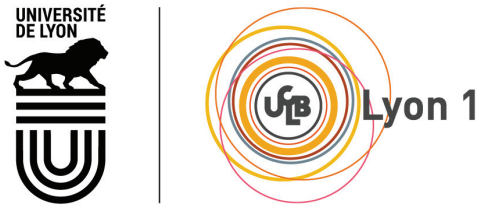
Traitement joint de nuage de points et d'images pour l'analyse et la visualisation des formes 3D

Maximilien Guislain



N° d'ordre NNT : 2017LYSE1219

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

École Doctorale ED512
Informatique et Mathématique de Lyon

Spécialité de doctorat : Informatique

Soutenue publiquement le 19/10/2017, par :
Maximilien Guislain

Traitement joint de nuage de points et d'images pour l'analyse et la visualisation des formes 3D

Devant le jury composé de :

Morin Luce, Professeure des Universités, INSA Rennes	Présidente
Hétroy-Wheeler Franck, Professeur des Universités, Université de Strasbourg	Rapporteur
Monasse Pascal, Chargé de Recherche, Ecole des Ponts ParisTech	Rapporteur
Meinhardt-Llopis Enric, Maître de Conférences, ENS Cachan	Examinateur
Chaine Raphaëlle, Professeure des Universités, Université Lyon 1	Directrice de thèse
Digne Julie, Chargée de Recherche CNRS, Université Lyon 1	Co-directrice de thèse
Lefebvre-Albaret Pascal, Directeur technique, Technodigit	Invité

RÉSUMÉ

TRAITEMENT JOINT DE NUAGE DE POINTS ET D'IMAGES POUR L'ANALYSE ET LA VISUALISATION DES FORMES 3D

Au cours de la dernière décennie, les technologies permettant la numérisation d'espaces urbains ont connu un développement rapide. Des campagnes d'acquisition de données couvrant des villes entières ont été menées en utilisant des scanners LiDAR (Light Detection And Ranging) installés sur des véhicules mobiles. Les résultats de ces campagnes d'acquisition laser, représentants les bâtiments numérisés, sont des nuages de millions de points pouvant également contenir un ensemble de photographies. On s'intéresse ici à l'amélioration du nuage de points à l'aide des données présentes dans ces photographies. Cette thèse apporte plusieurs contributions notables à cette amélioration.

La position et l'orientation des images acquises sont généralement connues à l'aide de dispositifs embarqués avec le scanner LiDAR, même si ces informations de positionnement sont parfois imprécises. Pour obtenir un recalage précis d'une image sur un nuage de points, nous proposons un algorithme en deux étapes, faisant appel à l'information mutuelle normalisée et aux histogrammes de gradients orientés. Cette méthode permet d'obtenir une pose précise même lorsque les estimations initiales sont très éloignées de la position et de l'orientation réelles.

Une fois ces images recalées, il est possible de les utiliser pour inférer la couleur de chaque point du nuage en prenant en compte la variabilité des points de vue. Pour cela, nous nous appuyons sur la minimisation d'une énergie prenant en compte les différentes couleurs associables à un point et les couleurs présentes dans le voisinage spatial du point.

Bien entendu, les différences d'illumination lors de l'acquisition des données peuvent altérer la couleur à attribuer à un point. Notamment, cette couleur peut dépendre de la présence d'ombres portées amenées à changer avec la position du soleil. Il est donc nécessaire de détecter et de corriger ces dernières. Nous proposons une nouvelle méthode qui s'appuie sur l'analyse conjointe des variations de la réflectance mesurée par le LiDAR et de la colorimétrie des points du nuage. En détectant suffisamment d'interfaces ombre/lumière nous pouvons caractériser la luminosité de la scène et la corriger pour obtenir des scènes sans ombre portée.

Le dernier problème abordé par cette thèse est celui de la densification du nuage de points. En effet la densité locale du nuage de points est variable et parfois insuffisante dans certaines zones. Nous proposons une approche applicable directement par la mise en œuvre d'un filtre bilatéral joint permettant de densifier le nuage de points en utilisant les données des images.

Mots-clés : nuage de points ; données multimodales ; enrichissement de données ; recalage image/géometrie ; colorisation ; détection d'ombres

ABSTRACT

JOINT POINT CLOUDS AND IMAGES PROCESSING FOR THE ANALYSIS AND VISUALIZATION OF 3D MODELS

Recent years saw a rapid development of city digitization technologies. Acquisition campaigns covering entire cities are now performed using LiDAR (Light Detection And Ranging) scanners embedded aboard mobile vehicles. These acquisition campaigns yield point clouds, composed of millions of points, representing the buildings and the streets, and may also contain a set of images of the scene. The subject developed here is the improvement of the point cloud using the information contained in the camera images. This thesis introduces several contributions to this joint improvement.

The position and orientation of acquired images are usually estimated using devices embedded with the LiDAR scanner, even if this information is inaccurate. To obtain the precise registration of an image on a point cloud, we propose a two-step algorithm which uses both Mutual Information and Histograms of Oriented Gradients. The proposed method yields an accurate camera pose, even when the initial estimations are far from the real position and orientation.

Once the images have been correctly registered, it is possible to use them to color each point of the cloud while using the variability of the point of view. This is done by minimizing an energy taking into account the different colors associated with a particular point and the potential colors of its neighbors.

Illumination changes can also impact the color assigned to a point. Notably, this color can be affected by cast shadows. These cast shadows are changing with the sun position, it is therefore necessary to detect and correct them. We propose a new method that analyzes the joint variation of the reflectance value obtained by the LiDAR and the color of the points. By detecting enough interfaces between shadow and light, we are able to characterize the luminance of the scene and to remove the cast shadows.

The last point developed in this thesis is the densification of a point cloud. Indeed, the local density of a point cloud varies and is sometimes insufficient in certain areas. We propose a directly applicable approach, using a joint bilateral filter, to increase the density of a point cloud using multiple images.

Keywords: point cloud; multimodal data; data enrichment; image to geometry registration; colorization; shadow detection

RÉSUMÉ SUBSTANTIEL

TRAITEMENT JOINT DE NUAGE DE POINTS ET D'IMAGES POUR L'ANALYSE ET LA VISUALISATION DES FORMES 3D

Ces dernières années ont vu un développement rapide des technologies permettant la numérisation d'espaces urbains. Parmi l'ensemble de ces techniques, la télémétrie laser terrestre suscite un intérêt grandissant. Des campagnes d'acquisition de données couvrant des villes entières ont été menées en utilisant des scanners LiDAR (Light Detection And Ranging) installés sur des véhicules mobiles. Les résultats de ces campagnes d'acquisition sont des nuages de millions de points, représentants les bâtiments numérisés par laser. Cette acquisition ne se limite souvent pas à la géométrie. Par exemple, en couplant le scanner avec d'autres appareils de mesure, il est possible de récupérer des photographies de la scène. On s'intéresse ici à l'amélioration du nuage de points à l'aide des données présentes dans les photographies. Pour ce faire, il est nécessaire de mettre en correspondance ces deux types d'informations. Ce traitement joint du nuage de points et des couleurs provenant d'images est effectué dans un but de recalage, de colorisation, d'extraction des ombres et de densification du nuage de points.

Le premier chapitre détaille les caractéristiques de ce type de données urbaines particulières, constituées à la fois d'un nuage de points et de photographies prises dans 6 directions à intervalles réguliers. Les problématiques habituellement rencontrées lors de l'acquisition, le traitement et l'affichage de ces informations sont exposés, en même temps qu'un bref aperçu des applications directement issues de ces données.

Le second chapitre concerne la mise en relation des données 3D (nuage de points) avec les données 2D (images). En effet, bien qu'elles représentent intrinsèquement la même chose, leurs modalités et leur dimensionalité diffèrent. Il est donc nécessaire de les rendre comparables pour pouvoir exploiter conjointement ces informations. Pour cela le nuage de points est projeté en deux dimensions sur un plan image dans les mêmes conditions que l'image originale. Cette projection est habituellement faite selon un modèle sténopé (*pinhole*), sur lequel s'ajoutent des distorsions optiques dues aux lentilles des appareils photo. Une fois l'ensemble des points du nuage projetés sur ce plan image, la valeur de couleur ou d'intensité attribuée à chaque pixel doit être déterminée. Cette donnée dépend essentiellement du type d'information associé à chaque point du nuage. On peut donc choisir de définir la couleur en fonction d'une illumination dépendant de l'orientation de la normale estimée pour chaque point, ou si le système d'acquisition le permet, d'utiliser l'intensité de retour laser. Il est également possible d'utiliser directement la couleur de la scène à l'endroit considéré si cette donnée est disponible en chaque point. Ces images, générées par projection, présentent cependant deux problèmes résiduels, à savoir, une absence d'occultation des points entre les différents plans de l'image et une faible densité des informations disponibles une fois projetées sur le plan image.

Le troisième chapitre traite du recalage d'images sur un nuage de points représentant la même géométrie. En effet les séries d'images et le nuage de points acquis lors des campagnes d'acquisition urbaines sont généralement alignés à l'aide de dispositifs embarqués, comme par exemple des accéléromètres et des GPS. Cependant, en utilisant uniquement ces données, l'alignement entre images et nuage de points reste imparfait. Ces problèmes peuvent être dus à de nombreux facteurs tels qu'une "dérive" des capteurs, ou encore à des problèmes de stabilité de fixation des appareils photo sur le véhicule. Pour corriger ce problème, il est possible de ré-aligner manuellement les images sur la géométrie. Toutefois, cette étape est relativement chronophage et donc impossible à réaliser sur de gros nuages et des groupes de photos importants. Il est donc nécessaire d'avoir une méthode permettant de corriger le positionnement des images automatiquement à partir de leur position et de leur orientation originales. Ainsi pour permettre ce recalage, une nouvelle métrique de comparaison entre images a été introduite. Celle-ci permet une meilleure résistance au manque d'information et à la différence entre les modalités de deux images. En effet, en utilisant une métrique globale s'appuyant sur l'entropie jointe, et en ajoutant une métrique locale basée sur la différence des gradients orientés, il est possible d'améliorer significativement la résistance de cette métrique aux erreurs. L'utilisation de cette métrique, pour un recalage d'images sur de la géométrie, en multi-échelle et en deux étapes, donne des résultats supérieurs aux techniques précédemment proposées pour les données urbaines.

Le chapitre quatre se propose d'apporter une couleur cohérente au nuage de points en utilisant de multiples photographies. En effet, bien que de plus en plus de LiDAR modernes permettent d'associer une couleur à chaque point du nuage, cette couleur peut parfois contenir des artéfacts dus à une mauvaise acquisition de la scène (*e.g.* personne en mouvement, occultation partielle de mobilier urbain) ou à une erreur du capteur due à des conditions extérieures (*e.g.* soleil sur la lentille, changement de luminosité brutal). La qualité et la précision de cette couleur peuvent aussi être assez faibles sur certains appareils d'acquisition, ce qui donne une impression de mauvaise résolution du modèle géométrique. Ainsi, obtenir une couleur cohérente, de qualité et de haute définition est une tache difficile dans des environnements urbains complexes. La plupart des méthodes proposées se limitent à une approche naïve, ou considèrent des scènes relativement simples, avec peu d'occultations et d'incohérences entre les différentes prises de vues. L'utilisation d'une méthode basée sur la minimisation d'une énergie utilisant de multiples images donne de bons résultats, lorsqu'elle est modifiée pour résister aux problèmes d'incohérence des scènes urbaines.

Cette couleur haute définition et fidèle est néanmoins dépendante de l'éclairage et de la luminosité à laquelle les images ayant servi à la colorisation ont été acquises. Ainsi pour que cette couleur devienne indépendante du moment de l'acquisition il est nécessaire de détecter les ombres et de corriger les couleurs. La méthode de détection proposée s'appuie sur les propriétés physiques du LiDAR. Pour un point d'acquisition, la valeur d'intensité de retour du laser est liée uniquement au matériau visé. La couleur associée à ce point dépend du matériau ainsi que de la luminosité ambiante et directe. Grâce à ces deux informations on peut détecter automatiquement des zones d'interfaces entre ombre et lumière. Une fois ces zones détectées, les points situés à l'ombre sont classés dans le plan image en utilisant une technique par graphe (*graph-cut*) s'appuyant sur les propriétés physiques de l'ombre, déterminées lors de l'étape précédente. Finalement, connaissant ces propriétés et à l'aide d'un modèle d'illumination simplifié, il est

possible de corriger la luminance et la chrominance pour l'ensemble des points de sorte à obtenir une illumination neutre de la scène.

Le chapitre cinq se concentre quant à lui sur la densification et la complétion d'un nuage de points en utilisant de multiples images. En effet, même lorsqu'ils sont très performants les LiDAR mobiles ne peuvent acquérir toute la géométrie d'une scène urbaine complexe. En effet, de nombreuses zones d'occultations apparaissent autour du mobilier urbain. En plus de zones d'occultations locales, on peut observer d'importantes variations de densité du nuage de points. Il est possible de corriger ces problèmes locaux en densifiant le nuage de points après son acquisition en se reposant sur l'utilisation des photographies pour guider la densification.

CONTENTS

LIST OF FIGURES

ACRONYMS

LiDAR Light Detection And Ranging

ToF Time of Flight

MI Mutual Information

NMI Normalized Mutual Information

HOG Histogram of Oriented Gradients

RGB Red Green Blue

PCA Principal Component Analysis

GPS Global Positioning System

TLS Terrestrial Laser Scans

SIFT Scale-Invariant Feature Transform

SURF Speeded Up Robust Features

RANSAC RANdom SAmple Consensus

GOM Gradient Orientation Measure

SFM Structure From Motion

ICP Iterative Closest Point

CCD Charge Coupled Device

DHOG Distance between Histogram of Oriented Gradients

HDR High Dynamic Range

LOD Level of Detail

UAV Unmanned Aerial Vehicle

4PCS 4-points Congruent Sets for Robust Surface Registration

GP-GPU General-purpose computing on graphics processing units Registration

NOTATIONS

O the optical center of the camera.

f the focal distance of the camera.

ppa the camera principal point located at coordinates (u_0, v_0), the location of the projection of the optical center in the image plane around which the distortions are centered.

t_1, t_2 the two tangential distortion parameters.

k_1, k_2, k_3 the radial distortions coefficients.

Ω_0 Original camera pose.

P a point located at coordinates X, Y and Z.

Q a point located at coordinates X_q, Y_q and Z_q.

x, y, z the coordinates X, Y and Z of the point P within the local camera pose Ω_0.

x_0, y_0 the homogeneous coordinates of the point P within the local camera pose Ω_0.

x', y' the radially distorted equivalent of x_0, y_0.

x'', y'' the tangentially distorted equivalent of x', y'.

p,q respectively the projection of the points P and Q on the image plane.

u, v the coordinates of the projection of the point P on the image plane.

ω_P the solid angle of visibility of the point P.

ψ the visibility solid angle threshold.

α_q the cosine of the visibility angle between $\vec{OP}$ and $\vec{QP}$.

α_{sector} the maximum value of α_q for a given sector.

p_i the probability for an arbitrary pixel of an image I to be of intensity i.

$p_{(m,n)}$ the joint probability a pixel k has an intensity m in an image I_1 and n in an image I_2.

B_{ij} HOG block centered at coordinates (i, j).

wb, hb respectively the width and height of an image I in number of HOG blocks.

$w_{B_{ij}}$ weight of the block B_{ij}.

v^I_{cbij} the value in image I of a HOG bin b in a cell c belonging to a block B_{ij}.

α Weight parameter contained in MIDHOG.

δ_x, δ_y displacement in the image plane in pixels.

ω the yaw rotation correction to apply to pose Ω_0.

ϕ the pitch rotation correction to apply to pose Ω_0.

K_P the identifier of the 3D Point P

C the color vector (containing the L, a and b values) associated with each point P of the cloud.

D_P is the data term for the point P.

$S_{P,Q}$ is the smoothness associated to the points P and Q belonging to the neighborhood N.

λ is the weight of the smoothing term in the colorization energy.

v_P is the vector of all the possible colors for the point P.

c_P is the final color associated to P.

$Md(v_P)$ is the median color of all the possible colors for the point P.

α_P is a binary term that indicates if the data-term will be used or not.

Υ is the distance threshold that define if an image can be used to associate a potential color to the point P.

R_p the reflectance value associated with point P.

$L(p)$ the Luminance component at pixel p.

L_L, L_S the estimated Luminance component of the scene in sunlit and shadowed area respectively.

$L(P)$, $a(P)$, $b(P)$ the Luminance, a and b components of the $La*b*$ color at point P.

$L'(P)$, $a'(P)$, $b'(P)$ the corrected Luminance, a and b components of the $La*b*$ color at point P.

d_p the estimated depth at the pixel p.

w_p the weight normalizing factor of the joint bilateral and trilateral upsampling.

N_p the set of pixels in a limited neighborhood around the pixel p.

c^I_p the $La*b*$ color at the pixel p in the color image I.

σ_d and σ_c, parameters of a Gaussian filter for the distance and color respectively.

ϖ a distance threshold in meter for depth validation.

ξ_p validity score for the pixel p of an upsampled depth map.

INTRODUCTION

Recent years saw a rapid development of city digitization technologies. Among these technologies, terrestrial laser scanning knew a rapidly growing interest. Acquisition campaigns covering entire cities are now performed using LiDAR (Light Detection And Ranging) scanners embarked aboard mobile vehicles. These digitization campaigns yield 3D models of urban environments. These 3D models have numerous potential applications, for instance virtual tours or augmented reality, but also notably in urban development planning.

Since the introduction of 3D city models, the work performed by surveyors and topographers is evolving toward a fully digital work environment. This digital work environment tends to dissociate measurement performed outdoor, on the field, from measurement on the virtual model. Indeed, the measurements on the field can be long and tedious, but thanks to digitization devices, the time spent outdoor can be greatly reduced, while longer digital measurement on the virtual models can be performed in an office, by more qualified workers. To make this transition possible, the digital scene must be as close as possible from the real environment. Results of these digitizations are sets of unordered 3D points (point clouds), sometimes incomplete and noisy. The point cloud may be supplemented with a set of pictures taken at regular intervals along the acquisition path. Indeed, the addition of a digital camera to a LiDAR is about to become systematic, due to the spectacular progress of the digital optic devices in terms of cost and quality.

This thesis is performed in collaboration with Technodigit, part of Hexagon group, editor of 3DReshaper. 3DReshaper is a software dedicated to point cloud processing for various applications, such as architecture, cultural heritage, civil engineering or VFX and cinema. Their particular relationship with their customers in the surveying domain helped to unveil the new evolution of the surveying habits. While Technodigit already possesses software suits able to perform 3D reconstruction, the company wants to offer new workflows capable of handling large amounts of images and using them for digital model improvement.

Innovative solutions integrating LiDAR and cameras are now available on the consumer market, and large scale measurement campaigns have been performed. The objective of this thesis is to lay the theoretical basis to confront these different types of data that are point clouds obtained from LiDAR and images. The combined use of these two types of data allows for a coherent analysis of the digitized scenes, but also to improve the quality of the data visualization, with a rendering that should be close to the one observed on the images. To achieve this joint analysis, several challenges must be tackled. First, the accurate correspondence between the 3D object and its planar projection, that the images are, must be established. It is then necessary to automatically and properly register the images and the 3D surface. 3D point clouds are characterized by large areas that do not contain any information. This lack of information arises from occlusion from various elements present on the scene.

Furthermore, a photography does not always contain a meaningful color information since this information can be distorted by several factors, including the camera characteristics, reflections, shadows, direct and indirect illumination. The 3D position of the points from the cloud can also be erroneous depending on the photometric properties of the scene objects and the sensor itself. Information present on the images should allow to improve, remove noise or segment the point cloud. The objective is to solve the problems in a virtuous loop. The processes of improvement are not meant to be sequential but rather interlaced. Urban environments are a particularly difficult case, on which the use a joint image and point cloud improvement can make a great difference. The joint processing explored in this thesis has for goal to be able to generate high quality models of urban scenes, clear of artifacts.

In the following chapters we will first discuss about the particularities of the urban data, the means of acquisition of 3D surfaces, and the challenges given by such kind of data. The projection process to associate 3D and 2D data will be presented in chapter 2. In chapter 3 a method to improve the original images registration to the geometry is presented. Chapter 4 will present a method to colorize and remove cast shadows from the point cloud using multiple images. Finally, chapter 5 will present a way to improve the density of a point cloud from images.

CHAPTER 1

URBAN DATA: STATE OF THE ART AND PROBLEMATIC

Contents

As more and more cities embrace the digital revolution, there is a growing demand for digitization technologies able to reconstruct urban scenes from geometric measurements and pictures. The goal might be, for example, to plan urban evolution or allow virtual tours of the city. This chapter is an overview of the objectives, the tools used and the remaining challenges faced in urban environment digitization.

1.1 Goals of urban data acquisition

Acquisition and reconstruction of urban scenes is a research field of significant interest. The objective is to obtain digital models of either individual buildings or complete urban environment including streets and urban furniture.

These digital models are geometric representations of real buildings or scenery. The need for this type of digital model corresponds to a global movement of mass digitization of urban environment. More and more professions evolve toward the massive use of digital models to simplify their access to the data in their everyday work. Surveyors and civil engineers are not an exception to this tendency: with the advance of digital models, less time is spent taking measurements on the field and more time is spent in the office, working in a more comfortable environment on digital models.

The direct applications for 3D city models are numerous for urban surveying and city modeling, in particular for street extraction and analysis. As an example, EL-HALAWANY et al. [EH+11] and HERVIEU and SOHEILIAN [HS13] propose to use point clouds of urban environments to

detect the street curbs in order to model the street network. Higher level applications can also be designed, as illustrated by the method of SERNA and MARCOTEGUI [SM13] that uses point clouds to study the accessibility of urban environment. But urban models have much broader applications, as described by BILJECKI et al. [Bil+15]. Among other application domains, we can cite:

- City development planning and architecture.
- Cultural heritage.
- Security and defense.
- Risk assessment.
- Virtual reality.
- Transport planning.
- Solar energy planning / right to light.
- Viewshed analysis.
- Integrated storm water management plan.

The digitization effort of cities to keep tracks of their development leads a significant number of them to publish their data in public open access. Hamburg, Rotterdam, Montréal, New York and Lyon have available point cloud-based datasets. Some other private companies also ran acquisition campaigns, but the collected data are not always publicly available. For instance the city of Calgary and its surroundings were scanned by *ATLIS Geomatics, Inc* in 2017. This multiplication of acquisition campaigns is a sign of the recent interest in the potential application of digital city models.

The different acquisition techniques used to digitize cities are usually based on aerial acquisition methods, and only yield low precision models. If such models are sufficient for certain applications, they may not be detailed enough when a particular attention needs to be given to facades or street elements. Complete models of buildings with numerous details require a better precision in the acquisition process. For these purposes, ground-based campaigns are necessary to capture more precise data. Such high-precision data are more difficult to obtain. They require a much longer on-field acquisition time to capture the environment. The large amount of data collected introduces real challenges in terms of data handling and rendering performance.

Contrarily to these planned and time-consuming acquisitions, some works intend to use available data to reconstruct digital models. In this type of work, the goal is to reconstruct a well documented building from crowd data. It can for instance consist in taking a large number of images of a building from publicly available image database, such as *Flickr*, to reconstruct the 3D model of the landmark building. These methods usually do not obtain a very precise model but a visually plausible and pleasant one. However these methods can only work for

well-documented places (such as touristic landmarks or monuments) and do not scale to complete cites. An impressive example is for instance Phototourism [SSS06]. In this project, the authors are able to reconstruct buildings from a collection of photographs gathered from the Internet. This reconstruction coupled with image-based rendering allows users to perform photo tours.

Currently, the majority of tasks performed on these digital data still requires user-interaction. A higher amount of urban digital data allows for more automation of the data processing. This newly possible level of automation also allows to process a higher volume of data than what was previously possible. In order to properly automatize the analysis of complex scenes, the quality of the data must be as high as possible, since the presence of inconsistencies or artifacts can impair automatic processing. It is therefore necessary to ensure the high quality of the data.

As we outlined in the previous paragraphs, acquisition of the geometry can be divided into two different categories: range-based approaches and image-based approaches. These two approaches offer different results in term of quality but also have different settings requirements. In the following sections, a short overview of both acquisition techniques and their output data will be presented.

1.2 Methods for urban data acquisition

1.2.1 Data representation

Urban models can be represented in various ways, such as by point clouds. These point clouds are series of unordered 3D points that sample the buildings and streets surface. This is the simplest representation of geometric 3D data and the main data output of digitization devices and tools that are described in this section. Other types of representation are possible, such as meshes, which are triangular-based representations of the surface. A mesh model has the notable advantage to be able to support texturing. Having continuous texture quickly gives an overall quality feeling. Some higher level representations also exist, such as CityGML [GP12]. CityGML encompasses semantic, topological information and meshes that are divided in Levels Of Details (LOD) of increasing structure and geometric complexity. This representation is however complex and cannot be obtained directly from measurement devices. They must undergo a reconstruction step. Several options are available to turn raw point clouds into LODs of a 3D urban scene. While some automatic reconstruction methods are available, such as the one presented by VERDIE, LAFARGE, and ALLIEZ [VLA15], it is most often still performed interactively [Sin+08][Ari+13].

This thesis will focus solely on the use of point clouds as geometric information, as they are the simplest representation of real buildings and the direct output of digitization devices. By using this low-level information obtained directly from the scanning devices, biases that may be introduced by reconstruction techniques are avoided.

1.2.2 Image-based acquisition

Photogrammetry, introduced by Albrecht Meydenbauer in 1858, is a relatively old research field. Since 2005, this field has raised a surge of interest and is a widely explored research area. A lot of

different reconstruction methods are available, producing high quality results. All those methods mostly follow the same workflow. The typical workflow consists in detecting and matching common elements in images while estimating the spatial position of these images via Structure from Motion (SfM) [KVD91]. Once all the cameras and images are properly located, a dense point cloud is computed using Multi-View Stereo (MVS) [Fio+12]. The massive amount of image descriptors that followed the introduction of Scale Invariant Features (SIFT) [Low04] offered an important improvement of digital photogrammetry.

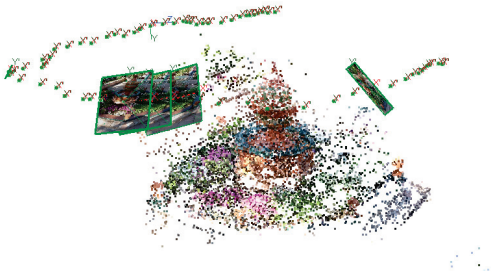
The first step of the SfM pipeline is to find similar features in image pairs. This point pairing is usually performed using descriptors obtained by SIFT [Low04] or less standard image feature detectors and descriptors such as SURF [BTVG06], ASIFT [MY09], ORB [Rub+11], BRIEF [Cal+10] and many others. Salient pixels are located in images using blob[1] or corner detectors. Then for these salient pixels, a vector of features is computed as a unique descriptor. In the case of SIFT descriptors, this feature vector size is 128. This descriptor is supposed to be fairly resilient but it can still generate outliers that are usually removed using Random Sample Consensus (RANSAC) [FB81]. The knowledge of matching points in two different images allows to determine the relative pose of the cameras from the object they represent. By applying this principle on all images, a process called bundle adjustment [Tri+00], one can obtain consistent poses for all the images all together. It is also possible to triangulate the 3D coordinates of the keypoints present in several images. Different variants and implementations of this SfM pipeline are available, such as a version with a hierarchical image management [FFG09], an incremental image addition version [Kim+13], or a method adapted to catadioptric cameras [Lhu08].

The SfM step yields a very sparse point cloud composed only of the triangulated points of interest found in multiple images. Such a sparse point cloud is hardly usable in any case since it only very coarsely represents the geometry of the scene. This point cloud needs to be densified to represent the geometry in more details before being of any use. Furthermore, the obtained point cloud is at an arbitrary scale and is not usable to perform measurements. In order to scale it properly, it is necessary to either manually perform the scaling knowing a distance that appears in the geometry, or to possess the georeferenced position of the images. Once the images have been positioned relatively to each other, the dense reconstruction is a critical step to obtain quality results. Different methods exist, using either stereo or multi-view reconstruction. In the case of multi-view reconstructions, several approaches are possible. In their work, FURUKAWA and PONCE [FP10] propose a patch-based reconstruction that yields good results (see Figure 1.1(c)). Another method by VU et al. [Vu+12] is based on a global optimization and photometric consistency giving impressive results. SURE [Rot+12] is another MVS available technique which is based on semi global matching. A full review of existing multi-view stereo methods can be found in [Rem+14]. Some techniques are commercially distributed and the algorithms used are not entirely known, such as the method available in the *Pix4D* software or in *Agisoft PhotoScan*.

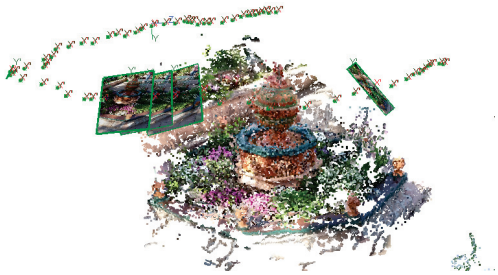
Photogrammetry and computer vision methods can produce high quality point clouds using low cost materials and are therefore more and more used for acquisition campaigns. Aerial

[1] A blob is, informally, a region of an image where the pixel properties are similar. Blob detection consists in finding interfaces between similar regions.

(a) Keypoint detection/matching (SIFT)



(b) After SfM step.



(c) After PMVS.

Figure 1.1: Sequential illustration of the steps used to generate a point cloud from images (Keypoint detection, SfM, MVS). Results presented here were obtained using VisualSFM [Wu+11].

photogrammetry is particularly indicated as it is a high quality data acquisition system that covers large areas at once. Aerial photogrammetry can also nowadays be a low cost, light 3D data acquisition system that can for instance easily be integrated aboard unmanned aerial vehicles (UAV). When combined with proper georeferencing data, output airborne photogrammetry can compete against terrestrial laser scanning for terrain acquisition (Figure 1.2).

Methods using images to reconstruct 3D geometry generate well-known artifacts. The most commonly found artifacts in the reconstructed geometry are the following:

Lack of precision. In low-textured and homogeneous areas, the 3D reconstruction is less precise. Points estimated in these areas have a higher error rate. One of the visible effect of this lack of precision can be that a flat area becomes a thick, fuzzy surface (Figure 1.1).

Shadows. Point clouds obtained using photogrammetry have the advantage to possess a rather good colorization. However, this color attributed to each point is dependent on the color at the acquisition time. It means that the reconstructed point set will display the cast shadows and bright regions depending on the scene illumination at capture time. These illuminations might be inconsistent over the scene.

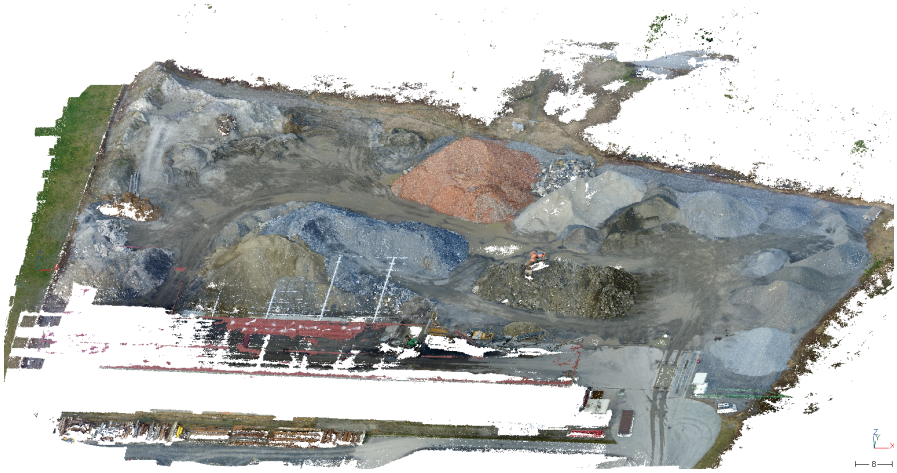


Figure 1.2: Photogrammetric reconstruction result of a stockpile using a Phantom 3 drone with 200 images using 3DReshaper. Despite an overall satisfactory reconstruction, the red, uniform roof of the building present in the scene was only partially reconstructed.

Partial structure. In completely homogeneous and texture-less areas, the triangulation of 3D points can be impossible. These homogeneous areas tend to create acquisition holes on the model that do not possess any related 3D information. This is visible in Figure 1.2 where a large homogeneous area in images is completely empty of points in the reconstructed cloud.

While these photogrammetry methods produce good quality results at an affordable cost, these acquisitions are not accurate at long range. Another set of 3D acquisition systems are range based devices. These devices are more expensive, but also more precise and are very frequently used by civil engineering teams and land surveyors.

1.2.3 Range-based acquisition

Range-based geometry acquisition makes use of an active acquisition device to measure distance between the device and the environment. These devices evolved quickly from rather simple sonar-based or infrared-based systems to modern Light Detection And Ranging (LiDAR). Sonar-based acquisitions are still used in the particular case of submarine data acquisition [Jou+12], for coast or wreck analysis for instance. For terrestrial or airborne geometry acquisition they have been replaced by faster and more accurate LiDAR systems. Infrared-based (IR-based) detection methods have recently known a surge of interest thanks to the Microsoft Kinect device [Zha12]. This inexpensive device uses an IR camera coupled with a structured IR light pattern projection to obtain depth data of a scene. However this device, despite recent uses in context specific applications, such as open field agricultural acquisition [RP+17], is not adapted to outdoor large scale urban acquisition and is mostly limited to indoor scanning [KE12][Hen+12].

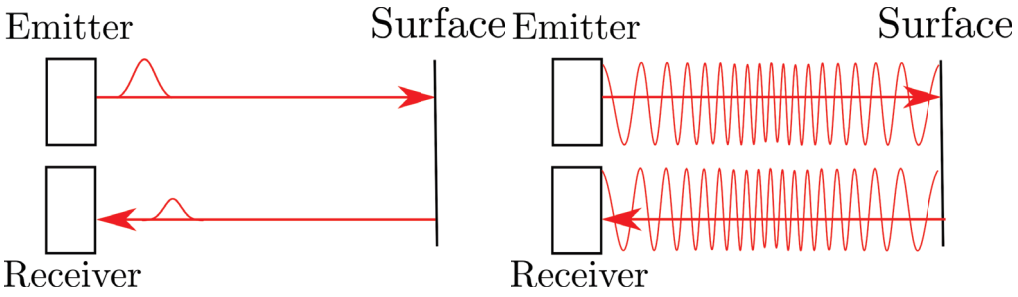
LiDAR technologies are based on the use of a 600nm to 1000nm laser to obtain the geometry,

and can be used to measure at long range. Most types of LiDAR systems nowadays outperform largely other measurement devices, even in specific condition such as airborne bathymetric measurements [FD+14][KK17]. There are different ways to obtain the distance between the device and the targeted surface, the most common ones being triangulation-based and time of flight methods.

Triangulation-based measurement systems (including the Kinect infrared system) are built using a laser emitter and a camera fixed with a constant angle between each other. The distance and the orientation angle between the camera and the laser are known. Using this two known values, it is possible to determine the distance of a laser spot projected on the object using trigonometry. For this type of measurement device, the measure error is directly linked to the distance of the measured object, therefore its use is usually limited to small range scanning (*i.e.* less than $10m$) [Mou14].

A more common LiDAR acquisition technique is the Time of Flight (ToF) method. ToF is similar in principle to sonar range finding. A laser beam is emitted in a direction, and its reflection is received. The time interval between the emission and the reception of the beam, knowing the speed of light in the medium, allows to compute the distance of the object hit by the beam. Pulsed ToF scanners (Figure 1.3(a)) use a simple principle that consists in emitting a laser beam and measuring its return value for each possible measurement of its field of acquisition. This acquisition method range is directly dependent on the laser energy, and can be used for potentially long distance scanning. However, it implies to emit one burst of energy and wait for its echo, leading to a slow acquisition speed.

Phase shift ToF scanners (Figure 1.3(b)) use a continuous laser beam to scan their entire field of view. This laser is modulated in amplitude or frequency using a sinusoidal function. This modulation allows a much faster acquisition rate. However, this measure might be uncertain depending on the sinusoidal function frequency and thus limits the maximum usable range. Indeed, as the modulation is cyclical, depending on the sinusoidal frequency, the original emission time of the received signal is uncertain.



(a) Principle of the pulsed ToF LiDAR. (b) Principle of the phase shift ToF LiDAR.

Figure 1.3: Illustration of the basic function of ToF LiDARs.

The laser beam of LiDAR devices is usually emitted toward a rotating mirror. This mobile mirror is used to orient the laser direction and measure points at different angles from the emitter/receiver couple. Different degrees of freedom are possible for the acquisition. The

rotation of the mirror can be one of these degrees of freedom, but the frame comprising both the mirror and the emitter/receiver device can sometimes also move, providing another degree of freedom. Such devices providing two degrees of freedom can yield almost complete spherical scans of their surroundings.

It is possible to differentiate static and mobile LiDAR systems. Static LiDARs produce independent, partial models of the world around the acquisition device. To obtain a complete and detailed model, several scans must be merged together. This merging process is usually performed manually using common overlapping areas on the different scans. Recent methods for automatic registration have been proposed and produce good results, such as for example 4-points Congruent Sets for Robust Surface Registration [AMCO08] [TWS14] or methods based on Gaussian Mixture Model [JV11]. Whether this registration is manual or automatic, a refinement step using Iterative Closest Point (ICP) is usually applied [Che+02]. In case of mobile LiDAR, scans can correspond to several static scans taken along the path of the vehicle. These scans are enriched with their global position and orientation obtained from GPS and inertial sensors. This is the case of the KITTI dataset [Gei+13] where a Velodyne LiDAR gives 360° scans at different positions (Figure 1.4). In other cases, continuous scan can be obtained, with much smoother results in term of density. Using once again GPS and inertial sensors, a consistent continuous point cloud is generated (Figure 1.5). Mobile LiDAR can now be carried in a backpack [Rön+16] or be hand-held [Zlo+14], allowing for the acquisition of streets or city places that cannot be accessed in a vehicle.

Point clouds acquired using the LiDAR technology are not exempt of artifacts and geometrical aberrations. The most commonly observed artifacts are:

Transparency. The transparent nature of windows and store fronts produces errors during the acquisition. Information of the geometry captured behind the windows can either be only partial and particularly noisy, or be simply missing.

Rooftops. While roofs can be acquired using airborne LiDAR, with terrestrial LiDAR, most roofs are only partially acquired, and depending on their shapes, can have severely occluded areas.

Static object occlusion. Vegetation and urban furniture can produce occlusions during the acquisition. This is particularly visible in urban environments where a lot of objects on the foreground make the acquisition of several parts of the background impossible (Figure 1.5).

Dynamic object occlusion. Pedestrians and cars also produce important occlusions during the geometry acquisition. This type of occlusion is particular since the movement of the occluding object itself will produce both a disturbed object acquisition and its occlusion counterpart.

Color. Color information is not always available for each point of the cloud. This information is however important to correctly analyze the scene. In some cases, the LiDAR device can provide a color for each point. Unfortunately, this color is often low resolution and suffers from acquisition and projection artifacts.

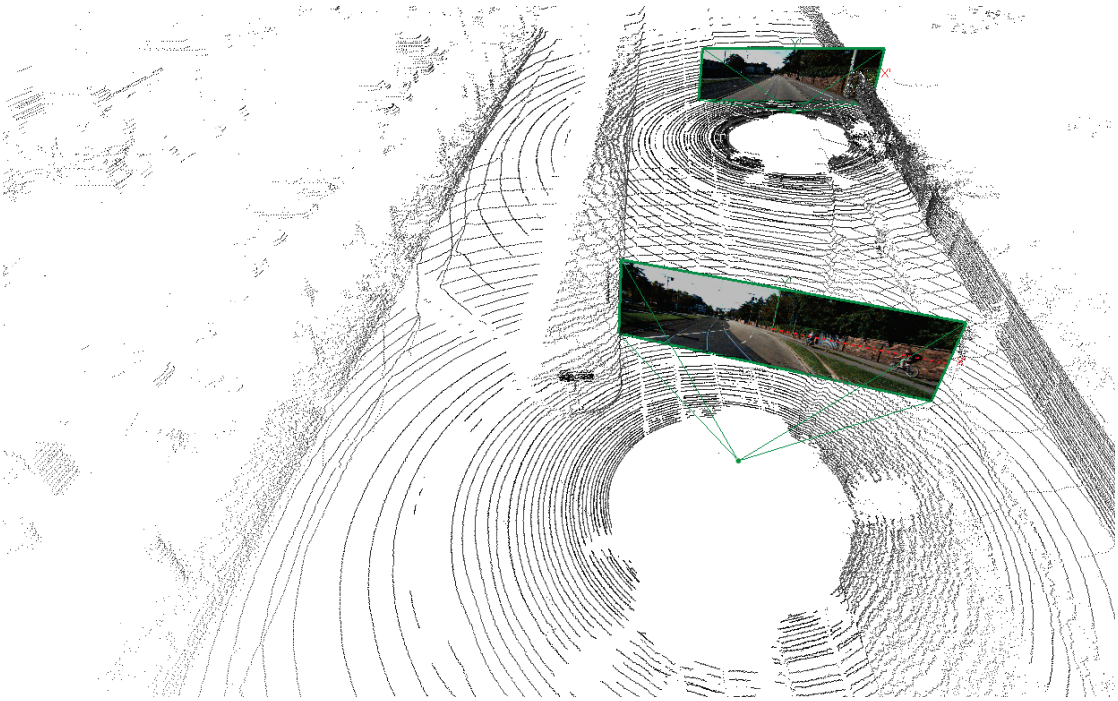


Figure 1.4: Two complete scans of the KITTI dataset seen from a top-down angle in 3DReshaper. LiDAR device position for each scan is visible in the blind circular areas that coincide with the image position. Static scans taken from multiple positions produce similar point clouds.

1.2.4 Combined datasets

As stated previously, more and more mobile acquisition devices now embed both a LiDAR and cameras to obtain as much information as possible on the scene. This is illustrated for example by the research-oriented dataset KITTI by GEIGER et al. [Gei+13] or the Stereopolis II dataset by PAPARODITIS et al. [Pap+12]. A significant number of commercial LiDAR systems also provide a 3D point cloud coupled with images, such as the *Leica Pegasus* and *Leica Pegasus Backpack* or the *Faro Road-Scanner C*.

Even if these systems provide interesting multi-modal data of the environment, surprisingly few works use these different modalities in a joint analysis. DECHESNE et al. [Dec+17] use both aerial LiDAR and multispectral imagery to improve segmentation of forest stand. MOUSSA, ABDEL-WAHAB, and FRITSCH [MAWF12] propose to enhance terrestrial range scans using terrestrial photogrammetry to fill acquisition holes. SIBBING et al. [Sib+13] discuss a method to render a point cloud using surfels textured from images. The objective of this rendering is to obtain an image that can be given to a SIFT detection computation.

We focus below on two datasets that will be used extensively in this thesis.

Figure 1.5: Top-down rendering of a continuous scan performed with a vehicle mounted LiDAR in 3DReshaper. The rotation axis of the LiDAR was inclined which created particular oblique occlusion areas visible on this rendering.

KITTI dataset

The KITTI dataset [Gei+13], is a widely used mobile LiDAR dataset. This dataset is composed of several different LiDAR scans taken by a Velodyne LiDAR coupled with four cameras. It offers the advantage to provide a consistent set of data that are reliably usable for multiple purposes, such as Simultaneous Localization And Mapping (SLAM), object detection and tracking or scene flow evaluation. Several different acquisition drives are available, representing different kinds of environments such as urban scenes or open road sceneries. We limit our tests and investigations on the particular *City* and *Residential* categories that are the most likely to represent urban environments. These datasets were acquired in the city of Karlsruhe, Germany. The KITTI dataset also provides precise meta information, such as the camera calibration that are mandatory

to jointly use images and geometry information (*cf.* chapter 2). It also contains the necessary transformation to obtain the pose (position + orientation) of the Velodyne LiDAR and all the 4 embedded cameras from the vehicle pose. For each captured time frame we also have the GPS coordinates and the car orientation. This position and orientation information allows to combine all the scans together. The 4 acquired images are composed of 2 color and 2 grayscale images (Figure 1.6). All these images are perspective images oriented toward the front of the moving vehicle with a resolution of 1242×375 pixels.

(a) Camera 1. (b) Camera 2.

(c) Camera 3. (d) Camera 4.

Figure 1.6: Example of images taken by the KITTI cameras, from the $2011/09/26$ data, drive 0009.

While the KITTI dataset is an interesting dataset to test and compare results, it is unfortunately not comparable to current state of the art acquisition system results such as the Shrewsbury dataset. Indeed, as it will be further discussed in chapter 3, the KITTI dataset contains a lot of noise and has a limited acquisition height.

Shrewsbury dataset

The Shrewsbury dataset is a company-owned dataset acquired during one of the first test of the Leica Pegasus II acquisition platform. This particular LiDAR device is mounted on top of a vehicle. It was acquired in the city of Shrewsbury, United Kingdom. The acquisition path itself is a loop of $2.5km$ containing approximately 260 millions of points. This set of points contains a digitization of the streets and includes some particular landmarks whose proximity with the driveway allowed to be captured by the LiDAR scanner, such as *Ireland's Mansion* (Figure 1.7(d)), *St Mary's Church Bell Tower* (Figure 1.7(b)) or *Major General Robert Clive statue* (Figure 1.7(c)). These few landmarks will be present throughout the different results rendering in this thesis. Besides these landmarks, the city itself contains fine examples of typical Tudor and Georgian architecture.

This dataset was acquired by a continuous LiDAR stream that provides a smooth point density along the direction of the vehicle. The laser return intensity is also available in high quality,

depending almost only on the material reflectivity, which provides an important information for displaying results as it will be further detailed in chapter 2. The point cloud is also fully colorized.

(a) Global top view of the acquisition path.

(b) St Mary's Church Bell Tower. (c) Maj Gen. Robert Clive statue. (d) Ireland's Mansion.

Figure 1.7: Illustrations of the point cloud acquired during the digitalization by the Pegasus II in the city of Shrewsbury. The point cloud is rendered in 3DReshaper using laser intensity values.

While the laser scanner is continuously acquiring points, the images are acquired every meter in contrast with the Kitti dataset where both point sets and images were captured at fixed time intervals. These images are taken in six directions simultaneously (front, back, front right, front left, back right and back left, as illustrated in Figure 1.8). A seventh spherical camera is located on top of the acquisition device and captures a picture of the sky and potentially very high buildings. In the presented cases, this camera will not be used, as we focus on higher definition perspective cameras. A total of 2452 different image sets were taken in each direction for a total amount of 14712 images. These images have a resolution of 2046×2046 pixels and are JPEG encoded.

Figure 1.8: The 6 images of the environment acquired every meter along the acquisition path of the Pegasus II scanning device. All images contain an important overlap. Original car outline is from the Kitti dataset presentation article [Gei+13].

1.3 Context of this work

As stated throughout this chapter, large city acquisitions are used in a variety of applications, from digital city model to specific object extraction. The increasing interest for automation of point cloud analysis and processing calls for higher data quality. Indeed, the quality of the geometric data is crucial for proper visualization and analysis. This data improvement can be done by jointly using information from images and information from point clouds. Indeed, images and point clouds are complementary data. Point clouds are high quality geometric representations but are also sparse. Images, on the other hand, despite not containing any raw geometric data, provide dense color information. The combined use of images and 3D data, as the one described in section 1.2.4, can turn to an advantage since they meet specific needs for visualization and analysis. This thesis investigates the combined use of these two types of data to improve the data quality. Methods presented afterwards aim at producing high quality data, that cannot be obtained using geometry or images separately.

To jointly use information from the 3D world (point clouds) in combination with the 2D data (images), it is necessary to be able to link these two information sources together. The projection of a set of 3D points on a particular 2D image plane is a well-known problem. A simple perspective projection method will be used in this thesis to bind the 2D and 3D information together. The details of the used projection method are described in chapter 2.

Using this projection to correlate the two data requires a near perfect concordance between them. The projection of a building on the image plane must fit the pixel representation of this building on the photograph. However, even if some datasets have, due to their data acquisition process, a nearly perfect concordance between the images and the geometry, some others may have slight misalignment problems. This misalignment problem can be even worse if the geometry data and the images are acquired at different times. In this case the camera poses relatively to the point cloud will need to be estimated from scratch. If the geometry projection discussed in chapter 2 does not correspond to the information located on the image, the complete process associating images and geometry will be corrupted, making the joint analysis impossible. The chapter 3 is dedicated to the problem of matching exactly the information on images and geometry by correctly registering images to the geometry.

In order to visualize the full acquisition result, the color information must be added to the point cloud. When no color information is available, being able to recognize elements in the point cloud is a difficult task, both for human and computer alike. This is illustrated by Figure 1.10 where color information could easily enhance the scene analysis. However, even when point clouds possess color information, this color may be badly assigned. Indeed, in the Shrewsbury dataset presented in section 1.2.4, some areas are afflicted by inconsistencies. Among others, the data can contain bad projections due to static or mobile objects (*e.g.* sidewalk colored by the car color, Figure 1.9(a)), wrong color assignment due to external influences (*e.g.* pink color of the brick wall, likely due to a lens flare, Figure 1.9(b)), color variation depending on the acquisition time (Cast shadows, Figure 1.9(c)), or even a total lack of information (black area without any information, Figure 1.9(c)). While a human operator can easily ignore those inconsistencies to analyze the scene, a computer analysis would be sensitive to such problems. These problems

are also obvious when a representation is expected to be as realistic as possible for visualization purposes. Chapter 4 addresses this colorization problem and focuses on the addition of high quality colors to a point cloud from multiple images and on the removal of cast shadows.

(a) Wrong color attribution. (b) Wrongly assigned color. (c) Cast shadows.

Figure 1.9: Example of wrong colorization in a colorized point cloud. In Figure 1.9(a) a wrong color attribution is visible: the sidewalk was colored with the car and the parking place panels. Figure 1.9(b) displays a wrong color assigned to a building wall, probably due to a lens flare. In Figure 1.9(c), shadows influence the color given to the point cloud depending on the time of the acquisition. The black line in the lane center is data without any color information.

The last topic addressed in this thesis is the completion of the point cloud data using multiple images. Indeed, the point sets acquired by LiDAR devices still have numerous problems. One major problem is the presence of acquisition holes within the point cloud. This missing parts problem can have different origins, it may come from a distance acquisition limit of the LiDAR scanner, but it can also come from occlusions of static (Figure 1.7 and Figure 1.9) or dynamic objects. These occlusions produce acquisition holes. The nature of urban data itself implies that the acquisition will produce a multitude of elements. For instance, the rendering of several layers of buildings and different streets (Figure 1.7(a)) can produce visualization problems. Such troubles can be benign, for example the fact that we see the inside of building walls on the bottom right of Figure 1.7(d). However, it can lead to wrong assignation between point and pixels during projection of the geometry that can impair a proper processing. Furthermore, the laser scanner provides a density of points that varies depending on the position of the acquired surface relatively to the acquisition device. For instance, the KITTI dataset has a point density that varies with the distance to the acquisition device. Chapter 5 proposes to improve the point cloud density by exploiting the picture information together with the acquired point set.

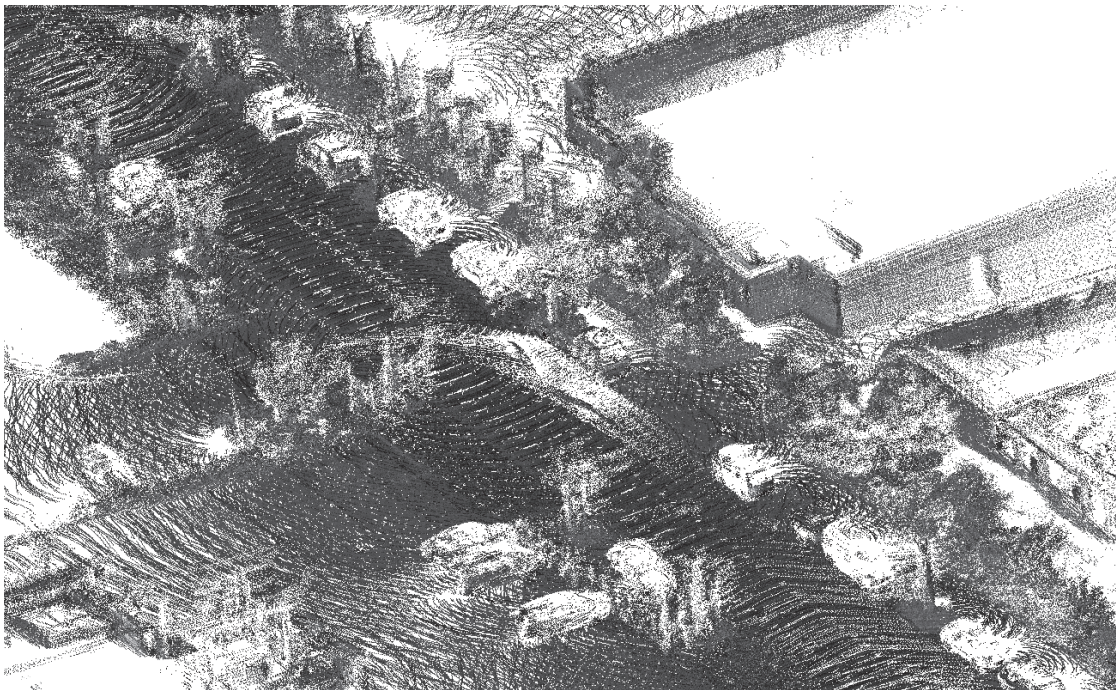


Figure 1.10: Several complete scans of the KITTI dataset seen from a top-down angle in 3DReshaper. The lack of color makes it difficult to distinguish the different elements of the scene.

CHAPTER 2

SYNTHETIC IMAGE GENERATION

Contents

More and more acquisition campaigns produce massive amount of data consisting in both point cloud and images. To be able to use a combination of both kind of data, it is necessary to be able to associate each point of the cloud to a pixel in an image. The usual solution is to cast the 3D data into 2D data, using a projection. To do so, we generate a synthetic image of the point cloud as seen by the camera that took one of the picture with the exact same projection parameters. This chapter details the synthetic image generation process, whose steps are depicted in Figure 2.1.

(a) Point cloud projection. (b) Occlusion estimation. (c) Interpolation.

Figure 2.1: Different steps to generate the synthetic image. The pixel intensities are computed using the point normals.

2.1 Projection model

Digital perspective cameras are far more represented than other types of cameras, such as Panoramic cameras, using cylindrical or fish-eye projection types. Such perspective cameras can be considered as relatively close to a pinhole camera system. In this ideal camera model, rays of light coming from the scene pass through an infinitely small hole in a closed box. These rays intersect at the camera center which is the pinhole. They are then projected on the box wall opposed to the pinhole, the image plane, where the rays form the inverted image of the scene. This projection model notably preserves the lines during the projection.

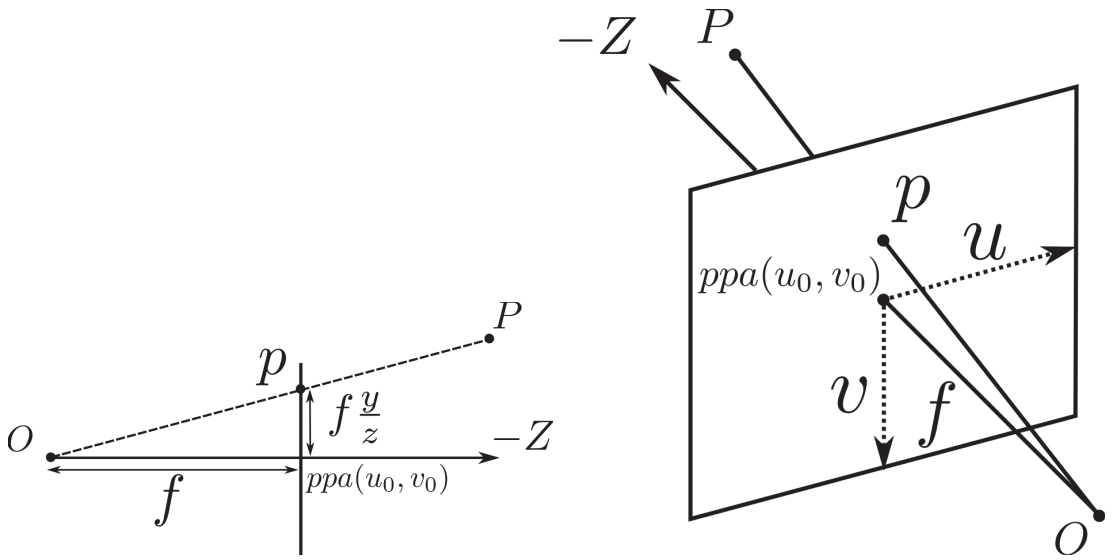


Figure 2.2: Visual representation of the pinhole camera model and internal parameters.

This simple projection model works well to represent the perspective projection that occurs in standard camera. This model is parametrized by two types of parameters: intrinsic and extrinsic. The intrinsic parameters represent the inner structure and mechanisms that happen within the camera. The extrinsic parameters represent the interaction between the environment and the camera. In the following section, the principle and the equations of this model will be detailed.

2.1.1 Camera coordinate system and intrinsic parameters

As one can see on Figure 2.2, to represent the geometry inside the camera, we define the camera origin O as the projection center (pinhole or lens position). The principal axis is the axis orthogonal to the image plane passing through O. In order to be consistent with the traditional image coordinates system, the principal axis will be the Z axis, and the scene will be oriented toward -Z.

To facilitate the representation of the projection model, we will also consider that the image

plane is in front of the projection center, leading to a non inverted image when observed from the camera origin. This image plane is situated at the *focal* distance f of the camera.

We define the camera principal point ppa as the intersection of the principal axis with the image plane. This principal point may not always be located in the center of the image plane, but may be shifted. The shift of the camera principal point to the origin of the image coordinate system is a part of the intrinsic parameters and can lead to a great difference in the projection. We consider (u_0, v_0), the position of the projection of the principal point. We also consider that ppa is the optical center, *i.e.* the center of the potential distortions that affect the rays. If the camera was perfect, this translation would be 0 ($u_0 = 0$, $v_0 = 0$).

The shear of the image coordinates system is sometimes taken into account to correct the skew between the image plane axes, but it is usually not present in real cameras. Hence, we discard this term in our projection model but one can safely ignore this shear for real camera.

To summarize, in the presented projection model we will consider the following intrinsic parameters:

- f, the focal distance of the camera.
- O, the optical center of the camera.
- ppa , the camera principal point located at coordinates (u_0, v_0), the location of the projection of the optical center in the image plane.

2.1.2 Distortions

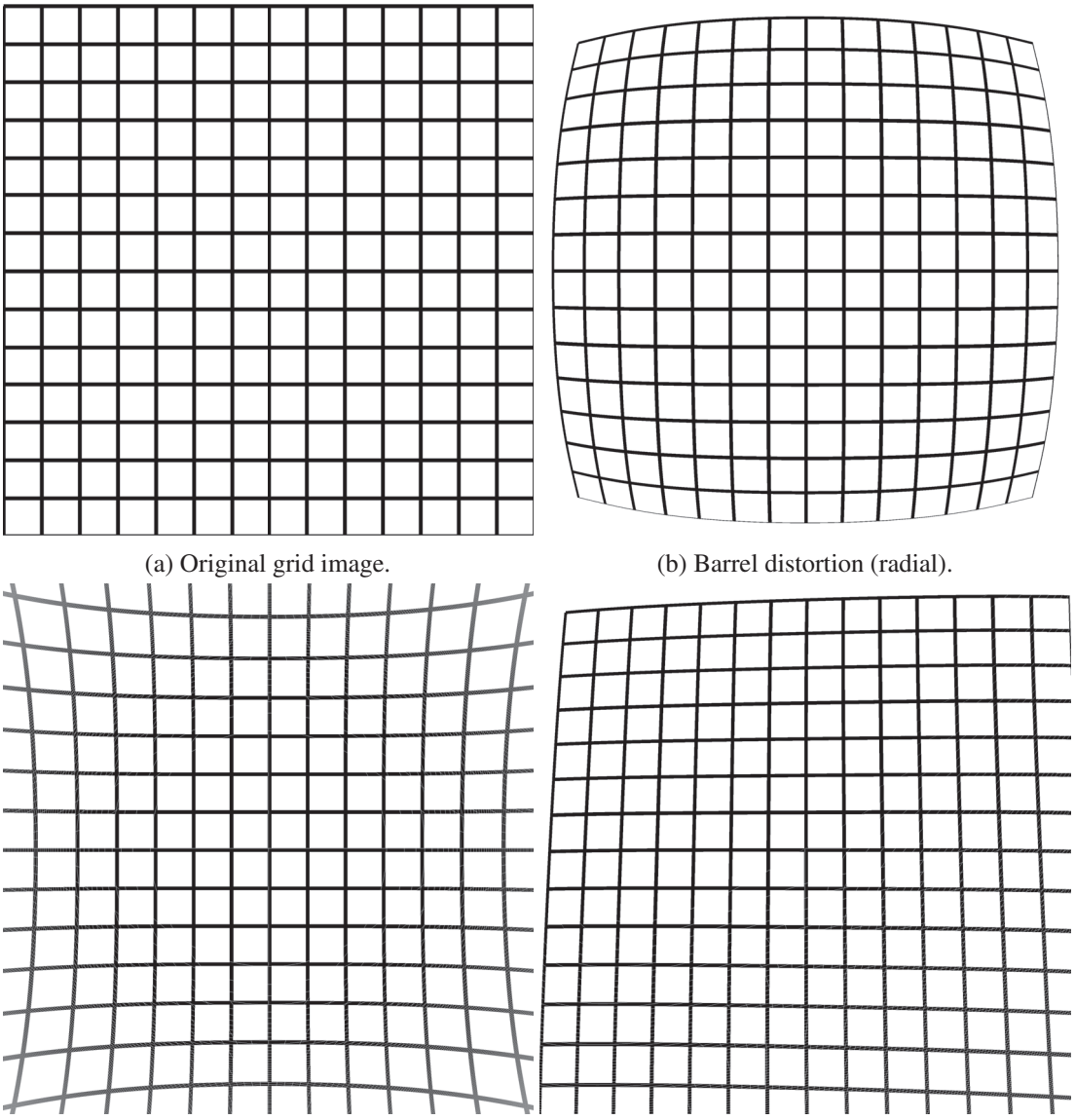
Assuming that these intrinsic parameters are known, this model does not render perfectly the observable reality: synthetic images will be different of the original real image. Indeed the use of lenses produces optical aberrations that leads to a deviation from the rectilinear projection expected with a pinhole camera. This implies that observable straight lines will not be preserved in projection. To model and correct these distortions, we use the Brown-Conrady camera distortion model [Bro66]. Using 6 radial and 2 tangential distortion parameters, this model is able to successfully remove (or at least mitigate) the distortion effects we observed in our data. These observable distortions are classified as (see Figure 2.3):

Barrel distortion (Figure 2.3(b)) is a radial distortion where the lens magnification decreases with the distance to the optical center. For instance images obtained using a Fisheye lens display significant barrel distortions. This type of distortion is modeled using the $k_{1...6}$ radial distortion parameters.

Pincushion distortion (Figure 2.3(c)) is a radial distortion where the lens magnification increases with the distance to the optical center. It is dual to barrel distortions and is also modeled using the $k_{1...6}$ radial distortions parameters.

Complex distortion is a radial distortion that starts as a Barrel distortion around the optical center, to evolve into a pincushion distortion further away. This type of distortion is less common than the Barrel and Pincushion distortions.

Tangential distortion (Figure 2.3(d)) arise from construction defects where the lens is not perfectly parallel to the image plane as stated by WENG, COHEN, and HERNIOU [WCH92]. These distortions, while usually less impacting than radial distortions, must still be taken into account. Tangential distortion is modeled by the t_1 and t_2 parameters.



(a) Original grid image.

(b) Barrel distortion (radial).

(c) Pincushion distortion (radial).

(d) Tangential distortion.

Figure 2.3: Different types of radial and tangential image distortions.

The details of the distortion equations are given in section 2.1.3. If the original distortion model takes into account up to 6 distortion parameters to model the radial distortions, we will only consider the first 3 in our calculation. Indeed, most calibration results only possess the first

3 parameters, leaving the 3 last to 0, as these last three parameters are negligible. To summarize, the considered distortions parameters are:

- t_1 and t_2, the two tangential distortion parameters.
- k_1, k_2 and k_3 the first radial distortion coefficients.

2.1.3 Projection equations

Considering we have the following camera intrinsic parameters, as defined in section 2.1.1 and 2.1.2:

- f, the focal distance of the camera and O the optical center of the camera.
- ppa , the camera principal point located at coordinates (u_0, v_0), the location of the projection of the optical center in the image plane around which the distortions are centered.

Let us also assume that we have an estimation of the camera extrinsic parameter (*i.e.* pose) in the scene Ω_0, composed of a rotation matrix R and a translation vector T. Let X, Y and Z be the coordinates of a point in the world coordinate system, x, y and z its coordinates relative to the camera coordinates system R (with the origin at the optical center O, and the z axis aligned with the principal axis, see Figure 2.2):

$$[x,y,z] = R \cdot [X,Y,Z] + T. \tag{2.1}$$

The projection of points along the optical axis is expressed using homogeneous coordinates: $x_0 = \frac{x}{z}$ and $y_0 = \frac{y}{z}$ so that it is possible to compute the distortion effects using the Brown-Conrady model [Bro66] and obtain the radially distorted points coordinates x' and y' as follows:

$$r = x_0^2 + y_0^2 \tag{2.2}$$

$$x' = x_0(1 + \sum_{n=1}^{3} k_n r^{2n}) \tag{2.3}$$

$$y' = y_0(1 + \sum_{n=1}^{3} k_n r^{2n}). \tag{2.4}$$

By taking tangential distortions into account, we obtain the values x'' and y'':

$$x'' = x' + t_2(r^2 + 2x_0^2) + 2t_1 x_0 y_0 \tag{2.5}$$

$$y'' = y' + t_1(r^2 + 2y_0^2) + 2t_2 x_0 y_0. \tag{2.6}$$

Using these notations, the image coordinates (u_{cam} and v_{cam}) of the projected point can be expressed as :

$$u_{cam} = fx'' + u_0 \tag{2.7}$$

$$v_{cam} = fy'' + v_0. \tag{2.8}$$

(a) Original photography. (b) Rectified image.

Figure 2.4: Comparison between the original photography and the rectified image. The original image suffers from barrel distortion, causing a loss of information near the image borders during rectification.

Another possible method to handle the distortions is to rectify the original photography of the scene (Figure 2.4). Starting from the rectified image allows to use the pinhole model directly, without taking distortions into account. Both methods (rectifying the image and applying the simple pinhole model, or taking distortions into account in the model and applying it to the original image) provide similar results.

2.2 Color information on synthetic image

The point cloud projection method described in section 2.1 only yields a binary image: pixels are lit if there is projected information and off if there is none. This projection is sufficient if we want to associate points from a point cloud to pixels of an image. However, this kind of representation is neither suitable for rendering nor for synthetic versus real image comparison. Therefore, a color should be assigned to the projected points. Several choices are possible to attribute a color

(a) Original corresponding photography. (b) Reflectance values. (c) Associated colors.

(d) Normal as *RGB*. (e) Sky based normal illumination. (f) Spotlight based illumination.

Figure 2.5: Different types of point cloud projection color attribution. Points are interpolated for visibility purpose.

to defined pixels. This choice depends mainly on two factors: the amount and type of information we have associated with each point of the cloud and the future application of the synthetic image.

The information associated with each point is the main limiting factor. The most basic form of unordered point cloud only possesses the 3D coordinates for each point. However, it is possible to reliably estimate the normal orientation of each point using standard procedures such as Principal Component Analysis (PCA) [Pea01]. This normal information can already allow for a variety of different information to be displayed. Indeed, one could consider that displaying the normal orientation in X,Y and Z axis as color in the *RGB* colorspace (Figure 2.5(d)) might be suitable for some visualization purpose. On another hand such information may not be adapted to outline scene surface salience when converted to a grayscale image. In this case, using the scalar product between the normal and the "up" axis, as proposed by TAYLOR and NIETO [TN13], to reflect the fact that most of the luminosity comes down from the sky can be a good choice (Figure 2.5(e)). However such lighting conditions, although they may be adapted to natural environment, give bad results in urban environment where the walls, which are usually perpendicular to the ground, offer a large variety of details that should be taken into account. Instead of using a direction

coming from the sky, one can use any other arbitrary direction and a natural choice would be one that enhances the details (Figure 2.5(f)). In our experiments, the best results are achieved using a lighting direction from the center of the camera. In that case, the computation of each point grayscale intensity breaks down to using the absolute value of the cosine angle between the camera direction and the point normal. This choice enhances interesting details on the building walls that would not appear as efficiently using other coloration methods.

A very interesting information that is often provided by modern LiDAR devices is the laser return intensities, or reflectance values. This value corresponds to the amount of photons that bounce back to the laser source for a given measurement. Reflectance values associated with each point are strongly dependent on the material that was hit by the laser beam. This value is also impacted by the angle of the beam to the measured surface, but the observed outputs of LiDAR are usually resilient to such variations and the reflectance values tend to vary only with material changes. This material dependency yields a strong correlation to other information, such as the color (Figure 2.5(b)).

Sometimes, color information can be given directly by the LiDAR. However, this given color is unfortunately often flawed with defects and an overall low spatial resolution. This is particularly true when considering urban data, where the scene contains moving objects and a high amount of furniture. A method to reliably attribute a high quality color information to the point cloud from multiple images is discussed in chapter 4.

Furthermore, it can happen that several points coming from different surfaces project onto the same pixel. It can be due either to a sampling density higher than the pixel size or to the fact that points from several surfaces (eg. buildings in different streets) can be seen through the foreground surface, since there is no watertight model and since we are dealing with large scale aggregated scans. In that case a choice should be performed to decide which piece of surface occludes the other by keeping the point that is closest to the camera position.

However, if there is not enough points on the closest surface, sparse sampling artifacts will appear as shown in Figure 2.1(a). The next step focuses on overcoming this limitation.

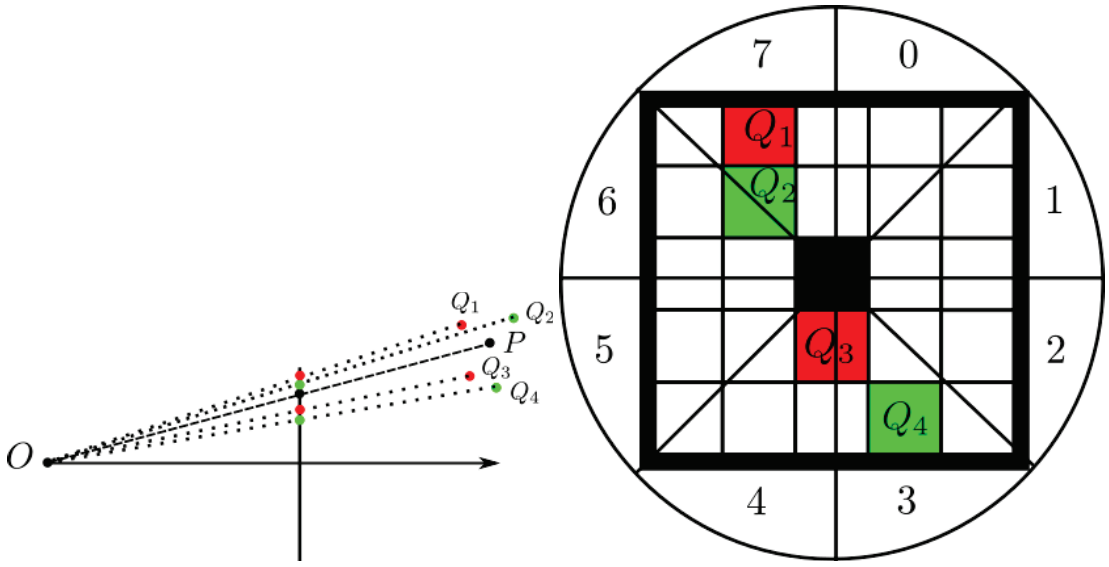
2.3 Reducing the sparse sampling artifacts

Since the image generation method typically operates on large point clouds created from large scale scans, a simple projection produces visual artifacts on sparsely sampled areas that occlude each other. These artifacts are created by points that become visible when they should not. This has a number of consequences, ranging from visual artifacts, to badly assigning information when working on real and synthetic images. It is possible to get rid of these artifacts in pinhole images, following the method proposed by PINTUS, GOBBETTI, and AGUS [PGA11] which we summarize briefly:

For each pixel p in the image plane corresponding to the projection of a 3D point P, the region in a $l \times l$ (usually $l = 9$) neighborhood around it is divided into 8 angular sectors. In a sector, for each 3D point Q projection which corresponds to a pixel q of this sector, we define the cosine of the visibility angle α_q relative to the pixel q as the angle between the vector $\vec{OP}$ and the

vector $\vec{QP}$ (see Figure 2.6) namely $\alpha_q =< \vec{OP}, \vec{QP} >$. For each of the sectors, only the highest α_q will be kept as the sector horizon such as : $\alpha_{sector} = max(\alpha_q)$.

This is illustrated in Figure 2.6(b) where the neighborhood around the considered point is divided into 8 enumerated sectors. For visualization purposes Figure 2.6(b) uses $l = 5$ instead of $l = 9$. Sectors 0, 1, 2 and 5 are empty. In sectors 3 and 4, the horizon is defined by point Q_3. In sector 6 the horizon is defined by the only point Q_2. However in sector 7 where both Q_1 and Q_2 project, the horizon is defined by point Q_1 that, as it can be seen in Figure 2.6(a), forms a much smaller angle.



(a) Side view. (b) Image view.

Figure 2.6: Projection of 5 points onto an image. The considered point and its projection on a pixel are depicted in blue. The visibility angle is the angle $< \vec{OP}, \vec{QP} >$. An horizon pixel is a pixel corresponding to the point with the smallest angle in a sector. These pixels are depicted in red, they are the points that best occlude the central point. Other pixels are in green.

Knowing the horizon in each sector allows to compute the visibility solid angle ω_p of the point P as:

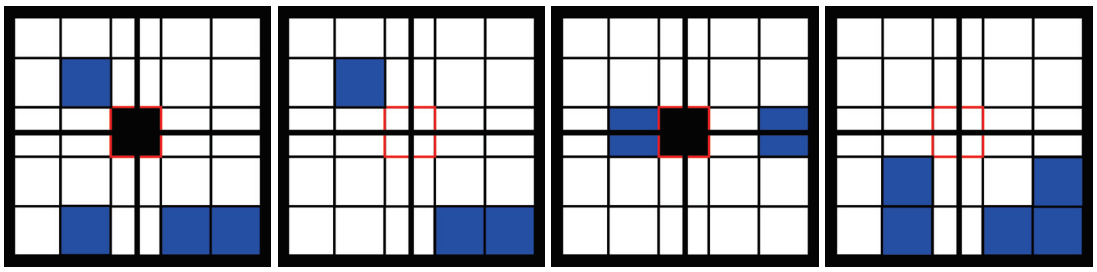
$$\omega_p = \frac{\pi}{4} \sum_{sector=0\cdots7} (1 - \alpha_{sector}) \tag{2.9}$$

If the solid angle ω_p is larger than a threshold value ψ the central pixel is classified as being visible. The value $\psi = 2.0sr$ is used for all experiments, except if stated otherwise.

This method successfully removes points that should not be visible, as shown in Figure 2.1(b). It can be noted that points may remain visible through large holes such as windows, as shown in Figure 2.1(b). This method however reduces the number of such erroneous points.

2.4 Interpolation

If the image is taken too close to the surface, a lot (if not most) of pixels do not correspond to a point of the point cloud. This information sparsity is not a problem in the case of information assignment. It can however be problematic for visualization purpose, or to compare real and synthetic images. To deal with missing information in the generated image, one has to retrieve information for undefined pixels. A simple bilinear interpolation on the available data, as proposed in [GARGGL09], leads to a widening of the edges (Figure 2.8), which could cause registration inaccuracy. To avoid that widening we propose to divide the neighborhood of the considered pixel into 4 sectors, and the interpolation is performed only if at least 3 of these 4 sectors contain pixels with data (see Figure 2.7). In some cases this may leave a lot of undefined pixels, but this is a good trade-off between edge location preservation and information addition. In our implementation we used a neighborhood of 5×5 pixels, a choice that reveals much needed details in the generated images, without affecting the edges of the scene (Figures 2.1(c) and 2.8).



(a) With 3 sectors containing data, the considered pixel will be interpolated.

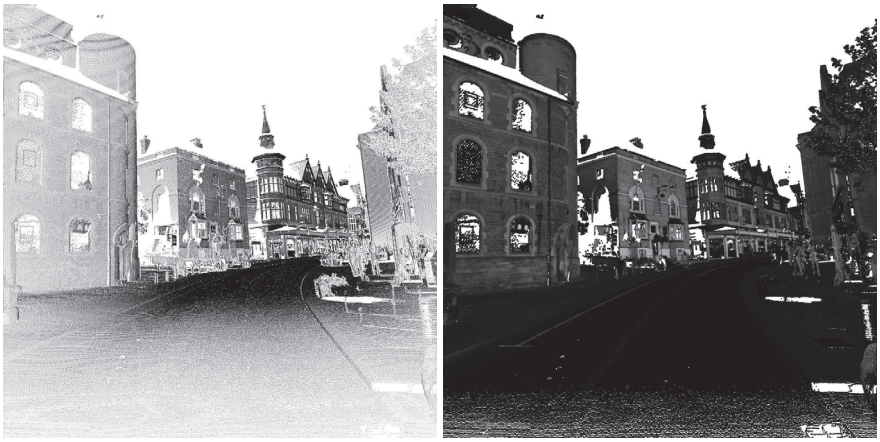
(b) With 2 sectors containing data, the considered pixel will not be interpolated.

(c) With 4 sectors containing data, the considered pixel will be interpolated.

(d) With 2 sectors containing data, the considered pixel will not be interpolated.

Figure 2.7: Example of the proposed interpolation scheme and its results. A 5×5 neighborhood is defined around the considered pixels. If a pixel is defined in this neighborhood, its corresponding sector will be considered as containing data. The interpolation of the considered pixel can happen only if 3 of its 4 sectors contain data.

Surface splatting is an alternative to produce synthetic images, which is relatively robust with respect to sparse sampling [Zwi+01]. In its simplest form, it consists in rendering points as surface elements. For instance, these points are often represented by oriented disks. Orienting the disks using their normal and with a radius that allows partial overlap of the disks allows for a good rendering of closed surfaces from a point cloud. Some methods propose to clip the surfels around sharp edges to take into account brutal variations of the surfaces. However, using a simple surfel rendering, as described in [Bot+05], also yields a widening of the edges (Figure 2.8(c)). Furthermore it generates artifacts when the surfel orientation is not well defined (such as in the trees).



(a) No interpolation. (b) Bilinear interpolation.

(c) Splat based rendering. (d) Edge preserving interpolation (ours).

(e) Bilinear (detail). (f) Splatting (detail). (g) Edge preserving (detail) (ours).

Figure 2.8: Effect of the bilinear interpolation versus an edge preserving bilinear interpolation. As can be seen in these images, our proposed edge preserving interpolation improves the data density sufficiently to give an idea of the image entropy, while preserving the details localization compared to a classical bilinear interpolation, whereas other methods give dense images but at the cost of an edge dilatation. The original data without interpolation is shown in magenta.

CHAPTER 3

IMAGE TO GEOMETRY REGISTRATION

Contents

Recent years have seen a fast development of acquisition technologies for acquiring urban scenes. Among all techniques, terrestrial laser scanners have gathered an important research interest. Acquisition campaigns covering whole cities have been led using LiDAR (Light Detection And Ranging) scanners onboard moving vehicles. The output of these campaigns consists in large, potentially unorganized, point clouds representing the buildings measured by the laser. These campaigns are often not limited to acquiring the geometry as a point cloud, but also embed other devices to measure various data. For example, this additional data can be a set of pictures taken at the same time as the points were measured. The set of pictures and the point clouds are usually aligned using onboard information such as GPS information or accelerometers. However, this initial alignment is often flawed which can lead to wrong interpretations in further processings using both points and pictures. This may be due to various factors such as sensor drift and uncertainty or even stability problems in the way the cameras are fixed to the vehicle.

This misalignment can be corrected interactively but it is time-consuming and intractable for large point sets and picture sets. It is therefore necessary to devise an automatic way to correct the registration starting from the approximate camera pose given by the acquisition device.

3.1 State of the art

Image to point cloud registration is a domain that was extensively explored in the past few years. Existing approaches can be roughly divided into four categories: *2D feature-based methods*, *statistical methods*, *3D based methods* and *skyline based methods*. The first two categories are the most explored in the literature. They share the common approach to cast the problem of image to point cloud registration as an iterative process of image to image registration. This implies that the point cloud should be turned into a synthetic picture on which the real picture can be registered. Using both images a first camera pose is estimated, a synthetic image is regenerated using this new pose and the process is iterated. The synthetic image can either be obtained directly from the LiDAR scanner which sometimes provides a spherical image, or by projecting the point cloud on the image plane and giving each point a color corresponding to some geometry properties (estimated normals for example, as explained in chapter 2).

2D Feature-based methods

Feature based registration methods rely on establishing correspondences between feature points obtained using methods such as SIFT [Low04], or SURF [BTVG06] on real and synthetic 2D images. These methods need to be applied on point clouds that already possess either a reliable color information, or a reflectance value as presented in [SW10], where a complete reflectance image is used to perform a SURF detection and matching.

Similarly, MOUSSA, ABDEL-WAHAB, and FRITSCH [MAWF12] rely on RGB coloration of the point cloud given by the laser to make comparisons between a synthetic image and a real image, using ASIFT [MY09] descriptors. Inconsistent correspondences are then removed by applying RANSAC [FB81]. Using the correspondences between 2D and 3D points, the camera pose is finally obtained by solving a Perspective-n-Point (PnP) problem using the EPnP algorithm [LMNF09]. YANG, BECKER, and STEWART [YBS07] propose a method to register an image on a shape using another image with a perfectly defined pose. First, SIFT descriptors are computed on the image with a known pose and associated with the point cloud by backprojecting them on the geometry. Real image SIFT keypoints are then computed and compared to the point cloud descriptors and the best matches are kept. Finally a two-step refinement is performed to obtain the camera position. This method does not require any prior estimation of the camera pose, but still needs a real image that has its pose perfectly defined relatively to the point cloud. GONZÁLEZ-AGUILERA, RODRÍGUEZ-GONZÁLVEZ, and GÓMEZ-LAHOZ [GARGGL09] propose a methodology that registers LiDAR range images generated from Terrestrial Laser Scans (TLS) and digital camera images using image descriptors. The real image is preprocessed to remove the distortions and its contrast is increased, followed by radiometric equalization and bilinear interpolation. Then a manual resizing operation is performed on the synthetic image to

fit the real image as well as possible. Feature points are detected and matched by combining cross-correlation, least squares matching and epipolar constraints. Finally the camera position and orientation is obtained using RANSAC. However it relies on high definition images to detect common features between LiDAR scans and photos. Furthermore, manual interaction is not possible for large datasets. A method guaranteeing the global optimality of the registration in case of points and lines within indoor scenes has also been proposed [BWG15].

PLOTZ and ROTH [PR15] recently described a feature based registration method using the average shading gradients to successfully register an image onto an untextured mesh object without any prior pose information.

Statistical methods

Methods using statistical analysis are widespread for aligning image to image (and thus image to geometry). Among all statistical methods, the most common metric is Mutual Information (MI). Proposed by VIOLA and WELLS III [VWI97], MI is a measure of the mutual dependencies between two random variables based on the Shannon entropy. For image registration, the two variables are the pixel intensities of both images. MI measures the similarity between two images based on the level of dependency of the intensity distributions. Thus in order to align an image to another, a good strategy is to find the pose that maximizes their Mutual Information ([Gon+14],[KKZ03]). Considering image to geometry registration as an iterative image to image registration, Mutual Information can once again be used to measure the quality of the registration. Variations on the original metric have been later proposed: for example the Normalized Mutual Information (NMI) [SHH99] uses normalized values, while the method from GONG et al. [Gon+14] add SIFT information to the mutual information. Several works have investigated Mutual Information in the context of comparing an image with some geometric information. CORSINI et al. [Cor+09] presented an in-depth discussion about which combination of geometric properties should be used to achieve the best results, *e.g.* normal maps, intensities or even a mixture of several modalities. MASTIN, KEPNER, and FISHER [MKF09] successfully used Mutual Information to correct small rotational errors in the registration of urban aerial images on the corresponding aerial Lidar data using elevation and reflectance data of the Lidar. In the case of data acquired by a mobile LiDAR acquisition system, MI has been used to obtain the position and orientation of an image relatively to the point set, using the similarities between images and scanner intensities [Pan+12]. TAYLOR and NIETO [TN13] propose a calibration framework that estimates the camera pose with no other information than the normal of the scanned points. To do so, a modified form of the Mutual Information is maximized using particle swarm optimization. Although this method gives good results, it suffers from several drawbacks. First, its high memory demand makes it impractical for large complex point clouds such as the ones we consider in the paper. Furthermore, it does not propose a way to cope with multiple depths layers that can be seen from a same viewpoint when the sampling is not dense enough. Another drawback lies in the dependency on panoramic spectral photography, a type of photography which provides a larger area of common information between the point cloud and the image leading to a more robust registration. When applying this method to regular images, as the ones available in our datasets, it does not work as well since regular images have less overlapping information. TAYLOR, NIETO,

and JOHNSON [TNJ13] further improved their method by introducing a gradient based metric called Gradient Orientation Measure (GOM) instead of Mutual Information. GOM computes the difference of the gradient orientation angle between the synthetic image and the real image. This method improves the accuracy of the result compared to Normalized Mutual Information (NMI). To alleviate the computation cost of the synthetic image after each particle motion, another improvement is to use spherical images. However we will show that using a single metric of comparison is generally not discriminative enough and that better results can be obtained by combining several measures to highlight the differences at different scales.

Statistical methods for image to geometry registration are an active field of research. For example, PASCOE, MADDERN, and NEWMAN [PMN15] recently introduced a Normalized Information Distance metric, based on Mutual Information and entropy variation, to retrieve the camera position in an urban environment.

3D based methods

In sharp contrast with the first two categories, some methods propose to use 3D reconstruction and then 3D matching to achieve proper registration. For example, CORSINI et al. [Cor+13b] start by performing a Structure From Motion (SFM) reconstruction from an image dataset. SFM is a powerful and widely used method that reconstructs a set of 3D points using a set of images capturing the scene with a small variation in position and orientation. Usually, common features are identified on this set of images using descriptors such as SIFT or SURF. The variations of these identified descriptors allow to reconstruct the descriptor 3D position and thus camera positions. This type of reconstruction is powerful but only gives a small amount of 3D points. The reconstructed points and the relative pose of the images are then fitted to an existing, denser point cloud using a scale independent version of 4-Points Congruent Sets [AMCO08]. After merging the sparse point cloud with the complete point cloud, the camera pose is estimated and then refined using Mutual Information maximization. MOUSSA [Mou14] also use Structure From Motion to perform image to geometry registration, but instead of doing a full 3D registration, they register images to the polar laser intensity images that are generated during the scanning process. Correct matches can be obtained from real and synthetic images using standard image descriptors. Thus, the polar image pose is obtained through SFM which finally yields the actual image pose estimate.

Skyline based methods

Skyline registration methods aim at retrieving the camera pose by analyzing the uniqueness of the skyline in urban environments. The color differences between the sky and the buildings are used to register single camera images on a corresponding point cloud, starting from an estimated pose. Although this approach is less spread, HOFMANN, EGGERT, and BRENNER [HEB14] proposed to rely on the skyline of the buildings. The sky is first automatically extracted in the real images and independently in the synthetic image using pixel intensity thresholding. The outline is then computed and refined. Extracted skylines in both images are then merged using a modified ICP method in the image plane. After this fusion, a better camera pose is estimated. These methods

are adapted to large scale cities but unfortunately suffer from relying only on the skyline. Hence, every problem on the skyline such as missing data, jagged skyline, or even too much vegetation, is likely to affect the results, or not give any result at all if no building skyline is visible in the images.

Our work focuses on the case of complex, large scale urban scenes. The data is acquired using a mobile LiDAR system, which gives as output clouds of several million points with coarsely registered corresponding images. The images are given by *in situ* CCD perspective cameras, that can be considered as standard digital cameras with a narrow field of view. Despite the fact that some scanners provide interesting properties such as the reflectance for each scanned point, we develop a more general framework that is able to perform the whole registration process using only the geometry data, *i.e.* the spatial positions. This is interesting when the laser intensity information is not embedded in the data format, or with a view to extending the approach to other acquisition devices. When the synthetic images are based only on the geometry, the descriptor-based methods fail to align our rendered images. Besides, the geometry of urban scenes makes it difficult to use 3D based methods since the presence of vegetation, moving vehicles and pedestrians make Structure From Motion methods fail. This would yield a flawed reconstructed point cloud leading to a wrong registration. These methods also require a large number of images taken with a small variation in space. Such data may not always be available. Similarly, skyline based registration is not always applicable in urban contexts due to the presence of either jagged or partial skyline or even a total lack of skyline in some images. Statistical methods are thus a better choice in our case. Yet in our urban context the Mutual Information objective exhibits a highly non convex profile because of the sparsity of the synthetic images. To be able to minimize this objective, one could resort to the proposed solution of [TNJ13] to perform particle swarm optimization. Unfortunately this method does not provide a way to cope with occlusions arising from multiple scenes layers, such as the ones described in section 2.3.

3.2 Robust comparison of synthetic and real images

Our image to geometry registration method heavily relies on 2D images comparison. We must therefore define an image comparison metric, to determine if two images represent the same scene. In our case, this metric should be resilient to noise and incomplete data. This is even more important since the modality is not the same in both images. Indeed, in our case the real image encodes the color information while the synthetic image encodes an intensity derived from point normals or point reflectance. In this section we present a new way to compare images, building on two existing approaches, Normalized Mutual Information (NMI) and Histogram of Oriented Gradients (HOG), which are detailed below.

3.2.1 Normalized Mutual Information

Mutual Information, as introduced in section 3.1, is widely used for image registration based on image comparison even in the case of different modalities, as stated by KIM, KOLMOGOROV, and ZABIH [KKZ03]. We focus here on Normalized Mutual Information (NMI), a modified

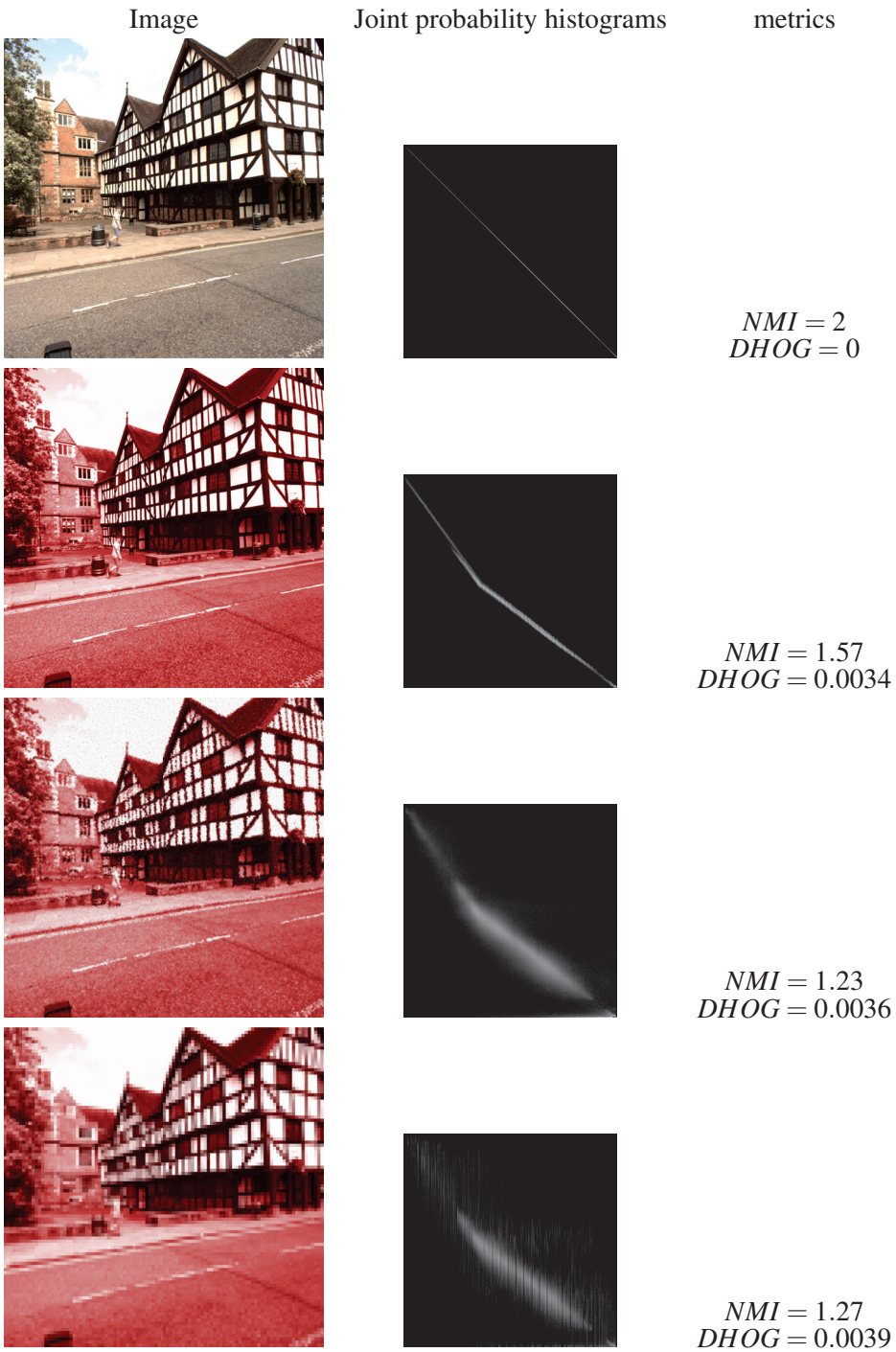


Figure 3.1: Comparison of the NMI and DHOG metric on several variations of the same image. All results are obtained by performing a comparison of the image depicted on each row to the image of the first row. The second column represents the joint probability histogram used in the NMI computation which gives a good visual clue of the image similarity.

version of MI proposed by STUDHOLME, HILL, and HAWKES [SHH99]. This normalized version ensures that the MI values are bounded. This also allows to compare images when the amount of known data varies.

The NMI between two images I_1 and I_2 is defined as:

$$\mathrm{NMI}(I_1, I_2) = \frac{H(I_1) + H(I_2)}{H(I_1, I_2)}. \tag{3.1}$$

where $H(I)$ is the image entropy defined as:

$$H(I) = \sum_i p_i \log(\frac{1}{p_i}) = \sum_i -p_i \log(p_i). \tag{3.2}$$

and p_i is the probability for an arbitrary pixel k of image I to be of intensity i (*i.e.* $p(I(k) = i)$). Hereafter, N is the total number of pixels in the image I and $T(\cdot = \cdot)$ equals 1 if $\cdot = \cdot$ is true, and 0 otherwise. Undefined pixels in the synthetic images are not considered in the NMI calculation. The probability p_i is defined as follows:

$$p_i = \frac{1}{N} \sum_k T(I(k) = i). \tag{3.3}$$

Let $p_{(m,n)}$ be the joint probability that pixel k has an intensity m in image I_1 and n in image I_2. The joint entropy $H(I_1, I_2)$ is defined as:

$$H(I_1, I_2) = \sum_m \sum_n -p_{(m,n)} \log(p_{(m,n)}). \tag{3.4}$$

Image to image registration using NMI is efficient in most cases, giving a measure varying between 1.0 for no mutual information to 2.0 for two identical images. This variation can be seen in Figure 3.1 where a single image is compared to several slightly modified versions of itself. Increasing the differences with the original image will change the mutual information value. Use of this metric in a maximization framework allows for the correct registration between two different images.

However, due to the amount of missing data in our synthetic images, NMI sometimes exhibits important flaws. Among others, non-convex profile of this measure might lead to an error in the maximization process causing a wrong registration, or the global maximum might not correspond to the actual image superposition (see Figure 3.3). This can be explained by the fact that NMI, and MI in general, takes the whole image into consideration. We propose to re-introduce some spatial locality in the analysis. Interestingly, another attempt at localizing MI was proposed in Pixel-Wise Mutual Information [Gon+14], but in case of images using only normal information this metric exhibits a highly nonconvex profile making it impossible to recover a good registration. Our proposed approach works differently as it combines NMI with local gradient histograms.

3.2.2 Local image metric

In this section, we introduce a metric based on the spatial distribution of intensity gradients called *Distance between Histograms of Oriented Gradients* (DHOG). It corresponds to a localized integration of distances between local Histogram of Oriented Gradients (HOG) that we briefly describe.

HOG

HOG, introduced by DALAL and TRIGGS [DT05], is a feature descriptor characterizing image areas using their gradient information. HOG is widely used to compare and match images [Shr+11], or to detect objects in images.

The Histograms of Oriented Gradients of an image can be obtained by computing the gradients on the whole image. Then the image is divided into regularly sized patches called cells. Orientation-based histograms are then computed within each cell. Each pixel in each cell contributes to one bin of the histogram with a weight depending on the magnitude of the gradient. Once a histogram has been obtained within each cell, the cells are grouped by blocks. For each of those blocks, a normalization factor is computed. The blocks overlap to produce resilience to illumination and contrast change. Therefore we have $n_{blocks} \times n_{cells\,per\,blocks}$ normalized histograms.

DHOG

HOG is usually computed on sliding windows, to detect known size patterns in an image. However, here we consider whole images, with fixed cells position. Let us consider two images I_1 and I_2 on which we compute the cells histograms of oriented gradients as described previously. To quantify the similarity between these two images, we integrate the square distance between each corresponding pair of histograms.

Since we operate on an inaccurate projection model, it is better to favor image similarity in areas that are less subject to distortion. In pinhole camera models, radial distortions affect the borders of the image, rather than the image center. Using a weighted sum of squared differences between HOG will give more importance to the registration error near the center of the image, and help the registration even when the image distortions are not well defined.

Besides alleviating the bad calibration, this weighting scheme also increases the registration accuracy: on the 45 images groundtruth, it improved the registration accuracy by 3 pixels and the registration success ratio by 4% in average.

Denoting *wb* and *hb* the image width and height in numbers of blocks, we define the weight $w_{B_{ij}}$ of a block B_{ij} centered at coordinates (i, j) as:

$$w_{B_{ij}} = \exp - \frac{(i - \frac{wb}{2})^2 + (j - \frac{hb}{2})^2}{(\frac{wb}{2})^2 + (\frac{hb}{2})^2}. \tag{3.5}$$

This weight will be close to 1 around the image center, and will decay as the considered blocks are closer to the picture's border.

Due to the particular structure of the HOG data, given a real image I_r and a synthetic image I_s this integration is a function of the values $v^{I_r}_{cbij}$ and $v^{I_s}_{cbij}$ of a HOG bin b in a cell c belonging to a block B_{ij} (located at coordinates i, j):

$$\text{DHOG} = \frac{\sum_{i,j}\sum_{c}\sum_{b}\left(v^{I_r}_{cbij} - v^{I_s}_{cbij}\right)^2 \times w_{B_{ij}}}{\sum_{i,j} w_{B_{ij}}}. \tag{3.6}$$

When applied on images with only little texture, such as normal based synthetic images, DHOG performs much better than NMI. However on images with a lot of texture, NMI gives more accurate results. Thus, by combining NMI and DHOG, we are able to overcome the failure cases of both as illustrated in Figure 3.3.

3.2.3 Combined image metric

As shown in Figure 3.3, the metric variation of NMI and DHOG are different for the same transformation. Interestingly their defects appear in different cases. Based on this observation we combine NMI and DHOG so that a proper coarse registration can be achieved where either one of the metric would tend to drift to a wrong position. MIDHOG is based on the dissimilarity of the images, a value of 1.0 represents two images where gradients are opposite from one another, which is rather uncommon. On a set of 20 images we observed that NMI values usually vary between $[0.87, 0.96]$ whereas DHOG varies between $[0.033, 0.058]$. Their combination gives best results using a simple addition of the components (see equation 3.7), DHOG being weighted by a parameter α.

$$\text{MIDHOG} = (2.0 - \text{NMI}) + \alpha \cdot \text{DHOG} \tag{3.7}$$

MIDHOG inherits the properties of both MI and DHOG, it is zero when the two compared images are totally identical and it is symmetrical.

An error study of the coarse registration error compared to our ground truth (section 3.4.1) shows that an α value between 5 and 20 gives similar and satisfactory results (see Figure 3.2) whereas relying too much on NMI fails to properly register the images. On the other hand, increasing the weight of DHOG too much produces unsatisfactory registration in some images. A good trade-off was obtained using $\alpha = 10$ which is used in the remainder of this work.

3.3 Registration method

Given an input point cloud data and a real image with initial pose estimate Ω, our goal is to find a refined pose estimate Ω'. We introduce a two step registration consisting in a fast but coarse registration, optimizing only for the rotation, followed by a slower fine-scale registration optimizing for both rotation and translation.

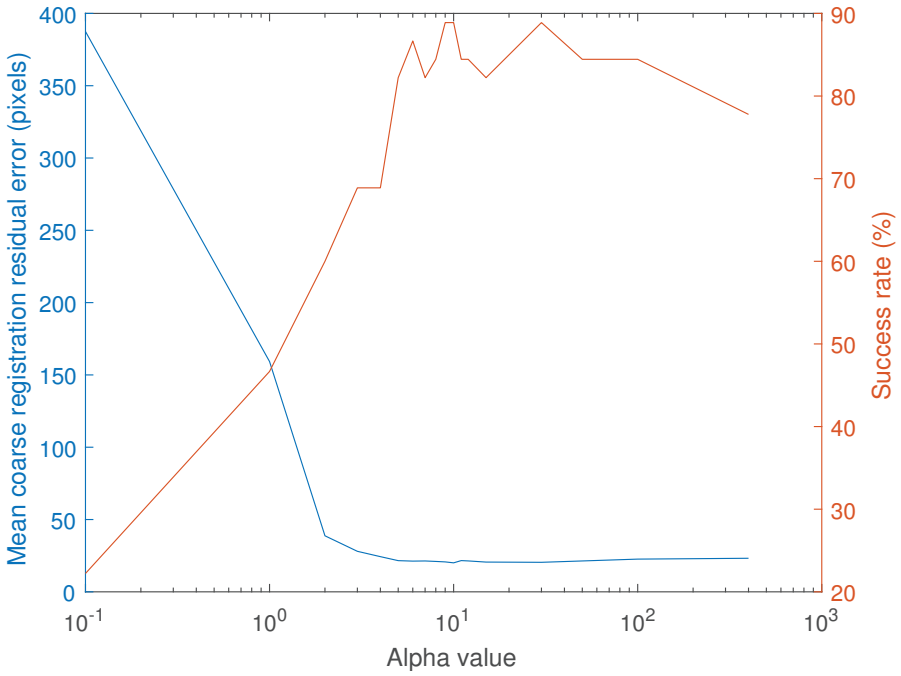
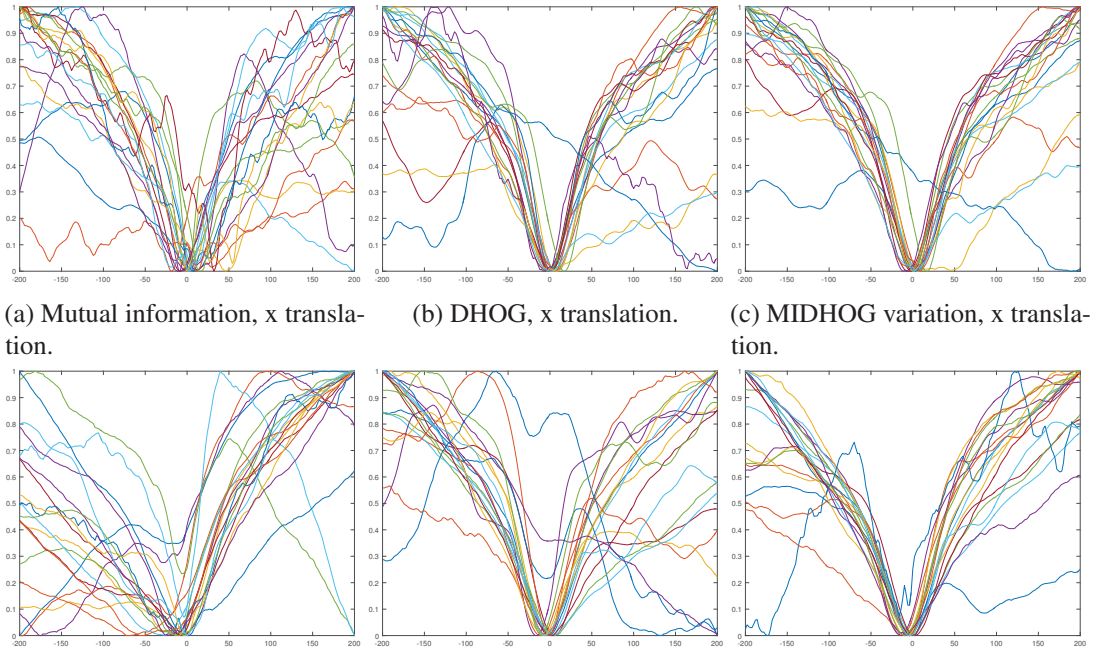


Figure 3.2: Average error after a coarse registration for different choices of α values for 45 groundtruth images with random initial disruptions.

3.3.1 Overview

Our method, summarized in Figure 3.4, takes as input a point cloud of an urban scene and a corresponding picture with initial pose estimate and known intrinsic parameters. Such an image can be acquired during the scanning process by a camera mounted on a vehicle or from a standard digital camera handled independently. We propose a two-step registration method to refine the camera pose, knowing the camera intrinsic parameters and a reasonable initial pose estimation. First a wide angle synthetic image is generated and used to optimize the camera pose with 3 degrees of freedom (optimizing only for a rotation). This rotation estimation is performed, in a multiscale fashion thanks to our new metric, called MIDHOG, that combines Normalized Mutual Information and Histogram of Oriented Gradients (HOG) descriptors to measure the consistency of the real image with a part of the wide angle synthetic image.

Starting from the refined orientation, the fine registration step gradually performs a full 6 degrees of freedom pose estimation. During this second step, two strategies are available: either using the MIDHOG metric, to ensure the accuracy, or to replace it only with DHOG, to improve the computation time at the cost of losing some precision.



(a) Mutual information, x translation.

(b) DHOG, x translation.

(c) MIDHOG variation, x translation.

(d) Mutual information, y translation.

(e) DHOG, y translation.

(f) MIDHOG variation, y translation.

Figure 3.3: Variation of three image comparison metrics: NMI, DHOG and MIDHOG. The top row corresponds to a per pixel translation on the horizontal axis of the subimage in a wide angle image (see Figure 3.5), and the bottom row corresponds to a translation in the vertical axis.

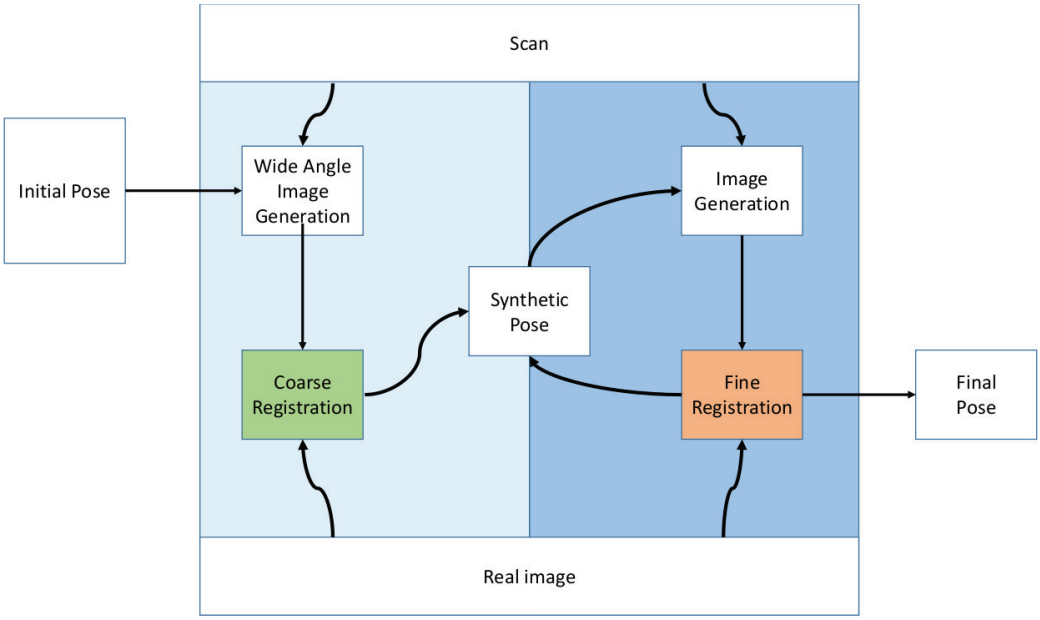


Figure 3.4: Overview of our method.

3.3.2 Coarse registration

A first and fast registration step is performed by generating a *wide angle* image of the point cloud from the initial camera pose and by refining 3 degrees of freedom on the location of this pose. For the sake of clarity, an overview of this coarse registration step is given in Algorithm 1.

Data: Ω: initial pose
$\mathscr{P}$: camera intrinsic parameters
I_R: real image
Result: Ω_1 a coarsely estimated camera pose
$scale = 1/4$;
PixelMotion $= 0$;
$\Omega_1 = \Omega$;
while $scale \leq 1/2$ **do**
  I_S = Generate synthetic wide image using $\mathscr{P}$, Ω_1 and $scale$;
  $(\delta x, \delta y, \theta)$ = Minimize MIDHOG between I_R and I_S using an initial step of $1/20$ of the image pixel size;
  Ω_1 = Obtain 3D pose from triplet $(\delta x, \delta y, \theta)$;
  PixelMotion = $\max(\delta x, \delta y)$;
  if $PixelMotion \leq 10$ **then**
    $scale = 2 \times scale$;

ALGORITHM 1 – Coarse image registration step.

The rationale behind this first step is that a single wide-angle image can be substituted to several steps of regular-size synthetic image generation. A small pitch and yaw rotation or translation of the pose will only marginally distort the pixels but will affect the position of the image center in the image plane. Thus a small pitch and yaw rotation or translation of the pose can be approximated by a small translation in the wide-angle image plane as depicted in Figure 3.5. On the contrary the roll rotation corresponds to a rotation around its center in the wide-angle image plane. Instead of generating a new synthetic image after each small motion, a single wide-image is thus generated and its sub-images are considered as good approximates of smaller images after a *small* viewpoint change. As a side-effect, it will also produce smoother metric variations than the one observed when performing a 3D rotation of the camera.

This approximation can hold for both small rotations and small translations, however we observed experimentally, during manual image registration, that the errors in the image registration are mostly due to rotations. Therefore we omit the translation in this coarse step and optimize only for the rotation (3 degrees of freedom). To be even faster, we do not optimize for all rotations as real rotations but rather approximate the Yaw rotation by a translation along the x axis in the image plane and the Pitch rotation by a translation along the y axis in the image plane. The last rotation, the Roll, is computed as a rotation in the image plane around the image center.

To generate a wide angle image, the internal parameters of the camera are kept identical to those given as input, but the size of the captor and the resolution of the image are increased. In the following explanations, images were generated using a double sensor size and resolution.

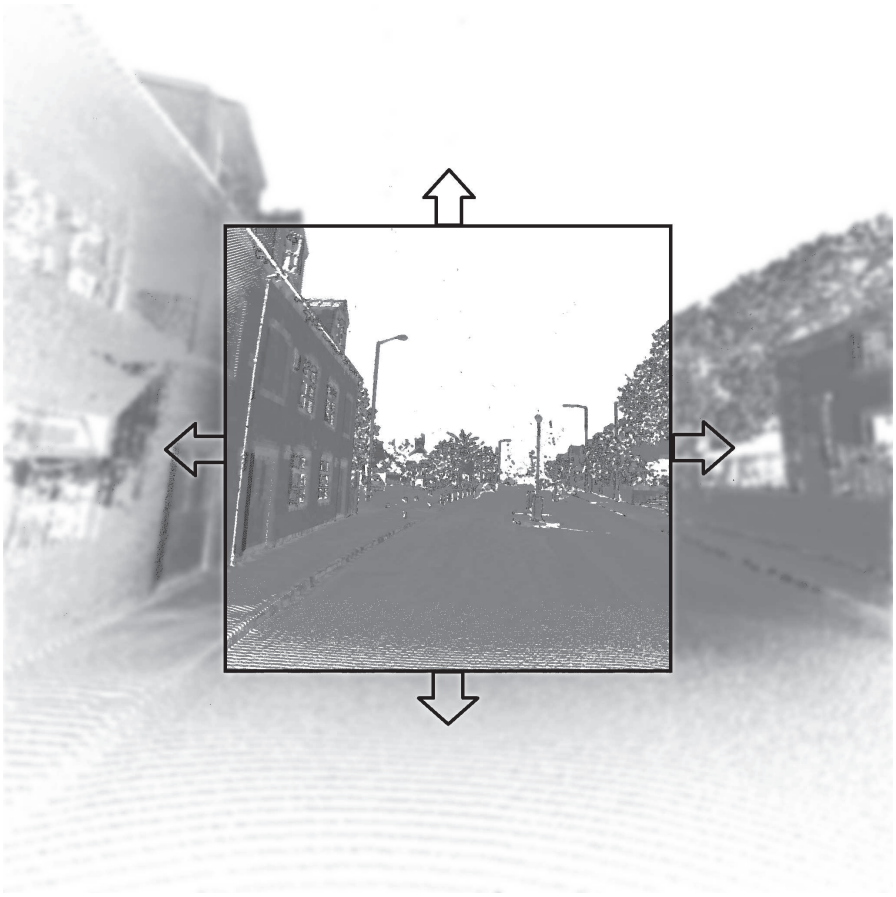


Figure 3.5: A small pitch and yaw rotation or translation of the pose can be approximated by a small translation in the wide-angle image plane.

Starting from the wide image, the sub-image that would correspond to an image generated with the standard parameters is extracted. Applying a translation in the X and Y direction of the image is, if the variation is small enough, very close to performing a real world rotation (see Figure 3.5). This way, we approximate the image synthesis without having to regenerate a synthetic view of the point cloud. By performing a match limited to 3 degrees of freedom (x-y translation and a θ rotation around the central pixel), an approximation of the image with a *pseudo-pose*, *i.e.* a pose in the image plane, can be obtained quickly.

Unfortunately, for larger camera motions the hypothesis does not hold. To cope with this problem, we iterate this step several times, regenerating a wide angle image after each step, or if the estimated change is superior to 10 pixels. This leads to images with less deformation after each iteration, allowing for an efficient convergence toward the metric minimum.

To further improve both the computation time and the convergence of the method, we perform this wide angle image registration at different scale levels. A Gaussian pyramid is first built

from the image. Then, the smoothest image is considered. Indeed, the details of the image are smoothed out as is the noise, leading to a smoother cost function easier to minimize. Once this first minimum is found, the next level of the pyramid is considered and the objective is once again minimized starting from the pose found at the previous level. Each iteration provides thus a better accuracy by increasing the resolution of the image and re-generating the view while taking into account the pose robustly estimated from the previous iteration. If the resulting estimated plane translation is too large, the assumption that the translation in the panoramic image plane approximates a real-world rotation is not valid anymore. Therefore the image is re-generated at the same scale and the process is repeated.

The quality of the registration is evaluated using our MIDHOG metric. To find the camera pose minimizing MIDHOG (Eq. 3.7), we use the BOBYQA algorithm [Pow09] and stop when the pose variation is small enough (2 pixels in our implementation). BOBYQA is a deterministic, derivative-free optimization algorithm that relies on an iteratively constructed quadratic approximation. It shows the same kind of flaws as the method presented by [MKF09], such as the difficulty to overcome local minimums, as pointed out by TAYLOR and NIETO. A better alternative would be to use Particle Swarm Optimization with a meaningful number of particles but it would become computationally intractable. Fortunately, in our case the local minimum problem that prevents the use of BOBYQA, is smoothed by the multi-scale approach proposed in this coarse registration.

For each iteration, we consider the two vector $\vec{dir}(0,0,f)$ and $\vec{ndir}(\delta_x,\delta_y,f)$, the original view direction and the modified view direction respectively, with δ_x and δ_y the translation found in pixels and f the focal length in pixels. We define $\vec{d_{yaw}}$ and $\vec{d_{pitch}}$ projections of $\vec{ndir}$ on the image horizontal and vertical plane passing through the image optical center. Using these vectors we can determine the rotations ω (yaw) and ϕ (pitch) corresponding to the pseudo-pose estimation using the equations 3.8 and 3.9. The roll is itself not considered as a translation in pixels, but is estimated by performing a rotation of the pixels in the image plane around the transformed image central axis.

$$\omega = -sgn(\delta x)atan2(\|\vec{dir} \times \vec{d_{yaw}}\|_2, \vec{dir} \cdot \vec{d_{yaw}}) \tag{3.8}$$

$$\phi = -sgn(\delta y)atan2(\|\vec{dir} \times \vec{d_{pitch}}\|_2, \vec{dir} \cdot \vec{d_{pitch}}) \tag{3.9}$$

$$atan2(y,x) = \begin{cases} arctan(\frac{y}{x}) & if\, x > 0, \\ arctan(\frac{y}{x}) + \pi & if\, x < 0\, and\, y \geq 0, \\ arctan(\frac{y}{x}) - \pi & if\, x > 0\, and\, y < 0, \\ +\frac{\pi}{2} & if\, x = 0\, and\, y > 0, \\ -\frac{\pi}{2} & if\, x = 0\, and\, y < 0, \\ undefined & if\, x = 0\, and\, y = 0. \end{cases} \tag{3.10}$$

The result of this coarse step is a modified pose Ω_1, that will be further refined in the following fine registration step.

	Coarse Registration	Fine Registration
Cost function	MIDHOG	MIDHOG (precision) / DHOG (speed)
Multi-resolution	Yes	No
Parameters	Rotation (3DoF)	Rotation + Translation
Parallax	No	Yes

Table 3.1: Comparison of the differences and similarity between the coarse and the fine step of the presented method. We do not consider any parallax in the coarse registration step, as we do not modify the camera position, but first try to determine the best viewing angle of the scene.

3.3.3 Fine registration

Having obtained a first estimation of the pose efficiently, the fine scale registration consists in estimating the real pose Ω' not far from the coarse estimation Ω_1 by considering the full 6 degrees of freedom. Despite the non convex form of the similarity metric with respect to the pose, we can find a satisfactory local minimum since Ω_1 is close to Ω'. For that, we rely once again on the BOBYQA algorithm [Pow09] to perform the derivative free minimization. Similarly to the coarse step, the MIDHOG metric defined in equation 3.7 allows for a better camera pose estimation, especially if the synthetic image is sparse. However, if the priority is given to the computation speed at the risk of losing some precision, it is safe to drop the NMI component of MIDHOG to rely solely on DHOG. This increases drastically the processing speed. Interestingly, this substitution can be done relatively safely only in the fine registration step since the search is limited to a narrow band around the pose Ω_1 found in the coarse registration step.

Contrarily to TAYLOR, NIETO, and JOHNSON [TNJ13], we do not need to perform particle swarm optimization since the first step has given an approximation *close to the global minimum*, where the metric behaves like a smooth convex function. This leads to a much lower computation time.

To recap our two step method, Table 3.1 lists the differences between the coarse and the fine registration steps.

3.4 Results and comparison

3.4.1 Groundtruth dataset

To evaluate the performances of the algorithm, we built a special groundtruth dataset using manual registration. First a series of 45 images was randomly selected among all pictures taken along the path of the vehicle mounted LiDAR. These images were then rectified using the given intrinsic parameters and manually registered to the point cloud by selecting 20 to 40 corresponding points both in the images and on the geometry. This dataset of 45 registered images is considered as an acceptable ground truth. However it should be noted that even if a special care was taken to select relevant common points, the registration in some images is still imperfect and may

eventually lead to small disturbance. Besides the rectified images are only an approximation of the undistorded reality and may contain residual errors that might impact the quality of the registration. To evaluate quantitatively our proposed method, a perturbation was applied to each groundtruth camera pose, consisting of a random uniform variation up to 5° in both yaw and pitch, up to 2° in roll and up to $10cm$ in X, Y and Z. This yields a set of images with perturbed camera pose that can be registered to the pointset using our method. Since the perturbation is known, the quality of the registration can now be evaluated with respect to the manual groundtruth. Setting up this groundtruth led us to the observation that the average registration error is around $6cm$ with errors up to $30cm$ in the estimation position and around 1° in yaw and pitch with peaks up to 4°. Errors in roll are always smaller than 1°.

The point cloud was automatically cut using a bounding box in a large area around the selected camera initial pose to limit the memory impact. Our tests were run on a laptop (Intel Core i7 2.7GHz CPU, NVIDIA Quadro K3100M), with approximately 100 million points processed at a time. Of course, larger point clouds can be loaded at once if enough memory is available. A first pre-processing step was performed on the real images to convert them from RGB to grayscale images using the standard Luma rec 601 conversion. The whole method was implemented in C++ using the NLOPT [Joh] library for the BOBYQA algorithm to minimize MIDHOG.

3.4.2 Interpolation scheme effect

As it was briefly outlined in section 2.4, the sparsity of the information of synthetic images can be a problem to perform proper comparison. As previously stated, a simple bilinear interpolation or a simple splatting of the point cloud would lead to a widening of the geometry edges.

The influence of our edge-preserving interpolation scheme on the whole registration process is evaluated in Table 3.2. Performing a registration without any interpolation produces a significantly higher residual error. The gain of using a border preserving interpolation method further improves the accuracy at a very small additional computation cost. Even if this improvement is not as big on average as one could expect, it has proven to be useful in some cases (as illustrated in Figure 3.6).

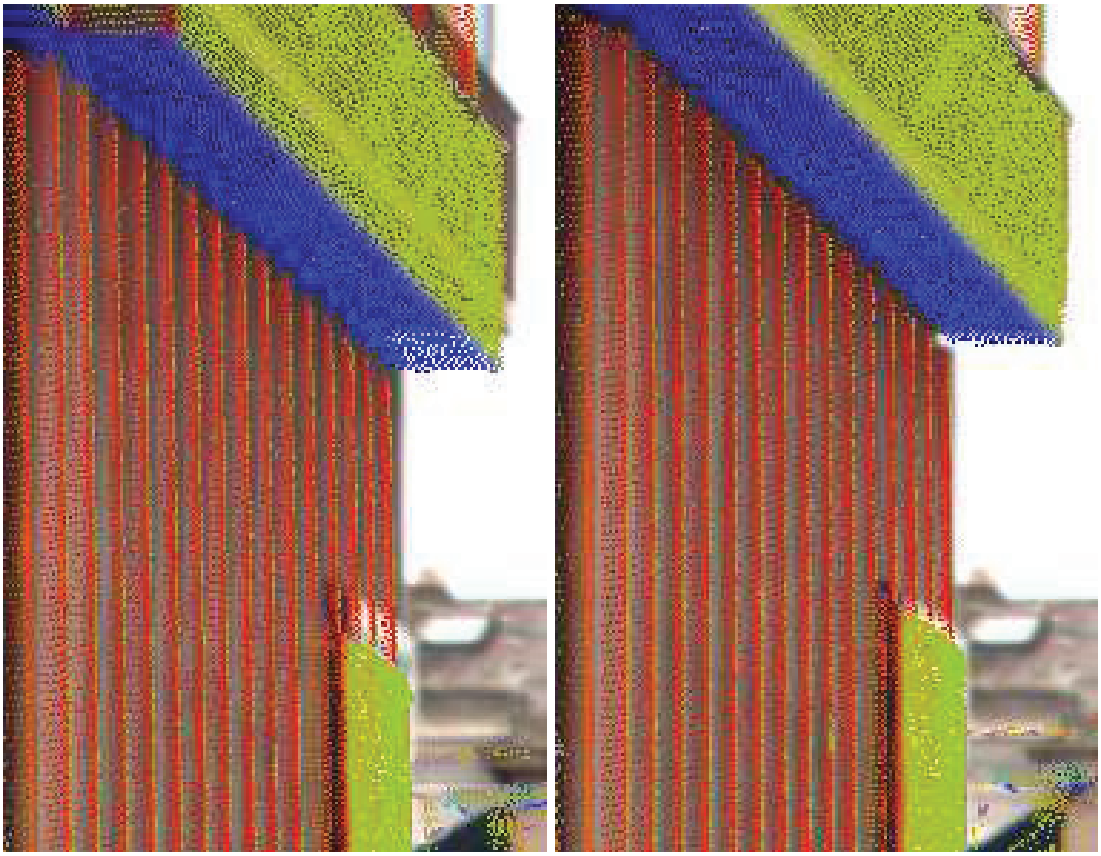
3.4.3 Coarse image to geometry registration

The panoramic image registration is an important step since it generates an excellent rotation approximation for a very small computation time. To compute DHOG, we used square cells of constant 32×32 pixels size, with 9 bins in the histograms. Blocks were composed of 4×4 cells and their ℓ^2-norm was used as the normalization factor. For this coarse registration step, we found that scales $1/4$ and $1/2$ are enough to get a good registration approximation. For MIDHOG minimization, we used an x and y step starting at $1/20$ of the image size in pixels at the finest scale. For the roll θ, since the initial error is in general much lower (less than 1°), a small step of 0.6° is used.

Figures 3.7 and 3.8 present registration results obtained solely using the coarse registration method. It appears clearly on all these figures that the coarse registration improves the camera

	No interpolation	Splatting	Bilinear	Ours
Accuracy (pixels)	41.89	23.25	21.49	20.05
Standard deviation	109.63	16.55	7.57	6.98
Success ratio (%)	73	82	82	89
Time (s)	40.61	263.46	58.65	56.11

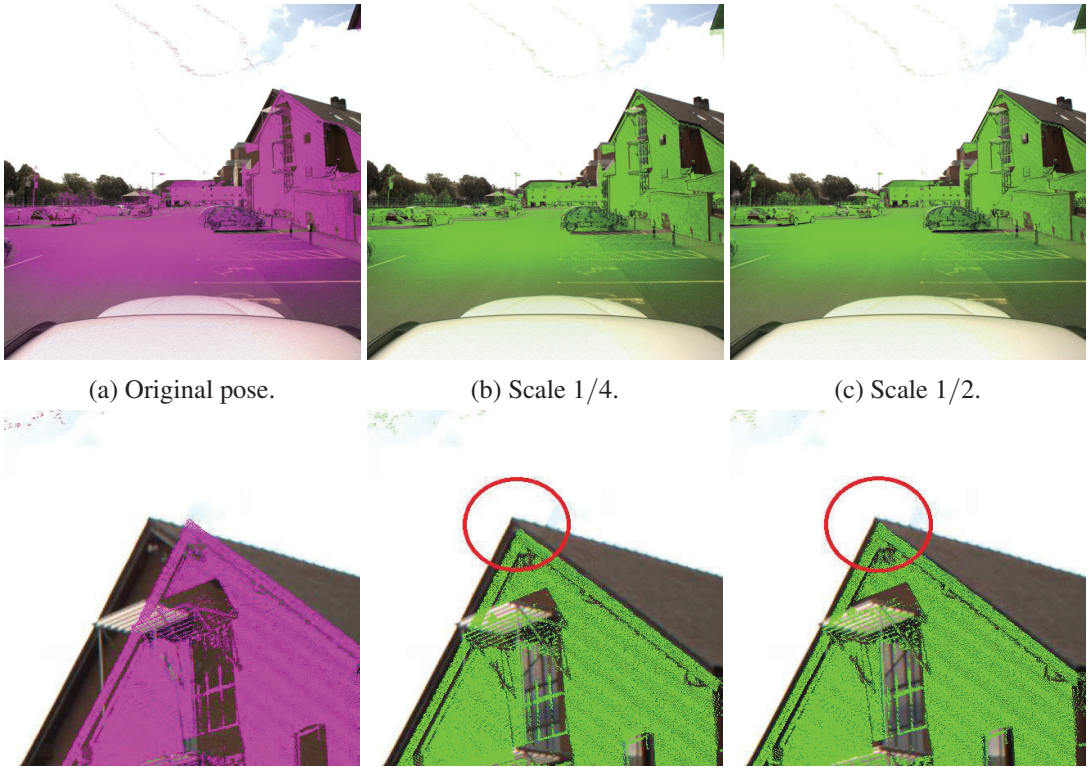
Table 3.2: Average automatic registration error in pixels observed after different kinds of interpolation. These registration results were obtained using coarse registration only from 45 images. See section 3.4 for evaluation methodology details.



(a) Bilinear interpolation.

(b) Proposed bilinear interpolation.

Figure 3.6: Details of a registration of the same region, using bilinear interpolation, or our edge preserving bilinear interpolation.



(a) Original pose. (b) Scale $1/4$. (c) Scale $1/2$.

(d) Detail (original pose). (e) Detail (scale $1/4$). (f) Detail (scale $1/2$).

Figure 3.7: Coarse Registration comparison on Council street data, with initial registration error of 2.4° (yaw) and 0.5° (pitch). Magenta color is the original registration and green color is the coarse registration results at different scales. The computation took around 60s.

pose estimation compared to the original registration. Detailed analyses are provided for each of these figures in the next paragraphs.

Council street

The first example set is acquired near the town council and offers generic city geometrical properties, with square shaped buildings, and low angle rooftops. Figure 3.7 shows the evolution of the registration through the mutiscale registration steps. Our method performs well in areas with missing building pieces or jagged skyline (Figure 3.7(a)). A different view of the same location (Figure 3.8(a)) offers different challenges such as missing rooftops, missing wall parts and occluding shadows around the foreground elements such as the fence. While the details in Figure 3.8(b) show an improvement after the coarse registration step, the registration is only roughly accurate.

(a) Coarse Registration - Council Street.

(b) Detail - Council Street.

(c) Coarse Registration - Shopping Street.

(d) Detail - Shopping Street.

(e) Coarse Registration - Castle Street.

(f) Detail - Castle Street.

Figure 3.8: Different effects of the coarse registration method using an input image taken from a lateral camera. Original registration (magenta) and coarse registration (green).

Castle street

This is a complex geometry area with high angle rooftops, two moving vehicles and some pedestrians at some distance from the acquisition vehicle. Results of the coarse registration are less accurate than on council street (Figure 3.8(b)), however, the pose estimation has clearly improved compared to the initial pose, as visible in Figure 3.8(e), and in Figure 3.8(f) for larger input errors.

Shopping street

The third example is a narrow city street environment with high buildings around the street and no access to any skyline (Figure 3.8(c)). Since the surfaces are close to the camera position, the point cloud density in the image plane in this area is low. We can observe here that the improvement provided by the coarse registration is marginal (Figure 3.8(c) and Figure 3.8(d)), mostly due to the fact that we are already close to the best possible solution. The coarse step provides thus a fast pose approximation, but there is still a residual error (Figure 3.8(e) and Figure 3.8(f)) that will be reduced in the next step.

3.4.4 Fine image to geometry registration

Although the coarse registration might look visually satisfactory (figures 3.7, 3.8(a) and 3.8(c)), a closer inspection on the details of the image/point cloud superposition reveals that the coarse registration still produces important errors. These errors are particularly visible in figures 3.8(b) and 3.8(d). In those cases the fine registration step improves drastically the registration, as shown on Figure 3.9. When comparing the results of the coarse only registration against the coarse plus fine registration, one can clearly see an improvement in the fitting of the image to the point cloud.

As explained in section 3.3, the fine scale registration step should be performed with MIDHOG to obtain a precise registration. However it comes at the cost of higher computation times. In order to alleviate this drawback it is possible to drop the NMI part of MIDHOG and thus to use solely DHOG, possibly yielding larger residual errors but faster computations. Table 3.3 compares the accuracy, computation times and successful registration ratio of both alternatives. The success ratio is obtained by considering the registration to be successful when the registration error is below a threshold (25 in our case). The remaining error is the error computed on the images considered as correctly registered. As expected the computation time is improved by using only DHOG with a gain of about 40s in average. However, when relying only on normals, the overall registration quality suffers from the unique use of DHOG, whereas using MIDHOG as the image comparison metric leads to an average registration error 4 pixels lower and a higher success rate.

All previous experiments were run by considering synthetic images created from estimated normals or reflectance. In the difficult case of normal based synthetic images, we showed that our proposed metric was able to perform the registration while other methods failed. As explained in chapter 2.2 LiDAR devices can provide an additional information: the reflectance of the laser. This information produces synthetic images that can be compared more easily with real images,

(a) Coarse.

(b) Coarse+fine.

(c) Coarse (detail).

(d) Coarse+Fine (detail).

Figure 3.9: Details of the registration on a part of Castle street, for coarse registration only, or for coarse and fine registration. The improvement of the registration with the fine method is clearly visible around the street lights.

and most methods work on this type of modality. Table 3.3 also shows that when using the reflectance information for synthetic image generation, both MIDHOG and DHOG give similar results and our method outperforms state-of-the art methods in terms of successful registration. Using the groundtruth described in section 3.3, the registration efficiency was compared using NMI with either simple geometrical information or reflectance values and the same comparison

	Error	Std	Time	Ratio	Remaining Error
Original disruption	123.57	40.82	N/A	N/A	N/A
NMI (normals)	448.94	633.3	302s	31%	20.05
NMI (reflectance)	24.25	77.23	585s	91%	8.56
DHOG (normals)	20.74	20.35	498s	89%	15.34
DHOG (reflectance)	15.25	21.05	572s	98%	12.28
MIDHOG (normals)	16.77	10.77	541s	93%	14.85
MIDHOG (reflectance)	15.25	21.05	597s	98%	12.28

Table 3.3: Comparison of the average error in pixels, standard deviation, convergence time, successful registration ratio and remaining error on the successful registration case for 45 images using either NMI, DHOG or MIDHOG as image comparison metric. The remaining error is the error computed on the images considered as correctly registered. Lines containing (reflectance) mark are based on the reflectance values rather than on the normal value for the metric calculation. Using the reflectance values leads to a major improvement in the results quality.

was done using MIDHOG. As can be seen in table 3.3 and in figure 3.10, using NMI without the reflectance values had a huge impact on the final registration. MASTIN, KEPNER, and FISHER [MKF09] observed that for aerial scans and photo, the use of reflectance only marginally improved the registration. However in our case the reflectance values, if available, improve greatly the final registration. This statement is also true but less spectacular when using MIDHOG, in which case the reflectance values improve the final registration only slightly. The NMI registration based on the normals fails to properly register the image and the point cloud. On the other hand, MIDHOG and NMI based on the reflectance give similar and satisfactory results. MIDHOG based solely on the estimated normals also gives satisfactory results, but is slightly less accurate than using the reflectance.

Table 3.4 gives the average errors on the groundtruth dataset for a complete MIDHOG registration using either the normals or using the laser reflectance. These results were computed with randomly generated errors, different than the ones present in table 3.3, which explain the slightly different residual error values.

We also applied our registration method to the KITTI dataset [Gei+13]. This dataset was obtained using a Velodyn LiDAR and co-registered camera. However, the Velodyn LiDAR outputs a point cloud that covers only a fraction of the space around the moving vehicle (see Figure 3.11). When projecting the point cloud on the image plane, the captured geometry covers

	$x(m)$	$y(m)$	$z(m)$	$\omega(^\circ)$	$\phi(^\circ)$	$\kappa(^\circ)$	pixels
Normals	0.051	0.061	0.058	0.516	0.745	0.458	16.6
Reflectance	0.056	0.060	0.058	0.344	0.401	0.401	9.74
KITTI	0.082	0.055	0.057	1.031	0.115	0.458	14.29

Table 3.4: Average registration error of our two-step method compared to the ground truth for 45 random images and for the KITTI dataset. x, y, z: translation, ω, ϕ, κ: Euler angles; and error measured in the image (in pixels). KITTI is the result on the KITTI dataset (normals based).

only roughly half the image, and points located too far from the Lidar are also not acquired at all. This is a huge difference with our data where point clouds have neither height nor depth limit, since the scans were previously merged and consolidated. To assess the registration quality of our approach, a process similar to the one described in the beginning of section 3.4 was applied to the KITTI data. Considering one image/scan pair at a time, a random error between -5° and $+5^\circ$ was applied in pitch and yaw, and a random error between $-0.1m$ and $+0.1m$ was added to the position. Our two step registration was applied on drive set number 71. It appears that due to the nature of the data, the registration of a single image/scan pair does not give satisfactory results. In this particular case, the method proposed by [TN13] appears to work better, and is applicable due the low size of image and the low size of the point cloud. However, if we consider several image/scan pairs at once, in a similar way to [Pan+12] to compute MIDHOG, our method successfully registers the images to the scans as shown in table 3.4 (last row).

The two-step registration method exhibits several advantages compared to a direct 6 degrees of freedom registration. One of these advantages is its resilience to important rotations. Indeed, as can be seen in table 3.5 , applying directly a 6 degrees of freedom pose estimation in a non optimal environment yields far larger errors. This can be explained by the sparse nature of the images, errors in the point cloud and missing data. Clearly, the two-step registration outperforms a single step registration. As visible in tables 3.6 and 3.7, our method can handle rather large input errors with acceptable accuracy results. However errors above 15° tend to be too high to be reliably overcome.

3.4.5 Comparisons

We compared our approach with two recent works for registering images on a point cloud. The first one is the original algorithm from TAYLOR and NIETO [TN13] based on Normalized Mutual Information and the second one is the GOM metric of TAYLOR, NIETO, and JOHNSON [TNJ13]. Comparisons were run on a subset of our real dataset, around Castle street, limited to 16 Million points due to the memory limitation of the Matlab implementation. First, the GOM method does not address point visibility problems which leads to areas blurred by inconsistent information superposition, as shown on Figure 3.12(a), influencing the algorithm convergence. This test was run with particle swarm of $0.5m$ variation in translation, a 2.5° range in yaw, pitch and roll. Figure 3.12 shows that using the GOM metric does not lead to a proper registration.The GOM metric is not robust enough to sparsity and missing parts of the synthetic images. The sparsity and

	Error (pixels)	Std	Time	Ratio	Remaining Error
Original disruption	123.57	40.82	N/A	N/A	N/A
NMI (fine only)	169.93	204.12	550s	7%	28.9
NMI (coarse + fine)	448.94	633.3	302s	31%	20.05
DHOG (fine only)	107.83	54.50	366s	11%	12.69
DHOG (coarse + fine)	20.74	20.35	498s	89%	15.34
MIDHOG (fine only)	104.78	56.99	435s	17%	12.36
MIDHOG (coarse + fine)	16.77	10.77	541s	93%	14.85

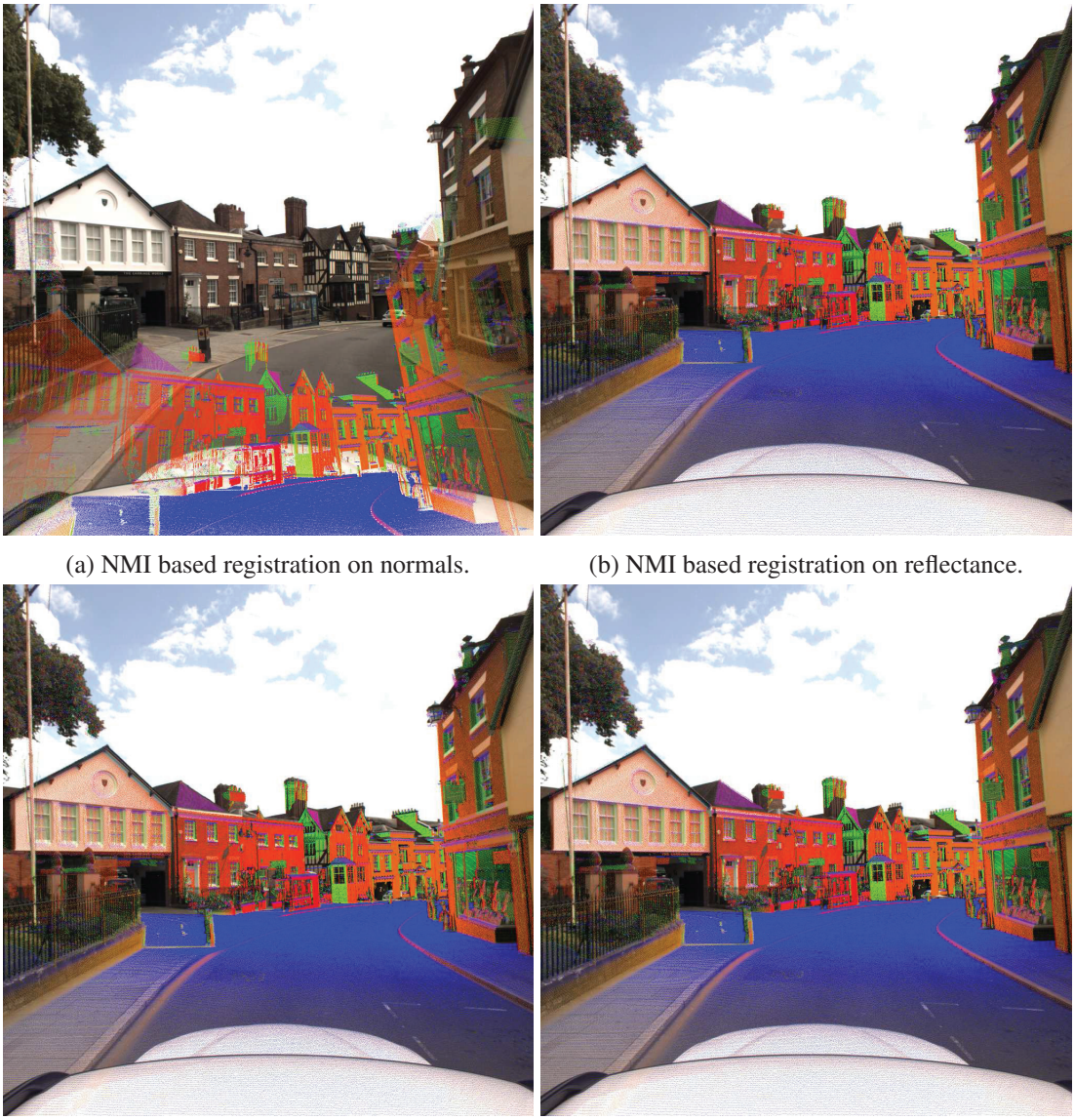
Table 3.5: Comparison of the average error, standard deviation, convergence time, successful registration ratio and remaining error for 45 images using either NMI, DHOG or MIDHOG as image comparison metric. The remaining error is the error computed on the images considered as correctly registered. The results obtained using the fine step only have largely worse results than the one obtained by the full coarse plus fine registration. All these data were obtained using the normal information of the point to compute the metric.

Angle (degrees)	3	6	9	12	14	17	20	23
Success ratio (%)	100	72	65.4	47	46.6	27.3	30	0

Table 3.6: Registration success ratio when applying a random disruption around a random rotation axis for 12 images. the displayed angle is the angle between the viewing direction of the original and disturbed camera, therefore even a rather small angle can represent important pitch, yaw and roll disruption.

Pitch Yaw	5°	10°	15°	20°
5°	100%	66%	50%	25%
10°	83%	66%	50%	33%
15°	66%	58%	33%	25%
20°	41%	16%	16%	16%

Table 3.7: Registration success ratio when applying a yaw and pitch disruption for 12 images. Ratio obtained for original disruptions lesser than 10° are quite acceptable, however, original disruptions superior to 15° dramatically decrease the registration quality.



(a) NMI based registration on normals.

(b) NMI based registration on reflectance.

(c) MIDHOG based registration on normals.

(d) MIDHOG based registration on reflectance.

Figure 3.10: Different metrics used for the registration based either on the normals or on the reflectance values of the point cloud.

the missing parts of the generated image disturbed the metric too much and actually prevented the registration. A similar test was run using the NMI metric (Figure 3.12(b)) and, once again, we can observe a failure to properly register the image on the point cloud.

Other tests were run on different scenes, applying the same search range that was comprised between $-0.1m$ and $+0.1m$ for translations and between $-5°$ and $+5°$ for rotations. This is illustrated by Figure 3.13 where the original image appears in magenta and the point cloud

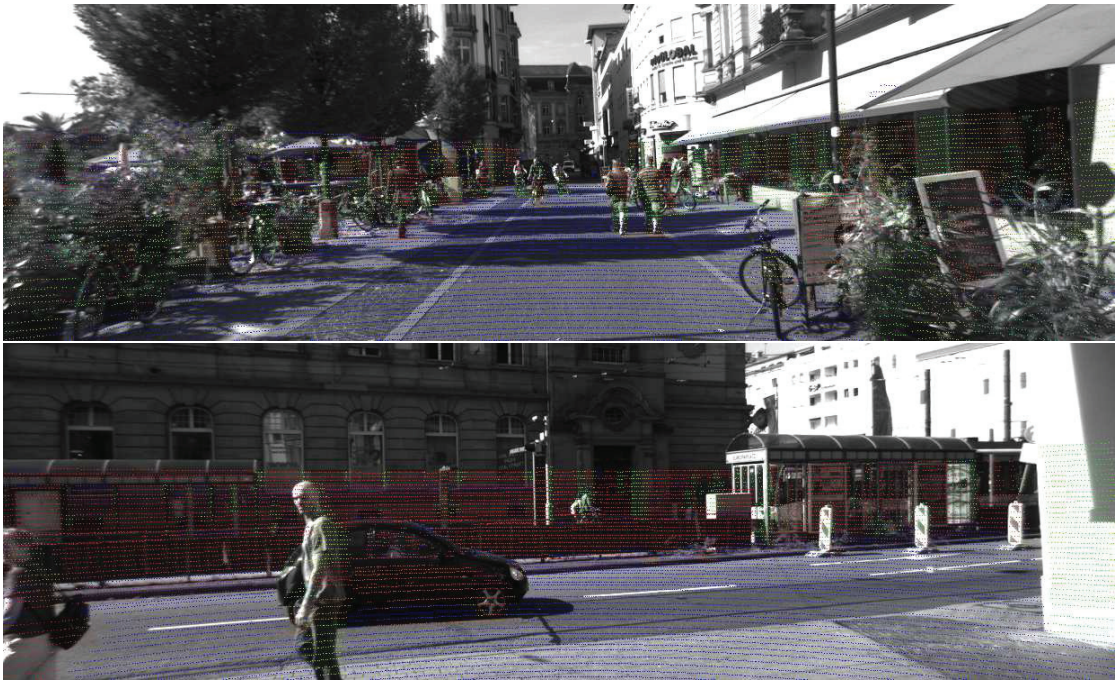
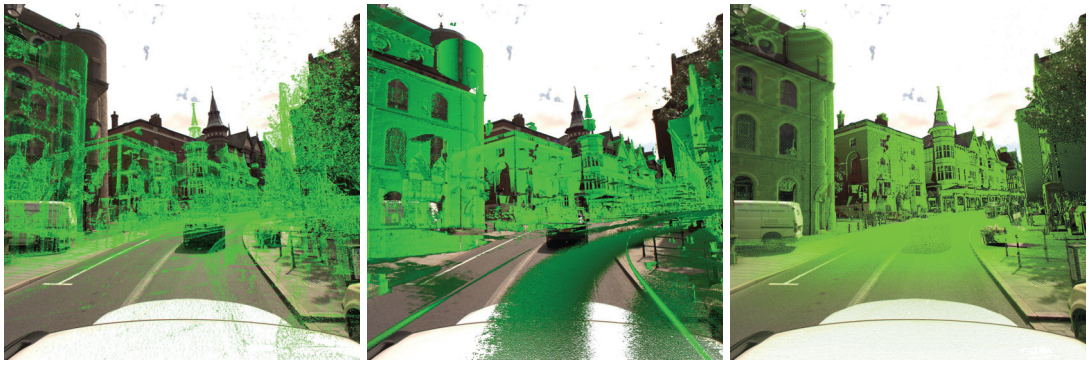


Figure 3.11: Projection of the point cloud on two of the images of the Kitti dataset containing image/scan pairs. The Velodyne LiDAR point cloud is sparse, and its scanning height is limited, which only offers a small amount of corresponding data.

projection appears in green. Results show once again that both NMI and GOM fail to properly register the image. Tests based solely on NMI failed to register properly (figures 3.13(a) and 3.13(f)), even when using the reflectance data (figures 3.13(b) and 3.13(g)). Tests using GOM metric also failed to register properly the image (figures 3.13(c), 3.13(h) and 3.13(i), except when using the reflectance value (Figure 3.13(d)), which yields an acceptable result, whereas our algorithm yields a good results in all cases (Figure 3.13(e) and 3.13(j)).

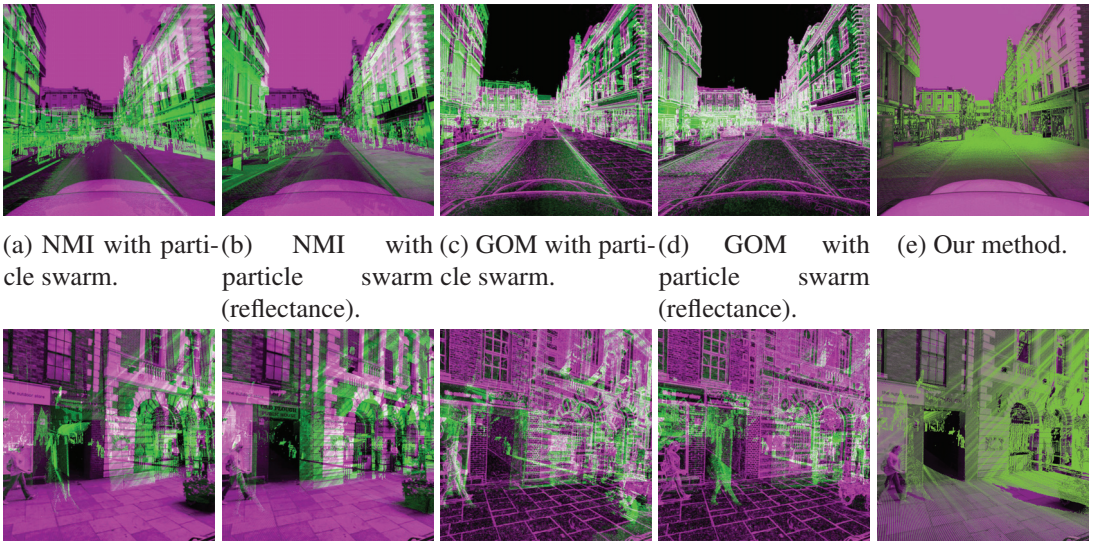


(a) GOM method [TNJ13] (40 minutes).

(b) NMI method [TN13] (62 minutes).

(c) Our two-step method (10 minutes).

Figure 3.12: Different registration results using various techniques on a subset of the point cloud. Our method clearly leads to a good registration whereas other methods fail. The high amount of noise and artifacts, characteristic of complex urban scenes, coupled with the lack of occlusion may be the origin of this registration failure.



(a) NMI with particle swarm.

(b) NMI with particle swarm (reflectance).

(c) GOM with particle swarm.

(d) GOM with particle swarm (reflectance).

(e) Our method.

(f) NMI with particle swarm.

(g) NMI with particle swarm (reflectance).

(h) GOM with particle swarm.

(i) GOM with particle swarm (reflectance).

(j) Our method.

Figure 3.13: Different metrics from Taylor TAYLOR and NIETO; TAYLOR, NIETO, and JOHNSON [TN13]; [TNJ13] used with a particle swarm optimization. Registration with NMI are clearly misaligned. GOM based on the estimated normals also lead to wrong registration. However using the reflectance values combined with GOM clearly gives acceptable results in one case (3.13(d)) whereas our algorithm yields a good results in all cases.

CHAPTER 4

POINT CLOUD COLORIZATION AND SHADOW REMOVAL

Contents

3D point cloud data is not necessarily restricted to the geometric coordinates. Additional information can be added to each point such as color, texture or material property. These additional information can be useful in many domains. For instance, the color information can be used for various improvements in geometry processing. ZHAN, LIANG, and XIAO [ZLX09], for example, propose to segment a point cloud by taking advantage of an additional color. This information can also be used for registration purpose, by modifying the Iterative Closest Point (ICP) algorithm into a color ICP [KHP]. It can also be used to ensure data geometric consistency over time using color [Kus+14]. Conservation of cultural heritage sites and objects is also a field where the model visual representation is overwhelmingly important. In these fields, the color and the aspect are not used to improve a process but rather to capture the visual aspect of the object as realistically as possible. Cultural heritage digitization and conservation is a widely studied area as the following publications suggest: [Ler+10], [Mou14],[AH04],[Bas+14], [Rem11].

4.1 Points colorization from multiple images

While some LiDAR devices can directly associate color information for each acquired point, the remaining ones produce uncolored point clouds. If no color is associated at all with the

point cloud, then it needs to be colorized from an external source. Even the point clouds that are automatically colored by the LiDAR device itself unfortunately suffer from a number of colorization defects. These artifacts (*e.g.* Figure 4.1) can impair the processing by color-based algorithm or deteriorate the visual aspect of interesting features for the user. Besides, the given color might be of low resolution, leading to a blend in the textures details that might be of importance for an observer.

(a) Original colorization. (b) Correct colorization.

Figure 4.1: Comparison between the original cloud colorization and the presented colorization result. Original color have an important stain located on the sidewalk. This problem is corrected with the proposed colorization process.

Considering the particular urban data case (see chapter 1), the objective is to obtain a sharp, accurate and correct color for each point of a massive urban point cloud, using multiple overlapping images. These overlapping images, despite being registered closely to the underlying geometry (see chapter 3), can still be afflicted by residual calibration and registration errors that will lead to incorrect color assignment. Moreover live outdoor urban scenes also contain several inconsistencies between the images themselves, such as partial occlusions created both by static and moving objects, and illumination variations as illustrated by Figure 4.2.

In the following section, we will first briefly review the existing methods to colorize a point cloud using images. Then, an in-depth explanation of a method capable of colorizing large urban point clouds using multiple images will be detailed, followed by a presentation of the obtained results, including a comparison to other colorization methods.

(a) Photo at a time t.

(b) Photo at a time t+0.33s.

Figure 4.2: Two consecutive frames taken 0.33s from each other. We can see that both the car in the center of the image and the pedestrian on the left moved slightly. The illumination conditions also changed and produced a difference in the perceived colors, particularly visible on the top left of the image.

4.1.1 State of the art

Point cloud colorization methods usually focus on the use of a single image, such as in the work of CROMBEZ, CARON, and MOUADDIB [CCM13], where a correctly registered image is used to colorize the point cloud. They use a bilinear interpolation of the 4 closest pixels in the *RGB* colorspace. However this method does not cover the use of multiple images to colorize complex point clouds such as the Shrewsbury dataset, as presented in section 1.2.4.

When multiple images are available, naive methods are usually used to perform the colorization, such as using the average or the median color for each point. These classical approaches typically do not yield good results, as discussed in section 4.1.3. Another possibility is to cast the problem to a single image colorization scheme to obtain results. These methods usually consist in taking the image whose point of view is the closest to the considered point (as in [ZZ14]). Another commonly used method is to consider the image for which the view direction is the most similar to the estimated point normal.

Another approach discussed by ABDELHAFIZ [Abd09] is called "*Point Cloud Painter*" algorithm. This algorithm proposes to get rid of the occlusion problem by pairing images with the closest *RGB* value difference. Then the average value of the pair with the smallest difference is used to colorize each point. This approach suffers from some important drawbacks. This approach will fail if multiple images of the same scene are taken close to the same position, or during a very short time lapse. Indeed, in this case, pairs of images could be chosen to colorize a

point even if they both represent an occluding object (such as the car in Figure 4.2).

LIM [Lim11] proposes to overcome the occlusion problem appearing in standard colorization process by de-occluding an image sequence and obtaining a consensus image to perform the colorization process. This method seems to be effective at correcting the differences between LiDAR data and image data, even when a high number of occlusions is present. However it seems more adapted to images taken with a static point of view, and might be difficult to use in a multiple point of view and multiple timeline image sequence.

Finally, in their work CHO et al. [Cho+14] propose a colorization that seems particularly adapted to urban environment containing both a great number of images and points. Their method consists in creating a neighboring graph from an unordered point cloud. From this newly ordered point cloud, the most correct color is attributed to each point by minimizing an energy based on the average color given by all images and the color differences with the neighboring points. This method is the basis of our own colorization process that will be explained in further details in section 4.1.2.

4.1.2 Colorization by global optimization

The colorization method described here aims at giving an accurate color for each individual point of a massive urban point cloud, using multiple overlapping images.

Neighborhood graph creation

The process as it was described by CHO et al. [Cho+14] must first create a neighborhood graph from the unordered point cloud. Several methods are possible to do so. In their paper, CHO et al. [Cho+14] compare the use of a Delaunay triangulation graph, with a KNN graph and a multiple Z-Ordering neighboring graph. According to their statement, the KNN based graph can produce isolated point clusters which is harmful for the final colorization quality. The Delaunay triangulation on the other hand gives good results without isolated clusters but suffers from a much longer processing time. The multiple Z-Ordering gives results similar to the one observable with the Delaunay triangulation but with a faster processing rate. The observation we conducted on our data (Figure 4.3) are consistent with this statement comparing the Delaunay triangulation and the Z-Ordering. As stated by CHO et al., the use of a Z-ordering neighborhood or a Delaunay triangulation based neighborhood yield similar results, as visible in Figure 4.3(a) and Figure 4.3(b). However, using the 5 nearest neighbors leads to bad results as shown in Figure 4.3(c). This kind of behavior was expected as the number of neighbors is low and the sampling of the point cloud is not uniform. This observation led us to use this Z-ordering graph for the neighborhood definition.

The Z-ordering implementation consists in determining an identifier (or a key) for each point. Considering a point P with 3D coordinates X, Y, Z, we define its key K_P by interleaving the binary values of its different coordinates. If we consider that each coordinate is defined by n bits, we can compute a key of $3n$ bits as:

(a) Z-Order result. (b) Delaunay result. (c) 5NN result.

Figure 4.3: Influence of the neighborhood variation on the colorization result. While Delaunay based and Z-ordering based neighborhood yield similar results, 5*NN* result displays some unexpected color stains. Data were interpolated for visualization purpose.

$$K_P = b_X^n b_Y^n b_Z^n \dots b_X^0 b_Y^0 b_Z^0 \tag{4.1}$$

Using these keys, we are able to define a neighborhood graph by connecting the points considered in an ascending order. This way, we obtain a simple graph where each point is linked to at least one and at most two other points. However, having at most two nodes for all the points may lead to imprecision for neighbor-based coloring. To overcome this limitation, CHO et al. introduce a multiple pass Z-ordering. This multiple pass consists in applying a Z ordering graph creation with varying points coordinates, namely:

(1) (X,Y,Z)

(2) $(-X,Y,Z)$

(3) $(X,-Y,Z)$

(4) $(X,Y,-Z)$

Combining the different neighboring graphs obtained with different Z ordering offers a much higher number of neighbors for each point. This strengthens the graph and largely improves the colorization quality, but also lowers the optimization of the structure as discussed in the computation paragraph.

Color estimation

Once a consistent neighborhood graph is obtained for the considered point cloud, the next step is to associate all the possible colors to one point. These possible colors correspond to the color that would be associated to the point in each different image. This possible color assignation is

performed using the standard point cloud projection as defined in chapter 2. Instead of using the *RGB* colorspace, as in the original paper, we switched to a more adapted colorspace: the $La*b*$. This colorspace has the advantage to be perceptually uniform, and particularly adapted to color difference computation.

Once these colors have been assigned to each point, we can define the energy to optimize. This energy is similar to the one introduced by CHO et al. [Cho+14] and defined as:

$$E(C) = \sum_{p \in P} D_p(C) + \lambda \sum_{(p,q) \in N} S_{p,q}(C). \tag{4.2}$$

where:

- C is the color vector (containing the L, a and b values) associated with the point P.
- D_P is the data term for the point P.
- $S_{P,Q}$ is the smoothness associated to the points P and Q belonging to the neighborhood N.
- λ is the weight of the smoothing term that will be kept to 1.0 throughout the method, as suggested by the original authors.

While this energy is similar to the one presented in [Cho+14], the data and smoothness term are slightly different.

Data term

The data term encourages the final color to be as close as possible to the median color of all the potential colors associated with the point p.

$$D_P(C) = \alpha_P ||c_P - Md(v_P)||^2 \tag{4.3}$$

where:

- c_P is the color that will be assigned to P.
- $Md(P)$ is the median color of all the possible colors for the point P.
- α_P is a binary term that indicates if the data-term will be used or if we will rely solely on the smoothness term to attribute the color to the point P. Following the recommendations of CHO et al. [Cho+14], this term was randomly set to 1 for 10% of the points, and to 0 for the rest.

As stated previously, the use of the average color in [Cho+14] was replaced here by the median color. This median color is not defined exactly as a geometric median, but as the color of the set that minimizes its distance with all the other colors of the set.

Smoothness term

Similarly to the *data term*, the *smoothness term*, that is the difference of color gradients, is edited to consider the median of the color differences instead of the average of the color differences:

$$S_{P,Q}(C) = ||(c_P - c_Q) - g(P,Q)||^2 \tag{4.4}$$

with $g(P,Q)$ being the median of the color differences such as:

$$g(P,Q) = Md(v_P^i - v_Q^i) \; \forall i \in U_P \cap U_Q \tag{4.5}$$

where U_p is the set of images within a threshold distance between points of view. This threshold distance is defined here as taking 20 % of the closest images for each point. This value yet arbitrary is important as it is one of the key to keep an accurate color while removing the occlusion effects.

Computation

The energy defined by equation 4.2 can be represented as a linear system of the form $Ax = b$ as follows:

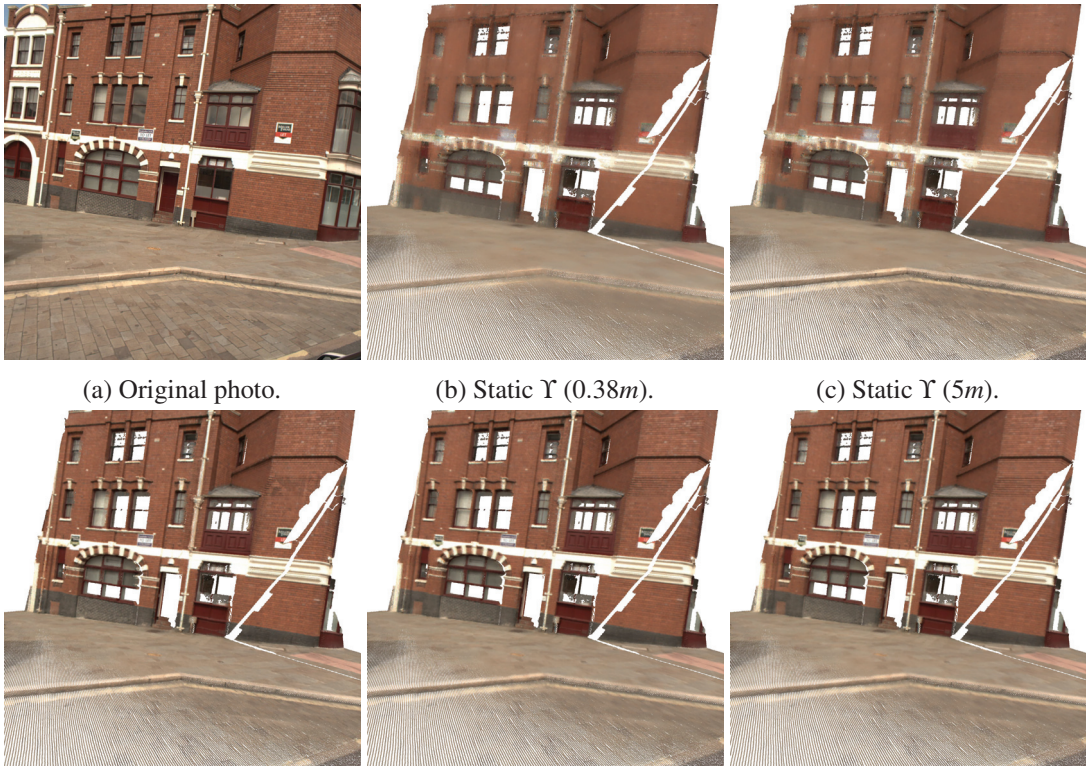
$$c_P(a_P + \sum_N (1)) - \sum_N c_Q = a_P Md(v_P) + \sum_N g_{(P,Q)} \tag{4.6}$$

with A being a gigantic sparse matrix of size $3m$, with m the number of points in the point cloud. According to CHO et al., the matrix A by being tri-diagonal is optimized for solving the energy optimization. However, this is not the case when we consider multiple possible neighbors for one single point (*i.e.* using multiple pass Z-ordering). In this case we obtain a sparse but not tri-diagonal matrix. According to the sparse properties of the matrix A, SuiteSparseQR [Dav11] is used to solve this sparse linear system efficiently.

4.1.3 Results and discussion

Parameters study

The original method uses a threshold parameter Υ to tune the final color given to a point. This parameter has a strong impact on the final results quality. However in their original paper CHO et al. do not explicitly indicate the best distance parametrization to use with their method. We therefore conducted a comparative study of different parameters to determine which was the most suited for the urban point cloud colorization. As we do not have a method to quantitatively rate the quality of the colorization, comparisons were made for identical rendering compared to a reference photography. Different threshold distances were applied to the colorization process to try to determine the most suitable one. As shown in the comparison on Figure 4.4, a static threshold leads to artifacts. Indeed, a too small threshold value will lead to a blurry colorization (Figure 4.4(b)). A threshold that would be correct for some point distance, will still be blurry for further points, and will also be blurry for closer points (Figure 4.4(c)). The use of a dynamic



(a) Original photo. (b) Static Υ (0.38m). (c) Static Υ (5m).

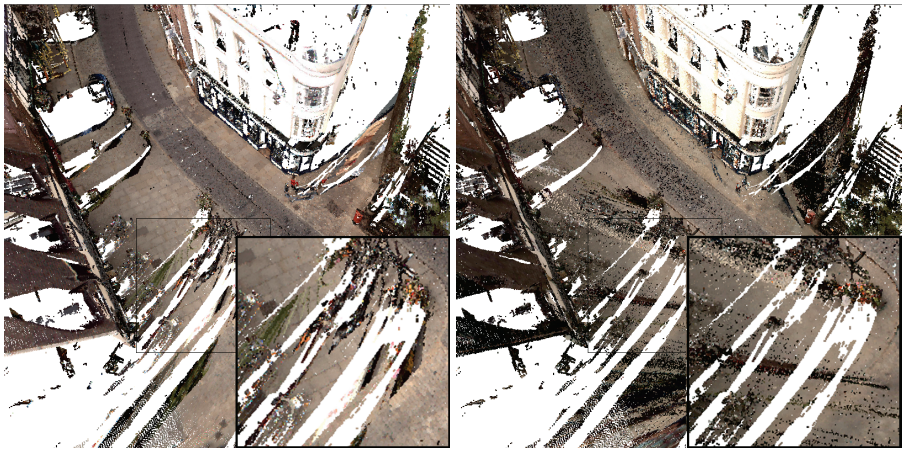
(d) Dynamic Υ (20%). (e) Dynamic Υ (50%). (f) Dynamic Υ (75%).

Figure 4.4: Different colorization results depending on the distance threshold parameter Υ. Static threshold is the limit distance between the point and an image. Dynamic threshold is the limit distance in order to take into account a certain percentage of images.

threshold distance that only allows a certain percentage of images to be used is preferable and yields better results. However that percentage of images is still subject to possible variations, and if a higher percentage tends to somewhat enhance the results details (bricks), it degrades the results for some others (publicity panels, ground markings). In our tests, we defined this threshold distance Υ as the distance needed to use 20% of the available images. This threshold value produces good results, and is the common threshold value used to obtain all the results presented in this section.

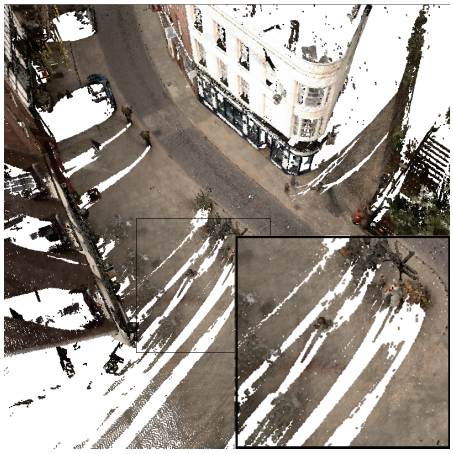
Comparison

Various locations were selected to perform the colorization tests. They vary both in content, in size of the point cloud and in number of images. As illustrated in Figure 4.5, it is difficult to assess the quality of the colorization of the point cloud from a distance. However, if the rendering camera is set too close to the point cloud, visual information is too sparse, even when an interpolation is applied on the data as illustrated by Figure 4.10.

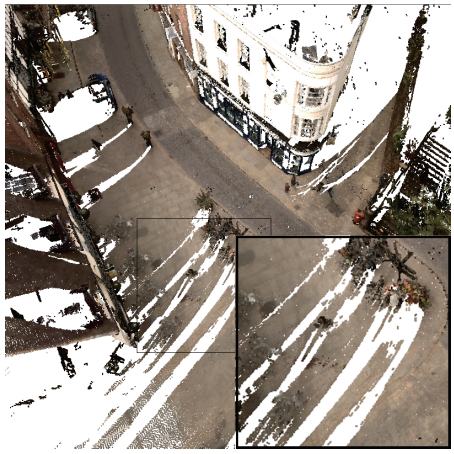


(a) Original point cloud color.

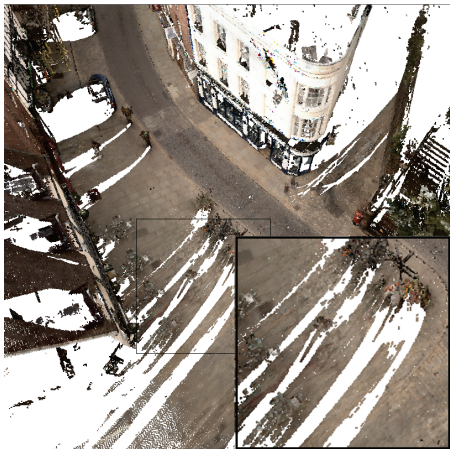
(b) Closest image use.



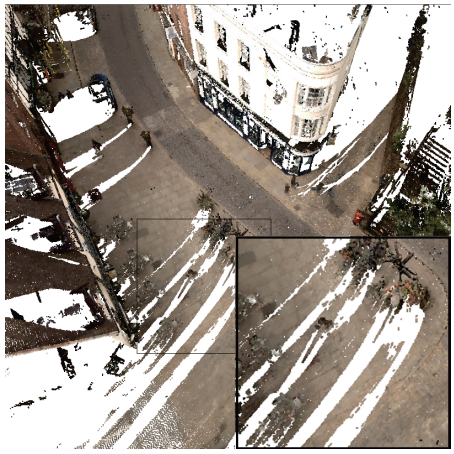
(c) Average color.



(d) Median color.



(e) Color from CHO et al.



(f) Our color.

Figure 4.5: Top-down view of the crossroad sub-set cloud colored with different methods. This subset is composed of 9 millions of points colored by 360 images.

Comparison with naive colorization methods

Both Figure 4.5(a) and Figure 4.5(b) contain artifacts associating wrong colors to points. These artifacts are most notably visible on the ground in Figure 4.5(a) with the projection of flower pots and pedestrian on the sidewalk. A lot of artifacts were added on the ground level in Figure 4.5(b) due to static and mobile occluding objects such as flower pots or cars.

It is unfortunately almost impossible to differentiate the four remaining figures (Figure 4.5(c), Figure 4.5(d), Figure 4.5(e), Figure 4.5(f)) from afar. However, Figure 4.6, which represents a small area of the same scene as Figure 4.5, we can see with more details the effect of our method compared to naive approaches. In this image (Figure 4.6) it appears clearly that both the average color and the median color produce a low resolution colorization. As a consequence, a strong blur appears in Figure 4.6(a) and Figure 4.6(b) compared to Figure 4.6(c). These blurry effects are most likely due to residual calibration or pose estimation problems. If the images are even slightly off the geometry they represent, the addition of color assignation errors will result in a blurry and unclear colorization. This blur effect is noticeable almost everywhere in Figure 4.6, but reduced in the proposed colorization scheme (Figure 4.6(c)). The color sharpness is also conserved by the presented technique compared to naive approaches. This is particularly visible with the red postal bin. These effects also appear clearly in Figure 4.7 where the ground markings appear less precisely in both Figure 4.7(a) and Figure 4.7(b) compared to Figure 4.7(c) and Figure 4.7(d).

Comparison with the original method from CHO et al.

As stated in section 4.1.2, the original colorization method from CHO et al. uses the average color and the average difference of color in its energy minimization. Using such values in urban environments, which contain occluding objects, may lead to the apparition of several artifacts. This is visible in Figure 4.8(b) and Figure 4.8(d) that respectively use the naive average color and the method of CHO et al. On these two figures, the traffic light pole color is projected onto the ground and the wall, mixed with the proper color of the surface. This creates darker stains that should not have the color assigned to the points. A slightly darker, thin form appears near the base of the pole in Figure 4.8(e). It looks like the kind of artifact visible also in Figure 4.8(d), however it is in fact the shadow of the traffic light, that appears in the original image (Figure 4.8(a)). The same problem leads to more diffuse errors in Figure 4.9. In this case the presence of trees introduces a green component by assigning the green color through the leaves to the red bricks.

The color artifacts visible in figures 4.8(d) and 4.9(a) with bright color spots, are the result of the optimization producing an overflow of the *RGB* color values. This could have been prevented on our side by forcing the values to stay in the defined range of the *RGB* colorspace. However this effect does not appear when using the $La*b*$ colorspace (Figure 4.8(e) and 4.9(b)).

When compared to the original color given to the point cloud (Figure 4.10), one can observe that the ground and walls artifacts are removed, the color is unified, the missing colors are recovered and the overall look and assignation of the colors is more consistent with the reality.

(a) Average color.

(b) Median color.

(c) Our color.

Figure 4.6: Top-down view of a small colored part of the crossroad subset. The use of a global optimization method sharpens the colors compared to the simple use of median or average color. This is particularly visible with the red postal bin, on the bricks wall and on the signs.

(a) Average color.

(b) Median color.

(c) Color from CHO et al.

(d) Our color.

Figure 4.7: Top view of the colored point cloud in Castle street near the train station. Average and median color are blurry compared to other colorization methods. Stains of the traffic light pole are visible on the top left part of Figure 4.7(a) and Figure 4.7(c).

Limitations

Although our method improves the result compared to a naive colorization and to the method of CHO et al., some problems persist. For instance, on Figure 4.8(e) the train station in the far background on the right of the image suffers from a color mis-alignement. Indeed, the sky appears on the right area of the building in place of stone masonry. This kind of artifact is likely due to a

(a) Original image. (b) Average color. (c) Median color.

(d) Color from CHO et al. (e) Our color.

Figure 4.8: Views of the colored point cloud in Castle street near the train station. An interpolation has been performed to improve the visibility. In Figure 4.8(d) the traffic light pole color is projected onto the ground and the wall, mixed with the proper color of the surface, whereas it is not present in Figure 4.8(e).

residual pose or calibration error in the camera parameters. However, these kinds of artifacts are likely to affect only distant structures, where the registration errors have more impact.

Another noticeable problem is a small definition of the ground details when compared to the wall details. This is visible in Figure 4.6 where the ground tiles, while being distinguishable, are not as sharply defined as the wall stones. This is likely due to the actual image acquisition angle (cf. chapter 1 - Figure 1.8). Indeed, the color information obtained for the ground will always be from a small angle, giving less information for each point compared to points on a wall that can be directly targeted by the cameras. This also produces a slightly diminished accuracy of the colorization of the ground (Figure 4.10(a)) compared to the original cloud color (Figure 4.10(b)).

(a) Color from CHO et al. (b) Our color.

Figure 4.9: Image view of the Bell Tower. Even without the presence of artifacts, the colorization of the tower for the left image is slightly greenish due to the contribution of the trees in the colorization of the bricks.

(a) Original cloud colorization. (b) Our color.

Figure 4.10: Image view of the square around the Statue of the Major-General Robert Clive. Original colorization has several artifacts around static occluding objects that were successfully removed. Other artifacts on building walls were also removed by our recolorization.

Finally, inconsistencies between the images and the point cloud appear with moving objects. Those moving objects silhouettes are colorized according to the background in the proposed colorization process. It means that pedestrians who are not static in the scene will surely appear to be made of bricks or other masonry materials (Figure 4.10). Unfortunately, there is no solution to this problem in the colorization process, and this kind of mobile objects should be removed from the geometry prior to any colorization by a detection algorithm, that is beyond the scope of this thesis.

4.2 Cast shadows removal from colored point cloud

The application of good quality colors, as explained previously, does not take into account exterior factors influencing the color. Indeed, factors such as the local weather, the time of the acquisition (*e.g.* dawn, noon or dusk) and transient geometry (*e.g.* pedestrian) can affect the color given to a point. Cast shadows are, in conjunction with light chrominance, the most noticeable color variations and should be corrected to not impact the color of the point cloud. In the following section, we will focus on automatic detection of cast shadows and on their removal by luminance and chrominance correction. This problem is closely related to intrinsic image decomposition [Bon+17], which consist in separating color from illumination in images.

4.2.1 State of the art

Shadow detection and removal is already a widely explored field. Methods have been developed to detect shadows either on a single image, a sequence of images or datasets containing both geometric information such as a LiDAR point set and registered images. Each of these problem settings requires a specific problem formulation, although some common ideas can be found in all these approaches. We review below relevant works emphasizing concepts that are linked to our approach.

Single image shadow detection

The detection and correction of shadows in a single image is a difficult problem that has been extensively studied. We restrict our analysis to some recent works that not only propose methods to detect the shadows but also address the re-illumination problem.

LALONDE, EFROS, and NARASIMHAN [LEN10] present a method to locate and remove the shadows based on machine learning. First the shadows are detected using a watershed segmentation and a Canny edge detection. Descriptors composed of histograms and skewness of the pixels intensities are analyzed on both sides of the detected edges. Shadow edges are then consolidated using a probabilistic model. Finally the scene layout (decomposition between ground, sky, and vertical surfaces) is added to only detect and consider the shadows laying on the ground.

CORKE et al. [Cor+13a] propose a method to recover a grayscale intrinsic image from a standard RGB image by considering the physical properties of the light and using also the *Log-Chromaticity* colorspace. In this particular colorspace, the colors of all materials under varying illuminations change along parallel directions. By projecting the points on a line perpendicular to these directions, an illumination-free color can be found. It however requires user interaction to define two regions of the same material under different illumination conditions and a perfect knowledge of the camera sensor response.

XIAO et al. [Xia+13] remove shadows from a single image using a multiscale illumination transfer in the $La*b*$ colorspace. Shadowed and lighted regions are roughly sketched by the user and then segmented using a Gaussian Mixture Model. A multi-scale illumination transfer between

the shadowed and lighted regions is then performed assuming lambertian surface illumination properties. Finally the shadow boundaries are reprocessed using a Bayesian framework to remove relighting artifacts.

Images sequence and 3D proxy

Over the past ten years, fast and robust photogrammetric methods have been developed to compute a 3D model from a series of pictures (*cf.* chapter 1). These multiview methods have been exploited to enrich the information of 2D images in order to retrieve specific lighting properties.

LAFFONT, BOUSSEAU, and DRETTAKIS [LBD12] propose a complete and precise process to retrieve the intrinsic characteristic from a set of images. This algorithm requires a set of HDR images, a set of LDR images from which a proxy 3D model is built, and a measured environment map. User interaction is also required to define the sun orientation and obtain two gray color values in the sunlight and in the shadow. This set of input data is used to decompose the illumination of each vertex of the proxy model into the original albedo and the sun, sky and indirect luminance. Finally these vertices are projected onto the image planes to retrieve the intrinsic images. Although this method proves efficient, it requires a lot of additional information that are not available in our context, in addition to user interaction, something we avoid.

In a somewhat similar manner, WEHRWEIN, BALA, and SNAVELY [WBS15] automatically detect the shadows and the sun direction in a series of picture. Applying SfM to the pictures yields a 3D model. A set of colors is then associated to each vertex of the mesh that comes from all the images that see the vertex. From the statistics of the proposed colors, the algorithm can guess which points lie in the shadow in each image. These shadow point labels are used to consolidate the shadow edges and estimate the orientation of the sun.

Shadow detection on 3D LiDAR data

Detection of shadows on 3D LiDAR data is a less explored environment compared to single image and images collections.

TROCCOLI and ALLEN [TA05] relight a 3D point cloud model using multiple HDR overlapping images taken under different illumination conditions. Their strategy is to compute and analyze illumination ratio in overlapping areas and use these ratios to relight the whole image. This approach however requires a mesh reconstruction step to get rid of the point cloud and work on a watertight surface.

RAMAKRISHNAN, NIETO, and SCHEDING [RNS15] propose to correct the colors of a point cloud using a consistent and continuous illumination model by removing the direct sun illumination and normalizing the sky illumination. The indirect illumination from the other part of the scene is considered negligible. The sun orientation is obtained using the GPS coordinates of the scene as well as the known sun position relative to this position at the time of the scene acquisition. Similarly to [Cor+13a], pairs of points of the same material are selected by the user both in sunlit and in the shadowed parts of the cloud. These pairs are used to compute the sun illumination and sky illumination contributions which yield an estimation of the illumination

values for each point of the cloud. Unlike the presented approach, this method requires user interaction. Furthermore, this algorithm is tested on LiDAR scans measured on separated buildings, a setting quite different from complex urban scene point clouds, where the buildings are concentrated and the streets narrow. Finally, this method relies heavily on having a clean 3D model. However in the context of complex urban data, the 3D model is only imperfectly measured, which can affect the quality of the sun visibility estimation (there can be missing building parts) and alter the final results.

4.2.2 Shadow detection

The presented shadow detection method relies on the observation that a shadow barely impacts the reflectance value of a point while it strongly impacts the color value. Indeed the laser reflectance only depends on the material properties of the surface whereas the color takes into account the illumination of a point. Therefore the first step is to look for shadow interface areas that have low reflectance gradients and high color gradients.

Shadow interface detection

In this first step, the algorithm works directly on the point cloud and finds pairs of close points that are likely to lie on each side of a shadow interface.

The firsts steps are to define the neighborhood of a point as the set of its K nearest neighbors and to estimate the color gradient ∇c and reflectance gradient ∇R in this neighborhood. Since the value of K impacts the computation time, we use a small value $K = 4$ at the risk of loosing some interfaces. As will be seen below, we still get enough interfaces to detect the shadows. The gradients between two neighboring points p and q are defined as follows:

$$\nabla R = \frac{|R_p - R_q|}{R_p + R_q}$$

$$\nabla c = E_{00}(c_p, c_q)$$

where E_{00} is a weighted difference of the color expressed in the $La*b*$ colorspace to render the gradient perceptually meaningful. The exact formula for E_{00} can be found in the work of SHARMA, WU, and DALAL [SWD05]. To detect relevant pairs of points we use a threshold that loosely selects a set of shadow interfaces. This threshold is set experimentally: two points are considered to lie at a shadow interface if $|\nabla c| \geq 2.33$ and $|\nabla R| \leq 0.05$.

It may appear counterintuitive to use the color gradient instead of the luminance gradient since the luminance should contain all illumination information. However, in the case of outdoors scenes, not only the luminance but also the chromaticity of the points is affected by the sun visibility, as noted by KHAN and REINHARD [KR05] and Corke CORKE et al. [Cor+13a]. Indeed, sunlit areas tend to be more yellowish (due to the sun light spectrum), while unlighted areas tend to appear more blueish (due to Rayleigh scattering). Therefore, it is better to use the full color difference rather than the luminance only. This assumption was verified experimentally by a higher number of meaningful detection.

Since the threshold is not tight, irrelevant detected pairs of points are filtered out based on a local density criterion: if a single pair is detected in a large area then it is likely that this pair is a false positive. In practice we remove the pairs of points that do not contain at least $1/4$ of their K' nearest neighbors as potential shadow interface ($K' = 21$). While this step does not remove all the outliers it is sufficient to reduce the computational burden of the next filtering step.

Shadow interfaces filtering

A further filtering of the shadow interfaces is achieved by analyzing the luminance histograms around a detected shadow interface pair of points. Indeed, histograms around a pair corresponding to a true shadow interface should have a bimodal distribution reflecting the light and shadow parts, as illustrated by Figure 4.11.

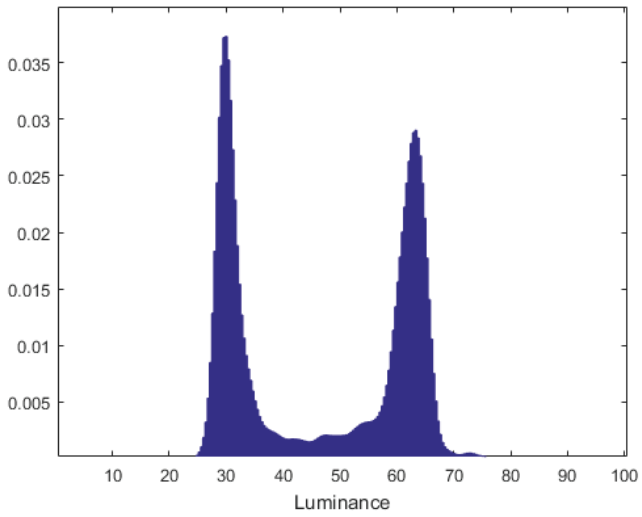
Following this observation, points pairs should only be kept if their histograms indeed display a bimodal distribution. Luminance histograms are computed by considering points that lie close to the pair and exhibit similar normal and reflectance values. This avoids considering luminance coming from potentially different materials in the vicinity of the pair. Histograms are then smoothed using a convolution and the peaks are detected by computing the first derivative of the distribution function. Small peaks are discarded if their heights are less than 75% of the highest peak. Only interface pairs that display a bimodal distribution are kept. The values of the two peaks define L_L and L_S the luminance in sunlit areas and in the shadowed areas respectively.

Shadow consolidation

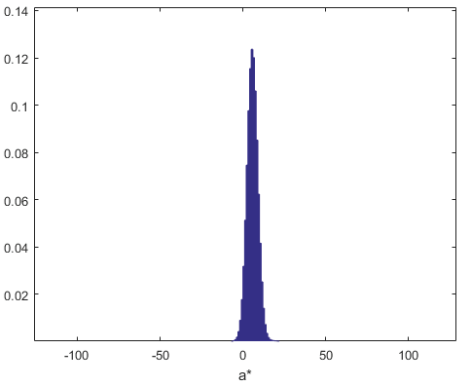
Previous steps allowed for the detection of interface points pairs which should next be turned into a complete segmentation method. To do so, we momentarily set aside the point cloud, and exploit the detected interfaces directly in the image plane, since an image is a denser information source. Thus, the shadow interface pairs are projected on the image plane, using the camera known pose and intrinsic parameters.

Using these interface points to estimate the sun position and deduce the segmentation might look like an appealing idea but it would require a complete and consistent geometry to be able to retrieve the shadow mask. However if it is indeed the case on most terrestrial scans, some scanners may either miss occluding geometry or even do not account for all the geometry of the scene. For example, the Velodyn scanner of the KITTI dataset measures points that are less than around 2 meters off the ground. We propose a different approach, relying on the images to compute a shadow segmentation: we look for a labeling δ of the image pixels, equal to 1 if the pixel lies in a shadow and 0 otherwise. The segmentation problem can be stated as an energy minimization with the following objective:

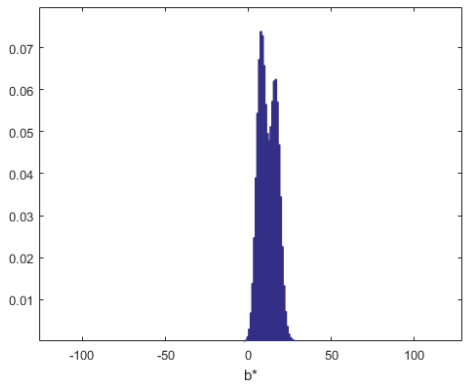
$$E(\delta) = \sum_{p:\text{image pixel}} E_{data}(\delta, p) + \gamma \sum_{(p,q):\text{ neighboring pixels}} E_{smooth}(\delta, p, q) \tag{4.7}$$



(a) Luminance histogram.



(b) a* histogram.



(c) b* histogram.

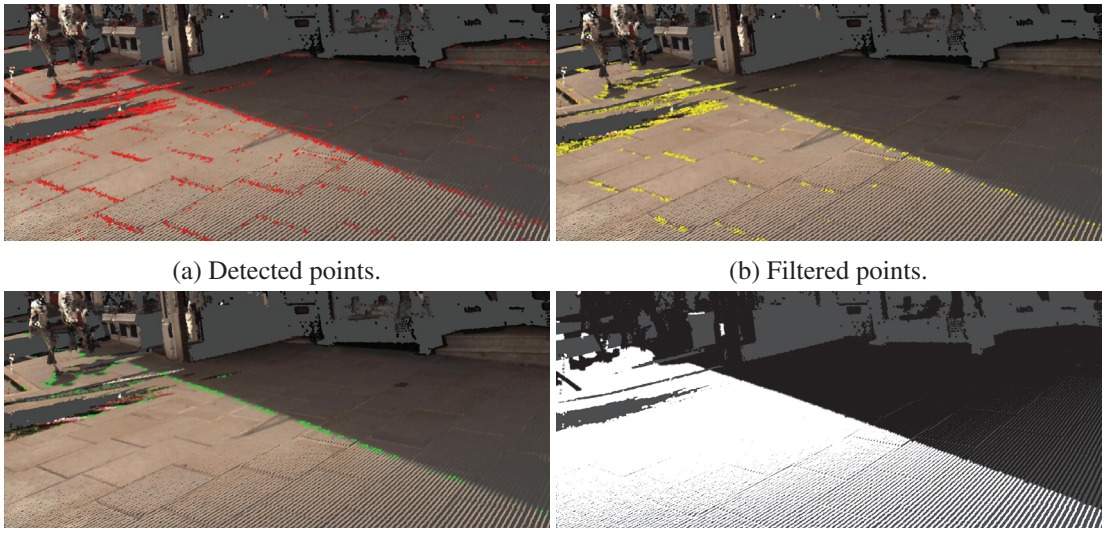
Figure 4.11: Histogram of the different $La*b*$ channels around a shadow interface.

where $E_{smooth} = e^{-\frac{1}{2\sigma^2}(L(p)-L(q))} \mathbb{1}_{\delta(p)\neq\delta(q)}$,

$$E_{data}(\delta, p) = \begin{cases} e^{-\frac{1}{2\sigma^2}(L(p)-L_S)^2} & \text{if } \delta(p) = 0 \\ e^{-\frac{1}{2\sigma^2}(L(p)-L_L)^2} & \text{if } \delta(p) = 1 \end{cases}$$

and γ is a weighting term set to 1 in our experiments.

This type of energy can be easily minimized using graph cuts, as introduced by BOYKOV, VEKSLER, and ZABIH [BVZ01] and BOYKOV and KOLMOGOROV [BK04], by using E_{data} as the source and sink edge costs and E_{smooth} as the inter-pixel edge cost. The labels obtained by the graph cut are then back-projected on the point cloud to assign the points to the shadow mask. Figure 4.12 shows the results of initial interface detection, interfaces filtered by density,



(a) Detected points. (b) Filtered points.

(c) Shadow interface points. (d) Shadow mask.

Figure 4.12: Points detected as potential shadow interfaces (4.12(a)), density filtered points (4.12(b)), shadow interface points (4.12(c)), and shadow mask from the graph cut. The shadow appearing at the bottom left of the graph cut mask (4.12(d)) is due to the acquisition vehicle that is visible on the pictures but not on the point cloud.

by histogram analysis and the final graph cut result, demonstrating how the graph cut turns the sparse set of shadow edges into a full segmentation. This step retrieved the proper shadowed areas in a scene where the surface reflectance changes widely due to material variation (*i.e.* , white building wall vs darker building wall).

4.2.3 Shadow correction

Once the shadows are segmented, the next step is to project the labelling on the point cloud, and to modify the color of the points so as to obtain a shadow-free point cloud.

Illumination model

According to YU et al. [Yu+99], LAFFONT, BOUSSEAU, and DRETTAKIS [LBD12] and RAMAKRISHNAN, NIETO, and SCHEDING [RNS15] the luminance of a point lying on an lambertian outdoor surface can be defined as:

$$L = R(S_{sun} + S_{sky} + S_{indirect}) \tag{4.8}$$

$$S_{sun} = V_{sun} cos\alpha E_{sun} \tag{4.9}$$

$$S_{sky} = \int_{\Omega_{sky}} cos\theta_{sky} E_{sky} \tag{4.10}$$

$$S_{indirect} = \int_{\Omega_{indirect}} cos\theta_{indirect} E_{indirect} \tag{4.11}$$

With I the illumination, R the albedo of the material, S_{sun} the lighting contribution of the sun, S_{sky} the lighting contribution of the sky (ambient light) and $S_{indirect}$ the lighting contribution of the light reflected by the objects in the scene. All notations are summarized on Figure 4.13. Using this simplified model, the presented algorithm proceeds in two steps, correcting first the luminance component and second the chrominance component at each point.

Luminance correction

The luminance correction step requires an approximation of the sunlight orientation to be able to separate the different component of the scene lighting. Although the method proposed by WEHRWEIN, BALA, and SNAVELY [WBS15] locates the sun orientation without *a priori* information, it cannot be applied on large urban scans due to the lack of precision on the surface to detect attached shadows. However, the sun azimuth and elevation can be estimated if the approximate GPS position and time of the data acquisition are known [RA04], leading to the sun orientation estimation. This is the method that is used to obtain the sun ray orientation relative to the scene. Knowing the position of several shadow points, the amount of illumination provided by the sky can be roughly estimated. Indeed, for shadow points the sun contribution can be safely ignored and the luminance of these points writes: $L = R(S_{Sky} + S_{Indirect})$, where R is the surface reflectance.

The problem is simplified by considering that $S_{Sky} + S_{Indirect} = \beta \times E_{sky}$ where $\beta = 0.5 \times \langle \mathbf{n}(P), \mathbf{n}_{ground} \rangle + 1$. Thus β varies proportionally to the dot product of the ground normal $\mathbf{n}_{ground}$ and the point normal $\mathbf{n}(P)$.

This estimated lighting ratio β from the sky is error-prone in the case of urban environment where streets can be tightly enclosed by buildings, greatly reducing the real contribution of the sky compared to the contribution of indirect lightening. However it is still a good first approximation of the amount of light coming from the environment. It is thus possible to make the luminance uniform for shadowed points by computing $\bar{\beta}$ the average β in the shadowed area:

$$L'(P) = L(P) \frac{\bar{\beta}}{\beta(P)} \tag{4.12}$$

The matter is quite different for sunlit points. In a similar manner as RAMAKRISHNAN, NIETO, and SCHEDING [RNS15] we define the sun-sky-ratio (SSR) of lighting between the sun and the sky:

$$SSR = \frac{\bar{L_L}\bar{\beta_S} - \bar{L_S}\bar{\beta_L}}{\bar{L_L}(\bar{cos\alpha})_L} \tag{4.13}$$

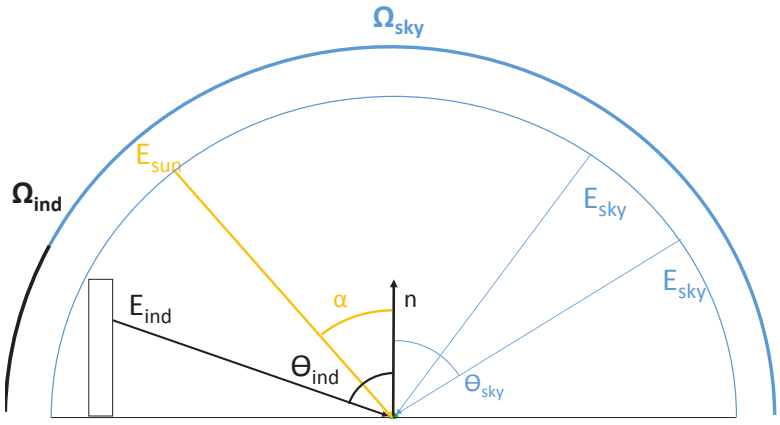


Figure 4.13: Illumination model of a point. The illumination is decomposed into 3 components: a sky contribution, a sun contribution and an indirect lightening contribution. Each sky ray ($\theta_{sky} \in \Omega_{sky}$) contributes to an energy E_{sky}, the sun contributes to an energy E_{sun} in direction α and the rays corresponding to indirect illumination ($\theta_{ind} \in \Omega_{ind}$) contribute to an energy E_{ind}. In our simplified model the indirect contributions are omitted.

where α is the angle between the sun direction and the points normal. The straight bar above the notation means that the quantity is averaged on a light or a shadow area respectively, depending on the indice $_L$ or $_S$. More details on the derivation of this formula can be found in the original publication.

This ratio, allowing us to recompute the luminance of points in sunlight, is defined as:

$$L'(P) = L(P)\frac{\bar{\beta}_S}{SSR\cos\alpha + \beta(P)}. \tag{4.14}$$

Applying these transformations to the luminance of the point successfully relight them in a realistic manner. Once the luminance is corrected, the chrominance can also be adapted.

Chrominance correction

The chrominance correction is simpler: it uses a simple chrominance ratio between the sunlit points and the shadowed points. It is performed for both the a and b components expressed in the $La*b*$ colorspace.

$$b'(P) = b(P)\cdot\frac{\bar{b_S}}{\bar{b_L}} \text{ and } a'(P) = a(P)\cdot\frac{\bar{a_S}}{\bar{a_L}} \tag{4.15}$$

This trivial update is enough for chrominance correction, as presented below in our experiments.

Penumbra zones

After this two-step correction, there may remain some artifacts near the boundaries of the shadowed and sunlit areas which should be corrected (Figure 4.14). Indeed, having a binary shadow mask may lead to unwanted effects along the shadow edges where the boundary between light and darkness is not sharply defined. These effects are typically due to under- or over-compensation in the luminance and chrominance correction around the shadow edges, which induces the apparition of a strong linear artifact along the border. This artifact is clearly visible in Figure 4.14(c). To alleviate this effect an in-painting step is performed around the known shadow border. Since the area to inpaint is very small, we avoid any sophisticated inpainting method such as variational or patch-based inpainting and use a median filter to guess the missing pixel colors. This last step, mitigates the visual artifacts caused by overcompensation as depicted in Figure 4.14(d).

4.2.4 Results

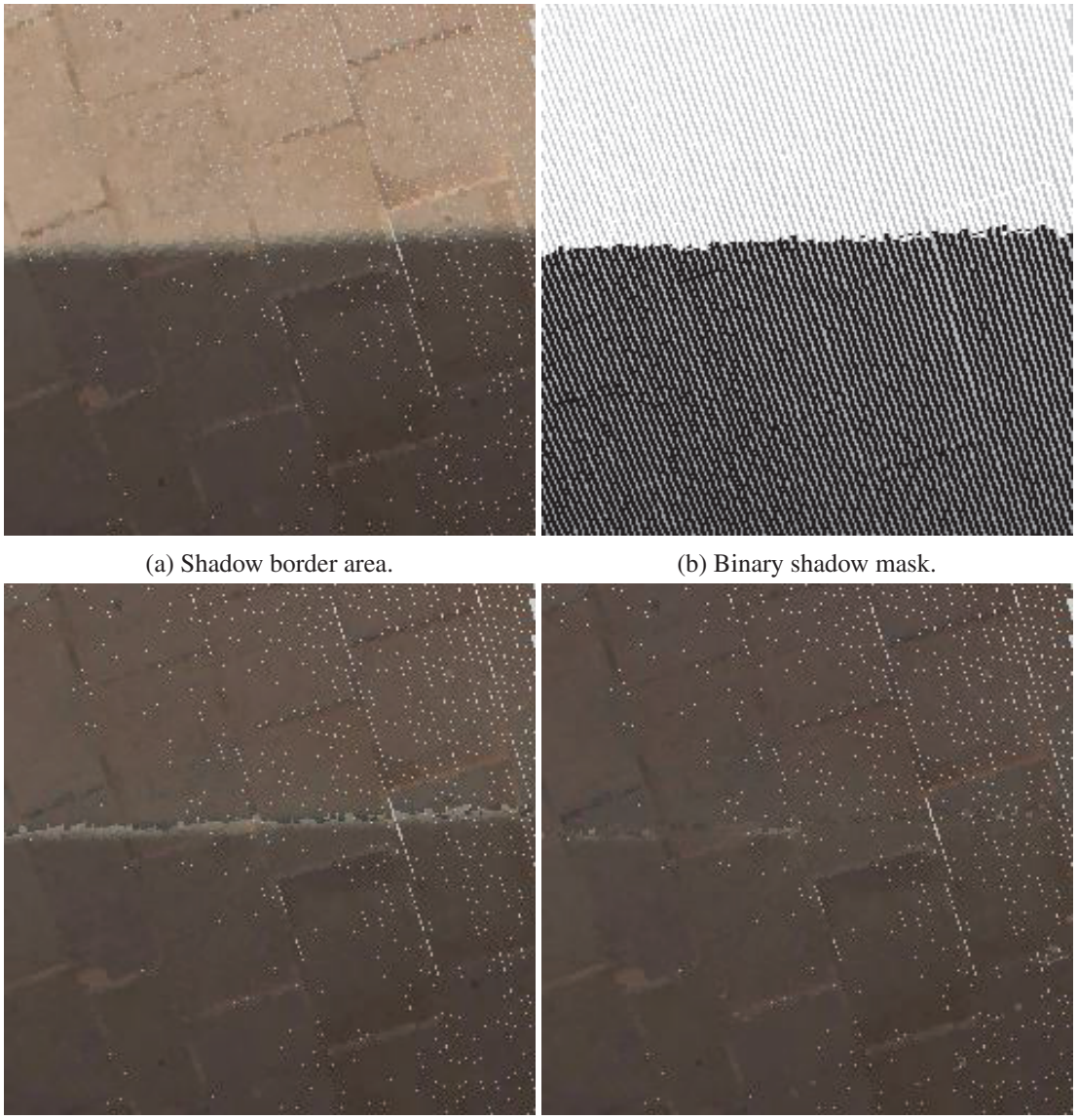
The efficiency of the proposed shadow detection and correction schemes is demonstrated on two different sets of data. These two datasets are detailed in chapter 1. In these two sets, images and point clouds have been acquired at the same time.

Shrewsbury Dataset

The Shrewsbury dataset offers smooth reflectances features that reflect well the luminance variation of the materials, almost without any noise. As can be seen in Figure 4.15, it is possible to relight a point cloud by switching its dark points to a sunlit state or switching the sunlit points to a shadow state (see Figure 4.15(d)). Other locations displaying strong shadows were tested as well, as shown on Figure 4.16, Figure 4.17 and Figure 4.18. In most cases the shadow is properly detected on the ground plane and extracted using the graph cut.

KITTI Dataset

Our second test case is the KITTI Dataset [Gei+13]. Due to the nature of data, some pre-processing was performed to obtain a point cloud dense enough with smooth reflectance values. First a set of 20 scans around an image were merged together. From this merged point cloud, the point density was unified to regularize the density. This point cloud was then colored using the color camera. This small pre-processing yielded a point cloud dense enough to be exploited jointly with the images. The proposed algorithm was then run with the standard parameters defined in sections 4.2.2 and 4.2.3. Figure 4.19 and Figure 4.20 show that strong shadows were successfully detected by the process. However, some soft shadows located on very bright regions (*e.g.* , the shadow of the tree on the white wall on the left) have not been correctly identified as shadows.



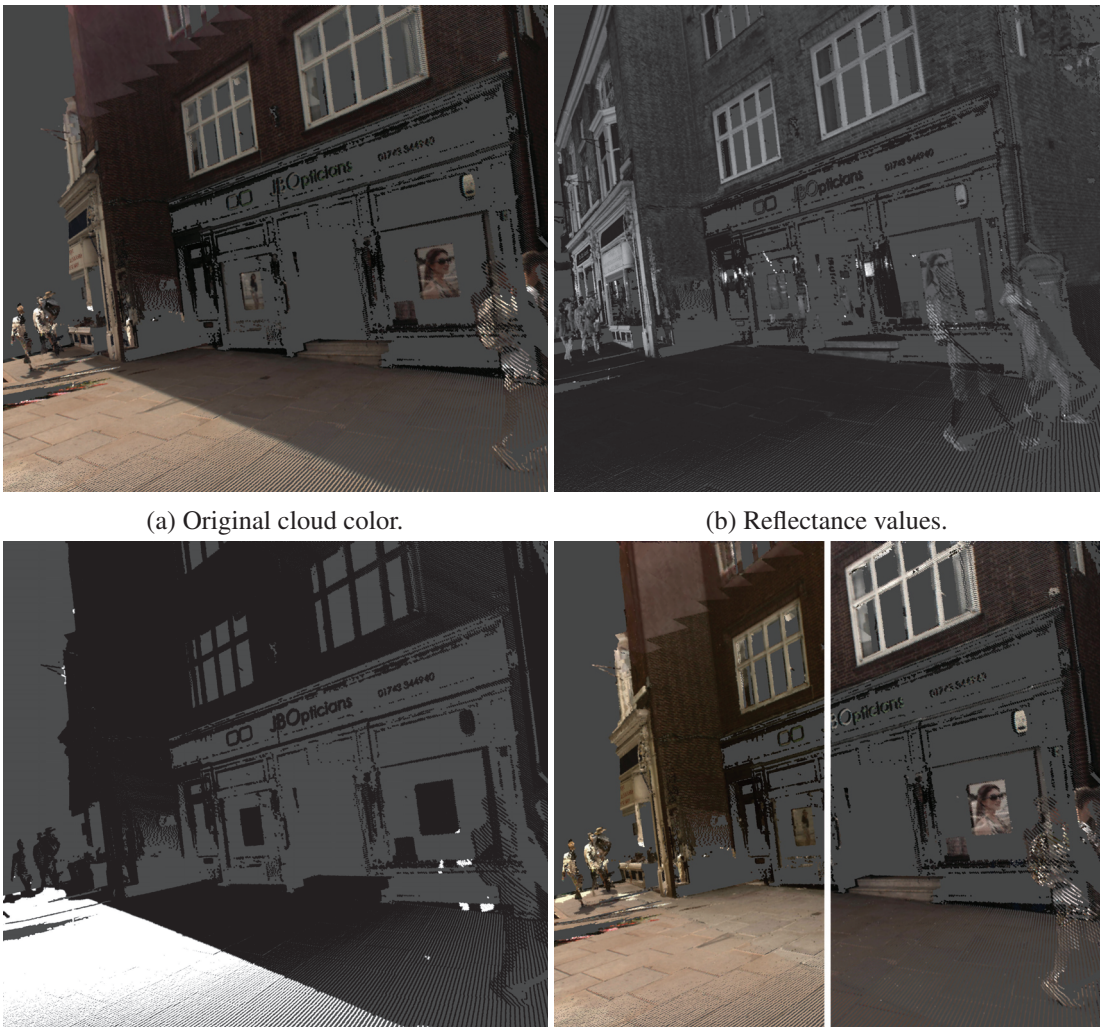
(a) Shadow border area.

(b) Binary shadow mask.

(c) Re-lighted area.

(d) Re-lighted area with local median filtering.

Figure 4.14: Small penumbra zones on the boundary of a shadow can have non negligible effect when re-lighting a point cloud. A simple median filtering of points located on the boundary of the shadow mask mitigate the apparition of artifacts.



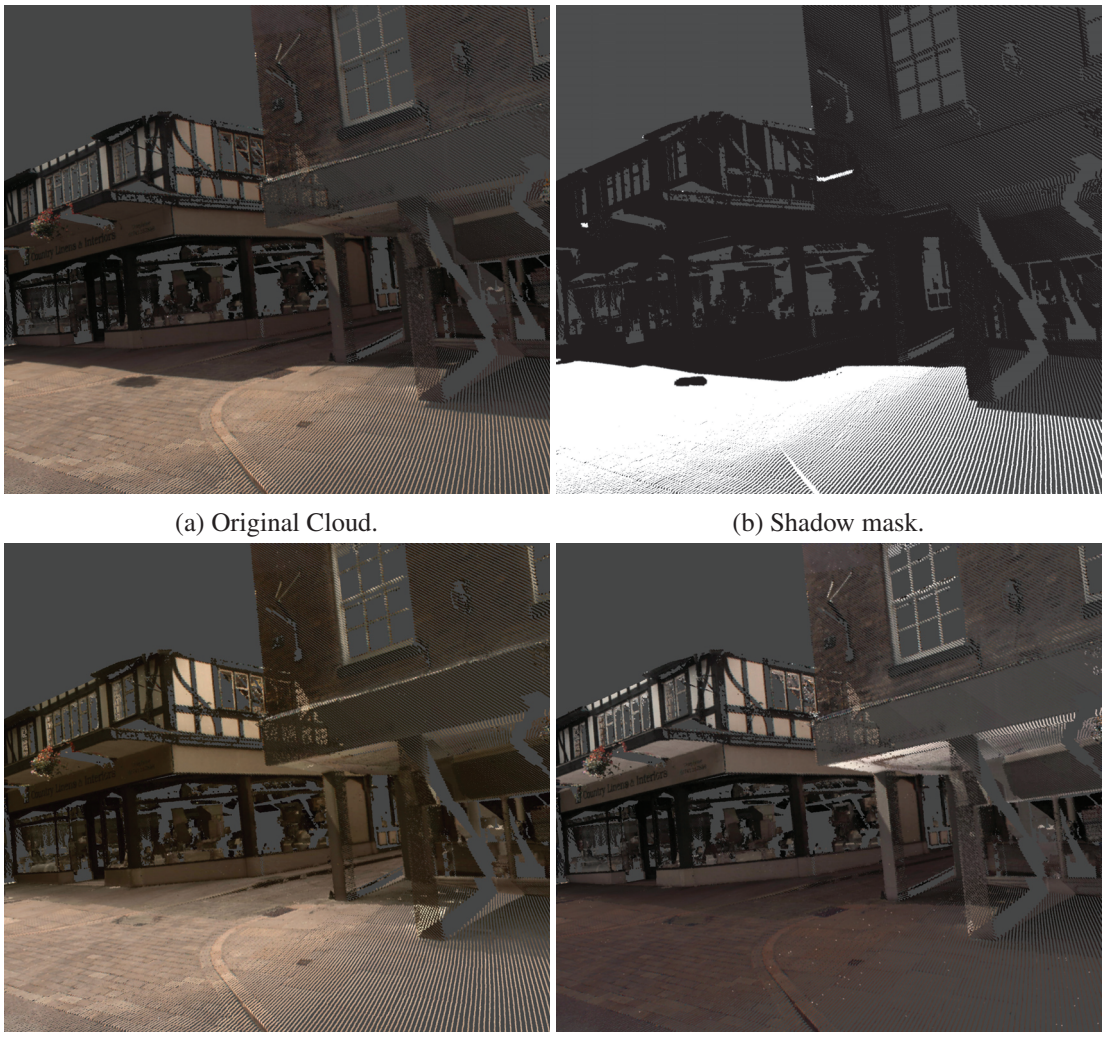
(a) Original cloud color.

(b) Reflectance values.

(c) Shadow mask.

(d) Relighted cloud (light/dark).

Figure 4.15: Comparison between the original color and the relighted colors of a fragment of the Shrewsbury point cloud where the shadows were removed.



(a) Original Cloud. (b) Shadow mask.

(c) Relighted cloud (light). (d) Relighted cloud (dark).

Figure 4.16: Comparison between the original color and the relighted colors of a fragment of the Shrewsbury point cloud.

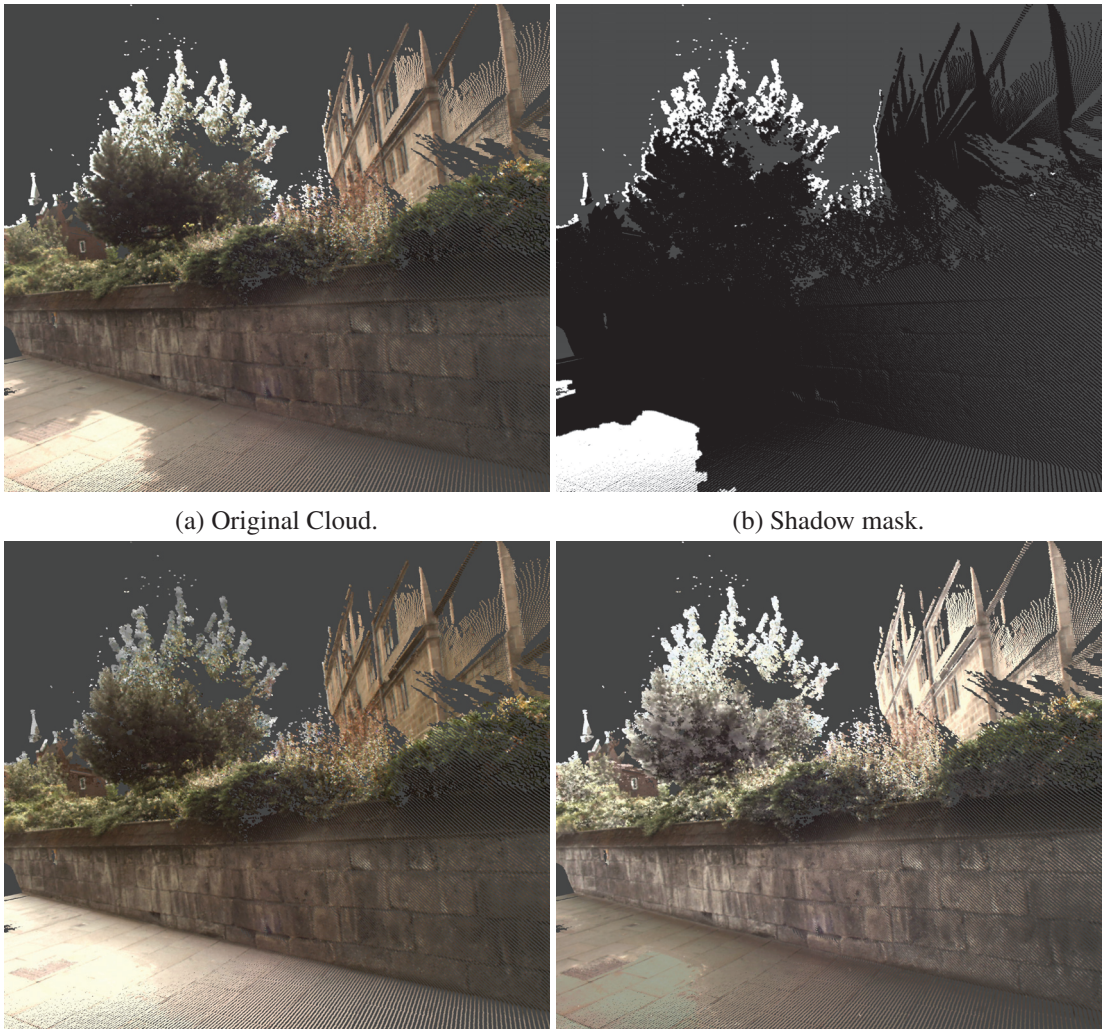
(a) Original Cloud.

(b) Shadow mask.

(c) Relighted cloud (light).

(d) Relighted cloud (dark).

Figure 4.17: Comparison between the original color and the relighted colors of a fragment of the Shrewsbury point cloud.



(a) Original Cloud. (b) Shadow mask.

(c) Relighted cloud (light). (d) Relighted cloud (dark).

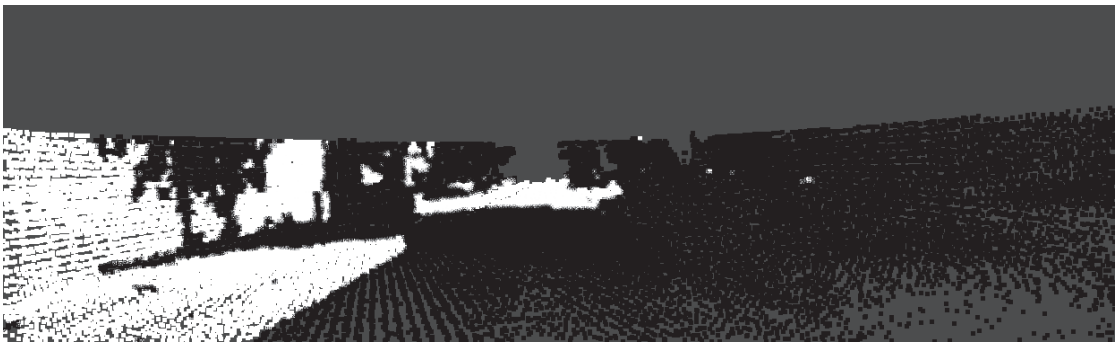
Figure 4.18: Comparison between the original color and the relighted colors of a fragment of the Pegasus point cloud.

4.2.5 Discussion

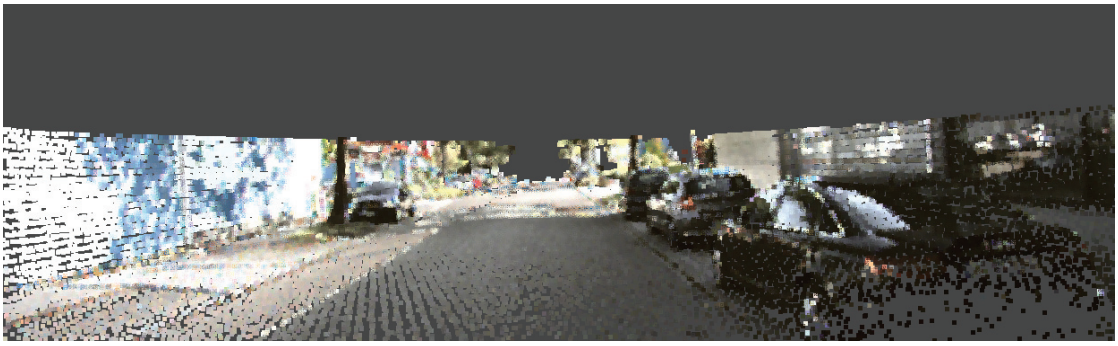
As previously explained, sharp shadows can be correctly detected and corrected by our process. Unfortunately, in some cases, the relighted areas do not possess the exact same luminance and chromaticity as its original counterpart. This is slightly noticeable in Figure 4.16 and Figure 4.17 but is very clear on Figure 4.19. Several factors may be responsible for this. First, the illumination model is only an approximation and does not perfectly reflect the illumination of the object (indirect and sky illumination). Second, the light source is supposedly uniform for each point of the cloud. However, illumination power may also vary in the case of a partially clouded sky. These two factors as well as the existence of soft shadows may account for the residual observable color difference. Another limitation lies in the lack of robustness to moving vehicles or pedestrians. They might not be at the same place in the image or in the point cloud which makes the joint exploitation fail.

(a) Photo.

(b) Colored cloud.



(c) Shadow mask.

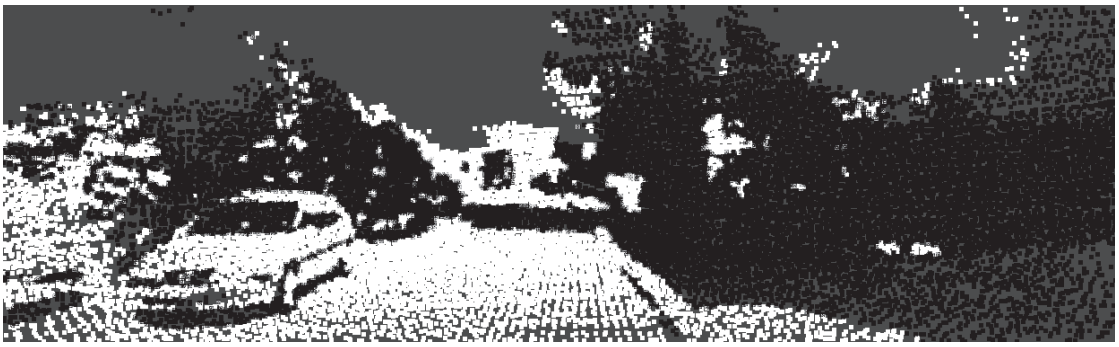


(d) Relighted cloud.

Figure 4.19: Example of shadow detection and cloud re-lighting on the KITTI dataset.

(a) Photo.

(b) Colored cloud.



(c) Shadow mask.

(d) Relighted cloud.

Figure 4.20: Example of a shadow detection and cloud re-lighting on the KITTI dataset.

CHAPTER 5

TOWARD POINT CLOUD DENSIFICATION

Contents

Point clouds acquired using mobile LiDAR systems are sometimes incomplete or display important variations in sampling density. In almost all cases, the acquired point density is significantly smaller than the information density visible in a color image. This low density has drastic consequences when projecting the point cloud on an image plane as described in chapter 2. We discuss in this chapter of a joint model that would be capable of improving the density of the geometric information from a sparse point cloud, using multiple registered images.

5.1 Related works

The problem of upsampling a point cloud is closely related to the upsampling and disocclusion of a depth map, which has been extensively studied. We can distinguish several concepts to perform a depth map upsampling. Some methods rely solely on the original depth map to estimate the unknown depth of certain pixels, but the majority of them uses the color image corresponding to the depth map. This color image, that renders the scene with the same parameters as the depth map, allows for an improvement of the disocclusion or upsampling process. Within these methods using a color image, we can further distinguish methods that rely on a global variational approach to fill the unknown pixels from the methods that rely on the local color propagation.

Single depth map method

Some authors prefer to not use color images at all in their depth map densification process, even if they are usually available. PREMEBIDA et al. [Pre+16] propose a method based on a joint

bilateral filter to obtain the depth map of a LiDAR point cloud without using the color values of an image. They introduce a modified version of a bilateral filter that uses the estimated depth of pixels in order to retrieve a consistent depth map.

VALLET and PAPELARD [VP15] propose to generate a ground orthophoto of a road network from the corresponding terrestrial LiDAR scans. To do so, they use Poisson interpolation to fill the visible gaps in the projection of the LiDAR point cloud on the orthoimage plane. This method yields both a completed height and reflectance orthoimage. Similarly, BIASUTTI et al. [Bia+17] discuss a method to create a dense orthoimage by diffusing depth and reflectance, and by inpainting the visible occlusion gaps. The presented results of these last two methods are however limited to road networks aerial views.

Variational methods

Some other kind of methods try to estimate all the unknown depth pixels at once, so that they are consistent between each other and with the reference color image. Such methods include for instance the work of BEVILACQUA et al. [Bev+17] who propose to simultaneously estimate depth, reflectance and visibility using a variational optimization. Their method to address the visibility problem yields an improved result compared to simple nearest neighbor interpolation as well as other variational methods. DROZDOV, SHAPIRO, and GILBOA [DSG16] propose to combine a super-pixel decomposition with a total-generalized-variation to recover heavily degraded depth maps. This method successfully recovers complete depth maps from very sparse depth maps that are similar to LiDAR point cloud projection. Variational depth map densification methods yield consistent and dense depth maps. However, solving such problems is computationally expensive and no formulation was developed to handle several depth maps simultaneously.

Depth propagation methods

Several methods solve the depth map inference problem by propagating information from neighboring pixels. The simplest of these methods consist in propagating the color from nearby known depth pixels to unknown ones while using the color image to guide the propagation process.

In their work, BUYSSENS et al. [Buy+17] propose to use inpainting in order to fill the occlusions produced during the creation of synthetic RGB-D view of a scene. Their depth inpainting method relies on a diffusion of the depth in unknown zones, coupled with a structure propagation step. This structure propagation step uses T-junction to fill missing areas of the depth image and boundaries of the objects. The proposed framework also reconstructs a color image that keeps a sharp edge definition even on occluded areas.

Joint Bilateral Upsampling as introduced by KOPF et al. [Kop+07] is an interesting image upsampling method based on a bilateral filter. For each unknown pixel it uses its neighbors, but also information gathered in a high resolution color image to determine missing information. This color information introduces an anisotropic component and allows to better respect the discontinuities around the color edges. This method is also interesting due to its low complexity. Among all possible applications of the Joint Bilateral Upsampling, the authors identify the possibility to improve the density of depth maps. DOLSON et al. [Dol+10] used this technique on

depth maps created from LiDAR point clouds with the addition of priors on object motion in the particular case of dynamic environments.

In the wake of the joint bilateral upsampling, LIU, TUZEL, and TAGUCHI [LTT13] propose the Joint Geodesic upsampling, which suggests to upsample a low resolution depth map using the geodesic distance in a full resolution color image. While a complete computation of the shortest geodesic path for each pixel would become intractable in term of computation time on large images, the authors propose an approximate formula with linear complexity. The presented results are comparable or better than the joint bilateral upsampling in term of edge preservation.

HE, CHEN, and LI [HCL15] further develop the principle of using a monocular color camera coupled with a sparse point cloud projection to obtain a dense depth map. In their work, they introduce an anisotropic diffusion tensor that uses geodesic distance to complete missing elements of a depth map. Their method has the advantage to be almost parameter free and seems to keep sharp depth discontinuities on LiDAR based depth maps.

Most of the works presented for the improvement of the point cloud either rely on complex variational problems to increase the density of a depth map or take only a single image and depth map pair into account. None of them considers a full dataset of images registered to the point cloud to improve its density by adding more consistent points. Moreover, in our case we dispose of another unexploited information that is a consistent and high quality color associated with each point of the cloud as it has been introduced in chapter 4.

5.2 Proposed approach

We propose a method that increases the density of a point cloud using the information from a set of images. While some areas of point clouds tend to have a regular sampling, other parts may have either irregular sampling or a complete lack of information due to occluding objects.

There are three main objectives for the presented method:

- Recover partially scanned structures.
- Fill occlusion gaps.
- Increase the point cloud density.

These objectives must be completed while retaining the geometrical features of the surface and a high geometrical precision.

5.2.1 Overview

Depth map upsampling algorithms are not directly applicable to increase the density of a point cloud. Indeed, depth map density improvement on a single image is not adapted to urban point clouds and will not fill partial structures nor occlusion gaps. To overcome these problems, a multi-scale density improvement is proposed, coupled with a multi-image validation process. This

way, each independently upsampled projection of a point cloud on an image plane corresponding to a real color image is consistently verified. The overview of the method detailed below is also summarized in algorithm 2.

The multiscale approach aims to mitigate the influence of visibility problems while improving the capability of upsampling methods to handle large gaps in the original depth map. This aspect of the method is discussed in section 5.2.3. At different scales, starting from a scale as low as $1/8$ of the original image size, the dense depth maps corresponding to all the color images representing the point cloud are computed. The sparse depth maps are obtained by projecting the point cloud on image planes corresponding to real photos. These depth maps are upsampled using a modified version of the joint bilateral upsampling (section 5.2.2). Once these depth maps are properly upsampled, the next step is to validate all newly estimated depth pixels. This validation step is performed by comparing the consistencies of depth pixels between all the upsampled depth maps (section 5.2.4). Once the inconsistent depth pixels are removed from the upsampled depth maps, 3D points are estimated using the newly validated depth pixels. These new 3D points are added to the point cloud in order to increase its density. This process is repeated several times until a defined scale is reached. The result is a point cloud that was densified at several scales using only validated, consistent 3D points.

Data: PC_0 a point cloud
S_I a set of registered images
Result: PC a densified point cloud
$scale = 1/8$;
$PC = PC_0$;
while $scale \leq 1/2$ **do**
    Project the images to obtain a set of sparse depth maps S_{DM};
    Fill the empty parts of the depth maps S_{DM};
    Validate each depth map of S_{DM} with each other depth map from S_{DM};
    Extract the cloud PC_{scale} from the new depth pixels of the set of depth maps S_{DM};
    $PC = PC + PC_{scale}$;
    $scale = scale \times 2$;

ALGORITHM 2 – Point cloud density improvement.

5.2.2 Sparse depth map upsampling

As introduced in chapter 2, the projection of a point cloud, acquired using a LiDAR scanner, on an image plane yields a very sparse image. Since we know the 3D position of the points relative to the camera position, we can compute the depth map that corresponds to the color image. The idea is to upsample this depth map to increase the amount of known depth pixels, in order to then increase the amount of 3D points in the cloud.

Overview of the joint bilateral upsampling

The presented method is based on the Joint Bilateral Upsampling [Kop+07]. As it is already envisioned by its authors, the joint bilateral upsampling can be used to upsample a low resolution depth map. The principle of this upsampling method applied to depth maps is to determine the depth of unknown pixels by using its known neighbor pixels. In comparison to a standard bilinear interpolation that leads to an oversmoothing of important depth changes, the bilateral upsampling proposes to better respect these depth changes. The method is built on the fact that brutal depth changes in the depth maps are often related to color change in a corresponding color image. In this case, the addition of a weight allows to introduce an anisotropic component in the upsampling. This weight is computed by comparing the color of the neighbor pixels, used to determine the unknown pixel depth, with this unknown pixel color in the high resolution (color) image. Thus, a pixel with a known depth whose color is really different from the considered unknown depth pixel color will have little to no influence on this pixel final estimated depth (see Figure 5.1). On the contrary, a neighbor pixel that shares the exact same color will have a much more important influence.

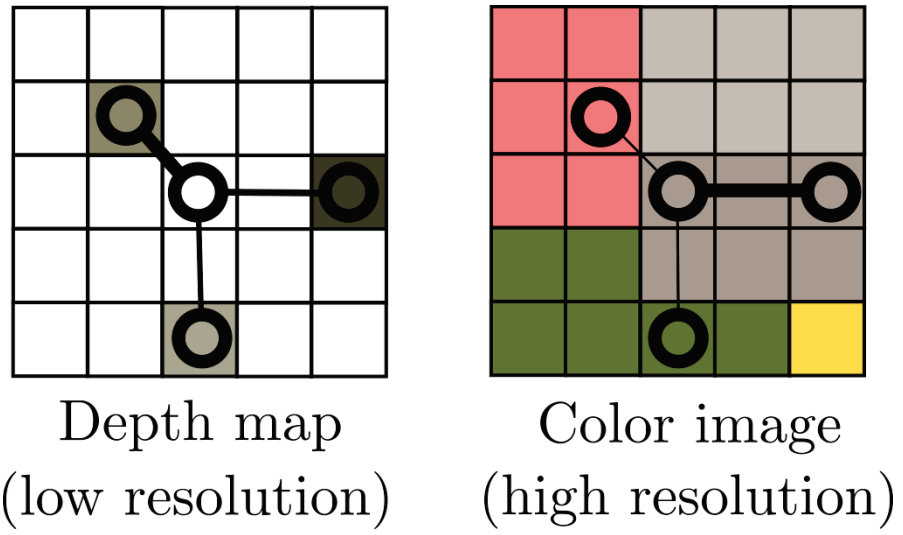


Figure 5.1: Illustration of the joint bilateral upsampling. It uses the information contained in a low resolution image (*e.g.* a depth map) combined with the information contained in a high resolution image (*e.g.* a color image) to infer the information of unknown pixels of the upsampled low resolution image.

Considering a sparse depth map, the objective is to determine the depth of unknown pixels. As described by its authors, for each unknown pixel p of the sparse depth map, it is possible to determine the potential depth d_p using the following joint bilateral upsampling:

$$d_p = \frac{1}{w_p} \sum_{q \in N_p} d_q e^{-\frac{\|p-q\|^2}{2\sigma_d^2}} e^{-\frac{\|c_p^I - c_q^I\|^2}{2\sigma_c^2}} \tag{5.1}$$

Where:

- d_p is the estimated depth at the pixel p.
- w_p is the normalizing factor.
- N_p is the set of pixels in a limited neighborhood around the pixel p.
- c_p^I is the color at the pixel p in the color image I.
- σ_d and σ_c are parameters of the Gaussian filter for the distance and color respectively.

Joint trilateral upsampling

As it is mentioned in section 5.1, this joint bilateral upsampling is perfectly capable of filling the missing pixels of a sparse depth in a rather consistent manner. However, as we are working on properly colored point clouds (see chapter 4), we can slightly improve this upsampling method. Indeed, by adding a third filter based on the color given to a point and the color as it is perceived in an image, we can avoid inconsistent depth pixels upsampling. By comparing the consistency between those two colors, we can decrease the weight given to inconsistent points.

We propose to determine the potential depth d_p for each unknown pixel p of the sparse depth map using the following joint trilateral upsampling:

$$d_p = \frac{1}{w_p} \sum_{q \in N_p} d_q e^{-\frac{||p-q||^2}{2\sigma_d^2}} e^{-\frac{||c_p^I - c_q^I||^2}{2\sigma_c^2}} e^{-\frac{||c_P - c_q^I||^2}{2\sigma_c^2}} \tag{5.2}$$

The notations are similar to the ones given in the original joint bilateral upsampling equation (equation 5.1). The only newly introduced term is c_P, which is the $La*b*$ color associated with the point P. To improve the consistency of the color difference estimation, the color c_p^I is also converted to the $La*b*$ colorspace.

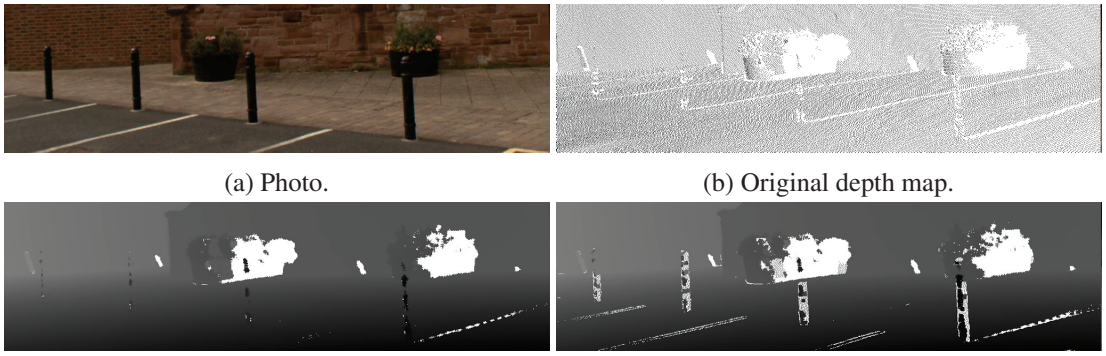
This upsampling allows to better preserve the visible edges in the color image during the depth estimation. Once a dense depth map has been obtained, it is possible to estimate the 3D position of the points that correspond to newly estimated depth pixels. Knowing the image position O_I in the world, the position of each pixel center C_p in the world and the estimated depth d_p for each of these pixels, we can roughly estimate the new 3D point P position as:

$$P = O_I + d_p \cdot \frac{\vec{O_I C_P}}{||\vec{O_I C_P}||} \tag{5.3}$$

However even with the anisotropic component, the estimated depth still tends to be over-smoothed around brutal depth changes. The very sparse nature of the point cloud also induces errors during the upsampling where partial structures are not sampled properly.

Residual upsampling artifacts

The acquisition of some particular urban furniture, such as posts, lamp poles or road signs can be only partial. Partial acquisition is characterized by a low density of points, with spaced out clusters separated by empty areas. In such cases, the direct application of the joint bilateral upsampling will produce artifacts. Indeed, background points will be projected on the image plane in place of another point, assigning a bad depth value with a certain color that will result in a blending of inconsistent depth values. These artifacts can be seen in Figure 5.2(c) where the posts are not upsampled with a consistent depth along their entire length. This effect is produced when the sampling is so sparse that points from the background are projected on the image plane where points from the post should be. The disocclusion step of chapter 2 is only able to fix the problem for relatively small areas. As this point is associated with a certain pixel value, the joint bilateral upsampling still uses its depth value without any consistency check. The joint trilateral upsampling avoids these inconsistent smoothing but also fails to properly fill missing areas (Figure 5.2(d)).



(a) Photo. (b) Original depth map.

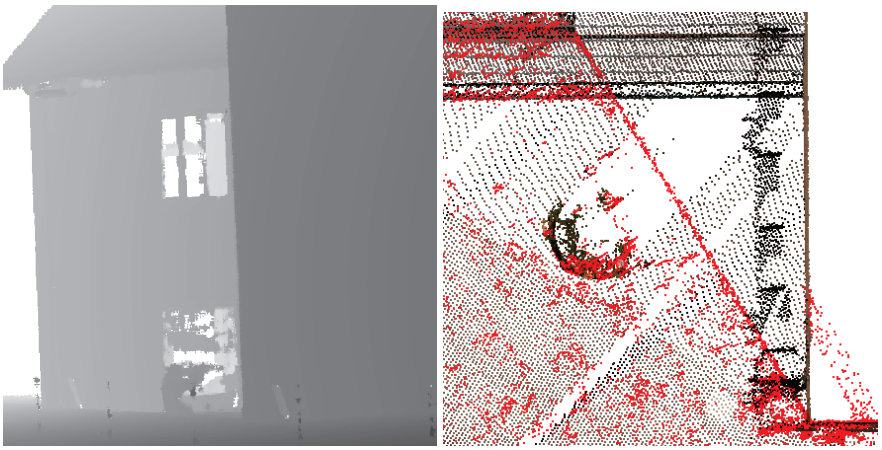
(c) Upsampled depth map (Joint Bilateral). (d) Upsampled depth map (Joint Trilateral).

Figure 5.2: Effect of the direct application of the joint bilateral and trilateral upsampling. The very low sample density of the road posts produce a bad upsampling.

As the bilateral and trilateral upsamplings use a small kernel size, not all the points are successfully upsampled. The remaining areas visible behind the flower pots in Figure 5.2(c) and around the posts in Figure 5.2(d) do not contain any depth value. These large occlusion gaps behind the flowers pots are not filled with new depth values.

Despite the use of bilateral or trilateral upsampling, some estimated depth values tend to be oversmoothed. This can be mostly explained by points assigned to wrong pixels in the color image. Using an upsampled depth map to generate new points for a cloud produces a high number of artifacts along the viewing direction on abrupt depth changes. While it is difficult to estimate the amount of artifacts around brutal depth changes by observing a depth map, this evaluation is fortunately made straightforward when re-projecting the pixel depths as 3D points, as illustrated by Figure 5.3(b). In this particular top view, the new points obtained using the depth map (Figure 5.3(a)) are displayed in red. The color similarity between the two walls produced a smooth interpolation that appears as the diagonal line of red points in Figure 5.3(b) where a

discontinuity should be clearly visible.



(a) Upsampled depth map.

(b) Point cloud (top view) obtained by depth map 5.3(a).

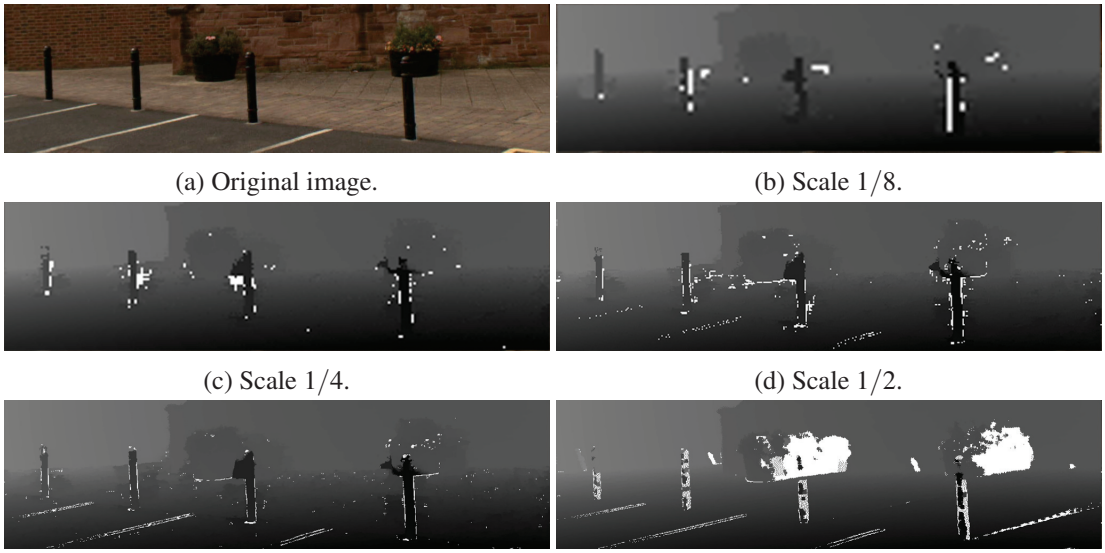
Figure 5.3: The upsampled depth map in Figure 5.3(a) was used to estimated new points appearing in red in 5.3(b). The diagonal of red points is composed solely of outliers whose positions were badly estimated.

5.2.3 Multi-scale upsampling

We resort to a multi-scale approach to address the observable problems around partial structures. By starting at low resolution depth map and progressively increasing the resolution, partial structure upsampling problems tend to be mitigated. Indeed, for a given depth map, a lower resolution depth map is a coarser representation of the same 3D data. This coarse representation contains fewer undefined holes and is less subject to sparse-sampling artifacts.

The process used to perform the multi-scale upsampling is straigthforward. Starting at a given smaller scale version of the original image (*e.g.* $1/8$), a depth map is created at this scale by projecting the points of the cloud on the image plane. This depth map is then upsampled using the process described in section 5.2.2. Once the depth map is upsampled, the newly estimated points are added to the cloud. This enriched cloud can then be used to repeat the process at a higher scale (*e.g.* $1/4$) by creating a new depth map at this scale. Points added at lower resolutions help to reduce the visible artifacts compared to a simple direct full scale upsampling as visible in Figure 5.4. In this figure we can clearly see that the posts are almost completely reconstructed at scale 1, whereas the direct upsampling at full scale leaves undefined surfaces. Large gaps are also filled properly within the depth map which is not achieved by direct upsampling.

However, using this multi-scale approach also produces artifacts. As visible in Figure 5.4(e) the depth of the posts actually bled on the background flower pots. This is partly due to the fact that the flower pots and the posts have almost the same color. Another effect is the apparition of noise on smooth surfaces. For each iteration, errors sum up and produce low frequency variation



(a) Original image. (b) Scale $1/8$.

(c) Scale $1/4$. (d) Scale $1/2$.

(e) Final upsampling (scale 1). (f) Direct upsampling.

Figure 5.4: Sequential upsampling of a single depth map with 4 different scale levels compared to a direct upsampling.

artifacts on planar surfaces, such as the ground. While most of these effects can be removed during the validation step, a better precision during this step could improve the complete process.

5.2.4 Multiple depth map validation

To overcome the problem introduced by the wrongly estimated depth pixels, it is possible to combine information from multiple images. Indeed, with different points of view, the produced artifacts should not be the same in all images. In this case, by comparing the upsampled results from one image to several other images, outliers can be easily removed. Different possibilities exist to validate upsampled points with other images. We present here a simple scoring system, illustrated by Figure 5.5, that works reasonably well in consolidating upsampled points.

For each newly upsampled pixel p of a depthmap I, we determine the corresponding 3D point P by backprojecting the depth data. This 3D point P is then re-projected in each upsampled depth map I_n. Let p'_{I_n} be the pixel where the point P is projected in a depth map I_n. By comparing the depth estimated or known at pixel p'_{I_n} with the distance of point P to the image position O_{I_n} we can either increase or decrease the validity score ξ_p. If $|||P - O_{I_n}|| - d_{p'_{I_n}}| < \varpi$ the score ξ_p is increased by one. In the other case, the validity score ξ_p is decreased by one. The threshold value ϖ is an absolute value, in meter, that gives an approximate control on the final depth map precision.

Once the point P is tested against all the other depth maps, if its score ξ_p is less than 0 the point is discarded. This validation method ensures that a strict majority of depth maps agree on the points that are to be added to the cloud. Pixels with unknown depth are not taken into

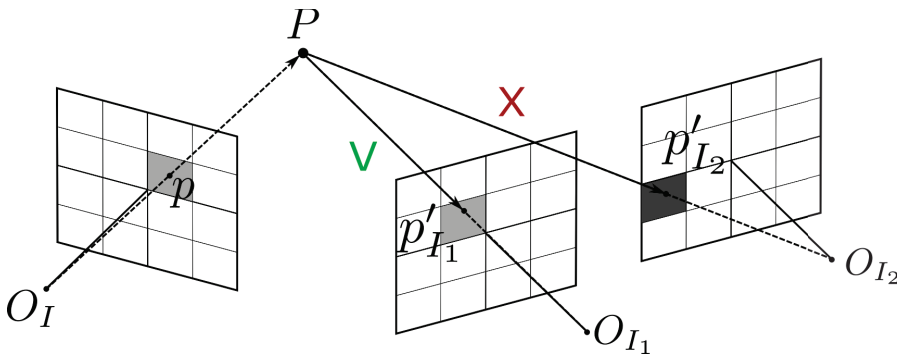


Figure 5.5: Illustration of the multi-depth map validation process. For each upsampled pixel p of a depthmap I, its 3D projection P is reprojected in n other depthmaps. Pixels p'_{I_n} of the depthmap I_n that verify the condition $|||P - O_{I_n}|| - d_{p'_{I_n}}| < \varpi$ increment the validity score ξ_p, and decrease it otherwise.

account for the majority vote, which allows to neglect areas where the upsampling did not add any meaningful data.

If the majority of depth maps agrees on a certain point, it is only up to the ϖ precision. Moreover, the multiple projections tend to accumulate errors which can result in a significant noise level. This inaccuracy can be easily tackled down by using the average of agreeing reprojected points $\overline{P}$ rather than the original estimated point P (see Figure 5.6). This averaging of the point significantly improves the final results and reduces the amount of noise in the final cloud.

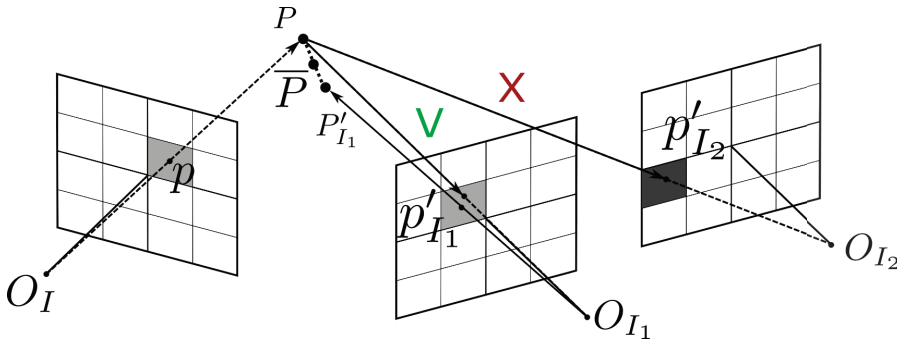


Figure 5.6: Illustration of the multi-depth map validation process. For each upsampled pixel p of a depthmap I, its 3D projection P is reprojected in n other depthmaps. The consolidated point $\overline{P}$ is the average of P and the 3D reprojection P'_{I_n} of the pixel p'_{I_n} that verify $|||P - O_{I_n}|| - d_{p'_{I_n}}| < \varpi$.

This validation step is applied independently at each scale, to remove the upsampled outliers.

(a) Before density improvement. (b) After density improvement.

Figure 5.7: Comparison before and after a density improvement (up to scale $1/2$) of a point cloud projected on an image plane as a depth map.

5.3 Preliminary results and discussion

In the following examples, the results of the direct application of the proposed process are presented. In all these examples, the upsampling was limited in a 7×7 neighborhood, with $\sigma_d = 2.0$, $\sigma_c = 4.66$ and $\varpi = 0.1m$.

A first test was performed in an area that contained a high number of partial structures, important depth variation and occlusion gaps which lie far away from the camera. The original point cloud on which it is applied contains 15 million points, and 70 images were used to upsample the point cloud up to a $1/2$ scale.

The result is a point cloud that contains 15 more million points for a total of 30 million. The close range partial structures (posts) are upsampled and now possess enough points to be properly identifiable as such in the point cloud. The difference of density is visible when comparing the reprojected depth map of the new point cloud, as visible in Figure 5.7.

The global amount of noise in the upsampled cloud is within the accepted tolerance defined by ϖ. Indeed, as visible in Figure 5.8, more than 98% of the new points are within $0.1m$ of their neighbor in the initial point cloud. This measure only partially accounts for the success of the upsampling since points in occlusion gaps or on partial structures have by definition no close neighbor in the original point cloud. The notable exception are the points located on the edges of the buildings that have been added far from all previously existing points. An observation on a more limited subset of the cloud (see Figure 5.9), located around the posts, shows that most of the noise on the ground is inferior to $0.025m$.

The use of several images to validate the points successfully removed the smooth artifact problems, as shown in Figure 5.10. The outliers that were added by only one depth map (Figure 5.3(a)) are removed during the validation process when compared to the data from other depth maps (Figure 5.3(b)).

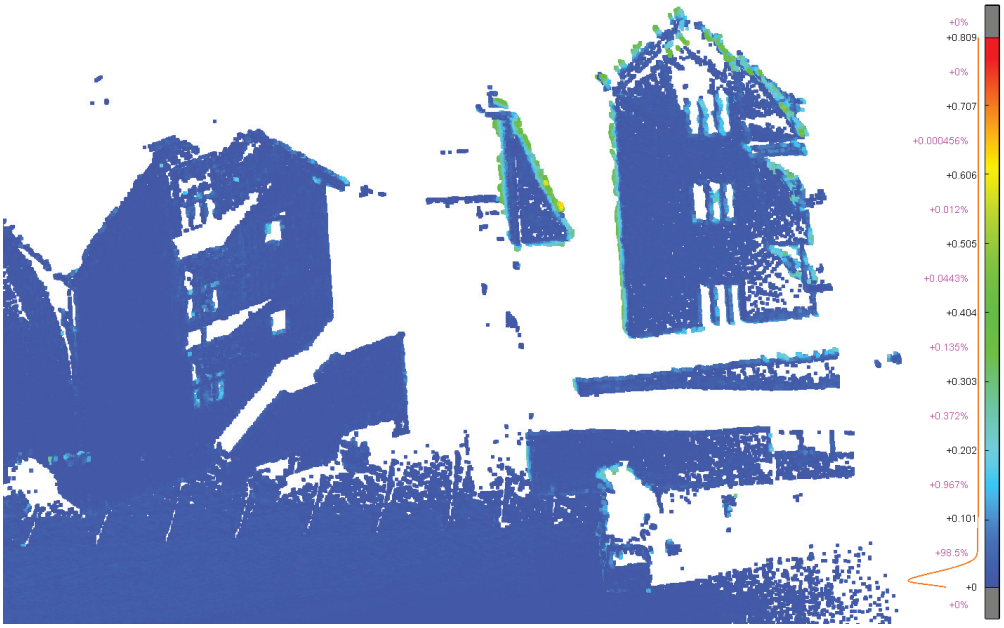


Figure 5.8: Visualization of the distance between the new upsampled points and their closest neighbors in the original point cloud.

Figure 5.11 shows a point cloud with an addition of 14 million points starting from 1 million points and using 40 images. This example also shows that elements such as the sidewalk conserve an acceptable accuracy where small but brutal depth changes are not completely oversmoothed.

The method presented in this section uses color images coupled to an existing point cloud in a simple way to improve the density of this cloud. This density improvement and the noise it creates can be controlled by an accuracy parameter. The proposed method improves the density of the point cloud and fills certain partial structures while retaining a proper geometric accuracy. However, certain areas are not upsampled properly. This is mostly the case on undefined point cloud areas created by occlusion shadows. Another problem that is not addressed by the method is the widening of the point cloud near the edges of the building. These points are due to residual registration errors, that produced small colorization errors, most notably on the edges of the roofs. These registration errors lead to the projection of building points in inconsistent image areas (such as the sky). The assignation of a building point with a sky pixel and a sky color will for instance lead to the assignation of the building depth to the sky pixels. The density of the upsampled point cloud is also not consistent. Obviously, a higher density will be observed closer to the image positions. The introduction of an average consolidated point also introduces the side effects that the newly added points tend to "flock" to the same areas. This behavior produces mottled areas (visible in Figure 5.9 and Figure 5.10), but any standard density equalization method should improve the result.

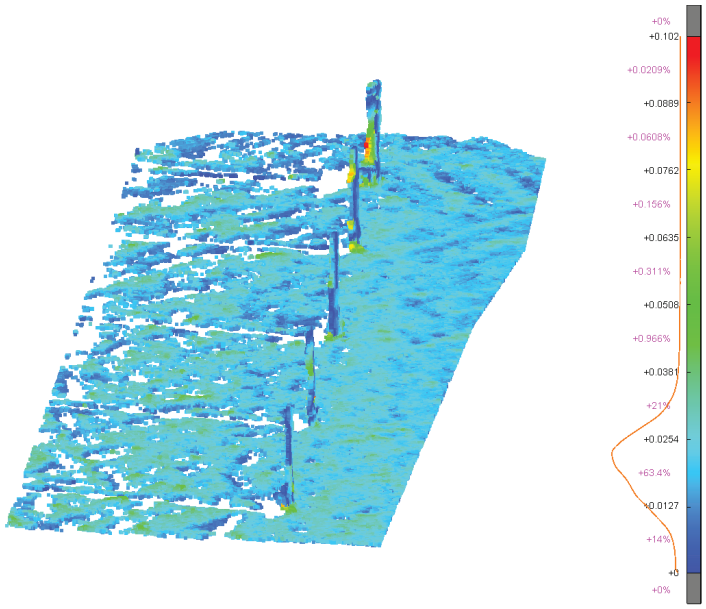
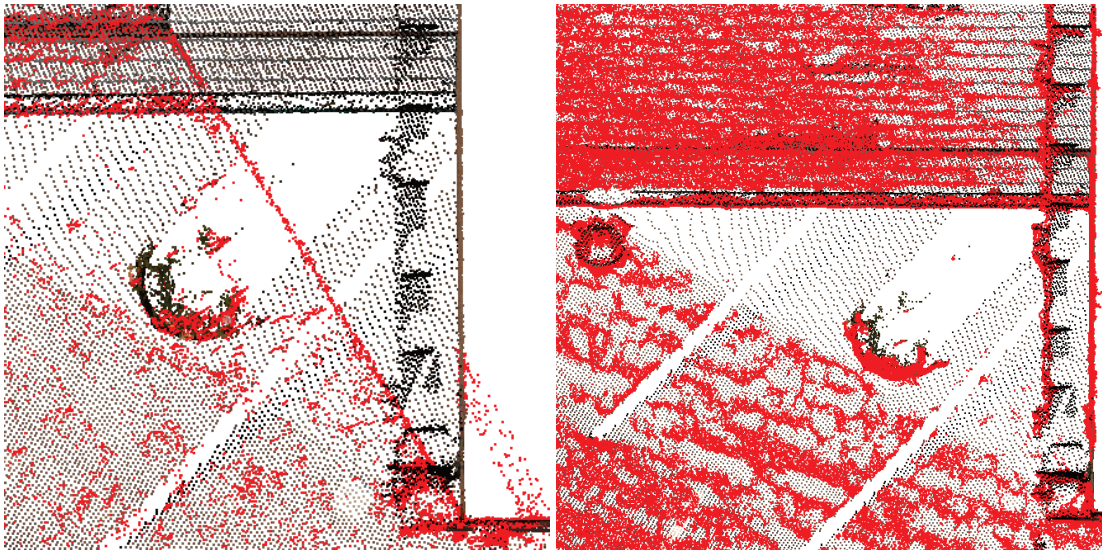


Figure 5.9: Details of the distance visualization between the new upsampled points and their closest neighbors in the original point cloud.



(a) Point cloud (top view) with a single depth map. (b) Point cloud (top view) with multiple depth maps.

Figure 5.10: Comparison between a single depth map point addition (5.3(a)) and multiple depth maps point additions with a validation step 5.3(b). The newly added points appear in red. The diagonal of outliers appearing in Figure 5.3(a) is successfully removed by the validation process.

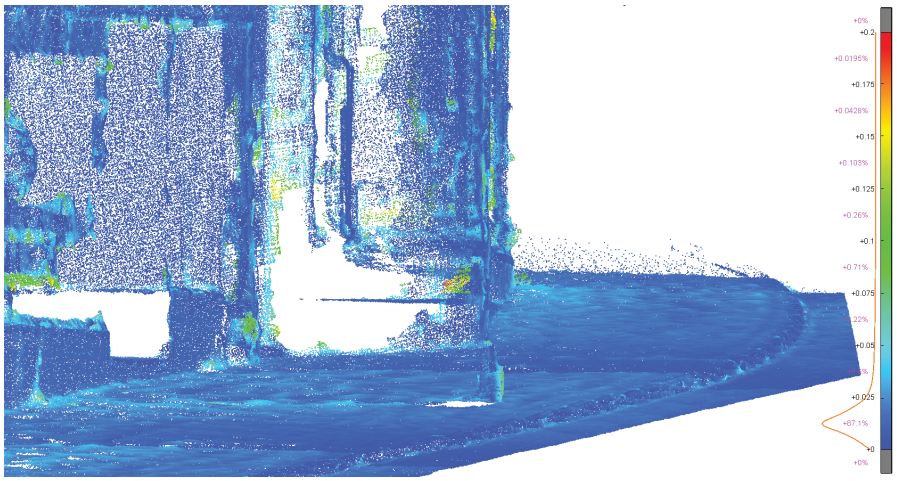


Figure 5.11: Visualization of the distance between the new upsampled points and their closest neighbors in the original point cloud.

CONCLUSION

Summary

This thesis demonstrated that the combined use of both images and point clouds was an advantage to handle large scale urban data. Throughout this thesis, the different steps necessary to jointly use images and point clouds and some of their potential applications for data improvement were explored. The synthetic image generation process is crucial to obtain image representations of a point cloud that can be compared to a real photography. Several possible image generation processes are possible, but the one described in this work is a good compromise between complexity and result quality. An image to geometry registration method was introduced, building on a novel image comparison metric. The metric and the proposed registration method are more robust to sparse and missing information, especially when no reflection information is available with the point cloud. An improvement of an existing colorization method was proposed by minimizing an energy based on the median color given by several images. The proposed colorization method is compatible with large numbers of images and gives an adequate and consistent color to each point of a cloud with a high level of detail. In the wake of this colorization method, a new process to automatically detect and correct shadows in a colored point cloud was introduced. The presented algorithm successfully locates and removes the cast shadows on scenes that present strong direct illumination from the sun. Finally possible directions for the combined use of images and point clouds to improve the geometrical information were explored. This preliminary work outlines the strong possibilities to combine information from multiple images to improve a single point cloud. Among them, a method was presented to successfully increase the point cloud density and complete certain partially acquired surfaces.

Limitations and perspectives

While several problems were treated by the combined use of image and point clouds throughout this thesis, some improvements can still be done in a significant number of domains. Indeed, the combined use of different types of data to improve each other was only partially grazed in the presented works and can be expended.

The image to geometry registration method could be improved in several ways. One possible improvement would be to adapt its principle to re-adjust the camera calibration and distortions parameters, that may be sightly off. A better fitting model would lead to more precise results, and a more accurate registration. Another possible improvement would be to adapt the method in

order to use it on GP-GPU[1], to further accelerate the process. The use of a spherical image could also improve the registration time, but it would imply to redesign almost completely the synthetic image generation process explained in chapter 2.

The colorization process described in section 4.1 produces coherent colors. However it is computationally expensive. This long processing time could be drastically reduced by considering smaller point cloud sections, and parallelizing the coloring process on this cluster. Another improvement that can be considered is the use of the laser reflectance data, if available, to guide the colorization. Indeed, if the laser intensity value cannot be used directly for colorization process, the gradients of reflectance intensity are similar to the color gradients. This information was not exploited in this work but could improve the final quality of the colorization and help to refine the contrast in difficult regions, such as the ground.

The shadow detection and correction capabilities presented in chapter 4 can be extended to correct the illumination in the images themselves. Indeed, as presented, the method can only be used to correct the illumination and remove the cast shadows for each point of a colored cloud. An even more useful application would be to extend this process to the registered images. By using the proposed approach and re-projecting the information on the pixels, a shadow-free image can be obtained. The illumination model presented in chapter 4 for shadow correction can also be greatly improved by considering a localized version of the illumination. Indeed, the shadow characteristics are considered uniform along the entire point cloud. However, since all the acquisition is not performed at the same time, the shadow properties can change in the different regions of the point cloud. Having a local characterization of the shadow properties could improve the results presented in section 4.2.4. Similarly, the current proposed illumination model does not take into account indirect illumination and the contribution of environmental light reflections in the final color aspect. This indirect illumination is however a crucial part of the perceived color of a point, especially in urban environment where large amounts of furniture are scattered in the scene. Thus, the addition of the indirect lighting in the model would further improve the results, and fix the remaining artifacts that appear after the shadow removal process.

The point cloud geometric improvement using multiple images was briefly addressed in chapter 5. A further consolidation of the density improvement model presented in this chapter is possible. For instance, the effect of using a geodesic upsample scheme rather than bilateral or trilateral filter has to be studied. Another limitation of this approach is the amount and dependency of the parameter, which should also be addressed.

These potential improvements are built directly on the work presented in this thesis. However, other applications of the combined use of point cloud and images are possible. Among these potentially efficient applications, the automatic detection and removal of occluding objects such as urban furniture or pedestrians can be greatly facilitated by the use of images in conjunction with point cloud. Indeed, the automatic detection of elements in the point cloud can be facilitated by the inconsistencies they produce in the images. In a similar way, the acquisition holes created by the occluding objects could be more easily filled by using the information contained in the images.

[1] General-purpose computing on graphics processing units

Finally, the powerful segmentation tools that have been developed in the image domain can be transferred to the point cloud. Indeed, by transferring the results of images segmentation algorithm on the point cloud, problems of automatically detecting and segmenting pedestrians, cars and urban furniture, is greatly simplified. Such segmentation could have beneficial effects on the colorization and the densification method discussed in this thesis.

BIBLIOGRAPHY

[Abd09] Ahmed ABDELHAFIZ. *Integrating digital photogrammetry and terrestrial laser scanning*. Techn. Univ., Inst. für Geodäsie und Photogrammetrie, 2009 (cited on page 79).

[AMCO08] Dror AIGER, Niloy J MITRA, and Daniel COHEN-OR. "4-points congruent sets for robust pairwise surface registration". In: *ACM Transactions on Graphics (TOG)*. Volume 27. 3. ACM. 2008, page 85 (cited on pages 28, 52).

[AH04] Yahya ALSHAWABKEH and Norbert HAALA. "Integration of digital photogrammetry and laser scanning for heritage documentation". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 35 (2004), B5 (cited on page 77).

[Ari+13] Murat ARIKAN, Michael SCHWÄRZLER, Simon FLÖRY, Michael WIMMER, and Stefan MAIERHOFER. "O-snap: Optimization-based snapping for modeling architecture". In: *ACM Transactions on Graphics (TOG)* 32.1 (2013), page 6 (cited on page 23).

[Bas+14] Paola BASTONERO, Elisabetta DONADIO, Filiberto CHIABRANDO, and A SPANÒ. "Fusion of 3D models derived from TLS and image-based techniques for CH enhanced documentation". In: *INTERNATIONAL ARCHIVES OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES* 2 (2014), pages 73–80 (cited on page 77).

[BTVG06] Herbert BAY, Tinne TUYTELAARS, and Luc VAN GOOL. "Surf: Speeded up robust features". In: *Computer vision–ECCV 2006*. Springer, 2006, pages 404–417 (cited on pages 24, 50).

[Bev+17] Marco BEVILACQUA, Jean-François AUJOL, Pierre BIASUTTI, Mathieu BRÉDIF, and Aurélie BUGEAU. "Joint inpainting of depth and reflectance with visibility estimation". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 125 (2017), pages 16–32 (cited on page 110).

[Bia+17] Pierre BIASUTTI, Jean-François AUJOL, Mathieu BRÉDIF, and Aurélie BUGEAU. "Diffusion and inpainting of reflectance and height LiDAR orthoimages". working paper or preprint. Oct. 2017. URL: `https://hal.archives-ouvertes.fr/hal-01322822` (cited on page 110).

[Bil+15] Filip BILJECKI, Jantien STOTER, Hugo LEDOUX, Sisi ZLATANOVA, and Arzu ÇÖLTEKIN. "Applications of 3D city models: State of the art review". In: *ISPRS International Journal of Geo-Information* 4.4 (2015), pages 2842–2889 (cited on page 22).

[Bon+17] Nicolas BONNEEL, Balazs KOVACS, Sylvain PARIS, and Kavita BALA. "Intrinsic decompositions for image editing". In: *Computer Graphics Forum*. Volume 36. 2. Wiley Online Library. 2017, pages 593–609 (cited on page 92).

[Bot+05] Mario BOTSCH, Anja HORNUNG, Matthias ZWICKE, and Kif KOBBELT. "High-quality surface splatting on today's GPUs". In: *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*. IEEE. 2005, pages 17–141 (cited on page 46).

[BK04] Yuri BOYKOV and Vladimir KOLMOGOROV. "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision." In: *IEEE transactions on Pattern Analysis and Machine Intelligence* 26.9 (2004), pages 1124–1137 (cited on page 96).

[BVZ01] Yuri BOYKOV, Olga VEKSLER, and Ramin ZABIH. "Efficient Approximate Energy Minimization via Graph Cuts". In: *IEEE transactions on Pattern Analysis and Machine Intelligence* 20.12 (2001), pages 1222–1239 (cited on page 96).

[Bro66] Duane C BROWN. "Decentering distortion of lenses". In: *Photometric Engineering* 32.3 (1966), pages 444–462 (cited on pages 39, 41).

[BWG15] Mark BROWN, David WINDRIDGE, and Jean-Yves GUILLEMAUT. "Globally Optimal 2D-3D Registration from Points or Lines Without Correspondences". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pages 2111–2119 (cited on page 51).

[Buy+17] P. BUYSSENS, O. L. MEUR, M. DAISY, D. TSCHUMPERLÉ, and O. LÉZORAY. "Depth-Guided Disocclusion Inpainting of Synthesized RGB-D Images". In: *IEEE Transactions on Image Processing* 26.2 (2017), pages 525–538. ISSN: 1057-7149. DOI: 10.1109/TIP.2016.2619263 (cited on page 110).

[Cal+10] Michael CALONDER, Vincent LEPETIT, Christoph STRECHA, and Pascal FUA. "Brief: Binary robust independent elementary features". In: *Computer Vision–ECCV 2010* (2010), pages 778–792 (cited on page 24).

[Che+02] Dmitry CHETVERIKOV, Dmitry SVIRKO, Dmitry STEPANOV, and Pavel KRSEK. "The trimmed iterative closest point algorithm". In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Volume 3. IEEE. 2002, pages 545–548 (cited on page 28).

[Cho+14] Sunyoung CHO, Jizhou YAN, Yasuyuki MATSUSHITA, and Hyeran BYUN. "Efficient Colorization of Large-Scale Point Cloud Using Multi-pass Z-Ordering". In: *2014 2nd International Conference on 3D Vision*. Volume 1. IEEE. 2014, pages 689–696 (cited on pages 80–83, 85, 86, 88–90).

[Cor+13a] Peter CORKE, Rimi PAUL, Winston CHURCHILL, and Paul NEWMAN. "Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, pages 2085–2092 (cited on pages 92–94).

[Cor+09] Massimiliano CORSINI, Matteo DELLEPIANE, Federico PONCHIO, and Roberto SCOPIGNO. "Image-to-Geometry Registration: a Mutual Information Method exploiting Illumination-related Geometric Properties". In: *Computer Graphics Forum* 28.7 (Oct. 2009), pages 1755–1764. ISSN: 01677055 (cited on page 51).

[Cor+13b] Massimiliano CORSINI, Matteo DELLEPIANE, Fabio GANOVELLI, Riccardo GHERARDI, Andrea FUSIELLO, and Roberto SCOPIGNO. "Fully automatic registration of image sets on approximate geometry". In: *International journal of computer vision* 102.1-3 (2013), pages 91–111 (cited on page 52).

[CCM13] Nathan CROMBEZ, Guillaume CARON, and El Mustapha MOUADDIB. "Colorisation photo-réaliste de nuages de points 3D". In: *Orasis, Congrès des jeunes chercheurs en vision par ordinateur*. 2013 (cited on page 79).

[DT05] Navneet DALAL and Bill TRIGGS. "Histograms of oriented gradients for human detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Volume 1. IEEE. 2005, pages 886–893 (cited on page 56).

[Dav11] Timothy A DAVIS. "Algorithm 915, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization". In: *ACM Transactions on Mathematical Software (TOMS)* 38.1 (2011), page 8 (cited on page 83).

[Dec+17] Clément DECHESNE, Clément MALLET, Arnaud LE BRIS, and Valérie GOUET-BRUNET. "HOW TO COMBINE LIDAR AND VERY HIGH RESOLUTION MULTISPECTRAL IMAGES FOR FOREST STAND SEGMENTATION?" In: *Proc. of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (2017) (cited on page 29).

[Dol+10] Jennifer DOLSON, Jongmin BAEK, Christian PLAGEMANN, and Sebastian THRUN. "Upsampling range data in dynamic environments". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pages 1141–1148 (cited on page 110).

[DSG16] Gilad DROZDOV, Yevgeny SHAPIRO, and Guy GILBOA. "Robust Recovery of Heavily Degraded Depth Measurements". In: *2016 Fourth International Conference on 3D Vision*. 2016 (cited on page 110).

[EH+11] Sherif EL-HALAWANY, Adel MOUSSA, Derek D LICHTI, and Naser EL-SHEIMY. "Detection of road curb from mobile terrestrial laser scanner point cloud". In: *Proceedings of the ISPRS Workshop on Laserscanning, Calgary, Canada*. Volume 2931. 2011 (cited on page 21).

[FFG09] Michela FARENZENA, Andrea FUSIELLO, and Riccardo GHERARDI. "Structure-and-motion pipeline on a hierarchical cluster tree". In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE. 2009, pages 1489–1496 (cited on page 24).

[FD+14] Juan Carlos FERNANDEZ-DIAZ, Craig L GLENNIE, William E CARTER, Ramesh L SHRESTHA, Michael P SARTORI, Abhinav SINGHANIA, Carl J LEGLEITER, and Brandon T OVERSTREET. "Early results of simultaneous terrain and shallow water bathymetry mapping using a single-wavelength airborne LiDAR sensor". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.2 (2014), pages 623–635 (cited on page 27).

[Fio+12] Torsten FIOLKA, Jörg STÜCKLER, Dominik KLEIN, Dirk SCHULZ, and Sven BEHNKE. "SURE: Surface Entropy for Distinctive 3D Features". In: *Spatial Cognition VIII*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2012, pages 74–93 (cited on page 24).

[FB81] Martin A FISCHLER and Robert C BOLLES. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6 (1981), pages 381–395 (cited on pages 24, 50).

[FP10] Yasutaka FURUKAWA and Jean PONCE. "Accurate, dense, and robust multiview stereopsis." In: *IEEE transactions on pattern analysis and machine intelligence* 32.8 (Aug. 2010), pages 1362–76. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2009.161. URL: http://www.ncbi.nlm.nih.gov/pubmed/20558871 (cited on page 24).

[Gei+13] Andreas GEIGER, Philip LENZ, Christoph STILLER, and Raquel URTASUN. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* 32.11 (2013), pages 1231–1237 (cited on pages 28–30, 33, 70, 100).

[Gon+14] Maoguo GONG, Shengmeng ZHAO, Licheng JIAO, Dayong TIAN, and Shuang WANG. "A novel coarse-to-fine scheme for automatic image registration based on SIFT and mutual information". In: *Geoscience and Remote Sensing, IEEE Transactions on* 52.7 (2014), pages 4328–4338 (cited on pages 51, 55).

[GARGGL09] Diego GONZÁLEZ-AGUILERA, Pablo RODRÍGUEZ-GONZÁLVEZ, and Javier GÓMEZ-LAHOZ. "An automatic procedure for co-registration of terrestrial laser scanners and digital cameras". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 64.3 (2009), pages 308–316 (cited on pages 46, 50).

[GP12] Gerhard GRÖGER and Lutz PLÜMER. "CityGML–Interoperable semantic 3D city models". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 71 (2012), pages 12–33 (cited on page 23).

[Gui+16a] M. GUISLAIN, J DIGNE, R. CHAINE, D. KUDELSKI, and P. LEFEVBRE-ALBARET. "Detecting and Correcting Shadows in Urban Point Clouds and Image Collections". In: *International Conference on 3D Vision*. Stanford University, California, USA, 2016 (cited on page 141).

[Gui+16b] Maximilien GUISLAIN, Julie DIGNE, Raphaëlle CHAINE, and Gilles MONNIER. "Recalage d'image dans des nuages de points de scènes urbaines". In: *Actes des Journées du Groupe de Travail en Modélisation Géométrique 2016*. Marc Neveu, Sandrine Lanquetin, Christian Gentil, Lionel Garnier. Dijon, France, Mar. 2016. URL: https://hal.archives-ouvertes.fr/hal-01320263 (cited on page 141).

[Gui+17] Maximilien GUISLAIN, Julie DIGNE, Raphaëlle CHAINE, and Gilles MONNIER. "Fine scale image registration in large-scale urban LIDAR point sets". In: *Computer Vision and Image Understanding (CVIU)*. Special Issue on Large-Scale 3D Modeling of Urban Indoor or Outdoor Scenes from Images and Range Scans 157 (Apr. 2017), pages 90–102. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2016.12.004. URL: https://hal.archives-ouvertes.fr/hal-01468091 (cited on page 141).

[HCL15] Yuhang HE, Long CHEN, and Ming LI. "Sparse depth map upsampling with RGB image and anisotropic diffusion tensor". In: *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE. 2015, pages 205–210 (cited on page 111).

[Hen+12] Peter HENRY, Michael KRAININ, Evan HERBST, Xiaofeng REN, and Dieter FOX. "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments". In: *The International Journal of Robotics Research* 31.5 (2012), pages 647–663 (cited on page 26).

[HS13] Alexandre HERVIEU and Bahman SOHEILIAN. "Semi-automatic road/pavement modeling using mobile laser scanning". In: *ISPRS Annals* 3 (2013), W3 (cited on page 21).

[HEB14] S HOFMANN, D EGGERT, and C BRENNER. "Skyline matching based camera orientation from images and mobile mapping point clouds". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2014), pages 181–188 (cited on page 52).

[JV11] Bing JIAN and Baba C VEMURI. "Robust point set registration using gaussian mixture models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (2011), pages 1633–1645 (cited on page 28).

[Joh] Steven G. JOHNSON. "The NLopt nonlinear-optimization package". http://ab-initio.mit.edu/nlopt (cited on page 64).

[Jou+12] Bruno JOUVENCEL, Xianbo XIANG, Lei ZHANG, and Zheng FANG. "3D Reconstruction of seabed surface through sonar data of AUVs". In: *Indian Journal of Geo-Marine Sciences (IJMS)* 41.6 (2012), pages 509–515 (cited on page 26).

[KR05] Erum A KHAN and Erik REINHARD. "Evaluation of color spaces for edge classification in outdoor scenes". In: *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. Volume 3. IEEE. 2005, pages III–952 (cited on page 94).

[KE12] Kourosh KHOSHELHAM and Sander Oude ELBERINK. "Accuracy and resolution of kinect depth data for indoor mapping applications". In: *Sensors* 12.2 (2012), pages 1437–1454 (cited on page 26).

[KKZ03] Junhwan KIM, Vladimir KOLMOGOROV, and Ramin ZABIH. "Visual correspondence using energy minimization and mutual information". In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE. 2003, pages 1033–1040 (cited on pages 51, 53).

[Kim+13] Kichang KIM, Takayuki SUGIURA, Akihiko TORII, Shigeki SUGIMOTO, and Masatoshi OKUTOMI. "Instant surface reconstruction for incremental SfM". In: *Proceedings of the IAPR Conference on Machine Vision Applications (MVA 2013)*. 2013, pages 371–374 (cited on page 24).

[KVD91] Jan J KOENDERINK and Andrea J VAN DOORN. "Affine structure from motion". In: *Journal of the Optical Society of America, A, Optics, Image & Science* 8.2 (1991), pages 377–385 (cited on page 24).

[Kop+07] Johannes KOPF, Michael F COHEN, Dani LISCHINSKI, and Matt UYTTENDAELE. "Joint bilateral upsampling". In: *ACM Transactions on Graphics (TOG)*. Volume 26. 3. ACM. 2007, page 96 (cited on pages 110, 113).

[KHP] Michael KORN, Martin HOLZKOTHEN, and Josef PAULI. "Color Supported Generalized-ICP 0". In: *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*. Volume 3, pages 592–599 (cited on page 77).

[KK17] Aarno Tapio KOTILAINEN and Anu Marii KASKELA. "Comparison of airborne LiDAR and shipboard acoustic data in complex shallow water environments: Filling in the white ribbon zone". In: *Marine Geology* 385 (2017), pages 250–259 (cited on page 27).

[Kus+14] Claudia KUSTER, Jean-Charles BAZIN, Cengiz ÖZTIRELI, Teng DENG, Tobias MARTIN, Tiberiu POPA, and Markus GROSS. "Spatio-temporal geometry fusion for multiple hybrid cameras using moving least squares surfaces". In: *Computer Graphics Forum* 33.2 (May 2014), pages 1–10. ISSN: 01677055. DOI: 10.1111/cgf.12285. URL: http://doi.wiley.com/10.1111/cgf.12285 (cited on page 77).

[LBD12] Pierre-Yves LAFFONT, Adrien BOUSSEAU, and George DRETTAKIS. "Rich Intrinsic Image Decomposition of Outdoor Scenes from Multiple Views." In: *IEEE transactions on visualization and computer graphics* XX.X (Apr. 2012), pages 1–16. ISSN: 1941-0506. DOI: 10.1109/TVCG.2012.112. URL: http://www.ncbi.nlm.nih.gov/pubmed/22508899 (cited on pages 93, 97).

[LEN10] Jean-François LALONDE, Alexei A EFROS, and Srinivasa G NARASIMHAN. "Detecting ground shadows in outdoor consumer photographs". In: *Computer Vision–ECCV 2010*. Springer, 2010, pages 322–335 (cited on page 92).

[LMNF09] Vincent LEPETIT, Francesc MORENO-NOGUER, and Pascal FUA. "Epnp: An accurate o (n) solution to the pnp problem". In: *International journal of computer vision* 81.2 (2009), pages 155–166 (cited on page 50).

[Ler+10] José Luis LERMA, Santiago NAVARRO, Miriam CABRELLES, and Valentín VILLAVERDE. "Terrestrial laser scanning and close range photogrammetry for 3D archaeological documentation: the Upper Palaeolithic Cave of Parpalló as a case study". In: *Journal of Archaeological Science* 37.3 (2010), pages 499–507 (cited on page 77).

[Lhu08] Maxime LHUILLIER. "Automatic scene structure and camera motion using a catadioptric system". In: *Computer Vision and Image Understanding* 109.2 (2008), pages 186–203 (cited on page 24).

[Lim11] Ee Hui LIM. "3D Urban Modelling". PhD thesis. Monash University Clayton Victoria 3800 Australia: Monash University, 2011 (cited on page 80).

[LTT13] Ming-Yu LIU, Oncel TUZEL, and Yuichi TAGUCHI. "Joint geodesic upsampling of depth images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pages 169–176 (cited on page 111).

[Low04] David G. LOWE. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pages 91–110. ISSN: 0920-5691 (cited on pages 24, 50).

[MKF09] Andrew MASTIN, Jeremy KEPNER, and Jonathan FISHER. "Automatic registration of LIDAR and optical images of urban scenes". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pages 2639–2646 (cited on pages 51, 62, 70).

[MY09] Jean-Michel MOREL and Guoshen YU. "ASIFT: A new framework for fully affine invariant image comparison". In: *SIAM Journal on Imaging Sciences* 2.2 (2009), pages 438–469 (cited on pages 24, 50).

[Mou14] Wassim MOUSSA. "Integration of digital photogrammetry and terrestrial laser scanning for cultural heritage data recording". PhD thesis. University of Stuttgart, 2014 (cited on pages 27, 52, 77).

[MAWF12] Wassim MOUSSA, Mohammed ABDEL-WAHAB, and Dieter FRITSCH. "An Automatic Procedure for Combining Digital Images and Laser Scanner Data". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 39 (2012), B5 (cited on pages 29, 50).

[Pan+12] Gaurav PANDEY, James R MCBRIDE, Silvio SAVARESE, and Ryan EUSTICE. "Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information." In: *AAAI*. 2012 (cited on pages 51, 71).

[Pap+12] Nicolas PAPARODITIS, Jean-Pierre PAPELARD, Bertrand CANNELLE, Alexandre DEVAUX, Bahman SOHEILIAN, Nicolas DAVID, and Erwann HOUZAY. "Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology". In: *Revue française de photogrammétrie et de télédétection* 200.1 (2012), pages 69–79 (cited on page 29).

[PMN15] Geoffrey PASCOE, Will MADDERN, and Paul NEWMAN. “Robust Direct Visual Localisation using Normalised Information Distance”. In: *British Machine Vision Conference (BMVC), Swansea, Wales*. Volume 3. 2015, page 4 (cited on page 52).

[Pea01] Karl PEARSON. “LIII. On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pages 559–572 (cited on page 43).

[PGA11] Ruggero PINTUS, Enrico GOBBETTI, and Marco AGUS. “Real-time rendering of massive unstructured raw point clouds using screen-space operators”. In: *VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*. Edited by Franco NICCOLUCCI, Matteo DELLEPIANE, Sebastian Pena SERNA, Holly RUSHMEIER, and Luc Van GOOL. The Eurographics Association, 2011. ISBN: 978-3-905674-34-7 (cited on page 44).

[PR15] Tobias PLOTZ and Stefan ROTH. “Registering Images to Untextured Geometry using Average Shading Gradients”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pages 2030–2038 (cited on page 51).

[Pow09] Michael JD POWELL. *The BOBYQA algorithm for bound constrained optimization without derivatives*. Technical report DAMTP 2009/NA06. University of Cambridge, 2009 (cited on pages 62, 63).

[Pre+16] Cristiano PREMEBIDA, Luis GARROTE, Alireza ASVADI, A Pedro RIBEIRO, and Urbano NUNES. “High-resolution LIDAR-based depth mapping using bilateral filter”. In: *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE. 2016, pages 2469–2474 (cited on page 109).

[RNS15] Rishi RAMAKRISHNAN, Juan NIETO, and Steve SCHEDING. “Shadow compensation for outdoor perception”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pages 4835–4842 (cited on pages 93, 97, 98).

[RA04] Ibrahim REDA and Afshin ANDREAS. “Solar position algorithm for solar radiation applications”. In: *Solar energy* 76.5 (2004), pages 577–589 (cited on page 98).

[Rem11] Fabio REMONDINO. “Heritage recording and 3D modeling with photogrammetry and 3D scanning”. In: *Remote Sensing* 3.6 (2011), pages 1104–1138 (cited on page 77).

[Rem+14] Fabio REMONDINO, Maria Grazia SPERA, Erica NOCERINO, Fabio MENNA, and Francesco NEX. “State of the art in high density image matching”. In: *The Photogrammetric Record* 29.146 (June 2014), pages 144–166. ISSN: 0031868X. DOI: 10.1111/phor.12063. URL: http://doi.wiley.com/10.1111/phor.12063 (cited on page 24).

[Rön+16] Petri RÖNNHOLM, Xinlian LIANG, Antero KUKKO, Anttoni JAAKKOLA, and Juha HYYPPÄ. "QUALITY ANALYSIS AND CORRECTION OF MOBILE BACKPACK LASER SCANNING DATA." In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* (2016), pages 41–47 (cited on page 28).

[RP+17] Joan R ROSELL-POLO, Eduard GREGORIO, Jordi GENÉ, Jordi LLORENS, Xavier TORRENT, Jaume ARNO, and Alexandre ESCOLA. "Kinect v2 sensor-based mobile terrestrial laser scanner for agricultural outdoor applications". In: *IEEE/ASME Transactions on Mechatronics* (2017) (cited on page 26).

[Rot+12] Mathias ROTHERMEL, Konrad WENZEL, Dieter FRITSCH, and Norbert HAALA. "Sure: Photogrammetric surface reconstruction from imagery". In: *Proceedings LC3D Workshop, Berlin*. Volume 8. 2012 (cited on page 24).

[Rub+11] Ethan RUBLEE, Vincent RABAUD, Kurt KONOLIGE, and Gary BRADSKI. "ORB: An efficient alternative to SIFT or SURF". In: *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE. 2011, pages 2564–2571 (cited on page 24).

[SM13] Andrés SERNA and Beatriz MARCOTEGUI. "Urban accessibility diagnosis from mobile laser scanning data". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 84 (2013), pages 23–32 (cited on page 22).

[SW10] S SHAHZAD and M WIGGENHAGEN. "Co-registration of terrestrial laser scans and close range digital images using scale invariant features". In: *Allgemeine Vermessungs-Nachrichten* 117.6 (2010), pages 208–212 (cited on page 50).

[SWD05] Gaurav SHARMA, Wencheng WU, and Edul N DALAL. "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations". In: *Color Research & Application* 30.1 (2005), pages 21–30 (cited on page 94).

[Shr+11] Abhinav SHRIVASTAVA, Tomasz MALISIEWICZ, Abhinav GUPTA, and Alexei a. EFROS. "Data-driven visual similarity for cross-domain image matching". In: *ACM Transactions on Graphics* 30.6 (Dec. 2011), page 1. ISSN: 07300301 (cited on page 56).

[Sib+13] Dominik SIBBING, Torsten SATTLER, Bastian LEIBE, and Leif KOBBELT. "Sift-realistic rendering". In: *3DTV-Conference, 2013 International Conference on*. IEEE. 2013, pages 56–63 (cited on page 29).

[Sin+08] Sudipta N SINHA, Drew STEEDLY, Richard SZELISKI, Maneesh AGRAWALA, and Marc POLLEFEYS. "Interactive 3D architectural modeling from unordered photo collections". In: *ACM Transactions on Graphics (TOG)*. Volume 27. 5. ACM. 2008, page 159 (cited on page 23).

[SSS06] Noah SNAVELY, Steven M. SEITZ, and Richard SZELISKI. "Photo tourism: Exploring photo collections in 3D". In: *ACM transactions on graphics (TOG)*. Volume 25. 3. New York, NY, USA: ACM Press, 2006, pages 835–846. ISBN: 1-59593-364-6 (cited on page 23).

[SHH99] Colin STUDHOLME, Derek LG HILL, and David J HAWKES. "An overlap invariant entropy measure of 3D medical image alignment". In: *Pattern recognition* 32.1 (1999), pages 71–86 (cited on pages 51, 55).

[TN13] Zachary TAYLOR and Juan NIETO. "Automatic calibration of lidar and camera images using normalized mutual information". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. 2013 (cited on pages 43, 51, 62, 71, 75).

[TNJ13] Zeike TAYLOR, John NIETO, and David JOHNSON. "Automatic calibration of multi-modal sensor systems using a gradient orientation measure". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, pages 1293–1300 (cited on pages 51–53, 63, 71, 75).

[TWS14] PW THEILER, JD WEGNER, and K SCHINDLER. "FAST REGISTRATION OF LASER SCANS WITH 4-POINTS CONGRUENT SETS–WHAT WORKS AND WHAT DOESN'T". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2014), pages 149–156 (cited on page 28).

[Tri+00] Bill TRIGGS, Philip MCLAUCHLAN, Richard HARTLEY, and Andrew FITZGIBBON. "Bundle adjustment—a modern synthesis". In: *Vision algorithms: theory and practice* (2000), pages 153–177 (cited on page 24).

[TA05] Alejandro TROCCOLI and Peter K ALLEN. "Relighting acquired models of outdoor scenes". In: *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*. IEEE. 2005, pages 245–252 (cited on page 93).

[VP15] Bruno VALLET and Jean-Pierre PAPELARD. "Road orthophoto/DTM generation from mobile laser scanning". In: *International Annals of Photogrammetry Remote Sensing and Spatial Information Sciences* 3 (2015), W5 (cited on page 110).

[VLA15] Yannick VERDIE, Florent LAFARGE, and Pierre ALLIEZ. *Lod generation for urban scenes*. Technical report. Association for Computing Machinery, 2015 (cited on page 23).

[VWI97] Paul VIOLA and William M WELLS III. "Alignment by maximization of mutual information". In: *International journal of computer vision* 24.2 (1997), pages 137–154 (cited on page 51).

[Vu+12] Hoang-Hiep VU, Patrick LABATUT, Jean-Philippe PONS, and Renaud KERIVEN. "High accuracy and visibility-consistent dense multiview stereo." In: *IEEE transactions on pattern analysis and machine intelligence* 34.5 (May 2012), pages 889–901. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2011.172. URL: http://www.ncbi.nlm.nih.gov/pubmed/21844631 (cited on page 24).

[WBS15] Scott WEHRWEIN, Kavita BALA, and Noah SNAVELY. "Shadow Detection and Sun Direction in Photo Collections". In: *Proceedings of 3DV*. 2015 (cited on pages 93, 98).

[WCH92] Juyang WENG, Paul COHEN, and Marc HERNIOU. "Camera calibration with distortion models and accuracy evaluation". In: *IEEE Transactions on pattern analysis and machine intelligence* 14.10 (1992), pages 965–980 (cited on page 40).

[Wu+11] Changchang WU et al. "VisualSFM: A visual structure from motion system". 2011 (cited on page 25).

[Xia+13] Chunxia XIAO, Ruiyun SHE, Donglin XIAO, and Kwan-Liu MA. "Fast Shadow Removal Using Adaptive Multi-Scale Illumination Transfer". In: *Computer Graphics Forum*. Volume 32. 8. Wiley Online Library. 2013, pages 207–218 (cited on page 92).

[YBS07] Gehua YANG, Jacob BECKER, and Charles V STEWART. "Estimating the location of a camera with respect to a 3d model". In: *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*. IEEE. 2007, pages 159–166 (cited on page 50).

[Yu+99] Yizhou YU, Paul DEBEVEC, Jitendra MALIK, and Tim HAWKINS. "Inverse global illumination: Recovering reflectance models of real scenes from photographs". In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 1999, pages 215–224 (cited on page 97).

[ZZ14] Fanyang ZENG and Ruofei ZHONG. "The algorithm to generate color point-cloud with the registration between panoramic image and laser point-cloud". In: *IOP Conference Series: Earth and Environmental Science*. Volume 17. 1. IOP Publishing. 2014, page 012160 (cited on page 79).

[ZLX09] Q ZHAN, Yubin LIANG, and Y XIAO. "Color-based segmentation of point clouds". In: *Laser scanning* 38.3 (2009), pages 155–161 (cited on page 77).

[Zha12] Zhengyou ZHANG. "Microsoft kinect sensor and its effect". In: *IEEE multimedia* 19.2 (2012), pages 4–10 (cited on page 26).

[Zlo+14] Robert ZLOT, Michael BOSSE, Kelly GREENOP, Zbigniew JARZAB, Emily JUCKES, and Jonathan ROBERTS. "Efficiently capturing large, complex cultural heritage sites with a handheld mobile 3D laser mapping system". In: *Journal of Cultural Heritage* 15.6 (2014), pages 670–678 (cited on page 28).

[Zwi+01] Matthias ZWICKER, Hanspeter PFISTER, Jeroen VAN BAAR, and Markus GROSS. "Surface splatting". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM. 2001, pages 371–378 (cited on page 46).

CITED AUTHORS INDEX

PUBLICATIONS

International journals

[Gui+17] Maximilien GUISLAIN et al. “Fine scale image registration in large-scale urban LIDAR point sets”. In: *Computer Vision and Image Understanding (CVIU)*. Special Issue on Large-Scale 3D Modeling of Urban Indoor or Outdoor Scenes from Images and Range Scans 157 (Apr. 2017), pages 90–102. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2016.12.004. URL: https://hal.archives-ouvertes.fr/hal-01468091.

International conferences

[Gui+16a] M. GUISLAIN et al. “Detecting and Correcting Shadows in Urban Point Clouds and Image Collections”. In: *International Conference on 3D Vision*. Stanford University, California, USA, 2016.

National conferences

[Gui+16b] Maximilien GUISLAIN et al. “Recalage d’image dans des nuages de points de scènes urbaines”. In: *Actes des Journées du Groupe de Travail en Modélisation Géométrique 2016*. Marc Neveu, Sandrine Lanquetin, Christian Gentil, Lionel Garnier. Dijon, France, Mar. 2016. URL: https://hal.archives-ouvertes.fr/hal-01320263.