each pixel of the feature map, also referred to as cell, is in charge of the detection in the corresponding area of the input image. To that end, a fixed number of candidate bounding boxes is initialized in each cell. Then, the network learns to discriminate cells that contain objects from cells that do not, while inferring slight offsets for each candidate box in order to refine the detection. The detection can be done in real-time thanks to the lightweight of the architecture. A similar method was proposed in (Liu et al., 2016), but both methods were quickly outperformed by the YOLO9000 architecture, proposed in (Redmon and Farhadi, 2017). This method uses a similar backbone as (Redmon et al., 2016), but it is able to achieve scores that are close to the best RPNs, while being much simpler to train.

In the next section, we propose to adapt the YOLO9000 method (Redmon and Farhadi, 2017) to perform 3D detection on 2D range-images built from 3D LiDAR point clouds.

8.3 Methodology

In this section, each step of the proposed pipeline is being detailed. An illustration of this pipeline can also be seen in Figure 8.1.

8.3.1 3D detection and localization

Input range-image To overcome dimensional limitations of previously mentioned methods, we offer to perform the 3D detection on range-images derived from LiDAR point clouds of the KITTI dataset. These range-images, shown in Figure 8.3, are generated with a fixed size of 416×64 pixels. This horizontal dimension corresponds

to an opening of ≈ 80 degrees facing the front direction of the acquisition vehicle. The range-images are composed of two channels: the depth towards the center of acquisition (in logarithmic scale to compensate the increasing spacing between scan-lines) and the elevation.

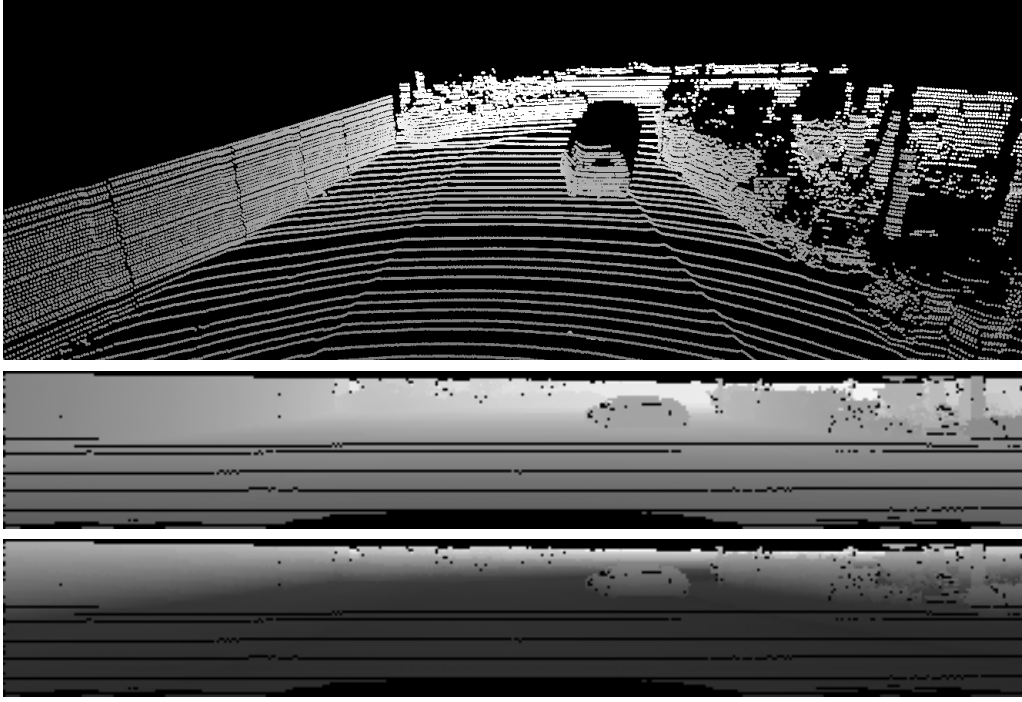


Figure 8.3: 3D LiDAR point cloud (top) and corresponding 2-channels range-image.

3D detection The first step of the proposed pipeline consists in the prediction of 3D coordinates as well as the dimensions and the orientation of the bounding boxes of each object from a range-image. To that extent, we offer to adapt the YOLO9000 architecture. This model is an encoder that estimates the presence of an object in each cell of the final layer. Each cell is divided into N possible objects, initialized with different dimensions. The *objectness* $o \in [0, 1]$ is computed for each object, indicating the probability that a cell contains an object or not. The strong spatial correlation between the prediction of each cell and the input image leads to a very accurate prediction of the 2D localization of the objects.

The prediction of the 3D position of an object in the range-image is similar to the prediction of the 2D localization, along with the prediction of the depth towards the sensor (Figure 8.2). Indeed, the coordinates of a pixel in the range-image directly correspond to the acquisition angles of the LiDAR sensor's beam. Therefore, we aim at predicting (θ, ϕ) , the horizontal and vertical acquisition angles respectively, as well as d the sensor depth of each object of the scene towards the center of acquisition.

Because of the perspective, the scale of objects in 2D varies proportionally to their depth. Thus, in YOLO9000, N possible objects are initialized with different dimensions to compensate the variation of sizes induced by the perspective. When working in 3D, the dimensions of an object do not change depending on the depth. Thus for each class, we define (H, W, L) the average dimensions of an object. Then, we aim at predicting (h, w, l) such that $(h*H, w*W, l*L)$ are equal to the dimensions of the predicted object.

In order to simplify the prediction of rotation, most of the 3D outdoor detection challenges (Everingham et al., 2010; Geiger et al., 2012) only consider the orientation of the object along the vertical axis (yaw) and they ignore the two other degrees of freedom (pitch, roll). This is assumed to be true in realistic scenarios as the ground on which the objects lie tends to be close to an horizontal plane. The range-image representation of the LiDAR point clouds correspond to a 360 degrees projection of the scene. Thus, two objects with similar yaw but different localizations in the 3D scene will have a different aspect in the range-image. Therefore, it is necessary to consider the θ angle of the object when predicting its rotation. We define r_y the rotation of an object along the vertical axis in the range-image, as illustrated in Figure 8.4.

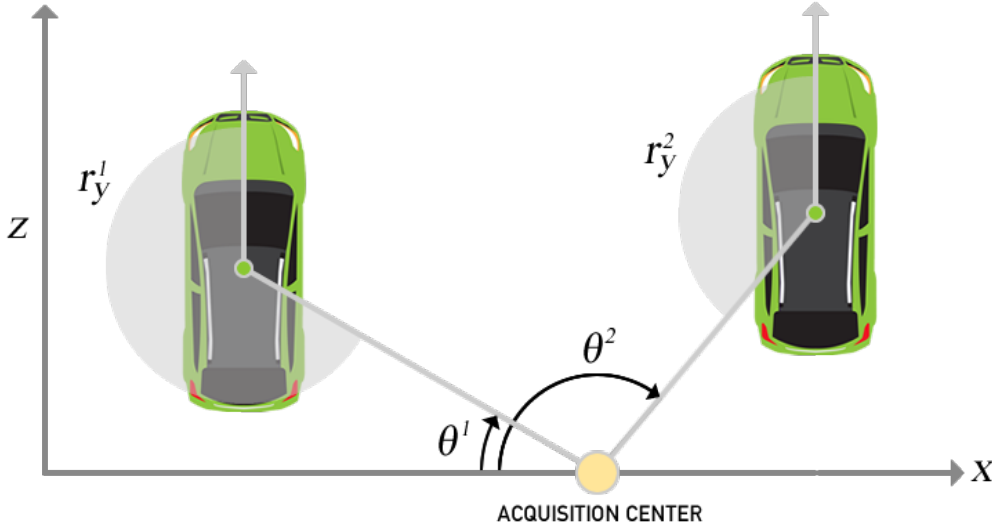


Figure 8.4: Example of r_y for two objects of similar yaw and different θ angles.

Training Let us define $\mathcal{F} = \{\theta, \phi, d, w, h, l, o\}$ the set of attributes that define an object without rotation. For each cell $c \in H \times W \times A$ that contains an object in the groundtruth, the following loss function is optimized:

$$\sum_{f \in \mathcal{F}} \lambda_f \|f(c) - \hat{f}(c)\|_2^2 + \lambda_{r_y} [1 - \cos(r_y(c) - \hat{r}_y(c))]$$

having \hat{x} the groundtruth value, λ_x the weight of each object attribute. The network is trained using the Adam optimizer with a learning rate set to 0.001. The batch size is set to 64 and the training is stopped after 10 epochs of the KITTI 3D object detection dataset (Geiger et al., 2012) that contains 7481 training examples.

Ambiguity of far objects Due to the low resolution of the LiDAR sensor, very few points hit the objects that are far from the sensor. Therefore, it is often very hard to distinguish these objects from the background as they are both represented by a couple of pixels only. This leads to ambiguous detections from the network, increasing the amount of false-positives. This is illustrated in Figure 8.5 on which one can see that the visual differences between the car as defined in the groundtruth and the prediction of our network is very small. To overcome this issue, we offer to couple the 3D detection model with a 2D detection method on optical images.



Figure 8.5: Example of ambiguous detections in the back of the scene. Both predictions (highlighted in red) look very much alike. However, only one detection really corresponds to an object in the groundtruth (in green). In this case, the object is a car.

8.3.2 2D detection on optical image

The accuracy of recent 2D optical detection methods (Ren et al., 2015; Dai et al., 2016; Lin et al., 2017b; Redmon and Farhadi, 2017; Liu et al., 2016; Lin et al., 2017c) has recently reached stunning scores on reference challenges. Some of these methods were specifically designed for urban 2D detection such as (Yang et al., 2016) or (Ren et al., 2017). In order to increase the robustness of our 3D detection model to the ambiguities brought by the lack of sampling on distant objects, we offer to perform a 2D detection of the same scene in optical images associated with the input point cloud. To that extent, we use the pre-trained version YOLO9000 trained on the COCO (Lin et al., 2014) as both code and weights are publicly available, and their performances are very satisfying as illustrated in Figure 8.6 on a KITTI dataset image.

8.3.3 Projection and fusion of the predictions

For each predicted 3D object, it is possible to compute the 3D coordinates of the 8 corners of the corresponding bounding box. Mobile Mapping Systems often provide accurate calibration settings of the system. Therefore, the coordinates of the corners can be projected in the optical image domain, assuming that the provided calibration is good enough as discussed in Chapter 5. It is then trivial to recover the smallest rectangle b_{3D} that contains the 8 projected corners. We define b_{2D} to be the 2D



Figure 8.6: 2D detection on an optical image from the KITTI dataset using the YOLO9000 method trained on the COCO dataset (Lin et al., 2014). We can see how accurate the prediction is (in red) compared to the groundtruth (green).

bounding box predicted by the 2D detector on the optical image. A 3D prediction is then considered valid whenever the intersection between its projection in the image domain and the 2D prediction is high enough. Therefore, a 3D prediction and its projection b_{3D} are valid if $\text{valid}(b_{3D}) > 0$ with:

$$\text{valid}(b_{3D}) = \sum_{b_{2D} \in B_{2D}} S(b_{3D}, b_{2D})$$

$$S(b_{3D}, b_{2D}) = \begin{cases} 1 & \text{if } \frac{|b_{3D} \cap b_{2D}|}{|b_{3D} \cup b_{2D}|} > t \\ 0 & \text{otherwise.} \end{cases}$$

where B_{2D} is the set of 2D predictions and t the intersection over union threshold above which we consider that a 3D prediction and a 2D prediction corresponds to the same object.

8.4 Results

In this section, we propose a qualitative and quantitative analysis of the proposed framework.

8.4.1 Qualitative analysis

Figure 8.7 highlights results of our method applied to the 3D detection of cars in urban scenes. We can see that the 3D detections (in light blue) are well aligned with the groundtruth (in green). Moreover, our method is able to distinguish close objects as well as objects that are far from the sensor, thanks to the coupling with the 2D detector. Finally, we can see that our method is robust to occluded objects, as it can be seen on many scenes where cars in the foreground are occluding the cars that are situated behind.

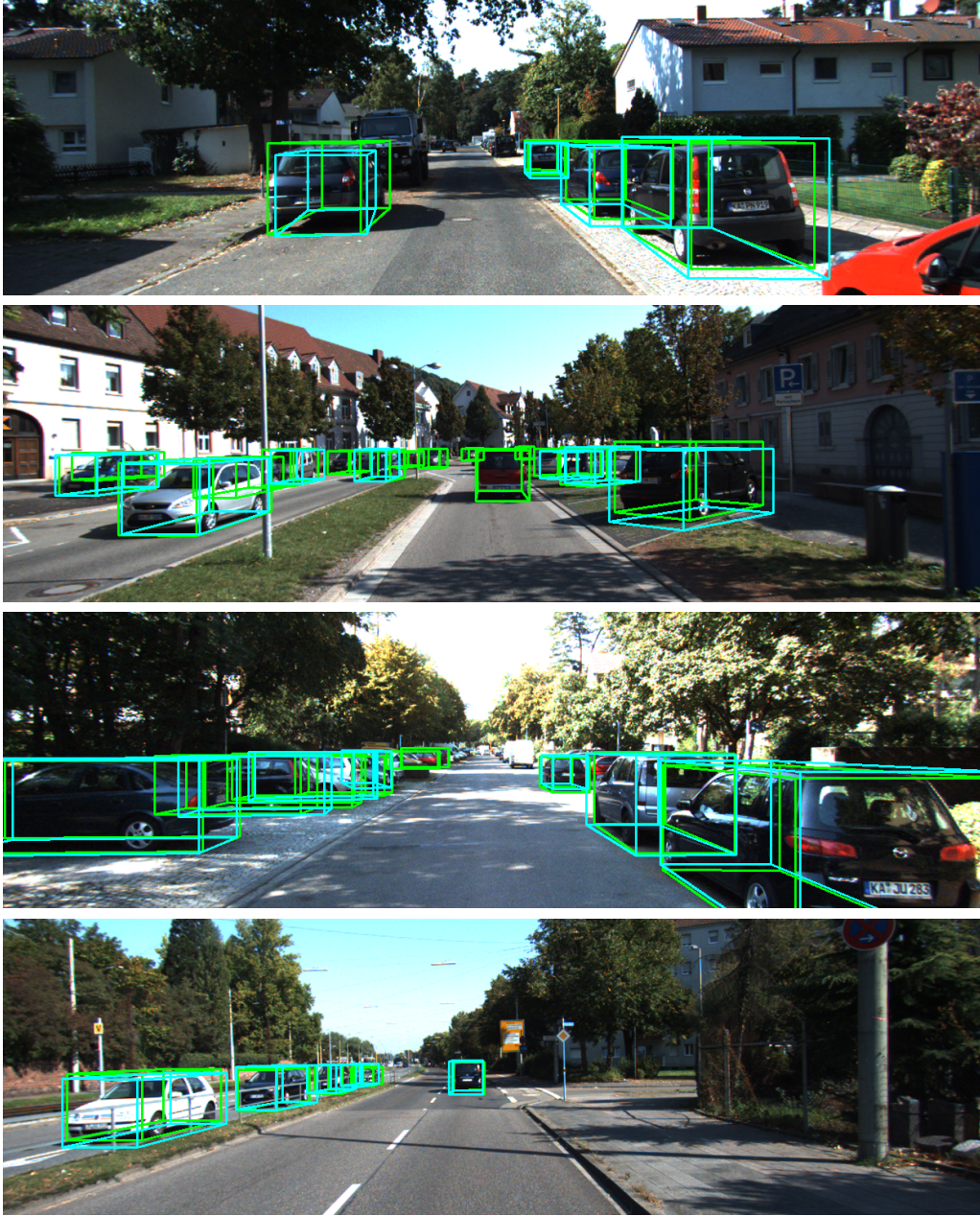


Figure 8.7: 3D detection produced by our pipeline (in blue) on scenes from the KITTI dataset and groundtruth (in green).

8.4.2 Quantitative analysis

Table 8.1 displays several accuracy metrics of the 3D detections against the groundtruth. We can see that for each detected object, the overall precision regarding each metric is very good. The 3D average distance between the center of the predicted bounding box and the center of the groundtruth is less than a meter, which is largely accept-

able in an urban scenarios in which a typical scene is spanned over 100 meters. Moreover, the average depth error shows that the localization error mostly comes from the prediction of d . This can be explained by the difficulty for the network to precisely learn the offset depth of the center of an object that is only observed from one side. Despite that, the average Intersection-over-Union in 3D is high. Finally, we can see that the average orientation error is low. The remaining error comes from the fact that it is hard to predict whether a car is facing towards the sensor or not directly from the range-image. Hence, some predictions happened to be rotated by 180 degrees.

	Score
Mean average 3D distance	0.53m
Mean average depth	0.51m
Mean 3D IoU	63%
Mean angular error	9.13 deg

Table 8.1: Mean average precision metrics of the 3D detection against groundtruth.

Nevertheless, as shown in Table 8.2, some objects are not detected by our method whereas some state-of-the-art methods succeed in better detection rates. Here, the evaluation method is the one presented in (Everingham et al., 2010). This is caused by the difficulty of correctly predicting the objectness of all objects in the scene. Moreover, the 2D detector that we use does not reach as good scores as the state of the art (Shi et al., 2018), which greatly impacts the performances of our method, as each missing detection drastically lowers the 3D mAP score. Despite that, our pipeline achieves much higher framerate than the state of the art.

Score KITTI (Geiger et al., 2012)	Proposed method	State of the art (Shi et al., 2018)
2D detection	63.67%	89.32%
3D detection	10.43%	75.76%
Orientation	29.87%	89.22 %
FPS (GPU)	312fps	10fps

Table 8.2: Scores of the proposed method against (Shi et al., 2018) on the KITTI 3D detection challenge (Geiger et al., 2012). Scores are measured as the mean Average Precision (mAP) with a 0.7 threshold of IoU against the groundtruth.

8.5 Conclusion

This chapter presents a novel approach for the 3D detection and localization of objects in LiDAR point clouds using range-images. The coupling of the 3D detector

with 2D optical detections lowers the ambiguities of false detections in the back of the scene. Our method succeeds in very fine localization of the objects. The detection of objects is satisfying, but it sometimes suffers in very crowded scenes. It has been the subject of the following publication: ([Biasutti et al., 2019b](#)).

Conclusion of the second part

In the second part of this document, we demonstrated how the topology of the sensor can be used to produce another type of 2D representation of the 3D LiDAR point cloud, named range-image. Range-images are dense, which implies that they can be employed as is without any preprocessing step. Thus, their use is often less costly than a 2D projection for many applications.

Range-image methodology We first showed how, depending on the nature of the sensor, the range-image could be generated. We also detailed how this result can be approximated despite the raw LiDAR data being unavailable by considering the local coordinates of the points compared to the acquisition center.

LiDAR to optical image alignment We developed a novel approach for the problem of LiDAR point cloud to optical image alignment. This method takes advantage of the topology of the sensor to reconstruct the mesh of the point cloud. A rendering of this mesh at optical camera location is then aligned with the optical image using a variational approach. The proposed method has shown its ability to accurately align both data.

Although this method provides very fine estimation of the 2D transformation that best aligns both modalities, certain applications require to estimate a 3D pose to improve the calibration of the system. In this case, our method could be improved by solving the Perspective-n-Points (PnP) problem ([Lepetit et al., 2009](#)) between the registered rendering and the actual 3D point cloud. The estimation could then be iteratively improved by computing the whole pipeline again with the newly estimated pose until some convergence is reached.

Segmentations We studied the problem of point cloud segmentation and semantic segmentation. We proposed a method for online point cloud segmentation that does not require any prior on the number of objects and that can operate on high resolution LiDAR acquisition seen from the topology of the sensor. The experiments conducted on this method have shown that it can perform with very high accuracy on large scenes. We also proposed a deep-learning based semantic segmentation approach for low-resolution LiDAR sensors. This method uses range-images as input of a CNN. The results have shown that this method achieves scores that are comparable with the state of the art, while using a very simple architecture, showing that range-image offer an effective way to treat 3D point clouds as images.

A first track of improvement of this method would be to study the interest

of using a loss that can compensate the imbalance between class representation. Indeed, as mentioned in Chapter 6, the dataset on which we trained our model contains an imbalanced number of examples depending on the class, resulting in pedestrians and cyclists being under-represented compared to cars. This can cause the network to affect more neurons to this class rather than for the other two. The use of the focal loss (Lin et al., 2017c) might therefore improve the results of the segmentation as it has been specifically designed to that end. We are also interested in the use of geometrical or spatial regularization in the loss in order to improve the spatial coherence of the prediction, either in 2D or in 3D.

Object removal We investigated the problem of object removal in a 3D point cloud by adapting the existing image inpainting literature to propose a variational inpainting method for range-images. This method reduces the dimension of the object removal problem by simply aiming at estimating new depth for each acquisition's ray. The proposed approach has shown its ability to remove and reconstruct accurately objects of the urban environment, while producing visually convincing results.

Due to the simplicity of the variational approach, the method is however unable to correctly reconstruct a strong gradient that would have been occluded by the sensor, as shown Figure 6.5. Although this case did not appear frequently in our experiment, it is theoretically possible and it would lead to unsatisfying reconstruction. Patch-based approaches represent a promising track to solve this problem as they are already widely used for images to preserve strong edges and texture frontiers. However, the lack of available texture in the range-image would require the development of novel patch-based metrics.

3D detection We also applied the range-image to the 3D detection problem by adapting a 2D detector for 3D detection. The proposed architecture is very lightweight and can easily be brought onboard low computational power systems despite showing interesting results.

However, as discussed in Chapter 8, the 3D detector sometimes misses object in the scene, as the estimation of the bounding boxes as well as the objectness is complicated in crowded environments. We believe that a multi-task version of the network would possibly improve the results while only increasing the memory requirement by a few. Indeed, multi-task architectures allow to create different branches for each wanted output. Thus, it would be possible to separate the objectness prediction from the bounding box prediction, which could leave more room for a better estimation in crowded spaces.

General conclusion and perspectives

1 General conclusion

This thesis has explored two ways to use image processing techniques on a 3D LiDAR point cloud that comes from MMS. The first manner is to project the 3D LiDAR point cloud onto a 2D pixel grid, and the second one is to take advantage of the topology of the sensor to produce a dense 2D image.

In the first part of this thesis, we have investigated how, given a projection model, the 3D LiDAR point cloud could be turned into an image.

We first showed that an orthogonal projection of the point cloud onto an horizontal grid could be used to generate high resolution orthoimages. We then demonstrated that RGB-D imaging could be built by projecting the point cloud into the domain of an optical image. Finally, we dealt with the problem of estimating the visibility of points by projecting the point cloud into the domain of an image.

We observed that the projection of the point cloud on a high resolution pixel grid ($> 0.5\text{M}$ pixels) creates a sparse image which cannot be directly used in many image processing methods. To solve this issue, we have shown how diffusion methods - especially variational methods - can be used to form a dense image from the projection. This step can rely on several channels of the LiDAR data, and/or by fusing the LiDAR data with an optical image. The sparsity of the projection also implies that the neighbors of a point cannot be retrieved by looking at adjacent pixels. We have shown how simple clustering methods (namely K-NN) could be used to retrieve the 2D neighbors of a point in the projection. All these steps have enabled the use of the densified projection of 3D LiDAR point clouds in the production of high resolution data.

In the second part of this document, we have explored how the topology of the sensor can be exploited to generate a dense 2D image from the 3D point cloud named range-image.

This type of image directly brings grid-like structure to the LiDAR data. We have shown how this structure can be used to easily create the mesh of the point cloud, which was later used in a LiDAR/optical alignment framework. Then, the problem of point cloud disocclusion was dealt with by using range-images in inpainting methods. Finally, we have investigated how range-images enabled the use of CNNs for deep-learning applications such as 3D detection and semantic segmentation.

Not only do range-images offer a way to avoid the preprocessing steps that are required by projections, they also offer a structured and canonical representation of the point cloud. We found out that in many applications, their use lead to better performances both in terms of accuracy and computational time. In deep-learning applications, range-images allow to benefit from the spatial properties of 2D convolutions of CNN architectures, while drastically reducing the memory usage compared to methods that directly process 3D point clouds. However, range-images provide the spatial structure of the sensors at the cost of loosing the spatial distribution of

the scenes.

While the projection of a LiDAR point cloud and its associated range-image present different advantages for representing the point cloud, they are equally interesting for LiDAR point cloud processing.

We found out that there are tendencies regarding the applications for which the range-image should be preferred to the projection, and vice-versa. On the one hand, the projection of a point cloud preserves the spatial distribution of the scene according to the point of view. Therefore, such representation satisfies most of the applications in which the output is a 2D representation of the point cloud, such as orthoimages. Moreover, a projection in the domain of an image offers an intuitive - yet efficient - way to fuse both modalities. Therefore, it seems to be a very good tradeoff for multimodal applications as it can be used to create correspondances between colors and 3D measurements.

On the other hand, a range-image is a very efficient way to represent the point cloud while bringing spatial structure. It is thus very interesting for 3D oriented tasks such as mesh reconstruction, ground filtering or geometric segmentation. The canonical aspect of such an image, along with the spatial structure, is meaningful for various real-time applications. In particular, it has shown promising results for deep-learning applications, such as semantic segmentation.

Although each representation provides separate advantages, they appear to be complementary for some applications. Indeed, we have shown in this thesis that some tasks can benefit from the use of both representations together, such as in LiDAR to optical image alignment or 3D detection.

2 Further works

Because of the diversity of the works proposed in this PhD thesis, several tracks of further works naturally appear. These tracks are grouped hereafter according to their general idea.

2.1 Densification with generative networks

The variational models proposed for the densification of the projection of the point cloud have shown great results for both orthoimage generation and RGB-D imaging (Chapters 1 and 2). Recent works on deep neural networks have led to significant improvements in many similar applications. In particular, generative networks have proven their strength for natural image inpainting (Yeh et al., 2017) and for image super resolution (Ledig et al., 2017). Thus, we believe that the use of such methods can improve the proposed densification approaches presented in this thesis, while being challenging as it has not yet been done for projected LiDAR data.

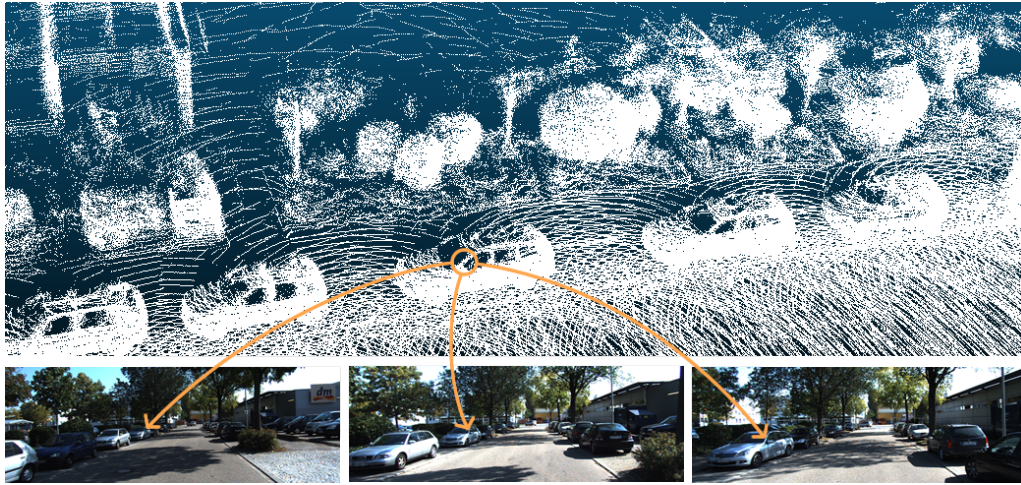


Figure 8.8: Example of multiple color candidates from different optical image for a same point in a LiDAR point clouds

2.2 Point cloud colorization

The result presented in Figure 3.9 shows how colors of optical images can be projected onto a point cloud. In this case, each point is associated with the color of the closest image. However, as a point is likely to be visible in many images, each point can be associated with a set of candidate colors. This idea is illustrated in Figure 8.8. Therefore, mixing the candidates is an intuitive extension of the proposed model. This task is related to image colorization methods that merge different candidates to enhance the results, such as in (Pierre et al., 2015).

2.3 Point cloud color prediction

MMS systems are built such that most of the points are visible from at least one optical image. To that end, optical cameras are generally placed to ensure maximum coverage of the scene. However, certain MMS such as in (Geiger et al., 2012) only present optical cameras turned in the front direction of the vehicle. Therefore, some points are never observed by the cameras. The colorization of such points cannot be done by projecting the color information of an image in the same way as it is done for Figure 3.9. The color of these points can only be predicted based on the information that they carry (coordinates, reflectance) and their neighborhood. Recent works in deep neural networks on the prediction of unobserved modalities, such as depth prediction from monocular images (Fu et al., 2018), offer an interesting starting point for color prediction on a LiDAR point cloud.

2.4 Multi-task learning

The tasks of semantic segmentation (Chapter 6) and 3D detection (Chapter 8) in a LiDAR point cloud present many similarities as they are applicable for the same objects. In particular, for non-permanent objects such as cars, the task of 3D detection consists in inferring a bounding box that only contains points that belongs to a car. Multitasks architectures for deep-learning approaches have demonstrated how all the tasks can mutually enhance the results, such as in (Liang et al., 2019). To that end, we believe that the strong correlation that exists between semantic segmentation and 3D detection can be exploited to create a more accurate, multitask architecture.

2.5 Spatial distribution in sensor topology

As discussed in Chapter 4, and later in Chapter 7, the sensor topology can be used to structure the point cloud in a range-image at the expense of loosing its spatial distribution. As a result, two similar objects in the same scene can appear differently in the sensor topology. This issue leads to several limitations. For example, and as discussed in Chapter 7, patch-based inpainting methods mostly rely on the self-similarity principle which requires that objects appear similarly in the image. For convolutional layers in CNNs, the learned kernels relevance largely depends on the redundancy of structures. If the structures that correspond to similar objects always appear differently, the network will have difficulties to learn meaningful features. Thus, the spatial invariance of the range-image is a crucial issue. Such invariance might be obtained by creating range-image where the channels correspond to 3D features extracted for each point.

2.6 Multimodal fusion

In Chapter 8, we have presented a method for 3D object detection where a late fusion of an optical image is done to enhance the results. Such fusion could be done earlier in the pipeline by merging optical features with LiDAR features in the 3D detection network. Recent works on 3D detection have showed that early multimodal fusion leads to better results such as in (Chen et al., 2017a) and (Ku et al., 2018). However, early fusion is challenging as optical image and LiDAR point clouds are expressed in different domains. In both (Chen et al., 2017a) and (Ku et al., 2018), the authors extract the features of each modality independently. The resulting feature-maps are then resized to a similar shape and concatenated to be later feed to the rest of the network. We believe that a better fusion would be possible by correlating the feature extraction of both modalities in the early layers of the network.

Appendix A

Primal-dual algorithm for solving Equation (2.7)

In order to provide a self-contained document, we include in this appendix the details of the primal-dual algorithm used to solve Equation (2.7). This proof is the result of the work of Marco Bevilacqua.

The optimization problem (2.7) turns out to be convex, but not smooth, due to ℓ_1 -type data-fidelity terms, $F(\zeta, v|\zeta_S)$ and $G(r, v|r_S)$, and the total variation regularization term $R(\zeta, r|I)$. Recently, in (Chambolle and Pock, 2011) a primal-dual first-order algorithm has been proposed to solve such problems. In Section 1 we provide the necessary definitions for the algorithm, which is subsequently described in Section 2.

1 Discrete setting and definitions

Images, considered in Section 2.2 as continuous functions in \mathbb{R}^2 , are here converted into real finite-dimensional vectors. Let M and N be the image dimensions in this discrete setting, and (i, j) the indices denoting all possible discrete locations in the Cartesian grid of size $M \times N$ ($1 \leq i \leq M$, $1 \leq j \leq N$). We then have ζ , ζ_S , r , r_S , v , I , and $\alpha \in X = \mathbb{R}^{MN}$, where X is a finite dimensional vector space equipped with a standard scalar product:

$$\langle \zeta, v \rangle_X = \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \zeta_{i,j} \zeta_{i,j}, \quad \zeta, v \in X. \quad (\text{A.1})$$

The gradient of an image $\zeta \in X$, $\nabla \zeta$, is a vector in the vector space X^2 with two components per pixel:

$$(\nabla \zeta)_{i,j} = ((\nabla_V \zeta)_{i,j}, (\nabla_H \zeta)_{i,j}). \quad (\text{A.2})$$

We compute the gradient components via standard finite differences with Neumann

boundary conditions, i.e.:

$$\begin{aligned} (\nabla_V \zeta)_{i,j} &= \begin{cases} \zeta_{i+1,j} - \zeta_{i,j} & i < M \\ 0 & i = M \end{cases} \\ (\nabla_H \zeta)_{i,j} &= \begin{cases} \zeta_{i,j+1} - \zeta_{i,j} & j < N \\ 0 & j = N \end{cases} \end{aligned} \quad (\text{A.3})$$

From the definition of gradient, it follows the expression of discrete coupled total variation, which matches the continuous one (2.5):

$$\text{TV}_\lambda(a, b) = \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \sqrt{(\nabla_H a_{i,j})^2 + (\nabla_V a_{i,j})^2 + \lambda^2 (\nabla_H b_{i,j})^2 + \lambda^2 (\nabla_V b_{i,j})^2} . \quad (\text{A.4})$$

As first suggested by (Chan et al., 1999), a total variation optimization problem can be recast into a primal-dual form that makes its solution easier, by rewriting the gradient norm by means of a vector-valued dual variable. To this end, in our case we first define a “coupled gradient” operator $\mathcal{K}_{\lambda b} : X \rightarrow Y$ ($Y = X^4$), which, applied to an image $a \in X$, expands its gradient to include the one of a reference image b according to a coupling parameter λ . I.e., we have the following element-wise definition:

$$(\mathcal{K}_{\lambda b} a)_{i,j} = ((\nabla_H a)_{i,j}, (\nabla_V a)_{i,j}, \lambda(\nabla_H b)_{i,j}, \lambda(\nabla_V b)_{i,j}) . \quad (\text{A.5})$$

The coupled gradient operator $\mathcal{K}_{\lambda b}$ can be further decomposed as $\mathcal{K}_{\lambda b} = \tilde{\mathcal{K}} + \beta_\lambda(b)$, according to the following element-wise definition:

$$\begin{aligned} (\mathcal{K}_{\lambda b} a)_{i,j} &= (\tilde{\mathcal{K}} a)_{i,j} + (\beta_\lambda(b))_{i,j} \\ &= ((\nabla_H a)_{i,j}, (\nabla_V a)_{i,j}, 0, 0) + (0, 0, \lambda(\nabla_H b)_{i,j}, \lambda(\nabla_V b)_{i,j}) . \end{aligned} \quad (\text{A.6})$$

$\tilde{\mathcal{K}}$ is the usual gradient operator “padded” with two zero components and it is linear in a ; $\beta_\lambda(b)$ is a bias term, depending on the gradient of the fixed variable b , which determines the last two components of the global coupled gradient operator.

Thanks to the definitions above, we can express alternatively the coupled total variation (A.4), by introducing the dual variable $p \in Y$:

$$\begin{aligned} \text{TV}_\lambda(a, b) &= \max_{p \in Y} \langle \mathcal{K}_{\lambda b} a, p \rangle_Y - \delta_P(p) \\ &= \max_{p \in Y} \langle \tilde{\mathcal{K}} a, p \rangle_Y + \langle \beta_\lambda(b), p \rangle_Y - \delta_P(p) , \end{aligned} \quad (\text{A.7})$$

where the scalar product in Y is defined as

$$\begin{aligned} \langle p, q \rangle_Y &= \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} p_{i,j}^1 q_{i,j}^1 + p_{i,j}^2 q_{i,j}^2 + p_{i,j}^3 q_{i,j}^3 + p_{i,j}^4 q_{i,j}^4 , \\ p &= (p^1, p^2, p^3, p^4), \quad q = (q^1, q^2, q^3, q^4) \in Y \end{aligned}$$

δ_P denotes the indicator function of the set P

$$\delta_P(p) = \begin{cases} 0 & \text{if } p \in P \\ +\infty & \text{if } p \notin P \end{cases} , \quad (\text{A.8})$$

and the feasibility set P for the dual variable p , is defined as

$$P = \{p \in Y \mid \|p_{i,j}\|_2 \leq 1, \forall i, j\} , \quad (\text{A.9})$$

i.e. $\|p\|_\infty \leq 1$.

We can now finally express the regularization term of our model $R(\zeta, r|I)$ (2.6) as the maximization over two dual variables. We then have:

$$\begin{aligned} R(\zeta, r|I) &= \max_{p \in Y} \max_{q \in Y} \langle \mathcal{K}_{\lambda_1 r} \zeta, p \rangle_Y + \langle \mathcal{K}_{\lambda_2 I} r, p \rangle_Y - \delta_P(p) - \delta_Q(q) \\ &= \max_{p \in Y} \max_{q \in Y} \langle \tilde{\mathcal{K}} \zeta, p \rangle_Y + \langle \beta_{\lambda_1}(r), p \rangle_Y + \langle \tilde{\mathcal{K}} r, q \rangle_Y + \langle \beta_{\lambda_2}(I), q \rangle_Y - \delta_P(p) - \delta_Q(q) . \end{aligned} \quad (\text{A.10})$$

This will let us formulate a discrete version of our joint inpainting problem (2.7), which falls into the primal-dual optimization framework. As for the other terms in (2.7), rewritten in discrete notation, we have:

$$\begin{aligned} F_1(\zeta|\zeta_S) &= \eta_1 \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \Phi_{i,j} \max(0, \zeta_{i,j} - \zeta_{S \ i,j}) \\ F_2(\zeta, v|\zeta_S) &= \eta_1 \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \Phi_{i,j} v_{i,j} \max(0, \zeta_{S \ i,j} - \zeta_{i,j}) \\ G(r, v|r_S) &= \eta_2 \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \Phi_{i,j} v_{i,j} |r_{i,j} - r_{S \ i,j}| \\ H(v|\zeta_S, r_S) &= \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \Phi_{i,j} \alpha_{i,j} (1 - v_{i,j}) \end{aligned} \quad (\text{A.11})$$

where Φ is a binary mask indicating the initial known pixels, i.e. belonging to the sparse image support Ω_S .

2 A primal-dual algorithm

Thanks to the previous definitions, we can express our model (2.7) in the form of the following saddle-point problem, which is an extension (including two extra variables) of the one presented in (Pierre et al., 2015):

$$\begin{aligned} \min_{\zeta \in X} \min_{r \in X} \min_{v \in X} \max_{p \in Y} \max_{q \in Y} \{ &\langle K_1 \zeta, p \rangle + \langle K_2 r, q \rangle - D_1^*(p) - D_2^*(q) \\ &+ A(\zeta) + B(r) + a(\zeta, v) + b(r, v) + C(v) \} . \end{aligned} \quad (\text{A.12})$$

It is a primal-dual problem with three primal variables (ζ , r , and v) and two dual variables (p and q) that evolve independently. Each dual variable is particularly linked to the gradient of a primal variable, i.e. p to ζ , and q to r . D_1^* , D_2^* , A , B , and C are convex functions; a and b are convex *w.r.t.* each of its respective variables. Globally, the functional is not convex *w.r.t.* the triplet (ζ, r, v) . By relating (2.7) and (A.12), and using the primal-dual expression of the regularization term reported in (A.10), we have the following equivalences:

- $K_1\zeta = \tilde{K}\zeta$;
- $K_2r = \tilde{K}r$;
- $D_1^*(p) = -\langle \beta_{\lambda_1}(r), p \rangle_Y + \delta_P(p)$;
- $D_2^*(q) = -\langle \beta_{\lambda_2}(I), q \rangle_Y + \delta_Q(q)$;
- $A(\zeta) = F_1(\zeta|\zeta_S) + \delta_{[\zeta_m, \zeta_M]}(\zeta)$;
- $B(r) = \delta_{[r_m, r_M]}(r)$;
- $a(\zeta, v) = F_2(\zeta, v|\zeta_S)$;
- $b(r, v) = G(r, v|r_S)$;
- $C(v) = H(v|\zeta_S, r_S) + \delta_{[0,1]}(v)$.

An algorithm to solve (A.12) can be derived within the primal-dual optimization framework of (Chambolle and Pock, 2011). It consists in a unique loop, where all variables are alternatively updated via proximal operators (see Algorithm 1). The algorithm takes as inputs the initial estimates of the complete depth and reflectance images (ζ_0 and r_0 , respectively), and the reference intensity image I . It also requires three parameters inherent to the algorithm: σ and τ , which are related to each other by the relation $16\tau\sigma \leq 1$ (Chambolle and Pock, 2011), and ρ , which is a parameter regulating the update speed of v .

Algorithm 1 Primal-dual based algorithm for depth and reflectance joint inpainting.

- 1: **Inputs:**
 $\zeta_0, r_0, I, \sigma, \rho, \tau$
 - 2: **Initialize:**
 $\zeta^0, \bar{\zeta}^0 \leftarrow \zeta_0, r^0, \bar{r}^0 \leftarrow r_0, v_{i,j}^0 \leftarrow 0.5,$
 $p^0 \leftarrow (\nabla \zeta_0, \lambda_1 \nabla r_0), q^0 \leftarrow (\nabla r_0, \lambda_2 \nabla I)$
 - 3: **for** $n = 0, 1, \dots$ **do**
 - 4: $p^{n+1} \leftarrow \text{prox}_{\sigma D_1^*}(p^n + \sigma K_1 \bar{\zeta}^n)$
 - 5: $q^{n+1} \leftarrow \text{prox}_{\sigma D_2^*}(q^n + \sigma K_2 \bar{r}^n)$
 - 6: $v^{n+1} \leftarrow \text{prox}_{\rho a(\bar{\zeta}^n, \cdot) + \rho b(\bar{r}^n, \cdot) + \rho C}(v^n)$
 - 7: $\zeta^{n+1} \leftarrow \text{prox}_{\tau A + \tau a(\cdot, v^{n+1})}(\zeta^n - \tau K_1^* p^{n+1})$
 - 8: $r^{n+1} \leftarrow \text{prox}_{\tau B + \tau b(\cdot, v^{n+1})}(r^n - \tau K_2^* q^{n+1})$
 - 9: $\bar{\zeta}^{n+1} \leftarrow 2\zeta^{n+1} - \zeta^n$
 - 10: $\bar{r}^{n+1} \leftarrow 2r^{n+1} - r^n$
-

Algorithm 1 involves the computation of the adjoints to the linear operators K_1 and K_2 (the “zero-padded” gradient operators). It is known that the adjoint

of the gradient operator is the negative divergence operator ($\nabla^* = -\text{div}$). In our case, the adjoint to the operator $K_1 : X \rightarrow Y$ is a linear operator $K_1^* : Y \rightarrow X$ consisting in the negative divergence computed only on the two first components of a four-component dual variable $p \in Y$, and by taking finite differences in the opposite direction than the gradient operator (A.3). These components are in fact the ones related to the primal variable to which the coupled gradient operator has been applied. We then have the following element-wise definition for K_1^*p (the same definition stands for K_2^*q):

$$(K_1^*p)_{i,j} = - \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{if } 1 < i < M \\ p_{i,j}^1 & \text{if } i = 1 \\ -p_{i-1,j}^1 & \text{if } i = M \end{cases} - \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{if } 1 < j < N \\ p_{i,j}^2 & \text{if } j = 1 \\ -p_{i,j-1}^2 & \text{if } j = N \end{cases}. \quad (\text{A.13})$$

Closed-form expressions for the update rules in Algorithm 1 can be easily computed by applying the definition of proximal operator. The resulting expressions are reported here below, where \mathcal{P} denotes the projection operation over a given real interval, i.e. values are clipped if exceeding the interval limits.

$$\text{prox}_{\sigma D_1^*}(\tilde{p}) = \frac{\tilde{p} + \sigma \beta_{\lambda_1}(r)}{\max(1, \|\tilde{p} + \sigma \beta_{\lambda_1}(r)\|_2)} \quad (\text{A.14})$$

$$\text{prox}_{\sigma D_2^*}(\tilde{q}) = \frac{\tilde{q} + \sigma \beta_{\lambda_2}(w)}{\max(1, \|\tilde{q} + \sigma \beta_{\lambda_2}(w)\|_2)} \quad (\text{A.15})$$

$$\begin{aligned} \text{prox}_{\rho a(\bar{\zeta}, \cdot) + \rho b(\bar{r}, \cdot) + \rho C}(\tilde{v}) = & \\ \begin{cases} \mathcal{P}_{[0,1]}(\tilde{v}) & \text{if } \Phi_{i,j} = 0 \\ \mathcal{P}_{[0,1]}(\tilde{v} + \rho\alpha - \rho\eta_2|\bar{r} - r_S|) & \text{if } \Phi_{i,j} = 1, \bar{\zeta}_{i,j} \geq \zeta_S \zeta_{i,j} \\ \mathcal{P}_{[0,1]}(\tilde{v} + \rho\alpha - \rho\eta_1(\zeta_S - \bar{\zeta}) - \rho\eta_2|\bar{r} - r_S|) & \text{if } \Phi_{i,j} = 1, \bar{\zeta}_{i,j} < \zeta_S \zeta_{i,j} \end{cases} \end{aligned} \quad (\text{A.16})$$

$$\text{prox}_{\tau A + \tau a(\cdot, v)}(\tilde{\zeta}) = \begin{cases} \mathcal{P}_{[\zeta_m, \zeta_M]}(\tilde{\zeta}) & \text{if } \Phi_{i,j} = 0 \\ \mathcal{P}_{[\zeta_m, \zeta_M]}(\tilde{\zeta} - \tau\eta_1) & \text{if } \Phi_{i,j} = 1, \tilde{\zeta}_{i,j} > u_S \zeta_{i,j} + \tau\eta_1 \\ \mathcal{P}_{[\zeta_m, \zeta_M]}(\tilde{\zeta} + v\tau\eta_1) & \text{if } \Phi_{i,j} = 1, \tilde{\zeta}_{i,j} < u_S \zeta_{i,j} - v\tau\eta_1 \\ \mathcal{P}_{[\zeta_m, \zeta_M]}(\zeta_S) & \text{otherwise} \end{cases} \quad (\text{A.17})$$

$$\text{prox}_{\tau B + \tau b(\cdot, v)}(\tilde{r}) = \begin{cases} \mathcal{P}_{[r_m, r_M]}(\tilde{r}) & \text{if } \Phi_{i,j} = 0 \\ \mathcal{P}_{[r_m, r_M]}(\tilde{r} - v\tau\eta_2) & \text{if } \Phi_{i,j} = 1, \tilde{r}_{i,j} > r_{S\ i,j} + v\tau\eta_2 \\ \mathcal{P}_{[r_m, r_M]}(\tilde{r} + v\tau\eta_2) & \text{if } \Phi_{i,j} = 1, \tilde{r}_{i,j} < r_{S\ i,j} - v\tau\eta_2 \\ \mathcal{P}_{[r_m, r_M]}(r_S) & \text{otherwise} \end{cases} \quad (\text{A.18})$$

The operations indicated in the proximal operators are pixel-wise, although the pixel coordinates have not been made explicit for clearer reading. ■

Appendix B

Supplementary experiments of Chapter 2

In order to provide a self-contained document, we include in this appendix further experiments related to Chapter 2. This proof is the result of the work of Marco Bevilacqua.

1 Parameters of the algorithm and model choices

Our finally resulting joint inpainting model (2.7) consists of four terms: two data-fidelity terms, $F(\zeta, v|\zeta_S)$ and $G(r, v|r_S)$, a “removal” cost depending solely on the variable v , $H(v|\zeta_S, r_S)$, and the two-fold regularization term $R(\zeta, r|I)$. As discussed in Section 2.2.1, for the data-fidelity terms we opt for a ℓ_1 measure of the error, in order to promote more contrasted solutions (Chan and Esedoglu, 2005). The visibility attribute v weights the data matching cost of each single pixel (data matching is more and more relaxed, as v tends to zero, i.e. when that particular point is considered to be excluded). However, over-estimated depths ($\zeta > \zeta_S$) are not weighted by v but are fully penalized. These values relate to pixels where either there is noise on a visible point that is slightly corrected ($\zeta - \zeta_S$ is small), or the value u_S represents an outlier (e.g. it is due to a mobile object). At present, we do not have a way to handle the latter case.

In $H(v|\zeta_S, r_S)$ (2.4), each point removal cost is the product between $(1 - v)$ (the level of “invisibility” of the point) and a coefficient α depending on the local input depth and reflectance: $\alpha = k_1\zeta_S + k_2r_S$. This choice has been made in order to balance all terms in (2.7) where v appears. Let us now observe the “complete” update rule for v (last case of (A.16), i.e. for points with under-estimated depth). According to it, we have that at each iteration v is incremented/decremented by a quantity $\Delta v = \rho(\alpha - \eta_1\Delta\zeta - \eta_2\Delta r)$. Let us suppose that the fluctuations on depth are significantly larger than the fluctuations on reflectance (the appearance of a hidden point can cause a big “jump” in depth, while the reflectance values might still be similar. For the sake of simplicity we can then adjust the value of α only on the basis of the depth input value. The proposed simplified expression for α is then:

$$\alpha = k\zeta_S. \tag{B.1}$$

With the assumptions made we therefore have $\Delta v \propto (k\zeta_S - \eta_1\Delta\zeta)$. The attribute v for a certain pixel increases (it gets a higher confidence as a visible point) if $\frac{\Delta\zeta}{\zeta_S} < \frac{k}{\eta_1}$, i.e. if the relative depth deviation is below a certain threshold. k is an adimensional parameter that contributes determining this threshold. Conversely, v decreases for relative depth deviations exceeding the threshold. As for the update of v for points with over-estimated depths (second case of (A.16)), if we hypothesize that α , adjusted on depth, is large enough w.r.t. the reflectance deviation, we have that v progressively tends to one (unless large absolute reflectance deviations occur).

As for the regularization term $R(\zeta, r|I)$, we proposed in Section 2.2.3 to combine two distinct coupled total variation terms: $\text{TV}_{\lambda_1}(\zeta, r)$ (depth is individually coupled with reflectance) and $\text{TV}_{\lambda_2}(r, I)$ (reflectance is individually coupled with the color image). By having two separate coupled TV terms, each one encoded by a dual variable that evolves independently from the other one, the reflectance gradient is constantly brought back to the reference gradient of the color image. At the same time the “correct” gradient information is transferred to the depth via the second term. Figure 2.5 shows an example of results obtained with the algorithm for the same test case as Figure 2.3.

For the example test of Figure 2.5, as well as for all the results reported hereinafter, the following parameters, found with multiple tests, have been used to characterize the model (2.7): $\eta_1 = 1.7$, $\eta_2 = 50$, $k = 0.05$ (the coefficient determining α according to (B.1)), $\lambda_1 = 0.5$, $\lambda_2 = 1$. These values have been found empirically by letting them vary one by one and observing the obtained visual results. The two data terms $F(u, v|u_S)$ and $G(r, v|r_S)$ are attributed different weights. The larger coefficient assigned to the reflectance data term ($\eta_2 > \eta_1$) means that a greater data fidelity is imposed on reflectance. Depth values have instead a greater “freedom” in deviating from their original values. The two coupling parameters λ_1 and λ_2 being in the same order of magnitude, it shows that the two coupling terms have a similar importance. As for the parameters, inherent to the primal-dual optimization scheme (Algorithm 1), the following values have been set after testing: $\rho = 10$, $\tau = 0.004$, $\sigma = 14$.

2 Results with urban data

We consider a data set acquired by a MMS system (Paparoditis et al., 2012) at *Place de la Bastille*, Paris, consisting of one LiDAR point cloud in the order of one billion of points and hundreds of optical images simultaneously acquired by 5 cameras mounted on the vehicle. Given a reference optical image, we project onto it the available LiDAR points to form the initial depth and reflectance incomplete images. Note that not all the points are effectively visible from the image view point. The incomplete depth and reflectance images, along with the reference color image chosen, represent the input of the algorithm (ζ_S , r_S , and I respectively).

Figures B.1–B.4 present results for four images (cropped w.r.t. the full size)

of the data set: *Column1*, *Column2*, *Buildings1*, *Buildings2*. For each reference image, the input sparse depth and reflectance images, obtained via projection, are shown, as well as the inpainted depth and reflectance images, obtained with four different methods. For the output depth images of Figure B.3 and B.4 we added some shading by modulating the color intensity of each pixel based on the zenith angle of the normal vector, to emphasize high-frequency changes. Moreover, for the inpainted depths, an alternative view of the resulting 3-D point cloud is proposed, where the coordinates of the points are retrieved thanks to the computed depths and color texture is applied to enrich the points. A color box is overlaid to the first of these 3-D views to highlight areas where the comparison between the different methods is particularly significant.

Our algorithm, presented in Section A, gives as output the two inpainted images ζ and r . As for the produced depth image, our algorithm is visually compared with nearest neighbor (NN) interpolation, the anisotropic total generalized variation (ATGV) method of (Ferstl et al., 2013b), and our previous depth inpainting method (Bevilacqua et al., 2016), which does not rely on reflectance information. We refer to the latter as Depth Inpainting with Visibility Estimation (*DIVE*). The optimization problem of DIVE is the following:

$$\begin{aligned} \min_{\substack{\zeta \in [\zeta_m, \zeta_M] \\ v \in [0,1]}} & \eta \int_{\Omega_S} (\max(0, \zeta - y))^2 dx_1 dx_2 + \eta \int_{\Omega_S} v(\max(0, y - \zeta))^2 dx_1 dx_2 \\ & + \int_{\Omega_S} (k\zeta_S)^2(1 - v) dx_1 dx_2 + \text{TV}_\lambda(\zeta, I) . \quad (\text{B.2}) \end{aligned}$$

The DIVE problem can be related to our proposed model (2.7), if we consider in the latter $\eta_1 = \eta$, $\eta_2 = 0$, $\lambda_1 = \lambda$, and we suppress the coupled TV term related to the reflectance (depth is instead coupled directly with the color image). Moreover, in (B.2) we have a ℓ_2 -norm data fidelity term; as a consequence of that, the coefficient of the removal cost term follows a quadratic law (we have $\alpha = (ku_S)^2$, instead of $\alpha = ku_S$, as in (2.7)).

As for the produced reflectance image, our algorithm is compared with nearest neighbor (NN) interpolation, the ATGV method of (Ferstl et al., 2013b) applied to reflectance, and a reduced version of our model (2.7) limited to reflectance. We refer to this method as Reflectance Inpainting with Visibility Estimation (*RIVE*). The RIVE method is derived from the solution of the following optimization problem:

$$\min_{\substack{r \in [r_m, r_M] \\ v \in [0,1]}} \eta \int_{\Omega_S} v|r - r_S| dx_1 dx_2 + \int_{\Omega_S} (kr_S)(1 - v) dx_1 dx_2 + \text{TV}_\lambda(r, I) . \quad (\text{B.3})$$

Also in this case we can derive the considered problem (RIVE) as a simplified version of our proposed model (2.7), where $\eta_1 = 0$, $\eta_2 = \eta$, $\lambda_2 = \lambda$, and the coupled TV term related to depth is suppressed. Moreover, the coefficient of the removal cost, while still following a linear law, here depends on the input reflectance r_S .

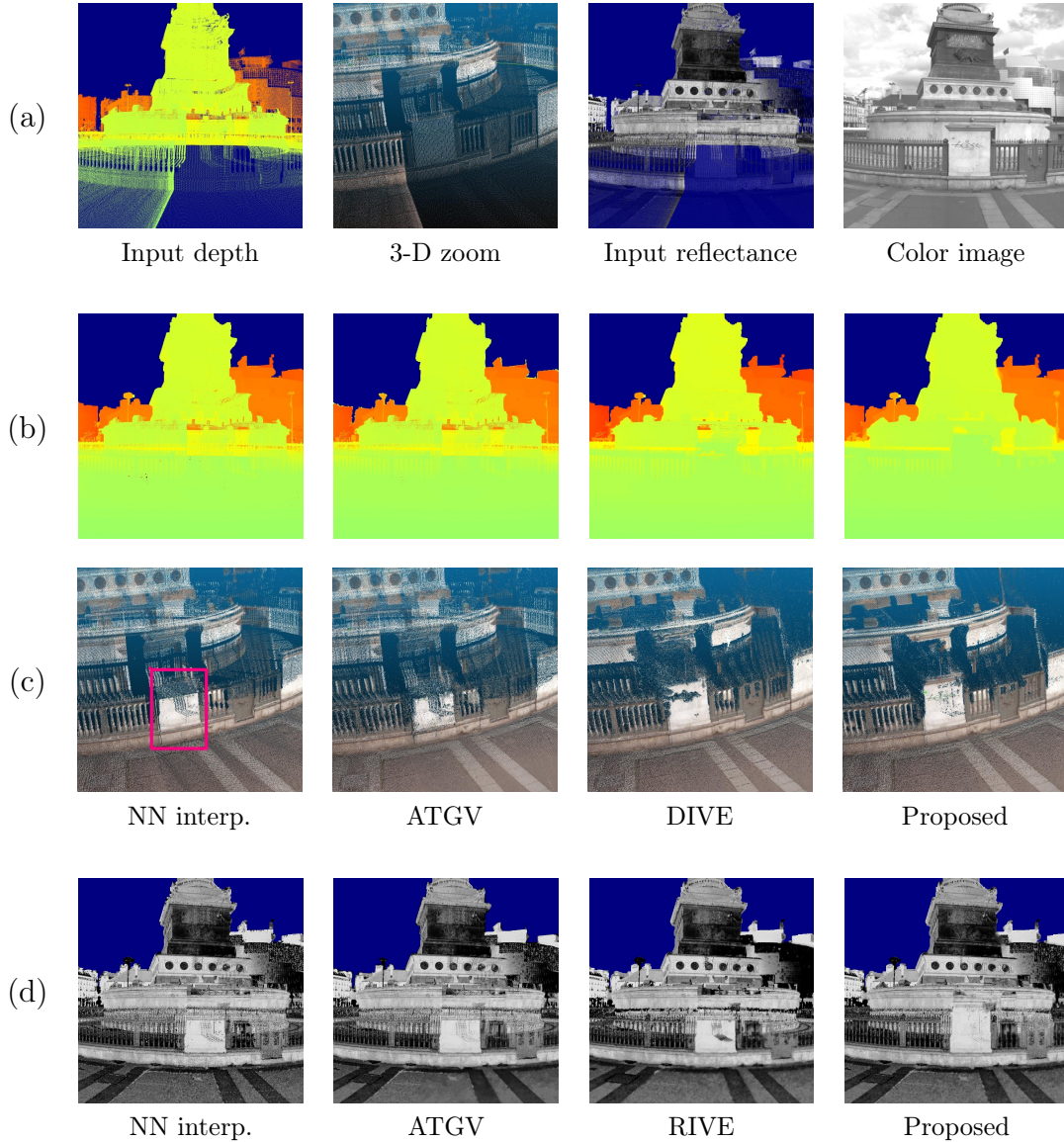


Figure B.1: Visual results for the image *Column1*. Row (a) shows the related input images: depth (with a 3-D zoom), reflectance, and reference color image. Rows (b) and (c) report the results obtained in terms of inpainted depth images (with related 3-D zoomed-in view) with the algorithms indicated below. Row (d) shows the inpainted reflectance images obtained with different methods, our proposed method always reported as last.

The four examples reported show the better performance of our algorithm in generating complete depth and reflectance images from real LiDAR measures. Results with the image *Column1*, reported in Figure B.1, particularly prove the effectiveness of our algorithm in detecting and removing hidden points appearing in the front, thus producing inpainted images correct from the image view point. These points,

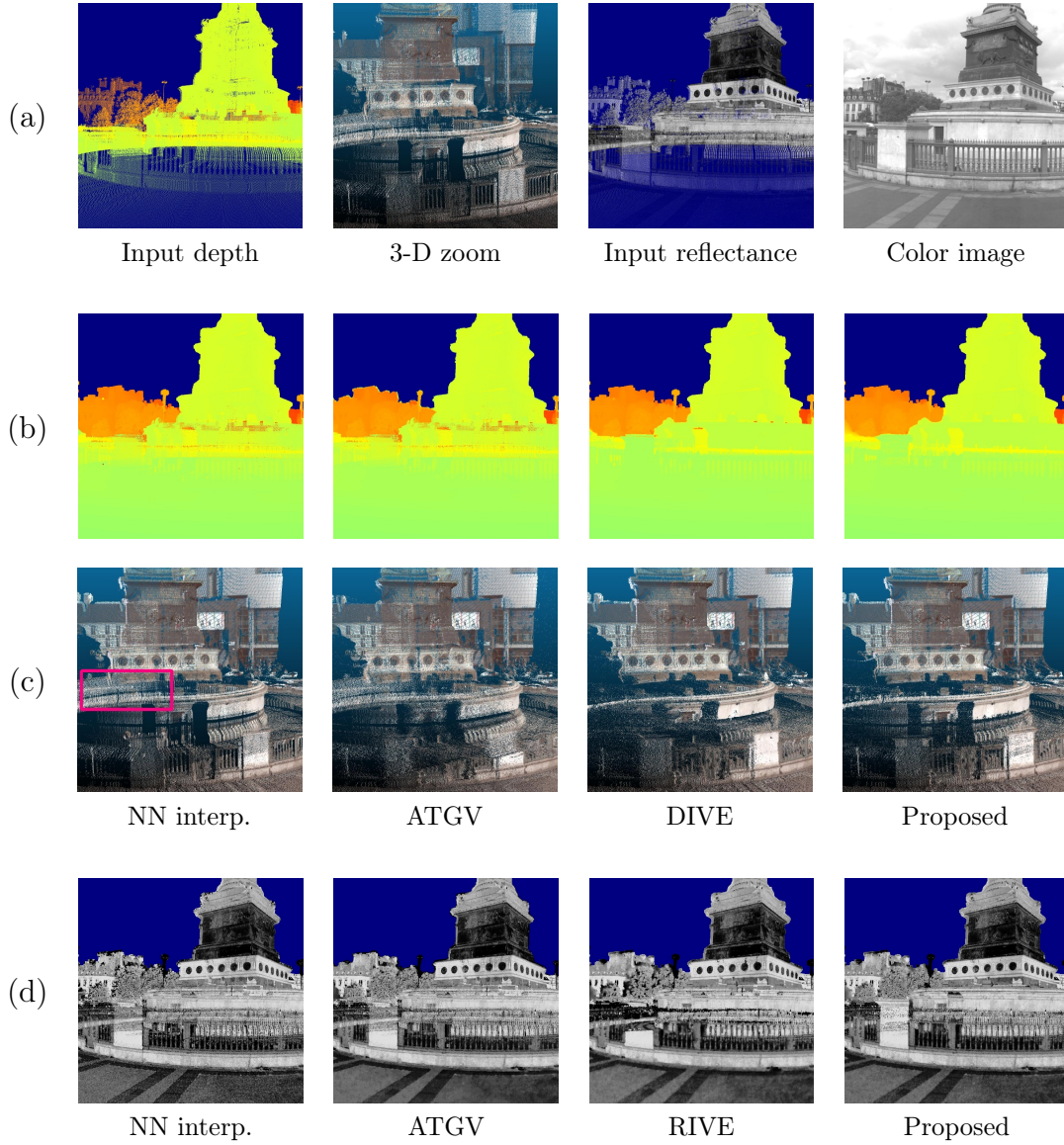


Figure B.2: Visual results for the image *Column2*. Row (a) shows the related input images: depth (with a 3-D zoom), reflectance, and reference color image. Rows (b) and (c) report the results obtained in terms of inpainted depth images (with related 3-D zoomed-in view) with the algorithms indicated below. Row (d) shows the inpainted reflectance images obtained with different methods, our proposed method always reported as last.

in yellow/orange according to the color code used for depth, appear mixed to visible points belonging to the column and the fence. By looking at the depth images generated (row (b)), our algorithm is the only one which is able to remove the misleading points and correctly reconstruct the foreground depth plane. This is even more visible by observing the main marble pole highlighted in the 3-D views (row (c)).

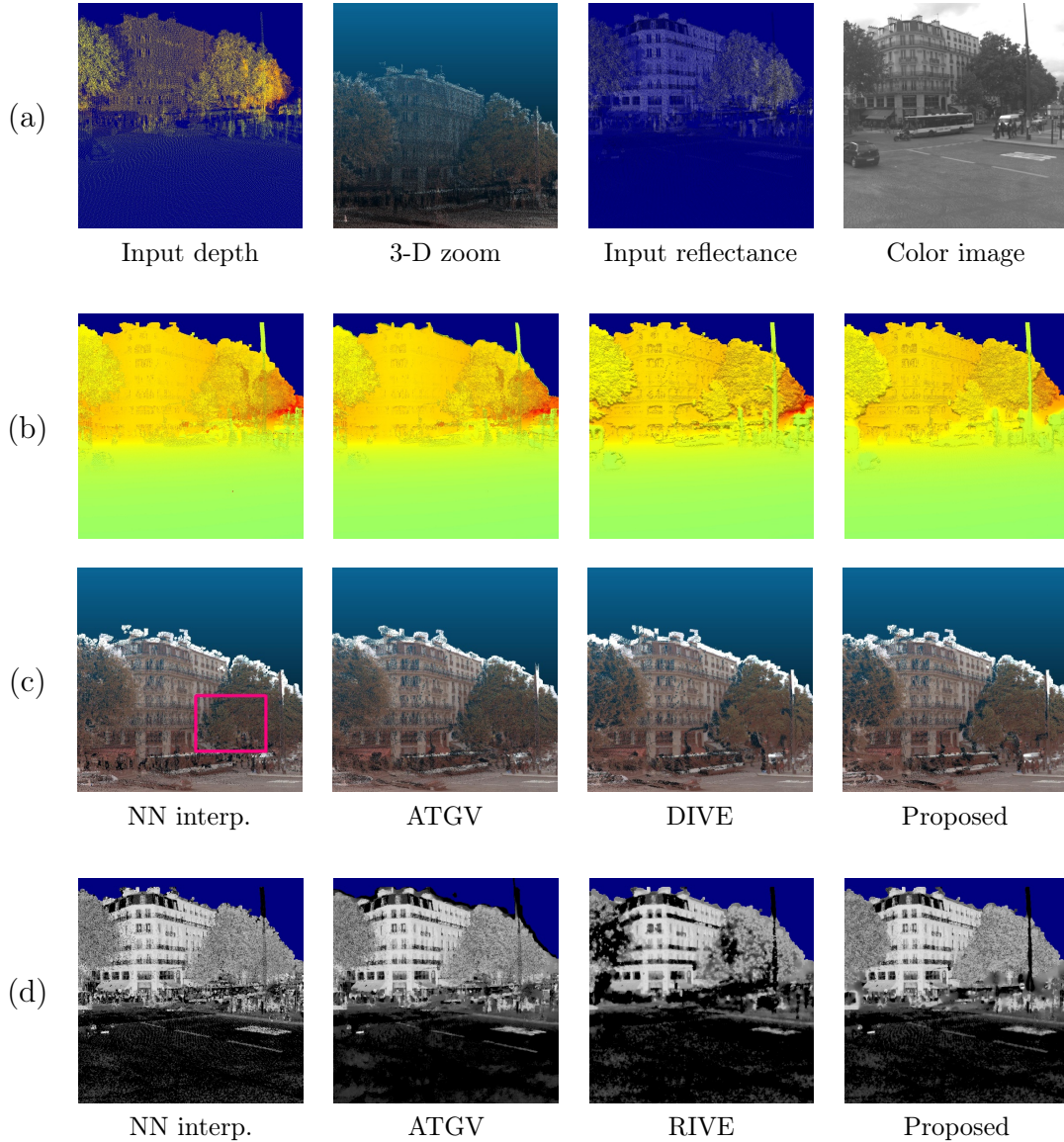


Figure B.3: Visual results for the image *Buildings1*. Row (a) shows the related input images: depth (with a 3-D zoom), reflectance, and reference color image. Rows (b) and (c) report the results obtained in terms of inpainted depth images (with related 3-D zoomed-in view) with the algorithms indicated below. Row (d) shows the inpainted reflectance images obtained with different methods, our proposed method always reported as last.

While other methods are not able to reconstruct the pole, since “distracted” by the interfering background depths, the reconstruction is better performed in our case. Results on the reflectance image confirm the trend. By observing again the main marble pole, we clearly see that the reflectance is better inpainted. This is possible thanks to the joint use of depth information, which helps detecting hidden points

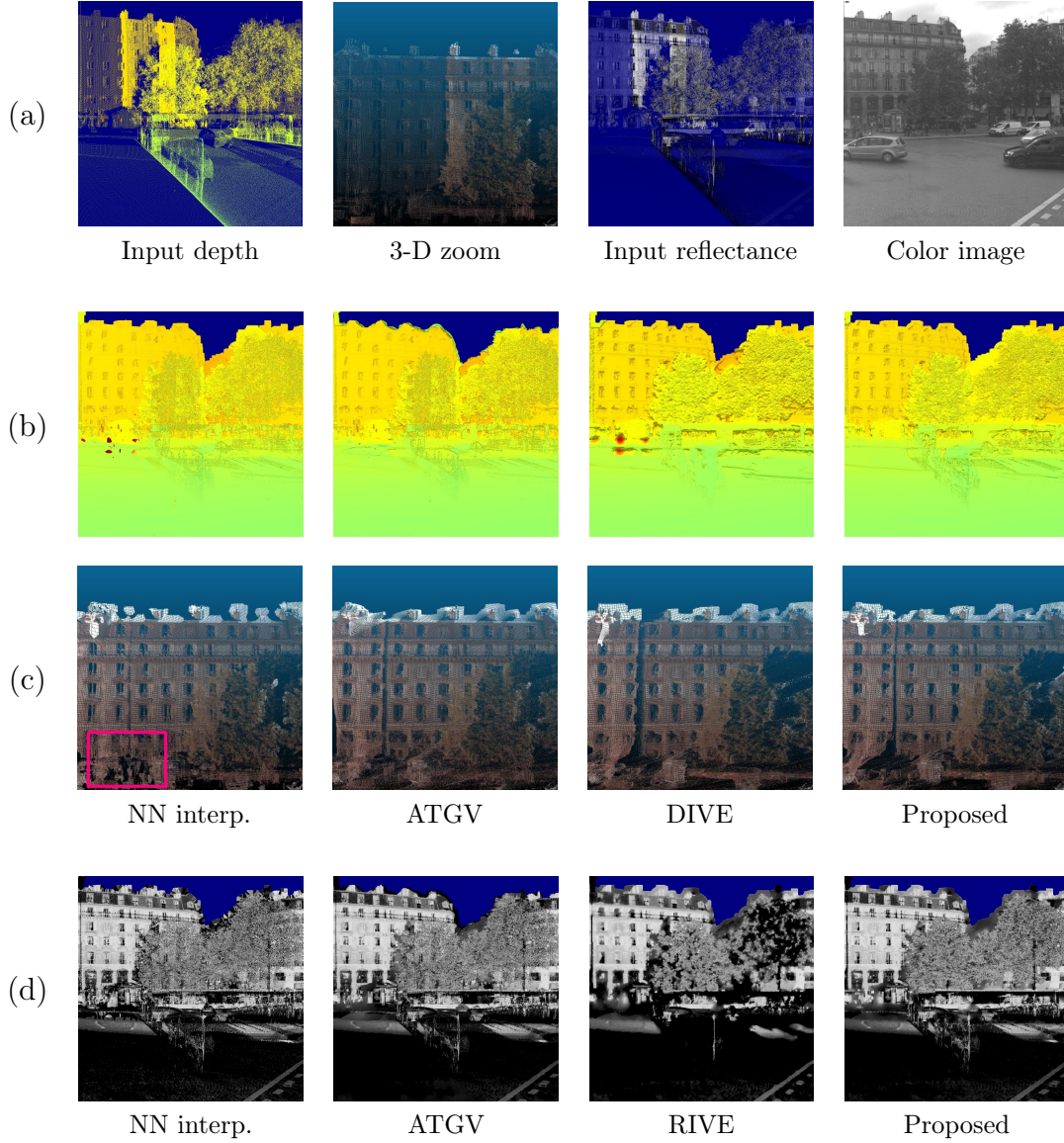


Figure B.4: Visual results for the image *Buildings2*. Row (a) shows the related input images: depth (with a 3-D zoom), reflectance, and reference color image. Rows (b) and (c) report the results obtained in terms of inpainted depth images (with related 3-D zoomed-in view) with the algorithms indicated below. Row (d) shows the inpainted reflectance images obtained with different methods, our proposed method always reported as last.

by leveraging depth over- and under-estimations, and the coupling with the color image gradient, which helps correctly restoring the edges. Similar considerations can be made for the image *Column2* (visual results are reported in Figure B.2). Here the box overlaid on the 3-D views indicates an area where points, non-visible from the reference image view point, should be removed. The removal of these points, as

well as the inpainting of depth and reflectance, is performed more efficiently by our method.

Figures B.3 and B.4 show results w.r.t. two other images taken peripherally to the scene. For the image *Buildings1*, we can observe that with our algorithm the inpainted depth and reflectance images look more satisfactory, the pole on the left being completely unveiled as a foreground element. The box overlaid on the 3-D views highlights a part of the scene where the depth values of two trees interfere. Our proposed algorithm (as well as the DIVE method (Bevilacqua et al., 2016)) makes a correct distinction between the two depth layers. Figure B.4, reporting results related to the image *Buildings2*, presents the problem of wrong LiDAR measures appearing in the front. Our method turns out to be the most effective one in clearing out these points, as also shown in the area highlighted by the box.

3 Performance on visibility estimation

While in the previous section we evaluated the performance of the algorithm in terms of produced inpainted images ζ and r , we now want to assess the quality of the third output of the algorithm, i.e. v , the visibility attribute.

As visibility is estimated while performing the depth and reflectance estimation, we can say that our algorithm fuses two problems: hidden point removal (HPR) and inpainting. Typically HPR is, instead, possibly performed as a preliminary operation. For HPR “stand-alone” the state of the art is represented by variations of (Katz et al., 2007) that relate the visible point set to the convex hull of a viewpoint-dependent transformation of it, discarding points based on a concavity threshold as seen from the view point. While this approach is effective, there is in general no globally satisfactory concavity threshold that would both correctly detect hidden surfaces and keep background points close to foreground silhouettes. To compare the two strategies for estimating visibility (the dedicated operation of (Katz et al., 2007) and our “soft” estimation), we show an example in Figure B.5, related to the image *Column1*. In our case, we consider as hidden points those depth values that are assigned $v = 0$ at the end of the algorithm. As for (Katz et al., 2007), a concavity parameter equal to 4 has been chosen after tuning.

The images obtained show that the “quality” of the visibility estimation process is comparable, if not higher with our method. If we observe closely the zoomed-in areas in Figure B.5, in fact, we can see that the HPR method wrongly selects points around the silhouettes (see first patch), while sometimes missing the detection of actual hidden points (see last two patches).

As a further test, we also compare our method (which jointly performs visibility estimation and inpainting), with a two-step approach, where visibility estimation (hidden point removal) is performed as a preliminary operation by the algorithm of (Katz et al., 2007). Depth is subsequently inpainted with the ATGV-based algorithm of (Ferstl et al., 2013b). Figure B.6 reports results for such comparison with two

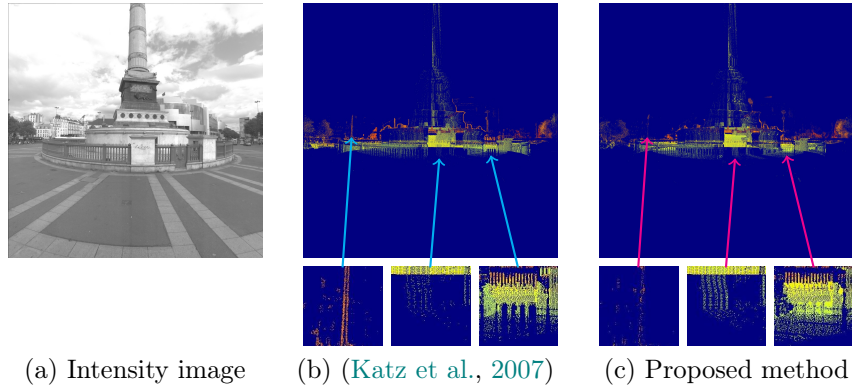


Figure B.5: Detected hidden points in the case of the image *Column1*, by the state-of-the-art method of (Katz et al., 2007) and our method. The three patches below each image represent zoomed-in areas of the images themselves at same locations.

images, the two-step approach being denoted as “HPR + ATGV”.

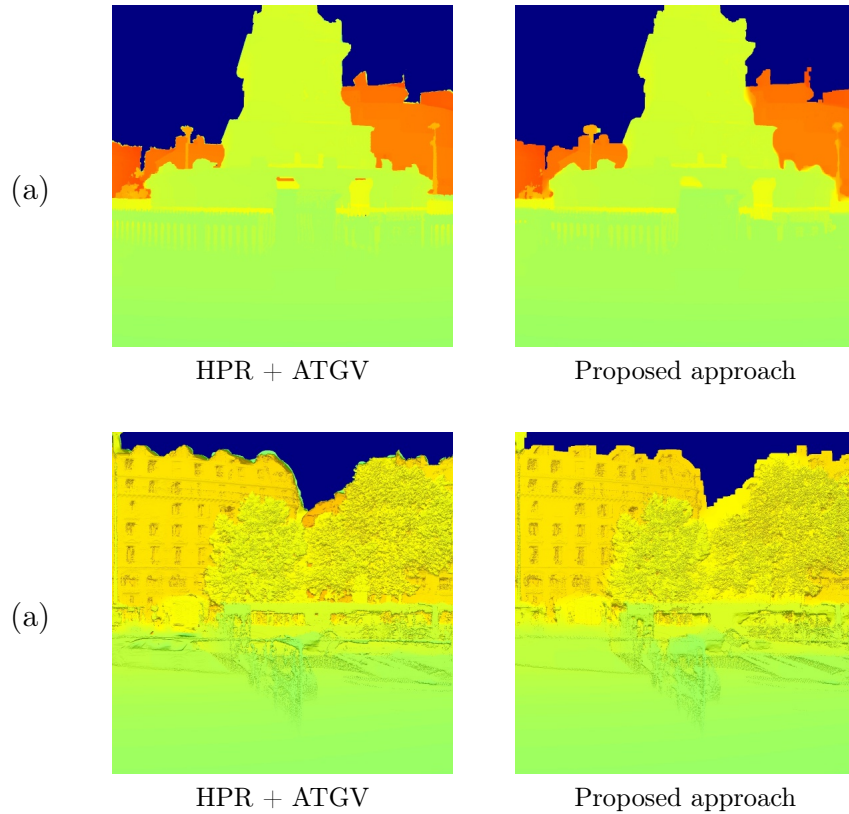


Figure B.6: Comparison between our joint approach and a two-step approach, where visibility estimation and inpainting are performed separately, on the images *Column1* (a) and *Buldings2* (b).

In the two cases of Figure B.6, we can observe a better outcome with our algorithm. For the image *Column1*, the preliminary point removal operation is not able to remove all the ambiguities in the central part of the image, where the depth values of the fence and the column are confused. For the image *Buildings2*, the HPR method of (Katz et al., 2007) exceeds in removing several points along the upper board of the image, causing blurred edges in the final reconstructed depth image. Besides the benefits observable in the qualitative assessment, the joint approach of our method has the advantage of not requiring an explicit parameter to be globally set (the concavity threshold in the case of (Katz et al., 2007)) to perform HPR. This is instead done in a “soft” way that adapts to the input image. ■

Bibliography

- [Abayowa et al. 2015] ABAYOWA, B. O. ; YILMAZ, A. ; HARDIE, R. C. : Automatic registration of optical aerial imagery to a LiDAR point cloud for generation of city models. In : *ISPRS Journal of Photogrammetry and Remote Sensing* 106 (2015), n. 1, pp. 68–81
- [Achanta et al. 2012] ACHANTA, R. ; SHAJI, A. ; SMITH, K. ; LUCCHI, A. ; FUA, P. ; SÜSTRUNK, S. : SLIC superpixels compared to state-of-the-art superpixel methods. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 34 (2012), n. 11, pp. 2274–2282
- [Aiger et al. 2008] AIGER, D. ; MITRA, N. J. ; COHEN-OR, D. : 4-points congruent sets for robust pairwise surface registration. In : *ACM Transactions on Graphics* 27, 2008, pp. 85–91
- [Allaire et al. 2008] ALLAIRE, S. ; KIM, J. J. ; BREEN, S. L. ; JAFFRAY, D. A. ; PEKAR, V. : Full orientation invariance and improved feature selectivity of 3D SIFT with application to medical image analysis. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008, pp. 1–8
- [Aubert and Kornprobst 2006] AUBERT, G. ; KORNPBST, P. : *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Springer, 2006
- [Auclair-Fortier and Ziou 2006] AUCLAIR-FORTIER, M-F. ; ZIOU, D. : A global approach for solving evolutive heat transfer for image denoising and inpainting. In : *IEEE Trans. on Image Processing* 15 (2006), pp. 2558–2574
- [Badrinarayanan et al. 2017] BADRINARAYANAN, V. ; KENDALL, A. ; CIPOLLA, R. : Segnet: A deep convolutional encoder-decoder architecture for image segmentation. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 39 (2017), n. 12, pp. 2481–2495
- [Barber et al. 1996] BARBER, C. B. ; DOBKIN, D. P. ; HUHDANPAA, H. : The quickhull algorithm for convex hulls. In : *ACM Transactions on Mathematical Software* 22 (1996), n. 4, pp. 469–483
- [Barnes et al. 2009] BARNES, C. ; SHECHTMAN, E. ; FINKELSTEIN, A. ; GOLDMAN, D. B. : PatchMatch: A randomized correspondence algorithm for structural image editing. In : *ACM Trans. Graph.* 28 (2009)
- [Bay et al. 2006] BAY, H. ; TUYTELAARS, T. ; VAN GOOL, L. : SURF: Speeded Up Robust Features. In : *Proc. of ECCV*, 2006, pp. 404–417
- [Becker et al. 2009] BECKER, J. ; STEWART, C. ; RADKE, R. J. : LiDAR inpainting from a single image. In : *IEEE Int. Conf. on Computer Vision* 1, 2009, pp. 1441–1448
- [Benenson et al. 2014] BENENSON, R. ; OMRAN, M. ; HOSANG, J. ; SCHIELE, B. : Ten years of pedestrian detection, what have we learned? In : *ECCV European Conference on Computer Vision*, 2014, pp. 613–627
- [Berger et al. 2017] BERGER, M. ; TAGLIASACCHI, A. ; SEVERSKY, L. M. ; ALLIEZ, P. ; GUENNEBAUD, G. ; LEVINE, J. A. ; SHARF, A. ; SILVA, C. T. : A survey of surface reconstruction from point clouds. In : *Computer Graphics Forum*, 2017, pp. 301–329

-
- [Bertalmio et al. 2000] BERTALMIO, M. ; SAPIRO, G. ; CASELLES, V. ; BALLESTER, C. : Image inpainting. In : *ACM Comp. graphics and interactive techniques* 1, 2000, pp. 417–424
- [Besl and McKay 1992] BESL, P. J. ; MCKAY, N. D. : Method for registration of 3D shapes. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 1611, 1992, pp. 586–607
- [Bevilacqua et al. 2016] BEVILACQUA, M. ; AUJOL, J-F. ; BRÉDIF, M. ; BUGEAU, A. : Visibility Estimation and Joint Inpainting of Lidar Depth Maps. In : *IEEE Int. Conf. on Image Processing*, 2016, pp. 1–5
- [Bevilacqua et al. 2017] BEVILACQUA, M. ; BIASUTTI, P. ; AUJOL, J-F. ; BRÉDIF, M. ; BUGEAU, A. : Joint inpainting of depth and reflectance with visibility estimation. In : *IJPRS International Journal of Photogrammetry, Remote Sensing and Spatial Information Sciences* 125 (2017), pp. 16–32
- [Biasutti et al. 2016] BIASUTTI, P. ; AUJOL, J-F. ; BRÉDIF, M. ; BUGEAU, A. : Diffusion anisotrope et inpainting d’orthophotographies LiDAR mobile. In : *RFIA Congr s national sur la Reconnaissance des Formes et l’Intelligence Artificielle*, 2016
- [Biasutti et al. 2017a] BIASUTTI, P. ; AUJOL, J-F. ; BR DIF, M. ; BUGEAU, A. : Disocclusion of 3D LiDAR point clouds using range images. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 4 (2017), n. 1, pp. 75–82
- [Biasutti et al. 2017b] BIASUTTI, P. ; AUJOL, J-F. ; BR DIF, M. ; BUGEAU, A. : D soccultation de nuage de points LiDAR en topologie capteur. In : *GRETSI Groupement de Recherche en Traitement du Signal et de l’Image*, 2017
- [Biasutti et al. 2018] BIASUTTI, P. ; AUJOL, J-F. ; BR DIF, M. ; BUGEAU, A. : Range-Image: Incorporating sensor topology for LiDAR point cloud processing. In : *Photogrammetric Engineering & Remote Sensing* 84 (2018), n. 6, pp. 367–375
- [Biasutti et al. 2019a] BIASUTTI, P. ; AUJOL, J-F. ; BR DIF, M. ; BUGEAU, A. : Diffusion and inpainting of reflectance and height LiDAR orthoimages. In : *Computer Vision and Image Understanding* 179 (2019), n. 1, pp. 31–40
- [Biasutti et al. 2019b] BIASUTTI, P. ; AUJOL, J-F. ; BR DIF, M. ; BUGEAU, A. : D tection et localisation d’objets 3D par apprentissage profond en topologie capteur. In : *GRETSI Groupement de Recherche en Traitement du Signal et de l’Image*, 2019
- [Biasutti et al. 2019c] BIASUTTI, P. ; AUJOL, J-F. ; BR DIF, M. ; BUGEAU, A. : Fast image and LiDAR alignment based on 3D rendering in sensor topology. In : *Pattern Recognition Letters (under review)* (2019)
- [Biasutti et al. 2019d] BIASUTTI, P. ; AUJOL, J-F. ; BR DIF, M. ; BUGEAU, A. : Visibility Estimation in Point Clouds with Variable Density. In : *VISAPP International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019, pp. 27–35
- [Biasutti et al. 2019e] BIASUTTI, P. ; BUGEAU, A. ; AUJOL, J-F ; BR DIF, M. : *RIU-Net: Embarrassingly simple semantic segmentation of 3D LiDAR point cloud*. 2019
- [Bichen et al. 2018] BICHEN, W. ; ZHOU, X. ; ZHAO, S. ; YUE, X. ; KEUTZER, K. : Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud. In : *arXiv preprint: 1809.08495* (2018)
-

- [Bitenc et al. 2011] BITENC, M. ; LINDENBERGH, R. ; KHOSHELHAM, K. ; VAN WAARDEN, A. P. : Evaluation of a LiDAR land-based mobile mapping system for monitoring sandy coasts. In : *Remote Sensing* 3 (2011)
- [Bletterer 2018] BLETTERER, A. : *Une approche basée graphes pour la modélisation et le traitement de nuages de points massifs issus d'acquisitions de LiDARs terrestres*, Université de Nice, PhD Thesis, December 2018
- [Bouchiba et al. 2017] BOUCHIBA, H. ; GROSCOT, R. ; DESCHAUD, J-E. ; GOULETTE, F. : High quality and efficient direct rendering of massive real-world point clouds. In : *Eurographics Annual Conference of the European Association for Computer Graphics*, 2017, pp. 1–6
- [Bredies et al. 2010] BREDIES, K. ; KUNISCH, K. ; POCK, T. : Total generalized variation. In : *SIAM Journal on Imaging Sciences* 3 (2010), pp. 492–526
- [Brédif 2013] BRÉDIF, M. : Image-Based Rendering of LOD1 3D City Models for traffic-augmented Immersive Street-view Navigation. In : *IJPRS International Journal of Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2013), n. 3, pp. 7–11
- [Brédif et al. 2015] BRÉDIF, M. ; VALLET, B. ; FERRAND, B. : Distributed Dimensionality-Based Rendering of LiDAR Point Clouds. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 3 (2015), pp. 559–564
- [Bresenham 1965] BRESENHAM, J. E. : Algorithm for computer control of a digital plotter. In : *IBM Systems journal* 4 (1965), pp. 25–30
- [Briot et al. 2018] BRIOT, A. ; VISWANATH, P. ; YOGAMANI, S. : Analysis of efficient CNN design techniques for semantic segmentation. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 663–672
- [Buyssens et al. 2015a] BUYSENS, P. ; DAISY, M. ; TSCHUMPERLÉ, D. ; LÉZORAY, O. : Depth-aware patch-based image disocclusion for virtual view synthesis. In : *SIGGRAPH International Conference on Computer Graphics and Interactive Techniques* 34, 2015, pp. 2–6
- [Buyssens et al. 2015b] BUYSENS, P. ; DAISY, M. ; TSCHUMPERLÉ, D. ; LÉZORAY, O. : Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions. In : *IEEE Trans. on Image Processing* 24 (2015), n. 6, pp. 1809–1824
- [Castorena et al. 2016] CASTORENA, J. ; KAMILOV, U. S. ; BOUFONOS, P. T. : Autocalibration of LiDAR and optical cameras via edge alignment. In : *ICASSP*, 2016, pp. 2862–2866
- [Chambolle and Pock 2011] CHAMBOLLE, A. ; POCK, T. : A first-order primal-dual algorithm for convex problems with applications to imaging. In : *Journal of Mathematical Imaging and Vision* 40 (2011), n. 1, pp. 120–145
- [Chan et al. 2008] CHAN, D. ; BUISMAN, H. ; THEOBALT, C. ; THRUN, S. : A Noise-Aware Filter for Real-Time Depth Upsampling. In : *Proc. of ECCV*, 2008, pp. 1–12
- [Chan and Esedoglu 2005] CHAN, T. F. ; ESEDOGLU, S. : Aspects of Total Variation Regularized L1 Function Approximation. In : *SIAM Journal on Imaging Sciences* 65 (2005), n. 5, pp. 1817–1837
- [Chan et al. 1999] CHAN, T. F. ; GOLUB, G. H. ; MULET, P. : A Nonlinear Primal-Dual Method for Total Variation-Based Image Restoration. In : *SIAM Journal on Imaging Sciences* 20 (1999), n. 6, pp. 1964–1977

-
- [Chen et al. 2012] CHEN, H. ; CHENG, M. ; LI, J. ; LIU, Y. : An iterative terrain recovery approach to automated DTM generation from airborne LiDAR point clouds. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 39 (2012)
- [Chen et al. 2018a] CHEN, L-C. ; PAPANDREOU, G. ; KOKKINOS, I. ; MURPHY, K. ; YUILLE, A. L. : Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 40 (2018), n. 4, pp. 834–848
- [Chen et al. 2018b] CHEN, L-C. ; ZHU, Y. ; PAPANDREOU, G. ; SCHROFF, F. ; HARTWIG, A. : Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In : *Proc. of ECCV*, 2018, pp. 801–808
- [Chen et al. 2016] CHEN, Q. ; WANG, H. ; ZHANG, H. ; SUN, M. ; LIU, X. : A point cloud filtering approach to generating DTMs for steep mountainous areas and adjacent residential areas. In : *Remote Sensing* 8 (2016)
- [Chen et al. 2017a] CHEN, X. ; MA, H. ; WAN, J. ; LI, B. ; XIA, T. : Multi-view 3D object detection network for autonomous driving. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017
- [Chen et al. 2017b] CHEN, Z. ; GAO, B. ; DEVEREUX, B. : State-of-the-art: DTM generation using airborne LIDAR data. In : *Sensors* 17 (2017)
- [Corsini et al. 2013] CORSINI, M. ; DELLEPIANE, M. ; GANOVELLI, F. ; GHERARDI, R. ; FUSIELLO, A. ; SCOPIGNO, R. : Fully automatic registration of image sets on approximate geometry. In : *Int. Jour. of Computer Vision* 102 (2013), n. 1, pp. 91–111
- [Criminisi et al. 2004] CRIMINISI, A. ; PÉREZ, P. ; TOYAMA, K. : Region filling and object removal by exemplar-based image inpainting. In : *IEEE Trans. on Image Processing* 13 (2004), n. 9, pp. 1200–1212
- [Dai et al. 2016] DAI, J. ; LI, Y. ; HE, K. ; SUN, J. : R-FCN: Object detection via region-based fully convolutional networks. In : *Advances in Neural Inf. Proc. Sys.*, 2016
- [Daribo and Pesquet-Popescu 2010] DARIBO, I. ; PESQUET-POPESCU, B. : Depth-aided image inpainting for novel view synthesis. In : *IEEE Trans. on Image Processing* 1, 2010, pp. 167–170
- [Delon et al. 2007] DELON, J. ; DESOLNEUX, A. ; LISANI, J-L. ; PETRO, A. B. : A nonparametric approach for histogram segmentation. In : *IEEE Trans. on Image Processing* 16 (2007), n. 1, pp. 253–261
- [Doria and Radke 2012] DORIA, D. ; RADKE, R. : Filling large holes in LiDAR data by inpainting depth gradients. In : *Int. Conf. on Pattern Recognition* 1, 2012, pp. 65–72
- [Efros and Leung 1999] EFROS, A. A. ; LEUNG, T. K. : Texture synthesis by non-parametric sampling. In : *IEEE Conf. on Computer Vision and Pattern Recognition* 2, 1999, pp. 1033–1038
- [Eigen et al. 2014] EIGEN, D. ; PUHRSCH, C. ; FERGUS, R. : Depth map prediction from a single image using a multi-scale deep network. In : *Advances in neural information processing systems*, 2014, pp. 2366–2374
- [Everingham et al. 2010] EVERINGHAM, M. ; VAN G., Luc ; WILLIAMS, C. K. ; WINN, J. ; ZISSERMAN, A. : The Pascal Visual Object Classes (VOC) challenge. In : *Int. Jour. of Computer Vision* 88 (2010), n. 2
-

- [Feng et al. 2014] FENG, C. ; TAGUCHI, Y. ; KAMAT, V. R. : Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In : *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6218–6225
- [Ferstl et al. 2013a] FERSTL, D. ; REINBACHER, C. ; RANFTL, R. ; RÜTHER, M. ; BISCHOF, H. : Image guided depth upsampling using anisotropic total generalized variation. In : *IEEE Int. Conf. on Computer Vision* 1, 2013, pp. 993–1000
- [Ferstl et al. 2013b] FERSTL, D. ; REINBACHER, C. ; RANFTL, R. ; RÜTHER, M. ; BISCHOF, H. : Image Guided Depth Usampling using Anisotropic Total Generalized Variation. In : *IEEE Int. Conf. on Computer Vision*, 2013, pp. 993–1000
- [Fischler and Bolles 1981] FISCHLER, M. A. ; BOLLES, R. C. : Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In : *Communications of the ACM* 24 (1981), n. 6, pp. 381–395
- [Friedman et al. 1977] FRIEDMAN, J. H. ; BENTLEY, J. L. ; FINKEL, R. A. : An algorithm for finding best matches in logarithmic expected time. In : *ACM Transactions on Mathematical Software* 3 (1977), n. 3, pp. 209–226
- [Fu et al. 2018] FU, H. ; GONG, M. ; WANG, C. ; BATMANGHELICH, K. ; TAO, D. : Deep ordinal regression network for monocular depth estimation. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 2002–2011
- [Furukawa and Ponce 2010] FURUKAWA, Y. ; PONCE, J. : Accurate, dense, and robust multiview stereopsis. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 32 (2010), n. 8, pp. 1362–1376
- [Garcia et al. 2010] GARCIA, F. ; MIRBACH, B. ; OTTERSTEN, B. ; GRANDIDIER, F. ; CUESTA, A. : Pixel weighted average strategy for depth sensor data fusion. In : *IEEE Int. Conf. on Image Processing*, 2010, pp. 2805–2808
- [Gehring et al. 2017] GEHRUNG, J. ; HEBEL, M. ; ARENS, M. ; STILLA, U. : An approach to extract moving objects from MLS data using a volumetric background representation. In : *ISPRS International Annals of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 4 (2017), pp. 107–114
- [Geiger et al. 2013] GEIGER, A. ; LENZ, P. ; STILLER, C. ; URTASUN, R. : Vision meets Robotics: The KITTI Dataset. In : *IJRR International Journal of Robotics Research* 32 (2013), n. 11, pp. 1231–1237
- [Geiger et al. 2012] GEIGER, A. ; LENZ, P. ; URTASUN, R. : Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012
- [Girshick 2015] GIRSHICK, R. : Fast R-CNN. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015
- [González et al. 2009] GONZÁLEZ, D. ; RODRÍGUEZ-GONZÁLEZ, P. ; GÓMEZ-LAHOZ, J. : An automatic procedure for co-registration of terrestrial laser scanners and digital cameras. In : *ISPRS Journal of Photogrammetry and Remote Sensing* 64 (2009), n. 3, pp. 308–316
- [Guan et al. 2014] GUAN, H. ; LI, J. ; YU, Y. ; ZHONG, L. ; JI, Z. : DEM generation from LiDAR data in wooded mountain areas by cross-section-plane analysis. In : *International Journal of Remote Sensing* 35 (2014)

-
- [Guinard and Vallet 2018] GUINARD, S. ; VALLET, B. : Sensor-topology based simplicial complex reconstruction from mobile laser scanning. In : *ISPRS International Annals of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences IV-2* (2018), pp. 121–128
- [Guislain et al. 2017] GUISLAIN, M. ; DIGNE, J. ; CHAINE, R. ; MONNIER, G. : Fine scale image registration in large-scale urban LIDAR point sets. In : *Computer Vision and Image Understanding* 157 (2017), n. 1, pp. 90–102
- [Hackel et al. 2017] HACKEL, T. ; SAVINOV, N. ; LADICKY, L. ; WEGNER, J. D. ; SCHINDLER, K. ; POLLEFEYS, M. : Semantic3D.net: A new large-scale point cloud classification benchmark. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* (2017)
- [Harrison and Newman 2010] HARRISON, A. ; NEWMAN, P. : Image and Sparse Laser Fusion for Dense Scene Reconstruction. In : *Field and Service Robotics (FRS)*, 2010, pp. 219–228
- [Hervieu and Soheilian 2013a] HERVIEU, A. ; SOHEILIAN, B. : Road side detection and reconstruction using LiDAR sensor. In : *IEEE Intelligent Vehicles Symposium* 4, 2013, pp. 1247–1252
- [Hervieu and Soheilian 2013b] HERVIEU, A. ; SOHEILIAN, B. : Semi-automatic road/pavement modeling using mobile laser scanning. In : *ISPRS International Annals of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 2, 2013
- [Hervieu et al. 2015] HERVIEU, A. ; SOHEILIAN, B. ; BRÉDIF, M. : Road Marking Extraction Using a Model&Data-driven RJ-MCMC. In : *ISPRS International Annals of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 2 (2015), n. 3, pp. 47–48
- [Himmelsbach et al. 2008] HIMMELSBACH, M. ; MUELLER, A. ; LÜTTEL, T. ; WÜNSCHE, H-J. : LIDAR-based 3D object perception. In : *Proceedings of international workshop on Cognition for Technical Systems*, 2008, pp. 1–7
- [Hu et al. 2018] HU, J. ; SHEN, L. ; SUN, G. : Squeeze-and-excitation networks. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141
- [Hu et al. 2015] HU, X. ; YE, L. ; PANG, S. ; SHAN, J. : Semi-global filtering of airborne LiDAR data for fast extraction of digital terrain models. In : *Remote Sensing* 7 (2015)
- [Huang et al. 2013] HUANG, H. ; WU, S. ; COHEN-OR, D. ; GONG, M. ; ZHANG, H. ; LI, G. ; CHEN, B. : L1-medial skeleton of point cloud. In : *ACM Transactions on Graphics*, 2013, pp. 65–72
- [Huang and Menq 2001] HUANG, J. ; MENQ, C-H. : Automatic data segmentation for geometric feature extraction from unorganized 3D coordinate points. In : *IEEE Trans. on Robotics and Automation* 17 (2001), n. 3, pp. 268–279
- [Huhle et al. 2010] HUHLE, B. ; SCHAIRER, T. ; JENKE, P. ; STRASSER, W. : Fusion of range and color images for denoising and resolution enhancement with a non-local filter. In : *Computer Vision and Image Understanding* 114 (2010), n. 12, pp. 1336–1345
- [Iandola et al. 2016] IANDOLA, F. N. ; HAN, S. ; MOSKEWICZ, M. W. ; ASHRAF, K. ; DALLY, W. J. ; KEUTZER, K. : SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. In : *arXiv preprint: 1602.07360* (2016)
- [Katz et al. 2007] KATZ, S. ; TAL, A. ; BASRI, R. : Direct visibility of point sets. In : *ACM Transactions on Graphics*, 2007, pp. 24–36
-

- [Kingma and Ba 2014] KINGMA, D. P. ; BA, J. : *Adam: A method for stochastic optimization*. 2014
- [Koenderink 1984] KOENDERINK, Jan J. : The structure of images. In : *Biological cybernetics* 50 (1984), pp. 363–370
- [Kolb et al. 2010] KOLB, A. ; BARTH, E. ; KOCH, R. ; LARSEN, R. : Time-of-Flight Cameras in Computer Graphics. In : *Computer Graphics Forum* 29, 2010, pp. 141–159
- [Kraus and Pfeifer 2001] KRAUS, K. ; PFEIFER, N. : Advanced DTM generation from LiDAR data. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 34 (2001), pp. 23–30
- [Ku et al. 2018] KU, J. ; MOZIFIAN, M. ; LEE, J. ; HARAKEH, A. ; WASLANDER, S. L. : Joint 3D proposal generation and object detection from view aggregation. In : *IEEE Int. Conf. on Intel. Robots and Systems*, 2018
- [Lafarge and Alliez 2013] LAFARGE, F. ; ALLIEZ, P. : Surface reconstruction through point set structuring. In : *Computer Graphics Forum*, 2013, pp. 225–234
- [Landrieu and Boussaha 2019] LANDRIEU, L. ; BOUSSAHA, M. : Point Cloud Oversegmentation with Graph-Structured Deep Metric Learning. In : *arXiv preprint: 1904.02113* (2019)
- [Ledig et al. 2017] LEDIG, C. ; THEIS, L. ; HUSZÁR, F. ; CABALLERO, J. ; CUNNINGHAM, A. ; ACOSTA, A. ; AITKEN, A. ; TEJANI, A. ; TOTZ, J. ; WANG, Z. et al. : Photo-realistic single image super-resolution using a generative adversarial network. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017
- [Lepetit et al. 2009] LEPETIT, V. ; MORENO-NOGUER, F. ; FUA, P. : E-PnP: An accurate $\mathcal{O}(n)$ solution to the PnP problem. In : *Int. Jour. of Computer Vision* 81 (2009), n. 2, pp. 155
- [Li 2017] LI, B. : 3D fully convolutional network for vehicle detection in point cloud. In : *IEEE Trans. on Intelligent Robots and Systems*, 2017, pp. 1513–1518
- [Li et al. 2011a] LI, Y. ; WU, X. ; CHRYSATHOU, Y. ; SHARF, A. ; COHEN-OR, D. ; MITRA, N. J. : Globfit: Consistently fitting primitives by discovering global relations. In : *ACM Trans. on Graphics*, 2011, pp. 52–64
- [Li et al. 2011b] LI, Y. ; ZHENG, Q. ; SHARF, A. ; COHEN-OR, D. ; CHEN, B. ; MITRA, N. J. : 2D-3D fusion for layer decomposition of urban facades. In : *IEEE Int. Conf. on Computer Vision*, 2011, pp. 882–889
- [Liang et al. 2019] LIANG, M. ; YANG, B. ; CHEN, Y. ; HU, R. ; URTASUN, R. : Multi-Task Multi-Sensor Fusion for 3D Object Detection. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 7345–7353
- [Lin et al. 2017a] LIN, G. ; MILAN, A. ; SHEN, C. ; REID, I. : Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 1925–1934
- [Lin et al. 2017b] LIN, T-Y. ; DOLLÁR, P. ; GIRSHICK, R. ; HE, K. ; HARIHARAN, B. ; BELONGIE, S. : Feature pyramid networks for object detection. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017

-
- [Lin et al. 2017c] LIN, T-Y. ; GOYAL, P. ; GIRSHICK, R. ; HE, K. ; DOLLÁR, P. : Focal loss for dense object detection. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017
- [Lin et al. 2014] LIN, T-Y. ; MAIRE, M. ; BELONGIE, S. ; HAYS, J. ; PERONA, P. ; RAMANAN, D. ; DOLLÁR, P. ; ZITNICK, C. L. : Microsoft COCO: Common objects in context. In : *Proc. of ECCV*, 2014
- [Lipman et al. 2007] LIPMAN, Y. ; COHEN-OR, D. ; LEVIN, D. ; TAL-EZER, H. : Parameterization-free projection for geometry reconstruction. In : *ACM Trans. on Graphics*, 2007, pp. 22–28
- [Liu et al. 2016] LIU, W. ; ANGUELOV, D. ; ERHAN, D. ; SZEGEDY, C. ; REED, S. ; FU, C-Y. ; BERG, A. C. : SSD: Single shot multibox detector. In : *Proc. of ECCV*, 2016
- [Long et al. 2015] LONG, J. ; SELHAMER, E. ; DARRELL, T. : Fully convolutional networks for semantic segmentation. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440
- [Lorenzi et al. 2011] LORENZI, L. ; MELGANI, F. ; MERCIER, G. : Inpainting strategies for reconstruction of missing data in VHR images. In : *IEEE Geoscience and Remote Sensing Letters* 8 (2011), n. 5, pp. 914–918
- [Lowe 2004] LOWE, D. G. : Distinctive image features from scale-invariant keypoints. In : *Int. Jour. of Computer Vision* 60 (2004), n. 2, pp. 91–110
- [Luo et al. 2018] LUO, W. ; YANG, B. ; URTASUN, R. : Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018
- [Maddern et al. 2017] MADDERN, Will ; PASCOE, Geoff ; LINEGAR, Chris ; NEWMAN, Paul : 1 Year, 1000km: The Oxford RobotCar Dataset. In : *The International Journal of Robotics Research (IJRR)* 36 (2017), n. 1, pp. 3–15
- [Mastin et al. 2009] MASTIN, A. ; KEPNER, J. ; FISHER, J. : Automatic registration of LIDAR and optical images of urban scenes. In : *CVPR IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2639–2646
- [Mehra et al. 2010] MEHRA, R. ; TRIPATHI, P. ; SHEFFER, A. ; MITRA, N. J. : Visibility of noisy point cloud data. In : *Computers and Graphics* 34 (2010), n. 3, pp. 219–230
- [Meng et al. 2010] MENG, X. ; CURRIT, N. ; ZHAO, K. : Ground filtering algorithms for airborne LiDAR data: A review of critical issues. In : *Remote Sensing* 2 (2010)
- [Menze and Geiger 2015] MENZE, M. ; GEIGER, A. : Object Scene Flow for Autonomous Vehicles. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3061–3070
- [Mikolajczyk and Schmid 2005] MIKOLAJCZYK, K. ; SCHMID, C. : A performance evaluation of local descriptors. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 27 (2005), n. 10, pp. 1615–1630
- [Miled et al. 2016] MILED, M. ; SOHEILIAN, B. ; HABETS, E. ; VALLET, B. : Hybrid online mobile laser scanner calibration through image alignment by mutual information. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 3 (2016), n. 1, pp. 25
-

- [Monszpart et al. 2015] MONSZPART, A. ; MELLADO, N. ; BROSTOW, G. J. ; MITRA, N. J. : RAPter: rebuilding man-made scenes with regular arrangements of planes. In : *ACM Trans. on Graphics*, 2015, pp. 103:1–103:12
- [Morel and Yu 2009] MOREL, J-M. ; YU, G. : ASIFT: A new framework for fully affine invariant image comparison. In : *SIAM Journal on Imaging Sciences* 2 (2009), n. 2, pp. 438–469
- [Moussa et al. 2012] MOUSSA, W. ; ABDEL-WAHAB, M. ; FRITSCH, D. : An automatic procedure for combining digital images and laser scanner data. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 39 (2012), n. 1
- [Muja and Lowe 2014] MUJA, M. ; LOWE, D. G. : Scalable Nearest Neighbor Algorithms for High Dimensional Data. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 36 (2014), n. 11, pp. 2227–2240
- [Nikolova 2004] NIKOLOVA, M. : A Variational Approach to Remove Outliers and Impulse Noise. In : *Journal of Mathematical Imaging and Vision* 20 (2004), n. 1-2, pp. 99–120
- [Paganelli et al. 2012] PAGANELLI, C. ; PERONI, M. ; PENNATI, F. ; BARONI, G. ; SUMMERS, P. et al. : Scale Invariant Feature Transform as feature tracking method in 4D imaging: a feasibility study. In : *IEEE International Conference of Engineering in Medicine and Biology Society*, 2012, pp. 6543–6546
- [Paparoditis et al. 2012] PAPARODITIS, N. ; PAPELARD, J-P. ; CANNELLE, B. ; DEVAUX, A. ; SOHEILIAN, B. ; DAVID, N. ; HOUZAY, E. : Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology. In : *Revue française de photogrammétrie and de télédétection* 200 (2012), pp. 69–79
- [Papon et al. 2013] PAPON, J. ; ABRAMOV, A. ; SCHOELER, M. ; WORGOTTER, F. : Voxel cloud connectivity segmentation-supervoxels for point clouds. In : *IEEE Conf. on Computer Vision and Pattern Recognition* 1, 2013, pp. 2027–2034
- [Park et al. 2011] PARK, J. ; KIM, H. ; TAI, Y-W. ; BROWN, M. S. ; KWEON, I. : High Quality Depth Map Upsampling for 3D-TOF Cameras. In : *IEEE Int. Conf. on Computer Vision*, 2011, pp. 1623–1630
- [Park et al. 2005] PARK, S. ; GUO, X. ; SHIN, H. ; QIN, H. : Shape and appearance repair for incomplete point surfaces. In : *IEEE Int. Conf. on Computer Vision* 2, 2005, pp. 1260–1267
- [Pérez et al. 2003] PÉREZ, P ; GANGNET, M ; BLAKE, A : Poisson image editing. In : *ACM Trans. on Graphics* 22, 2003, pp. 313–318
- [Perona and Malik 1990] PERONA, P. ; MALIK, J. : Scale-space and edge detection using anisotropic diffusion. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 12 (1990), pp. 629–639
- [Pierre et al. 2015] PIERRE, F. ; AUJOL, J-F. ; BUGEAU, A. ; PAPADAKIS, N. ; TA, V-T. : Luminance-Chrominance Model for Image Colorization. In : *SIAM Journal on Imaging Sciences* 8 (2015), n. 1, pp. 536–563
- [Pintus et al. 2011] PINTUS, R. ; GOBBETTI, E. ; AGUS, M. : Real-time rendering of massive unstructured raw point clouds using screen-space operators. In : *Eurographics International Conference on Virtual Reality, Archaeology and Cultural Heritage*, 2011, pp. 105–112

-
- [Pu et al. 2011] PU, S. ; RUTZINGER, M. ; VOSSELMAN, G. ; ELBERINK, S. O. : Recognizing basic structures from mobile laser scanning data for road inventory studies. In : *IJPRS International Journal of Photogrammetry, Remote Sensing and Spatial Information Sciences* 66 (2011), pp. 28–39
- [Qi et al. 2018] QI, C. R. ; LIU, W. ; WU, C. ; SU, H. ; GUIBAS, L. J. : Frustum pointnets for 3D object detection from RGB-D data. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018
- [Qi et al. 2017] QI, C. R. ; SU, H. ; MO, K. ; GUIBAS, L. J. : Pointnet: Deep learning on point sets for 3D classification and segmentation. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 652–660
- [Qu et al. 2015] QU, X. ; SOHEILIAN, B. ; PAPARODITIS, N. : Vehicle localization using mono-camera and geo-referenced traffic signs. In : *IEEE Intelligent Vehicles Symposium* 4, 2015, pp. 605–610
- [Rabbani et al. 2006] RABBANI, T. ; VAN DEN HEUVEL, F. ; VOSSELMANN, G. : Segmentation of point clouds using smoothness constraint. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 36 (2006), n. 5, pp. 248–253
- [Rabin et al. 2011] RABIN, J. ; PEYRÉ, G. ; DELON, J. ; BERNOT, M. : Wasserstein barycenter and its application to texture mixing. In : *Scale Space and Variational Methods in Computer Vision*. Springer, 2011
- [Redmon et al. 2016] REDMON, J. ; DIVVALA, S. ; GIRSHICK, R. ; FARHADI, A. : You only look once: Unified, real-time object detection. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 779–788
- [Redmon and Farhadi 2017] REDMON, J. ; FARHADI, A. : YOLO9000: Better, Faster, Stronger. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017
- [Ren et al. 2017] REN, J. ; CHEN, X. ; LIU, J. ; SUN, W. ; PANG, J. ; YAN, Q. ; TAI, Y-W. ; XU, L. : Accurate single stage detector using recurrent rolling convolution. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017
- [Ren et al. 2015] REN, S. ; HE, K. ; GIRSHICK, R. ; SUN, J. : Faster R-CNN: Towards real-time object detection with region proposal networks. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 39 (2015), n. 6
- [Ren et al. 2013] REN, Z. ; YUAN, J. ; MENG, J. ; ZHANG, Z. : Robust part-based hand gesture recognition using kinect sensor. In : *IEEE Transaction on Multimedia* 15 (2013), n. 5, pp. 1110–1120
- [Ronneberger et al. 2015] RONNEBERGER, O. ; FISCHER, P. ; BROX, T. : U-net: Convolutional networks for biomedical image segmentation. In : *MICCAI International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241
- [Roshni and Revathy 2008] ROSHNI, V. ; REVATHY, K. : Using mutual information and cross correlation as metrics for registration of images. In : *Journal of Theoretical & Applied Information Technology* 4 (2008), n. 6
- [Rottensteiner and Briese 2002] ROTTENSTEINER, F. ; BRIESE, C. : A new method for building extraction in urban areas from high-resolution LiDAR data. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 34 (2002), pp. 295–301
-

- [Rubinstein et al. 2008] RUBINSTEIN, M. ; SHAMIR, A. ; AVIDAN, S. : Improved seam carving for video retargeting. In : *ACM Trans. on Graphics* 17, 2008, pp. 16:1–16:9
- [Ruble et al. 2011] RUBLEE, Ethan ; RABAUD, Vincent ; KONOLIGE, Kurt ; BRADSKI, Gary : ORB: An efficient alternative to SIFT or SURF. In : *IEEE Int. Conf. on Computer Vision*, 2011, pp. 2564–2571
- [Schmeing and Jiang 2011] SCHMEING, M. ; JIANG, X. : Depth Image Based Rendering. In : *Pattern Recognition, Machine Intelligence and Biometrics*. 2011, pp. 279–310
- [Schnabel et al. 2009] SCHNABEL, R. ; DEGENER, P. ; KLEIN, R. : Completion and reconstruction with primitive shapes. In : *Computer Graphics Forum*, 2009, pp. 503–512
- [Schnabel et al. 2007] SCHNABEL, R. ; WAHL, R. ; KLEIN, R. : RANSAC based out-of-core point-cloud shape detection for city-modeling. In : *Proceedings of "Terrestrisches Laserscanning"* 26 (2007), pp. 214–226
- [Schneider et al. 2016] SCHNEIDER, N. ; SCHNEIDER, L. ; PINGGERA, P. ; FRANKE, U. ; POLLEFEYS, M. ; STILLER, C. : Semantically Guided Depth Upsampling. In : *German Conference on Pattern Recognition (GCPR)*, 2016, pp. 37–48
- [Serna and Marcotegui 2013] SERNA, A. ; MARCOTEGUI, B. : Urban accessibility diagnosis from mobile laser scanning data. In : *IJPRS International Journal of Photogrammetry, Remote Sensing and Spatial Information Sciences* 84 (2013), pp. 23–32
- [Serna and Marcotegui 2014] SERNA, A. ; MARCOTEGUI, B. : Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. In : *IJPRS International Journal of Photogrammetry, Remote Sensing and Spatial Information Sciences* 93 (2014), pp. 243–255
- [Serra 1982] SERRA, J. : *Image analysis and mathematical morphology*. 1. Academic Press, 1982
- [Shalom et al. 2010] SHALOM, S. ; SHAMIR, A. ; ZHANG, H. ; COHEN-OR, D. : Cone carving for surface reconstruction. In : *ACM Transactions on Graphics*, 2010, pp. 150–160
- [Shan and Toth 2008] SHAN, J. ; TOTH, C. K. : *Topographic laser ranging and scanning: principles and processing*. CRC Press, 2008
- [Shao and Chen 2008] SHAO, Y.-C. ; CHEN, L.-C. : Automated searching of ground points from airborne LiDAR data using a climbing and sliding method. In : *Photogrammetric Engineering & Remote Sensing* 74 (2008)
- [Sharf et al. 2004] SHARF, A. ; ALEXA, M. ; COHEN-OR, D. : Context-based surface completion. In : *ACM Trans. on Graphics* 23 (2004), n. 3, pp. 878–887
- [Shi et al. 2018] SHI, S. ; WANG, X. ; LI, H. : PointRCNN: 3D object proposal generation and detection from point cloud. In : *arXiv preprint: 1812.04244* (2018)
- [Sutour et al. 2015] SUTOUR, C. ; AUJOL, J-F. ; DELEDALLE, C-A. ; SENNEVILLE, B. Denis de : Edge-based multi-modal registration and application for night vision devices. In : *Journal of Mathematical Imaging and Vision* 53 (2015), n. 2, pp. 131–150

-
- [Tagliasacchi et al. 2011] TAGLIASACCHI, A. ; OLSON, M. ; ZHANG, H. ; HAMARNEH, G. ; COHEN-OR, D. : VASE: Volume-Aware Surface Evolution for Surface Reconstruction from Incomplete Point Clouds. In : *Computer Graphics Forum*, 2011, pp. 1563–1571
- [Toews et al. 2013] TOEWS, M. ; ZÖLLEI, L. ; WELLS, W. M. : Feature-based alignment of volumetric multi-modal images. In : *International Conference on Information Processing in Medical Imaging*, 2013, pp. 25–36
- [Tournaire et al. 2006] TOURNAIRE, O. ; SOHEILIAN, B. ; PAPARODITIS, N. : Towards a sub-decimeter georeferencing of groundbased mobile mapping systems in urban areas: Matching ground-based and aerial-based imagery using roadmarks. In : *IJPRS International Journal of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36 (2006)
- [Tschumperlé 2006] TSCHUMPERLÉ, D. : Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. In : *Int. Jour. of Computer Vision* 68 (2006), n. 1, pp. 65–82
- [Vallet 2016] VALLET, B. : *Analyse et reconstruction de scènes urbaines*, Université Paris-Est, Habilitation à diriger des recherches, December 2016
- [Vallet et al. 2015] VALLET, B. ; BRÉDIF, M. ; SERNA, A. ; MARCOTEGUI, B. ; PAPARODITIS, N. : TerraMobilita/IQmulus urban point cloud analysis benchmark. In : *Computers and Graphics* 49 (2015), pp. 126–133
- [Vallet and Papelard 2015] VALLET, B. ; PAPELARD, J-P. : Road orthophoto/DTM generation from mobile laser scanning. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 3 (2015), n. 1
- [Velas et al. 2018] VELAS, M. ; SPANEL, M. ; HRADIS, M. ; HEROUT, A. : Cnn for very fast ground segmentation in velodyne LiDAR data. In : *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2018, pp. 97–103
- [Viola and Wells III 1997] VIOLA, P. ; WELLS III, W. M. : Alignment by maximization of mutual information. In : *Int. Jour. of Computer Vision* 24 (1997), n. 2, pp. 137–154
- [Wack and Wimmer 2002] WACK, R. ; WIMMER, A. : Digital terrain models from airborne laserscanner data-a grid based approach. In : *ISPRS Arch. of the Photogrammetry, Remote Sens. and Spatial Inf. Sciences* 34 (2002)
- [Wang et al. 2008] WANG, L. ; JIN, H. ; YANG, R. ; GONG, M. : Stereoscopic inpainting: Joint color and depth completion from stereo images. In : *IEEE Conf. on Computer Vision and Pattern Recognition* 1, 2008, pp. 1–8
- [Weickert 1998] WEICKERT, J. : *Anisotropic diffusion in image processing*. 1. Teubner Stuttgart, 1998
- [Wu et al. 2018] WU, Bichen ; WAN, Alvin ; YUE, Xiangyu ; KEUTZER, Kurt : Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d LiDAR point cloud. In : *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1887–1893
- [Xiong et al. 2014] XIONG, S. ; ZHANG, J. ; ZHENG, J. ; CAI, .J ; LIU, L. : Robust surface reconstruction via dictionary learning. In : *ACM Trans. on Graphics*, 2014, pp. 201–205
-

- [Yan et al. 2018] YAN, Y. ; MAO, Y. ; LI, B. : Second: Sparsely embedded convolutional detection. In : *Sensors* 18 (2018), n. 10
- [Yang et al. 2016] YANG, F. ; CHOI, W. ; LIN, Y. : Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016
- [Yang et al. 2013] YANG, Q. ; AHUJA, N. ; YANG, R. ; TAN, K-H. ; DAVIS, J. ; CULBERTSON, B. ; APOSTOLOPOULOS, J. ; WANG, G. : Fusion of median and bilateral filtering for range image upsampling. In : *IEEE Trans. on Image Processing* 22 (2013), n. 12, pp. 4841–4852
- [Yeh et al. 2017] YEH, R. A. ; CHEN, C. ; YIAN LIM, T. ; SCHWING, A. G. ; HASEGAWA-JOHNSON, M. ; DO, M. N. : Semantic image inpainting with deep generative models. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 5485–5493
- [Yuan et al. 2018] YUAN, W. ; TIANYUE, S. ; PENG, Y. ; LEI, T. ; MING, L. : Pointseg: Real-time semantic segmentation based on 3d LiDAR point cloud. In : *arXiv preprint: 1807.06288* (2018)
- [Zach et al. 2007] ZACH, C. ; POCK, T. ; BISCHOF, H. : A globally optimal algorithm for robust TV-L 1 range image integration. In : *ICCV IEEE International Conference on Computer Vision*, 2007, pp. 1–8
- [Zakšek and Pfeifer 2006] ZAKŠEK, K. ; PFEIFER, N. : An improved morphological filter for selecting relief points from a LIDAR point cloud in steep areas with dense vegetation. In : *Technical report: Institute of Anthropological and Spatial Studies, Ljubljana, Slovenia* (2006)
- [Zhang and Lin 2013] ZHANG, J. ; LIN, X. : Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification. In : *ISPRS Journal of Photogrammetry and Remote Sensing* 81 (2013)
- [Zhang 2012] ZHANG, Z. : Microsoft Kinect sensor and its effect. In : *IEEE International Conference on Multimedia and Exposition* 19 (2012), n. 2, pp. 4–10
- [Zhao et al. 2018] ZHAO, H. ; QI, X. ; SHEN, X. ; SHI, J. ; JIA, J. : Icnnet for real-time semantic segmentation on high-resolution images. In : *Proc. of ECCV*, 2018, pp. 405–420
- [Zhou and Tuzel 2018] ZHOU, Y. ; TUZEL, O. : Voxelnet: End-to-end learning for point cloud based 3D object detection. In : *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499
- [Zhu et al. 2010] ZHU, X. ; ZHAO, H. ; LIU, Y. ; ZHAO, Y. ; ZHA, H. : Segmentation and classification of range image from an intelligent vehicle in urban environment. In : *IEEE Trans. on Intelligent Robots and Systems* 1, 2010, pp. 1457–1462
- [Zhuang and Bioucas-Dias 2018] ZHUANG, L. ; BIOUCAS-DIAS, J. M. : Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations. In : *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11 (2018)
- [Zinger and Do 2010] ZINGER, S. ; DO, L. : Free-viewpoint depth image based rendering. In : *Journal of Visual Communication and Image Representation* 21 (2010), n. 5, pp. 533–541

Table of contents

Résumé en Français	3
General introduction	12
I Image processing on sparse projection of 3D LiDAR point clouds	20
1 Orthoimage generation from onground LiDAR acquisition	24
1.1 Introduction	25
1.1.1 DSM generation from LiDAR data	26
1.1.2 Orthophotography from LiDAR data	27
1.2 Framework description	28
1.3 Projection of LiDAR point cloud	29
1.3.1 Filtering ground points	30
1.3.2 Sparse projections	31
1.3.3 Parameters	31
1.3.4 Dependency to the sensor	32
1.4 Diffusion of sparse images	32
1.4.1 Choice of the approach and requirements	32
1.4.2 Proposed algorithm	33
1.4.3 Comparison with other diffusion techniques	34
1.4.4 Parameters	35
1.5 Inpainting of occlusions	35
1.5.1 Occlusion hole detection	36
1.5.2 Exemplar-based inpainting	36
1.5.3 Modification to the original algorithm	37
1.5.4 Parameters	39
1.6 Results	39
1.6.1 Parameters	39
1.6.2 Qualitative analysis	40
1.6.3 Quantitative analysis	41
1.6.4 Computational speed	44
1.7 Conclusion and future work	48

2	Dense depth map from sparse projection	49
2.1	Introduction	50
2.1.1	Addressed problem and related works	51
2.2	Model	53
2.2.1	Visibility-weighted data-fidelity terms	54
2.2.2	Removal cost	55
2.2.3	Coupled Total Variation	56
2.3	Experimental results	57
2.3.1	Quantitative evaluation with a benchmark data set	59
2.4	Conclusion	61
3	Visibility estimation of a point cloud from a given point of view	62
3.1	Introduction	63
3.2	Related works	64
3.3	Visibility estimation method	66
3.4	Visibility estimation dataset for LiDAR point clouds	68
3.4.1	Overview of the dataset	69
3.5	Experiments & Results	70
3.5.1	Evaluation on the Visibility Estimation Dataset	70
3.5.2	Evaluation on constant density point cloud	74
3.5.3	Example of application to data fusion	77
3.6	Conclusion	77
	Part conclusion	78
II	Image processing on 3D LiDAR point clouds in sensor topology	81
4	Dense 2D representation of a 3D LiDAR point cloud	86
4.1	Problem statement	87
4.2	Range-images derived from the sensor topology	87
4.2.1	Sensor topology	88
4.2.2	From sensor topology to range-image	88
4.3	Interest and applications	90
5	Point cloud to image registration	92
5.1	Introduction	93
5.2	Mutli-modal alignment	94
5.2.1	Mutli-modal image registration	94
5.2.2	LiDAR to optical registration	95
5.3	Methodology	97
5.3.1	Fast mesh reconstruction in sensor topology	97
5.3.2	Depth to optical image alignment	99

5.4	Experiments and results	102
5.4.1	Quantitative analysis	102
5.4.2	Qualitative analysis	104
5.5	Conclusion	105
6	Object segmentation	106
6.1	Introduction	107
6.2	Point cloud segmentation	107
6.2.1	Region segmentation	107
6.2.2	Semantic segmentation	108
6.2.3	Tradeoff between region and semantic segmentation	110
6.3	Proposed region segmentation method	110
6.3.1	Methodology	110
6.3.2	Results & Analysis	112
6.4	Proposed semantic segmentation method	114
6.4.1	Input of the network	114
6.4.2	Architecture	114
6.4.3	Loss function	117
6.4.4	Training	117
6.5	Experiments	117
6.6	Conclusion	119
7	Object removal	121
7.1	Problem statement	122
7.2	Object removal methods	122
7.2.1	Image object removal	122
7.2.2	Point cloud object removal	124
7.3	Range-image disocclusion technique	124
7.4	Results & Analysis	127
7.4.1	Sparse point cloud	127
7.4.2	Dense point cloud	129
7.4.3	Quantitative analysis	130
7.4.4	Overlapping objects	132
7.5	Conclusion	133
8	Object detection	134
8.1	Introduction	135
8.2	2D detection architecture for 3D detection	136
8.3	Methodology	137
8.3.1	3D detection and localization	137
8.3.2	2D detection on optical image	140
8.3.3	Projection and fusion of the predictions	141
8.4	Results	141

8.4.1	Qualitative analysis	141
8.4.2	Quantitative analysis	141
8.5	Conclusion	143
Part conclusion		144
General conclusion and perspectives		149
1	General conclusion	149
2	Further works	150
2.1	Densification with generative networks	150
2.2	Point cloud colorization	151
2.3	Point cloud color prediction	151
2.4	Multi-task learning	152
2.5	Spatial distribution in sensor topology	152
2.6	Multimodal fusion	152
A Primal-dual algorithm for solving Equation (2.7)		154
1	Discrete setting and definitions	154
2	A primal-dual algorithm	156
B Supplementary experiments of Chapter 2		161
1	Parameters of the algorithm and model choices	161
2	Results with urban data	162
3	Performance on visibility estimation	168
Bibliography		171
Table of contents		185