Change detection from mobile laser scanning point clouds

Wen Xiao

UNIVERSITÉ PARIS-EST

ÉCOLE DOCTORALE Mathématiques et STIC

IGN INSTITUT NATIONAL DE L'INFORMATION GÉOGRAPHIQUE ET FORESTIÈRE

MATIS

Thèse de doctorat

présentée pour obtenir le grade de docteur
de l'UNIVERSITÉ PARIS-EST

École Doctorale MSTIC

Spécialité : Sciences et Technologies de l'Information Géographique

WEN XIAO

DÉTECTION DE CHANGEMENTS À PARTIR DE NUAGES DE POINTS DE CARTOGRAPHIE MOBILE

Soutenue publiquement le 12 novembre 2015 devant le jury composé de :

George VOSSELMAN	Rapporteur
Laurent TRASSOUDAINE	Rapporteur
Jon MILLS	Président
Beatriz MARCOTEGUI	Examinateur
Michel ROUX	Examinateur
Nicolas PAPARODITIS	Directeur de thèse
Bruno VALLET	Encadrant de thèse

UNIVERSITÉ PARIS-EST

ÉCOLE DOCTORALE UNIVERSITÉ PARIS-EST
Mathématiques et STIC

IGN
INSTITUT NATIONAL
DE L'INFORMATION
GÉOGRAPHIQUE
ET FORESTIÈRE

MATIS

DISSERTATION

presented to obtain the degree of doctor
of the UNIVERSITÉ PARIS-EST

Doctoral School MSTIC

Speciality : Geographic Information Science and Technology

WEN XIAO

CHANGE DETECTION FROM MOBILE LASER SCANNING POINT CLOUDS

Publicly defended on 12th November 2015 in front of the jury composed of:

George VOSSELMAN	Rapporteur
Laurent TRASSOUDAINE	Rapporteur
Jon MILLS	President
Beatriz MARCOTEGUI	Examiner
Michel ROUX	Examiner
Nicolas PAPARODITIS	Thesis director
Bruno VALLET	Thesis supervisor

The research is conducted at laboratory MATIS (Méthodes d'Analyses pour le Traitement d'Images et la Stéréorestitution) of the Geographic Information Science research unit (service de recherche en sciences de l'information géographique) at IGN (Institut National de l'Information Géographique et Forestière).

Laboratoire MATIS
Service de la Recherche
Institut National de l'Information Géographique et Forestière
73 avenue de Paris
94165 Saint-Mandé cedex FRANCE

Téléphone : 01 43 98 80 00

Reference BibTeX:

```
@PHDTHESIS{Xiao2015,
  author = {Xiao, Wen},
  title = {Change Detection from Mobile Laser Scanning Point Clouds},
  school = {Universit\'e Paris-Est},
  year = {2015}
}
```

Résumé

Les systèmes de cartographie mobile sont de plus en plus utilisés pour la cartographie des scènes urbaines. La technologie de scan laser mobile (où le scanner est embarqué sur un véhicule) en particulier permet une cartographie précise de la voirie, la compréhension de la scène, la modélisation de façade, etc. Dans cette thèse, nous nous concentrons sur la détection de changement entre des nuages de points laser de cartographie mobile.

Tout d'abord, nous étudions la détection des changements a partir de données RIEGL (scanner laser plan) pour la mise à jour de bases de données géographiques et l'identification d'objet temporaire. Nous présentons une méthode basée sur l'occupation de l'espace qui permet de surmonter les difficultés rencontrées par les méthodes classiques fondées sur la distance et qui ne sont pas robustes aux occultations et à l'échantillonnage anisotrope. Les zones occultées sont identifiées par la modélisation de l'état d'occupation de l'espace balayé par des faisceaux laser. Les écarts entre les points et les lignes de balayage sont interpolées en exploitant la géométrie du capteur dans laquelle la densité d'échantillonnage est isotrope. Malgré quelques limites dans le cas d'objets pénétrables comme des arbres ou des grilles, la méthode basée sur l'occupation est en mesure d'améliorer la méthode basée sur la distance point à triangle de façon significative.

La méthode de détection de changement est ensuite appliquée à des données acquises par différents scanners laser et à différentes échelles temporelles afin de démontrer son large champs d'application. La géométrie d'acquisition est adaptée pour un scanner dynamique de type Velodyne. La méthode basée sur l'occupation permet alors la détection des objets en mouvement. Puisque la méthode détecte le changement en chaque point, les objets en mouvement sont détectés au niveau des points. Un algorithme de détection et le suivi simultané est proposé afin de retrouver les trajectoires de piétons. Cela permet d'estimer avec précision la circulation des piétons des circulations douces dans les lieux publics.

Les changements peuvent non seulement être détectés au niveau du point, mais aussi au niveau de l'objet. Ainsi nous avons pu étudier les changements entre des voitures stationnées dans les rues à différents moments de la journée afin d'en tirer des statistiques utiles aux gestionnaires du stationnement urbain. Dans ce cas, les voitures sont détectés en premier lieu, puis les voitures correspondantes sont comparées entre des passages à différents moments de la journée. Outre les changements de voitures, l'offre de stationnement et les types de voitures l'utilisant sont également des informations importantes pour la gestion du stationnement. Toutes ces informations sont extraites dans le cadre d'un apprentissage supervisé. En outre, une méthode de reconstruction de voiture sur la base d'un modèle déformable générique ajusté aux données est proposée afin de localiser précisément les voitures. Les paramètres du modèle sont également considérés comme caractéristiques de la voiture pour prendre de meilleures décisions.

Dans cette thèse, certains sujets liés à la détection des changements comme par exemple, suivi, la classification, et la modélisation sont étudiés et illustrés par des applications pratiques. Plus important encore, les méthodes de détection des changements sont appliquées à différentes géométries d'acquisition de données et à de multiples échelles temporelles et au travers de deux stratégies: "bottom-up" (en partant des points) et "top-down" (en partant des objets).

Mots Clés : *Lidar, Détection de changements, Suivi d'objets, Classification, Modélisation*

Abstract

Mobile mapping systems are increasingly used for street environment mapping, especially mobile laser scanning technology enables precise street mapping, scene understanding, façade modelling, etc. In this research, the change detection from laser scanning point clouds is investigated.

First of all, street environment change detection using RIEGL data is studied for the purpose of database updating and temporary object identification. An occupancy-based method is presented to overcome the challenges encountered by the conventional distance-based method, such as occlusion, anisotropic sampling. Occluded areas are identified by modelling the occupancy states within the laser scanning range. The gaps between points and scan lines are interpolated under the sensor reference framework, where the sampling density is isotropic. Even there are some conflicts on penetrable objects, e.g. trees, fences, the occupancy-based method is able to enhance the point-to-triangle distance-based method.

The change detection method is also applied to data acquired by different laser scanners at different temporal-scales with the intention to have wider range of applications. The local sensor reference framework is adapted to Velodyne laser scanning geometry. The occupancy-based method is implemented to detection moving objects. Since the method detects the change of each point, moving objects are detect at point level. As the Velodyne scanner constantly scans the surroundings, the trajectories of moving objects can be detected. A simultaneous detection and tracking algorithm is proposed to recover the pedestrian trajectories in order to accurately estimate the traffic flow of pedestrian in public places.

Changes can be detected not only at point level, but also at object level. The changes of cars parking on street sides at different times are detected to help regulate on-street car parking since the parking duration is limited. In this case, cars are detected in the first place, then they are compared with corresponding ones. Apart from car changes, parking positions and car types are also important information for parking management. All the processes are solved in a supervised learning framework. Furthermore, a model-based car reconstruction method is proposed to precisely locate cars. The model parameters are also treated as car features for better decision making. Moreover, the geometrically accurate models can be used for visualization purposes.

Under the theme of change detection, related topics, e.g. tracking, classification, modelling, are also studied for the reason of practical applications. More importantly, the change detection methods are applied to different data acquisition geometries at multiple temporal-scales. Both bottom-up (point-based) and top-down (object-based) change detection strategies are investigated.

Keywords : *Mobile laser scanning, Lidar, Change detection, Moving object detection, Object tracking, Classification, Modelling*

Acknowledgements

Time flies. It is been more than three years already, I have done nothing but the PhD. Well, I have also travelled many places in Europe. But my French is still far from good. I am still not really fitted into the society. There are still so many amazing places to visit. The sad thing is, I am "old" already. Time also crawls. I am looking so much forward to the defence and so ready to finish these stressful days. Going back to see my family and friends only once a year is killing me. And I cannot wait for the new job.

Fist of all, my deepest gratitude goes to Nicolas Paparoditis and Bruno Vallet for recruiting me for such a rewarding but at the meantime challenging work. Nicolas is always encouraging and supportive. He gives me a lot of freedom to decide what to do, and he is always supporting our decisions. Whenever I have a request, he is always willing to help. His appreciation gives me a lot of confidence, even though sometimes I know my work is not that "excellent". This is very important for me. Bruno is such a great tutor. His knowledge is way beyond my imagination. I have learned a lot from him, and I am sure there are much more to learn. It is been a great pleasure to work with him. He is also a great team mate. I enjoyed playing Ultimate with him. Actually, we just had a game before I am writing this, and we won again, as usual.

I would like to thank all the jury members for accepting the invitations and for make themselves available for the defence. It is my great honour to have them in the jury, and I hope they enjoyed reading the manuscript. Specially I would like to thank George Vosselman, who was my MSc supervisor. Thank you for the recommendations.

Thank all the MATIS members, who are all very kind and helpful to me. Thank Valérie for the supports. Thank Mathieu and Clément for helping my work. Thank Marie-Cloud and Marie-Jean for bearing with my terrible French. Thank David for constantly helping me fix my computer problems. Thank Jean-Pascal who helped my a lot, especially when I just arrived. Thank Fabrice and Miloud for sharing the office with me, and thank Jérome for giving me the table. Thank Murat for sharing the wonderful road trip, and the terrible prefecture experience. Thank Alexandre for teaching me Backgammon. Thank Adrien for answering my silly questions. Thank Jean-Pierre, Nicolas, Emmanuel for data acquisition and preprocessing. Thank Neel, Quoc-Dinh, Murat for the exotic food. Thank my Chinese fellows, Yuchu, Xiaozhi and Lijun who make me feel less homesick, especially when they cook nice Chinese meal. Thank all the Ultimate warriors, all the gym tough guys. Thank Isabelle, Corina, Alexandre, Antonio, Athanasios, Arnaud, Bahman, Bertrand, Clémen, David, Fabien, Jean-Christophe, Kamel, Laman, Laurent, Lionel, Narut, Rémi, Sébastien for sharing the great time with me.

Thanks also go to the staff from UPE, even which is still no recognized in China. Thank Sylvie for the administrative help. Thank the acc&ss team for solving my ID problems and providing french courses. I would also like to thank Konrad Schindler for accepting me visiting his group. I met lovely people and really enjoyed my time there.

Last but not least, I cannot thank enough my parents, sister and my love for their selfless, endless and precious love.

Contents

Chapter 1

Introduction

Contents

The world is changing all the time. The study of change detection, in my opinion, reveals the interaction between beings and the world, and the evolution of themselves.

1.1 Motivation

This doctoral research is conducted under the frame of the TerraMobilita[1] project that aims to develop solutions for the 3D data acquisition and processing, the production of maps of urban roads and public space, the management and maintenance of roads, as well as applications and services for soft traffics, i.e. pedestrian, bicycle, roller skate, stroller, wheelchair, etc.

Two main innovations are expected. First, integration of the current process of creation and revision of 2D road maps with 3D data acquired from a mobile mapping vehicle. The objectives are: reduction of time and costs; possibility of more frequent updates (e.g. monthly); systematic provision of information not available on the current frame nor in the process of current surveys. Second, building complete 3D models of public space by integrating the surface of the roads and other public place or infrastructures (e.g. parking, trash-bin, lamppost) to develop applications and services for not only the development, but also management and maintenance of roads, e.g. production of urban mobility maps for travel itineraries, and roads accessibility diagnoses for disability access.

The theme of this dissertation is change detection, which will be used for the street map updating, environment understanding and other related applications.

1.1.1 Fine-scale change detection

Change detection is a classical research problem in the fields of photogrammetry and remote sensing. The physical world is changing all the time hence change detection techniques are needed for the purpose of monitoring, map updating, understanding the changing tendency, etc. For instance, land cover and land use changes are intensively studied in years using airborne or remote sensing imageries.

Classical change detection deal with data of massive coverages and of very long-term intervals, e.g. urbanization, deforestation, deformation. Change detection of a whole city after a natural disaster is also commonly encountered. Whereas, small scale changes, i.e. changes at street level, are equally or even more important for our daily life, e.g. new bus stops, installation of street lamps, open of public squares, construction sites.

We aim to detect changes at fine scales in street environments with flexible temporal-scales, meaning the time interval of change can be as long as years, days, hours or even instant depending on the applications. This requires a flexible data acquisition system which can easily access the same points/locations of interest.

A mobile mapping system (MMS) is normally a geo-referenced vehicle mounted with cameras and/or laser scanners for environment perception. The movability makes it a perfect system for street to city scale data acquisition. It can be used for street mapping, building reconstruction, road inventory, etc. Using MMS datasets, the iTowns[2] project aims to integrate 3D building footprints/models, street view images and laser scanning point cloud (LSPC) into a online platform for augmented city visualization.

Laser scanning, or Lidar (light detection and ranging), technology used on MMS is normally referred to as mobile laser scanning (MLS). It can be considered as a part of the MMS besides other type of data acquisition parts, e.g. optical cameras. Compared with MMS cameras, no

[1] www.terramobilita.fr
[2] www.itowns.fr

matter stereo or panoramic, laser scanning captures directly 3D locations with high accuracy, hence is suitable for geometrical data interpretation. It is active acquisition and insensitive to light conditions. Despite the lack of color information, we only focus on LSPC change detection in this research.

1.1.2 Applications

Change detection of MMS data can be used for many applications. The most obvious one is map/database updating, given that one has the original database. These changes is preferably to be "permanent", otherwise they should not be updated to the database. Whereas in most circumstances, temporary changes, e.g. moving urban objects, are not avoidable. In fact, if the time interval is not long, e.g. several days, most of the detected changes are moving objects, since no "real" change of the map database is likely to be occurred. One can take advantage of this fact to remove the temporal changes so that the dataset is cleaned, which can be used for better scene understanding, reconstruction and mapping. Moving object removal is helpful for mobility assessment since movable objects should not be considered as obstacles. Moreover, it can improve the quality of street-view images in a way that temporary moving objects will be remove so that they will not be projected on to street models (Vallet et al., 2015). This will also facilitate the bundle adjustment step (orientation and triangulation) of multi-view stereo from images, since moving objects will introduce false corresponding points.

Other practical and specific applications are:

- Pedestrian flow estimation: change detection method is able to detection moving objects, including pedestrians. Thus their trajectories can be recovered if the temporal sampling density is high, i.e. time interval is small. The trajectory pattern is important for public space design and management.
- On-street parking monitoring: change detection can be used to estimate the parking duration by checking whether two cars acquired at the same parking spot are the same. Actually, this application brings us more challenges other than change detection. For instance, cars have to be detected in the first place, including their accurate locations. Moreover, car types can also be classified to extract further information for parking lot management.

1.2 Problems

Street environment change detection can be much harder than classical large scale ones since the objects sizes are much smaller. The data acquisition has to cover more details (higher resolution), and the change detection method has to be more accurate.

Two different change detection strategies are top-down change detection (TDCD) and bottom-up change detection (BUCD). As for TDCD, interested objects are extracted or the whole data are classified first, then the changes are detected at object level. Meaning, the objects used for comparison are known. Whereas for BUCD, datasets are compared directly without knowing what has been compared and what has been changed. The detected changes are recognised afterwards. We aim to investigate both strategies with the intention of finding the most suitable method for specific situations.

Two typical laser scanners used in MMS are REIGL and Velodyne scanners. They have different scanning geometries hence they are often used for different purposes (detailed in the following section). Regarding change detection, apart from the scanning geometries, the temporal scales are significantly different. REIGL scanner scans the profiles of surroundings. The time interval of two epoch data depends on the repetition of acquisition. Whereas the Velodyne scanner scans the full surroundings for each rotation (panoramic view). The rotation speed is quite high such that instant changes can be detected. Most existing change detection methods are one-task or one-sensor oriented.

One of the main challenges for street level change detection is occlusion. All the objects on streets will induce occlusions which can be confused as changes in many cases, especially when using RIEGL laser scanner. Thus it is vital to distinguish them from real changes. This will be discussed in details in the related chapters.

Apart from the fundamental change detection question, we also tackle the aforementioned practical applications: pedestrian detection and tracking, and on-street parking monitoring. Velodyne scanner is used for the first one, and RIEGL scanner is used for the second one. A generic change detection method is needed, and other problems, e.g. tracking, classification, are to solve.

1.3 Mobile mapping system and data

The MMS used for this research is called Stereopolis (Paparoditis et al., 2012), which has been introduced by Demantké (Demantké, 2014) and Monnier (Monnier, 2014). The MMS is composed of three main parts:

- Mobile platform: vehicle. In general, it can be any movable carriers. In city environments, a vehicle is the most efficient platform. Still, there are many places where vehicles have no access to due to road limitations. Hence, alternative carriers, e.g. electric wheelchair, can be chosen.
- Positioning unit: GNSS + IMU + Odometer. Global navigation satellite system (GNSS) localizes the whole system in the global world coordinate system. Inertial measurement unit (IMU) or inertial navigation system (INS) enables the positioning when GNSS signals are unavailable, which occurs often in dense city areas of high buildings and narrow streets. Odometer measures the vehicle moving distance, which can rectify the localization errors.
- Acquisition unit: cameras and laser scanners. There are both optical cameras and laser scanners on Stereopolis MMS. Sometimes, there is only one type of sensor on the system depending on the specific task. Optical cameras can be used for street-view visualization, street environment reconstruction and even self localization. Laser scanners can be used for high precision street mapping, modelling and simultaneous localization and mapping (SLAM). These two type of sensors are complementary to each other. Note that cameras are significantly cheaper than laser scanners, especially high-end REIGL scanners.

On the MMS, there are one panoramic camera and two stereo cameras, one facing forwards and one backwards. Since we only focused on LSPC change detection, optical cameras on the system are neglected even though without doubt that imagery information will facilitate the

change detection results. In this research, we intend to apply our change detection method on different practical applications, which need different scanning techniques. So both RIEGL and Velodyne scanners are introduced.

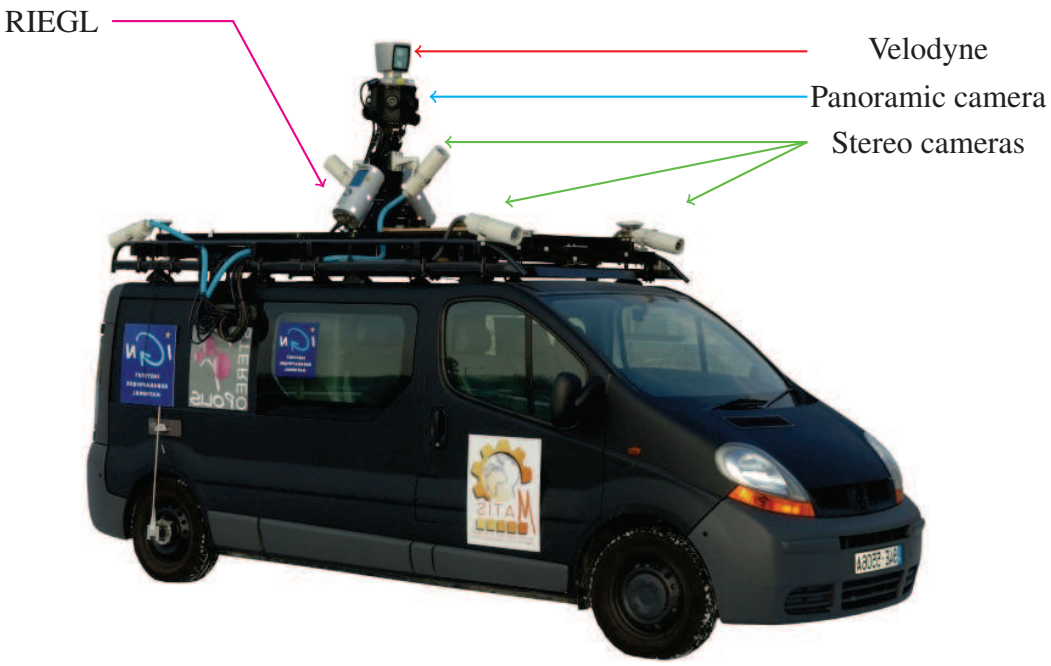


Figure 1.1 – Stereopolis: Mobile mapping system of IGN (Monnier, 2014).

1.3.1 REIGL laser scanner

Two different models of REIGL scanners are used on the system, REIGL LMS-Q120i (Figure 1.2a) and VQ-250 (Figure 1.2b). The former one has a field of view (FOV) of 80°, range accuracy 20 mm, rotation frequency 100 Hz (600 rpm). The number of scanned points can be upto 10 000 per second. The latter one has a vertical FOV of 360°, range accuracy 10 mm, rotation frequency 100 Hz. And the maximum measurement rate is 300 000 points/sec. The former scanner has very limited FOV hence there are two of them scanning each side of the vehicle. The latter has a wider FOV, better accuracy and higher measurement frequency. It rotates along an axis that is aligned with the vehicle moving direction, so it scans the vertical profile of the surroundings.

(a) (b) (c)

Figure 1.2 – Laser scanners on Stereopolis Mobile Mapping System. (a) RIEGL LMS-Q120i, (b) RIEGL VQ-250, (c) Velodyne HDL-64E.

Data presentation Data acquired by both two models are illustrated in Figure 1.3. Model LMS-Q120i covers pavements and lower part of buildings on one street side. Model VQ-250 has a full coverage of the environment.

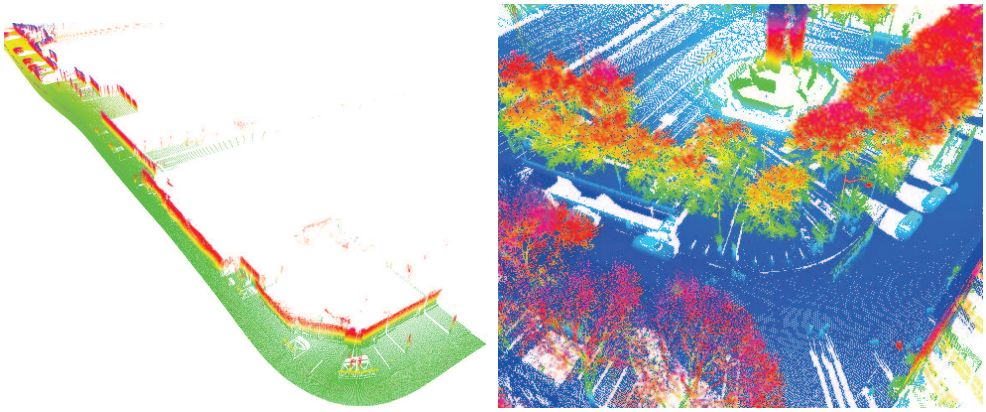


Figure 1.3 – Data acquired by RIEGL scanners. Left: LMS-Q120i, coloured by height from green (low) to red (high). Right: VQ-250, coloured by height from blue (low) to red (high).

1.3.2 Velodyne laser scanner

The Velodyne scanner model is HDL-64E (Figure 1.2c), which contains 64 lasers that are equally distributed along the vertical axis, and it rotates horizontally. It has horizontal FOV of 360°, vertical FOV 26.8° (+2° to −24.8°), range accuracy 20 mm, and rotation frequency 5-15 Hz. Since the scanner has a full/panorama view of the surrounding at a high frequency, it is perfect for instant environment perception and moving object detection. In the field of robotics, it is heavily used for SLAM and autonomous driving.

Data presentation Data acquired by Velodyne HDL-64E are illustrated in Figure 1.4. Differ from RIEGL VQ-250 scanner, which has a vertically full FOV, it has a horizontal 360° FOV. Only the lower part of the street is scanned. Notice that points will accumulate on static objects given that the positioning is accurate. Whereas points on moving objects will have different locations along the trajectories.

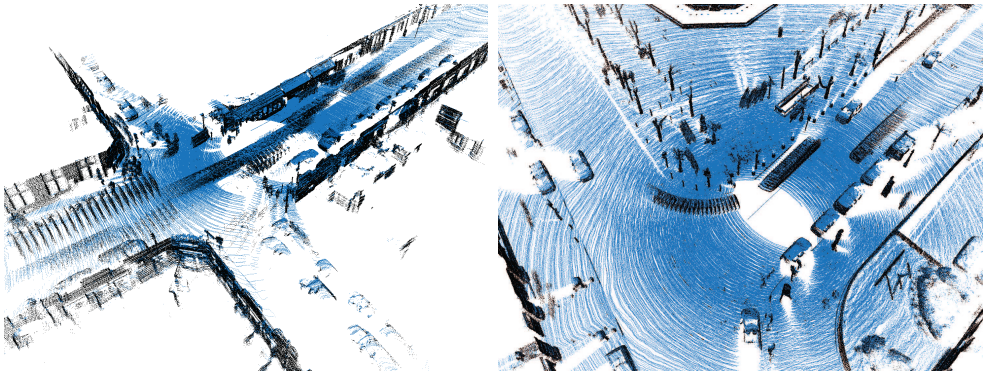


Figure 1.4 – Two datasets acquired by Velodyne HDL-64E, coloured by height from blue (low) to black (high).

1.4 Objectives

The objective of the research is to develop a generic change detection method at fine scale using both REIGL and Velodyne LSPC. REIGL data can be used for long-term change detection with higher point density and larger coverage. Velodyne data can be used for both long-term and short-term or even instant change detection. So that the method is applicable to multiple temporal-scale LSPC. Since occlusions can induce false changes, e.g. points in one epoch will have no corresponding points in the other if the space is occluded, a method that is robust to occlusions is needed. Hence, the change detection method is **generic**, **adaptable** and **robust**.

Apart from change detection, we also aim to tackle some practical problems that are closely related. The first problem is pedestrian flow estimation, which amounts to detect and track pedestrians to recover their trajectories. Moving object detection can be treated as instant change detection. The goal is not only to detect the changes, but also to recognise pedestrians and recover their trajectories.

The second problem is to monitor on-street parking lots, i.e. to extract statistics of occupancy rate, car type, parking duration, etc. This relates to several research topics, e.g. car detection, classification and change detection. The objective is to develop a comprehensive system that extracts as many information as possible from multi-temporal datasets.

1.5 Structure

We tried to apply the change detection method to some practical applications, hence the research problems are extended. The change detection method, as the core of this research, is presented as a whole chapter. The other two parts, pedestrian flow estimation and parking lot monitoring, can be considered as further investigations of change detection. One explores the applicability of the change detection to different scanning geometries and temporal scales, and also investigates moving object detection and tracking. The other focuses on the change detection at object level, i.e. TDCD, which is different from change detection at point level, i.e. BUCD.

The dissertation is organized into six chapters.

After this introduction chapter, the state-of-the-art of change detection is presented in Chapter 2.

Chapter 3 describes the essential change detection methodology, i.e. the occupancy-based method and the combination with distance-based method.

Chapter 4 extends the occupancy-based method to Velodyne LSPD, and investigates moving object detection and tracking.

Chapter 5 explores TDCD in the framework of street-side parking lot monitoring. Car detection and classification are also presented.

Each of these three chapters focuses on one research problem, hence there is a full method-experiment-conclusion pipeline for each of them.

The last chapter concludes the overall work and shares some perspectives.

1.6 Contribution

Some of the main contributions are highlighted here. Detailed ones are discussed in relative chapters.

A change detection method that is applicable to both REIGL and Velodyne data. Including:

- Combination of occupancy-based and distance-based change detection.
- No voxelization for occupancy-based change detection.
- k-D tree for neighbouring points enquiry
- Sensor coordinate instead of world coordinate for space interpolation.

Accurate pedestrian trajectory recovery in 3D. Including:

- Static and dynamic environment extraction.
- Simultaneously detection and tracking of pedestrians.

A full system for on-street parking monitoring. Including:

- Model-based car reconstruction.
- Object level change detection based on supervised learning.

Chapter 2

State-of-the-art

Contents

Change detection is a classical topic in the fields of remote sensing and photogrammetry for environment monitoring, database updating, etc. With the development of laser scanning technology, change detection has been gradually applied to airborne and static terrestrial laser scanning data. In most recent years, change detection of terrestrial laser scanning on mobile platforms has become increasingly popular. In this chapter, change detection methods are generally presented first, then they are summarized in details according to different data acquisition platforms. Change detection in related research domains, i.e. computer vision and robotics, is also addressed.

2.1 Change detection methods

Change detection techniques have been developing for decades. Several reviews have categorized change detection approaches from different perspectives. Singh (1989) generalizes the basic approaches into two, i.e. classification based comparison and comparison directly on raw multi-temporal data. Lu et al. (2004) summarize ten different applications of change detection and group the change detection approaches into seven categories, i.e. algebra, transformation, classification, advanced modelling, geographic information system (GIS) integration, visualization and others. Both the advantages and disadvantages of the approaches are discussed and the degree of complexity is ranked. Based on previous categorizations, Gong et al. (2008) classify change detection approaches into two general groups, i.e. bi-temporal change detection and temporal trajectory analysis regarding the data sources that are used. Furthermore, they also organize the algorithms into seven categories, i.e. direct comparison, classification, object-orientated method, modelling, time-series analysis, visualization and hybrid method. Recently, Hussain et al. (2013) summarize change detection approaches as pixel-based, e.g. image differencing, and object-based, e.g. classified object comparison, for remotely sensed images.

In general, change detection methods follow two main strategies: *direct comparison* and *post-classification comparison* (PCC). Direct comparison means comparing two raw datasets directly after registration. As for images, the comparing result is the difference of each pixel value, and is normally called *difference image*. Then meaningful changes can be found by clustering the changed pixels. More sophisticated methods also take the original image's pattern and the difference image's smoothness into account. As for 3D lidar data, the "difference image" is normally the minimum distance between a point and its reference point cloud. The distance can simply be the point-to-point distance, or preferably point-to-surface distance. The surface, e.g. triangle, plane, is calculated by a proper local subset. Sometimes, the distance between two local subsets, e.g. *Hausdorff distance*, is more suitable. This somehow corresponds to the region-based image differencing method, which compares a group of pixels instead of individuals. PCC method firstly classifies the data into meaningful objects, then the differences between objects are detected. The compared values are object features. Sometimes, objects are modelled to better characterize their features. Both supervised and unsupervised learning are used to classify changes in the feature space. The changes are detected at object level, hence the method is often regarded as object-based change detection.

Actually, the concepts of change of pixel/point-based and object-based methods are different. The former defines changes as the inconsistency of acquired data. Changes are detected at low level. The significance of changes has to be discovered afterwards. We call this type of method bottom-up change detection. Whereas, the later treats object's dynamics as changes. So the target object has to be specified in the beginning. This type of method can be called, on the contrary, top-down change detection.

2.2 Change detection from imageries

Image-oriented change detection approaches vary from pixel-based, region-based to object-based methods. Hussain et al. (2013) group all the approaches as only pixel-based and object-based, where the pixel-based approach includes both direct comparison and PCC. Tian et al. (2013), however, divide them into pixel-based and region-based methods. The region is treated as an general term, including both segment and object. They claim that region-based meth-

ods perform generally better than pixel-based methods. Benedek et al. (2015) prefer to use region-based and object-based methods for categorization. Apparently, researchers have different interpretations of the same terms.

Many change detection studies can be found using different methods or image types. Few recent work are presented here. Tian et al. (2013) use a region-based method for building and forest change detection. DSMs are generated firstly from Cartosat 1 stereo images. Then the original images are segmented by mean-shift into individual regions. Potential changing regions are found by intersecting two compared image segments and merging small regions. A multilevel, i.e. different number of neighbours, change vector is extracted including both change features from DSM heights and image pixel values. Results suggest that, by taking the region as a unit, the change detection results are mostly superior than simple pixel-based method in both forest and industrial areas. Benedek et al. (2015) comparatively studied three multilayer Markov Random Field (MRF) based change detection methods, namely *Multicue MRF*, *Conditional Mixed Markov model*, and *Fusion MRF*. The first two methods both use *direct comparison* strategy, but different features and inter-layer connects to ensure smooth segment results. The last method belongs to the PCC category. Multi-temporal images are fused and segmented. Changes are detected by comparing the classification results of image segments. Vallet (2013) detects changes from Digital Elevation Models (DEMs). After generating the height difference image, a multilevel shape variation is analysed based on the theory of persistent homology. This method is independent from threshold setting. Gressin et al. (2013) use object-based change detection method to update a landcover database. Not only the object changes are detected, the theme/type and number of object are also inspected. A supervised classification is implemented to detect changes at object level, using Support Vector Machines (SVMs).

2.3 Change detection from airborne lidar data

Similar to the image-oriented cases, change detection using airborne laser scanning (ALS) data also starts from pixel-based method by transform the point clouds to images. The potential of ALS system for change detection in urban area was discussed early on by Murakami et al. (1998), who also detected the changes of buildings using ALS data (Murakami et al., 1999). They subtract digital surface models (DSMs) generated from ALS data at different times. Then a simple shrinking and expansion filter was utilized to remove edges of unchanged features. Changes are detected by simple image differencing at 2.5D. Many later studies follow the same strategy for both urban and forest environment change detection (Steinle et Bahr, 2003; Matikainen et al., 2003; Vögtle et Steinle, 2004; Yu et al., 2004; Champion et al., 2009; Choi et al., 2009; Rutzinger et al., 2010). Walter (2004) uses pixel-based and object-based classification of multispectral and lidar data for change detection in geographic information system (GIS) databases. Vosselman et al. (2004) apply change detection of lidar data for updating medium scale map. First, the laser data are segmented and classified as bare-earth, building and vegetation. Then the segments of buildings are matched against the building objects of a topographical database. In the change detection experiment, all newly constructed buildings are detected reliably. Choi et al. (2009) present a feature based approach to automatically detect changes in urban areas. The main processes are first detecting changed areas through the subtraction of two DSMs, then segmenting and classifying the changed patches to predefined classes, ground, vegetation and building, and last determining the types of changes based on the classes and properties. This method is able to detect the changes in a sufficient degree of

accuracy semi-automatically. However, there is no quantitative evaluation.

Multi-temporal lidar data have been a major source for forestry studies in many researches (Mallet et Bretar, 2009). Yu et al. (2004) detect harvested trees and forest growth using airborne lidar data. The estimation of height growth is accomplished by individual tree delineation and a tree to tree matching algorithm. First, trees are located by detecting the local maxima in the CHM, and then the crown shape is determined by watershed segmentation. Last, trees are matched by the location with a threshold distance at 0.5m. The precision is about 5cm and 15cm at stand level and plot level respectively based on field and statistical analysis. Individual tree height growth is detected again by Yu et al. (2006), who present three change detection manners, i.e. differentiation between DSMs and CHMs, canopy profile comparison and analysis of height histograms. Hausdorff distance, the maximum distance from a point in one set to the closest point in a different set, is applied to improve the tree to tree matching result.

On the contrary of most early studies, which detect changes in 2.5D, recent studies tend to process data directly in 3D. Xu et al. (2014) first classify the ALS data into six types of urban objects, i.e. ground, water, vegetation, roof, wall, and roof element. Features are extracted from multiple entities, e.g. individual point, planner segment, mean shift cluster. Five classifiers are use for supervised classification. The overall accuracy is as high as 97.0%. Then changes on buildings are detected using a surface separation map, i.e. a point-to-plane distance map. The map is segmented and the changed segmented are detected based on contextual reasoning (Xu et al., 2013). Xiao et al. (2012) focus on the changes of urban trees. After refining the classification result of trees, the tree crowns are modelled in 3D and parameters are derived from the models. Changes are detected by parameter comparison. Hebel et al. (2013), however, use *occupancy grid* from robotics for direct ALS data comparison in 3D without pre-classification. This work is discussed in detail in section 2.5.

2.4 Change detection from terrestrial lidar data

Complementary to aerial lidar, terrestrial laser scanning (TLS) can be either static or mobile. In practice, TLS often refers to the static case, and mobile TLS is called mobile laser scanning (MLS).

Change detection from TLS TLS data demand more accurate detection methods since the point density is very high and the 3D position is very precise. Object-based change detection can be affected by the object recognition accuracy, thus point-based and region-based methods are often used. Point-to-point distance-based direct comparison method is practical for TLS data because changes can be detected directly in 3D with high degree of accuracy. Nevertheless, point-to-point distance is very sensitive to point density. A local surface model can be helpful, since for example point-to-triangle distance or point-to-plane distance are more robust than simple point-to-point distance. Girardeau-Montaut et al. (2005) propose a framework to detect changes from terrestrial lidar data semi-automatically. Point clouds are directly compared using three methods, i.e. average distance, best fitting plane orientation and the Hausdorff distance (the maximum distance among the points in one set to the closest point in another set). Results show that the Hausdorff distance performs best. A local model for distance calculation is suggested in order to avoid density variation issues. Occlusion problem is solved using visibility map. Invisible points were dealt with separately. Kang et al. (2013) detect changes on building façades from TLS data. Datasets are matched first using reflectance images, then buildings are segmented. Hausdorff distance is also used to detect the changes between building segments.

Zeibak et Filin (2007) treat 3D TLS scans as range panoramas. Range images are compared from the sensor perspective, which avoids false detection on occluded parts. This is similar to the occupancy grid method. Changes are detected by thresholding the range difference. However, false alarms on trees are left as unsolved.

Change detection from MLS MLS change detection is quite similar to TLS except different scanning geometry and scope. Aijazi et al. (2013) firstly classify MLS data into permanent and temporary objects. Then permanent parts from all different scans are retained to incrementally build a complete 3D urban map. A similarity map is constructed on 3D voxels for multiple epoch data fusion. Changes are detected to fill perforated areas. Qin et Gruen (2014) detect changes at street level using MLS point clouds and terrestrial images. After co-registration, points are projected onto each image. Then, stereo pairs of terrestrial images are compared with point clouds to find the geometrical consistency. Finally, initial changed areas are optimized by graph cut. Lindenbergh et Pietrzyk (2015) summarize change detection methods using both static and mobile laser scanning, including ALS, as binary change and quantified change. Various applications, e.g. structural monitoring, geomorphology, urban and forest inventory, are also presented.

2.5 Change detection in related domains

Change detection in computer vision. 3D change detection has been applied to moving object detection and urban environment monitoring in computer vision. Yin et Collins (2007) detect moving objects by a Belief Propagation approach using a 3D Markov Random Field (MRF). A similar method has been presented by Košecka (2013) to detect changes from street scene images. Changes are differentiated as structural, appearance change or temporary dynamically moving objects. Sakurada et al. (2013) detected changes of 3D urban structures from a vehicle-mounted camera. The similarity of the local image patches is computed from multi-view images. The method is compared with Multi-View Stereo (MVS) based methods. Many investigations are based on voxelized space, which performs better than MVS models as compared by Taneja et al. (2011). Structural changes have been detected by voxelizing places of interest. Geometric consistencies between voxels are evaluated. Inconsistency indicates a change in the scene. They extend the work to city-scale in order to detect changes in cadastral 3D models for facilitating the model updating process (Taneja et al., 2013). Pollard et Mundy (2007) store probability distributions for surface occupancy and image appearance in 3D voxel grids. Then they are updated by new images based on Bayesian decision theory. The changes are detected by thresholding the probability to obtain a binary mask. The work has been extended to 4D by Ulusoy et Mundy (2014). 3D changes are detected on 3D models in a time series for model updating instead of rebuilding models at each time.

Change detection using occupancy grids from robotics. Pagac et al. (1996) use occupancy grids for constructing and maintaining a map of an autonomous vehicle's environment for navigation purposes. A sensor beam is projected on a rectangular grid assigned probabilities of cells being empty, full and ignorance outside the beam. Every cell is initialized, $m(empty) = m(full) = 0$ and $m(ignorance) = 1$, then the Dempster-Shafer Theory (DST) is used to fuse the sensor readings. The DST has proved to outperform the Bayesian method which needs to specify all conditional probabilities even if no *a priori* information exists. Wolf et Sukhatme (2004) also use an occupancy grid for SLAM in dynamic environments. The states of the occupancy grid are defined as *Free*, *Unknown* and *Occupied*. Two different grids are used to model

static and dynamic parts of the environment and then combined. The algorithm is capable of detecting dynamic objects in the environment. Most of the previous studies model occupancy in 2D grid cells. Underwood et al. (2013) detect changes in 3D from lidar data using ray tracing which is similar to occupancy modelling. Changes are categorized as *additions*, *subtractions* and *discrepancies* w.r.t. the previous occupancy state. A spherical reference frame is proposed to present the relative position of two scans. Laser beams are treated as cone-shaped. Then changes are detected by setting thresholds on range distances. Hebel et al. (2013) detect changes in ALS point clouds using voxel indexing. The occupancy of laser rays is modelled by sigmoid functions. Then the DST is applied to combine multiple measurements. Vegetation is modelled with different parameters from others due to self-conflicting. Changes in buildings and cars are successfully detected.

A summary of the change detection using laser scanning data is listed as in Table 2.1.

Table 2.1 – Summary of change detection using lidar data

Author	Data	Object	Method	Other
Murakami et al. (1998)	ALS	Urban features	Differential image	2.5D
Murakami et al. (1999)	ALS	Buildings	DSM1-DSM2	2.5D
Steinle et al. (1999)	ALS	Buildings	DHM vs. CAD model	2.5D
Matikainen et al. (2003)	ALS	Building map updating	DSM, segmentation, classification, vs. old building map	2.5D
Steinle et Bahr (2003)	ALS	Urban	Pixelwise height differences, mobile/stationary changes	2.5D
Vosselman et al. (2004)	ALS	Building map updating	Point segmentation, classification, vs. geodatabase	3&2.5D
Vu et al. (2004)	ALS	Buildings	Grid image differencing, histogram thresholding	2.5D
Vögtle et Steinle (2004)	ALS	Buildings	DSM, segmentation, classification	2.5D
Walter (2004)	ALS	GIS database	Lidar-DTM=DHM, pixel-based, object-based classification	2.5D
Hsiao et al. (2004)	TLS	Landslide terrains	2m grid vs. topomap(altitude)	2.5D
Walter (2005)	ALS	Urban	image classification based on existing GIS	2.5D
Girardeau-Montaut et al. (2005)	TLS	Construction site	Octree, point-to-point, visibility maps	2.5D
Yu et al. (2006)	ALS	Forest, canopy height	CHM=DSM-DTM, morphological filtering,h1-h2	2.5D
Zeibak et Filin (2007)	TLS	Urban	Range panoramas	2.5D
Butkiewicz et al. (2008)	ALS	Urban	Mesh(Delaunay triangle), point-to-vertex, error model	3D
Champion et al. (2009)	ALS	Buildings	4 approaches comparison, 3 data types	2.5D
Choi et al. (2009)	ALS	Urban	DSMs subtraction, segmentation,classification	2.5&3D
Rutzinger et al. (2010)	ALS	Building footprints	DSM, vegetation mask, classification	2.5D
Xiao et al. (2012)	ALS	Trees	Classification, modelling	3D
Xu et al. (2013)	ALS	Buildings	Classification, segmentation, height difference	3D
Hebel et al. (2013)	ALS	Buildings, cars	Occupancy grid	3D
Kang et al. (2013)	TLS	Buildings	Segmentation, Hausdorff distance	3D
Aijazi et al. (2013)	MLS	Street scenes	Similarity map	3D
Qin et Gruen (2014)	MLS/Image	Street scenes	Co-registration, geometrical consistency	3&2.5D

Chapter 3

Street Environment Change Detection

Contents

Mobile laser scanning (MLS) has become a popular technique for road inventory, building modelling, infrastructure management, mobility assessment, etc. Meanwhile, due to the high mobility of MLS systems, it is easy to revisit interested areas. However, change detection using MLS data of street environment has seldom been studied. In this chapter, an approach that combines occupancy grids and a distance-based method for change detection from MLS point clouds is proposed. Unlike conventional occupancy grids, our occupancy-based method models space based on scanning rays and local point distributions in 3D without voxelization. A local cylindrical reference frame is presented for the interpolation of occupancy between rays according to the scanning geometry. The Dempster-Shafer theory (DST) is utilized for both intra-data evidence fusion and inter-data consistency assessment. Occupancy of reference point cloud is fused at the location of target points and then the consistency is evaluated directly on the points. A point-to-triangle (PTT) distance-based method is combined to improve the occupancy-based method. Because it is robust to penetrable objects, e.g. vegetation, which cause self-conflicts when modelling occupancy. The combined method tackles irregular point density and occlusion problems, also eliminates false detections on penetrable objects.

3.1 Introduction

Change detection techniques have been applied in different fields such as environment monitoring (Tian et al., 2013), 3D city model updating (Taneja et al., 2013), street environment inventory (Pu et al., 2011), simultaneous localization and mapping (SLAM) (Wolf et Sukhatme, 2004; Moras et al., 2011), moving object tracking (Yin et Collins, 2007; Irani et Anandan, 1998; Lindström et Eklundh, 2001), surveillance systems (O'Callaghan et Haga, 2007) and so on. The spatial scale can be as large as a whole country, a forest, a city or as small as a street. Objects of interest vary from ground surfaces, vegetation, buildings, cars to pedestrians.

In remote sensing studies, large coverage images are usually used for large spatial scale change detection in forest or urban areas for land-cover and land-use monitoring (Hussain et al., 2013; Tian et al., 2013). Airborne laser scanning (ALS) data is also used for similar applications with high geometric precision due to accurate 3D acquisition (Xu et al., 2013; Hebel et al., 2013; Yu et al., 2004). In recent years, 3D maps and virtual city models have been under fast development, therefore many studies have focused on street environment monitoring and city model updating (Früh et Zakhor, 2004; Kang et al., 2013).

Mobile mapping systems (MMSs) can easily scan streets multiple times, therefore allow us to detect changes at street or even city-scale. Laser scanning provides precise 3D geometric information on the environment, which is of great interest for 3D mapping, localization, scene perception, motion tracking and navigation purposes. Studies from computer vision mainly use imagery for city and street scene change detection (Pollard et Mundy, 2007; Sakurada et al., 2013; Košecka, 2013; Eden et Cooper, 2008; Taneja et al., 2011, 2013). However, lidar (**li**ght **d**etection **a**nd **r**anging) data (also referred to as laser scanning data, range data or lidar point clouds) have been proven to be an accurate data source for 3D urban reconstruction (Lafarge et Mallet, 2011; Chauve et al., 2010; Verma et al., 2006; Zhou et Neumann, 2010; Toshev et al., 2010; Banno et al., 2008; Poullis, 2013), infrastructure management and road inventory (Pu et al., 2011; Zhou et Vosselman, 2012). Thus, mobile laser scanning (MLS) data is intensively studied nowadays (Weinmann et al., 2014; Demantké et al., 2011; Monnier et al., 2012; Yang et Dong, 2013; Aijazi et al., 2013; Serna et Marcotegui, 2014; Qin et Gruen, 2014).

Change detection methods specific to MLS point clouds have been seldom investigated, therefore the development of corresponding approaches becomes urgent. Conventional distance-based methods, e.g. point-to-point, point-to-plane or point-set to point-set distances, may be used for this purpose (Girardeau-Montaut et al., 2005). However, irregular point density and occlusions still remain major challenges. In this chapter, we aim to develop a street environment change detection method that is robust to point density variations and capable of distinguishing occlusions from real changes. First, related work and our contribution are discussed in Section 5.2. Then, the concept of occupancy-based change detection is explained in Section 3.3. Section 3.4 presents the PTT distance-based method and the combination with the occupancy-based one. Section 3.5 describes the experiments and the corresponding results. Quantitative evaluation is demonstrated in Section 3.6. Finally, conclusions are drawn and limitations are discussed in Section 5.5.

3.2 Contribution

Our method is inspired by the previous studies (Pagac et al., 1996; Wolf et Sukhatme, 2004; Underwood et al., 2013; Hebel et al., 2013) using occupancy grids for space modelling. Our preliminary occupancy-based method has shown the capability and reliability of urban environment change detection at object level (Xiao et al., 2013). Urban objects, e.g. cars, garbage bins, bicycles, pedestrians, were successfully detected in the conducted experiments. However, the datasets were small, field of view was narrow (only lower part of the street, no vegetation scanned), and objects were close to the sensor. Most importantly, occupancy around the point was simply modelled along the ray direction, which leads to misdetections (details in Section 3.3.2). In this chapter, we present both the original and the improved methods of occupancy modelling around a ray and, especially, around the point considering its normal. Moreover, we utilize completely new challenging datasets, which cover a large area in a complex street environment without human intervention, and have massive number of points and irregular point density. In addition, we combine the occupancy-based method with a PTT distance-based method, which is robust to penetrable objects, to further improve the change detection result. Statistical analysis and comparison are also conducted.

Compared to related work, our main contribution is a method that is robust to irregular point density and able to distinguish occlusions from real changes in MLS data. In more detail:

- *Combination of occupancy and distance:* Occupancy-based method is able to distinguish occlusions from real changes, and distance-based method is robust to penetrable objects. They are complementary to each other. We take the advantages of both to improve the final change detection result.

- *Local cylindrical reference frame:* A local cylindrical reference frame is defined from the sensor's perspective according to the scanning geometry to interpolate evidence when it is missing between rays.

- *K-d tree for point query:* The scope of a scan ray is not considered as a cone but a prism. A K-d tree is used to optimise the query of points inside the prism.

- *No voxelization:* Consistency is evaluated directly on points without voxelization. We demonstrate that conflicts (changes) may only occur near the points at one of the compared epoch data. Thus, evidence is accumulated directly at points' locations to compare with the other epoch.

- *Occupancy modelling considering uncertainties:* Occupancy states are modelled elaborately over a scan ray and around the point in 3D with physical meanings, taking into account the acquisition geometry and the underlying uncertainties.

- *Point-to-triangle distance (PTTD):* Due to the inconsistent point density and the complexity of street objects, we propose PTTD instead of point-to-point or point-to-plane distances. The Delaunay triangulation moderately generalizes the object surfaces.

3.3 Occupancy-based change detection

The principle of our occupancy-based approach is that a laser scan ray indicates the occupancy of space. Scenes covered by laser scans can be represented by such occupancy. If scenes have changed, the occupancy of space will change as a result. The inconsistency of occupancies

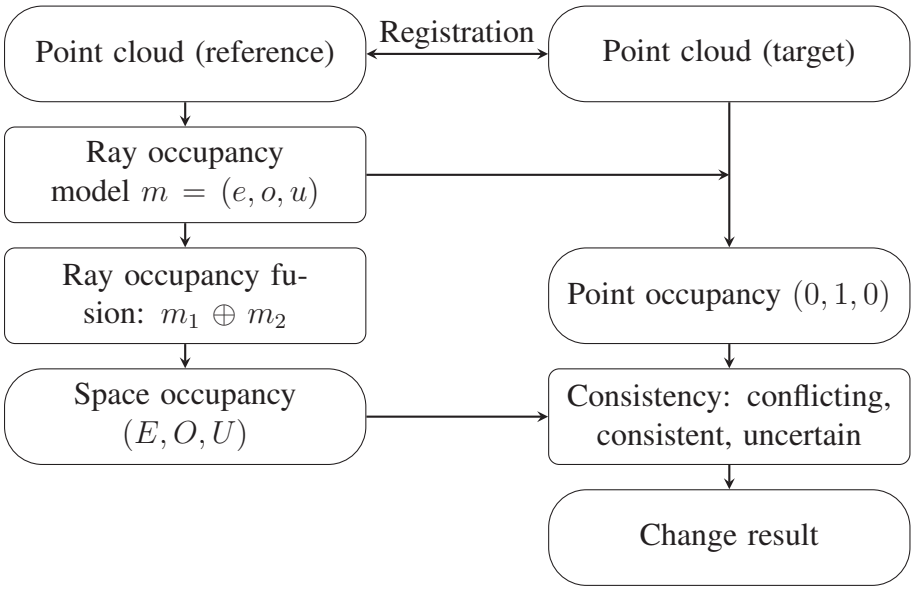


Figure 3.1 – Flowchart of our occupancy-based method including three main steps: single ray occupancy modelling, ray occupancy fusion in reference data, and consistency assessment between reference and target point clouds.

between datasets will then indicate the changes.

The workflow including three main steps is illustrated in Figure 3.1. Two epoch data are taken for comparison, one target, one reference. To obtain the change information of the target points, first, occupancy of a single ray from the reference data is modelled, considering the measurement and registration uncertainties. Then, occupancies of all the rays from the reference data are fused together. Last, the occupancy of the reference space is compared with the occupancy of each target points resulting in the consistency between these two. Then changes are detected based on the consistency assessment (details in Section 3.3.4). The workflow depicts the detection of changes in the target data w.r.t. the reference data, whereas the two datasets are exchangeable since the method is symmetrical.

3.3.1 Laser Scanning Geometry

The scanner mounted on the MMS Stereopolis Paparoditis et al. (2012) scans the profiles of streets. The scan ray rotates on a vertical plane perpendicular to the trajectory. Figure 3.2 shows the geometry of data acquisition.

In addition to the xyz coordinates, the origin and rotation angle of each point are known. The ray is not a perfect line but a cone-like beam with a particular footprint. The angular resolution $\alpha = 0.4$ degrees and the range accuracy $\sigma_m = 25mm$. The distance D between two scan lines varies because it depends on the inconsistent speed of the vehicle. Point densities in vertical and horizontal directions are inconsistent since the horizontal sampling depends on the vehicle speed while the vertical density depends on the object distance and the incidence angle.

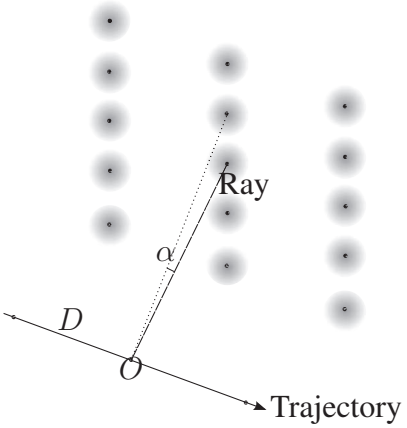


Figure 3.2 – Laser scanning geometry

3.3.2 Occupancy modelling for an individual ray

This section firstly presents the DST and its application to occupancy modelling, following which the improved modelling method is presented. Then, uncertainties are integrated with the occupancy functions.

Occupancy modelling along a ray

One advantage of the DST is that the combination rule is commutative and associative so that evidence can be fused in any order, which is essential for handling objects scanned from different perspectives and datasets acquired at different times. It allows to combine evidence from different sources and arrive at a degree of belief representing all possible states of a system. The state of space occupancy can be *empty* or *occupied*, represented by a universal set $X = \{empty, occupied\}$. The power set of X, $2^X = \{\emptyset, \{empty\}, \{occupied\}, \{empty, occupied\}\}$, contains all of its subsets. Due to occlusions in data acquisition, no information is obtained in shadow areas; hence, the occupancy state is unknown. When the occupancy of space is *unknown*, it can be either *empty* or *occupied*, so the set $\{empty, occupied\}$ represents state *unknown*. According to the DST, the mass (m) of each element of the power set is within the range [0, 1]. Moreover, the mass of the empty set is 0 and the masses of the remaining subsets add up to a total of 1 :

$$m : 2^X \rightarrow [0, 1], m(\emptyset) = 0, \sum_{A \in 2^X} m(A) = 1 \tag{3.1}$$

The DST represents all the three states of occupancy and sets a range for each state. This avoids the effect of redundancy due to repetitive scanning. The mass of each state, $m(\{empty\})$, $m(\{occupied\})$ and $m(\{unknown\})$ are abbreviated as e, o and u respectively.

The mass of occupancy is denoted by m, and for each element, $m(\{empty\})$, $m(\{occupied\})$ and $m(\{unknown\})$ are denoted by e, o and u respectively. To define the mass of occupancy of scan rays, a local reference frame is presented. Figure 3.3 illustrates the relative position of point P and Q in 3D. Origin O_t is the scanning origin of point Q. $P^{'}$ is the projection of P on the rotation plane that is perpendicular to the trajectory. The trajectory direction ($norm_{traj} = \overrightarrow{O_{t-1}O_{t+1}}$) is obtained by the scanner centers from the scan lines before (O_{t-1})

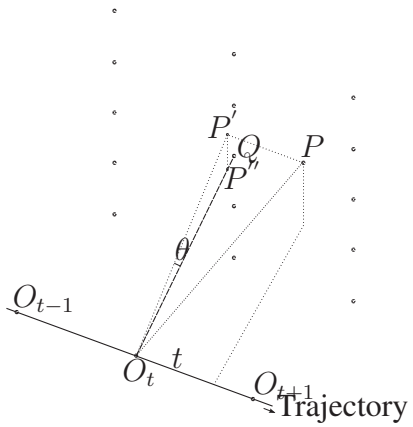


Figure 3.3 – Relative position of P and Q

and after (O_{t+1}). P'' is the projection of P on $\overrightarrow{O_tQ}$. θ is the angle between $\overrightarrow{O_tP'}$ and $\overrightarrow{O_tQ}$, t is the length of the projection of $\overrightarrow{O_tP}$ along the trajectory and r is the difference between the length of $\overrightarrow{O_tP''}$ and $\overrightarrow{O_tQ}$. θ, r, t represent the relative position of point Q_t and P in a local cylindrical coordinate system.

$$\theta = \arccos \frac{\overrightarrow{O_tQ} \cdot \overrightarrow{O_tP'}}{||\overrightarrow{O_tQ}|| \cdot ||\overrightarrow{O_tP'}||} \tag{3.2a}$$

$$r = ||\overrightarrow{O_tP''}|| - ||\overrightarrow{O_tQ}|| \tag{3.2b}$$

$$t = \overrightarrow{O_tP} \cdot norm_{traj} \tag{3.2c}$$

First we define the occupancy masses along the ray direction (e_r , o_r , u_r). When P is between O_t and Q, it is empty, so $e_r = 1; o_r = 0$. When P is at the position of Q, $e_r = 0; o_r = 1$. When P is behind Q, the mass of occupancy is represented by a Gaussian function. o_r decreases from 1 to 0, and u_r increases from 0 to 1 accordingly (Figure 3.4).

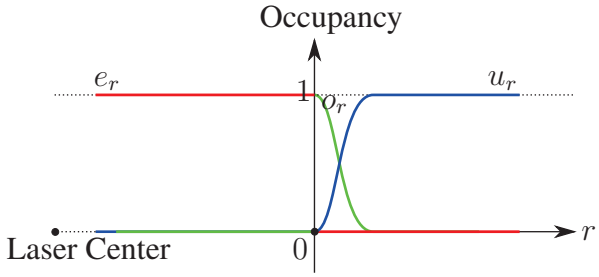


Figure 3.4 – Occupancy in the ray direction: *empty* between a laser point and its scanning origin; *occupied* at the location of the point; *unknown* behind the point (Origin is the point location, r represents the distance to the point in the ray direction).

Uncertainties are taken into consideration by convolving with the occupancy functions (Figure 3.5). When $r = 0$, the point is on the edge of an object (interface of *empty* and *occupied*). Due to uncertainties, the point can be on either side of the interface hence $e_r^{'} = o_r^{'} = 0.5$. The occupancy here is half *empty* and half *occupied* which is different from *unknown* where no information is obtained. It is more certain that the point is on the *occupied* side when it is behind the interface. Thus the mass of *occupied* increases and reaches a maximum behind the point

and then deceases because of a lack of information. The shift s between the maximum value location and the original point depends on uncertainties.

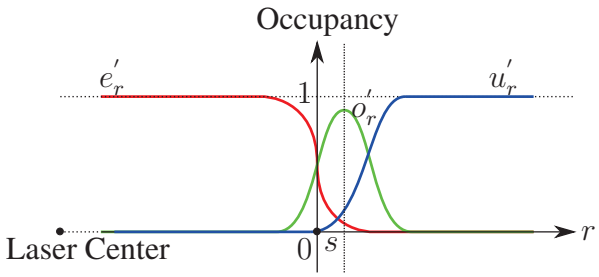


Figure 3.5 – Occupancy in ray direction considering uncertainties.

A laser ray is not a perfect line but a cone-shaped beam with a particular footprint. The occupancy around the ray can be parametrised by the angle of the cone. A simple cone-shaped ray is given in Figure 3.7. Due to uncertainties, the edge of the laser beam is fuzzy. The three elements of occupancy are within the scope of 0 and 1 and they add up to 1 so they can be considered as barycentric coordinates. They are illustrated by a triangle (Figure 3.6).

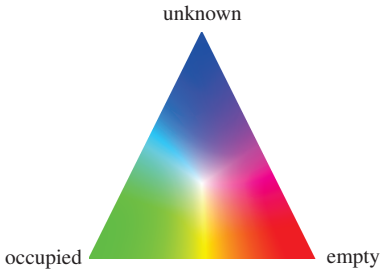


Figure 3.6 – Occupancy triangle

Figure 3.7 – Cone-shaped laser beam (red: empty, green: occupied, blue: unknown).

In reality, point clouds, unlike images, do not cover the whole scanned space. There are gaps between rays. And the gaps vary due to many factors, e.g. vehicle speed, object distance and incidence angle. A simple cone shape will not cover the gaps, therefore will induce under-detection. One single parameter of the ray (angle of the cone) is not sufficient for occupancy interpolation.

Improved occupancy modelling around a point

A common MMS configuration (Paparoditis et al., 2012) is scanning the profile of surroundings with a laser scanner rotating vertically to the vehicle trajectory. A local reference frame (Figure 3.8) is presented to define the occupancy around rays to interpolate the gaps between rays. The origin O is the sensor center, θ represents the vertical rotating angle, and t is the distance

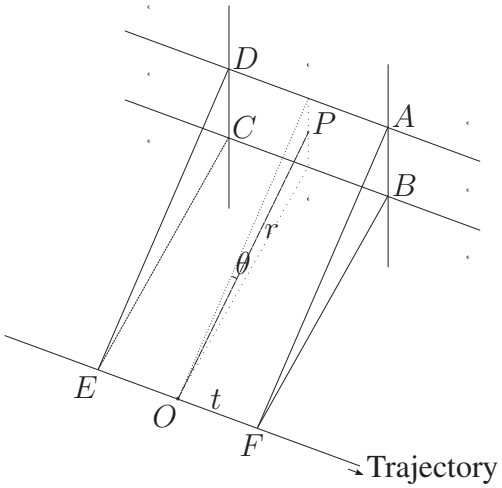


Figure 3.8 – Cylindrical local reference frame for occupancy modelling around a ray OP and its triangular prism shaped vicinity *ABCD-EF* .

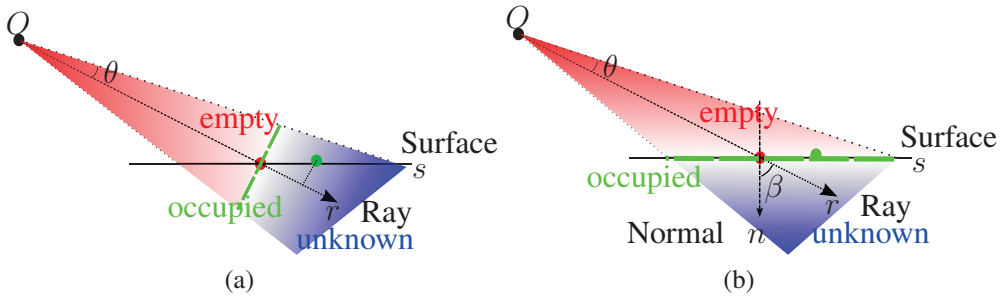


Figure 3.9 – Occupancy around a point (red: empty, green: occupied, blue: unknown). If occupancy is modelled along the ray (a), the green point will be considered in the shadow of the red one since it falls behind in the ray direction r. However, in reality, they are on the same surface s, so occupancy is modelled along the point normal n (b).

in the trajectory direction. Together with the distance in the ray $\overrightarrow{OP}$ direction r, θ and t define the occupancy around the ray in a local cylindrical coordinate system.

In the ray direction, occupancy was previously simply defined as in Figure 3.4 (Xiao et al., 2013). However, the default assumption is that point normals are always in the same direction as the rays, which is obviously false in reality. As shown in Figure 3.9a, the green point is behind the red one in the ray direction, then it will be considered as located in the *unknown* part of the ray. However, they are actually on the same surface. This happens when the incidence angle is large.

To overcome this issue, we improve the method by considering the normal directions of points and modelling occupancy around the surface (Figure 3.9b). Occupancy is modelled as: *empty* between the origin of a point and its surface ($e_n = 1$); *occupied* around the surface ($o_n = e^{-\frac{1}{2}(\frac{n}{\lambda_n})^2}$); and *unknown* behind the surface ($u_n = 1 - o_n$). λ_n represents the mass of occupancy in the normal direction and behind the surface of the modelled object. Occupancy on the surface along the scan line direction is modelled by a Gaussian function $f_s = e^{-\frac{1}{2}(\frac{s}{\lambda_s})^2}$, ranging from 0 to 1. And $\lambda_s = r \cdot \lambda_\theta / cos\beta$, r is the distance between a point and its origin in the ray direction, β is the angle between the ray and the point normal, λ_θ represents the angular mass of occupancy on the sensor rotating plane.

Now the occupancy has been modelled for the ray and angular directions, which are transformed onto the point normal direction and scan line direction on the corresponding surface. Theoretically, the occupancy masses in the rotating direction should be the same as in the trajectory direction since the ray is cone-shaped. However, sampling densities are not the same in these two directions. The horizontal sampling depends on the vehicle speed while the vertical density depends on the object distance and the incidence angle. To overcome this anisotropic sampling, these two directions are considered independently to interpolate the gaps. Thus, in the trajectory direction t, the mass of occupancy is modelled as $f_t = e^{-\frac{1}{2}(\frac{t}{\lambda_t})^2}$. The value of λ_t depends on the gap between scan lines.

When it is far from a ray, there will be no mass contributing by this ray, namely a ray only has mass in its vicinity (defined as 3 times λ_θ, λ_t and λ_r). The vicinity is a triangular prism *ABCD-EF* according to the cylindrical reference frame (Figure 3.8). Occupancy mass out of the prism is significantly small hence can be neglected. The prism is used for *K*-d tree corresponding point searching (details in Section 3.3.3). The local reference frame is defined for a specific MMS configuration, however it is adaptable as long as the whole space is covered. The idea is to interpolate the space from the sensor's perspective.

Uncertainty modelling

Uncertainties of measurement (ranging) and registration are taken into consideration since they can induce changes at fine scales. Both of them are represented by normal distributions with standard deviations σ_m and σ_r. Since the three directions, n , s , t are pairwise perpendicular, uncertainties are considered separately in each direction. In the normal direction, the error of measurement has partial effects since the ranging measurement error only exists in the ray direction. Thus, the overall uncertainty σ_n is the combination of the registration error and the projection of measurement error. They are taken into account by convolving the two error distributions, resulting in a Gaussian function F_n with standard deviation $\sigma_n = \sqrt{\sigma_r^2 + (\sigma_m \cdot cos\beta)^2}$. Similarly, perpendicular to the normal, the uncertainty on the surface σ_s is modelled as a Gaussian function F_s with standard deviation $\sigma_s = \sqrt{\sigma_r^2 + (\sigma_m \cdot sin\beta)^2}$. Since in the trajectory direction, measurement error does not exist, the uncertainty only comes from the error of registration, therefore $\sigma_t = \sigma_r$ for the uncertainty function F_t.

Uncertainties are taken into account by convolving with occupancy functions. In the normal direction, three occupancy states are convolved individually with the uncertainty function:

$$e_n^{'} = e_n \otimes F_n; o_n^{'} = o_n \otimes F_n; u_n^{'} = u_n \otimes F_n = 1 - e_n^{'} - o_n^{'} \tag{3.3}$$

And in the other two directions, occupancy functions after convolution are:

$$f_s^{'} = f_s \otimes F_s; f_t^{'} = f_t \otimes F_t \tag{3.4}$$

The overall occupancy of ray $\overrightarrow{OP}$ is a function of parameters n, s, t: n defines the state of occupancy and also the value of each state; s and t interpolate the occupancy around the ray. The occupancies in s and t directions drop from 1 to 0 when values of s and t increase. They can be considered as weights of the occupancy states in the normal direction. Thus they are multiplied with occupancy state functions as follows:

$$m(\overrightarrow{OP}) = \left\{ \begin{array}{c} e \\ o \\ u \end{array} \right\} = \left\{ \begin{array}{c} e_n^{'} \cdot f_s^{'} \cdot f_t^{'} \\ o_n^{'} \cdot f_s^{'} \cdot f_t^{'} \\ 1 - e - o \end{array} \right\} \tag{3.5}$$

3.3.3 Occupancy fusion and corresponding point retrieval

After modelling occupancy for each single ray, the occupancy of a whole epoch data can be obtained by combining occupancies of all the rays from the epoch, which can then be compared with the target data. In practice, we first search the corresponding target points using the prism-shaped ray scope. Then the occupancy is modelled directly at the location of each point. A point can be within the scope of multiple rays, whose occupancies are then fused to compare with the point's occupancy.

Corresponding point retrieval

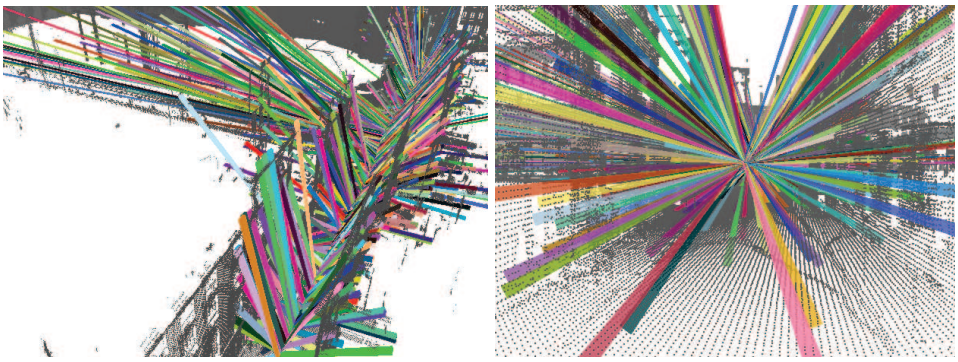


Figure 3.10 – Corresponding point search with prism-shaped rays using *K*-d tree (points in gray, sub-sampled rays in random colors): (left) perspective view and (right) view along trajectory.

For each ray in the reference data, a prism is built according to the local cylindrical frame. It is then used to retrieve the points that are inside the scope of this ray. The problem is to query a *K*-d tree of 3D points, which is built on the target point cloud, with a convex polyhedron (prism) (Figure 3.10). The bounding box (BBox) of each non-leaf cell of the *K*-d tree is checked whether it intersects with the prism. In 3D, it is the intersection between two convex polyhedra: a cell BBox B and a prism P. They do not intersect when there exists a separating plane lying in between which is (i) parallel to a face of B, (ii) parallel to a face of P, or (iii) parallel to an edge of B and an edge of P (Greene, 1994).

Separating planes in the three cases are estimated: (i) plane separation supported by a face of B amounts to testing if all the vertices $p \in P$ are on the same side of the face. In fact, only the nearest vertex to the face is necessary to be verified. Here, the nearest vertex means the one with the smallest coordinates w.r.t. the face normal. The nearest vertex $p_N \in P$ for each face of B is tested; (ii) similarly, for each face of P, the normal direction and the face plane are constructed. And the nearest vertex $b_N \in B$ is checked whether it crosses the plane; (iii) separating planes are constructed by an edge of B and an edge of P. For each edge $E_P \in P$, three planes that are passing through E_P and parallel to the three axes of B are constructed. Then the nearest vertex $b_N \in B$ for each of the three planes is checked whether it is on the same

side as P. In any of the cases, if the nearest vertex does not cross the separating plane, B and P do not intersect with each other.

To gather points of the *K*-d tree contained in the query prism, the *K*-d tree is traversed from the root by pruning sub-trees whose BBoxes do not intersect with P. Finally, a *point-in-convex-polyhedron* test is performed on all the points of the traversed *K*-d tree nodes to filter the exact list of points inside P. Based on test performances, approximate point query without case (iii) is even more computationally efficient because case (iii) is rather complex. It is the trade-off between pruning more sub-trees and conducting more *point-in-convex-polyhedron* tests. For each point inside the prism, the occupancy at the location of this point is then modelled according to the ray in the local reference frame.

Occupancy fusion

A target point can be inside of multiple ray prisms, each of which contributes its own occupancy. Thus the overall occupancy at the point location needs to be fused. Occupancy is fused by the DST. Dempster's rule of combination is as follows:

$$m_{1,2}(A) = (m_1 \oplus m_2)(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} (m_1(B) \cdot m_2(C)) \tag{3.6}$$

where $B \in 2^X$, $C \in 2^X$ and K is the conflict between two mass sets:

$$K = \sum_{B \cap C = \emptyset} (m_1(B) \cdot m_2(C)) \tag{3.7}$$

In our case, $\{e\}$ and $\{o\}$ are the subsets of $\{u\}$ because space can be both *empty* or *occupied* when the state is *unknown*. The combined occupancy of two rays $m_1 \oplus m_2$ is:

$$m_1 \oplus m_2 = \left\{ \begin{array}{c} e_1 \\ o_1 \\ u_1 \end{array} \right\} \oplus \left\{ \begin{array}{c} e_2 \\ o_2 \\ u_2 \end{array} \right\} = \frac{1}{1-K} \left\{ \begin{array}{l} e_1 \cdot e_2 + e_1 \cdot u_2 + u_1 \cdot e_2 \\ o_1 \cdot o_2 + o_1 \cdot u_2 + u_1 \cdot o_2 \\ u_1 \cdot u_2 \end{array} \right\} \tag{3.8}$$

in which $K = o_1 \cdot e_2 + e_1 \cdot o_2$. The combination rule is commutative and associative, so the order of combination is arbitrary.

Then for a given reference point P, which lies inside the scope of I rays R_i $(i \in I)$, the overall occupancy at this location is updated by combining all the rays' occupancies:

$$m(P) = \bigoplus_{i \in I} m(R_i) \tag{3.9}$$

An example of ray occupancy fusion is given in Figure 3.11. Due to large amount of data, Lidar point clouds are usually cut into small blocks and stored separately in different files. One object can be scanned from different perspectives, therefore points from the same object will be stored in several files. Since Dempster's rule is associative, we can process any file in any order. The *K*-d tree keeps the processing time at a relatively low level (tens of minutes instead of hours).

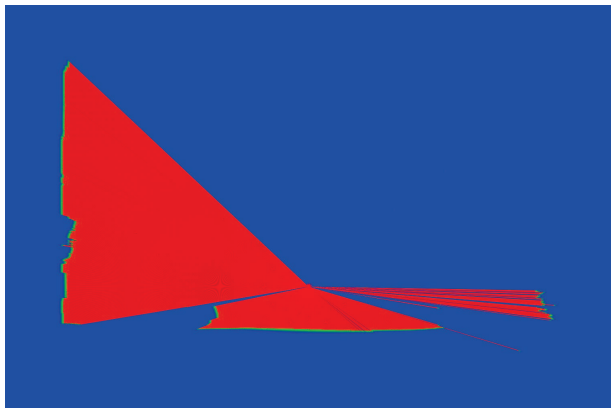


Figure 3.11 – An example of ray occupancy fusion (red: empty, green: occupied, blue: unknown).

3.3.4 Consistency assessment between different epochs

To detect the changes between two epoch data, we define consistency relations between their occupancies. They are *conflicting* when one is *empty* whereas the other one is *occupied* or vice versa, *consistent* when they have the same occupancy state, and *uncertain* if one is *unknown* whereas the other is known. The consistency relations between two datasets, target (E_1, O_1, U_1) and reference (E_2, O_2, U_2), are defined as:

$$\begin{aligned} Conflicting &= E_1 \cdot O_2 + O_1 \cdot E_2 \\ Consistent &= E_1 \cdot E_2 + O_1 \cdot O_2 + U_1 \cdot U_2 \\ Uncertain &= U_1 \cdot (E_2 + O_2) + U_2 \cdot (E_1 + O_1) \end{aligned} \tag{3.10}$$

To compute the consistency relations, one simple method is to voxelize space, compute the occupancy of each dataset on each voxel using the same size, and then compare the occupancy values of two registered datasets on every voxel. This method provides a consistency result on the whole scene at once. However, it is computationally expensive. There is no need to compute the occupancy over the whole space since we are interested in changes (conflicts) which, based on its definition (Equation 3.10), only occur at places that are empty at one epoch and occupied at another. A point gives the evidence that the space is occupied. So to know if the point has changed or not, we only need to compute consistency at the location of this point. The result of our algorithm is therefore the consistency information (*consistent*, *conflicting*, *uncertain*) of each point, which is more straightforward than representing this information on a voxel grid. An occupancy voxel grid will only be useful when the scene is scanned a high number of times when the number of acquired points becomes higher than the number of voxels required to model the scene with sufficient resolution.

According to Section 3.3.2, the occupancy at a target point is $m(P) = (e_1\ ,\ o_1\ ,\ u_1) = (0,\ 1,\ 0)$. After considering uncertainties, the maximum *occupied* value shifts slightly behind to P_s, $m(P_s) = (e_1^{'}\ ,\ o_1^{'}\ ,\ u_1^{'}) \simeq (0,\ 1,\ 0)$ (Xiao et al., 2013). Then to obtain the consistency information of this point, we need to compare it with the occupancy of the reference dataset at P_s, which is the combination of occupancies of neighbouring rays. The consistency relations at P_s are as follows:

$$\begin{aligned} Conf(P_s) &= e_1^{'} \cdot O_2 + o_1^{'} \cdot E_2 && \simeq E_2 \\ Cons(P_s) &= e_1^{'} \cdot E_2 + o_1^{'} \cdot O_2 + u_1^{'} \cdot U_2 && \simeq O_2 \\ Unce(P_s) &= u_1^{'} \cdot (E_2 + O_2) + U_2 \cdot (e_1^{'} + o_1^{'}) && \simeq U_2 \end{aligned} \tag{3.11}$$

If the location at P_s is empty in the reference data, then it conflicts with the occupancy of the target point, which indicates a change. If the location is also occupied, it is consistent, i.e., there is no change. If the location is unknown, it means that either the point is in the shadow or there are no counterparts in the reference data, thus it is uncertain whether it has changed or not.

3.4 Combination with distance-based change detection

Occupancy-based approach may detect false changes on objects that are penetrable, e.g. trees, fences, because rays can pass through them, meaning the space around them is empty. Thus points on these objects will conflict with rays that travelled through. Hebel et al. (2013) treat this situation separately using different parameters for vegetation which is pre-classified automatically. In our case, we have no *a priori* class information on the data.

Conventional point-to-point distance-based methods may avoid this failure because neighbouring points can be found for points on these objects from another epoch if the distance threshold is large enough. Direct point-to-point comparison suffers from irregular point density. To minimize the effect of anisotropy, a local surface model or a small point cluster is recommended (Girardeau-Montaut et al., 2005). However the size or the number of points for the model can be tricky to chose. Due to the complexity of street objects, we use the point-to-triangle distance.

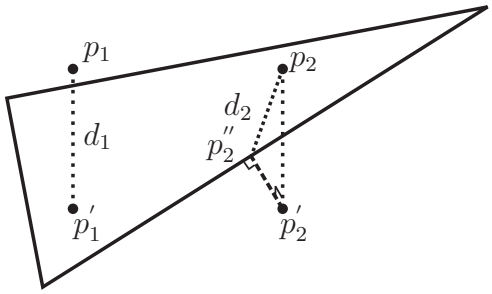


Figure 3.12 – Point-to-triangle distance (PTTD) in 3D: if the projection is inside the triangle as $p_1^{'}$, the projection distance is the PTTD, $d_1 = p_1 p_1^{'}$; whereas if it is outside as $p_2^{'}$, the distance to the nearest edge of the triangle is considered as the PTTD, $d_2 = p_2 p_2^{''}$.

Distance to the nearest triangle in the reference data is computed instead of the nearest point. A certain number of nearest points are retrieved first from the reference data, then a local Delaunay triangulation is implemented on these points. The number of points is not vital because we only look for the nearest triangle, which should not be different as long as the number is not too small. Afterwards, the nearest triangle is found considering the distance from the target point to each triangle in 3D. If the target point projects inside the triangle, the distance to the projection is the PTTD. If the projection is outside the triangle, the distance to the nearest edge of the triangle is considered as the PTTD (Figure 3.12). Then points that are changed are detected by thresholding the minimum distance (d_{min}).

$$m(P) = \begin{cases} 1 & \text{if } PTTD < d_{min}, \\ \bigoplus\limits_{i \in I} m(R_i) & \text{else.} \end{cases} \tag{3.12}$$

The occupancy-based and PTTD-based methods are integrated as shown in Equation 4.9. If a point has nearby corresponding triangles ($PTTD < d_{min}$), the occupancy at its location is set to be 1. Otherwise, the occupancy is the fusion of neighbouring rays' occupancies according to Equation 3.9. Points that are changed should be far from other points as well as be *conflicting* with neighbouring rays.

3.5 Experiments and result

Two experimental data at different sites were acquired in Paris by the *Stereopolis* MMS (Paparoditis et al., 2012) using two RIEGL laser scanners, LMS-Q120i and VQ-250. Both rotate perpendicularly to the vehicle trajectory, scanning profiles of the street environment. One has only 80° vertical field of view (FOV) (data referred to as Data 1), whereas the other scans a full circle (360° FOV) (Data 2). Each site has two epoch data. Datasets are registered by the method proposed by Monnier et al. (2013) with about 0.1 m accuracy using non-rigid iterative closest point (ICP). Point normals are estimated by the method from Demantké et al. (2011) in which the optimal spherical neighbourhood is automatically defined. Three dimensionality (linear, planar, volumetric) features are computed at multiple radius scales. Then the best radius is automatically selected when one feature is mostly dominant over the two others. Point normals are given by the principle component analysis (PCA) on the optimal neighbourhood.

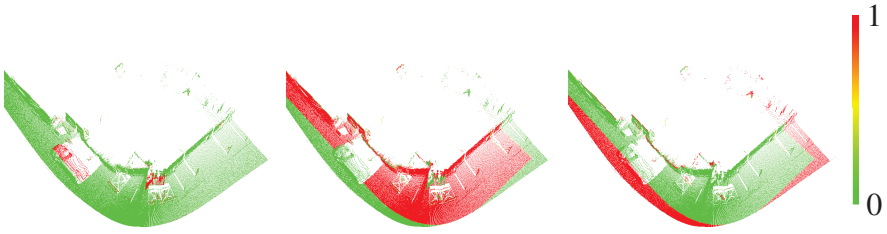
Many variables and parameters have been defined in previous sections, Table 3.1 summarizes the definition of notations and parameter settings. Parameters are set according to the scanning geometry. The scan ray rotates perpendicularly to the vehicle trajectory. Rotation angle and sensor origin are recorded for each point. The range accuracy σ_m depends on the sensors. The distance between scan lines varies because it depends on the vehicle speed. Thus λ_t should be large enough to avoid the gap and meanwhile not too large either. The average distance between scan lines is about 0.1 m . λ_s depends on λ_θ which covers the gaps along the rotating plane, therefore it should be larger than half of the angular resolution. λ_n defines the occupancy scope in the normal direction. It should be bigger than the registration error ($\sigma_r =$ 0.1 m), however not too big to avoid over-modelling. Based on experimental results, it is set as 0.3 m.

If the time span is large between two epochs, changes may be found on buildings or street facilities which is useful for database updating. If the time span is short, e.g. two epochs scanned in the same day, most changes are caused by moving objects such as cars or pedestrians which is also important because these movable objects need not to be modelled or stored in a database. Data with clean street environment is of great importance for street inventory and especially mobility assessment. Movable objects may be considered as obstacles if they are located on pavements (Serna et Marcotegui, 2013).

Partial results of the first dataset (Data 1) using occupancy-based and PTTD-based methods are presented in our work (Xiao et al., 2013), in which many urban objects, e.g. garbage bins, chairs, motorcycles, are illustrated. Each point has the three consistency elements ranging from 0 to 1. The largest value among them can be considered as dominant hence it is the final consistency result. Figure 3.13 shows the result of consistency assessment.

Table 3.1 – Summary of notation definitions and parameter settings

Notations	Definitions and settings
Occupancy notations:	
(e, o, u)	abbreviations of the masses (empty, occupied, unknown)
(e_n, o_n, u_n)	(e, o, u) in the normal direction
$(e^{'}, o^{'}, u^{'})$	(e, o, u) after convolving with uncertainty functions
f_n, f_s, f_t	occupancy function in normal, scan line and trajectory directions
(E, O, U)	(empty, occupied, unknown) for a whole epoch data after fusion
Geometric notations:	
r, θ, t	local reference frame of ray, angular and trajectory directions
n, s, β	local geometry between ray and point normal (Figure 3.9b)
Data features:	
σ_m, σ_r	variance of measurement and registration
$\sigma_n, \sigma_s, \sigma_t$	variance of uncertainty in each direction
Physical parameters:	
λ_s	occupancy in scan line direction $\lambda_s = r \cdot \lambda_\theta / cos\beta$
Tunable parameters:	
λ_n	occupancy in n direction, thickness of urban objects (0.3 m)
λ_θ	occupancy in θ direction, half of the angular resolution
λ_t	occupancy in t direction, covers gaps between scan lines (0.1 m)



(a) Conflicting (b) Consistent (c) Uncertain

Figure 3.13 – Consistency assessment results

Figure 3.14 shows the final result of change detection. Similar to the occupancy triangle, red, green and blue demonstrate conflicting, consistent and uncertain respectively. On site 1, the man and the billboard were detected. The van was mostly detected as change and a small part as uncertain because in the second data the MMS vehicle passed through the position of the volume that the van occupied in the other data and only a part of the volume was scanned. On site 2, the car was detected as change. And the place where the van from the second scan was parked was detected as uncertain since this part was in the shadow.

PTT distance-based result is depicted in Figure 3.15, the changes at the two sites were also detected correctly. However, all the changes are labelled with the same color (red). Thus it is impossible to distinguish between change and occlusion or absence of counterpart. Occlusions are commonly encountered in MMS data. It is essential to distinguish real changes and those caused by them. However, they were falsely detected as changes by the PTT method as shown on site 2, points in the shadow of the van were considered the same as other changes. If the

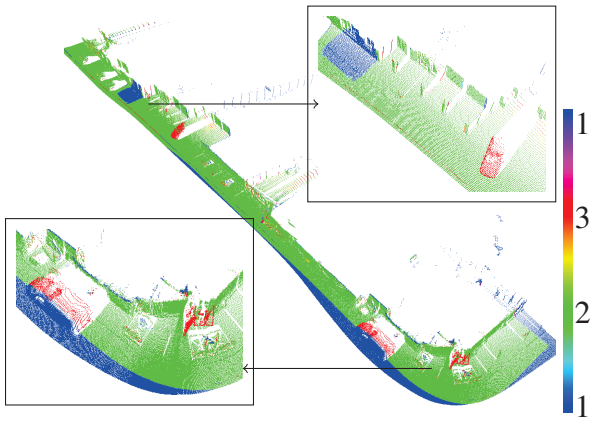


Figure 3.14 – Occupancy consistency difference (1: uncertain; 2: consistent; 3: conflicting)

two scans do not have the same boundary, points that have no counterparts in the other dataset will also be considered as change. On site 1, all the points on the van were detected as change, including the points that were not in the scanning scope of the other dataset. In addition, the global threshold (0.3m) may not be suitable for all datasets when there are more than two multi-temporal datasets.

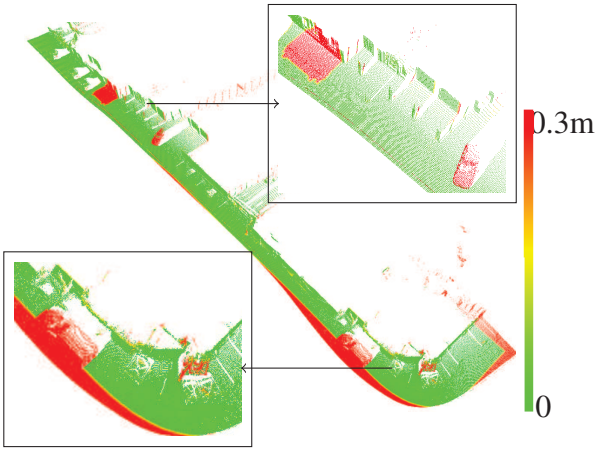
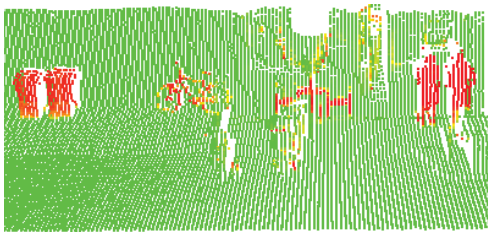


Figure 3.15 – Point to triangle distance

To verify the change detection results, some urban objects were moved for different scans. As shown in Figure 3.16, large objects, e.g. garbage bins, pedestrians, a motorcycle, were well detected. Hollow objects, e.g. a bicycle, chairs and a table, and even a small pole-like object, about 20cm wide, were detected. The results indicate that the detection is reliable and the detectable changes can be quite small.

The minimum detectable size of changed objects depends on point density. To avoid false detections, a single point or a few points in one line are eliminated. Thus the object size should be wider than at least twice the gap between scan lines and also tall enough to be detected. Registration quality will directly affect the change detection results. With high point density, many false changes will be found even small objects can be detected if registration quality is low.

Figure 3.17 shows the comparison with the improved occupancy-based method by considering point normals. Previous method performs weak for the points whose normal direction are extremely different from the directions of their rays, e.g. ground points that are far from the

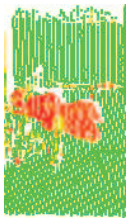


(a) Garbage bins, bicycle, chairs and table, pedestrians

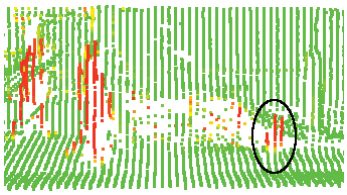
(b) Garbage bins

(c) Bicycle

(d) Chairs and table



(e) Motorcycle

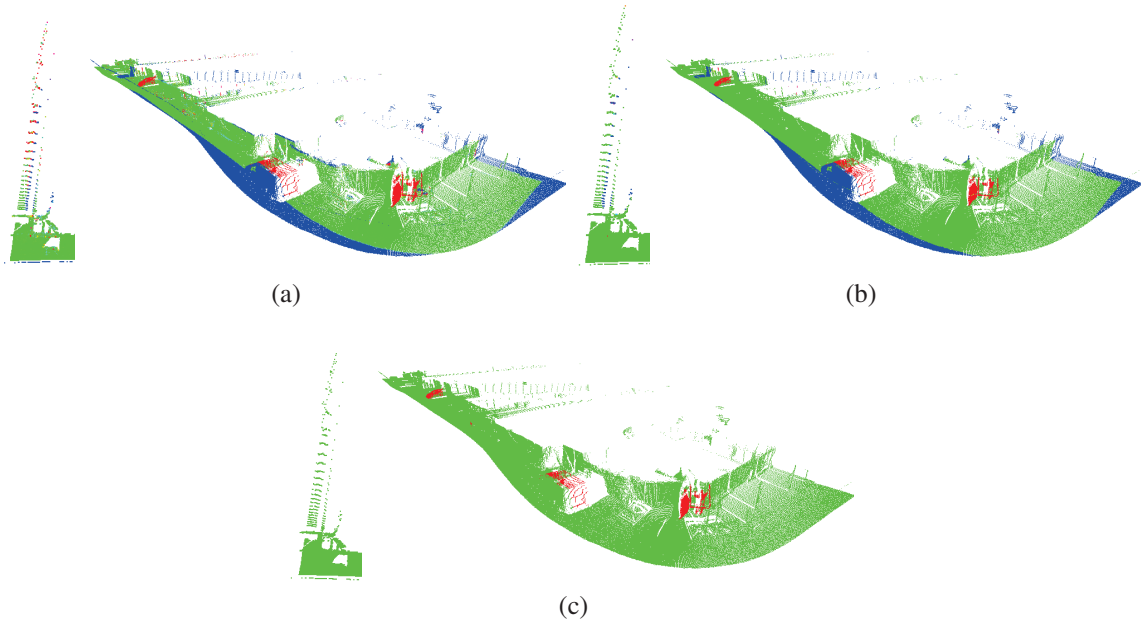


(f) Pedestrians and pole-like object

(g) Pole-like object

Figure 3.16 – Detected changed urban objects

laser center.



(a)

(b)

(c)

Figure 3.17 – Change detection result of Data 1 with 301903 points. (a) occupancy-based result (changes in red); (b) improved by considering point normals. Especially in the enlarged part (on the left), ground points that are far from the laser center are rectified (green in (b), red in (a)); (c) hand labelled ground truth.

Figure 3.18 depicts the study site of Data 2 and some detected movable objects, e.g. pedestrians, bicycles and cars, using the occupancy and PTTD combined method.

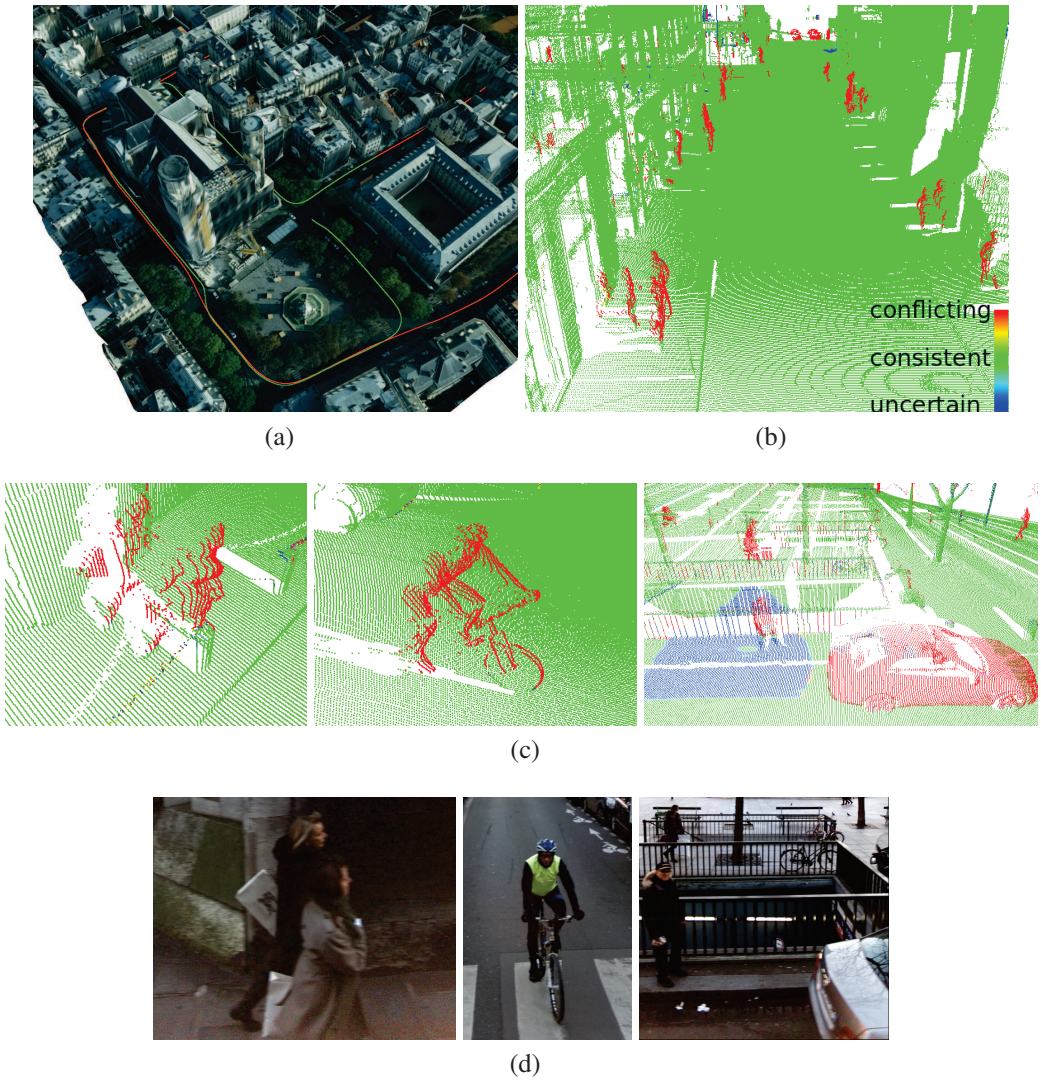


Figure 3.18 – Study site and some change detection examples. (a) trajectories of two epoch datasets in red and green around a cathedral; (b) one site of detected changes (red); (c) successfully detected pedestrians that are close to façade and bollards, a cyclist and a car and the shadow (blue) of another car in the reference data. Notice the correctly detected pedestrian in the middle of the shadow; (d) corresponding images (the laser scanner and cameras have different points of view).

Figure 3.19 illustrates the results of a large area using different methods. The occupancy-based method successfully differentiates points that are occluded or have no neighbouring rays with real changed points. Occluded points are detected as *uncertain* (blue), and unchanged points are *consistent* (green), whereas changed points are *conflicting* (red) with reference data. Many points on the ground that are far from the laser center are falsely detected as changed (Figure 3.19b) using previous occupancy-based method. There are rectified by considering point normals (Figure 3.19c). However, many points on trees and fences are detected as *conflicting*. The PTTD-based method is also able to detect all the changes (Figure 3.19d). It has better performance for penetrable objects, because neighbouring triangles can be found from the reference data. However, all points in shadows are incorrectly detected as changed. The integrated approach combines the advantages of both (Figure 3.19e). Occlusion changes are distinguished from real changes and false detections on penetrable objects are excluded. No-

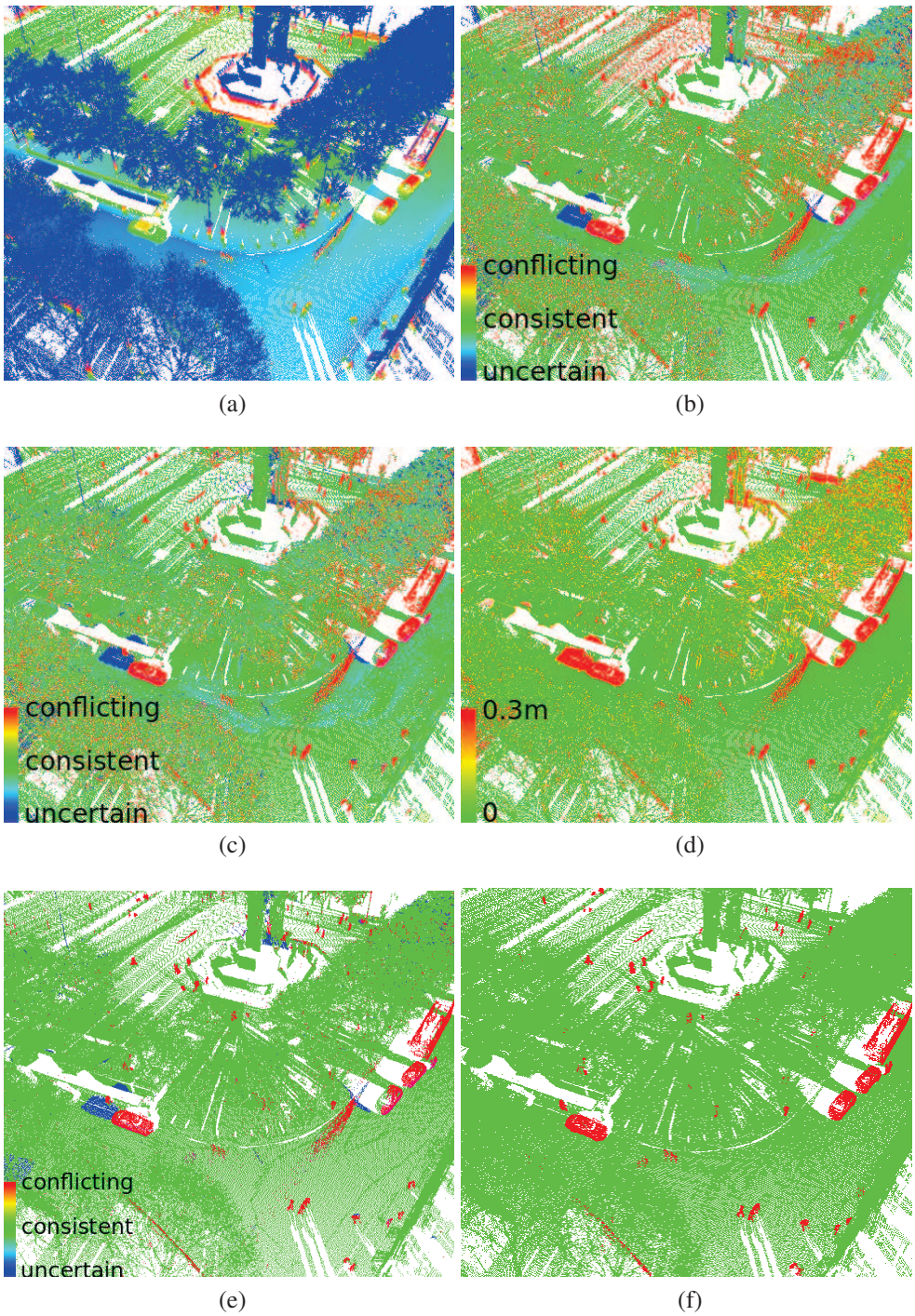


Figure 3.19 – Change detection results by different methods. (a) original data coloured by height; (b) occupancy-based method: occluded points (blue) are distinguished from real changed points (red), many points on the ground are incorrectly detected as changed; (c) occupancy-based method with point normals: some points on trees are still incorrectly detected as changed; (d) PTTD-based method: occluded and real changed points are both detected as changed (red); (e) combination method: real changes are differentiated from occlusions, incorrectly detected points on trees are excluded; (f) manually labelled ground truth (changes in red).

tice that there are many false detections on trees in Figure 3.19c and 3.19d, whereas much less in Figure 3.19e. This is because only points that are both conflicting with reference rays and far from nearest triangles are detected as changed. Figure 3.19f shows the manually labelled ground truth. Some points on a fast moving object on the street have not been labelled due to the irregular structure. However they are correctly detected.

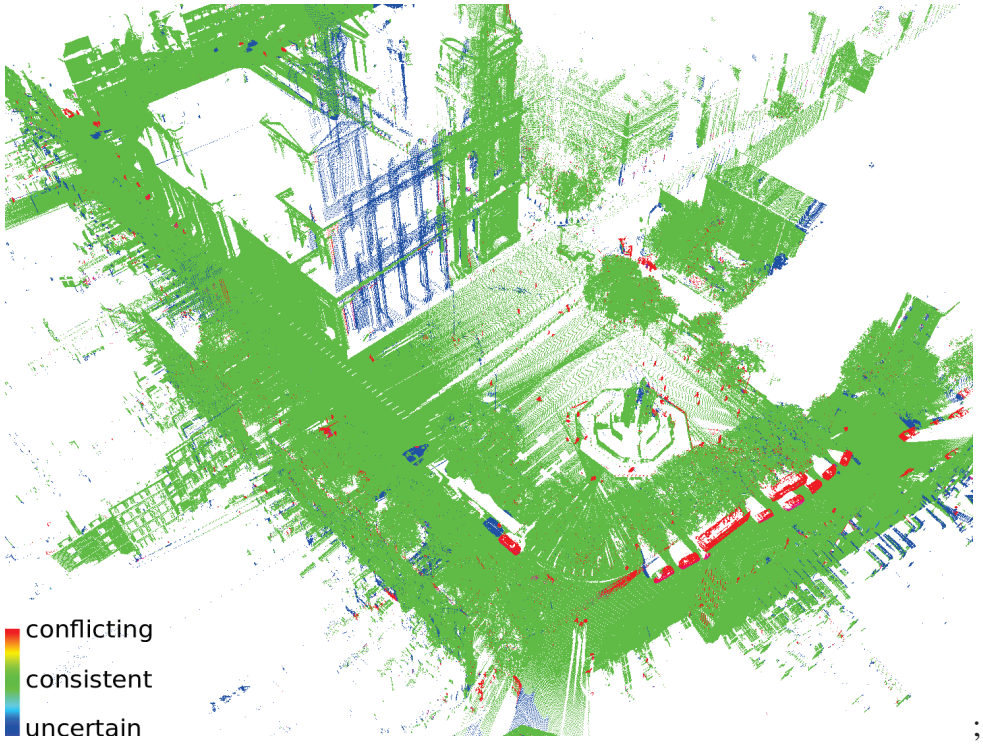


Figure 3.20 – Street environment change detection result of the combination of occupancy-based and PTTD-based methods: real changes are differentiated from occlusions, incorrectly detected points on trees are excluded (data sub-sampled due to huge size).

3.6 Evaluation and discussion

Since our method detects changes for each point, the accuracy is firstly assessed at point level. Ground truth has been labelled manually. Four methods, i.e. PTTD-based method, occupancy-based method without point normals (Occ-Norm), occupancy-based method with point normals (Occ+Norm) and its combination with PTTD-based method (Combined = PTTD+Occ+Norm), are evaluated against the ground truth. True positive (TP) is the number of changed points correctly detected. False positive (FP) is the number of unchanged points that are detected as changed. False negative (FN) is the number of changed points detected as unchanged. True negative (TN) is the number of unchanged points correctly detected. The majority of the points are unchanged and more than 97.8 % of the whole dataset are correctly detected as unchanged, so it is unnecessary to quantify TN rate and the overall accuracy (including TN). Recall (R), precision (P), Jaccard coefficient (JC) (Radke et al., 2005) and F_1 score are quantified. Evaluation

Method	Data 1				Data 2			
	R	P	JC	F_1	R	P	JC	F_1
PTTD	0.917	0.105	0.104	0.189	0.865	0.277	0.265	0.419
Occ-Norm	**0.987**	0.760	0.753	0.859	**0.902**	0.363	0.349	0.517
Occ+Norm	0.963	0.875	0.846	0.917	0.876	0.467	0.438	0.609
Combined	0.907	**0.946**	**0.862**	**0.926**	0.812	**0.857**	**0.715**	**0.834**

Table 3.2 – Accuracy assessment of methods using PTTD, occupancy, occupancy with point normals (Occ+Norm) and its combination with PTTD.

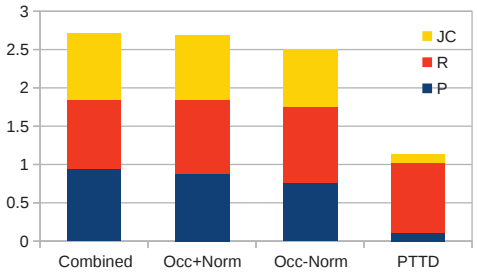


Figure 3.21 – Change detection results of Data 1 using different methods. PTTD-based method performs the worst due to occlusions. Considering point normals improves the occupancy-based method. The combined method performs the best.

results of the two data are given in Table 4.1.

$$\begin{aligned} R &= TP/(TP+FN) \\ P &= TP/(TP+FP) \\ JC &= TP/(TP+FN+FP) \\ F_1 &= 2\cdot P\cdot R/(P+R) \end{aligned} \tag{3.13}$$

The two datasets give similar results. The occupancy-based method shows highest recall, and the combined method dominates all the other values. In general, the values are higher for Data 1 than for Data 2, because Data 1 is simpler and smaller. Especially, the occupancy-based method has much higher precision in Data 1 than Data 2 because there are less penetrable objects. The improved occupancy-based method (Occ+Norm) significantly increases the precision, however decreases the recall, which can be caused by normal estimation errors. Both occupancy-based and PTTD-based methods give a low level of precision, especially for Data 2. However the reasons are different. For the occupancy-based method, errors come from the points on penetrable objects. As depicted in Figure 3.19c, there is a large number of falsely detected tree points. For the PTTD-based method, it is due to occlusion. As shown in Figure 3.20, a big part of the façade (blue) is occluded. Therefore, the combination of them avoids these drawbacks resulting in a high level of precision. The recall of combined method is smaller than the others because points need to meet both the criteria as mentioned in the previous section.

Figure 3.21 illustrates the add-up of precision, recall and JC values. It is clear that occupancy-based method is improved by considering point normals. And the combination with PTTD helps to further improve the final result. Figure 3.22 shows the results of each subset of Data 2 (from number 1 to 11) and the whole dataset (number 12). Due to large amount of data, the laser scanner automatically stores acquired points into different files. Each subset corresponds to one

file and contains three million points. Subsets 1 to 7 are acquired behind and beside the cathedral in Figure 3.18(a) and there are barely penetrable objects on the narrow streets as shown in Figure 3.18(b). Whereas the others contain many points on trees that are in the front of the cathedral, such as in Figure 3.19. The results varies due to different street scenes. In general, occupancy-based results are better than PTTD-based when there are few penetrable objects (subsets 1 to 7), and vice versa. Considering point normals improves the occupancy-based results when point normals differ from their ray directions (subsets 2, 8, 9, 10 ,11), however it also hinders the results to some extent depending on the normal estimation accuracy (subsets 1, 3, 4, 5, 6). The combined method has the best results for all the subsets except the first one, because the PTTD-based result is extremely bad. The overall result (dataset 12) shows the same pattern as in Figure 3.21, meaning the combined method performs the best in a complex urban street environment. However, individual methods have strengths at certain street scenes.

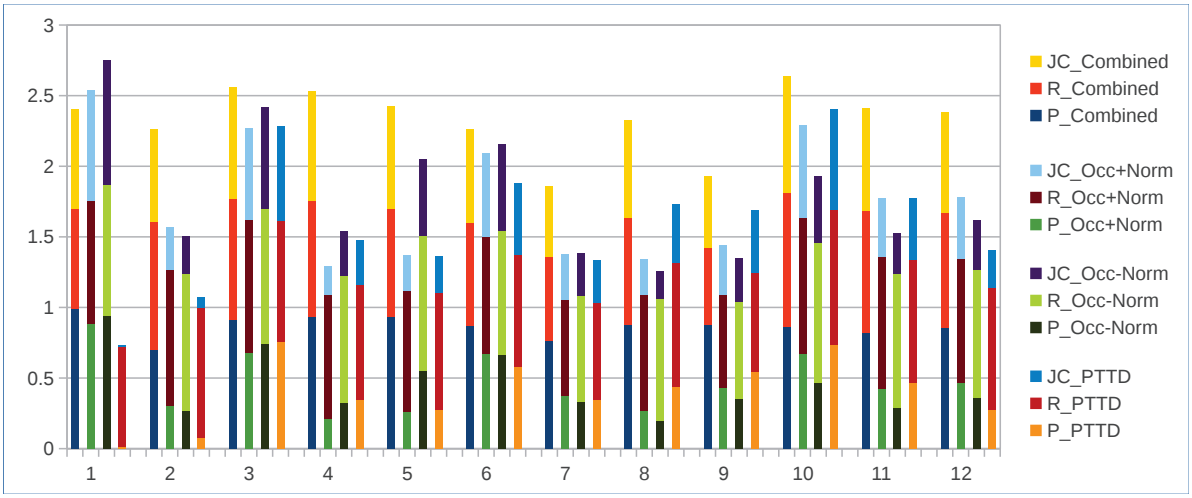


Figure 3.22 – Change detection results of Data 2. Number 1 to 11 are the 11 subsets, number 12 is the whole dataset. Results of subsets vary according to different scenes. Result of the whole dataset is consistent with Data 1.

Failure of detection is caused by (i) object replaced by another: the object has changed but locates at the same location as a previous one (Figure 3.23a); (ii) object points are near the ground: points on the ground give evidence of consistency to nearby points and are also within the PTTD threshold (Figure 3.23b); (iii) noise: some points presented in the data are too sparse to represent meaningful street objects, e.g. objects that are too far from the laser scanner.

To evaluate the accuracy at object level, we consider an object as detected if a large portion (90%) of its points are correctly detected. In Data 1, three changed objects, a pedestrian, a van and a car, are all correctly detected. In Data 2, a total of 229 objects are observed in the target dataset. All of these objects are successfully detected except one (Figure 3.23a), which is partially detected (lower than 90%) because another object from the reference dataset is located at the same space. No unchanged objects are incorrectly detected. Most of the falsely detected changed points are sparsely distributed so that they can be treated as noise in the following process. For instance, after clustering all the changed points into objects, those objects with a small number of points can be eliminated.

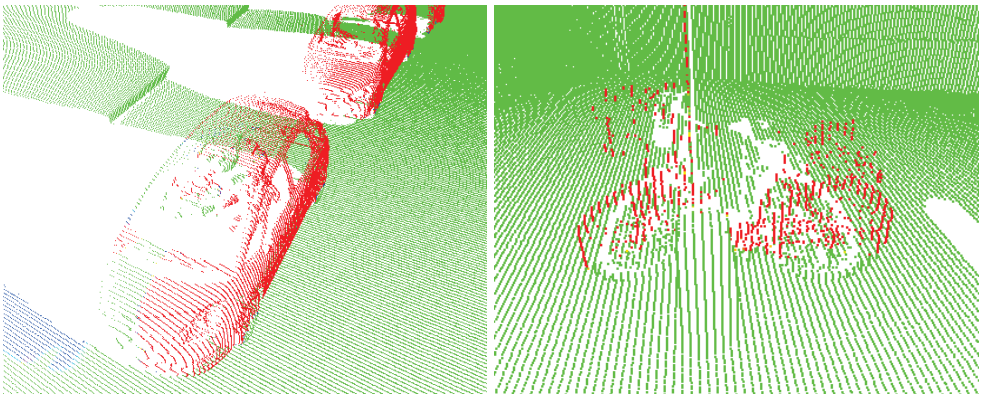


Figure 3.23 – Examples of false detections. (a) a car has been partially detected as changed whereas the other part as unchanged because this part is overlapped with another car in the reference data; (b) a bicycle is leaning against a pole and the majority part of it is correctly detected, however the very lower parts of the wheels are misdetected as unchanged; few points on the pole are also misdetected as changed.

3.7 Conclusion

A general framework for change detection in urban street environments was presented. The method combines both advantages of occupancy grids from robotics and conventional distance-based methods. The former indicates occluded areas thus helps to distinguish occlusions and points without counterparts from real changes. The latter is robust to penetrable objects which are self-conflicting in terms of occupancy. They are complementary to each other.

The occupancy-based method is based on the physical laser scanning mechanism. The prerequisite is to know the origin of each scanned point. A local cylindrical reference frame is built for each ray and its point. Then the occupancy of space is modelled around the ray and the point. Gaps between rays are interpolated by treating the ray shape as a triangular prism instead of a cone. Thus it is robust to irregular point density. Point normals are considered to improve the occupancy modelling around the point and its surface. Changes are detected directly at point level without voxelization. A distance-based method, PTTD is chosen to optimise the occupancy-based result. PTTD keeps the flexibility to be suitable to complex street objects, and also the robustness to irregular point distributions. Urban object changes are automatically detected at point level in complex street environments.

The limitation of the proposed method is that objects replaced by another can not be detected. Future work will focus on fine scale feature-based change detection.

Chapter 4

Simultaneous detection and tracking of pedestrian

Contents

Pedestrian traffic flow estimation is essential for public place design and construction planning. Traditional data collection by human investigation is tedious, inefficient and expensive. Velodyne laser scanner, which scans surroundings repetitively at a high frequency, has been increasingly used for 3D object tracking. In this chapter, a simultaneous detection and tracking (SDAT) method is proposed for precise and automatic pedestrian trajectory recovery. First, the dynamic environment is detected by assessing the consistency of space occupancy in a spherical reference system centred at the laser scanner location. Then, all the rest points on moving objects are transferred into a space-time (x, y, t) coordinate system. The pedestrian detection and tracking amounts to assign the points belonging to pedestrians into continuous trajectories in space-time. We formulate the point assignment task as an energy function which incorporates the point evidence, trajectory number, pedestrian shape and motion. A low energy trajectory will well explain the point observations, and have plausible trajectory trend and length. The method inherently filters out points from other moving objects and false detections. The energy function is solved by a two-step optimization process: tracklet detection in a short temporal window; and global tracklet association in the whole time span. Results suggest that the proposed method can automatically recover the pedestrians trajectories with accurate positions and high completeness.

4.1 Introduction

It is common that public places, e.g. squares and concourses, need to be renovated, expanded or redesigned. One of the main factors to be considered is the pedestrian traffic flow. Usually, field data for the flow estimation are collected by human visual counting. The process is expensive and inefficient. In this chapter, we investigate the potential of using laser scanning techniques for automatic pedestrian trajectory estimation. The objective is to detect the moving pedestrians and recover their trajectories. Pedestrian, or in general, moving object detection and tracking has been studied in both computer vision and robotics for various applications, e.g. surveillance, autonomous driving. Target's behaviour is mostly studied individually in order to predict its future move. Here, we aim for large scale long term monitoring in order to study the general moving patterns. Accurate geo-located moving patterns can be incorporated into GIS platforms for precise agent-based modelling.

Moving object detection and tracking (MODAT) have been traditionally studied using image sequences and video. Objects are detected in the camera reference frame or 2D world coordinate system (Milan et al., 2014). Stereo matching enables us to detect and reconstruct objects in 3D, then their 3D trajectories can be reconstructed (Schindler et al., 2010). Whereas the field of view (FOV) is still limited by the stereo cameras. Panoramic image stereo-based tracking is still immature (Koyasu et al., 2001). With the development of laser scanning technology, especially the laser scanners with 360° horizontal FOV, e.g. Velodyne HDL-64E, 3D moving object detection and tracking using laser range data has become increasingly popular (Shackleton et al., 2010; Kaestner et al., 2012; Moosmann et Stiller, 2013).

A Velodyne scanner is usually composed of a number of vertically configured laser sensors covering a wide enough vertical FOV (depending on the number of sensors). It rotates around the vertical axis such that it generates a panoramic view of the surroundings. The rotation frequency ranges from 6 to 15 Hz. Same as other types of laser scanners, the Velodyne range data are directly recorded in 3D in the form of 3D point clouds. The measurement distance ranges from 2 m to 100 m, and the range accuracy is about 2 cm. The Velodyne scanner constantly scans the full surroundings hence it is an ideal technique for MODAT, especially when the area of interest is located around the sensor (Moosmann et Stiller, 2013). It is commonly mounted on a mobile mapping system (MMS), together with optical cameras, for the purposes of environment perception, and simultaneous localization and mapping (SLAM) (Moosmann et Stiller, 2011). Stereo image-based MODAT can be complementary, but it is difficult to incorporate with laser scanning geometry.

A popular MODAT method is tracking-by-detection, where the moving objects are detected first in each frame, then the trajectories are reconstructed by associating plausible candidates (Andriluka et al., 2008; Wu et Nevatia, 2007). Objects are typically detected by extracting discriminative features from the segments, then classifying them into objects of interest. However, the detection results can be affected by many factors, such as occlusion, miss-classification. To avoid miss-classifications, some try to track generic objects without knowing the specific classes, which, however, can have limited applications (Kaestner et al., 2012). In both cases, the detection accuracy will limit the overall tracking performance.

In this chapter, we aim to reconstruct the pedestrian trajectories without specifically detecting the objects for each frame. As for Velodyne laser scanning data, usually a full turn (360° view) is treated as a frame, which means that, in our case, the data do not have to be partitioned into frames. We support the idea that partial data acquisitions, which are commonly

caused by self-occlusion and limited FOV, should be retained for better trajectory recovery. Instead of detecting the individual object, we detect its trajectory directly by assigning the point data to a trajectory hypothesis. The data assignment is formalized as an energy function which incorporates the point evidence, trajectory number, pedestrian shape and motion. First, points belonging to moving objects are separated from the static environment/background (Section 5.3.3). Then pedestrians and their trajectories are detected simultaneously from the moving points which include false alarms and points on other moving objects, e.g. cars (Section 4.3.2). Experiments are carried out on two datasets, one of which is publicly available. Both qualitative and quantitative results are illustrated (Section 5.4). Some discussions (Section 4.5) are presented before the conclusion and perspective (Section 4.6).

4.2 Related work

MODAT is a classical question in both computer vision and robotics. Imageries are still the primary data source for object tracking in computer vision, whereas laser scanning technology is, in general, getting more and more popular in robotics. Thus pedestrian tracking using images in computer vision and using laser scanning data are reviewed respectively.

4.2.1 Pedestrian tracking in computer vision

Object tracking using optical cameras has been studied in both camera reference frame (2D) and world coordinate system (2D and 3D), and it is applied to many applications, e.g. surveillance, collision prevention, driving assistance. The general pipeline is first to detect the moving object in each frame, then their future positions are predicted and updated for online tracking, or their complete trajectories are recovered by data fitting for offline tracking.

Schindler et al. (2010) detect and track pedestrians from a moving stereo rig. First, pedestrians are detected from each stereo pair in 3D. Then trajectory candidates are generated by a Extended Kalman Filter taking into account the object motion and appearance. An optimal subset of the trajectories are selected by maximizing a quadratic binary expression using multi-branch optimization (Schindler et al., 2006).

Milan et al. (2014) track multi-targets by continuous energy minimization. The energy function linearly combines six terms: data association, appearance, motion, exclusivity, track persistence and regularization term. Six types of jump moves, namely grow, shrink, merge, split, add and remove, are used to minimize the energy function.

Milan et al. (2015) enhance the state-of-the-art tracking-by-detection method by investigating low-level information from the video. They argue that important information have been ignored during the detection process. Thus they propose to retain the ID information of each super-pixel in the entire sequence which help to bridge the trajectory gap caused by absence of detections and occlusions.

Similar investigation of pixel-level segmentation has also been proven to be beneficial for pedestrian tracking by Aeschliman et al. (2010). They propose to use an implicit fine level segmentation instead of a simple mask such as a rectangle or an ellipse. The segmentation and tracking are jointly solved in a probabilistic framework.

One major component of tracking is connecting objects' presences over time to reconstruct

their trajectories, which is normally referred to as data association. Instead of bipartite matching, Zamir et al. (2012) propose a tracker that associates object detections globally taking into account both motion and appearance using clique graphs. Plausible association are taken as nodes for a global graph. Each node is weighted by the association cost, e.g. motion continuity, appearance consistency. Then the best trajectories are composed by the subgraph whose nodes have the minimal cost. The tracker has been improved by Dehghan et al. (2015).

In the same spirit as Milan et al. (2015) and Aeschliman et al. (2010), we preserve all the original acquired information without explicit classification to prevent information loss. Graph clique is also chosen, in our case, for tracklet association because it guarantees one-to-one node connection. The optimal clique is governed by data explanation.

4.2.2 Pedestrian tracking using laser scanning data

MMS are mostly using both cameras and laser scanners. Cameras provide with richer information besides the 3D locations generated from stereo matching. They are much cheaper than laser scanners, which, however, deliver more accurate 3D geometry. They are both used for SLAM and autonomous driving. Here we focus on object tracking using laser range data.

Moving object detection and tracking using a mobile laser scanner has been studied earlier on by Lindström et Eklundh (2001). A laser scanner is mounted on a moving robot. Points on the single moving object is detected by checking the violation of the static environment. Then points from each scan is tracked by matching with the nearest neighbour.

Shackleton et al. (2010) track indoor moving people using a Velodyne laser scanner. A Classical tracking pipeline has been used. Points from each frame are segmented first, and then the segmented object hypotheses are classified so that moving people are identified. At last, the trajectories are estimated and verified by an extended Kalman filter. Reported results are promising, whereas the tested scene is rather simple.

Spinello et al. (2011) detect people in 3D using a bottom-up top-down approach. The results of conventional classification method, i.e. voting based classification after segmentation (bottom-up), are improved by a predefined volume which represents the 3D features of people (top-down). The detection is compared with spin images and template-based method, and is reported to outperform them. The detected objects are then applied to a standard multi-hypothesis tracking (MHT) procedure (Reid, 1979).

Kaestner et al. (2012) focus on generic object detection without explicit classification. Static and dynamic observations/points are segmented through a probabilistic interpretation. Then all the detected dynamic object hypothesis are tracked by optimizing the standard MHT using oriented bounding box and track splitting and merging. The tracking results are comparable to aforementioned discriminative detections (Spinello et al., 2011).

Moosmann et Stiller (2013) also track generic objects from Velodyne data. Normal and flatness features are precomputed for each points, then they are segmented based on the local convexity. A tracklet is generated for each segment/object hypothesis. And new tracklets are connected by counting the overlapping pixels in their range images.

Since we only need trajectories from pedestrians, the points on pedestrians have to be distinguished from those on static and other moving objects. One drawback using laser scanning points other than images is the lack of color information, i.e. appearance, which is important evidence for object discrimination. The point's intensity attribute is, however, affected by

various factors, e.g. object material, incidence angle, hence it is not consistent across time. HosseinyAlamdary et Yilmaz (2014) color the points by projecting them on to corresponding images. Whereas since the camera POV is limited, only a part of points are used for tracking.

4.3 Methodology

Since only moving objects are of interest in our case, we first distinguish them from the static background. The moving objects are firstly extracted in a spherical reference frame based on the occupancy information that the laser beam convey. The occupancy is modelled at the point level, so the detected moving objects are not yet clustered (Section 5.3.3). Next, all the moving points are transferred into a space-time (x, y, t) coordinate system. Object tracking amounts to cluster all the points belong to the same object in such space time cube. Points belonging to the same pedestrian are clustered regarding the unique shape and structure of a pedestrian in (x, y, t) in a short temporal window. Since the pedestrians are clustered/segmented over time, they are simultaneously detected and tracked from all the moving points. We formalize the simultaneous detection and tracking (SDAT) of pedestrian as an energy minimization problem, which is solved by a two-step optimization process (Section 4.3.2).

4.3.1 Moving object detection

Here, we differentiate between moving object and mobile object. The former refers to objects that are moving during the acquisition process. Whereas the latter defines the movability of an object, i.e. a mobile object is movable even if it is not moving during the acquisition. So a moving object is always a mobile object, but a mobile object is not necessarily moving, e.g. parked cars. Those mobile objects that stayed static during the whole acquisition process have no trajectories so that they are no detected.

Occupancy modelling for Velodyne data

The laser beam reveals the occupancy status of the scanned space, i.e. it is empty (e) between the laser center and the detected point, occupied (o) around the point, and unknown (u) behind. If an object is moving, the corresponding point position will move such that the occupancy status will change accordingly. Occupancy modelling in our previous work (Xiao et al., 2015) is extended to a spherical local reference frame. The Dempster-Shafer Theory (DST) is also used for information fusion. Here, the occupancy information fusion is adjusted to better detect the changed points on moving pedestrians. The occupancy is modelled by three mass functions:

$$m : (e, o, u) \in [0, 1]; e + o + u = 1 \tag{4.1}$$

A local reference frame representing the Velodyne data acquisition geometry is depicted as Figure 4.1. The origin O is the sensor center. Space around a ray $\overrightarrow{OP}$ is described by a vertical angle θ and a horizontal angle φ. Together with r, they compose a spherical local reference frame. The occupancy in the ray direction (e_r, o_r, u_r) is defined as (1, 0, 0) between $\overrightarrow{OP}$, (0, 1, 0) around P, and (0, 0, 1) behind P. The occupancy in horizontal and vertical directions are modelled by Gaussian functions as follows:

$$f_\theta = e^{-\frac{1}{2}(\frac{\theta}{\lambda_\theta})^2}; f_\varphi = e^{-\frac{1}{2}(\frac{\varphi}{\lambda_\varphi})^2} \tag{4.2}$$

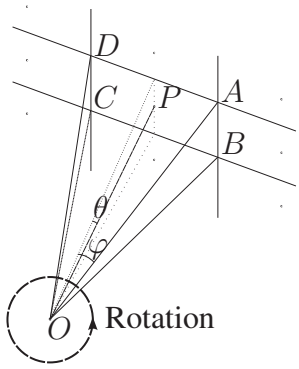


Figure 4.1 – Spherical local reference frame for Velodyne data.

in which f_θ and f_φ do not model the physical angle of the laser beam but interpolate the occupancy information between beams. Standard deviations λ_θ and λ_φ represent the mass of occupancy in corresponding directions. Sampling densities are not the same in these two directions since the horizontal sampling density depends on the rotating speed while the vertical density depends on the object's distance. In the vertical direction, lasers are equally spaced and the angle between rays is approximately 0.4°. λ_θ should be larger than half of the angle to cover the gaps in between. In the horizontal direction, angular resolution depends on the rotation speed. Similarly, λ_φ should be larger than half of the resolution. The vicinity scope of ray $\overrightarrow{OP}$ is a pyramid $O - ABCD$, which is used for K-d tree neighbouring point searching.

The overall occupancy of ray $\overrightarrow{OP}$ is formulated as the combination of all the occupancy functions in the three directions, hence is a function of the occupancy status in the 3D space with parameters r, θ, φ:

$$m(\overrightarrow{OP}) = \begin{pmatrix} e \\ o \\ u \end{pmatrix} = \begin{pmatrix} f_\theta \cdot f_\varphi \cdot e_r \\ f_\theta \cdot f_\varphi \cdot o_r \\ 1 - e - o \end{pmatrix} \tag{4.3}$$

Data fusion for moving object detection

Velodyne sensor scans the surroundings repetitively by rotating the sensor around the vertical axis. Each full 360° rotation captures an approximate static scene of the environment since the spinning speed is high. It is referred to as a frame. Points from one frame are compared with rays from other temporally nearby frames to find out whether they have moved.

Vallet et al. (2015) take n_{frame} frames before and after the target frame for comparison. Whereas, the occupied volumes of two successive frames can be overlapped for a moving object, which will lead to miss detection of the occupancy changes. Thus it is optimal to compare with frames where a pedestrian is supposed to be moved out of its occupied volume. Moreover, the same volume will most probably be occupied again by other objects, so the target frame should be compared within a short temporal window from the time that a typical pedestrian has moved out of its original volume until a certain time that the space is still assumed to be empty. The average speed sp and size sz of a pedestrian are assumed to be 1.5 m/s, 0.5 m respectively. Given the time t of current frame, the temporal window T is as follows:

$$\frac{sz}{sp} < |T - t| < \frac{sz}{sp} + \Delta t \tag{4.4}$$

in which Δt is the minimal time gap between two objects occupying the same space, e.g. 0.5 s.

Knowing the comparing frames, the occupancy information is combined directly at the location of the target frame point. The data fusion is conducted by the combination rule of the DST (Xiao et al., 2015) as follows:

$$\begin{pmatrix} e \\ o \\ u \end{pmatrix} = \begin{pmatrix} e_1 \\ o_1 \\ u_1 \end{pmatrix} \oplus \begin{pmatrix} e_2 \\ o_2 \\ u_2 \end{pmatrix} = \frac{1}{1-K} \begin{pmatrix} e_1 \cdot e_2 + e_1 \cdot u_2 + u_1 \cdot e_2 \\ o_1 \cdot o_2 + o_1 \cdot u_2 + u_1 \cdot o_2 \\ u_1 \cdot u_2 \end{pmatrix} \tag{4.5}$$

in which $K = o_1 \cdot e_2 + e_1 \cdot o_2$. Note that the combination rule is commutative and associative, so the order of combination is arbitrary. The final occupancy status at the location of target point will indicate whether it has moved, i.e. e means the space of the point is empty now so it has moved away; o indicates the point is static; u means no information about the space of the point which is mainly caused by occlusions.

Now the movement of individual point is detected by occupancy-based approach, meaning moving objects are detected at point level. Next sections will discuss the clustering of individual points and the estimation of trajectories for pedestrian tracking. It is worth mentioning that at this stage, points are labelled as moving or static, the object moving patterns can be visually interpreted from the moving points. A 2D density map/image can be generated by simply counting the number of points projected on each pixel.

4.3.2 Simultaneous detection and tracking of pedestrian

Most state-of-the-art tracking methods, e.g. tracking by detection, have to detect the targets in each frame in the first place. Then the targets are associated across the time so that the full trajectory is recovered. In our case, moving objects' points are detected, but are not yet clustered into structured objects. These points come from all types of moving objects. We aim to simultaneously detect the moving pedestrians and recover their moving trajectories. The points are translated into space-time (x, y, t) coordinate system. Then the detection and tracking are solved together as an energy minimization problem.

Notation

To ease reading, the notation is introduced first for the problem formulation.

- $\mathbb{N}^n = \{1, ..., n\}$ the first n positive integers.
- $\mathcal{X} = \{\mathbf{x}_i = (x, y, t)_i\}$, $\mathbf{x}_i \in \mathbb{R}^3$, $i \in \mathbb{N}^n$: the n points of detected moving objects. We work in 2D for simplicity, but extension to 3D is straightforward.
- $Part(\mathcal{X})$ the set of parts of $\mathcal{X}$:

$$Part(\mathcal{X}) = \{S | S \subset \mathcal{X}\}$$

- $\mathcal{P}(\mathcal{X})$ the set of partitions of $\mathcal{X}$:

$$\mathcal{P}(\mathcal{X}) = \{\{S_1, ..., S_m\} | S_i \neq \emptyset \text{ and } S_i \cap S_j = \emptyset \text{ and } \bigcup S_j = \mathcal{X}\}$$

- $|S|$ the number of elements in the set S (cardinal).
- $t_{min} = min_i t_i, t_{max} = max_i t_i$ the start and end times of the acquisition.
- $P : [t_{min}, t_{max}] \rightarrow \mathbb{R}^2$ a point trajectory giving a 2D position as a function of time.
- In this chapter, we will work with trajectories interpolated linearly piecewise through a set of control times $t_0 = t_{min} < t_1 < ... < t_{N_c-1} < t_{N_c} = t_{max}$. For practical reasons, we can choose the control times as:

$$t_c = t_{min} + N_c \Delta_t \quad \Delta_t = \frac{t_{max} - t_{min}}{N_c}$$

- A trajectory P will thus be defined by a start and end time t_{min} and t_{max} and a set of control points $\{P_c\}_{c=0...N_c}$:

$$P(t) = \frac{(t_{c+1} - t)P_c + (t - t_c)P_{c+1}}{\Delta_t} \tag{4.6}$$

- A simple 2D pedestrian model will be given by a disk of radius r centred at the trajectory point. Based on this model, we define the (adimensional) distance from a point to a pedestrian trajectory as:

$$D(\mathbf{x}_i, P) = max\left(\frac{dist\left((x_i, y_i), P(t_i)\right)}{r} - 1, 0\right)^2 \tag{4.7}$$

- Given a set $\{P^j\}$ of trajectories, we call $j^*(i) = min_i D(\mathbf{x}_i, P^j)$ the index of the closest trajectory to $\mathbf{x}_i$.

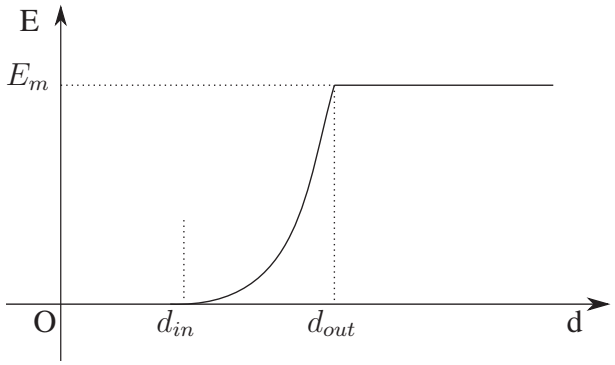


Figure 4.2 – Data attachment energy regarding the 2D point to trajectory distance d.

Global formalization

We formalize the simultaneous detection and tracking problem for pedestrians as finding an unknown number N_{traj} of trajectories P^j and an association of the points $\mathbf{x}$ to these trajectories defined by a trajectory index j_i for each points i, with the convention $j = 0$ for points not associated to a trajectory (outliers). These unknowns will be determined by minimizing the following energy:

$$E(P^j, j_i) = \sum_{i|j_i>0} \lambda_{out} D(\mathbf{x}_i, P^{j_i}) + \lambda_{track} \sum_j \frac{t_{max} - t_{min}}{\Delta_t} + \lambda_{traj} N_{traj} + \lambda_{rigid} \sum_j \sum_{c=1...N_c} ||\overrightarrow{P_{c-1}P_c} - \overrightarrow{P_cP_{c+1}}|| \quad (4.8)$$

This energy has following terms:

1. Adimensional data attachment term. r controls the radius size of pedestrian and the number of inlier points, and λ_{out} (adimensional) penalizes outliers (points not associated with a trajectory).
2. Density term. λ_{track} (in s^{-1}) gives the minimum rate of inliers per second required to track an object. It defines the sensitivity of the detector, i.e. higher value will favour clusters with high point density, and low value will allow to detect more clusters, especially those that are far from the laser center.
3. Generalization term. The data are to be explained by a minimum number of trajectories. A high λ_{traj} (adimensional) will favour the connection of trajectory pieces and discard trajectories associated with few points.
4. Smoothness term. It minimizes pedestrian acceleration. A high λ_{rigid} will smooth the trajectories whereas a low one will overfit to the data.

Now the outliers can have extremely large energy if the 2D distance to the trajectory $d = dist\left((x_i, y_i), P(t_i)\right)$ is large, thus we define a maximum energy E_m that a point can have, an outer radius r_{out} bigger than which the point is surely an outlier, and an inter radius r_{in} smaller than which the point is an inlier. The energy of a point $E_{\mathbf{x}_i}$ is illustrated as Figure 4.2.

$$E(\mathbf{x}_i) = \begin{cases} 0 & \text{if } d \leq r_{in}, \\ E_m(\frac{d-r_{in}}{r_{out}-r_{in}})^2 & \text{else if } r_{in} < d \leq r_{out}, \\ E_m & \text{else.} \end{cases} \quad (4.9)$$

The energy can be reformed as:

$$E(P^j) = \sum_i (E_{\mathbf{x}_i} - E_m) + \lambda_{track} \sum_j \frac{t_{max} - t_{min}}{\Delta_t} + \lambda_{traj} N_{traj} + \lambda_{rigid} \sum_j \sum_{c=1...N_c} ||\overrightarrow{P_{c-1}P_c} - \overrightarrow{P_cP_{c+1}}|| \quad (4.10)$$

This gives a simple interpretation to λ_{out} in terms of maximum distance of a point on a pedestrian to the pedestrian center.

4.3.3 Optimization

The problem is solved in two steps: (i) detect pedestrian tracklets by RANSAC (RANdom SAmple Consensus (Fischler et Bolles, 1981)) in each $[t_c, t_{c+1}]$ time window; (ii) connect detected tracklets by a clique graph.

RANSAC tracklet detection

The first step is to extract trajectory segments, i.e. tracklets, of length Δ_t, corresponding to the detection of all the points belonging to a pedestrian throughout this time window. For such a short time interval, e.g. 0.5 s, the trajectory can be approximated by a line segment following the interpolation of Equation 4.6. Noting δ_t the time for the laser scanner to acquire the $360°$ frame, RANSAC is used to find tracklets with starting time in $[t_c, t_c + \delta_t]$ and ending time in $[t_{c+1}, t_{c+1} + \delta_t]$. Two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are randomly selected as seed points of the segment in each of these intervals, then the impact ΔE on the energy of adding this segment P^{ij} is computed as follows:

$$\Delta E(P^{ij}) = \sum_i (E_{\mathbf{x}_i} - E_m) + \lambda_{track}\Delta_t \tag{4.11}$$

neglecting the last two terms that determine the tracklect association at next step. Two endpoints of the tracklet are extrapolated by the segment seed points at the starting and ending time of the interval. The tracklet with lowest ΔE is added to the solution set, and the corresponding inliers are removed from the point cloud. This process iterates until the lowest ΔE gets positive, meaning adding the best tracklet does not reduce the energy any more.

Clique graph tracklet association

Let us call P_c^{ij}, P_{c+1}^{kl} the tracklets extracted within temporal windows Δ_{t_c} and $\Delta_{t_{c+1}}$. We will now connect P_c^{ij} tracklet with P_{c+1}^{kl}. Because endpoints do not coincide exactly, they are connected by merging the endpoints $\mathbf{x}_j$ and $\mathbf{x}_k$ of the two tracklets at their middle position $\mathbf{x}_{jk} = (\mathbf{x}_j + \mathbf{x}_k)/2$. The corresponding energy variation is computed from Equation 4.10 for each possible connection.

Each one of the m tracklets from P_c^{ij} is possibly connected with all the n tracklets from P_{c+1}^{kl}. And each connection will have its energy change Δ_E. Only the connections with $\Delta_E < 0$ are kept. The connections can form a graph of $m \times n$ dimensions. Each plausible connection is a graph node. Since an endpoint can only move a short distance between two time windows, faraway endpoints can be pre-filtered out by verifying the distance. Thus majority of the hypothetical connections will be discarded, hence the graph is sparse. Two nodes can be linked by an edge if they are compatible, i.e. they do not contain the same endpoint, so that the track is not split or merged. Then the tracklet association amounts to select one set of edges such that the overall selected energy reduction is maximum. It means to find the maximum clique with the lowest energy. The optimal clique is solved using a clique-searching algorithm called Cliquer (Niskanen et Östergård, 2003; Vallet et al., 2014).

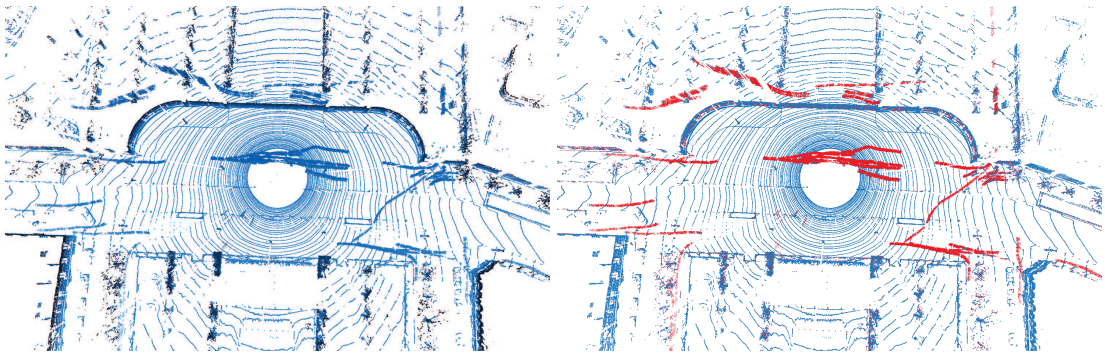
Since the first step split the data into parts with small temporal interval, the algorithm is inherent scalable. A long time-span dataset can be processed separately in multi-thread and on many machines. The first step will extract the tracklets from each subset. Since a tracklet is only composed of two points (start and end) in space-time cube, complete trajectories can be recovered for a long time-span dataset even if it does not fit in RAM.

4.4 Experiments and results

4.4.1 Moving object detection

The experimental data are acquired in Paris by the *Stereopolis* MMS (Paparoditis et al., 2012) using the HDL-64E Velodyne laser scanner, which is composed of 64 vertically distributed sensors. The scanner rotates around the vertical axis with a frequency around 10Hz. The vertical angular resolution is about 0.86°, and the horizontal one, in our case, is 0.23°.

To constantly monitor the place of interest, the MMS can either be stationed at certain observation points to scan the surroundings, or simply move around the place. We illustrate both static and moving acquisitions. Figure 4.3 shows a static scenario, in which points on moving pedestrians (red) are successfully detected. False detections can be observed on penetrable objects, e.g. fences, trees, which is the drawback of using occupancy-based method because penetrable objects are self-conflicting when modelling their occupancies (Xiao et al., 2015). These falsely detected points are of little amount hence can be considered as noise in the following process, and can be easily filtered out since they do not move over time. Figure 4.4 shows the result of moving object detection when the mapping system is moving.

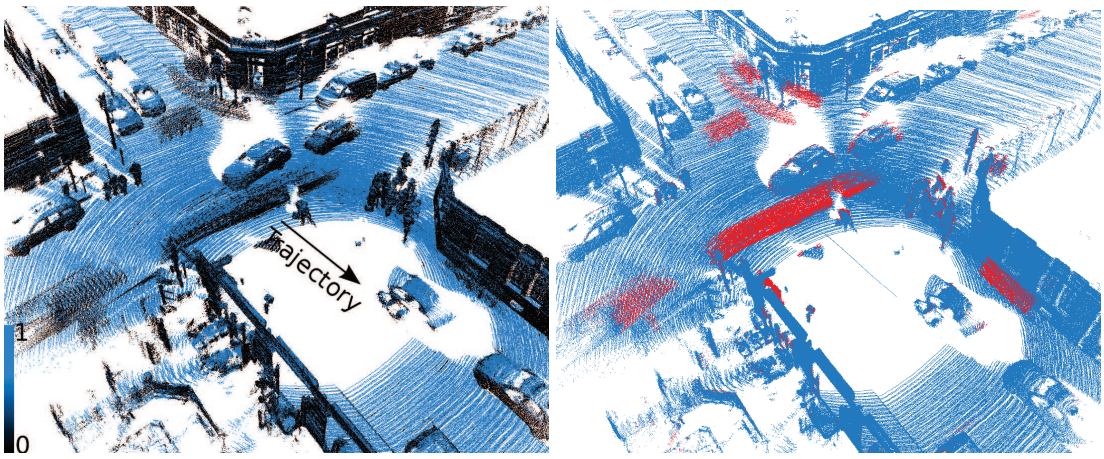


(a) (b)

Figure 4.3 – Moving object detection using static mapping system, (a) original data, (b) detected moving points in red.

The detection results are statistically evaluated against manually labelled ground truth (Table 4.1). Since the method detects points on moving objects, the results are evaluated at point level. The recall (R), precision (P) and F_1 score are assessed. It is obvious that in both scenarios, the recalls are relatively high. Whereas the precision of the static scene is significantly lower than the moving case. The reason is the challenging dataset which contains many penetrable objects. Theoretically, under similar circumstances, the method should perform no worse in static mode than in moving situation. Nevertheless, false detections and under detections are expected, so the tracking method should be robust to noise and certain degree of data missing.

There are simpler moving object detection methods when the MMS is static. The laser beam is not supposed to pass through the static environment scene. Thus one can take the furthest points of each of the i laser sensors over a long period as the static environment. Each circle is partitioned into j parts depending on the proper angular resolution. Then the static environment is represented by a distance map $D_{max} \in \mathbb{R}^{i \times j}$. Points between the laser center and the furthest reachable locations ($D^{i \times j} < D^{i \times j}_{max}$) are on moving objects. According to the



(a) (b)

Figure 4.4 – Moving object detection using MMS, (a) input Velodyne data acquired within 2 seconds (coloured by normal N_z), (b) output result: points on moving objects are detected (red).

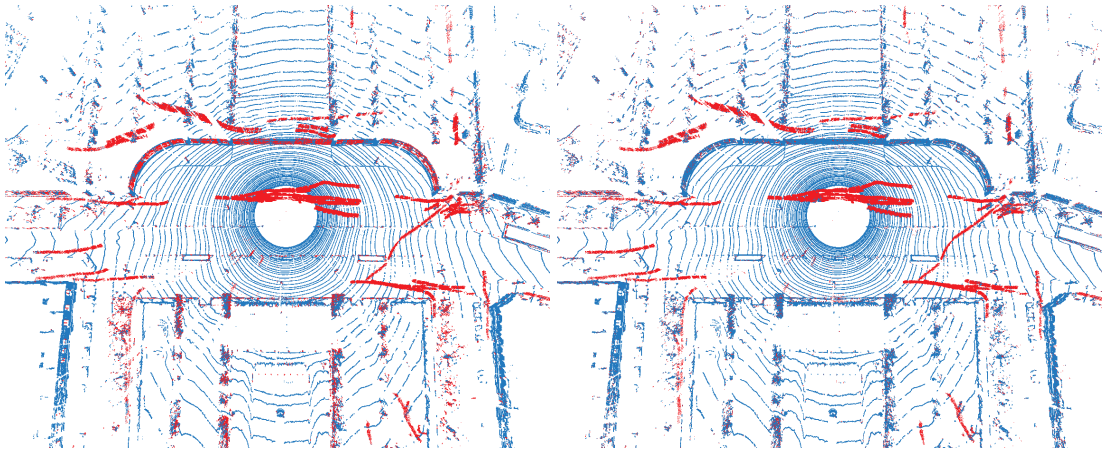
Table 4.1 – Accuracy assessment of moving object detection at point level. The recalls (R) are relatively high in both static and moving modes. Whereas the precision (P) is significantly lower in the static mode because the scene is more challenging.

Method	Occupancy-based			Max-Distance			Nearest-point		
	R	P	F_1	R	P	F_1	R	P	F_1
Static	83.0	57.3	67.8	96.2	20.9	34.3	60.1	70.1	64.7
Moving	80.5	84.7	82.5	–	–	–	–	–	–

results in Table 4.1, this method, i.e. *Max-Distance*, is able to detected almost all the moving points, whereas it has very low precision. Because it is very sensitive to penetrable objects, which are largely presented in the data.

Another method is to count the number of nearest points. Static data acquisition constantly scans the same locations, hence points will be accumulated on static objects over a period of time. Whereas moving objects will have many instances along the moving trajectories. These object instances are normally overlapped because of the high scanning frequency. So the number of nearest point is counted within a certain temporal window, which is the same as the occupancy-based method (Equation 4.4). Nearest points are searched for an object instance within this window where the object should be moving out of its original occupied volume, meaning no overlapped instances are taken. This method, *Nearest-point* (Table 4.1), does not require the scanning geometry, nor the furthest backscattered points. This means that the data acquisition can be artificially limited to a reasonable range where the point density is high enough thus the data size is reduced. Another advantage is that it is robust to penetrable objects given a proper distance threshold, as shown in Figure 4.5.

These points on moving objects can be accumulated on an image which will illustrate the moving pattern and density of objects. It can be used for visual interpretation of the moving object traffic flow (Figure 4.6). The next step is to recover the individual trajectory of each



(a) (b)

Figure 4.5 – Moving object detection using (a) Max-Distance, and (b) Nearest-point methods. The former is simple and fast, whereas the later is robust to penetrable objects, as seen in (b) there are much less false alarms (red) on fences and trees.

pedestrian.

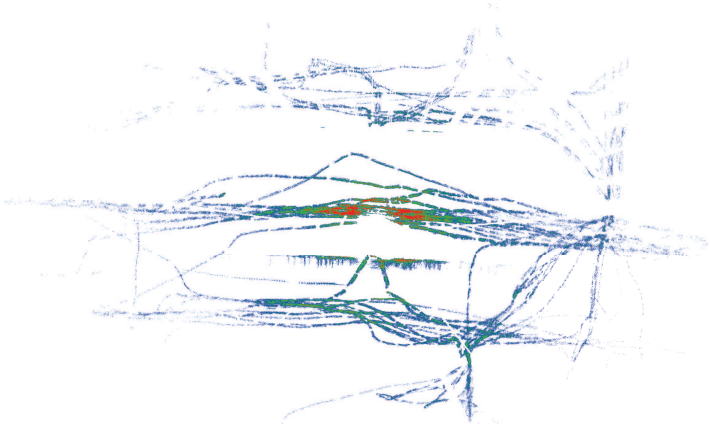
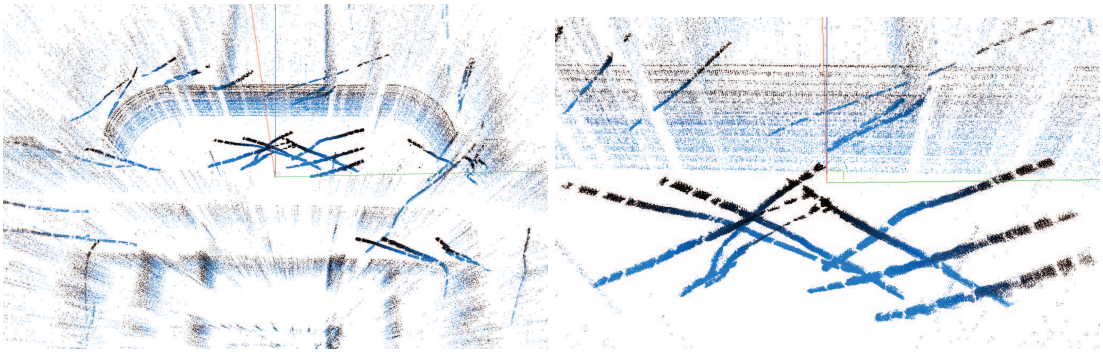


Figure 4.6 – Moving object point accumulation image. Points are projected to an image showing the number of points in each pixel. Coloured from blue to green, to red as the pixel value grows.

4.4.2 Pedestrian tracking

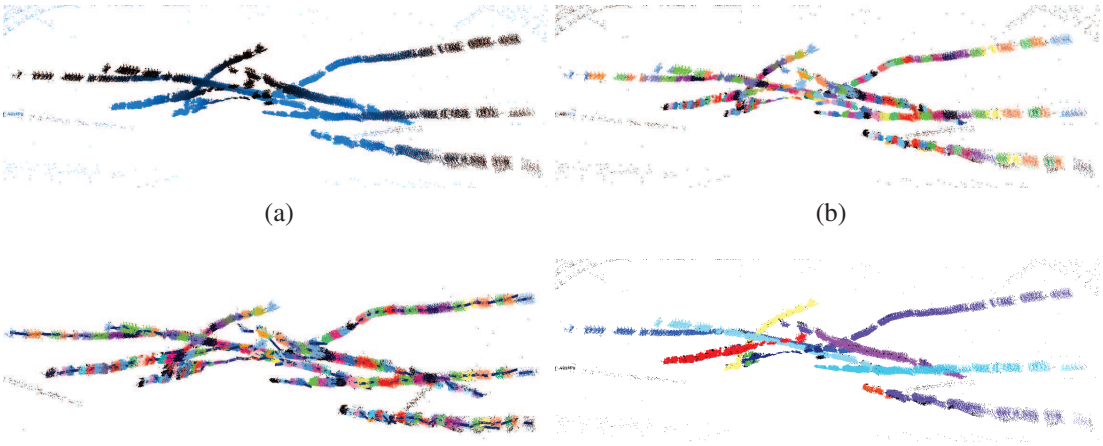
Detected moving points are firstly transfer into space-time cube (x, y, t) which are shown in Figure 4.7. The data is then partitioned into small temporal windows, such that in each time window, the pedestrian trajectories are assumed to be a straight line. Then the tracklets are estimated by associating the centralized points to the assumed tracklets hypotheses. In the end, the tracklets from different temporal windows are connected so that the overall trajectories are recovered (Figure 4.8).

One of the advantages of the SDAT method is that all the points on moving objects are retained, even those are partially scanned due to occlusion or scanning FOV. Figure 4.9 shows



(a) (b)

Figure 4.7 – Detected moving objects shown in space-time cube (x, y, t) as green, blue, red axis respectively. Dense point clusters are moving trajectories, and sparse points are false detections. Points are coloured from blue to black over time. (b) is zoom-in view of the central part of (a).



(a) (b)

(c) (d)

Figure 4.8 – Moving object tracking. (a) original detected moving points; (b) points are partitioned into small temporal windows coloured randomly; (c) detected tracklets in partitioned temporal windows (tracklets are in dark blue); (d) connected final trajectories.

an example where significant less points are acquired on the moving objects that are close to the laser scanner (highlighted by a black rectangle). The trajectories are still recovered because these small amount of points convey enough evidence for the tracklet searching. However, as for tracking-by-detection method, these points will hardly be detected as pedestrians because they are only of a a small portion of a whole people.

Figure 4.10 illustrates the pedestrian flow of the public place during approximate 3 minutes. General trajectory patterns can be easily observed. Detailed and accurate information, e.g. pedestrian location, moving direction, can be extracted from the results and used as input for accurate pedestrian flow simulations.

To evaluate our SDAT method, a benchmark ETH Zurich *Polyterrasse* dataset used by Spinello et al. (2011) and Kaestner et al. (2012) is adopted. Both work follow the detection-and-then-tracking convention. The difference is that the former specifically detects pedestrian using a bottom-up top-down (BUTD) detector, whereas the latter detects all types of moving

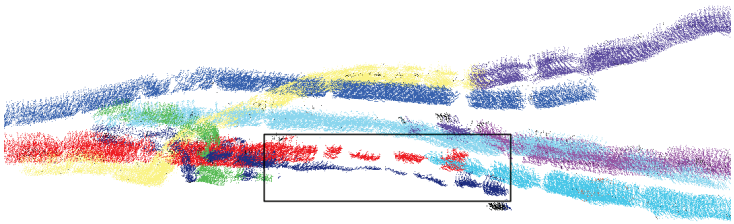


Figure 4.9 – Example of trajectory recovery by points that are partially scanned on objects (highlighted in the black rectangle).

Figure 4.10 – Pedestrian flow in a public place. Pedestrian trajectories are in random colors, and the background is in black.

objects in general. The tracking results are quantitatively evaluated using the CLEAR MOT metrics (Bernardin et Stiefelhagen, 2008). Three basic types of errors/values are determined for the estimated tracks against the ground truth:

• FN: The false negative ratio, i.e. the percentage of missing tracks that are supposed to exist regarding the ground truth.

• FP: The false positive ratio, i.e. the percentage of false alarms w.r.t the ground truth.

• MM: The number of mismatches in terms of track identity switches.

Two more indicators are also derived:

• MOTP: Multiple object tracking precision, i.e. the average Euclidean distance between the estimated track instances and the ground truth.

• MOTA: Multiple object tracking accuracy, i.e. the percentage of the number of correct track instances w.r.t the ground truth.

Table 4.2 – Tracking methods comparison using the CLEAR MOT metrics. MOTA, FN, FP are in %. The best value of each indicator is in bold.

Detector	MOTP	MOTA	FN	FP	MM
BU	$< 0.16m$	23.1	18.7	57.1	11
BUTD	$< 0.16m$	**89.1**	**2.6**	**7.6**	20
General	$< 0.14m$	77.7	8.5	10.1	n/a
SDAT	$<$ **0.13**m	62.7	8.3	28.4	**5**

The MOTA can be computed by the three basic values:

$$MOTA = 1 - \frac{FN + FP + MM}{\sum track_{groundtruth}} \tag{4.12}$$

The evaluation results are listed in Table 4.2, together with other results from comparative methods. The comparison reveals that the SDAT method has the most accurate track positions and least mismatches. The FN ratio is acceptable, however, the FP ratio is significantly lower than the better ones. One reason is that a duplicated track is sometimes observed if the object is taking/carrying large accessories, e.g. a stroller. But the most dominant factor is the non-FP tracks that are not annotated in the ground truth. These tracks should be accounted as true positives. The same issue is raised by Kaestner et al. (2012). Figure 4.11 illustrates the detected (blueish) tracks and the (reddish) ground truth. It is obvious that the majority of the FP tracks naturally extend from the ground truth tracks. It is clear that these FP tracks are supported by moving points as shown in Figure 4.12.

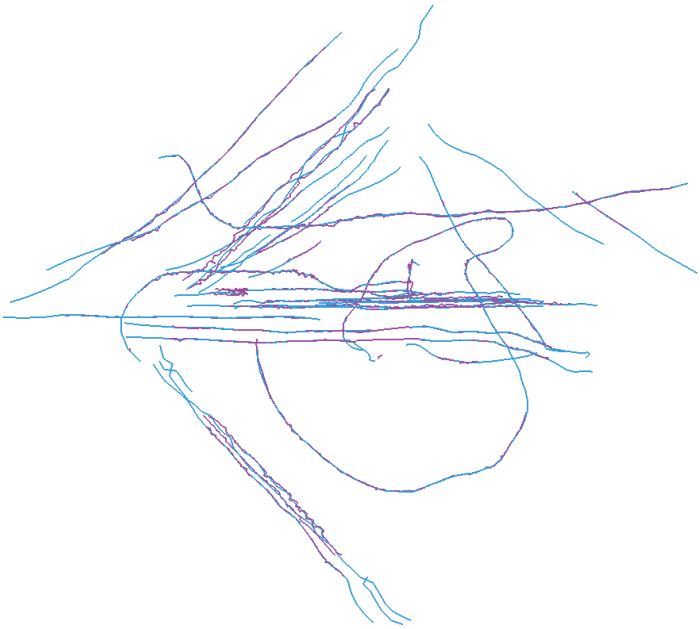


Figure 4.11 – Comparison between the detected tracks (cyan) and the ground truth (magenta). Most of the detected tracks are naturally longer than the ground truth tracks.

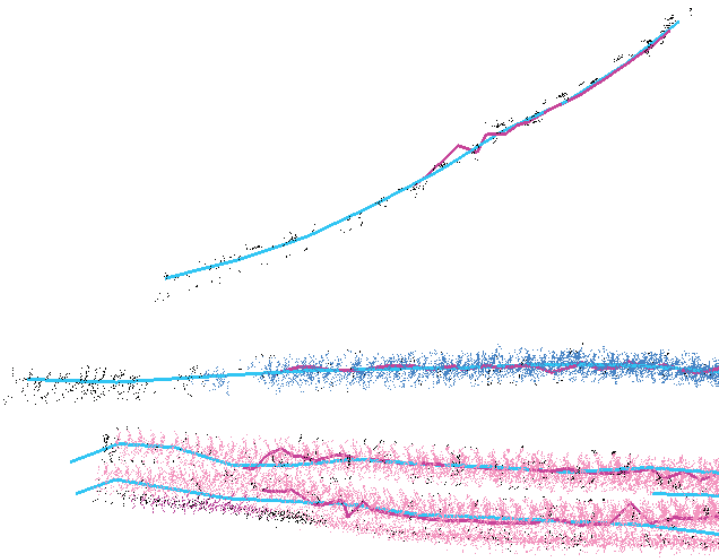


Figure 4.12 – Detected moving points and tracks (cyan) compared with ground truth (magenta).

4.5 Discussions

Points on moving objects are detected firstly based on the space occupancy. The *empty* status of each point ranges from 0 to 1, and naturally the threshold is set as 0.5. Some real moving points (true positive) are detected with $empty < 0.5$ and are treated as static. Smaller threshold will increase the recall, meanwhile allow more false alarms. The maximum-distance moving object detection method is simple and much faster. But it is specific to the static data acquisition mode and is more sensitive to penetrable objects. Whereas occupancy-based method can be used for both static and mobile mapping systems. Both methods need point recording behind the moving objects. If there is no light/pulse backscattered to the sensor, the scanning range from the sensor center to the furthest reachable location can no be determined. Then points on moving objects in middle of the scanning range will be treated as the furthest reachable points hence the static background. The third nearest-point method, however, only takes point locations into account hence does not required further information, e.g. scanning geometry, furthest range. Points far from the laser scanner are normally very sparse and no reasonable features can be extracted, so they can be discarded to reduce the data size if necessary. The static environment is automatically extracted at this stage. Even though there still might be mobile objects, the sub-cleaned data can be used as basis for database updating, road mapping, surface reconstruction, etc. Thus the moving object detection algorithm can serve as the pre-processing step for applications such as robotic navigation and mobile mapping.

Our SDAT algorithm takes all the detected moving points as input regardless false detections. The strengths are: (i) no need of segmentation for the detected moving points; (ii) no need of moving point detection refinement or explicit classification (pedestrian or not); (iii) partially scanned objects are also retained, to be robust to occlusions and under detections. However, the downside is that false alarms can be raised if a non-pedestrian object's spatial distribution is constantly similar to a pedestrian's. The only constraint is the number of points which lie inside of a pedestrian-sized 2D circle. Apparently, more comprehensive and discriminative features should be incorporated into the SDAT algorithm in the future work.

The tracking method inherently smooths the detected trajectories since tracklets within the temporal windows are treated as linear, and they are connected by averaging their endpoints. Due to the randomness of seed tracklet point selection, the results can slightly vary at each run. Still, they can be unified by increasing the number of random trials at the price of more time consumption. When an object is far from the laser scanner, its point density is small due to the radial scanning nature. Then there will be not enough points for tracklet detection. So the Velodyne laser scanner has a certain effective range, better within 20 m (Spinello et al., 2011; Kaestner et al., 2012).

4.6 Conclusion

A pedestrian-oriented simultaneous detection and tracking (SDAT) algorithm is proposed and is successfully implemented using laser scanning data. Moving object points are extracted firstly by assessing the consistency of space occupancy supported by the laser rays. The accumulated point density image/map can be used for qualitative visual interpretation of the pedestrian flow. Quantitative estimation is achieved by reconstructing the 3D trajectories of pedestrians. The SDAT method takes all the moving points and estimates the trajectories that best explain the points. The tracking, segmentation and classification are solved simultaneously. The method inherently handles occlusions and under detections. Results suggest that Velodyne laser scanning data can be used for efficient and accurate pedestrian flow estimation.

Future work will focus on improving the processing speed of the algorithm, and integration of multiple scans for large areas.

Chapter 5

Street-side car detection, classification and change detection

Contents

Statistics on street-side car parks, e.g. occupancy rates, parked car types, parking durations, are of great importance for urban planning and policy making. Related studies, e.g. vehicle detection and classification, mostly focus on static images or video. Whereas mobile laser scanning (MLS) systems are increasingly utilized for urban street environment perception due to their direct 3D information acquisition, high accuracy and movability. In this chapter, we design a complete system for car park monitoring, including car recognition, localization, classification and change detection, from laser scanning point clouds. The experimental data are acquired by a MLS system using high frequency laser scanner which scans the profiles of streets. The point clouds are firstly classified as ground, building façade, and street objects which are then segmented using state-of-the-art methods. Each segment is treated as an object hypothesis, and its geometric features are extracted. Moreover, a deformable vehicle model is fitted to each object. By fitting a explicit model to the vehicle points, detailed information, such as precise position and orientation, can be obtained. The model parameters are also treated as vehicle features. Together with the geometric features, they are applied to a supervised learning procedure for vehicle or non-vehicle recognition. The classes of detected vehicles are also investigated. Whether vehicles have changed across two datasets acquired at different times are detected to estimate the durations. Here, vehicles are trained pair-wisely. Two same or different vehicles are paired up as training samples. As a result, the vehicle recognition, classification and change detection accuracies are 96.5%, 84.9% and 98.7%, respectively. Vehicle modelling facilitate recognition rate, also precisely locate vehicles compared to bounding boxes.

5.1 Introduction

In many populated cities, on-street parking places have strict parking regulations which need to be controlled and monitored regularly. Types of cars using certain parking spaces are important for street parking design and management. For instance, the time span and type of cars parked on commercial streets are interesting to street managers and street-side shops. Ordinary cars are not supposed to park on reserved spots, e.g. disabled or delivery. Cars that do not belong to the residential areas are not encouraged to park there for a long time. Cars are supposed to park inside the parking spots (within the parking line) and not on the pavements. Utility cars and passenger cars usually have different parking rules, e.g. fees and durations. Many cities implement the alternate side parking rule: only one side of the street can be parked on, on a given day. In France, the law only allows cars to park at the same spot for maximum seven consecutive days. These regulations are normally controlled by investigators. And the process is tedious, time consuming and unproductive.

To facilitate the automation of on-street parking monitoring, we aim to accurately detect cars, including car recognition and precise localization. Also, car categories are important information for parking regulations. Furthermore, due to the demand of estimating the parking durations, we need to detect whether two cars are parked at identical locations at different times in a day.

Current technologies, e.g. video surveillance using static cameras, are able to detect cars in parking lots hence generating the occupancy rate. However, it is obviously not practical to install cameras on all the streets. Mobile mapping systems (MMSs) can be a promising alternative. They are moving vehicles mounted with cameras and/or laser scanners that are geo-referenced by a positioning unit so that the acquired data are in a geographical coordinate system and can be used directly for localizations (Paparoditis et al., 2012). 3D vision using images acquired from MMSs have been used for large scale applications (Zia et al., 2015; Taneja et al., 2013). However, since the 3D information are generated indirectly, precise geometry extraction, localization and 3D modelling still remain challenging tasks. Moreover, optical cameras are sensitive to light conditions, such that they might not work during night time. On the contrary, laser scanners measure the surroundings actively and directly in 3D with a high level of precision (a few centimetres). They open the potential to precisely locate and model cars which can be used for more detailed investigations, such as whether a car has parked properly in one parking spot or whether two same coloured cars are really the same.

There are studies focusing on one or two aspects of this practical problem, such as car detection. But precise localization, fine-grained classification and change detection of cars using laser scanning (also referred to as lidar) point clouds have not been investigated yet. Not to mention a complete system of all the tasks for car park monitoring.

Car oriented object detection using laser scanning data can only be found in few studies (Toth et al., 2003; Yao et al., 2008; Börcs et Benedek, 2012). Most studies deal with a generic classification problem, i.e classifying all objects in the data rather than a specific one (Serna et Marcotegui, 2014; Weinmann et al., 2015). One common procedure is to remove points belonging to the ground and building façades, and then cluster the remaining points into segments. Then features are extracted from the segments. Some categorize the features as at point or segment level, others extract features in different dimensions, mostly in both 2D and 3D. In addition to geometric features, other features, e.g. RGB and intensity are also utilized if available. Then the dataset is classified using supervised classification methods. Detected cars

can be roughly localized by a bounding-box using their 3D coordinates. However, lidar data are typically incomplete because cars are only scanned from one side. Hence a bounding-box can not recover the whole shape of a car and is mostly not accurate enough for fine detailed applications.

Car shapes are universalized due to the pursuing of high-profile outlook and aerodynamic designs. Some cars lie between two categories in terms of size and shape, e.g. crossovers, making the classification a difficult task. To identify whether cars have changed in between the different epochs, one straight forward method is to find out the corresponding cars in two datasets, then compare their geometries. Normally a threshold is set to identify whether the difference is significant enough to be a change. The challenge is that car comparison can be affected by many factors, e.g. registration error, scanning perspective, point density, anisotropic sampling, occlusion. Then the threshold of the difference can be hard to set.

In this chapter, a complete system for detailed on-street parked car information extraction is proposed, including car recognition, localization, classification and change detection, for the purpose of car park monitoring. A deformable car model is proposed to fit car hypotheses for precise localization, and its parameters are used as features for supervised learning. Furthermore, it can be used for visualization purposes.

5.2 Related work

5.2.1 Vehicle detection

Image-based vehicle detection has been intensively studied for the purposes of scene understanding (Bileschi et al., 2004; Arróspide et al., 2012; Zia et al., 2015), traffic monitoring (Gupte et al., 2002; Huang et Liao, 2004), intelligent vehicles (Betke et al., 2000; Jazayeri et al., 2011; Caraffi et al., 2012), etc.

Bileschi et al. (2004) detect vehicles using the part-based method (Agarwal et al., 2004). Keypoints are extracted from the test images, then vehicles are detected by comparing the keypoints against a vocabulary of vehicle-specific keypoints learned from the training set. Instead of using keypoints, Arróspide et al. (2012) utilize gradient-based descriptor, which is the simplified histogram of gradients (HOG) (Felzenszwalb et al., 2010), for efficient vehicle detection. Tuermer et al. (2013) detect vehicles from airborne images using also HoG features and disparity maps. Bensrhair et al. (2002) detection vehicle in 3D based on the feature extracted from stereo vision. Moreover, in recent years, vehicles have been increasingly studied using both airborne laser scanning (ALS) and MLS point clouds in 3D.

Toth et al. (2003) extract vehicles utilizing ALS data by thresholding height histograms, and then classify them into several main categories using rule-based and machine learning classifiers. Yao et al. (2011) also detect vehicles with ALS data using support vector machines (SVMs) after segmentation. Velocities are also estimated for the purpose of traffic monitoring and analysis. Börcs et Benedek (2012) add intensity values and number of echoes as features for vehicle detection using ALS data.

Himmelsbach et al. (2008) use a MLS system for object detection, then classify the detected objects as vehicle and non-vehicle using SVMs. Keat et al. (2005) extract vehicles from MLS data using Bayesian programming and finally map a whole parking lot. Golovinskiy et al. (2009) analyse point clouds generated from combined ALS and MLS. They first segment com-

pact objects, then describe them with shape and context features and classify them into different urban object classes, including vehicles and traffic signs. Velizhev et al. (2012) improve the vehicle and pole object detection results using the same data by incorporating the implicit shape model (ISM) framework. To recognize an object hypothesis, random keypoint descriptors are matched with a pre-generated dictionary of *geometric words* and then the potential object center is assigned by weighted votes. Serna et Marcotegui (2014) develop a full pipeline of segmentation and classification of urban objects using MLS data. Segmented objects are classified using SVMs with different set of features. The method is applied to three datasets, one of which is focused on vehicles. High precision and recall are reported.

Supervised learning for vehicle detection or classification are popular for laser scanning data. Different types of features at point or object/segment level are explored: geometric features, contextual features and intensity/color features. In many computer vision studies, detailed models are used to facilitate object detection and classification in images or video. However, object modelling towards scene understanding, including object detection and classification, using laser scanning data still remains a challenge.

5.2.2 Vehicle modelling

Detailed 3D object representations facilitate scene understanding (Zia et al., 2015). Geometrically accurate models help to improve the detection and localization of modelled objects, and can be used for visualization.

Kaempchen et al. (2002) estimate parked vehicle's pose facilitated by a generic 3D vehicle surface model for autonomous parking. The vehicle surface model is composed of three or four 3D planes, hence is rather simple and incomplete. Hinz et al. (2003) utilize an adaptive 3D model for automatic vehicle detection from high resolution aerial images. The explicit model incorporates both geometric and radiometric features. It is adaptive to different feature saliency but not to various vehicle shapes/types. The disadvantage is that a large number of models are needed. Therefore, a shape adaptive model, i.e. active shape model (ASM), is adopted in this paper.

The ASM is a statistical model whose shape is deformable to fit with new object instances proposed by Cootes et al. (1995). The model is learned from a training set and can only deform in ways consistent with the training set. The method was successfully implemented to locate and model different types of objects from images in 2D. Ferryman et al. (1995) build a 3D deformable vehicle model using the ASM. The model parameters are then used to classify the vehicles into different subclasses. Sullivan et al. (1995) use the same method to improve the performance of pose estimation, and to locate and classify vehicles in video (Sullivan et al., 1997). Zhang et al. (2012) manually set the model parameters and their value ranges instead of learning from the training set for vehicle localization and recognition. A local gradient-based method is proposed to evaluate the model-to-image fitness. Leotta et Mundy (2009) develop a multi-resolution meshed deformable vehicle model to facilitate edge detection and model-to-image fitting. The model is also used for simultaneous shape estimation and tracking in video (Leotta et Mundy, 2010).

Zia et al. (2013) used detailed representations for vehicle and bicycle recognition and modelling. Different from rather simple 2D bounding-boxes, representative 3D deformable ASMs are designed for both vehicles and bicycles. They are trained based on 3D CAD models and the deformation is learned from these samples. Then objects are detected by matching models with

images using discriminative parts.

Many studies use the ASMs for vehicle recognition, localization and tracking. One of the major differences is the model-to-image fitting method. However, as for the lidar point cloud, an innovative model-to-lidar data fitting method is needed.

5.3 Methodology

The full system is composed of four major parts: vehicle recognition, localization, classification and change detection. Vehicle recognition and localization together are considered as vehicle detection. They are interconnected and can be conducted simultaneously. They are followed by vehicle classification and change detection which can be implemented in parallel. In detail, the acquired 3D point clouds are first generally classified as ground, building and street object points. Then the object points are segmented and each segment is taken as an object hypothesis. The following processes of the system are depicted as Figure 5.1. Geometric features, i.e. features that describe shape and size, are extracted at object level. And a deformable vehicle model is fitted to each object for precise localization. The parametric vehicle model provides accurate vehicle position and orientation, also the model parameters are treated as object features. A supervised learning procedure is then applied for vehicle recognition. Here, it is a binary classification between vehicle and non-vehicle. Next, the detected vehicles are classified into different categories.The vehicle change detection problem is also formulated as a binary classification task, same or different vehicles, without explicitly comparing their geometries.

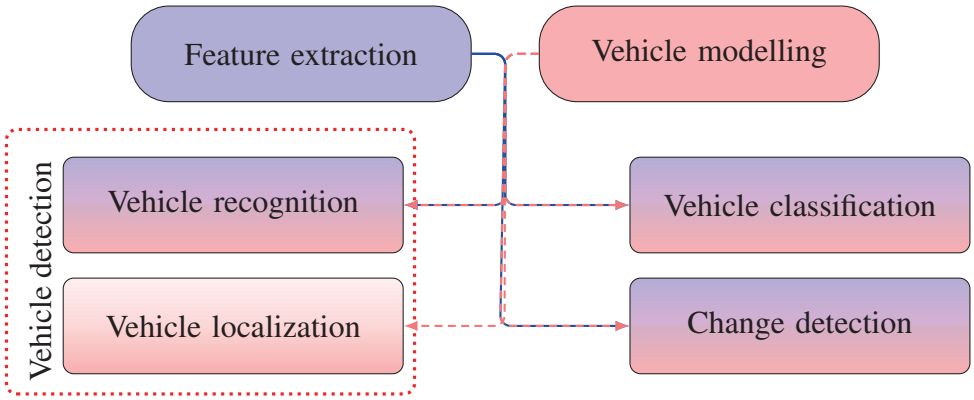


Figure 5.1 – Flowchart of the full system. Feature extraction and vehicle modelling serve as basis for the four main steps: vehicle recognition, localization, classification and change detection.

5.3.1 Segmentation and feature extraction

Segmentation

Data are segmented using the method proposed by Serna et Marcotegui (2014). Ground points are filtered out first using the λ-flat zones labelling algorithm. The point cloud is projected on to a 2.5D elevation image. The maximal elevation value is assigned to each pixel since in general it contains several points. This maximizes the height difference between ground pixels and neighbouring non-ground pixels. Two neighbouring pixels x, y belong to the same λ-flat zone if their height difference $|h_x - h_y| \leq \lambda$. The largest λ-flat zone is taken as the ground. λ is set as

20 cm to cover both roads and pavements. Building façade points are automatically segmented by checking the highest vertical and most elongated structures in the elevation images (Serna et Marcotegui, 2013). Then the rest points are segmented using the top-hat by filling holes (THFH) (Hernández et Marcotegui, 2009a), which amounts to a general connect component labelling. Nearby objects are still labelled as one component. Local maxima are detected for each component as object markers. They are refined by a morphological h-Maxima filter to eliminate the ones with a low local contrast, i.e. relative height is less than the threshold h. A constrained watershed (Hernández et Marcotegui, 2009b) is implemented to further segment the connect components.

Each segment is treated as an object hypothesis from which features are extracted. Since the goal is to detect vehicles, obvious non-vehicle objects, i.e. oversized and undersized objects, e.g. traffic signs, large trees, can be pre-filtered out in practice.

Feature extraction

Most studies use low level features based on individual point or a small group of points, e.g. dimensionality (Demantké et al., 2011), verticality, local density, point feature histogram, because there is no *a priori* segment information. In our case, the input is a segmented object hypothesis already. So we extract features at object level in 3D. The object is firstly transferred into its local coordinate systems using principle component analysis (PCA), and then fitted with a bounding box (Bbox). The features are extracted as follows:
• object dimension: length (l), width (w), height (h)
• volumetric feature: area (a), volume (v), extracted from 3D bounding box
• relative position: maximum height (H_{max}), mean height (H_{mean}), relative to the ground
• vertical point distribution histogram (VPDH): the input object is vertically divided into even parts from a certain height to the ground. Point proportion regarding the overall number of points at each vertical part varies between different urban objects. The feature is represented by a histogram containing the point proportion of each vertical part. We assume the maximum height of studied cars is lower than 2 m, so we divide it into 10 parts and each part is 20 cm high. The 10 VPDH features are named as $v_1 \dots v_{10}$

Now, we have 17 geometric features. Further features may help to improve the supervised classification results. Some classifiers are able to give a proper weight or importance to each feature, or to select the best feature subsets. Selecting optimal subsets of features has been proven to improve the final classification results and reduce the training time (Weinmann et al., 2015). Therefore, there is no need to extract a large number of features that are potentially redundant.

5.3.2 Car modelling

In MLS data, objects are inherently incomplete since most of the time they are scanned only from one side, hence we propose a model-based approach to reconstruct complete car models for precise localization. A deformable car model is reconstructed using the active shape model (ASM) to maintain the geometrical structure since cars share a common basic structure (Zia et al., 2013). A geometrically correct shape is created as the basis and only the variance of cars are focused when modelling. Since the ASM is a complete car model, the occluded part is guaranteed to be reconstructed. Thus this model is robust to partial data incompleteness and

various point densities.

Deformable car model

A global ASM is learned first from a training dataset. The points in the ASM are a list of n landmarks representing the basic shape of a car. These landmarks are manually picked up from the training cars. With more landmarks, the model will represent the original shape in more detail. However, cars are normally only scanned from the side for street-side parked ones. Hence the number of landmarks is limited by the completeness of the point clouds of cars. Since cars are symmetrical, we only need to select landmarks from one side. Then the other side is automatically obtained by symmetrical projection, given the width. In our case, 18 landmarks are selected on one side, two of which are on the center of wheels. One more landmark is selected on the other side to extract the width, and other one is selected on the edge of a wheel to obtain the wheel radius (Figure 5.2).

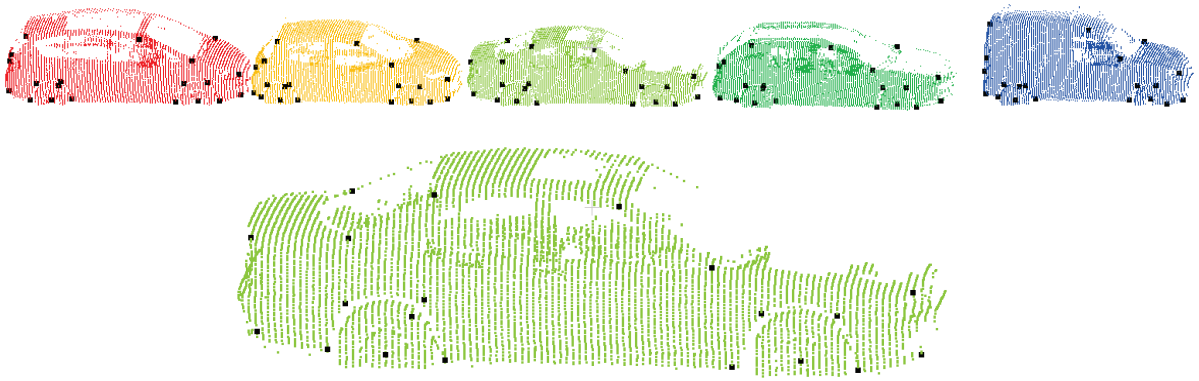


Figure 5.2 – Landmark selection from training samples (one enlarged sample below; landmarks in black). 20 points are selected in total: 16 on one side of the car, one on the wheel edge, two on wheel centres, and one on the other side.

Then the landmarks of one side of a car from all the training samples are aligned. The line passing through the two wheel centres are treated as the x axis, and the midpoint of them are the origin. There are a total of 20 landmarks for one sample, which are defined as a vector $\mathbf{X} \in \mathbb{R}^{3i}$

$$\mathbf{X} = (x_1, y_1, z_1, ..., x_i, y_i, z_i) \tag{1}$$

in which i = 20. The sample alignment and the mean of all samples $\bar{\mathbf{X}}$ are depicted in (x, z) plane as Figure 5.3.

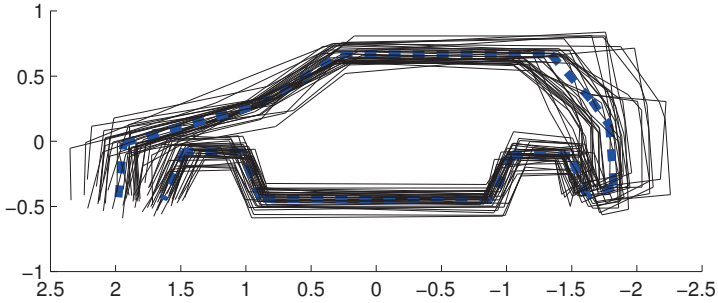


Figure 5.3 – Active shape model alignment, mean of samples in blue dashed line

Now, the samples are transformed into a local coordinate system and the 3D point distribution model can be obtained by projecting the points from one side to the other using the retrieved car width. This model needs to be fitted with lidar points. Unlike model-to-image fitting which is in 2D (Leotta et Mundy, 2009; Zia et al., 2013), model-to-points fitting can be conducted directly in 3D. The idea is to fit the model surface with lidar points since most of the acquired points are on car surfaces. The deformable car model is defined as a wireframe model based on the ASM by triangulating the landmark points. And the model is deformed by moving the point positions, meanwhile the connectivities of points are maintained. The triangulation is conducted manually (Figure 5.4). Then the model is fitted by minimizing the distance between lidar points and the model triangles.

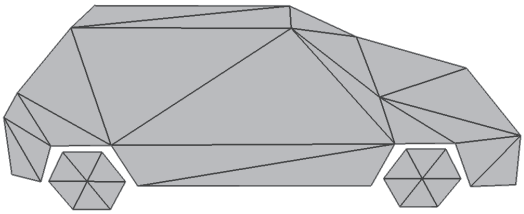


Figure 5.4 – Car model composed of triangles

Each element of the model vector $\mathbf{X}$ varies across the training samples, and the variance is assumed to be a Gaussian distribution in the vector space. The model vector space has relatively high dimensionality, whereas the car shape lives in a lower dimensional subspace since many elements have similar distributions. Dimensionality reduction can be carried out without greatly constraining the model, and has a denoising effect. The principle component analysis (PCA) is implemented to reduce the dimension of the model vector. The j eigenvectors $\mathbf{P} \in \mathbb{R}^{3i \times j}$ describe the principle modes of variation across the training set. The ASM can be defined by the linear combination of the mean shape with the eigenvalues in each component direction:

$$\mathbf{X}^{'} = \bar{\mathbf{X}} + \mathbf{w}\Sigma\mathbf{P} \tag{2}$$

where $\bar{\mathbf{X}}$ is the mean shape of all the training samples, Σ is the standard deviation vector of the principle component $\mathbf{P}$, $\mathbf{w}$ is a scaling vector for each component. So the deformable car model is a group of 3D triangles composing the basic shape of a car which is deformable by tuning the scaling vectors. To ensure that a newly generated model is represented by the training set, the scaling variable normally vary within three times standard deviation, which covers 99.7% of the values in the normal distribution. So elements of $\mathbf{w}$ are constrained between [-3, 3]. An example of the deformed models is depicted in Figure 5.5.

Model fitting

The input of the car modelling algorithm is an object hypothesis, meaning a segmented point cluster that is supposed to be an object. For each object hypothesis, the mean shape model $\bar{\mathbf{X}}$ is initialized at the geometric center. The orientation is approximated by PCA. The top j principle components are taken as the deformation directions, and the corresponding scaling vector $\mathbf{w}_k$, $k \in (1, j)$ are the deformation parameters. To simplify the model fitting, cars are assumed to stay on a horizontal ground plane, hence there is only one rotation parameter $\mathbf{R}_1$. There are also three translation parameters $\mathbf{T}_3$. The model is initialized at the geometric center hence may not lie on the ground, and the deformation may also change the wheel height, so the translation in z

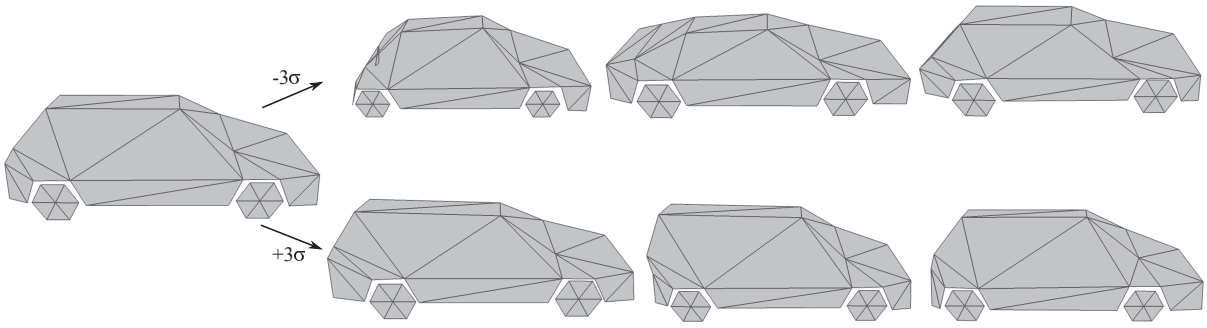


Figure 5.5 – Mean car model and the top three principle components (left to right) of the active shape model, each component is deformed by $\pm 3\sigma$.

is needed to ensure the model lies on the ground. And j is set to be seven, covering more than half of the sample variance.

The point to model triangle distance minimization is treated as an one-dimensional case, i.e. the parameters are fitted sequentially, in the following order: rotation, translation and deformation (Figure 5.6). For each parameter, an iterative minimum bracketing procedure is applied. The scope of rotation angle is from -180° to 180°, translation parameters are between -0.5 and 0.5, and all the model parameters are between -3 and 3. The stopping criterion s of the iteration is set by examining the difference between the mean squared point to triangle distances of two successive iterations. Here, the point to triangle distance is the distance between each point p of an object hypothesis P and its closest triangle t among all the model triangles T. Let $E_x = \sum D(p,t) : [m,M] \in \mathbb{R}$ be the point to triangle distance function that need to be minimized for each parameter. The parameter scope $[m,M]$ is discretized by a factor N at each iteration. The point to triangle fitting algorithm is as follows:

Algorithm 1: lidar point to car model triangle fitting
Input: T and P
Output: fitted transformation and deformation parameters ($\mathbf{R}_1$, $\mathbf{T}_3$, $\mathbf{w}_7$)
For each $x \in$ ($\mathbf{R}_1$, $\mathbf{T}_3$, $\mathbf{w}_7$), $E_x^k : [m,M] \in \mathbb{R}$ (k is outer iteration number)
1. Set $x_i^j = m + i(M-m)/N$ for $i = 0..N$, (j is inner iteration number)
2. Find the optimal $i_j^* = argmin_i E(x_i^j)$,
3. Set $x_i^{j+1} = x_{i_j^*-1}^j + i(x_{i_j^*+1}^j - x_{i_j^*-1}^j)/N$,
4. If $\lvert E(x_{i_j^*}^j) - E(x_{i_{j+1}^*}^{j+1})\rvert < s$, stop; Else go back to step 2.
End
Set $E_x^k = E(x_{i_{j+1}^*}^{j+1})$,
If $\lvert E_x^k - E_x^{k+1}\rvert < s$, stop; Else go back to the beginning.

To facilitate the model fitting, some constraints and optimizations are implemented. First, it happens that there are many points inside cars due to window penetration. Whereas the model triangles are on the car surfaces. Thus the model fitting should be weighted in favour of the outer points of the modelled car. The 3D alpha shapes algorithm is applied to discriminate points that are inside cars. Then they are treated as non contributing points by setting the weights to zero. Second, there are normally no points at the bottom of any cars where the laser scanner does not reach, which means that there is no constraint in this direction. Since models do not go under ground, the ground plane is used as constraint. Third, cars have outside attachments, e.g.

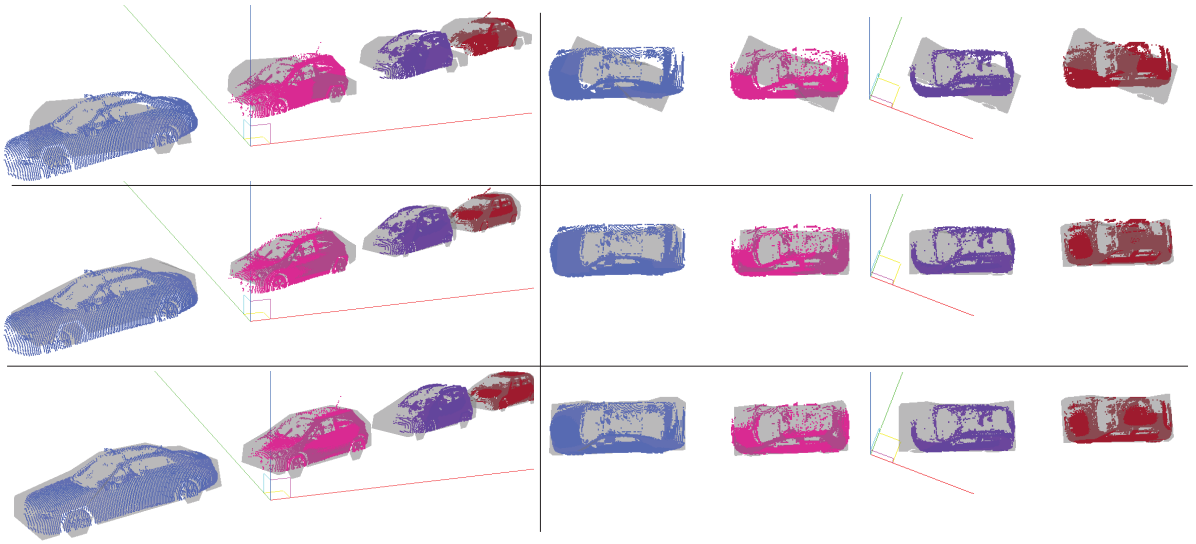


Figure 5.6 – Model fitting (left: perspective view, right: bird's eye view). Top: initialization, the mean model is assigned to each car hypothesis. Middle: first iteration, the orientation and general shape is determined. Bottom: final fitting result, object points to model triangle distances are minimized.

antennas, mirrors, which are outliers w.r.t. the car surface model. It also occurs that a nearby object is merged with a car due to under segmentation. Thus we use a robust, truncated distance function, i.e. the point to triangle distance is truncated by a maximum threshold (30 cm). The point to triangle distance minimization model fitting is comparable to the point-to-plane iterative closest point (ICP) algorithm (Pomerleau et al., 2013). Whereas in our case, the model triangles are transformable and deformable to fit the points which are kept fixed.

Model fitting residuals indicate the final mean squared distances between lidar points and model triangles. A small residual means a good fitting, but does not necessary mean a good model because the model can be over fitted. It happens when the number of points in one fitting direction is significantly lower than the another one which will then be over fitted. The model parameters convey the deformation information and represent the car shape w.r.t. the mean model. Then the representativeness of the model parameters is assessed by treating them as car features.

5.3.3 Car recognition and localization

To extract detailed information on street-side parking lots, we aim to not only recognize, but also precisely locate these cars. After segmentation, the next step is to detect whether a segment is a car, i.e. recognize cars from other street objects. It is a binary classification problem, car or non-car, which is commonly solved by supervised learning.

Feature extraction has been presented in Section 5.3.1. Apart from the extracted geometric features, model parameters are also treated as car features. Fitting a detailed accurate model to a street object can help to recognize whether it is a car, since a non-car object will have extreme fitting parameters or large residuals. Once a car is confirmed, its location and pose can be also retrieved from the fitted model. So the detailed car models can be used not only for recognition, but also for precise localization.

We experiment with two different classifiers, support vector machines (SVMs) using the

Gaussian radial basis function (RBF) (Cortes et Vapnik, 1995), and random forests (RF) (Breiman, 2001). SVMs try to maximize the margin between different classes in the training samples. Parameter C balances the complexity of the decision boundary and the cost of misclassification. The Gaussian kernel parameter γ defines the influence of the a training sample. A bigger γ means a smaller influence radius. Best C and γ are obtained by grid search based on 5-fold cross-validation results (Chang et Lin, 2011). We use F-score as the evaluation and feature ranking criterion instead of overall accuracy because the data are unbalanced (Chen et Lin, 2006). The RF takes a random number of features to build many decision trees, which are assembled and averaged. Again, the optimal parameters , e.g. number of trees, split quality function, tree depth, are exhaustively searched to have the highest cross validation result. One of the advantages of RF is that it will evaluate the importance of each feature, i.e. features are automatically ranked.

To figure out how the model features behave in the classification. Three sets of features are tested: only geometric features, only model features, and all the features. Here, the model parameters are indirectly compared with other geometric features.

5.3.4 Car classification

Fine-grained classification, i.e. detailed categorization of objects belonging to the same general class, has been increasingly studied for detailed 3D scene understanding (Stark et al., 2012; Lin et al., 2014; Zia et al., 2015). In our case, it is important to know the size and type of cars parking at a certain on-street parking lot, for instance a large delivery van is not welcome to park on a commercial street.

Given the detected vehicles and their features, vehicle classification can be achieved using the same supervised learning procedure. Notice that the distribution of vehicle categories may vary by region. For example, the majority of vehicles in most European cities are subcompact or compact vehicles, whereas in America, bigger vehicles, e.g. full size sedan, SUV, pickup truck, are generally preferred. Therefore, it is preferable to use a training set specifically chosen for the relevant region.

Once a vehicle is detected, its class can be determined using the previously extracted geometric and model features. Similarly, the three set of features are tested to check whether the model features are of any contribution. Both classifiers, SVMs and RF, are used to justify the results.

The vehicle recognition step and this classification/categorization step could be solved as one general classification problem by introducing a further class, non-vehicle, to the category. However, we prefer to treat the two steps separately as a two-step hierarchy, i.e. vehicles are first discriminated from other objects and then further classified into different categorises, which is usually reported to have better performances (Serna et Marcotegui, 2014).

It is worth mentioning that the classification is also useful for vehicle change detection, which compares whether two vehicles are identical. In theory, if two vehicles are determined to be in different categories, there is no need for further comparison. Thus even the classification step can be executed in parallel with the change detection, it is recommended to be processed priorly. The robustness of same vehicles being classified into the same categories is also tested.

5.3.5 Change detection

To check whether a car has moved away or is still parked in the same place, we scan the same street multiple times. Then the task is to find out whether cars from different passages (also referred to as scans in this chapter) at the same location are identical. Even though the MMS is geo-referenced, different passages are typically not well co-located due to positioning uncertainties. Thus registration is conducted using the method proposed by Monnier et al. (2013), who register all the lidar points to an existing 3D city model using non-rigid ICP. The overall registration accuracy is reported around 10 cm.

One straight froward way to detect car changes is to find out the corresponding cars in two datasets first, then compare them in the feature space. A general threshold is set to identify whether the difference is significant enough to be a change. However, a proper threshold can be difficult to find. We propose to use supervised learning to identify whether cars have changed across two data acquired at different times. The decision boundary between the same and different cars is learned from the training set rather then by setting a general threshold. The classifier is trained on pairs of cars. Same-car pairs and different-car pairs are both compared in the feature space and the difference is taken to compose a new training set. Since we need as many car pairs as possible for the training, all cars in the training data are compared with each other even if they are not corresponding cars so that the training data size is augmented. In practice, to find out whether a car has changed, only cars at approximately the same location or largely co-located need to be compared. As aforementioned, detected cars are precisely located. So if the distance to the corresponding car is much larger than the registration accuracy, there is no need for further change detection as this displacement is already evidence of a change.

Since the car features are extracted in the local coordinate system, the method is independent from absolute geo-locations. Moreover, different scanning perspectives, point densities, anisotropic sampling and partial occlusion are handled as long as they are represented by the training set.

Again, car model parameters are taken as features. The pair features are either the subtraction or the concatenation of features from each car. Both classifiers, SVMs and RF, are used to check the consistency of the results.

5.4 Experiments and results

5.4.1 Segmentation

The experimental data are acquired in Paris by the *Stereopolis* MMS (Paparoditis et al., 2012) using the VQ-250 RIEGL laser scanner. The laser beam rotates perpendicularly to the system moving trajectory, scanning profiles of the streets.

Figure 5.7 illustrates the segmentation result, in which each object is coloured randomly. There are 716 objects in the test area. Since cars are the objects of interest, we only evaluate the segmentation result of cars. Among all the street objects, there are 217 cars, 210 of which are correctly segmented. Two over and five under segmentations are observed in the result (Figure 5.8). Over segmentation is caused by low point density and severe occlusion so that cars fall apart. Under segmentations occur when an other object is too close to a car. The overall segmentation correct rate is 210/217 = 96.8%.

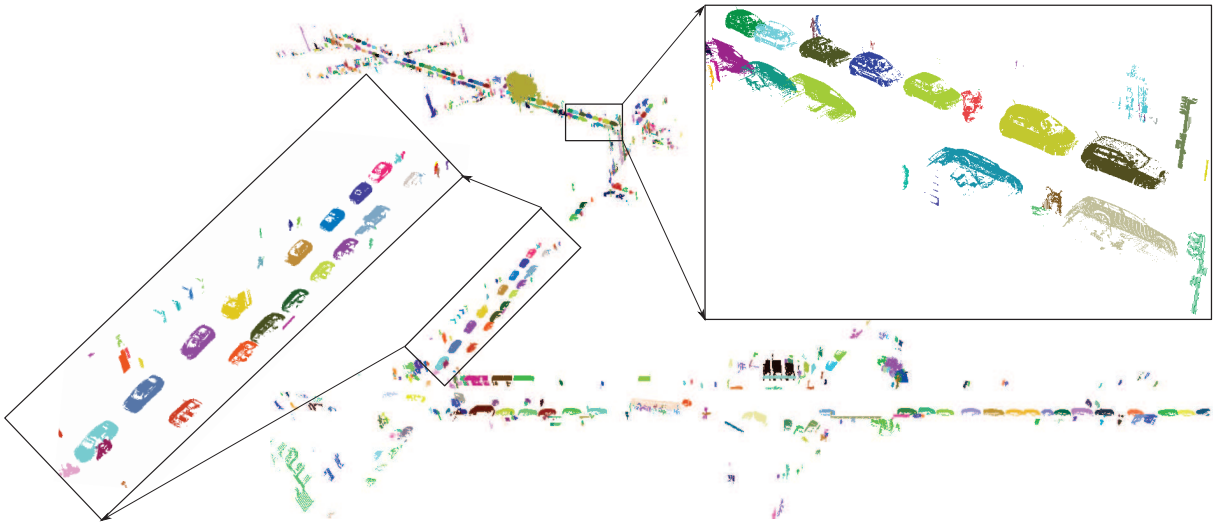


Figure 5.7 – Segmentation result, segments are coloured randomly

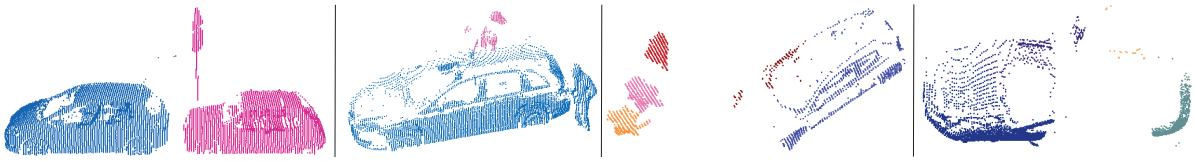


Figure 5.8 – Examples of segmentation failures. From left to right: (1) the reddish car is still connected to a road sign post since they are really close to each other; (2) a pedestrian is connected to a car, where as another one on the other side is correctly segmented; (3) a car is segmented into two parts due to low point density; (4) a car is also over segmented due to severe occlusion.

5.4.2 Car modelling

Figure 5.9 depicts the model fitting result in the test street. According to visual inspection, the car shapes are well modelled. Even the two big vans in the top right corner are well reconstructed considering that no vans are taken as training samples. It is clear that not only the precise location, but also the pose information can be retrieved from the fitted models. One can even easily estimate the location of car windows and doors by observing the car model.

However, there are cases that the model is not properly fitted as shown in Figure 5.10. The car has no points in the front part since it is scanned from the back, which causes false orientation when fitting because there is more constraints in the back. But this will not be the case for cars that are scanned from side hence will not affect the street-side parked car modelling.

Different cars will have different fitting parameters, and non-car objects will have extreme parameters (Figure 5.11). So they are used as features for the following classifications together with other geometric features.

Figure 5.12 illustrates both car models and their Bboxes. It is more reasonable to use the detailed models for visualization purposes. Moreover, models are more accurate in terms of localization because they are robust to outliers and certain degree of data incompleteness (Figure 5.13). More information, e.g. orientation, wheel's location, can also be retrieved from the

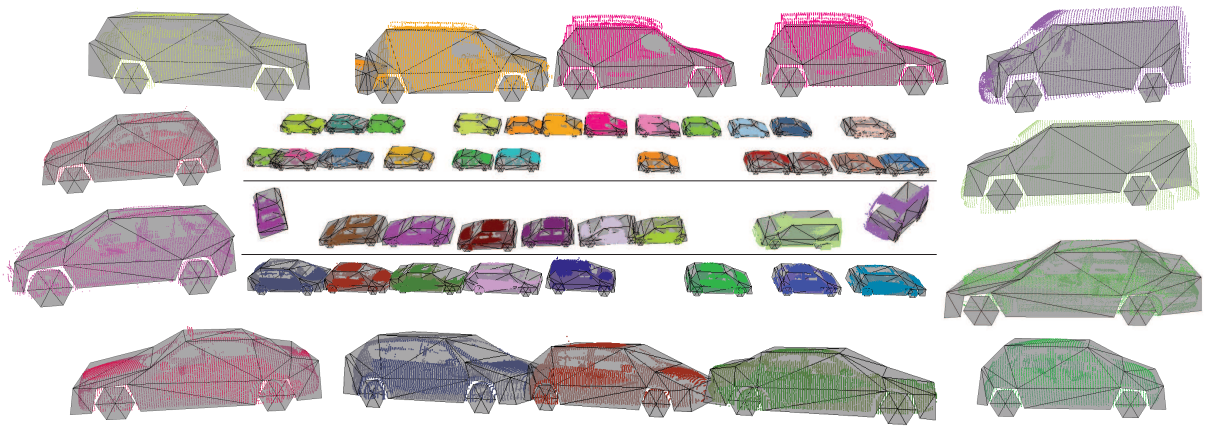


Figure 5.9 – Model fitting examples on test streets. Each car is illustrated by a random colour, and models are in light grey. Cars of different types, e.g. subcompact car, hatchback, sedan, SUV, minivan, van, are illustrated.

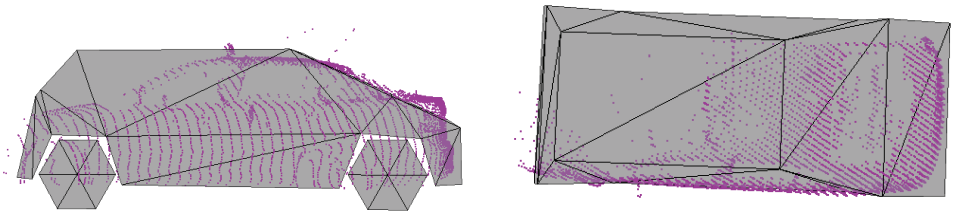


Figure 5.10 – Inappropriate fitting result (left: side view; right: bird's eye view). There are no points in the front since the car is scanned from the back. The model is misoriented because the back has more constraints.

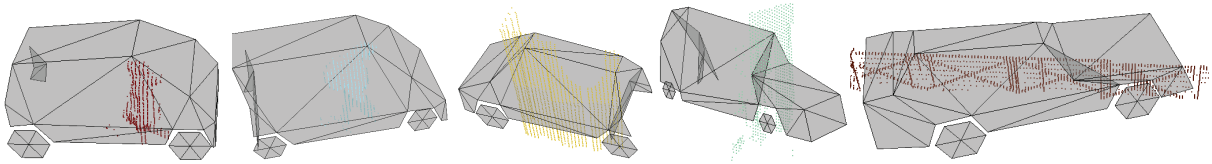


Figure 5.11 – Car model fitting with non-car objects, e.g. pedestrians, vertical surfaces, traffic barriers. Car models are extremely distorted.

models. Whereas for the Bbox, car attachments, e.g. mirrors, antennas, will severely affect the accuracy, and incomplete data will lead to inaccurate orientation determined by PCA.

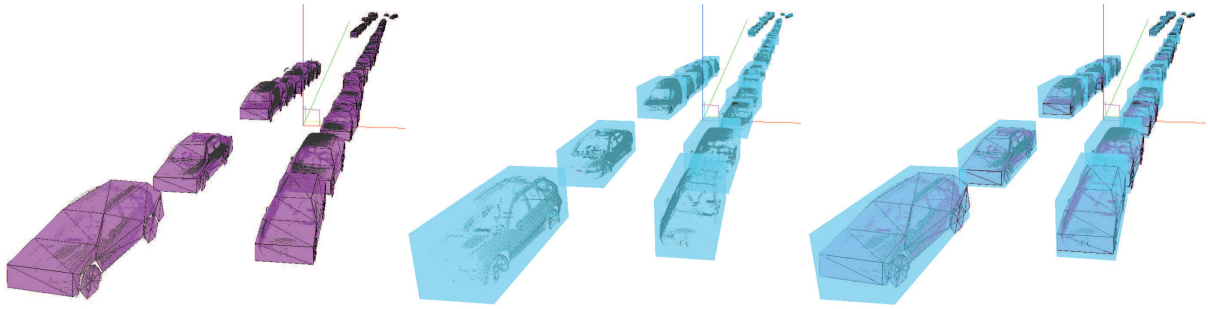


Figure 5.12 – Illustration of fitted models and Bboxes (lidar points in gray). Left: detailed models; middle: Bboxes; right: both (models in red, Bboxes in blue).

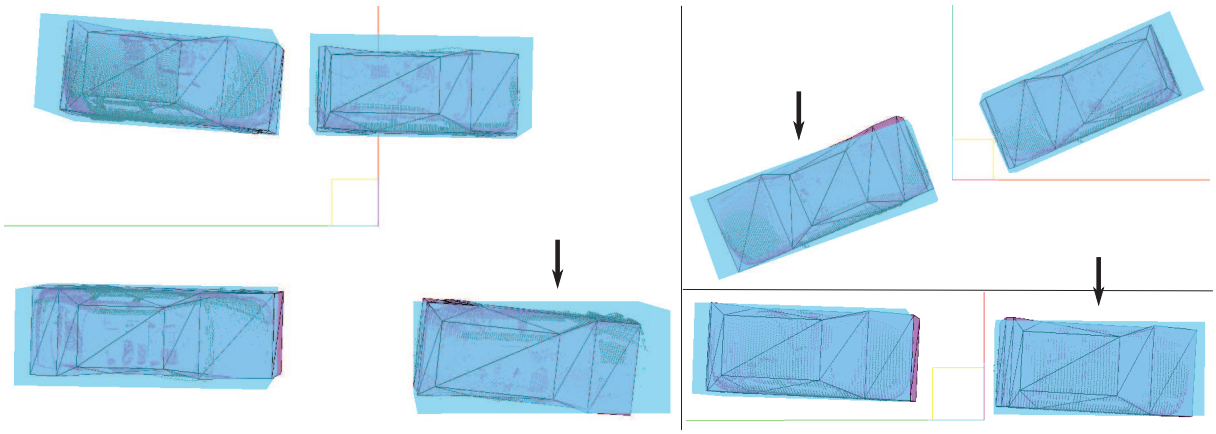


Figure 5.13 – Comparison examples of models (red) and Bboxes (blue). Bboxes' locations are less accurate than car models', especially the ones pointed by arrows.

5.4.3 Car recognition

After the segmentation, each segment is considered as an object hypothesis and its geometric features are extracted. Deformable car models are fitted to the objects. To evaluate the detection result, ground truth is manually labelled from the dataset as in Figure 5.14.

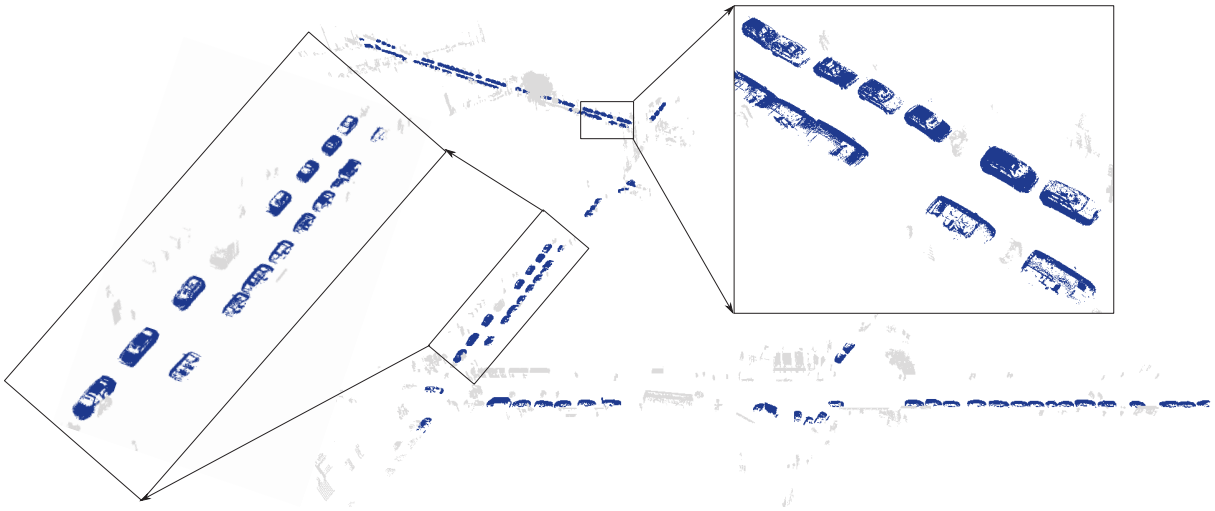


Figure 5.14 – Car detection ground truth data. Cars in blue, other objects in grey.

As described before, three sets of features are tested to check the importance of model features. The results are shown in Table 5.1. The accuracy, precision, recall and F-score are presented for each feature set using both SVMs and RF classifiers.

For the SVMs classifier, geometric features alone yield a result of high accuracy and recall, but relatively low precision. And the F-score is about 88%. Model features themselves do not perform as strong as geometric features since the recall is extremely low. However, the combination of these features (G+M) yields the best result with a balanced recall and precision, and highest accuracy and F-score. The RF classifier yields slightly higher values than the SVMs, whereas in general the value patterns are the same. Model features yield results with low recall, hence are not sufficient for the classification. But they contribute to the final results, generating higher accuracy and F-score than pure geometric features. Figure 5.15 illustrates the feature

Table 5.1 – Car detection binary classification results (in %) with different sets of features and classifiers. The highest value is in bold in each column.

Feature	SVMs				Random Forests			
	Acc	Rec	Pre	F-score	Acc	Rec	Pre	F-score
Geo	95.5	**94.3**	82.4	87.9	95.0	**86.8**	89.5	88.1
Model	89.4	50.6	82.2	62.6	90.1	53.0	84.0	65.0
G+M	**96.2**	90.1	**88.1**	**89.1**	**96.5**	86.7	**92.7**	**89.6**

importances calculated by the RF classifier, in which model features are in red. It is clear that they are of significant importance, especially the first three parameters.

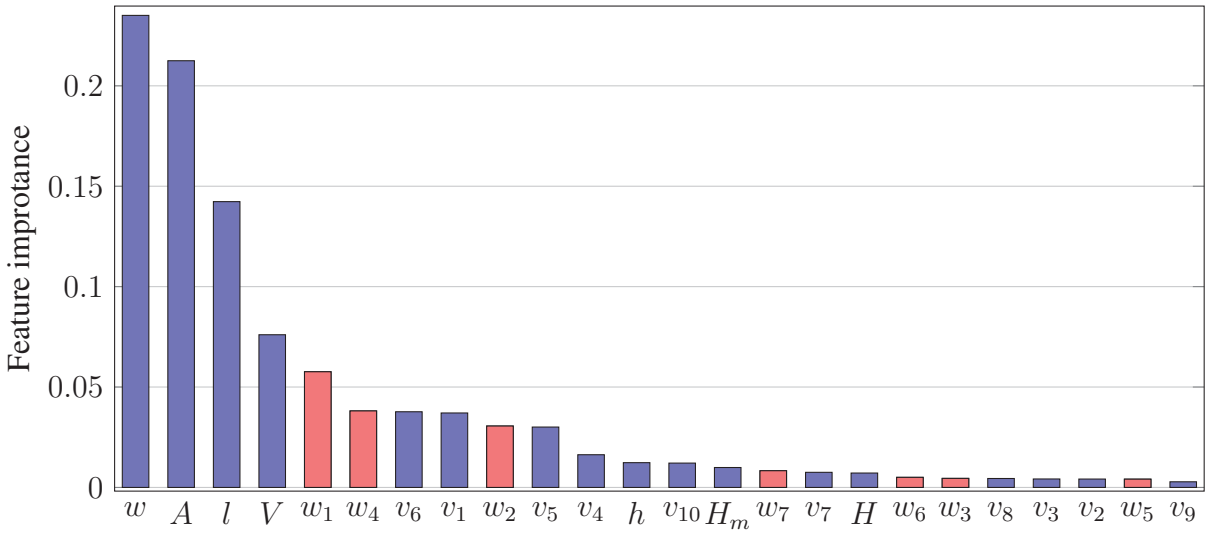


Figure 5.15 – Feature importance ranking. Model features ($w_1 \ldots w_7$) are in light red.

The results suggest that the model features improve the car recognition from lidar point cloud. Even the improvement is moderate, the car models convey more detailed information. Furthermore, they provide precise car location and orientation.

5.4.4 Car classification

Covering different types of cars, 73 training samples are selected as shown in Table 5.2. Since there are not many samples, they are grouped into four main categories: subcompact (mini or small), compact (hatchback), full-size cars (sedan, station wagon, SUV, MPV) and vans. The categories can be further divided if more training samples are available.

Table 5.2 – Car samples for categorization

Type	SubComp	Compact	FullSize	Van
Num.	37	17	11	8

The three sets of features, model features, geometric features, and the combination, are tested using both SVMs and RF classifiers. Since the training set is not large, the classifiers are trained by 2-fold cross validation. Then the two test results are added up. The normalized confusion matrices are depicted in Figure 5.16. The first class, subcompact car, has good results using all the three feature sets, since this type of car can be easily distinguished from others due to its small size. Whereas in other classes, size and shape are mostly mixed hence the accuracies are lower. SVMs and RF have slightly different values, but globally they show the same pattern. It is clear that geometric features perform better than model features, and the combined feature set performs the best using both classifiers.

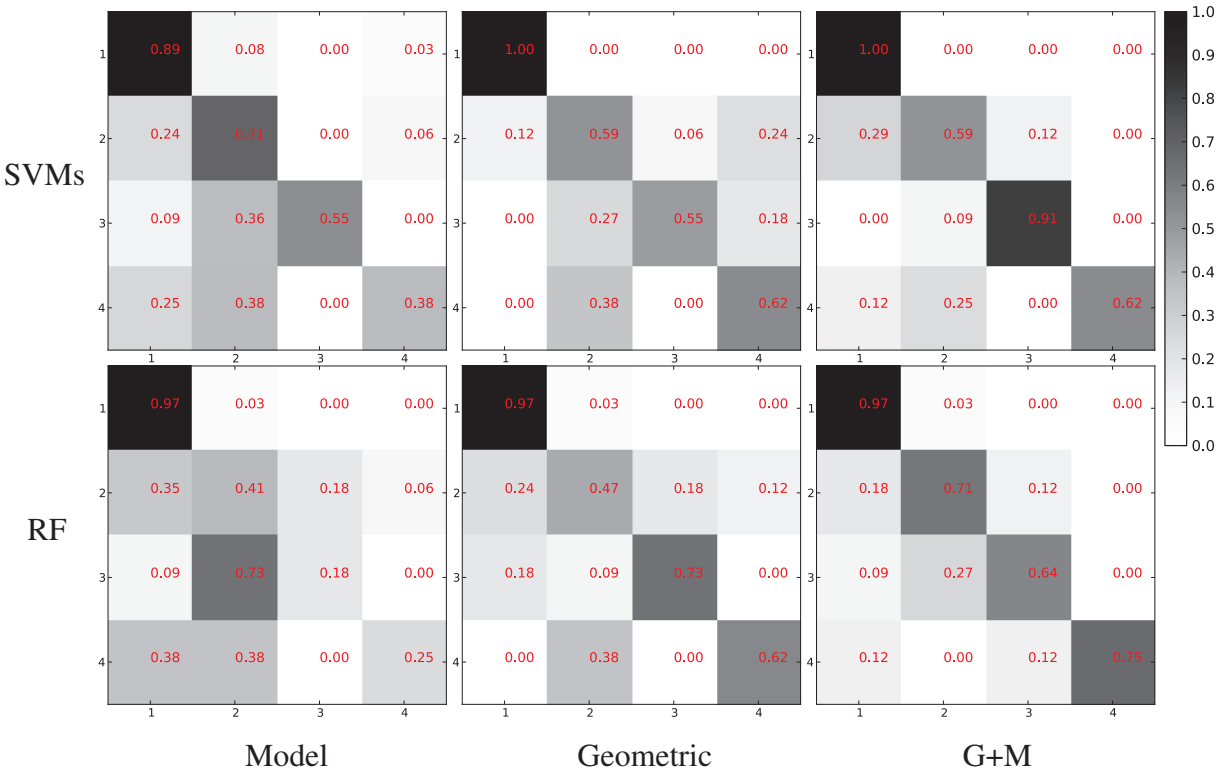


Figure 5.16 – Normalized confusion matrices of car classification using various feature sets and classifiers. Class 1: subcompact car; 2: compact car; 3: full size car; 4: van. Geometric features perform better than model features, and the combined feature set performs the best using both classifiers.

The overall accuracy using each feature set and each classifier is shown in Table 5.3. The accuracies are generally not very high, because the problem is more complicated and the training data size is small. However, the results still confirm that the model features have contributions for the classification.

Table 5.3 – Car classification accuracy using different feature sets and classifiers.

Feature	SVMs	RF
Geo	79.5	78.1
Model	73.97	64.4
G+M	84.9	83.6

As aforementioned, there is no need for change detection of car from different classes given

that the classification is accurate. 11 cars from four passages (44 cars in total) are tested, where only one car has changed. Despite the correctness of class, all unchanged cars are classified into the same classes by the SVMs classifier. And only two unchanged cars are classified twice into different classes by the RF classifier. It means that the classification results are highly reliable so that further change detection is unnecessary.

5.4.5 Change detection

A total of 987 car pairs, 83 of which are same-car pairs, are extracted from four passages of the same street. One of the passages is sub-sampled from another, so that different point densities are presented in the training data.

Again, three feature sets are tested shown in Table 5.4. Here, the features are the subtraction of the features from the paired-up cars. As for the SVMs classifier, model features themselves are not sufficient for the change detection. Geometric features yield better results than the combined features. Thus, in this case, the model features do not improve the classification. The results are consistent with the ones using the RF classifier, which yields the best accuracy and F-score using only the geometric features. Notice that the objective is to detect whether a car has changed across two epoch datasets. Even though model features do not improve the results, the final F-score and accuracy are high using only the geometric features. The concatenations of the paired car features are also tested. However, the results are significantly lower than the subtractions. The highest F-score is 63.2% using the geometric features alone by the SVMs classifier.

Table 5.4 – Car change detection results (in %) with different sets of features and classifiers. The highest value is in bold in each column.

Feature	SVMs				Random Forests			
	Acc	Rec	Pre	F-score	Acc	Rec	Pre	F-score
Geo	**98.7**	**92.8**	**91.7**	**92.2**	**97.8**	**81.6**	92.5	**86.7**
Model	92.1	33.7	54.9	41.8	93.0	15.7	66.0	25.4
G+M	98.5	**92.8**	89.5	91.1	97.4	78.0	**93.0**	84.8

If a car has moved away and there is no other car parking in the same place, there will be no correspondence. This type of change can be easily be detected. Then for cars with correspondences, they are paired-up and tested by the supervised learning to detect whether they are the same. Car change detection examples are shown in Figure 5.17. Same or different cars can be successfully detected despite point density, scanning perspective, car pose and type. However failures caused by under segmentations, e.g. cars clustered with pedestrians, are observed. Severe data missing caused by occlusions will also induce false detections (Figure 5.18).

5.5 Conclusion

We present a comprehensive system for on-street car park monitoring, which is composed of a series of research problems, i.e. car recognition, localization, classification and change detection. The system is able to recognize and precisely locate cars with the help of detailed

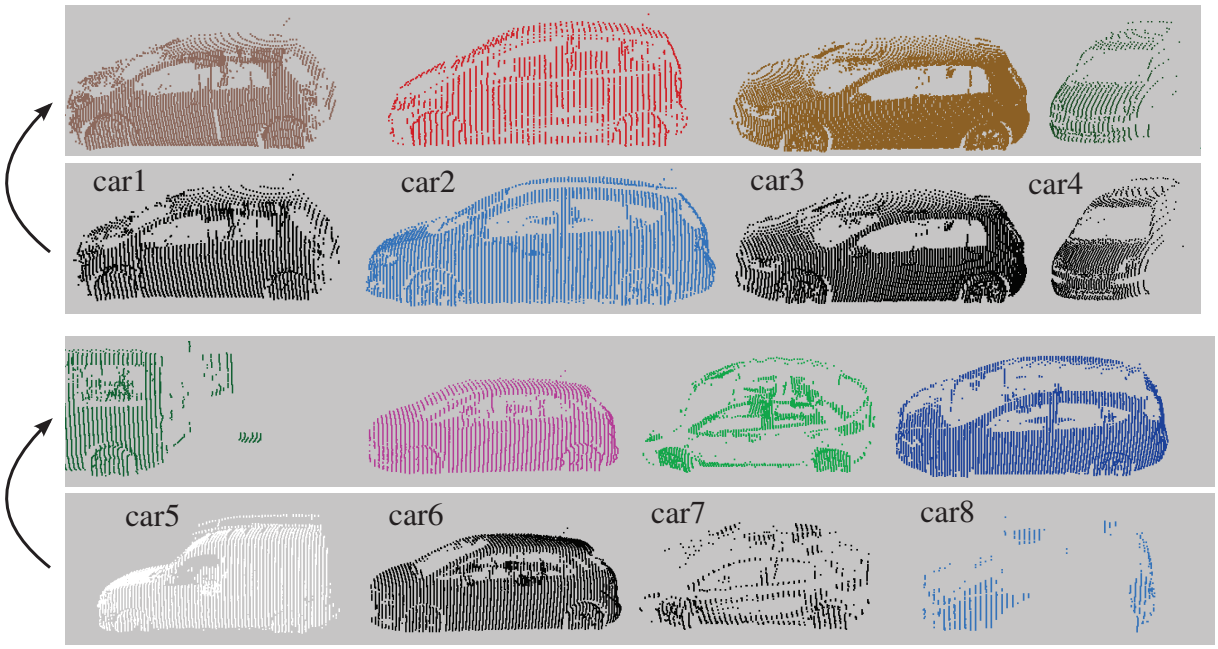


Figure 5.17 – Change detection examples. Cars in rows one and three are reference cars, each of which is in a random color. Even rows are change detection results. Detected unchanged cars are in black, and changed cars in blue. Cars without correspondences are in white.

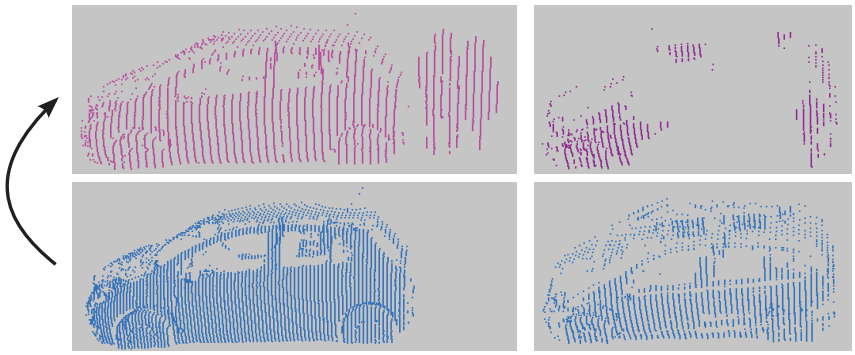


Figure 5.18 – Change detection failure examples. Both unchanged cars are detected as changed. The left one is caused by segmentation error, and the right one is due to sever data incompleteness of the reference car.

car modelling. It also identifies the general classes/categories of cars. Moreover, whether a parked car has changed is detected to potentially estimate the parking durations. The benefit of detailed car modelling is threefold: the model parameters can be used as car features for supervised classifications; the fitted model provides precise location and pose information; tangible and detailed models can be directly used for visualization purposes.

The system is powerful considering that it relies on supervised learning for decision making rather than setting specific thresholds. However, the performance is determined by the representativeness of training sets. The reported results can be improved since the test is limited by the training data size. Future work will investigate the automation of training sample annotation to improve the system performance.

Chapter 6

Conclusion and perspectives

Contents

As mentioned in the very beginning, the world is constantly changing. Studying of the world dynamics, and the evolution of human activities are always of great demand. Laser scanning is somewhat an accurate technique to record and witness the changes.
Specific conclusions have been drawn in each chapter. In this chapter, the work is summarized in general. And some perspectives is presented in the end.

6.1 Summary

This work is conducted as a part of the TerraMobilita project that aims to upgrade the current urban geo-database, management system and applications empowered by a mobile mapping system, particularly with state-of-the-art laser scanners. We focused on the change detection of LSPC for street map updating, temporary object removal (data cleansing), etc. In the application part, we utilized Velodyne LSPC for accurate pedestrian flow estimation to help the redesign and monitoring of urban public places. Moreover, we investigated the on-street car park monitoring using RIEGL LSPC.

The work is organized into three main parts:

1. Street environment change detection. Change detection at street level is challenging for several reasons, e.g the level of detail in high, street objects are of various shapes and sizes, and most importantly the occlusion problem. Classical distance-based method is straight-forward and efficient but is suffering from occlusions. Occupancy-grid helps to distinguish the space that are occluded. In fact, no actual grid is necessary because the occupancy is modelled in the laser scanner's reference frame and the space is represented by the points themselves other than artificial grids. By looking from the scanner's perspective, the gaps between scan lines can be easily interpolated. Changes are examined directly on each point by its neighbouring rays from the comparing epoch. The limit of occupancy-based method is its sensibility to penetrable objects, which is not the case for distance-based method. Thus it is used to refine the results from a point-to-triangle distance-based method. The combined method shows promising results on real data in a complex street environment. This part is presented in Chapter 3.

2. Pedestrian flow estimation. This part of work is based on the previous one. The occupancy-based method is adapted to the Velodyne laser scanning geometry. In this case, the moving objects are continuously detected over time. Moving object detection is considered as one type of change detection with short temporal intervals. Actually, the temporal interval between two comparing frames depends on the pedestrian moving speed. A moving object is not compared with its consecutive scanned instance but with the space that are supposed to be empty since it should have moved out of the space. In other words, a moving object will move out of its occupied space after some time, so to find out whether an object is moving, we need to check whether the occupied space become empty after a while, but not necessarily for a long time. The detected moving object instances over a long time can visually reveal the general pedestrian moving pattern. Still, to accurately estimate the pedestrian flow, individual trajectories are recovered. Since the points are not clustered/labelled yet, we can not simply link the object instance in each frame to form the trajectory. Instead, we have to cluster all the points belonging to the same object over time, and at the mean time distinguish pedestrians from other objects and noises. This is accomplished by a simultaneous detection and tracking (SDAT) algorithm, which formalizes the tracking as an energy minimization problem incorporating data attachment, sensitivity, generalization, and smoothness terms. The problem is solved by a two-step optimization: RANSAC tracklet detection, and tracklet association by clique graph. The method can recover trajectories that are partially scanned. Quantitative results are evaluated and compared with state-of-the-art methods using benchmark data. This part is presented in Chapter 4.

3. On-street car park monitoring. Parking lot monitoring includes not only the occupancy

rate status, but also the car location, category and parking duration, especially for the on-street parking places which normally have stricter regulations. Apparently, this topic deals with data at object level. So cars are recognized in the first place. Then their locations and categories are determined. Moreover, the parking duration can be regulated by checking whether the car parking at the same place is the same as the previous one. All the processes are handled in a supervised learning framework. Features are extracted, and training sets are labelled by hand. In particular, a model-based reconstruction method is proposed to recover the shape of cars, and also to provide additional information, e.g. precise location and model features. The reconstructed models can also be used for visualization purposes. In this application, changes are detected at object level which is different from previous two applications. The supervised classification procedure for change detection shows promising results. This part is presented in Chapter 5.

6.2 Conclusion

A specific conclusion has been drawn in each part. Here, general conclusions are presented. In this research, the change detection from laser scanning point clouds have been studied at multiple temporal-scales with different types of laser scanners, RIEGL and Velodyne. Both bottom-up and top-down change detection strategies are properly applied to solve practical problems. Related topics, e.g. tracking, object recognition, classification, and modelling are also explored.

Bottom-up vs. top-down Change detection at point level, i.e. bottom-up change detection, corresponds to the pixel-based method in remote sensing studies. Changes are detected at low level, meaning further meaningful information have to be extracted based on the changes. This step can generally serve as a pre-processing step, for data cleansing (short-term change, temporal objects), static environment mapping, database updating (long-term, permanent changes) and so on. The drawback of this strategy is that false alarms and miss detections (false negative) will affect the following processes. Top-down change detection, i.e. object-based change detection, can be considered, on the contrary, as high level data processing. Objects are identified firstly, then the changes are detected. The results can usually be directly used to support decision making. The limitation is that the object detection errors will induce false changes.

The first part uses bottom-up strategy which is an obvious choice because the majority of the lidar points are not changed. It can be an unaccomplishable task to classify all the data and then detect changes at object level. The detected changed points do not necessarily constitute a complete object, e.g. points on opened windows. They are difficult to be classified. This is one of the reasons that the changes are not yet recognized in this research. As for database updating, further information, e.g. changed point locations, sizes, can be generated to help the operator for verification. The second part follows the same strategy. Even other bottom-up change detection methods are proposed for the static data acquisition mode. Whereas, for the purpose of moving object tracking, one common method, i.e. tracking-by-detection, is to detect the object of interest first. The advantage of our bottom-up method is that partially scanned or occluded objects are retained as evidence of tracks. Mostly the object detection methods will omit these evidence. The third part naturally uses the top-down method because only cars are of interest in the data. Cars have to be extracted anyway for localization, categorization and modelling. Moreover, subtle changes can hardly be detected by neither occupancy nor distance-

based methods because the buffer can be greater than the car difference. Accurate features are needed for precise decisions.

Both strategies have their own pros and cons. Different tasks may require different strategies, choices have to be made accordingly.

Long-term vs. short-term Temporal scale can vary for different applications. In this research, change detection has not been limited by the temporal resolution. In fact, we also investigated instant change detection, better be referred to as moving object detection (MOD). MOD is important because the moving objects can be either removed from the data as temporary objects, or treated as objects of interest for tracking. In the scope of the Terramobilita project, one needs to assess the accessibility of urban public space for wheelchairs. Then temporary object, e.g. pedestrians, bikes, wheeled trash can, on pavements should not be stored as obstacles in the database. On the other hand, the trajectories of moving objects/pedestrians in public squares are useful information for the management and rearrangement.

The occupancy-based change detection method in this research is temporal scale independent. So it has been used for both long-term change detection and short-term moving object detection. In the first part, the temporal scale can vary from minutes to years, thanks to the convenience of the MMS. Long-term, e.g. months, years, changes will most probably be permanent constructions in the street environment, hence can be used for database updating. Whereas changes over minutes or hours are mostly temporary object presences, so they should be removed from the data. Then the rather cleaned data can be used for classification, modelling, mapping, and comparison with the current database for updating. As for the moving object detection, strictly speaking, it is not instant change detection as the current frame is compared with not the consecutive frames but those within a temporal window, during which the originally occupied space is supposed to be empty. Nevertheless, the occupancy-based method is successfully adapted to the new scanning geometry and overlapped moving objects of very small time gap are detected. The temporal scale of the on-street car change detection depends on the parking regulations. For instance, in certain areas of Paris, cars are allowed to park on the street side for maximum two hours. In most of the cases, cars can be parked at the same public parking spot for seven consecutive days. The monitoring frequency can be days or hours as the object-based change detection is time independent.

Lidar, Computer Vision, and Robotics Change detection can also be found in other closely related research domains, e.g. computer vision and robotics.

Lidar range data, especially Velodyne data, have been widely used in Robotics for environment perception, SLAM, navigation, etc. The occupancy-based change detection method is originally adapted from occupancy-grid in robotics. The grid is abandoned because the information is carried directly by the points themselves. The space is interpolated based on the scanning geometry, so that the method is adaptable to different laser scanners. The change detection method does not only detect the moving object, but also extract the static environment at the same time.

Besides moving object detection, we also investigated the tracking problem, which is a classical topic in both Robotics and Computer Vision. Moving object tracking has been conducted in the context of geo-pattern estimation for urban planning and management. Furthermore, model-based vision is firstly adopted in this research using lidar point clouds to facilitate scene understanding, including object recognition and localization. Models are directly reconstructed

in 3D. They can also be used for visualization and simulation. Even thought lidar range data are not the main data source for Computer Vision, they are increasingly used for modelling and scene understanding, especially the RGBD data. In this research, both Robotics and Computer Vision methods are used for lidar change detection.

6.3 Perspectives

Regarding the method of each part, further improvements can be investigated:

- As for the first part, distance-based and occupancy-based are simply processed one after another, a mathematical integration of these two methods is worth considering, even tough a better result is not guaranteed. Point-to-triangle distance is based on the Delaunay triangulation. The occupancy can be modelled according to the triangular prism composed by the scanner center and each triangle instead of each ray.
- For moving object tracking, the only criterion to distinguish pedestrian from others is just the point density in a 2D circle, which is not robust enough. Features, apart from pure geometric constraints, can be incorporated.
- For car park monitoring, a semi-automatic training data generation will further strengthen the whole system. More features can be added to the classifiers.

In this research, we used only lidar data since they are geometrically accurate. Obviously, image integration can enhance the feature information for change detection, object recognition, and tracking regardless the different data acquisition geometries. The occupancy-based change detection method uses only geometrical information, image information can be difficult to integrate. However, they can be used to rectify and therefore improve the results. Especially, object boundaries in lidar data are not clear as in images. Image information can also be used to improve pedestrian tracking by distinguish pedestrians from other and also among themselves. Moreover, for car change detection, color features should even be treated prior to geometric ones. Actually, intensity information of lidar data is to be used. However it is not consistent since it is affected by many factors, e.g. distance, incidence angle, object material.

Further studies can be explored based on the results of this work:

- The change detection can be used for temporary object removal to have cleaner data for modelling and accessibility assessment. Naturally, there will be data missing in the occluded areas, especially for RIEGAL data. Single passage of data acquisition will most probably be incomplete. So several cleaned data can be incrementally merged for complete street mapping. For Velodyne data, the static environment is generated. Even with movable objects, which can be classified and filtered out afterwards, it can be used for route planning and environment mapping.
- The moving objects in lidar data can be used as hint to remove moving objects in corresponding images. They are not supposed to be projected to street-view images or models. Moving object removal from images can also facilitate image matching, structure from motion (SfM), and street side modelling because it avoids tie points on moving objects that appear in multiple images. These tie points will hinder the camera orientation and hence deteriorate the matching results.

- Moving objects are only detected at point level, so point cloud segmentation and object recognition can be carried out for better scene understanding and tracking.
- Model-based object reconstruction has only been used for car modelling, it can be easily applied to other objects since the basic mean model is constructed by only a few number of points.

Bibliography

Aeschliman, C., Park, J., Kak, A. C., 2010. A probabilistic framework for joint segmentation and tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1371–1378.

Agarwal, S., Awan, A., Roth, D., 2004. Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11), 1475–1490.

Aijazi, A. K., Checchin, P., Trassoudaine, L., 2013. Detecting and Updating Changes in Lidar Point Clouds for Automatic 3D Urban Cartography. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. II-5/W2. pp. 7–12.

Andriluka, M., Roth, S., Schiele, B., 2008. People-tracking-by-detection and people-detection-by-tracking. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8.

Arróspide, J., Salgado, L., Marinas, J., 2012. HOG-like gradient-based descriptor for visual vehicle detection. In: IEEE Intelligent Vehicles Symposium. pp. 223–228.

Banno, A., Masuda, T., Oishi, T., Ikeuchi, K., 2008. Flying laser range sensor for large-scale site-modeling and its applications in Bayon digital archival project. International Journal of Computer Vision 78 (2-3), 207–222.

Benedek, C., Shadaydeh, M., Kato, Z., Szirányi, T., Zerubia, J., 2015. Multilayer Markov Random Field models for change detection in optical remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing 107, 22–37.

Bensrhair, A., Bertozzi, M., Broggi, A., Fascioli, A., Mousset, S., Toulminet, G., 2002. Stereo vision-based feature extraction for vehicle detection. In: IEEE Intelligent Vehicle Symposium. Vol. 2. pp. 465–470.

Bernardin, K., Stiefelhagen, R., 2008. Evaluating multiple object tracking performance: the CLEAR MOT metrics. Journal on Image and Video Processing 2008, 1–10.

Betke, M., Haritaoglu, E., Davis, L. S., 2000. Real-time multiple vehicle detection and tracking from a moving vehicle. Machine vision and applications 12 (2), 69–83.

Bileschi, S. M., Leung, B., Rifkin, R. M., 2004. Towards component-based car detection. In: ECCV Workshop on Statistical Learning and Computer Vision.

Börcs, A., Benedek, C., 2012. A marked point process model for vehicle detection in aerial LIDAR point clouds. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. I-3. pp. 93–98.

Breiman, L., 2001. Random forests. Machine learning 45 (1), 5–32.

Butkiewicz, T., Chang, R., Wartell, Z., Ribarsky, W., 2008. Visual analysis and semantic exploration of urban lidar change detection. In: Computer Graphics Forum. Vol. 27. Wiley Online Library, pp. 903–910.

Caraffi, C., Vojir, T., Trefny, J., Sochman, J., Matas, J., 2012. A system for real-time detection and tracking of vehicles from a single car-mounted camera. In: IEEE International Conference on Intelligent Transportation Systems. pp. 975–982.

Champion, N., Rottensteiner, F., Matikainen, L., Liang, X., Hyyppä, J., Olsen, B., 2009. A test of automatic building change detection approaches. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVIII-3/W4. pp. 145–150.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27.

Chauve, A.-L., Labatut, P., Pons, J.-P., 2010. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1261–1268.

Chen, Y.-W., Lin, C.-J., 2006. Combining SVMs with Various Feature Selection Strategies. In: Feature Extraction. Springer, pp. 315–324.

Choi, K., Lee, I., Kim, S., 2009. A feature based approach to automatic change detection from LiDAR data in Urban areas. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVIII. pp. 259–264.

Cootes, T. F., Taylor, C. J., Cooper, D. H., Graham, J., 1995. Active shape models-their training and application. Computer vision and image understanding 61 (1), 38–59.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine learning 20 (3), 273–297.

Dehghan, A., Assari, S. M., Shah, M., 2015. GMMCP Tracker: Globally Optimal Generalized Maximum Multi Clique Problem for Multiple Object Tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Vol. 1. p. 2.

Demantké, J., 2014. Reconstruction de modéles 3d photoréalistes de façades á partir de données image et laser terrestre. Thèse de Doctorat, Université Paris-Est.

Demantké, J., Mallet, C., David, N., Vallet, B., 2011. DIMENSIONALITY BASED SCALE SELECTION IN 3D LIDAR POINT CLOUDS. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVIII-5/W12. pp. 97–102.

Eden, I., Cooper, D. B., 2008. Using 3D line segments for robust and efficient change detection from multiple noisy images. In: European Conference on Computer Vision (ECCV). pp. 172–185.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9), 1627–1645.

Ferryman, J., Worrall, A., Sullivan, G., Baker, K., 1995. A Generic Deformable Model for Vehicle Recognition. In: British Machine Vision Conference 1995.

Fischler, M. A., Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24 (6), 381–395.

Früh, C., Zakhor, A., 2004. An automated method for large-scale, ground-based city model acquisition. International Journal of Computer Vision 60 (1), 5–24.

Girardeau-Montaut, D., Roux, M., Marc, R., Thibault, G., 2005. Change detection on points cloud data acquired with a ground laser scanner. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 36. pp. 30–35.

Golovinskiy, A., Kim, V. G., Funkhouser, T., 2009. Shape-based recognition of 3D point clouds in urban environments. In: International Conference on Computer Vision. IEEE, pp. 2154–2161.

Gong, J., Sui, H., Ma, G., Zhou, Q., 2008. A review of multi-temporal remote sensing data change detection algorithms. The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences 37 (Part B7), 757–762.

Greene, N., 1994. Detecting intersection of a rectangular solid and a convex polyhedron. In: Graphics Gems IV. Elsevier, pp. 74–82.

Gressin, A., Vincent, N., Mallet, C., Paparoditis, N., 2013. Semantic approach in image change detection. In: Advanced Concepts for Intelligent Vision Systems. pp. 450–459.

Gupte, S., Masoud, O., Martin, R. F., Papanikolopoulos, N. P., 2002. Detection and classification of vehicles. IEEE Transactions on Intelligent Transportation Systems 3 (1), 37–47.

Hebel, M., Arens, M., Stilla, U., 2013. Change detection in urban areas by object-based analysis and on-the-fly comparison of multi-view ALS data. ISPRS Journal of Photogrammetry and Remote Sensing 86, 52–64.

Hernández, J., Marcotegui, B., 2009a. Filtering of artifacts and pavement segmentation from mobile LIDAR data. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVIII. pp. 329–333.

Hernández, J., Marcotegui, B., 2009b. Point cloud segmentation towards urban ground modeling. In: Joint Urban Remote Sensing Event. pp. 1–5.

Himmelsbach, M., Müller, A., Lüttel, T., Wünsche, H.-J., 2008. LIDAR-based 3D object perception. In: Proceedings of 1st International Workshop on Cognition for Technical Systems.

Hinz, S., Schlosser, C., Reitberger, J., 2003. Automatic car detection in high resolution urban scenes based on an adaptive 3D-model. In: GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas. pp. 167–171.

HosseinyAlamdary, S., Yilmaz, A., 2014. Merging Trajectory and Point Clouds for 3D Object Tracking. In: Photogrammetric Computer Vision Symposium Tracking Challenge.

Hsiao, K., Liu, J., Yu, M., Tseng, Y., 2004. Change detection of landslide terrains using ground-based LiDAR data. In: XXth ISPRS Congress, Istanbul, Turkey, Commission VII, WG VII/5. Vol. 5.

Huang, C.-L., Liao, W.-C., 2004. A vision-based vehicle identification system. In: International Conference on Pattern Recognition.

Hussain, M., Chen, D., Cheng, A., Wei, H., Stanley, D., 2013. Change detection from remotely sensed images: From pixel-based to object-based approaches. ISPRS Journal of Photogrammetry and Remote Sensing 80, 91–106.

Irani, M., Anandan, P., 1998. A unified approach to moving object detection in 2D and 3D scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (6), 577–589.

Jazayeri, A., Cai, H., Zheng, J. Y., Tuceryan, M., 2011. Vehicle detection and tracking in car video based on motion model. IEEE Transactions on Intelligent Transportation Systems 12 (2), 583–595.

Kaempchen, N., Franke, U., Ott, R., 2002. Stereo vision based pose estimation of parking lots using 3D vehicle models. In: IEEE Intelligent Vehicle Symposium. Vol. 2. pp. 459–464.

Kaestner, R., Maye, J., Pilat, Y., Siegwart, R., 2012. Generative object detection and tracking in 3d range data. In: IEEE International Conference on Robotics and Automation. pp. 3075–3081.

Kang, Z., Zhang, L., Yue, H., Lindenbergh, R., 2013. Range Image Techniques for Fast Detection and Quantification of Changes in Repeatedly Scanned Buildings. Photogrammetric Engineering & Remote Sensing 79 (8), 695–707.

Keat, C. T. M., Pradalier, C., Laugier, C., 2005. Vehicle detection and car park mapping using laser scanner. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2054–2060.

Košecka, J., 2013. Detecting changes in images of street scenes. In: Asian Conference on Computer Vision (ACCV). pp. 590–601.

Koyasu, H., Miura, J., Shirai, Y., 2001. Real-time omnidirectional stereo for obstacle detection and tracking in dynamic environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 1. pp. 31–36.

Lafarge, F., Mallet, C., 2011. Building large urban environments from unstructured point data. In: IEEE International Conference on Computer Vision (ICCV). pp. 1068–1075.

Leotta, M., Mundy, J., 2009. Predicting high resolution image edges with a generic, adaptive, 3-D vehicle model. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1311–1318.

Leotta, M. J., Mundy, J. L., 2010. Vehicle Surveillance with a Generic, Adaptive, 3-D Vehicle Model. IEEE transactions on pattern analysis and machine intelligence 33 (7), 1457–1469.

Lin, Y.-L., Morariu, V. I., Hsu, W., Davis, L. S., 2014. Jointly Optimizing 3D Model Fitting and Fine-Grained Classification. In: European Conference on Computer Vision (ECCV). pp. 466–480.

Lindenbergh, R., Pietrzyk, P., 2015. Change detection and deformation analysis using static and mobile laser scanning. Applied Geomatics 7 (2), 65–74.

Lindström, M., Eklundh, J.-O., 2001. Detecting and tracking moving objects from a mobile platform using a laser range scanner. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 3. pp. 1364–1369.

Lu, D., Mausel, P., Brondizio, E., Moran, E., 2004. Change detection techniques. International Journal of Remote Sensing 25 (12), 2365–2401.

Mallet, C., Bretar, F., 2009. Full-waveform topographic lidar: State-of-the-art. ISPRS Journal of Photogrammetry and Remote Sensing 64, 1–16.

Matikainen, L., Hyyppä, J., Hyyppä, H., 2003. Automatic detection of buildings from laser scanner data for map updating. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 34-3/W13. pp. 218–224.

Milan, A., Leal-Taixé, L., Schindler, K., Reid, I., 2015. Joint tracking and segmentation of multiple targets. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5397–5406.

Milan, A., Roth, S., Schindler, K., 2014. Continuous energy minimization for multitarget tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (1), 58–72.

Monnier, F., 2014. Amélioration de la localisation 3D de véhicules mobiles á l'aide de cartes ou modéles 3D. Thèse de Doctorat, Université Paris-Est.

Monnier, F., Vallet, B., Paparoditis, N., Papelard, J.-P., David, N., 2013. Registration of terrestrial mobile laser data on 2D or 3D geographic database by use of a non-rigid ICP approach. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. II-5/W2. pp. 193–198.

Monnier, F., Vallet, B., Soheilian, B., 2012. Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. I-3. pp. 245–250.

Moosmann, F., Stiller, C., 2011. Velodyne slam. In: IEEE Intelligent Vehicles Symposium. pp. 393–398.

Moosmann, F., Stiller, C., 2013. Joint self-localization and tracking of generic objects in 3D range data. In: IEEE International Conference on Robotics and Automation. pp. 1146–1152.

Moras, J., Cherfaoui, V., Bonnifait, P., 2011. Credibilist occupancy grids for vehicle perception in dynamic environments. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 84–89.

Murakami, H., Nakagawa, K., Hasegawa, H., Shibata, T., Iwanami, E., 1999. Change detection of buildings using an airborne laser scanner. ISPRS Journal of Photogrammetry and Remote Sensing 54 (2), 148–152.

Murakami, H., Nakagawa, K., Shibata, T., Iwanami, E., 1998. Potential of an airborne laser scanner system for change detection of urban features and orthoimage development. International Archives of Photogrammetry and Remote Sensing 32, 422–427.

Niskanen, S., Östergård, P. R. J., 2003. Cliquer User's Guide, Version 1.0. Tech. rep., Communications Laboratory, Helsinki University of Technology, Espoo, Finland.

O'Callaghan, R., Haga, T., 2007. Robust change-detection by normalised gradient-correlation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8.

Pagac, D., Nebot, E., Durrant-Whyte, H., 1996. An evidential approach to probabilistic map-building. In: IEEE International Conference on Robotics and Automation (ICRA). Vol. 1. pp. 745–750.

Paparoditis, N., Papelard, J., Cannelle, B., Devaux, A., Soheilian, B., David, N., Houzay, E., 2012. Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology. Revue Française de Photogrammétrie et de Télédétection 200, 69–79.

Pollard, T., Mundy, J. L., 2007. Change detection in a 3-d world. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–6.

Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S., 2013. Comparing ICP variants on real-world data sets. Autonomous Robots 34 (3), 133–148.

Poullis, C., 2013. A Framework for Automatic Modeling from Point Cloud Data. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 2563–2575.

Pu, S., Rutzinger, M., Vosselman, G., Oude Elberink, S., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS Journal of Photogrammetry and Remote Sensing 66 (6), S28–S39.

Qin, R., Gruen, A., 2014. 3D change detection at street level using mobile laser scanning point clouds and terrestrial images. ISPRS Journal of Photogrammetry and Remote Sensing 90, 23–35.

Radke, R. J., Andra, S., Al-Kofahi, O., Roysam, B., 2005. Image change detection algorithms: a systematic survey. IEEE Transactions on Image Processing 14 (3), 294–307.

Reid, D. B., 1979. An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control 24 (6), 843–854.

Rutzinger, M., Ruf, B., Höfle, B., Vetter, M., 2010. Change detection of building footprints from airborne laser scanning acquired in short time intervals. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVIII. pp. 475–480.

Sakurada, K., Okatani, T., Deguchi, K., 2013. Detecting Changes in 3D Structure of a Scene from Multi-view Images Captured by a Vehicle-Mounted Camera. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 137–144.

Schindler, K., Ess, A., Leibe, B., Van Gool, L., 2010. Automatic detection and tracking of pedestrians from a moving stereo rig. ISPRS Journal of Photogrammetry and Remote Sensing 65 (6), 523–537.

Schindler, K., James, U., Wang, H., 2006. Perspective n-view multibody structure-and-motion through model selection. In: European Conference on Computer Vision (ECCV). pp. 606–619.

Serna, A., Marcotegui, B., 2013. Urban accessibility diagnosis from mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing 84, 23–32.

Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. ISPRS Journal of Photogrammetry and Remote Sensing 93, 243–255.

Shackleton, J., VanVoorst, B., Hesch, J., 2010. Tracking People with a 360-Degree Lidar. In: IEEE International Conference on Advanced Video and Signal Based Surveillance. pp. 420–426.

Singh, A., 1989. Review Article Digital change detection techniques using remotely-sensed data. International Journal of Remote Sensing 10, 989–1003.

Spinello, L., Luber, M., Arras, K. O., 2011. Tracking people in 3D using a bottom-up top-down detector. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1304–1310.

Stark, M., Krause, J., Pepik, B., Meger, D., Little, J., Schiele, B., Koller, D., 2012. Fine-Grained Categorization for 3D Scene Understanding. In: British Machine Vision Conference. pp. 1–12.

Steinle, E., Bahr, H., 2003. Detectability Of Urban Changes From Airborne Laser scanning Data. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 34-7/B. pp. 1135–1140.

Steinle, E., Oliveira, F., Bähr, H.-P., Loch, C., October 1999. Assessment of Laser Scanning Technology for Change Detection in Buildings. In: Proceedings XVIIth CIPA Symposium. Olinda, Brazil.

Sullivan, G., Baker, K., Worrall, A., Attwood, C., Remagnino, P., 1997. Model-based vehicle detection and classification using orthographic approximations. Image and Vision Computing 15 (8), 649–654.

Sullivan, G., Worrall, A., Ferryman, J., et al., 1995. Visual object recognition using deformable models of vehicles. In: Workshop on Context-Based Vision. Vol. 19. pp. 75–86.

Taneja, A., Ballan, L., Pollefeys, M., 2011. Image based detection of geometric changes in urban environments. In: IEEE International Conference on Computer Vision (ICCV). pp. 2336–2343.

Taneja, A., Ballan, L., Pollefeys, M., 2013. City-Scale Change Detection in Cadastral 3D Models Using Images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 113–120.

Tian, J., Reinartz, P., d'Angelo, P., Ehlers, M., 2013. Region-based automatic building and forest change detection on Cartosat-1 stereo imagery. ISPRS Journal of Photogrammetry and Remote Sensing 79, 226–239.

Toshev, A., Mordohai, P., Taskar, B., 2010. Detecting and parsing architecture at city scale from range data. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 398–405.

Toth, C., Barsi, A., Lovas, T., 2003. Vehicle recognition from LiDAR data. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 3/W13. pp. 163–166.

Tuermer, S., Kurz, F., Reinartz, P., Stilla, U., 2013. Airborne vehicle detection in dense urban areas using HoG features and disparity maps. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 6 (6), 2327–2337.

Ulusoy, A. O., Mundy, J. L., 2014. Image-based 4-d reconstruction using 3-d change detection. In: European Conference on Computer Vision (ECCV). pp. 31–45.

Underwood, J. P., Gillsjo, D., Bailey, T., Vlaskine, V., 2013. Explicit 3D change detection using ray-tracing in spherical coordinates. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 4735–4741.

Vallet, B., 2013. Homological persistence for shape based change detection between Digital Elevation Models. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 1. pp. 49–54.

Vallet, B., Soheilian, B., Brédif, M., 2014. Combinatorial clustering and Its Application to 3D Polygonal Traffic Sign Reconstruction From Multiple Images. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 1. pp. 165–172.

Vallet, B., Xiao, W., Brédif, M., 2015. Extracting Mobile Objects in Images Using a Velodyne LIDAR Point Cloud. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 1. pp. 247–253.

Velizhev, A., Shapovalov, R., Schindler, K., 2012. Implicit shape models for object detection in 3D point clouds. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 3. pp. 179–184.

Verma, V., Kumar, R., Hsu, S., 2006. 3d building detection and modeling from aerial lidar data. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Vol. 2. pp. 2213–2220.

Vögtle, T., Steinle, E., 2004. Detection and recognition of changes in building geometry derived from multitemporal laserscanning data. International Archives of the Photogrammetry, Remote sensing and Spatial Information Sciences 35 (B2), 428–433.

Vosselman, G., Gorte, B., Sithole, G., 2004. Change detection for updating medium scale maps using laser altimetry. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 34/B3. pp. 12–23.

Vu, T., Matsuoka, M., Yamazaki, F., 2004. LIDAR-based change detection of buildings in dense urban areas. In: IEEE International Geoscience and Remote Sensing Symposium. Vol. 5. IEEE, pp. 3413–3416.

Walter, V., 2004. Object-based evaluation of LIDAR and multispectral data for automatic change detection in GIS databases. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 35. pp. 723–728.

Walter, V., 2005. Object-based classification of integrated multispectral and LIDAR data for change detection and quality control in urban areas. In: Proceedings of the ISPRS WG VII/1 Human Settlements and Impact Analysis.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing 105 (0), 286 – 304.

Weinmann, M., Jutzi, B., Mallet, C., 2014. Semantic 3D scene interpretation: a framework combining optimal neighborhood size selection with relevant features. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. II-3. pp. 181–188.

Wolf, D., Sukhatme, G. S., 2004. Online simultaneous localization and mapping in dynamic environments. In: IEEE International Conference on Robotics and Automation (ICRA). Vol. 2. pp. 1301–1307.

Wu, B., Nevatia, R., 2007. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. International Journal of Computer Vision 75 (2), 247–266.

Xiao, W., Vallet, B., Brédif, M., Paparoditis, N., 2015. Street environment change detection from mobile laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 107, 38–49.

Xiao, W., Vallet, B., Paparoditis, N., 2013. Change Detection in 3D Point Clouds Acquired by a Mobile Mapping System. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. II-5/W2. pp. 331–336.

Xiao, W., Xu, S., Oude Elberink, S., Vosselman, G., 2012. Change detection of trees in urban areas using multi-temporal airborne lidar point clouds. In: SPIE Remote Sensing. Vol. 853207. pp. 853207–853207–10.

Xu, S., Vosselman, G., Oude Elberink, S., 2013. Detection and Classification of Changes in Buildings from Airborne Laser Scanning Data. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. II-5/W2. pp. 343–348.

Xu, S., Vosselman, G., Oude Elberink, S., 2014. Multiple-entity based classification of airborne laser scanning data in urban areas. ISPRS Journal of Photogrammetry and Remote Sensing 88, 1–15.

Yang, B., Dong, Z., 2013. A shape-based segmentation method for mobile laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 81, 19–30.

Yao, W., Hinz, S., Stilla, U., 2008. Automatic vehicle extraction from airborne LiDAR data of urban areas using morphological reconstruction. In: IAPR Workshop on Pattern Recognition in Remote Sensing. pp. 1–4.

Yao, W., Hinz, S., Stilla, U., 2011. Extraction and motion estimation of vehicles in single-pass airborne LiDAR data towards urban traffic analysis. ISPRS Journal of Photogrammetry and Remote Sensing 66 (3), 260–271.

Yin, Z., Collins, R., 2007. Belief propagation in a 3D spatio-temporal MRF for moving object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8.

Yu, X., Hyyppä, J., Kaartinen, H., Maltamo, M., 2004. Automatic detection of harvested trees and determination of forest growth using airborne laser scanning. Remote Sensing of Environment 90 (4), 451–462.

Yu, X., Hyyppa, J., Kukko, A., Maltamo, M., Kaartinen, H., 2006. Change detection techniques for canopy height growth measurements using airborne laser scanner data. Photogrammetric engineering and remote sensing 72 (12), 1339.

Zamir, A. R., Dehghan, A., Shah, M., 2012. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In: European Conference on Computer Vision (ECCV). pp. 343–356.

Zeibak, R., Filin, S., 2007. Change detection via terrestrial laser scanning. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 36-3/W52. pp. 430–435.

Zhang, Z., Tan, T., Huang, K., Wang, Y., 2012. Three-dimensional deformable-model-based localization and recognition of road vehicles. IEEE Transactions on Image Processing 21 (1), 1–13.

Zhou, L., Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. International Journal of Applied Earth Observation and Geoinformation 18, 293–304.

Zhou, Q.-Y., Neumann, U., 2010. 2.5D dual contouring: A robust approach to creating building models from aerial lidar point clouds. In: European Conference on Computer Vision (ECCV). pp. 115–128.

Zia, M. Z., Stark, M., Schiele, B., Schindler, K., 2013. Detailed 3D Representations for Object Recognition and Modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11), 2608–2623.

Zia, M. Z., Stark, M., Schindler, K., 2015. Towards Scene Understanding with Detailed 3D Object Representations. International Journal of Computer Vision 112 (2), 188–203.