Road Information Extraction from Mobile LiDAR Point Clouds using Deep Neural Networks

by

Lingfei Ma

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Geography

Waterloo, Ontario, Canada, 2020

©Lingfei Ma 2020

Examining Committee Membership

The following served on the Examining Committee members for this thesis. The final decision of the Examining Committee is by majority vote.

Supervisor:	**Dr. Jonathan Li** Geography & Environmental Management, University of Waterloo
Co-Supervisor:	**Dr. Michael A. Chapman** Civil Engineering (Adjunct GEM), Ryerson University
Internal Member:	**Dr. Richard Kelly** Geography & Environmental Management, University of Waterloo
Internal-External Member:	**Dr. John S. Zelek** System Design Engineering, University of Waterloo
External Member:	**Dr. Naser El-Sheimy** Geomatics Engineering, University of Calgary

AUTHOR'S DECLARATION

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This doctoral thesis is accomplished in the manuscript option, following the graduation guidelines administered by the joint Waterloo-Laurier Graduate Program in Geography. Three required manuscripts have been published as refereed journal papers, which present in Chapters 3-5, respectively. These three publications, as presented below, have been accordingly revised for a consistent format.

- **Ma L**, Li Y, *Li J, Tan W, Yu Y, Chapman M. A., 2019. Multi-Scale Point-Wise Convolutional Neural Networks for 3D Object Segmentation from LiDAR Point Clouds in Large-Scale Environments. *IEEE Trans. Intell. Transp. Syst*. Doi: 10.1109/TITS.2019.2961060.

- **Ma L**, Li Y, *Li J, Yu Y, Junior J, Gonçalves W, Chapman M. A., 2020. Capsule-based Networks for Road Marking Extraction and Classification from Mobile LiDAR Point Clouds. *IEEE Trans. Intell. Transp. Syst*. Doi: 10.1109/TITS.2020.2990120.

- **Ma L**, Li Y, *Li J, Junior J, Gonçalves W, Chapman M. A., 2020. BoundaryNet: Extraction and completion of road boundaries with deep learning using MLS point clouds and satellite imagery. *IEEE Trans. Intell. Transp. Syst*. (under minor revision).

For all three manuscripts, I am the first author, and my supervisor Prof. Dr. Jonathan Li is the corresponding author. They are dominated by my intellectual effort. I have major contributions in designing the ideas and methods, implementing experiments, and writing these papers. Other co-authors also contributed to these papers in terms of document preparation and review and paper editing.

Abstract

Urban roads, as one of the essential transportation infrastructures, provide considerable motivations for rapid urban sprawl and bring notable economic and social benefits. Accurate and efficient extraction of road information plays a significant role in the development of autonomous vehicles (AVs) and high-definition (HD) maps. Mobile laser scanning (MLS) systems have been widely used for many transportation-related studies and applications in road inventory, including road object detection, pavement inspection, road marking segmentation and classification, and road boundary extraction, benefiting from their large-scale data coverage, high surveying flexibility, high measurement accuracy, and reduced weather sensitivity. Road information from MLS point clouds is significant for road infrastructure planning and maintenance, and have an important impact on transportation-related policymaking, driving behaviour regulation, and traffic efficiency enhancement.

Compared to the existing threshold-based and rule-based road information extraction methods, deep learning methods have demonstrated superior performance in 3D road object segmentation and classification tasks. However, three main challenges remain that impede deep learning methods for precisely and robustly extracting road information from MLS point clouds. (1) Point clouds obtained from MLS systems are always in large-volume and irregular formats, which has presented significant challenges for managing and processing such massive unstructured points. (2) Variations in point density and intensity are inevitable because of the profiling scanning mechanism of MLS systems. (3) Due to occlusions and the limited scanning range of onboard sensors, some road objects are incomplete, which considerably degrades the performance of threshold-based methods to extract road information.

To deal with these challenges, this doctoral thesis proposes several deep neural networks that encode inherent point cloud features and extract road information. These novel deep learning models have been tested by several datasets to deliver robust and accurate road information extraction results compared to state-of-the-art deep learning methods in complex urban environments. First, an end-to-end feature extraction framework for 3D point cloud segmentation is proposed using dynamic point-wise convolutional operations at multiple scales. This framework is less sensitive to data distribution and computational power. Second, a capsule-based deep learning framework to extract and classify road markings is developed to update road information and support HD maps. It demonstrates the practical application of combining capsule networks with hierarchical feature encodings of georeferenced feature images. Third, a novel deep learning framework for road boundary completion is developed using MLS point clouds and satellite imagery, based on the U-shaped network and the conditional deep convolutional generative adversarial network (c-DCGAN). Empirical evidence obtained from experiments compared with state-of-the-art methods demonstrates the superior performance of the proposed models in road object semantic segmentation, road marking extraction and classification, and road boundary completion tasks.

Acknowledgements

First of foremost, I would like to express the sincerest gratitude to my supervisor Prof. Dr. Jonathan Li, who provides me not only professional research knowledge but also life guidance during my PhD program. Without his valuable insight, technical advice, and positive encouragement, it would be pretty challenging for me to complete this research. I also would like to express my special thanks to my co-supervisor Prof. Dr. Michael A. Chapman for his professional research guidance and financial support.

I would like to express gratitude and appreciation to the Thesis Examining Committee members, including Prof. Dr. Richard Kelly (internal examiner) from the Department of Geography and Environmental Management, University of Waterloo; Prof. Dr. John S. Zelek (internal-external examiner) from the Department of Systems Design Engineering, University of Waterloo; and Prof. Dr. Naser El-Sheimy (external examiner) from the Department of Geomatics Engineering, University of Calgary. Their insightful suggestions and valuable comments improve the clarity of this thesis.

Many thanks to my colleagues at the Geospatial Sensing and Data Intelligence (GSDI) Lab, University of Waterloo, Ying Li, Ming Liu, Zilong Zhong, Weikai Tan, Dedong Zhang, Zheng Gong, Juan Manuel Carrillo Garcia, Nannan Qin, and Hongjie He, for their excellent academic assistance and lifelong friendships. Many thanks to Mr. Mike Lackner from the Centre for Mapping, Analysis & Design (MAD), Faculty of Environment, University of Waterloo, for his necessary technical supports.

Finally and most significantly, I am deeply thankful to my dear parents and families in China, who provide their endless love and emotional encouragement to keep my positive attitude during my studies. Special thanks to my girlfriend for her help, understanding, and accomplishment.

Lingfei Ma

August 14, 2020

Table of Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two dimensional
3D	Three dimensional
ADAS	Advanced Driver-Assistance System
ALS	Airborne Laser Scanning
AV	Autonomous Vehicle
BCE	Binary Cross Entropy
BLS	Backpack Laser Scanning
BN	Batch Normalization
c-GAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DBM	Deep Boltzmann Machine
DBN	Deep Belief Network
DGCNN	Dynamic Graph Convolutional Neural Network
DL	Deep Learning
EKF	Extended Kalman Filter
FC	Fully connected
FPS	Farthest Point Sampling
FPFH	Fast Point Feature Histograms
GAN	Generative Adversarial Network
GNSS	Global Navigation Satellite System
GSD	Ground Sample Distance
GSPN	Generative Shape Proposal Network
HD	High-definition
HKS	Heat Kernel Signature
IDW	Inverse Distance Weighting
IMU	Inertial Measurement Unit
IoU	Intersection over Union
KNN	K-Nearest Neighbour
LiDAR	Light Detection and Ranging

LRF	Local Reference Frame
LSCF	Least Squares Curve Fitting
MCR	Misclassification Rate
mIoU	mean Intersection over Union
MLP	Multi-layer Perceptron
MLS	Mobile Laser Scanning
MSE	Mean Squared Error
MsGAN	Multi-supervised Generative Adversarial Network
MS-PCNN	Multi-scale Point-wise Convolutional Neural Network
MSE	Mean Squared Error
MSTV	Multiscale Tensor Voting
MV3D	Multi-view Three Dimensional
OA	Overall Accuracy
PCL	Point Cloud Library
ReLU	Rectified Linear Unit
RoPS	Rotational Projection Statistics
RTK	Real-time Kinematic
SAR	Synthetic Aperture Radar
SDF	Spatial Density Filtering
SHOT	Signature of Histograms of OrienTations
SIFT	Scale-invariant Feature Transform
SSIM	Structural Similarity Index
SURF	Speeded Up Robust Feature
TLS	Terrestrial Laser Scanning
WNDH	Weighted Neighbouring Difference Histogram

Chapter 1

Introduction

1.1 Motivations

Urban roads, as one of the essential transportation infrastructures, provide considerable motivations for rapid urban sprawl and bring notable economic and social benefits (Wang et al., 2012). Detailed road asset inventories are commonly applied to support extensive transportation applications, such as city planning, construction surveying, smart cities, high-definition (HD) maps, and autonomous vehicles (AVs) (Pu et al., 2011). Generally, road objects (e.g., roadside trees, buildings, power lines, road markings, and pavement cracks) and road geometries (e.g., lane width, curvatures, and slopes) are manually collected and generated by expert-annotated digital maps or field survey in large-scale road environments. However, such methods for inspecting road information and extracting road infrastructures are labour-intensive and cost-consuming.

Mobile laser scanning (MLS) systems comprising Light Detection and Ranging (LiDAR) sensors can collect highly dense and accurate point clouds in urban roads and highways (Ma et a., 2018). The point density collected by MLS systems can achieve over 10,000 pts/m^2 with mm-level absolute measurement accuracy, while airborne and terrestrial laser scanning (ALS/TLS) cannot achieve such precision and flexibility (Chen et al., 2019a). Different from remotely sensed imagery collected through various platforms and sensors, such as drone-based imagery and satellite imagery, MLS systems are less sensitive to weather and ambient luminance conditions. They have been widely used for many transportation-related studies and applications in road information inventory, including road object detection and segmentation (Li et al., 2019a), pavement detection (Ye et al., 2019), road marking classification (Rastiveis et al., 2020), and road boundary extraction (Wen et al., 2019a). Thus, road information from MLS point clouds is significant for road infrastructure planning and maintenance, and have an important impact on transportation-related policymaking, driving behaviour regulation, and traffic efficiency enhancement.

Although the notable development in MLS systems has motivated a technical transformation for HD maps in both remote sensing and cartography communities, the massive and increasing point cloud data volume requires intelligent data processing and analysis methods. The MLS point clouds are usually unorganized with varying point densities and intensities, resulting in the poor performance of many threshold-based and rule-based methods for road information extraction. Thus, to perform road infrastructure mapping and support the development of HD maps, intelligent point cloud processing techniques are required to solve many challenging problems, including road object semantic segmentation, road marking extraction and classification, and road boundary extraction and completion in complex urban conditions.

Road object segmentation usually serves as the first step to extract road information. Road object semantic segmentation classifies each point in the entire road point clouds into several homogeneous classes, and the points belonging to the same objects or regions will have the same semantic labels (Nguyen and Le, 2018). In practice, MLS systems capture different road objects in the form of 3D points. However, it is very challenging to achieve automated and effective point-wise segmentation because of the high redundancy, inevitable distortion, and inexplicit structure of MLS point clouds (Wen et al., 2019b). Thus, it is significant to introduce an effective and robust method for point-wise road object segmentation.

Road markings play a significant role in guiding, regulating, and forbidding all road participants (Bétaille and Toledo-Moreo, 2010). It is necessary to extract and classify road markings from raw data and then assign specific class labels for their different categories. Due to occlusions, distortions, and intensity variations from MLS systems, most of the existing threshold-based extraction methods and rule-based classification methods cannot deliver accurate and robust results (Wan et al., 2019; Guan et al., 2014; Jung et al., 2019). Moreover, manual editing and post refinement operations are required to improve the completeness and accuracy of extracted road markings. Still, it is time-consuming and labour-intensive. Recently, deep learning (DL) is taking off in remote sensing and 3D vision communities (Zhu et al., 2017). Deep neural networks are firstly trained using manually annotated road marking samples as training samples, to learn inherent features and establish relationships between predictions and labels, which has indicated a promising solution for road marking extraction and classification tasks.

Road boundaries designate allowable driving zones and provide auxiliary road information to develop HD maps and fully autonomous driving (Holgado-Barco et al., 2017). Road boundary extraction is generally implemented through segmenting road surfaces, followed by detecting road curbs from MLS point clouds. Although considerable improvement has been achieved, most of the off-the-shelf methods cannot completely and accurately extract road boundaries, due to occlusions and point density variations during raw MLS data acquisition (Soilán et al., 2019). Accordingly, road boundaries collected by MLS systems are usually incomplete with many missing parts. It is still challenging to automatically obtain complete and accurate road boundaries. One of the most straightforward ways is to use MLS systems to collect incomplete data multiple times. A more practical and effective alternative is to restore such incomplete road boundaries by using deep learning-based methods.

Furthermore, the point clouds obtained from MLS systems are in an irregular format and unstructured distribution, which requires upgrading not only the whole data collection infrastructure (i.e., platforms, workstations, processing software, network communication, and data analysis) but also intelligent point cloud processing algorithms to extract road information. Efficient road information extraction requires well-designed models that capture the inherent features of different road objects and additional information, taking advantage of the increased performance of computational resources. One promising solution is to use deep learning models, e.g., deep convolutional neural networks (CNNs), to encode deeper and more distinctive feature representations. Thus, benefiting from many publicly accessible MLS point cloud datasets, such as KITTI (Geiger et al., 2012), Paris-Lille-3D (Roynard et al., 2018), and Toronto3D (Tan et al., 2020), both advanced MLS techniques and state-of-the-art deep learning networks (e.g., PointCNN) provide the underlying motivations for this thesis to develop robust and efficient road information extraction methods.

Compared to the existing threshold-based and rule-based road information extraction methods, recent studies indicate that deep neural networks could achieve superior performance in 3D road object segmentation and classification tasks. However, three main challenges remain that impede deep learning-based methods from precisely and robustly extracting road information from MLS point clouds:

(1). **Massive and unstructured point clouds**: Point clouds captured by MLS systems are always in large-volume and irregular formats, which is significantly challenges for managing and processing such massive unstructured points. Various methods, including voxel-based methods (Maturana and Scherer, 2015), multiview-based methods (Chen et al., 2017a), auto-encoder-based methods (Wang et al., 2016), graph-cut-based methods (Simonovsky and Komodakis, 2017), and symmetric function-based methods (Qi et al., 2017a), have been proposed to use CNNs for 3D data analysis, object recognition, and point-wise semantic segmentation. However, it is still challenging to effectively and automatically manipulate MLS point clouds with unstructured 3D points, various point densities, outliers, and occlusions, which are inevitable in complex urban environments.

(2). **Point density and intensity variations:** Point clouds are usually acquired by vehicle-mounted MLS systems that are driven through changing lanes at varying driving speeds. Because of the profiling scanning mechanism of MLS systems, the incident angle of laser beams grows larger with an increased scanning range. Consequently, point clouds have higher intensity values and point densities if they are closer to the trajectory of MLS systems. Still, it is difficult to effectively extract road information for most threshold-based methods, which assume that intensities and point densities are uniformly distributed.

(3). **Incomplete and worn road conditions**: Due to occlusions and the limited scanning range of LiDAR sensors, some road objects are usually incomplete during data acquisition. Road surface damages caused by on-road overloaded trucks and severe weather conditions, such as acid-alkali erosion, could lead to worn and decaying pavements. Moreover, occlusions from all road participants (e.g., vehicles and cyclists) also complicate the accurate extraction and classification of road objects. Accordingly, manual editing and post refinement can improve the completeness and accuracy of extracted road objects, but reducing the generalization ability and efficiency of proposed methods.

1.2 Objectives and Contributions

MLS point clouds have shown a promising solution for road information extraction, contributing to the generation of HD maps. To tackle the challenges mentioned above, this thesis proposes several deep neural networks that encode inherent point cloud features to deliver robust and accurate results in road information extraction, compared to state-of-the-art deep learning methods in complex urban conditions. More specifically, the three main objectives of this thesis, focusing on road object segmentation, road marking extraction and classification, and road boundary recovery, respectively, from MLS point clouds to support the HD map generation, are presented as follows:

(1). To develop a 3D semantic segmentation deep learning model, called MS-PCNN, that facilitates point-wise CNNs on unstructured 3D point clouds. Although there have been remarkable improvements in semantic HD map and fully autonomous driving domains, most of the existing point cloud segmentation models cannot deliver high feature representativeness and remarkable robustness. The main difficulties lie in completely and efficiently extracting high-level 3D point cloud features, specifically in large-scale urban road environments. The novel architecture of the proposed neural network can directly consume unstructured 3D points and implement a point-wise semantic label assignment network to learn fine-grained layers of feature representations and reduce unnecessary convolutional computations. Compared to existing point cloud segmentation methods based on traditional CNNs, the proposed method is less sensitive to data distribution and computational powers. The experimental results acquired by using different point cloud scenarios indicate that the MS-PCNN model can achieve state-of-the-art semantic segmentation performance in feature representativeness, segmentation accuracy, and technical robustness.

(2). To propose two capsule-based neural network architectures for road marking extraction and classification using MLS point clouds. Road markings captured by MLS systems are usually incomplete and worn, making it challenging to extract and classify accurately. To address this, a capsule-based deep learning framework is proposed for road marking extraction and classification from massive and unstructured MLS point clouds. The innovation of this study is to demonstrate the practical application of combing capsule networks with hierarchical feature encodings of georeferenced feature images for updating road information and supporting HD maps. The experimental results have demonstrated that capsule-based networks effectively extract inherent features from massive MLS point clouds and achieve superior performance in road marking extraction and classification tasks.

(3). To introduce a novel deep learning-based framework, called BoundaryNet, that uses MLS point clouds and satellite imagery to complete road boundaries and calculate road geometries. Still, the varieties and uncertainties of missing parts in urban road boundaries complicate whether these gaps should be filled or not. The innovation is to demonstrate the practical application of deep learning models for road boundary completion from multi-source data. By testing satellite imagery and MLS point cloud datasets with varying densities and road conditions in urban environments, the experimental results indicate that the BoundaryNet model can solve road boundary completion and road geometry estimation.

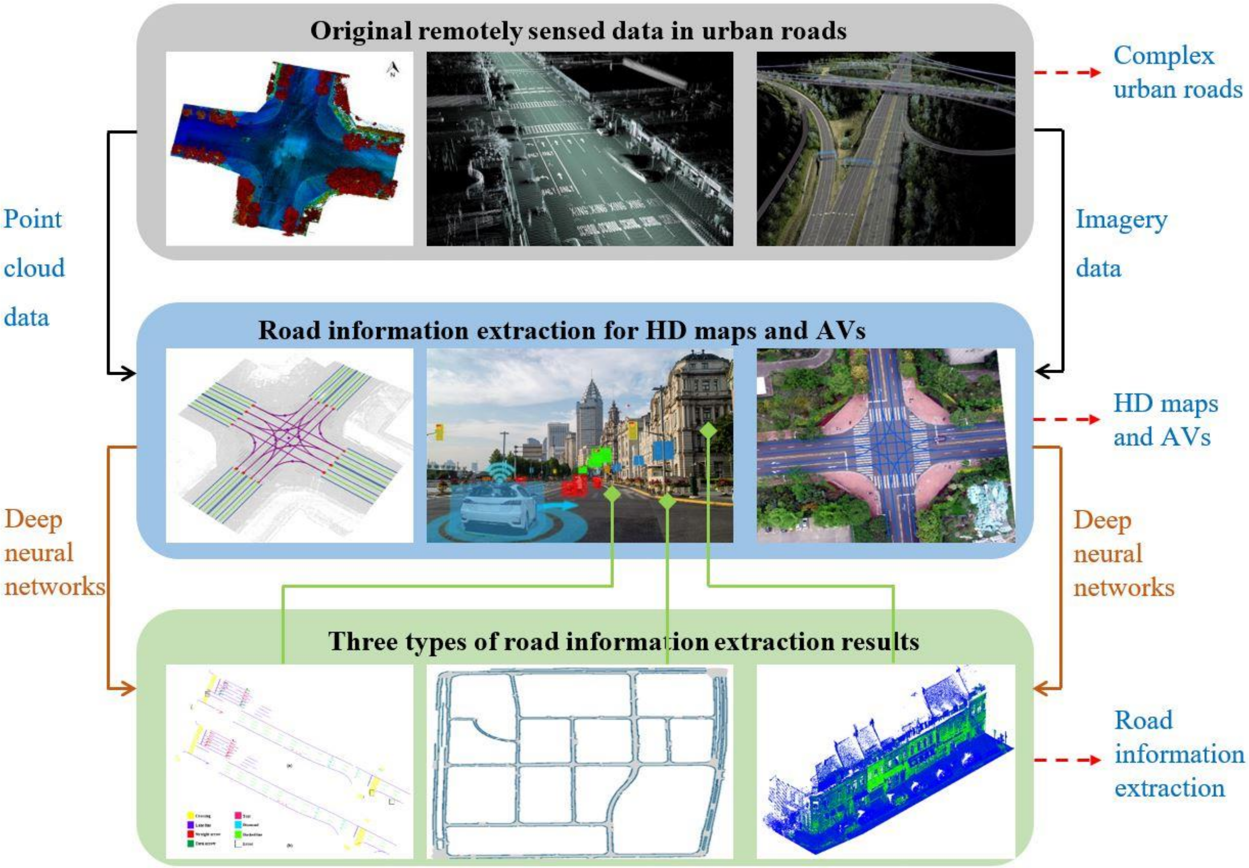

Original remotely sensed data in urban roads
Complex urban roads
Point cloud data
Imagery data
Road information extraction for HD maps and AVs
HD maps and AVs
Deep neural networks
Deep neural networks
Three types of road information extraction results
Road information extraction

Figure 1. 1 Logical-flow framework of this thesis.

1.3 Structure of This Thesis

This doctoral thesis aims to accurately and efficiently extract road information from MLS point clouds by employing deep neural networks. Figure 1.1 shows the logical-flow framework of this thesis. Accordingly, the main structure is provided in the following aspects:

Chapter 2 presents a fundamental literature review of the existing studies on the object semantic segmentation, road marking extraction and classification, and road boundary completion by using MLS point clouds.

Chapter 3 describes an end-to-end deep learning model, to be incorporated with Conditional Random Field (CRF) for point-wise semantic segmentation in multiple scales. By developing a revised point-based 3D convolution, the proposed models can directly consume 3D point clouds without data conversion and transformation.

Chapter 4 presents two capsule-based deep learning networks from massive and unstructured MLS point clouds for road marking extraction and classification, which provide a promising solution for HD map generation and autonomous driving.

Chapter 5 introduces a novel deep learning framework to recover and complete road boundaries using MLS point clouds and satellite imagery, which effectively solves completeness reduction and curvature loss when processing massive MLS point clouds with many missing parts.

Finally, Chapter 6 concludes this thesis and indicates future research directions.

Chapter 2

Mobile LiDAR Point Clouds for Road Information Extraction: An Overview

The MLS systems have indicated a superior strength in providing highly accurate and dense 3D point clouds and attracted considerable attention in the interdisciplinary field at the interface between 3D vision and geodata intelligence, specifically for accurate and efficient road information extraction (Ma et al., 2018). Accordingly, to promote the development of HD maps and autonomous driving, there is an increasingly large number of studies and applications that have been designed to extract road information by using mobile LiDAR point clouds. Although various threshold-based and rule-based methods have delivered promising solutions for road information extraction, massive point clouds with high redundancy, point density and intensity variations, and irregular road structures still pose significant challenges to effectively and automatically manipulate MLS point clouds. Moreover, many deep learning-based methods have been developed to strengthen the feature descriptiveness and representation and learn inherent features by taking advantage of the increased performance of computational resources (Li et al., 2020). However, it is still challenging to effectively and automatically manipulate MLS point clouds with unstructured 3D points, various point densities, outliers, and occlusions, which are inevitable in complex urban environments. Therefore, the accurate understanding and technical expression of 3D road information using deep learning methods have become an urgent demand in extensive intelligent transportation-related applications.

2.1 High-definition Maps

Compared to conventional 2D navigation roadmaps, 3D HD maps provide highly precise and realistic representations of urban road networks with decimetre-level localization accuracy, which could record and update traffic information in real-time, including road hazards, traffic congestion, road construction, and driving speed limitations (Máttyus et al., 2016). Such HD maps are integrated and preloaded on autonomous vehicles by providing an extended monitoring range, which allows AVs to deal with challenging road conditions (e.g., intersections and roundabouts) far beyond the scanning ranges of onboard sensors more rapidly, accurately, and effectively (Bétaille and Toledo-Moreo, 2010).

As shown in Figure 2.1, an HD map is a typical multi-layer structure that contains the base map layer, geometric layer, semantic layer, map priors layer, and dynamic perception layer (Seif and Hu, 2016). As the most significant element of HD maps and the bottom-most layer, the base map layer contains rich geometrical and semantic road information about the physical and static parts of the urban roads, which are well-organized in considerable details to support precise localization and navigation services.

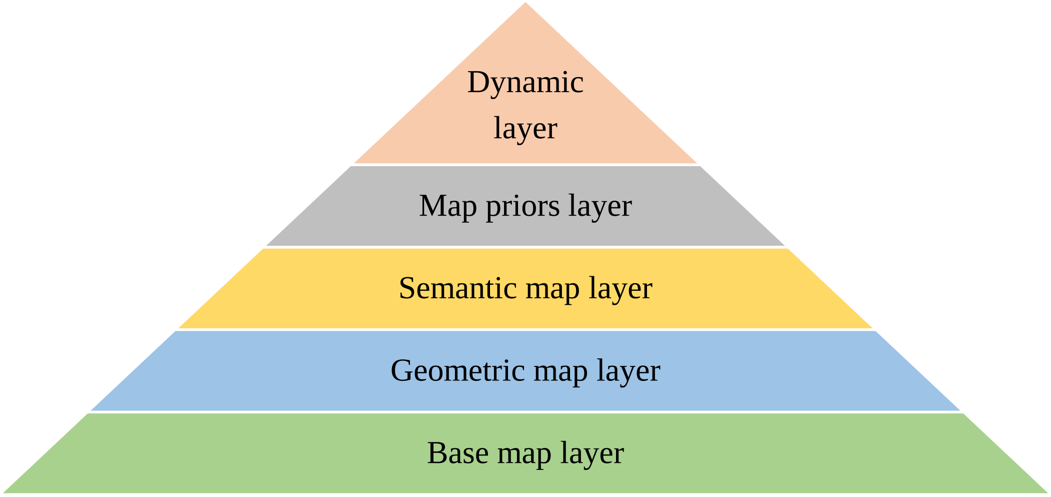


Figure 2. 1 Multi-layer structure of HD maps.

Accordingly, many studies focus on extracting sub-lane level road information and highly detailed road inventories from survey-grade MLS systems with mm-level absolute measurement accuracy, including traffic signs, pole lights, roadside trees, lanes, boundaries, curbs, and all other essential road assets, contributing to the highly precise base map layer assembled in live HD maps (Chu et al., 2018). These road objects comprise essential metadata associated with them, including road widths and turn restrictions for road users. Thus, it is increasingly necessary to propose effective and robust segmentation and classification methods to identify and classify 3D points for different road objects, defined as the geometric and semantic road parts of HD base maps.

2.2 Road Object Segmentation

3D point-wise segmentation is to classify each point in the entire point clouds into several homogeneous classes, and semantic labels will be assigned to the points belonging to the same objects or regions (Nguyen and Le, 2018). For the past several years, many methods have been developed for 3D object segmentation (Ma et al., 2018; Che et al., 2019). This section provides an in-depth review and investigation from the perspectives of 3D point clouds. Generally, the commonly employed 3D point cloud segmentation methods are classified into two groups: hand-designed feature related algorithms and deep learning related algorithms.

2.2.1 Hand-designed Feature Related Studies

Hand-designed feature descriptors, including both global feature descriptors (e.g., shape distribution descriptors) and local feature descriptors (e.g., the spin image feature descriptor), are created to derive inherent features from 3D point clouds, such features are afterward input into off-the-shelf classifiers (e.g., random forests) (Dong et al., 2018). Global feature descriptors are commonly obtained from the geometrical information of entire 3D point clouds. A 3D statistical moment descriptor was developed for the coarse representation of shapes of 3D objects (Paquet et al., 2000). Furthermore, a shape distribution descriptor was proposed to measure geometrical information of 3D objects (Osada et al., 2002). The essential idea is to convert arbitrary 3D object models into parameterized functions that can be directly compared with others. This shape distribution descriptor can effectively eliminate shape segmentation problems to the comparison of probability distributions, which is more robust and straightforward than other shape segmentation methods that need data registration, model fitting, and feature matching. Yet the performance of these global feature descriptors is dramatically impacted by the selection of patch sizes and patch locations. These feature descriptors are highly vulnerable to occlusions, distortions, and background interferences (Luo et al., 2019a). Moreover, due to the complexity of 3D objects, especially for large-scale MLS point clouds, the computational cost will exponentially increase for the extraction of global feature descriptors.

Compared to global feature descriptors, local feature descriptors generally calculate geometrical information and statistical distributions of key points in the local neighbourhoods to construct feature description vectors (Shen et al., 2018). As one of the most representative local feature descriptors, Spin Image feature descriptor (Johnson and Hebert, 1999) has been regarded as the benchmark for the performance evaluation of the other local feature descriptors. However, the feature representation ability of Spin Image is relatively poor (Ma et al., 2018). To enhance the feature descriptiveness, a 3D Shape Context feature descriptor (Frome et al., 2004) was proposed through reconstructing 2D shape context methods on 3D point clouds. Besides, the spatial transformation based feature descriptors, including Heat Kernel Signature (HKS) (Sun et al., 2009) and 3D Speeded Up Robust Feature (SURF) (Knopp et al., 2010), first transformed the spatial domain to other domains (e.g., spectrum domain), then used the specific information in the transformed domains to describe the key points within local neighbourhoods.

Accordingly, Rusu et al. (2009) developed the Fast Point Feature Histograms (FPFH) descriptor by taking the angle differences from a key seed to its neighbours into consideration. Meanwhile, Salti et al. (2014) proposed a histogram-based descriptor, called Signature of Histograms of OrienTations (SHOT), to extract local surface features. Guo et al. (2013) presented a Rotational Projection Statistics (RoPS) method, which has been included in the open-access Point Cloud Library (PCL). Rather than learning global and local features or constructing grid-based data formats, Wang and Jia (2019) introduced a Frustum ConvNet (F-ConvNet) for 3D object segmentation on outdoor KITTI datasets. Firstly, F-ConvNet generated a collection of frustums to assemble points in local regions. Then, point-wise features represented by frustum-level feature vectors were learned via a fully convolutional network within each frustum. Most significantly, F-ConvNet expects no prior knowledge of the data scenarios and is therefore dataset-agnostic.

Nevertheless, for the above methods, the unavoidable task of unstructured point clouds triangulation could lead to considerable computational complexity and original information loss. Other local feature descriptor generation methods based on the geometric information histograms and orientation gradient histograms depend on the first and second derivatives of the point cloud mesh surfaces, which are prone to noise interferences. Moreover, the majority of local feature descriptors require to first detect and extract key points and then construct the local reference frames (LRFs). Therefore, the robustness of an LRF has a great impact on the performance of the generated local feature descriptors. In addition, the larger size of local neighbourhoods, the more information the local feature descriptors describe, and the more sensitive to occlusions and background interferences. Moreover, such methods could capture few geometrical details (e.g., shape and pose information) of 3D objects. Hence, the representativeness of developed feature descriptors is yet far from satisfaction.

2.2.2 Deep Learning Related Studies

Compared to hand-designed feature descriptors, deep learning related algorithms follow end-to-end pipelines, where the multilayer architectures can learn inherent feature representations of high-dimensional data with multiple levels of abstraction (Lecun et al., 2015; Schmidhuber, 2015). Various DL-based methods have remarkably improved the state-of-the-art in many domains, including image recognition, machine translation, and environmental perception (Karpathy and Li, 2015). However, the irregular format and uneven distribution of 3D point clouds make direct applications of traditional CNNs challenging. Thus, the fundamental problem of DL-based algorithms is to address feature representations of 3D point clouds. Several end-to-end DL-based methods have been investigated to deal with the dilemmas of irregular data formats and uneven distributions. They are usually categorized into three groups based on the following data processing methods: voxelization-based, multiview-based, and 3D point-based methods (Ma et al., 2018).

(1) Voxelization-based methods: Volumetric methods can transfer 3D point clouds with irregular format into structured voxel data, on which CNN-related neural networks are thus commonly performed. To overcome the over-segmentation and under-segmentation issues normally occurred in complex urban road environments, Luo et al. (2019b) introduced a probability occupancy grid-based method for real-time ground segmentation tasks by employing a single laser scanner. Maturana and Scherer (2015) proposed a VoxNet architecture by integrating volumetric occupancy grid representation with a supervised CNN framework for 3D object recognition and autonomous robot operation. Meanwhile, Wu et al. (2015) developed 3D ShapeNets to describe 3D geometric shapes as probability distributions of binary variables on 3D volumetric grids, then applied a convolutional deep belief network (DBN). However, such methods lead to sparse volumes and need lots of memory space and computational powers with an increasing voxel size. Accordingly, space partition methods including Octree-based methods (Riegler et al., 2017; Tatarchenko et al., 2017) and KD-tree-based methods (Klokov and Lempitsky, 2017; Zeng and Gevers, 2018; Wang and Lu, 2019) were created to tackle voxel size and memory explosion problems. Nevertheless, the above methods solely depend on the partition of a bounding voxel rather than the locally geometrical structures. That is, if the point density is relatively low, there will be not enough points located in the sparsely sampling neighbourhoods for volumetric convolutional operation. It normally leads to an excessive requirement of memory footprints and high computation cost.

(2) Multiview-based methods: To fully take advantages of well-developed DL-based models in image processing and computer vision fields, such as AlexNet (Krizhevsky et al., 2012) and Mask R-CNN (He et al., 2017), many studies converted 3D point clouds into 2D images. The multiview CNN methods were proposed by projecting 3D point clouds into a set of 2D images derived from multiple views. A basic CNN architecture was then employed to train these rendered images and learn representative features (Su et al., 2015). In order to support autonomous driving, Chen et al. (2017a) developed Multi-View 3D (MV3D) networks for onboard sensor fusion and 3D object detection based on the mechanism of multiple views. Bai et al. (2016) proposed a 3D shape matching and retrieval framework by using projective images of 3D objects. Wen et al. (2019b) first transformed mobile LiDAR point clouds into 2D georeferenced intensity images with 4 cm^2 resolution. An autoencoder-based U-Net was afterward proposed for road marking segmentation.

Additionally, Qi et al. (2016) designed an automated pipeline by combining both 3D voxelization and multiview CNNs for 3D object classification and segmentation. Instead of constructing proposals from RGB-D images or converting point clouds into multiple views or volumetric data blocks, Shi et al. (2019) proposed the PointRCNN model that directly generates 3D object proposals from raw point clouds using a bottom-up strategy. Then, the resampled points in each proposal were transformed into canonical coordinates to capture more local spatial features. Although these 3D-2D dimensional transformation methods can achieve dominating performances, they introduce the resulting data with redundant volumes and ignore the rich 3D geometric information and spatial correlation of points. Still, it is challenging to ascertain both the number and direction of views so that they can cover the whole 3D scenes for self-occlusion prevention.

(3) 3D point-based methods: Compared to volumetric methods and multiview-based methods, 3D point-based methods could directly consume 3D points without data format transformation. Considering the permutation invariance and transformation invariance of point clouds, a CNN-based model, called PointNet (Qi et al., 2017a), was proposed to learn inherent features for classification and segmentation tasks. However, PointNet cannot capture local features of point clouds, which decreases its strength to identify fine-grained patterns and generalizability to large-scale point clouds. Subsequently, an improved version, called PointNet++ (Qi et al., 2017b), was developed to learn more local features than the PointNet by calculating the metric space distances. PointNet++ used the farthest point sampling (FPS) and multi-scale grouping algorithms to leverage local features from coarse layers to fine layers at multiple scales for robustness improvement. In general, both PointNet and PointNet++ are pioneers in DL-based models that directly use 3D point clouds for classification and segmentation in complex scenes.

The fundamental structure developed in both PointNet and PointNet++ for feature aggregation from various input points is max-pooling operation. Nevertheless, a max-pooling layer uniquely remains the largest activation on different features in local neighbourhoods or global regions, which leads to inevitable information loss for segmentation tasks. Furthermore, the lack of deconvolution operation also limits their performances.

To solve these problems, many PointNet-derived deep learning models apply PointNet recursively and optimize their performances to deliver state-of-the-art. Li et al. (2018) developed a PointCNN framework to use a hierarchical convolution structure and an X-Conv operator that aggregate input points into fewer points with richer features. However, PointCNN is not capable of achieving permutation invariance, which is significant for point cloud segmentation. Jiang et al. (2018) proposed the PointSIFT model applying a scale-invariant feature transform (SIFT) descriptor to capture the shape representation of input points. Additionally, dynamic graph CNN (DGCNN) implemented a framework that is able to dynamically update the graph of point clouds (Wang et al., 2019b).

Moreover, Yi et al. (2019) introduced a Generative Shape Proposal Network (GSPN) for 3D object segmentation by employing an analysis-by-synthesis approach and reconstructing shapes as object proposals from noisy observation, which achieves state-of-the-art performance on KITTI LiDAR datasets. Other methods, such as SpiderCNN (Xu et al., 2018), also have demonstrated their superior performance in point cloud object detection and semantic segmentation tasks. However, there are very few applications that apply CNN-based models for segmentation using MLS point clouds, especially in large-scale urban road environments due to high computational complexity and memory occupation. Besides, it is challenging to directly apply traditional CNNs on point clouds regarding to their irregular formats. Additionally, such CNN-based methods always utilize fix-sized filters (e.g., 1 × 1 and 5 × 5) to apply convolution on unorganized point clouds, resulting in remarkably redundant convolutional operations and extra memory overhead.

To summarize, most of the existing hand-designed feature descriptors or feature-related methods that only concentrate on either global or local statistical information, which leads to a performance reduction in representativeness and descriptiveness. Also, traditional CNN convolution applied to 3D point clouds could lead to high computational consumption and edge information loss. Thus, it is necessary to propose an end-to-end deep learning framework that can directly consume 3D point clouds and implement a point-wise semantic label assignment network to learn fine-grained layers of feature representations and reduce unnecessary convolutional computations.

2.3 Road Marking Extraction and Classification

2.3.1 Road Marking Extraction

Road markings are decorated on asphalt concrete pavements with highly light-reflective coatings, the intensities backscattered from road markings are considerably higher than surrounding pavements (Zai et al., 2017). Accordingly, threshold-related approaches have been widely applied to achieve road marking extraction (Soilán et al., 2017; Soilán et a., 2019). To overcome the unevenly distributed intensities and point densities, a multi-threshold approach was developed by first segmenting raw point clouds into data blocks with trajectory support. Next, each block was divided into different profiles with a certain width. Finally, road markings were extracted based on the peak values of intensity in each profile, followed by the spatial density filtering (SDF) algorithm for noise removal (Ma et al., 2019a). Combined with the multi-threshold method, Ye et al. (2019) employed geometric feature filtering to segment lane markings.

Furthermore, by converting 3D point clouds into 2D georeferenced intensity images, a multiscale tensor voting (MSTV) algorithm was proposed by Guan et al. (2015) for discrete pixel elimination and road marking preservation. A weighted neighbouring difference histogram (WNDH) algorithm was first performed to compute the intensity histogram of raw point clouds and determine adaptive thresholds. Subsequently, the MSTV and upward region-growing approaches were applied to ascertain candidate road marking pixels, accompanied by a morphological nearest algorithm for road marking extraction. However, it is still very challenging for such methods to effectively extract road markings from unorganized and high-density point clouds, especially with distinctive concavo-convex features (Yu et al., 2014).

Deep learning techniques have been widely applied in the domains of object segmentation and object classification. He et al. (2016) proposed a lane marking extraction method based on the CNNs from MLS point clouds. A CNN framework designed for learning hierarchical features from upsampling-downsampling modules was first introduced to detect lane-shaped markings effectively. Then, both the length and spatial information related filters were utilized to optimize the extracted road markings. Moreover, Wen et al. (2019b) developed an improved U-Net encoder-decoder framework by learning inherent features of road markings embedded in different data patches, which achieved promising flexibility and performance on point clouds with low-intensity contrast ratios. Nevertheless, these methods mainly concentrate on regular-shaped road markings (e.g., dashed lines and zebra crossings), it remains a challenge to deliver satisfactory results for complicated road markings (e.g., texts). Although it dramatically reduces the computational complexity by converting 3D point clouds into 2D rasterized images, these neural networks cannot capture pose or spatial information that is quite significant for road marking extraction in fluctuant terrain environments (Wen et al., 2019b).

2.3.2 Road Marking Classification

Following the extraction process, many classification methods were developed to classify road markings into various categories for specific applications (Guan et al., 2016; Che et al., 2019). Yu et al., (2014) implemented a Euclidean distance-based clustering approach, followed by a voxel-based normalized segmentation algorithm for clustering unorganized road marking point clouds into large-size and small-size clusters. Afterward, large-size road markings were classified with the assistance of trajectory data and curb-lines. Then, a jointly trained Deep Boltzmann Machine (DBM) neural network, followed by a multi-layer classifier, was developed to recognize and categorize small-size markings effectively. Additionally, based on the geometric parameters (e.g., perimeter, area, and calculated width), Cheng et al. (2017) classified the extracted road markings by constructing a manually defined decision tree. However, it is challenging for this rule-based method to effectively classify complex road markings, such as words and arrows. Due to discrete noise, faded markings, and varied road environments, it is also difficult for these methods to accurately classify the incomplete road markings.

To achieve the superior road marking classification performance, Soilán et al. (2017) designed a hierarchical classification framework by employing a multi-layer neural network to recognize arrows and pedestrian crossings. Then, the Structural Similarity Index (SSIM) algorithm was carried out to classify different types of arrows. Furthermore, Wen et al. (2019b) introduced a two-stage CNN-based hierarchical classification framework. At first, a multi-scale Euclidean clustering algorithm was implemented to classify large-size road markings (e.g., zebra crossing). Then, the remaining small-size road markings (e.g., texts and diamonds) were successfully classified into different groups by using a four-layer convolution network, followed by an optimized conditional generative adversarial network (c-GAN) to enhance the completeness of the extracted road markings. Although their experimental results indicated a highly promising solution in road marking classification, it is still a challenge to eliminate the influences of small incompletions and deliver an end-to-end deep learning framework. Therefore, it is increasingly necessary to develop a deep learning framework capable of encoding more salient features embedded in intensity values and the pose and spatial information of the road markings for extraction and classification purposes.

2.4 Road Boundary Extraction and Completion

2.4.1 Road Boundary Extraction

Generally, the commonly applied boundary extraction methods are categorized into three groups: remotely sensed image-driven methods (Chen et al., 2019b, Wang et al., 2015b, Zhang et al., 2019), 3D point-driven methods (Kang et al., 2012, Yang et al., 2013, Cabo et al., 2016, Lin et al., 2018), and multi-source data-driven methods (Homayounfar et al., 2018, Liang et al., 2019). Different kinds of remotely sensed imagery acquired by various platforms and sensors, such as drone-based imagery, satellite imagery, and synthetic aperture radar (SAR) imagery, provide significant spectral, spatial, and texture information for the robust and effective road feature extraction in large-scale terrains. For instance, Zhang et al. (2019) proposed a Multi-supervised Generative Adversarial Network (MsGAN) framework to extract road boundaries and centerline maps from satellite images, which is jointly trained by the topology and spectral information of road networks. By estimating an approximate prediction of the road edges for guidance, Zang et al. (2016) developed an anisotropic shock filtering framework to extract roads. Additionally, Chu et al. (2019) proposed a graph-driven neural turtle graphics network to detect urban road layouts on the SpaceNet dataset, while nodes in the graph indicate spatial control points of the road networks and edges represent road segments. However, various environmental and topological factors, including ambient lighting conditions, the complexity of road scenarios, and undulating terrains, have remarkable impacts on the performance of road boundary extraction from 2D images.

Meanwhile, 3D point clouds acquired by MLS, ALS, and TLS systems have been an appropriate data source for road boundary extraction. Zai et al. (2017) developed a two-stage method for 3D road boundary extraction from MLS point clouds in complex road scenes. First, super voxels were generated based on smooth seed points and different geometric and spatial attributes. Then, road boundaries were extracted by performing the α-shape algorithm, followed by the graph-cut related energy minimization algorithm. By partitioning roads to a collection of LiDAR point cloud blocks with the assistance of vehicle trajectory, Wang et al. (2015b) proposed a method to construct saliency maps and extract salient points for road boundary detection. Moreover, according to independent patches of the road network, Boyko and Funkhouser (2011) developed a method to generate 3D road maps by using Cardinal splines under continuity constrains and an attractor function for curb detection from both ALS and MLS point clouds. However, it is difficult to accurately and completely extract road boundaries regarding unstructured and noisy point clouds. The information loss is inevitable during the process of voxelization and occlusions.

Furthermore, because of the limitations of employing a single data source, some methods that take multi-source data into consideration are essential to obtain rich road information and complete road boundaries with high accuracy and robustness. Li et al. (2019b) used a transfer-learning-based neural network for road feature encoding and a U-Net framework for road centerline and edge extraction by integrating aerial images with taxi trajectories. Ravi et al. (2019) proposed an intensity-based multi-threshold method to extract lane markings (e.g., road boundaries), and then combined the extracted lane markings from MLS point clouds with RGB camera images for accurate lane width estimation and road boundary measurement. Although such methods have significantly improved the accuracy of road boundary extraction by introducing spectral or texture information, they still cannot address existing gaps in road boundaries.

2.4.2 Road Boundary Completion

To fill these existing gaps and enhance the completeness of road boundaries caused by occlusions and method drawbacks, Xu et al. (2016) employed an energy function for the candidate curb point segmentation from MLS point clouds and further refined these candidate points by conducting a least-cost path model. Likewise, Ma et al. (2019a) performed a B-spline least-square fitting method to generate smooth road boundaries from candidate road curb points. However, it is a challenging task for such methods to deal with compound or spiral curved road sections with changing curvatures (Ma et al., 2019a). Inspired by image inpainting, Wen et al. (2019a) first proposed a CNN-based model to complete gaps and recover road boundaries from 3D point clouds.

Then, a conditional generative adversarial network (cGAN) framework was implemented to deal with the uncertainties of missing parts of road boundaries and refine them accordingly. The existing methods have certain limitations and considerable challenges to provide promising solutions (Zai et al., 2017, Wen et al., 2019a). Accordingly, to tackle the problems of completeness reduction and curvature loss when processing massive MLS point clouds with many missing parts, a deep learning framework should be proposed to identify and restore the missing parts of road boundaries by using multi-source data (e.g., LiDAR points, camera images, and GNSS points) under challenging road scenes more accurately and robustly.

2.5 Chapter Summary

In this chapter, a variety of existing methods proposed for road object segmentation, road marking extraction and classification, and road boundary extraction and completion using MLS point clouds were comprehensively reviewed and discussed. It can be concluded that MLS point clouds are suitable for extracting highly precise road information in complex urban road conditions. Although various threshold-based and rule-based methods have delivered promising solutions for road information extraction, massive point clouds with high redundancy, point density and intensity variations, and irregular road structures still pose significant challenges to effectively and automatically manipulate MLS point clouds.

More recently, based on the more and more publicly available point cloud datasets with labels, deep learning-based methods have demonstrated that they are capable of learning deeper and more distinctive feature representations of different road objects by taking advantage of the powerful computational resources. Accordingly, intelligent point cloud processing and road information extraction by using deep neural networks are investigated in the following aspects: a point-wise 3D convolution operation embedded in a U-shaped downsampling and upsampling framework is proposed for road object semantic segmentation in Chapter 3; two hybrid capsule-based deep neural networks are developed to extract and classify different types of road markings in Chapter 4; followed by CNN-based and GAN-based networks for road boundary recovery in Chapter 5.

Chapter 3

MS-PCNN: Multi-scale Point-wise Convolution for Road Object Segmentation

Although significant improvement has been achieved in fully autonomous driving and semantic HD map domains, most of the existing 3D point cloud segmentation methods cannot provide high representativeness and remarkable robustness. The principally increasing challenges remain in completely and efficiently extracting high-level 3D point cloud features, specifically in large-scale road environments. This chapter provides proposed a novel end-to-end neural network, named MS-PCNN, by combining point-wise CNNs with dynamic edge convolutions in multiple scales for 3D point cloud segmentation. Compared to existing point cloud segmentation methods that are commonly based on traditional convolutional neural networks, the proposed method is less sensitive to data distribution and computational powers. The experimental results acquired by using different point cloud scenarios indicate the MS-PCNN method can achieve state-of-the-art semantic segmentation performance in feature representativeness, segmentation accuracy, and technical robustness.

More specifically, Section 3.1 introduces the research backgrounds. Section 3.2 presents a stepwise algorithm framework in detail. The implementation details of deep neural networks are presented in Section 3.3. The datasets used in this study are presented in Section 3.4. The experimental results and discussions are presented in Section 3.5, followed by the efficiency evaluation in Section 3.6. Section 3.7 concludes this chapter. © [2020] IEEE. Reprinted, with permission, from [Lingfei Ma, Ying Li, Jonathan Li, Weikai Tan, Yongtao Yu, and Michael A. Chapman. Multi-scale Point-wise Convolutional Neural Networks for 3D Object Segmentation from LiDAR Point Clouds in Large-scale Environments. *IEEE Trans. Intell. Transp. Syst.*, doi:10.1109/TITS.2019.2961060].

3.1 Introduction

With the increasing market demands of the Advanced Driver-Assistance Systems (ADAS), Level-5 fully autonomous driving, autonomously operating robotics, smart cities, and semantic high-definition (HD) maps, mobile laser scanning (MLS) or mobile Light Detection and Ranging (LiDAR) systems have attracted extensive attention of many researchers over the past few years

(Bresson et al., 2017). Such MLS systems could effectively collect high-density and precise 3D point clouds in large-scale road environments (Yu et al., 2016). Accordingly, 3D point clouds have been commonly applied in many industrial applications, including 3D object extraction in urban road networks (Ye et al., 2019; Ma et al., 2019a), object registration, object tracking (Luo et al., 2019c), object modeling and 3D reconstruction, object classification (Wen et al., 2019b), and semantic segmentation (Luo et al., 2015). As a significant requirement of 3D digital cities, 3D semantic segmentation aiming to assign the per point semantic label for all input point clouds is crucial in exploiting the informative values of point clouds for the aforementioned applications (Lin et al., 2018). Therefore, in this paper, it specifically concentrates on the foundational and theoretical problems of 3D semantic segmentation using MLS point clouds in large-scale urban environments.

3D point-wise segmentation is to classify each point in the entire point clouds into several homogeneous classes, and semantic labels will be assigned to the points belonging to the same objects or regions. However, it is very challenging to achieve automated and effective point-wise segmentation regarding the high redundancy, uneven point density, and inexplicit structure of MLS point clouds (Wen et al., 2019b). Generally, 3D semantic segmentation is performed by creating hand-designed feature descriptors. The most representative feature descriptors are comprised of global feature descriptors and local feature descriptors. Such global feature descriptors, e.g. 3D statistical moment (Paquet et al., 2000) and spherical harmonics descriptor (Funkhouser et al., 2003), are commonly obtained using the geometrical information of entire MLS point clouds. However, these feature descriptors are very sensitive to occlusions, distortions, and background interferences, resulting in segmentation ambiguities (Wang et al., 2018). In addition, local feature descriptors including Spin Image (Johnson and Hebert, 1999), Signature of Histograms of OrienTations (SHOT) (Salti et al., 2010), Fast Point Feature Histograms descriptor (FPFH) (Rusu et al., 2009), and Fourier power spectrum (FPS) (Masuda, 2009), mainly concentrate on the descriptive information of point clouds in local regions. However, such methods could capture few geometrical information (e.g., shape and pose features) of 3D objects. Hence, the representativeness of developed feature descriptors is yet far from satisfaction.

To strengthen the descriptiveness and feature representation of these existing methods, it is effective to learn features at middle and high levels for inherent and additional information taking advantage of the increased performance of computational resources. One promising solution is to use deep learning (DL) models, e.g. deep convolutional neural networks (CNNs) and generative adversarial networks (GANs), to learn deeper and more distinctive feature representations (Zhu et al., 2017). Accordingly, various methods including voxel-based methods (e.g., VoxNet (Maturana and Scherer, 2015)), multiview-based methods (e.g., MV3D (Chen et al., 2017a)), auto-encoder-based methods (e.g., CAE-ELM (Wang et al., 2016)), graph cut-based methods (e.g., ECCNet (Simonovsky and Komodakis, 2017)), and symmetric function-based methods (e.g., PointNet (Qi et al., 2017a)), have been proposed to use MLPs for 3D data analysis, object recognition, and semantic segmentation. Although these existing CNN-based methods have achieved a significant enhancement in the representativeness and descriptiveness on several publicly available datasets (e.g., ShapeNet-Part and ModelNet40), it is still challenging to effectively and automatically manipulate MLS point clouds with unstructured 3D points, various point densities, outliers, and occlusions, which are inevitable in complex urban environments.

To overcome these challenges, the feasibility of embedding point-wise CNNs with hierarchical feature representations of point clouds is investigated. Yet CNNs were initially developed to deal with 2D images with structured pixel arrays. Such images are organized with regular lattice grids in a specific order, which can be directly fed into CNN-based architectures. It is not feasible to directly perform CNNs on 3d MLS point clouds since they are not in a regular data format or an inherent order. Therefore, to solve this dilemma, a 3D semantic segmentation model aiming to facilitate collaboration between point-wise CNNs and unstructured 3D point clouds is developed. The novel architecture of the proposed neural network is to directly consume unstructured 3D points and implement a point-wise semantic label assignment network to learn fine-grained layers of feature representations and reduce unnecessary convolutional computations. To this end, an end-to-end DL framework comprised of the following four modules is proposed: (1) point-based 3D convolution, (2) U-shaped downsampling-upsampling framework, (3) dynamic graph edge convolution, and (4) conditional random field (CRF)-based postprocessing. This proposed neural network is capable of robustly and efficiently extract global and local features of input point clouds in multiple scales. Furthermore, these proposed and revised models can directly consume 3D point clouds without data conversion and transformation.

The proposed model has been evaluated on publicly accessible large-scale MLS point cloud dataset. Experimental outputs conclusively demonstrate that the proposed method could achieve superior performance in feature representation, computational efficiency, and robustness. The significant contributions of this paper are described as follows: (1) the PointCONV as a multi-scale density-based reweight convolution was revised, which can completely and efficiently approximate the 3D convolution on large-scale unstructured point clouds with an efficient computation fashion; (2) a hierarchical U-shaped downsampling-upsampling framework was designed to implement both PointCONV and PointDeCONV for better point-wise segmentation outcomes; (3) the Edge-Conv descriptor was revised by optimizing both symmetric aggregation function and edge function to achieve dynamically update the graph of edges and learn more representative features between adjacent points in local neighbourhoods; and (4) a CRF algorithm was performed for the label assignment refinement generated by the proposed end-to-end model.

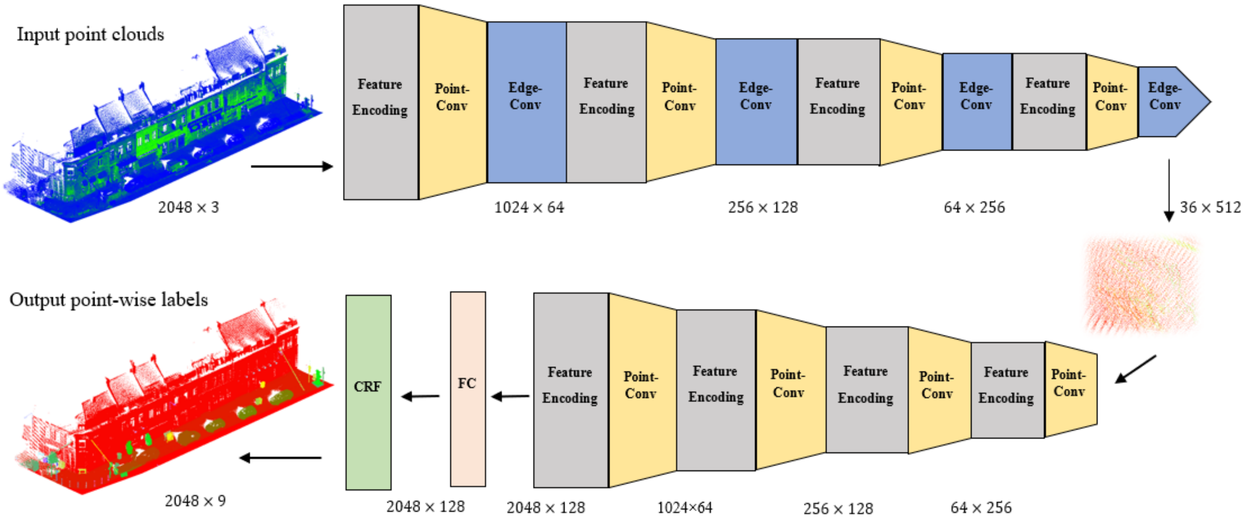


Figure 3. 1 Illustration of the proposed MS-PCNN model architecture.

3.2 Algorithm Description

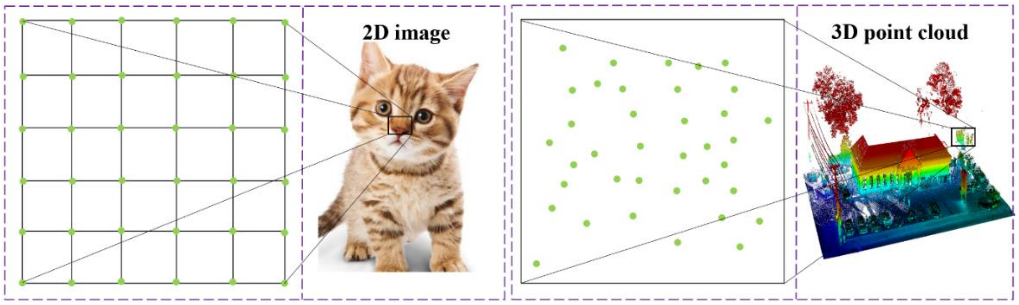
In this section, the theoretical and logical principles of the proposed model are presented for 3D road object segmentation from MLS point clouds. This novel model, named MS-PCNN, mainly contains four modules: convolution on 3D points, multi-scale feature extraction, dynamic edge feature extraction, and conditional random field post-processing. More specifically, Module I is designed to construct a revised convolutional kernel, particularly for 3D point clouds. A Monte Carlo approximation of the 3D continuous convolutional operators is first applied followed by dynamic density scales to re-calculate the optimized weight functions. In Module II, a U-shaped downsampling-upsampling architecture is proposed to leverage both global and local features in multiple scales. Next, in Module III, high-level local edge features in 3D point neighbourhoods are further extracted by using an adaptive graph convolutional neural network based on the K-Nearest Neighbour (KNN) algorithm. Finally, in Module IV, a conditional random field algorithm is developed for postprocess and segmentation result refinement, and 3D road objects are therefore segmented. Figure 3.1 presents the detailed workflow of the proposed MS-PCNN model.

3.2.1 Convolution on 3D Points

Although CNN-based methods have demonstrated the superior performance on recognition, classification and segmentation tasks using regular data formats, such as 2D images or 3D voxelized grids, it is very difficult to provide promising solutions that directly apply convolutions on 3D point clouds. Inspired by Wu et al. (2019), as a revised convolutional operation, MS-PCNN extending conventional 2D image convolutions into 3D point clouds is accordingly proposed. In general, convolutional operations are determined by using:

$$(F * G)(x) = \iint F(\Delta x)G(x + \Delta x)d(\Delta x)\,, \Delta x \in \mathbb{R}^d \tag{3.1}$$

where $F(x)$ and $G(x)$ are two functions, x is a d-dimensional vector, and $\mathbb{R}^d$ denotes a d-dimensional Euclidean space. 2D images represented by grid-structured matrices are normally regarded as discrete functions. In traditional CNNs, various kernel filters (e.g., 1 × 1, 5 × 5, and 7 × 7) are assigned to focus on small-sized local neighbourhoods. Moreover, the relative positions among different pixels are always certain in each local region, as illustrated in Figure 3.2(a). Diverse filters can be effectively employed to calculate the sum of real-valued weights for different locations in the given local neighbourhood.



(a) (b)

Figure 3. 2 Data format comparison between 2D images and 3D point clouds: (a) 2D images. (b) 3D point clouds.

In contrast, point clouds are considered as a collection of discrete 3D points p_i (i = 1, 2, . . . , n) containing xyz coordinate information and related characteristics including color, intensity, and normal. Compared to grid-structured images, point clouds have an irregular format with the unfixed arrangement. Hence, as shown in Figure 3.2(b), the relative positions of point clouds are different within different local regions, resulting in traditional convolutional filters used on regular data formats (e.g., images) cannot be directly utilized on point clouds.

In order to make full use of convolutional operations on 3D point clouds, Wu et al. (2019) proposed a permutation-invariant convolutional filter, called PointCONV. The main idea of PointCONV is to define the 3D convolutions for continuous functions by the following equation:

$$3DConv(H,J)_{xyz} = \iiint H(\varphi_x, \varphi_y, \varphi_z) \cdot J(x+\varphi_x, y+\varphi_y, z+\varphi_z) d\varphi_x \varphi_y \varphi_z \, , (\varphi_x \varphi_y \varphi_z) \in E \quad (3.2)$$

where $H(x)$ and $J(x)$ are two functions, $J(x+\varphi_x, y+\varphi_y, z+\varphi_z)$ represents the feature of a point p_i (i = 1, 2, . . . , n) in the local neighbourhood E, where (x, y, z) is the center position of this local region. Specifically, point clouds are interpreted as non-uniform samples in the continuous 3D space. Therefore, PointCONV is defined as follows:

$$(F, H, J)_{xyz} = \sum_{(\varphi_x \varphi_y \varphi_z) \in E} F\left(\varphi_x, \varphi_y, \varphi_z\right) H(\varphi_x, \varphi_y, \varphi_z) \times J(x+\varphi_x, y+\varphi_y, z+\varphi_z) \quad (3.3)$$

where $F\left(\varphi_x, \varphi_y, \varphi_z\right)$ indicates the inverse density given the point $\left(\varphi_x, \varphi_y, \varphi_z\right)$. $F\left(\varphi_x, \varphi_y, \varphi_z\right)$ is significant since the downsampled point clouds are non-uniformly distributed. However, the point densities in different local neighbourhoods are various across the entire point clouds. The key idea is to employ multi-layer perceptrons (MLPs) for the weight function approximation based on the 3D positions $\left(\varphi_x, \varphi_y, \varphi_z\right)$ and the inverse density values $F\left(\varphi_x, \varphi_y, \varphi_z\right)$ using a density estimation algorithm. However, Wu et al. (2019) considered the approximation of the density scale in a fixed threshold rather than multi-scale or dynamic scales, which leads to approximations of the 3D convolutional operator far from satisfactory.

Different from Wu et al. (2019), a multi-scale kernelized point density calculation algorithm is proposed in this study followed by a non-linear transformation algorithm, which is implemented during feature extraction stages. Different colors in Figure 3.3 represent different point densities. The MS-PCNN network is designed to capture multi-scale patterns by grouping 3D points in multiple scales followed by according MLPs to extract inherent features within each scale. Then, features learned from various scales are concatenated together for the multi-scale feature encoding purpose. The raw points are randomly dropped out with a randomized probability for each point. According to prior knowledge, the randomized dropout rate θ was ascertained in the range of [0, ρ], where ρ = 0.9 to avoid the sampling deficiency. To achieve invariant permutation of 3D points, the weights learned from different MLPs in the revised PointCONV are shared in the whole point clouds. According to the proposed multi-scale kernel density estimation (MKDE) and non-linear transformation algorithms, the inverse density scales $F(\varphi_x, \varphi_y, \varphi_z)$ can be adaptively calculated with multi-scale point density estimation in local regions.

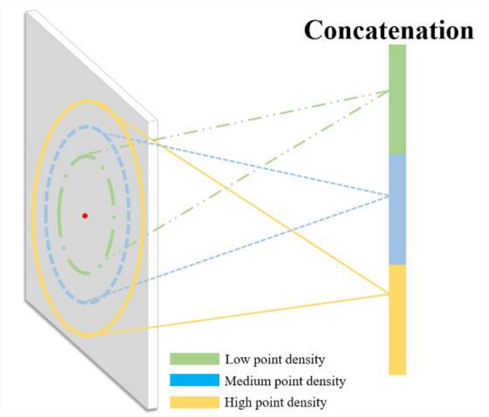


Figure 3. 3 Illustration of the multi-scale kernelized point density estimation.

Figure 3.4 indicates the revised PointCONV framework within a local neighbourhood. As can be seen, the white rectangles represent the features by concatenating interpolated features with features learned from MLPs with the same resolution using across-level skip connections. Blue color bars indicate feature extraction results by using MLPs, while light green bars denote the downsampling and grouping modules that are similar to the ones employed in PointNet++ (Qi et al., 2017b). More specifically, the iterative farthest point sampling was conducted to subsample the raw point clouds by calculating the Euclidean distances from 3D points to the given centroids, which can generate receptive fields in a data-dependent fashion. Assuming that C_i and C_o be the number of channels about the input features and output features, respectively. (k, C_i, C_o) is regarded as the index of K-th neighbour, C_i-th channel of input features and C_o-th channel of output features. The input p_i (i = 1, 2, . . . , n) provide 3D coordinates p_i = (x_i , y_i , z_i) where $p_i \in \mathbb{R}^{3\times K}$, which is calculated by subtracting the centroid coordinate and the input feature $c \in \mathbb{R}^{C_i\times K}$ of the local neighbourhood. 1 × 1 convolutional kernel size is ascertained to perform MLPs. The outputs of the weight functions are $W \in \mathbb{R}^{(C_i\times C_o)\times K}$. Accordingly, $W(k, C_i) \in \mathbb{R}^{C_o}$ denotes a weight vector, and the density scale is $F \in \mathbb{R}^K$. In order to capture more high-level local features at multiple scales in each local region, the multiple thresholds of K are selected based on the diverse distributions of point clouds, and the average value of MKDE is then estimated. According to prior knowledge and the accessible computation capability, K is predefined as 128, 64, 32, and 16, respectively. After convolutional operations, the input features F_i captured in each local region with multi-scale K points are fed into the following equation to obtain the output features $F_o \in \mathbb{R}^K$:

$$F_o = \sum_{k=1}^{K} \sum_{c_i} F(k)W(k,c_i)F_i(k,c_i) \tag{3.4}$$

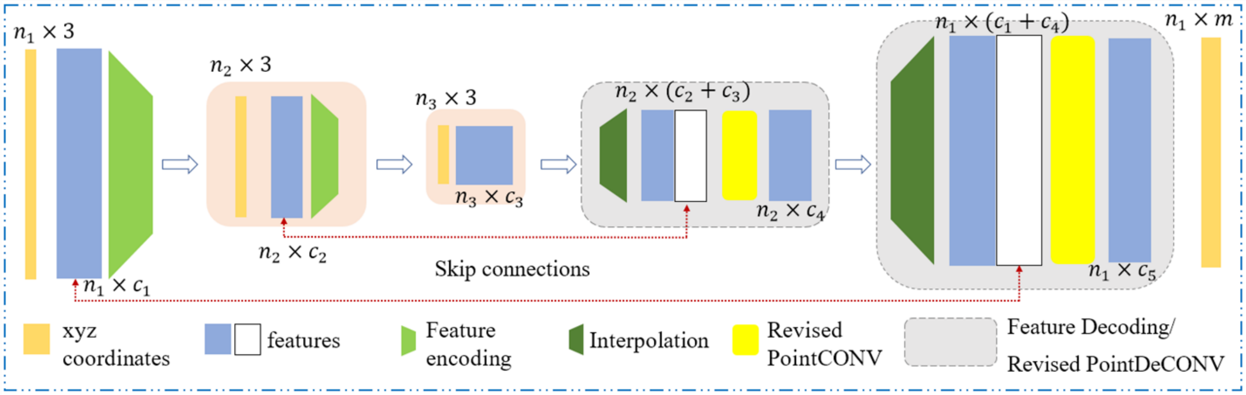


Figure 3. 4 Revised PointCONV framework with feature encoding and propagation.

However, such revised PointCONV operations are time-consuming and huge memory-overhead, especially for the weight approximation. For a certain point cloud, each local neighbourhood is assigned to the equivalent weight functions that are encoded from MLPs. Nevertheless, the weights calculated by different weight functions from various point clouds are different. Accordingly, the sizes of the weight filters can be determined as follows:

$$S_w = B \times N \times K \times C_{in} \times C_{out} \tag{3.5}$$

where S_w is the size of weight filters computed by MLPs. B is the mini-batch size, N represents the number of points within each point cloud, K indicates the number of points within each local neighbourhood, C_{in} is the number of input channels, and C_{out} denotes the number of output channels. For instance, if B = 64, N = 1024, K = 64, C_{in} = C_{out} = 128, respectively, the memory size for the generated weight filters are over 16 GB for each layer, which results in huge memory consumption in the training phase. Therefore, to tackle this problem, the PointCONV implementation is further refined by optimizing matrix multiplication and 2D convolution operations. The revised PointCONV is equivalent to the following equation:

$$F_{out} = CONV_{3\times3}(H, (F \cdot F_{in})^T \otimes M) \tag{3.6}$$

where $M \in \mathbb{R}^{K\times C_{mid}}$ denotes the inputs fed into the last layer of MLP to calculate the weight function, $H \in \mathbb{R}^{(C_{in}\times C_{out})\times C_{hid}}$ is the weights in the last layer of MLP, F indicates the density scale, and $CONV_{3\times3}$ is 3 × 3 convolutional operation. Since the last layers of MLPs are generally linear layers, $\tilde{F} = F \cdot F_{in}$ is therefore rewritten within each local neighbourhood. Accordingly, let the weight function $W = CONV_{3\times3}(H, M) \in \mathbb{R}^{(C_{in}\times C_{out})\times K}$, k is the index of points in local regions, and c_{in}, c_{hid}, c_{out} are the indices of the input, hidden and output layer, respectively. Therefore, the revised PointCONV can be expressed as follows:

$$F_{out} = \sum_{k=0}^{K-1}\sum_{c_{in}=0}^{C_{in}-1}(W(k, c_{in})\widetilde{F_{in}}(k, c_{in})) \tag{3.7}$$

$$W(k, c_{in}) = \sum_{c_{hid}=0}^{C_{hid}-1}(M(k, c_{hid})H(c_{hid}, c_{in})) \tag{3.8}$$

According to both Eqs. (3.7) and (3.8), the revised PointCONV is thus determined by:

$$F_{out} = \sum_{k=0}^{K-1}\sum_{c_{in}=0}^{C_{in}-1} CONV_{3\times3}(H, \widetilde{F_{in}^{T}}M) \tag{3.9}$$

Consequently, the previous PointCONV is equivalently converted into a 2D 3 × 3 convolution and a matrix multiplication. In this revised model, the matrix multiplication is refined by dividing the weight filters into two parts: the convolutional kernel H and the intermediate output M. In addition, instead of using the 1 × 1 convolution, the 3 × 3 convolution is employed to deliver promising outputs and effectively reduce computational costs. Assuming that C_{hid} = 64, the memory usage is about 0.251 GB for each layer, which is only 1/64 of the original PointCONV with the same parameters as shown in Figure 3.4.

Therefore, this revised PointCONV operation can effectively construct a network and approximate the continuous weights for convolutions on point clouds. Compared to the traditional convolutions, the revised PointCONV-based convolution that only considers the relative coordinates as inputs could output multi-scale densities and weights across the whole point clouds, which considerably decreases the computational cost caused by traditional discretized and fix-sized convolutions.

3.2.2 U-shaped Downsampling-Upsampling Architecture

After implementing PointCONV operations, the original input point clouds have been subsampled into various resolutions. However, for object segmentation task especially as semantic labeling, the point wise segmentation for the entire point clouds is needed. To acquire high-level features for the whole point clouds in both global and local scales, a hierarchical framework that could propagate features from subsampled point clouds to relatively dense ones is required. Therefore, a U-shaped downsampling-upsampling architecture is proposed by taking PointCONV operations into consideration. According to the revised PointCONV mentioned in Section 3.2.1, more high-level features are captured by regarding a revised PointDeCONV layer as deconvolutional operations.

As shown in Figure 3.4, PointDeCONV implementation mainly contains two processes: interpolation and revised PointCONV. First, the interpolation operation is conducted to assemble different level features from previous layers. According to the three nearest points, the interpolation is carried out to linearly interpolate features. Subsequently, such interpolated features are concatenated with features learned from MLPs with the same resolution using across-level skip links. Finally, the revised PointCONV is thus employed on the concatenated features to catch the final deconvolution outputs. Accordingly, this recursive process will not terminate until the features learned from all point clouds have been propagated back to the initial resolution.

3.2.3 Dynamic Graph Edge Convolution

Although the proposed MS-PCNN hierarchical framework embedded with revised PointCONV and PointDeCONV operations could obtain features for all input point clouds in multiple scales, edge features between a point and its adjacent neighbours have not been taken into consideration. To address this drawback, a PointCONV-based dynamic graph edge convolution operator is proposed to capture local geometrical edge structures based on the EdgeConv descriptor (Wang et al., 2019b). As shown in Figure 3.1, a local neighbourhood graph is constructed followed by PointCONV-based convolutional operations on the connected edges. Compared to conventional graph CNNs, the graphs introduced in this study are dynamically updated rather than fixed after feature extraction layer. Specifically, the K-nearest neighbours of a point dynamically change between two adjacent layers of the model and are accordingly calculated the sequence of embeddings. Figure 3.5 illustrates the principle of revised EdgeConv operation compared with revised PointConv operation. As can be perceived, by adding dynamic graph edge convolutions into the proposed model, not only point-wise geometric information but also edge informative features between a certain point and its neighbours are considered to capture more descriptive features in a local region.

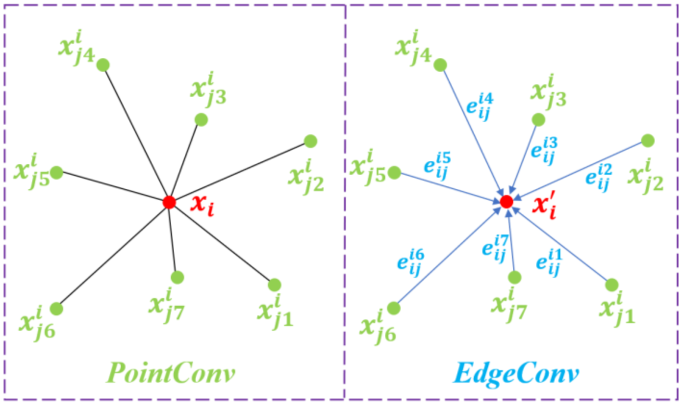


Figure 3. 5 Illustration the principles of the revised PointCONV and EdgeConv operations.

Assuming that a directed graph $G = (V, E)$ denoting the local structure of each point cloud, $V = (1, 2, \dots, n)$ and $E \subseteq V \times V$ are the vertices and edges, respectively. To simplify this problem, a graph G is built as the KNN-graph in D-dimensional space (generally $D = 3$ representing xyz coordinates of point clouds) and edge features are defined as $e_{ij} = g_\phi(x_i, x_j)$, where $g_\phi \in \mathbb{R}^D \times \mathbb{R}^D$ denote parameterized nonlinear functions with a collection of parameters ϕ. After that, the EdgeConv operation is defined by conducting a channel-wise symmetric aggregation implementation (e.g., sum) on the edge features from each point. Therefore, the output of EdgeConv operation at the i-th vertex is performed as follows:

$$x_i' = \boxdot_{j:(i,j)\in E}\ g_\phi(x_i, x_j) \tag{3.10}$$

where $\boxdot$ represents a symmetric aggregation function. In this study, differently from Wang et al. (2019b), the max operation is applied as the aggregation function rather than the sum to reduce the computational consumption. Instead of $g_\phi(x_i, x_j) = g_\phi(x_i), g_\phi(x_i, x_j) = g_\phi(x_j - x_i)$, or $g_\phi(x_i, x_j) = g_\phi(x_i, x_j - x_i)$ tried in Wang et al. (2019b), $g_\phi(x_i, x_j) = g_\phi(x_i, x_j + x_i)/2$ is defined in this study as an symmetric edge function. By combining both the global structures and local neighbourhood characteristics, such a function is capable of acquiring more inherent and high-level features in an effective manner. Moreover, due to the variations of the number of points in each local neighbourhood, the average-based asymmetric edge function $g_\phi(x_i, x_j) =$

$g_\phi(x_i, x_j + x_i)/2$ is prone to error reduction and keep much information for further feature encodings.

Furthermore, it is remarkably significant to recalculate a new graph using the KNN algorithm within the D-dimensional feature space generated by previous layers. Thus, a new graph $G^l = (V^l, E^l)$ is constructed at each layer. Consequently, the D^{l+1}-dimensional outputs are calculated by using the revised EdgeConv to the D^l-dimensional outputs of the l-th layer from the following equation:

$$x_i^{l+1} = \boxdot_{j:(i,j)\in E^l}\, g_\phi^l(x_i^l, x_j^l) \tag{3.11}$$

where $g_\phi^l \in \mathbb{R}^{D^l} \times \mathbb{R}^{D^l}$. Such revised EdgeConv can be easily fed into existing architectures to boost the segmentation performance. The revised EdgeConv is combined with the basic version of hierarchical PointCONV and PointDeCONV framework. As depicted in Figure 3.1, an EdgeConv layer is employed after each revised PointCONV layer, followed by a fully connected (FC) layer then fed back to PointDeCONV layers. Within each EdgeConv module, $g_\phi^l\left(x_i^l, x_j^l\right) = g_\phi(x_i^l, x_j^l + x_i^l)/2$ is applied as a shared edge function, and the max operation is performed as the aggregation function. Moreover, the number of nearest neighbours k is predefined to be 32 for the effective segmentation process.

3.2.4 CRF-based Post-processing

Both CNNs and CRFs have demonstrated dominating performance in semantic segmentation tasks for 3D point clouds (Roynard et al., 2018; Dai et al., 2017). Precise point-wise semantic segmentation requires completely understanding not only high-level features of road objects but also mid- or low-level details. Such details are essential to ensure the consistency of point-wise label prediction. For instance, if two points are close to each other and have similar reflectance values, it is reasonable that these two points pertain to the same road object and therefore have the same semantic label. Thus, a CRF algorithm is performed for the label map refinement produced by the proposed PointCONV. Typically, an energy function is applied to CRF models using the following equation:

$$E(l) = \sum_{i=1}^{n} u_i(l_i) + \sum_{i,j}^{n} v_{i,j}(l_i, l_j) \tag{3.12}$$

where l_i denotes the i-th predicted label, i = 1, 2…, n, and n is the total number of point clouds. Moreover, $u_i(l_i) = -logP(l_i)$ is defined as the predicted probability $P(l_i)$ from the revised PointCONV. The second term in Eq. (3.12) indicates the penalty to assign labels to a couple of points and is therefore determined by $v_{i,j}(l_i, l_j) = \mu(l_i, l_j) \sum_{p=1}^{P} w_p k^p(f_i, f_j)$, where $\mu(l_i, l_j) = 1$ if $l_i \neq l_j$ or 0 otherwise, k^p represents the p-th Gaussian kernel depending on extracted features f from points i and j, and w_p denotes constant coefficients. Herein, two Gaussian kernels are chosen as follows:

$$k_1 \exp\left(\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|x_i - x_j\|^2}{2\sigma_\beta^2}\right) + k_2 \exp\,(\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}) \tag{3.13}$$

where k_1 exp(·) is determined by the 3D coordinates (x, y, z) and angular positions p of two adjacent points, and k_2 exp(·) is calculated only relying on angular positions. σ_α, σ_β, and σ_γ are three predefined hyperparameters. Accordingly, the fine-grained point-wise label prediction is achieved by minimizing the CRF energy function defined in Eq. (3.12). Although accurate minimization of Eq. (3.12) is intractable, Chen et al.(2017b) developed and revised a mean-field iteration method to handle this problem effectively and appropriately. The CRF can effectively leverage the prediction and confidence produced by the PointCONV-based classifier, as well as semantic label assignment between two similar points in each local region.

To compute and minimize the loss generated by MS-PCNN model, the off-the-shelf softmax cross entropy loss function is utilized after implementing CRF-based post-processing. More specifically, the softmax cross entropy is defined as follows:

$$Loss = L(g, h(y)) = -\sum_{i=1}^{N} g_i logS_i \tag{3.14}$$

where g_i represents the one-hot label of i-th training sample, N denotes the batch size, and $S_i = e^{V_i} / \sum_j e^{V_j}$ is the softmax prediction score vector. The main objective is to minimize the loss function expressed in Eq. (3.14). Finally, the point-wise semantic label is determined using the prediction score vector S_i.

3.3 Implementation Details

In all designed experiments, the proposed neural networks were tested using Tensorflow on Nvidia® GTX 1080 Ti GPU and 32 GB RAM. Moreover, the networks were optimized using adaptive moment estimation (Adam) optimizer which is built-in in Tensorflow. Batch normalization (BN) and rectified linear unit (ReLU) were employed after each MLP layer, except for fully connected (FC) layers. Several hyperparameters such as the batch size and initial learning rate were optimized during the training phase to determine the optimal combination by using the grid search approach. Specifically, the batch size, initial learning rate, the momentum of Adam, and dropout rate were predefined in the range of [8, 16, 32], [0.01, 0.001, 0.001], [0.80, 0.85, 0.90], and [0.5, 0.6, 0.7], respectively.

The Overall Accuracy (OA) and Intersection over Union (IoU) were used as the performance evaluation matrices. By implementing many experiments with all possible hyperparameter combinations, an optimal combination was determined as [8, 0.001, 0.9, 0.5]. Namely, MS-PCNN model was trained using Adam with a momentum of 0.9, a dropout rate of 0.5, and a batch size of 8. The initial learning rate was 0.001 with a decrease rate of 50% in every 25 iterations. Each test dataset was divided into 70%, 20%, and 10% subsets for training, testing, and validating, respectively. Finally, a total of 200 epochs was applied for training purpose.

3.4 Datasets

A large-scale outdoor MLS dataset was employed, called Paris-Lille-3D (Roynard et al., 2018) collected in complex urban environments. Paris-Lille-3D point cloud dataset was collected in the two metropolitan areas, namely Paris and Lille, France, using an MLS system equipped with a Velodyne HDL-32E LiDAR. The Velodyne HDL-32E LiDAR sensor can obtain a maximal measurement rate of 700,000 points per second in an effective scanning range of 80 m to 120 m. Such a sensor is installed at the rear roof of the vehicle with an angle of 30° between the horizontal axis and rotation axis, which can achieve a 2 cm measurement accuracy at the speed of up to 60 km/hr, resulting in MLS point densities ranging from 1,500-2,000 points/m^2.

In Paris-Lill3-3D, there are four data acquisition trajectories: Lille1_1 is a length of 620 m urban road segment with 30.2 million points, Lille1_2 is a length of 530 m urban road corridor with 30.1 million points, Lille 2 is a length of 340 m urban road with 26.8 million points, and Paris provides a 450 m length of urban road with 45.7 million points, respectively. There are 9 object classes were manually labeled as Ground, Buildings, Poles, Bollards, Trash Cans, Barriers, Pedestrians, Cars, and Natural with a total number of 2,479 object instances. Moreover, a total of 30 million points without labels are released as official test datasets. This dataset is obtained from complex urban road environments, it shows surveying conditions with occlusions and varying point densities in the real-world scenarios, thus resulting in considerable difficulties for road object segmentation using this dataset.

3.5 Results and Discussion

3.5.1 Hyperparameter Optimization

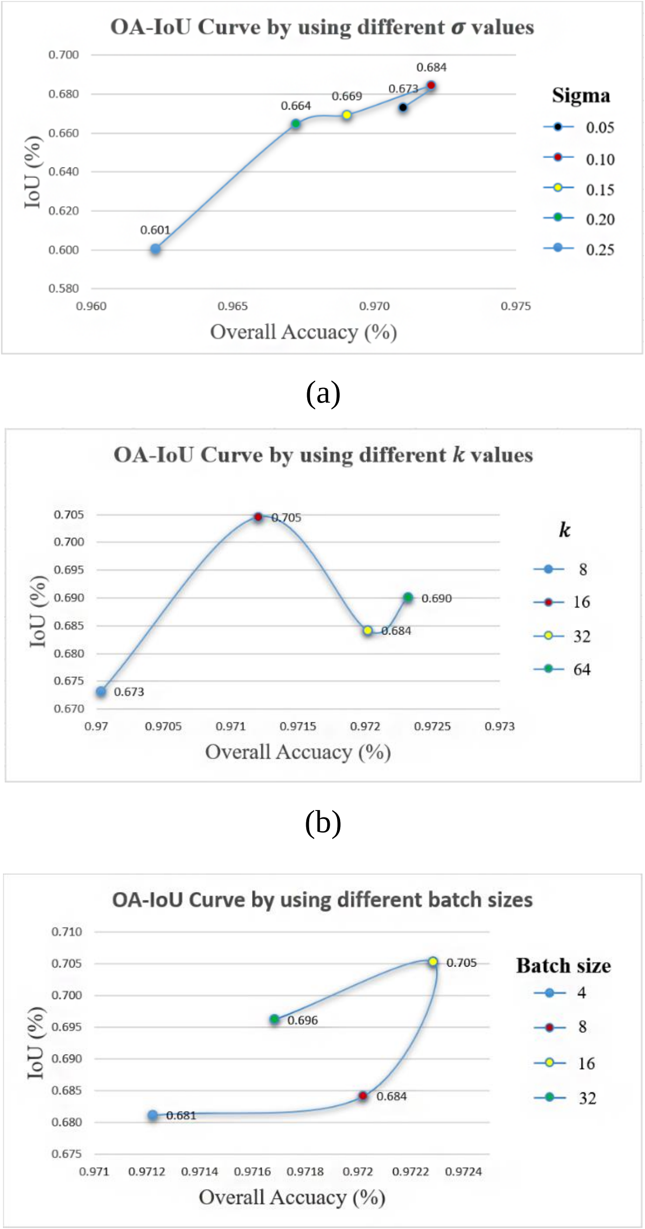
The proposed MS-PCNN framework has two essential hyperparameters: σ, the bandwidth in multi-scale kernel density estimation; and k, the number of points in each local neighbourhood. To achieve the optimal hyperparameter settings, MS-PCNN model performance was evaluated through multiple experiments based on two evaluation matrices, i.e., overall accuracy and intersection over union, which can be calculated as follows:

$$OA = \frac{\sum_{i=1}^{N} c_{ii}}{\sum_{j=1}^{N}\sum_{k=1}^{N} c_{jk}} \tag{3.15}$$

$$IoU = \frac{c_{ii}}{c_{ii}+\sum_{j\neq 1} c_{ij}+\sum_{k\neq 1} c_{ki}} \tag{3.16}$$

where *OA* metric represents the overall accuracy of segmentation results, and *IoU* metric measures the percent overlap between the target mask and the segmentation output. *N* is the number of classes, $c \in \mathbb{R}^{N\times N}$ is a confusion matrix of the segmentation method, where c_{ij} is the number of points from ground-truth class i predicted as class j.

Before conducting various experiments, the Paris-Lille-3D dataset was preprocessed by first downsampling input point clouds and then rotating and jittering them to enhance the robustness and applicability of the MS-PCNN network. Additionally, to ensure geospatial correlation among point clouds in the local neighbourhoods, the coordinates of all point clouds were normalized to [−1, 1] in a trajectory interval of 5 m. According to prior knowledge, the performance of MS-PCNN model was tested by using 5 (options of σ) × 4 (options of k) = 20 combinations. Since the number of combinations is relatively large, the different performance of each hyperparameter setting was evaluated through the variable-controlling approach. That is, only the value of one hyperparameter was changed each time, while remained the other hyperparameter values the same. To evaluate the influence of different hyperparameter combinations, the OA-IoU curve for all categories can be generated. Intuitively, the OA-IoU curve would fall in the top-right region of the plot, which indicates the MS-PCNN model can produce both high OA and IoU.



(a)

(b)

(c)

Figure 3. 6 Model performance evaluation through OA-IoU curves: (a) Using different σ values. (b) Using different k values. (c) Using different batch sizes.

(1) **Size of $\boldsymbol{\sigma}$**: The size of bandwidth σ has a significant impact on the performance of MS-PCNN model. An appropriate value of σ enables the model to learn more local features from input point clouds. The value of σ varies in the range of [0, 1]. More specifically, the smaller σ is, more local information the model can capture, but more computational energy consumes. To achieve an optimal balance between model performance and computational costs, the performance of MS-

PCNN network was tested by using different σ values, i.e., 0.05, 0.10, 0.15, 0.20, and 0.25 on three test datasets, while keeping k = 32 all the time by running 200 training epochs.

Figure. 3.6(a) presents the OA-IoU curve based on different σ values. Note that, the performance of MS-PCNN enhances with the decrease of σ, which achieves the best performance (i.e., OA = 97.2% and IoU = 68.4%) while setting σ = 0.10. The reason is that in the process of multi-scale kernel density estimation, MS-PCNN could capture more details with the decreasing values of σ. However, point-wise semantic segmentation performance decreases by 1.1% when changing bandwidth values from σ = 0.10 (IoU = 68.4%) to σ = 0.05 (IoU = 67.3%), that is because the MS-PCNN model is overfitting due to much redundant information used in the training phase. Therefore, σ = 0.10 was determined as the optimal hyperparameter value in order that the MS-PCNN model can deliver high robustness and computational efficiency.

(2) **Size of $\boldsymbol{k}$**: The number of points in each local neighbourhood, namely k, determines both the descriptiveness and robustness of MS-PCNN model in local feature extraction. It is normally predefined as k = 8, 16, 32, or 64 based on the different point densities for different test datasets. Accordingly, the performance of MS-PCNN model was evaluated by using different predefined k values on three test datasets, while keeping σ = 0.10 all through 200 training epochs.

Figure 3.6(b) shows the OA-IoU curve by using different k values. Different point densities have a significant impact on the selection of k values. More specifically, to obtain representative features in local regions, a relatively large k value should be selected for point cloud scenes with high point densities. Note that, the MS-PCNN model can achieve the best performance (i.e., OA = 97.1% and IoU = 70.5%) for point-wise segmentation while setting k = 16 on Paris-Lille-3D dataset. Obviously, for the per-point segmentation task, the mIoU increases by 2.1% by changing k = 32 to k = 16. Moreover, compared to k = 32 or 64, the computational efficiency at the stages of k-nearest neighbour searching and edge convolutions is considerably improved by setting k = 16. Thus, in MS-PCNN, k = 16 was defined as the optimal hyperparameter value for high segmentation accuracy and relatively low computational costs.

Moreover, the influence of using different batch sizes during the training phase was also evaluated. Figure 3.6(c) presents the OA-IoU curve by varying batch sizes from 4 to 24 (i.e., 4, 8, 16 and 24) based on the accessible computational power, while keeping other hyperparameters the same (e.g., σ = 0.10 and k = 16). Generally, the larger the batch size, the more global feature the model captures (Qi et al., 2017a), yet the more computational power the model requires. As can be seen, the MS-PCNN model can deliver the best segmentation accuracy by setting the batch size to be 16 ($mIoU$ = 70.5%). However, the relatively low segmentation accuracy ($mIoU$ = 69.6%) is achieved when setting the batch size as 32, because the MS-PCNN model captures relatively fewer local features than global features resulting in lower model performance. Accordingly, an optimal hyperparameter combination was ascertained as σ = 0.10, k = 16, and batch size to be 16, respectively.

3.5.2 Segmentation Results on Paris-Lille-3D

According to different experiments by using various combinations, the optimal combination was determined as σ = 0.10 and k = 16 on the Paris-Lille-3D test dataset. Moreover, the initial learning rate, batch size, momentum of Adam, dropout rate and epochs are 0.001, 16, 0.9, 0.5, and 200, respectively, which can deliver the best segmentation result. Since the design of the MS PCNN architecture depends on experience, other parameters are thus ascertained through trial and error. For instance, when determining the dimension of the output channel, it is common to utilize an increasing size (e.g., from 64 to 512) in the encoding layers and a decreasing size (e.g., from 512 to 128) in the decoding layers.

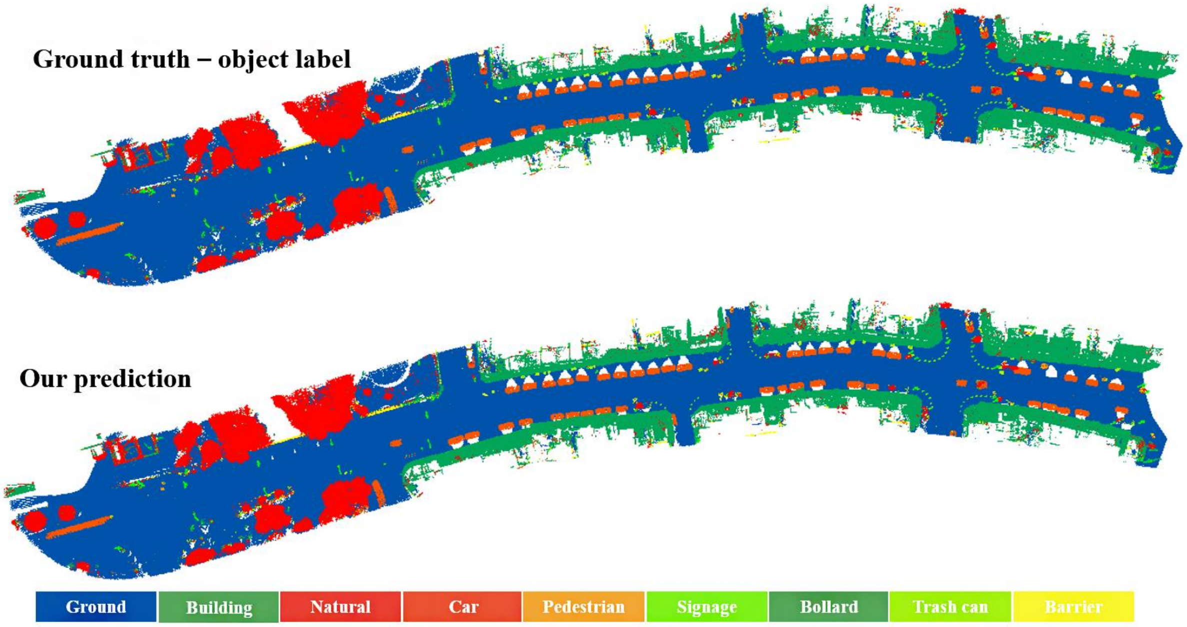


Figure 3. 7 Point-wise segmentation results by using MS-PCNN network on Paris-Lille-3D dataset.

Figure 3.7 illustrates the experimental result by testing with Lille2 dataset, which demonstrates that MS-PCNN model is able to achieve promising solutions for point-wise segmentation tasks in large-scale urban environments. Although mobile LiDAR point clouds collected in urban road scenes are very different from small-scale CAD models, the segmentation results indicate a large number of road objects (e.g., buildings and poles) were effectively segmented and the road surfaces were completely extracted. However, some points failed to be segmented, which indicates that certain points were mis-assigned as other semantic labels, as illustrated in Figure 3.8 The complexity of road scenarios has a significant impact on the descriptiveness of the MS-PCNN network. Based on the zoom-in visual inspection, the decay, ground settlement, occlusion, and moving obstacles (e.g., cyclists) in the Paris-Lille-3D dataset could lead to the false point-wise label assignment. Figure 3.8 also shows some pedestrian points were misclassified as natural and some points belonging to cars were predicted as barriers. Such unavoidable errors evolving in the process of data preprocessing, such as batch normalization, also conduce to overall accuracy reduction of point cloud segmentation.

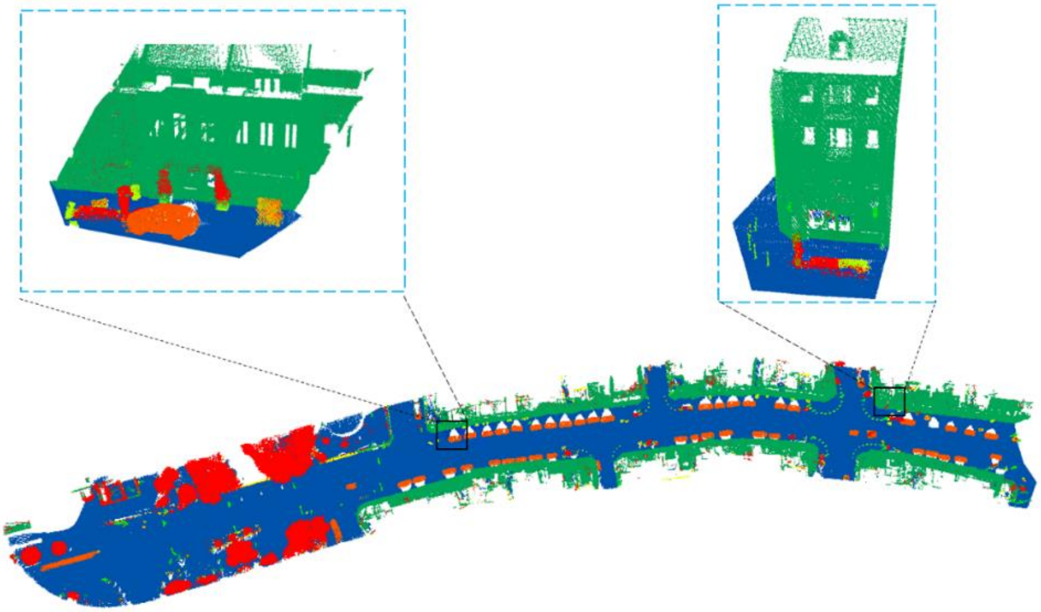


Figure 3. 8 Two zoom-in views of point-wise segmentation results from Paris-Lille-3D dataset.

Accordingly, based on the same testing protocols, the proposed MS-PCNN model was compared with these existing networks. Table 3.1 presents the performance comparison results by calculating the mean IoU ($mIoU$) matrix, which is the mean of IoU across all the object categories.

As can be perceived, the proposed method dramatically outperforms both PointNet (38.6% $mIoU$) and PointNet++ (32.0% $mIoU$), which are pioneers that directly consume point clouds using deep learning. In addition, compared to the DGCNN, the proposed MS-PCNN method could dynamically update K-nearest neighbours between two adjacent layers of the model and accordingly calculate the sequence of embeddings, resulting in a 17.6% $mIoU$ improvement. Moreover, PointSIFT (62.7% $mIoU$) obtain lower segmentation accuracy than MS-PCNN. Most importantly, for certain types of road objects including signages, bollards, pedestrians and cars, the proposed model can deliver the dominating performance in semantic segmentation. In conclusion, the MS-PCNN model can achieve state-of the-art point-wise segmentation performance in large-scale urban environments. Meanwhile, the comparative study inspires us to optimize the MS-PCNN model by using more low-level features of point clouds, e.g., RGB and normal vectors.

Table 3. 1 Semantic segmentation results on Lille2 by using different methods.

Methods	Ground	Building	Signage	Bollard	Trash Can	Barrier	Pedestrian	Car	Natural	mIouU(%)
PointNet	97.3	90.4	22.9	8.7	3.2	2.5	24.3	71.9	26.3	38.6
PointNet++	96.6	78.6	15.2	4.3	1.2	0	19.3	46.3	26.1	32.0
DGCNN	98.3	93.1	52.9	36.1	19.5	15.0	16.6	88.6	56.5	52.9
PointSIFT	98.4	95.6	51.2	44.8	53.9	31.4	31.3	87.4	70.7	62.7
MS-PCNN	98.1	95.4	57.6	64.6	63.0	34.1	57.7	95.2	68.3	70.5

3.6 Efficiency Evaluation

Although the CRF-based postprocessing could strengthen robustness, it would have a direct influence on the memory consumption and time complexity (i.e., forward and backward propagations) of the whole framework. Most notably, it may affect the segmentation results. To estimate these influences, the MS-PCNN network was tested using a desktop equipped with Intel$^{®}$ i7 8700K CPU @ 4.7GHz and Nvidia$^{®}$ GTX 1080 Ti GPU with and without the CRF module. 200 epochs were run on the Paris-Lille-3D dataset. Additionally, the average processing time and the highest GPU memory size were recorded for two different models.

Table 3.2 presents the comparison results. It is notable that the model size and GPU-memory usage using the network architecture with CRF module is about 260 MB and 5,015 MB, respectively. The reason is that performing the CRF module could linearly increase the number of parameters of MS-PCNN network. Besides, the mean IoU increases by 2.7% by introducing the CRF module into MS-PCNN architecture, which demonstrates the CRF module is capable of further achieving the point-wise segmentation refinement. Additionally, the time consumption of each forward and backward propagation process in MS-PCNN is over two times than that in the network without the CRF operation. Obviously, the point-wise semantic segmentation performance is greatly improved by employing a CRF post-processing module. Since CRFs are able to directly model spatial structures and capture more inherent geometric characteristics (e.g., connectivity between two adjacent points), the MS-PCNN model can be fine-tuned by formulating the CRF module. Furthermore, the proposed MS-PCNN network can achieve state-of-the-art point-wise segmentation performance in outdoor environments with different data distributions and requires less GPU memory usage. By predefining the batch size as 16, the proposed MS-PCNN baseline only consumes 4,824 MB GPU memory space compared to 11,450 MB used in PointSIFT network, which demonstrates the MS-PCNN model is less sensitive to data distributions and computational consumptions.

Table 3. 2 Performance evaluation of MS-PCNN network with and without the CRF module on Paris-Lille-3D dataset.

Model	Size (MB)	GPU memory usage (MB)	Time spend of each forward propagation (ms)	Time spend of each backward propagation (ms)	mIoU (%)
MS-PCNN	224.3	4,824	108.06	307.25	67.8
MS-PCNN + CRF	259.9	5,015	235.27	631.61	70.5

3.7 Chapter Summary

This paper tackles the problems related to 3D point cloud segmentation tasks, particularly in large-scale scenes. Such problems result in computation complexity and robustness reduction when dealing with 3D highly dense point clouds, most notably due to its various point density and irregular data format, as well as occlusion and background interference in the real world. In this paper, a novel end-to-end neural network, MS-PCNN, is proposed by combining point-wise CNNs with dynamic edge convolutions for 3D point cloud segmentation. The proposed network was evaluated by estimating efficiency and robustness on the real urban-scene LiDAR datasets, i.e., Pairs-Lille-3D.

In conclusion, the proposed neural network has four main strengths: First, the revised point-wise convolutional filters that can learn spatial relationships and extract geometric information of point clouds in local regions contributing to permutation invariance and translation invariance. Second, MS-PCN applies a hierarchical PointCONV-based downsampling and DePointCONV-based upsampling architecture in order that more high-level features are extracted in multiple scales. Third, by improving the dynamic graph edge convolution, MS-PCNN can learn edge features between a point and its adjacent neighbours to improve the descriptiveness. Finally, a CRF post-processing algorithm is used to ensure the consistency of point-wise label prediction and refine segmentation results. MS-PCNN model is robust to diverse point density and intensity distributions for the complex urban-scene point clouds. Therefore, this paper demonstrates that the MS-PCNN model can provide promising solutions in industrial applications, such as fully autonomous driving. Compared to other point-based networks, e.g., PointSIFT and PointCNN, MS-PCNN is less memory-consuming and time-consuming in both forward and backward propagations, which can considerably save the training time. Additionally, the comparative study certainly indicates that MS-PCNN is superior to other DL-based methods in the testing scenarios in segmentation accuracy and computational complexity. Overall, it is concluded that the proposed neural network can achieve dominating performance in 3D point cloud segmentation under large-scale point cloud scenes more effectively and robustly.

Chapter 4

Capsule-based Networks for Road Marking Extraction and Classification

Accurate road marking extraction and classification play a significant role in the development of AVs and HD maps. Due to point density and intensity variations from MLS systems, most of the existing threshold-based extraction methods and rule-based classification methods cannot deliver high efficiency and remarkable robustness. This chapter details the theoretical and mathematical implementations of the developed capsule-based network architectures for road marking extraction and classification by using mobile LiDAR point clouds. Section 4.1 introduces the research backgrounds. Section 4.2 presents the detailed workflow of the proposed framework, which mainly contains three modules: data-preprocessing, road marking extraction, and road marking classification. Section 4.3 presents the datasets and implementation details. Section 4.4 presents the experimental results and discussion, followed by the computational efficiency evaluation in Section 4.5. A chapter summary is presented in Section 4.6. © [2020] IEEE. Reprinted, with permission, from [Lingfei Ma, Ying Li, Jonathan Li, Yongtao Yu, José Marcato Junior, Wesley Nunes Gonçalves, and Michael A. Chapman. Capsule-based Networks for Road Marking Extraction and Classification from Mobile LiDAR Point Clouds. *IEEE Trans. Intell. Transp. Syst*., doi: 10.1109/TITS.2020.2990120].

4.1 Introduction

Nowadays, many leading digital mapping corporations (e.g., Here, TomTom, Google Maps, and Bing Maps) and multinational courier services companies (e.g., UPS, FedEx, and SF Express), are investing increasingly and dedicating themselves to produce high-definition (HD) maps. Such HD maps are capable of providing sub-lane level road information and highly detailed road inventories, including traffic signs, pole lights, roadside trees, lanes, boundaries, curbs, and all other essential road assets required for the development of autonomous vehicles (AVs) and intelligent service robotics (ISRs) (Chu et al., 2018). As a critical element in HD maps, road markings play a significant role in guiding, regulating, and forbidding all road participants (Bétaille and Toledo-Moreo, 2010). For instance, lane lines regulate driving zones, painted texts indicate traffic rules, and arrows show allowable driving directions. Therefore, accurately extracting and classifying road markings have a significant impact on transportation-related policymaking, driving behaviour regulation, and traffic collision reduction.

A series of research has been conducted for road marking segmentation and classification using 2D images obtained from vehicle-borne optical imaging systems (Jung et al., 2009). A group of geometric feature functions in a probabilistic Random Under Sampling Boost (RUSBoost) and Conditional Random Field (CRF) classification framework was employed to automatically learn the rules embodied in road markings from stereo images (Mathibela et a., 2015). A trainable multi-task model was developed for pavement marking recognition and segmentation from images acquired under complex road topotaxy and varying traffic conditions (Lee et al., 2017). Moreover, line markings were extracted by creating a novel line proposal unit embedded in a fully convolutional network (FCN) for valid feature encodings (Li et al., 2019c), which achieved the promising performance on MIKKIand TuSimple image datasets. However, such image-related methods are highly susceptible to weather and illumination variations (Máttyus et al., 2016).

Mobile laser scanning (MLS) systems comprising a combined Global Navigation Satellite System and Initial Measurement Unit (GNSS/IMU) subsystem, a Light Detection and Ranging (LiDAR) subsystem, a Radio Detection and Ranging (RADAR) subsystem, CCD cameras, and a central computing subsystem, can collect highly dense and accurate point clouds with intensity or reflectance information in largescale urban environments and highways (Ma et al., 2018). Compared to vehicle-mounted cameras, LiDAR sensors are less sensitive to ambient lighting conditions (Yang et al., 2017). The point density collected by MLS systems can achieve over 10,000 pts/m^2 with cm-level resolutions, while it is quite challenging for both terrestrial and airborne laser scanning (TLS/ALS) platforms to deliver such precision and flexibility (Chen et al., 2019a).

Therefore, many studies focusing on the road marking extraction and classification have been addressed by using MLS point clouds (Wan et al., 2019). However, massive and unevenly distributed 3D point clouds make the intelligent point cloud processing challenging. Occlusions and distortions, intensity variations, density variations, noisy points, and incomplete pavements during MLS data acquisition also result in considerable difficulties. Since threshold-based methods at a global scale cannot effectively extract road markings from georeferenced images with various point distributions, multi-threshold methods are accordingly proposed by partitioning road surface point clouds into a set of data blocks and determining an adaptive threshold within each data block (Guan et al., 2014). Nevertheless, such methods highly rely on suitable data block sizes. Meanwhile, Jung et al., 2019 performed a normalized intensity-based approach to minimize the impacts of different intensity values due to varying distances from the onboard laser scanners to scanning objects. However, the normalization parameters defined in such methods are different from scene to scene.

Generally, there exist three main challenges for road marking extraction and classification from mobile LiDAR data: (1) the contrast between pavements and road markings is relatively low. Road damage is inevitable regarding poor maintenance, which leads to the unevenly distributed intensity, thereby resulting in intensity-related methods ineffective. (2) The intensity values and point densities are varying. Point clouds are generally acquired by vehicle-based MLS systems that are driven through changing lanes at varying driving speeds. Depending on the profiling scanning mechanism of MLS systems, the incident angle of laser beams grows larger with an increased scanning range. Consequently, road markings have higher intensity values and point densities if they are closer to the trajectory of MLS systems. It is challenging for threshold-based extraction methods to effectively extract road markings by assuming that intensity and point density are uniformly distributed. (3) Some road markings are incomplete. The damage of road surfaces resulting from on-road overloaded trucks and severe weather conditions, such as acid-alkali erosion, could create worn and decaying road markings. Moreover, occlusions from all road participants (e.g., vehicles and cyclists) also bring in dilemmas and uncertainties for the accurate extraction and classification of road markings. Accordingly, manual editing and post-refinement are required to improve the completeness and accuracy of extracted road markings. However, it is time-consuming and labour-intensive.

To deal with these challenges, the feasibility of combing capsule networks with hierarchical feature encodings of georeferenced feature images is investigate. Compared to the conventional convolutional neural networks (CNNs), capsule networks have achieved superior performance in image segmentation and classification tasks, which captures more intrinsic features in pose and spatial relationships of different objects in images (Jaiswal et al., 2018, Duarte et al., 2019). In this paper, two capsule-based neural network architectures are developed for road marking extraction and classification by using MLS point clouds. To this end, a pixel-wise U-shaped road marking extraction network is proposed to segment road markings from input images. At first, the road surface is partitioned into a collection of image patches. Then, the Intersection-over-Union (IoU) loss is employed, rather than cross-entropy, to guide weight updates in the U-shaped segmentation architecture. Finally, road markings are extracted based on binary classification. Moreover, combined with the fully connected (FC) capsule layers, a capsule-based network is constructed to classify road markings. First of all, the extracted road markings are resized to 28 × 28 pixels for computational complexity reduction. Then, two sibling classification networks (i.e., a capsule-based network and a fully connected capsule network) are trained to encode both low-level and high-level features for different road marking classes, followed by a revised dynamic routing algorithm. Meanwhile, a large-margin Softmax (L-Softmax) loss function is adopted in the capsule-based classification model to guide training, instead of a standard Softmax loss. Finally, road markings are effectively categorized into seven groups, including lane lines, dashed lines, zebra crossings, straight arrows, turn arrows, diamonds, and texts.

The whole road marking extraction and classification framework provides a promising solution for preloaded HD map creation, which further produces an essential road inventory dataset for road marking updates to support the development of AVs. The significant contributions of this paper are as follows. (1) A novel U-shaped convolution-deconvolution capsule network is constructed to extract road markings. The impacts of low-intensity contrast between road markings and their surrounding pavements, as well as varying point densities, are remarkably decreased through encoding the image patches at various locations. (2) A hybrid capsule network is proposed to categorize road markings with the assistance of a revised dynamic routing algorithm. The sibling framework of the capsule model and the fully connected capsule model achieves more effective performance for road marking classification. (3) To date, it is the first use of capsule-based neural networks for road marking extraction and classification in literature. And (4) a road marking dataset containing both 3D point clouds and manually labeled reference data in three types of road scenes (i.e., urban roads, highways, and underground garages) is constructed, which will be publicly accessible to motivate relevant research.

4.2 Algorithm Description

4.2.1 Data Pre-processing

Since this study mainly concentrates on road markings, the off-ground point clouds (e.g., trees, traffic lights, fences, and buildings) are first filtered out to strengthen the computational efficiency and reduce GPU memory consumption in the following processes. In the previous work (Ma et al., 2019a), a revised curb-based road surface extraction method was introduced. Given the fact that urban roads are constructed with concrete curbs as road separation zones, road surface point clouds can be accurately and effectively segmented from the input point clouds depending on the fitted curb-lines. Herein, this curb-based extraction method is employed to segment road surface point clouds.

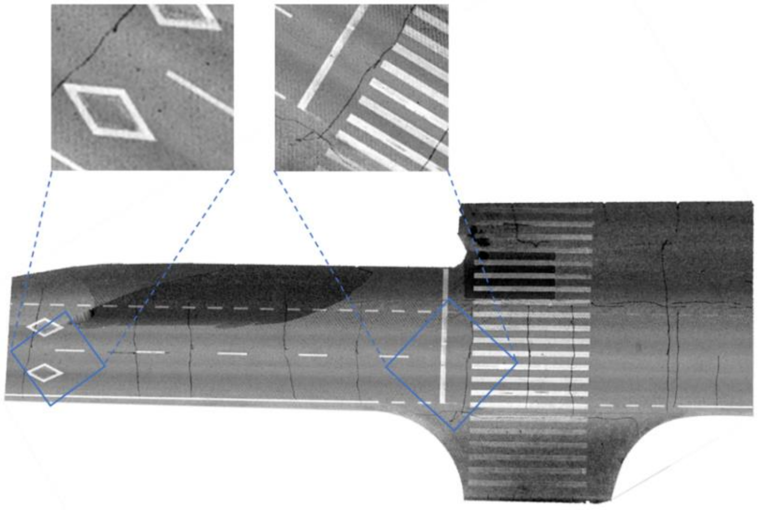


Figure 4. 1 Intensity image generated by using IDW interpolation.

Moreover, existing studies have demonstrated that the height information of road point clouds conduces little to road marking segmentation (Yang et al., 2013). Thus, in this study, road surface point clouds are projected to a 2D xy-plane and transformed into georeferenced intensity raster images. To this end, an inverse distance weighting (IDW) interpolation algorithm was employed to produce 2D intensity images by calculating the grey-scale value of a specific cell from its surrounding neighbours. Two rules are designed to determine the weight associated with each point: (1) a point with a larger intensity value has a higher weight, and (2) a point closer to the trajectory has a higher weight. The grid cell size should adequately preserve the details of different road markings and dramatically decreases the number of data that should be handled. Theoretically, A larger grid cell size is selected when performed on point clouds with lower density.

Based on the prior knowledge (Cheng et al., 2016), several grid cell sizes from 2.5 cm to 10 cm were tested. The generated intensity images became blurred, and the computational cost was reduced. Moreover, with a grid cell size of 4 cm, the thinnest road markings (i.e., lane lines with a width of 15 cm) are well preserved in the generated intensity images, and the gaps among 3D point clouds are accordingly interpolated.

Furthermore, a high-pass filtering operation with a suitable kernel size is performed on the generated intensity images to minimize the influence of varying intensity values. The kernel size of this high-pass filter is ascertained based on prior knowledge and multiple experiments. Specifically, this kernel size should be not only large enough to comprise both road marking and road surface pixels but small enough to reduce the influence of the spatial variance and uneven distribution of the intensity. Herein, the raster grid size of the generated intensity image and the kernel size of the high-pass filter are defined as 4 cm and 25 × 25, respectively. Figure 4.1 indicates an example of the generated intensity image after implementing IDW interpolation and high-pass enhancement.

4.2.2 U-shaped Capsule Network

Since a rasterized cell either denotes some road marking pixels or pavement pixels in the intensity images, the road marking extraction process can be regarded as a basic binary classification task. Meanwhile, although capsule networks introduced by Sabour et al. (2017) have achieved remarkable success for a broad range of computer vision problems particularly for digit recognition and small image classification, no studies yet exist in literature that employs capsule networks for road marking extraction from MLS point clouds. Comparing with conventional CNNs, capsule networks utilize vectorial neurons rather than scalar neurons to encode entity features. The instantiation parameters of different capsules indicate varying types of entities, while different lengths of capsules encode the probabilities of the existence of these entities, and different directions indicate their pose information (Yu et al., 2019). Therefore, to demonstrate the effective performance of capsule networks in extracting road markings, a U-shaped capsule-based network is designed using the 2D georeferenced intensity images.

In the training process, the generated intensity images are first manually labeled into two groups: positive training samples containing road marking pixels and negative training samples containing road surface pixels. Subsequently, a collection of local image patches with the size of

512 × 512 pixels are derived from the generated intensity image based on a sliding window mechanism. To ensure complete and extensive coverage of the training image, two adjacent image patches are generated with an overlapping size of p_s pixels. Moreover, such patches are fed into the multi-layer capsule networks for intrinsic feature extraction.

Figure 4.2 shows a U-shaped convolution-deconvolution capsule network, which can learn not only intensity variance from massive labeled image patches but the shape and position information of road markings. This U-shaped capsule network consists of traditional convolutional layers, primary capsule layers, convolutional capsule layers, and deconvolutional capsule layers. The traditional convolution layers are designed to encode locally shallow features (e.g., edges and shapes) from the input local image patches via convolutional encodings. Such low-level features are afterward fed into high order capsules to learn in-depth features. Herein, the commonly employed rectified linear unit (ReLU) is used as the activation function.

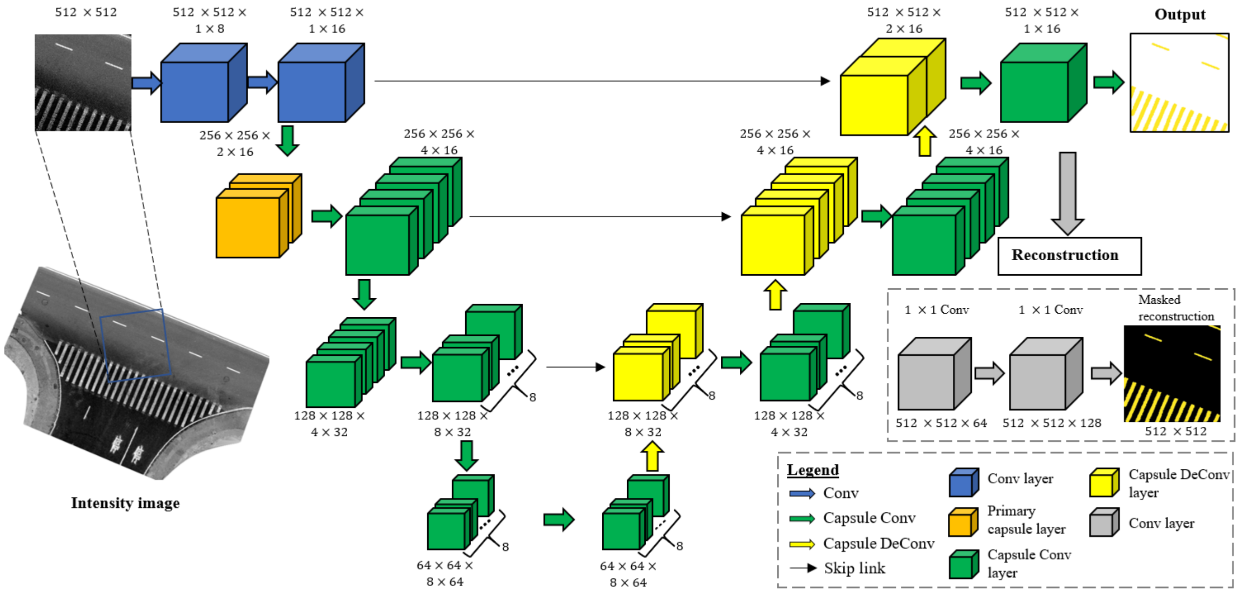


Figure 4. 2 Architecture of the proposed U-shaped capsule network.

In primary capsule layers, the shallow scalar feature encodings are transformed into high-order vectorized capsule representations. Assuming F_m and D_c are the number of feature maps and the dimension of capsules, respectively. Then, the kernels with the size of $F_m \times D_c$ are implemented in the following convolutional layer. Finally, the output feature maps are categorized as F_m groups, each of which consists of D_c feature maps.

The following convolutional capsule networks focus on encoding high-level capsule features and orientations by using capsule convolution operations. In general, the whole inputs to a capsule j are a weighted sum over all outputs from the capsules in the convolutional kernel in the previous layer:

$$C_j = \sum_i h_{ij} \cdot \hat{V}_{j|i} \tag{4.1}$$

where C_j denotes the whole input to the capsule j, h_{ij} is the coupling coefficient showing the level of significance that capsule i in the previous layer activates capsule j, and $\hat{V}_{j|i}$ indicates the predictions between capsule i and capsule j, which is calculated by:

$$\hat{V}_{j|i} = W_{ij} V_i \tag{4.2}$$

where W_{ij} denotes the weight matrix and V_i is the outcomes of capsule i. The sum of the weighting coefficients between capsule i and all its linked capsules in the previous layer is equal to 1, which is ascertained through a dynamic routing mechanism (Sabour et al., 2017).

Moreover, a nonlinear "Squashing" activation function is employed to guarantee that different lengths of vectors in capsules are shrunk in the range of [0, 1] and results in the different probabilistic predictions. This squashing function is calculated by:

$$S_j = \frac{\|C_j\|^2}{1+\|C_j\|^2} \cdot \frac{C_j}{\|C_j\|} \tag{4.3}$$

where C_j and S_j is the total input and output vector of capsule j, respectively. Furthermore, three deconvolutional capsule layers are designed to construct a diverse set of capsule types and propagate learned features from downsampled images to the original images, thereby allowing the capsules to capture the most critical and intrinsic parameters of the input images. Based on the IDW interpolation algorithm, the feature propagation process is performed by interpolating feature values f of p_n pixels at coordinates of the p_{n-1} pixels. Then, such interpolated features are locally-constrained and concatenated with skip linked pixel features from the convolution capsule layers. Since this process only focuses on the distributions of the positive input class (i.e., road marking pixels) and regard the remaining pixels as background, all capsules except whose class labels match to the input image patch are masked out. This reconstruction process is conducted by employing three 1×1 convolutional layers.

Compared to the standard capsule network, the parameters is iteratively updated by using the IoU as the cost function for model performance refinement, rather than binary cross-entropy. The IoU loss function, namely L_{IoU}, is calculated as follows:

$$L_{IoU} = -\frac{\sum_{i=1}^{N}(p_{pred}^{i} \cap p_{gt}^{i})}{\sum_{i=1}^{N}(p_{pred}^{i} \cup p_{gt}^{i})} \tag{4.4}$$

where p_{pred}^{i} and p_{gt}^{i} is the i-th predicted road marking pixel and corresponding ground truth pixel, respectively. To minimize the L_{IoU}, the proposed U-shaped capsule network can extract more accurate and complete road markings in image patches. Moreover, images with various intensity are utilized as training data to decrease the impacts of intensity variation.

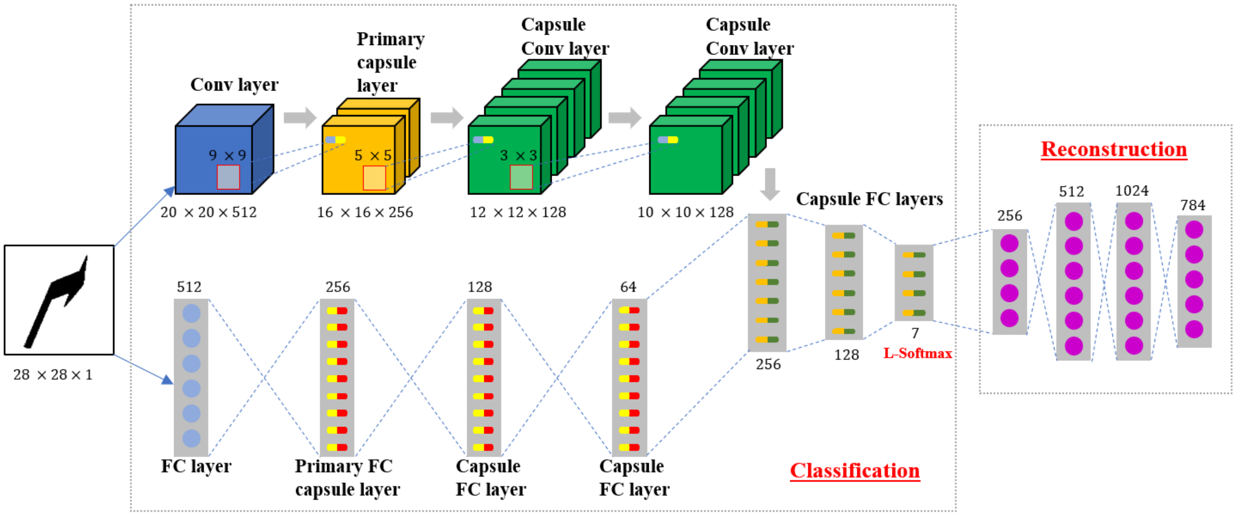


Figure 4. 3 Architecture of the proposed hybrid capsule network.

4.2.3 Hybrid Capsule Network

After road markings are segmented from the georeferenced intensity images, a hybrid capsule-based network is further proposed to categorize these road markings into seven classes, including lane lines, dashed lines, zebra crossings, straight arrows, turn arrows, diamonds, and texts. Figure 4.3 shows the workflow of the hybrid capsule framework, which mainly consists of two hierarchical networks (a convolutional capsule network and an FC capsule network) for, respectively, encoding high-level and low-level features from input images. As indicated in Figure 4.3, the convolutional capsule network comprises a standard convolutional layer and a primary capsule layer, followed by two convolutional capsule layers. By taking advantage of convolutional operation, the first convolutional layer functions to encode locally low-level features from the input images. Such extracted features are then fed into high order vectorial capsules for further feature encodings.

Algorithm 1 Revised dynamic routing algorithm.

```
procedure DYNAMIC ROUTING ($\hat{\mathbf{V}}_{\mathbf{j|i}}$, $\mathbf{r}$, $\mathbf{k}_h$, $\mathbf{k}_w$)
    for all capsule $\mathbf{i}$ within a $\mathbf{k}_h * \mathbf{k}_w$ kernel in layer $\mathbf{l}$ and all capsule $\mathbf{j}$ in layer $(\mathbf{l}+\mathbf{1})$ : $\mathbf{b_{ij}} \leftarrow \mathbf{0}$
    for $\mathbf{r}$ iterations do
        for all capsule $\mathbf{i}$ in layer $\mathbf{l}$: $\boldsymbol{l_i} \leftarrow \boldsymbol{L}_{\text{Softmax}}\,(\boldsymbol{b_i})$          ⇒ L-Softmax computes Eq. (4.5)
        for all capsule $\mathbf{j}$ in layer $(\mathbf{l}+\mathbf{1})$: $\boldsymbol{c_j} \leftarrow \sum_i \boldsymbol{h_{ij}} \cdot \widehat{\boldsymbol{V}}_{j|i}$
        for all capsule $\mathbf{j}$ in layer $(\mathbf{l}+\mathbf{1})$: $\boldsymbol{s_j} \leftarrow$ squash $(\boldsymbol{c_i})$          ⇒ Squash computes Eq. (4.3)
        for all capsule $\mathbf{i}$ in layer $\mathbf{l}$ and all capsule $\mathbf{j}$ in layer $(\mathbf{l}+\mathbf{1})$: $\boldsymbol{b_{ij}} \leftarrow \boldsymbol{b_{ij}} + \widehat{\boldsymbol{V}}_{j|i} \cdot \boldsymbol{s_j}$
    return $\boldsymbol{s_j}$
```

To enhance the weight update efficiency and improve inter-class separability, a revised dynamic routing algorithm is proposed. Different from the dynamic routing method conducted by Sabour et al. (2017), the revised dynamic routing algorithm only route the child capsules within the user-specified kernel to the parent, rather than routing all child capsules to all parent capsules. The revised dynamic routing algorithm is described in Algorithm 1. Moreover, a large-margin Softmax loss (Liu et al., 2016) is adopted to emphasize intra-class compactness and overcome inter-class imbalance, which usually poses challenges by using standard Softmax loss. The large-margin Softmax loss is calculated as follows:

$$L_{softmax} = -\log\left(\frac{e^{\left\|W_{y_i}\right\|\|x_i\|\psi(\theta_{y_i})}}{e^{\left\|W_{y_i}\right\|\|x_i\|\psi(\theta_{y_i})}+\sum_{j\neq y_i} e^{\left\|W_{y_i}\right\|\|x_i\|\cos(\theta_j)}}\right) \tag{4.5}$$

where W_{y_i} indicates the y_i-th column of a FC-capsule layer W, θ_j $(0 < \theta_j < \pi)$ represents the angle between the vector W_j and x_i. Generally, $\varphi(\theta) = \cos(m\theta)$ is defined, $0 \leq \theta \leq \frac{\pi}{m}$ and $\varphi(\theta) = \mathcal{F}(\theta)$, $\frac{\pi}{m} \leq \theta \leq \pi$. m presents an integer that is closely correlated to the classification margin. With smaller m, the classification margin becomes smaller, and the learning objective becomes easier accordingly. Based on prior knowledge, m is defined as 5. Furthermore, $\mathcal{F}(\theta)$ is a function that monotonically decreases in the range of [0, π], while $\mathcal{F}(\frac{\pi}{m})$ equals to $\cos(\frac{\pi}{m})$. By such an introduction, the large-margin Softmax loss not only gains the main strengths from Softmax loss but encodes inherent features at a large angular margin between different classes.

As perceived in Figure 4.3, the FC-capsule network contains a standard FC layer, a primary FC-capsule layer, and two FC-capsule layers. Intuitively, the FC layer is employed to encode shallow global features from the raw images. Then, such extracted global features are fed into high order capsules. Likewise, according to traditional fully-connected operations, the primary FC-capsule layer is generated. The corresponding units are equally divided into categories to construct a group of capsules. Meanwhile, two FC-capsule layers focusing on extracting inherent capsule features on a global scale are employed.

The two hierarchical networks encoding both local and global capsule features are combined through flattening and concatenation operations and further input into three FC-capsule layers for the classification task. Finally, four FC layers are employed to rebuild the input images, thus enable capsules to learn the most intrinsic and critical instantiation parameters of the raw images. Accordingly, all classification capsules are masked out, except for the remaining capsules whose class labels correspond to the raw image. The instantiation parameters of these capsules are input to the reconstruction module for reconstruction.

The training parameters are effectively fine-tuned at the error back-propagation stage. Accordingly, the following multitask loss function is adopted to refine the whole framework:

$$L = \sum_{i=1}^{M} \sum_{c=1}^{N} L_1^c + \delta \sum_{i=1}^{M} L_2^i \tag{4.6}$$

where L_1^c and L_2^i denote the classification loss and reconstruction loss, respectively. M and N are, respectively, the number of input training images and class-related capsules within the L-Softmax layer. δ indicates a regularization term. Accordingly, the classification loss for the specified class c is calculated as follows:

$$L_1^c = T_c \cdot \max(0, m^+ - \|S_c\|)^2 + \lambda(1 - T_c) \cdot \max(0, \|S_c\| - m^-)^2 \tag{4.7}$$

where T_c = 1, if the training image corresponds to class c. Otherwise, T_c = 0. m^+ and m^- are thresholds, respectively, that determine if a training image belongs to class c or not. In this study, m^+ = 0.93 and m^- = 0.07 are predefined based on a set of experiments. λ represents a regularization term. Additionally, a smooth-L loss proposed by Girshick (2015) is adopted to determine the reconstruction loss.

In the classification process, a group of road marking training data is manually labeled. Then, such training samples are augmented through rotation, scaling, cropping, and illumination changes. In total, 7,000 training samples are generated, with 1,000 samples for each road marking type. Moreover, to reduce the computational cost, all training samples are resized to 28×28 pixels before feeding into the classification network. Finally, the class label of a road marking image is ascertained using the following equation:

$$K^* = argmax_c \| S_c \| \tag{4.8}$$

where $\| S_c \|$ denotes the output of the classification network. This class label is therefore assigned to the image as its predicted road marking type.

4.3 Datasets and Implementation Details

The mobile LiDAR point clouds were captured using a RIEGL VMX-450 MLS system in both urban and highway road sections with free-flowing traffic. The RIEGL VMX-450 MLS platform contains two full-circle RIEGL VQ-450 laser heads, which could achieve a 400 lines/sec scan frequency in a “Butterfly” configuration pattern. The average point density of these data is over 4,500 pts/m^2, and the absolute measurement accuracy from laser heads can reach 8 mm. Additionally, some pavement point clouds were collected in underground garage environments using a self-assembled backpack laser scanning (BLS) system. This BLS system is equipped with two Velodyne VLP-16 laser heads, which can achieve a scanning distance of 100 m with 1,700 pts/m^2 point density and 3 cm measurement accuracy. Compared to point clouds obtained by MLS systems, the point clouds captured by the BLS system are of low quality with low point densities due to low-end LiDAR sensors and poor illumination conditions in underground garage environments.

Moreover, the raw data was manually annotated to build a road marking benchmark dataset. To end this, all road markings were first labeled pixel-by-pixel on the generated intensity images based on visual interpretation. Then, all these road markings were segmented as separate training samples by employing a clustering method. Thus, each labeled image only contains one type of road marking. Finally, the 2D coordinates and class type of each road marking pixel were recorded. Since the number of some road markings in certain classes (e.g., different Chinese words) is limited, such classes were merged and accordingly augmented through rotation, translation, and scaling operations. As listed in Table 4.1, a total of seven categories of road markings were generated.

Table 4. 1 Category and quantity of labeled road markings.

Category	Quantity of labeled road markings	Quantity of labeled road markings after augmentation
Lane line	366	1, 000
Dashed line	423	1, 000
Zebra crossing	219	1, 000
Straight arrow	330	1, 000
Turn arrow	298	1, 000
Diamond	305	1, 000
Text	318	1, 000

Since the road marking types in these three road scenes (e.g., urban roads, highways, and underground garages) are similar, 7,000 samples were utilized to extract and classify road markings. The whole dataset was split into 60%, 20%, and 20% subsets for training, validation, and testing, respectively. According to prior knowledge and multiple experiments, different hyperparameters, i.e., the batch size, initial learning rate, and dropout rate, were fine-tuned in the training process for the optimal combination. Accordingly, the batch size, initial learning rate, dropout rate, and epochs were [8, 0.0001, 0.80, 400] for the U-shaped road marking extraction model, respectively, and [32, 0.0005, 0.80, 300] for the hybrid capsule-based road marking classification model. The proposed models were tested using TensorFlow 2.0 on Intel® i7-8700K CPU @3.70 GHz, Nvidia® 1080-Ti GPU, and 32 GB RAM.

4.4 Results and Discussion

4.4.1 Road Marking Extraction Results

In this study, according to the manually labeled reference data, the following three evaluation metrics, i.e., precision, recall, and F$_1$-score (Powers, 2011), were adopted to conduct the quantitative performance evaluation of road marking extraction:

$$\text{Precision} = \frac{T_P}{T_P+F_P} \tag{4.9}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{4.10}$$

$$F_1\text{-}score = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{4.11}$$

where T_P, F_P, and F_N present true positive, false positive, and false negative segmentation outputs, respectively. Specifically, the precision shows the percentage that the extracted road markings are valid, while the recall represents the completeness of the extracted road markings. Moreover, F_1-score is a weighted average score of by analyzing both precision and recall.

The U-shaped convolution-deconvolution capsule architecture was proposed for road marking extraction on the generated intensity image patches with 4 cm resolutions. Figure 4.2 presents the fine-tuned network configurations based on multiple experiments, which details the number of capsules and the sizes of feature maps after standard or capsule-based convolutional operations. The training samples captured in urban roads, highways, and underground garages were used to evaluate the extraction performance. A series of experiments were performed to determine the optimal parameters, such as the overlapping size p_s between two adjacent image patches. In fact, an increasing overlapping size can produce not only better classification performance but also more image patches resulting in slow training speed. Therefore, to balance the classification performance and computational burden, p_s was defined as 512.

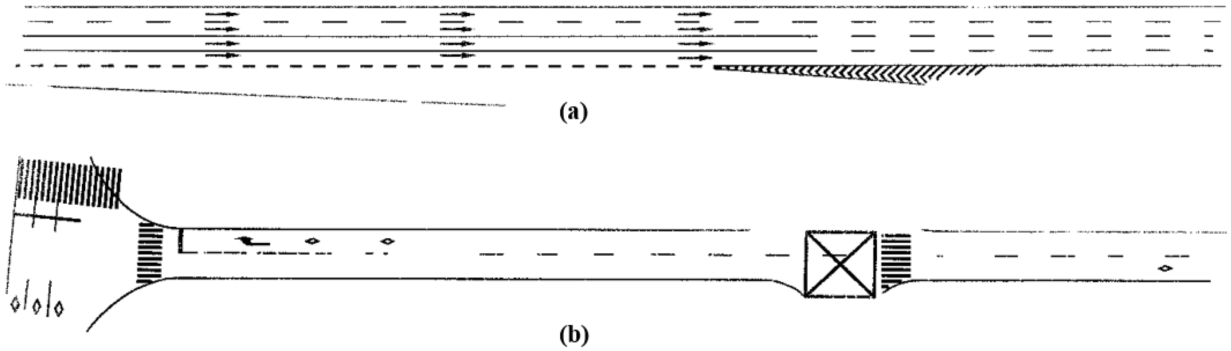


Figure 4. 4 Road marking extraction results using the proposed U-shaped capsule network. (a) Highway scene, and (b) urban road scene.

Figure 4.4(a) indicates the road marking extraction results in a highway scene. These highways are newly built and well maintained with clear road markings (i.e., lane lines and dashed lines), which enables the vectorial capsules to effectively learn inherent road marking features (e.g., varying intensities) at the training stage. Although the binary classification was conducted to minimize the influence of low intensity contrast between road surfaces and road markings, some pixels belonging to the lane lines were misclassified into road surfaces. Compared to highway road scenes, road markings painted on urban road surfaces are usually worn, which leads to dilemmas and uncertainties for high-accuracy road marking extraction. As shown in Figure 4.4(b), most road markings were successfully extracted, which demonstrates the proposed network in this study can effectively extract road markings even in complex urban road scenes. However, due to heavy traffic flow and pavement corrosion, some road markings are heavily worn and incomplete, which brings in enormous difficulties for road marking extraction. Besides, some road markings are covered by thick dust due to late maintenance, thus leading to varying intensities and low contrasts with the surrounding pavements. Although these conditions inevitably occurred in urban road scenes rather than highways, the proposed U-shaped capsule network could achieve reliable performance and deliver promising results for road marking extraction under various road conditions.

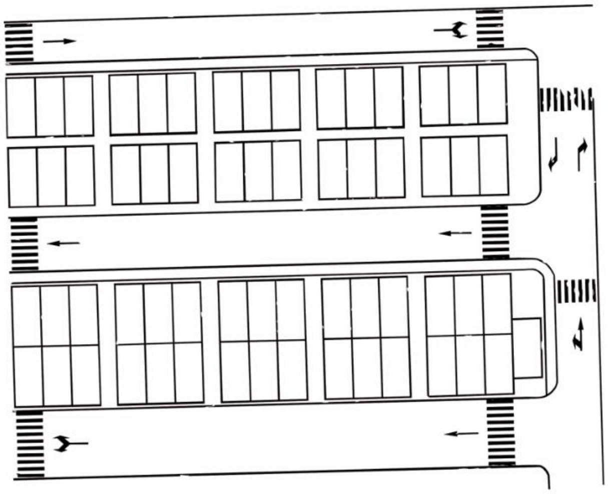


Figure 4. 5 Road marking extraction results in an underground garage scene.

Additionally, Figure 4.5 illustrates the road marking extraction results by utilizing the proposed networks in a 50 × 40 m^2 underground garage scene. Although the point density of these underground garage data is lower than point clouds obtained from the MLS systems, it can be solved by converting 3D point clouds into 2D images. The intensity value of a certain pixel is calculated based on its surrounding neighbours. Moreover, capsule convolutions encode not only intensity contrast but pose (e.g., shape and position) information. Consequently, the majority of road markings were accurately detected and extracted, while only a few pixels belonging to arrows and zebra crossings were misclassified into road surfaces due to inevitable occlusions and poor illuminations.

Table 4. 2 Accuracy assessment of road marking extraction in urban roads, highways, and underground garages.

Test dataset	Precision (%)	Recall (%)	F_1-score (%)
Urban scene			
1	94.97	90.16	92.50
2	94.76	89.31	91.95
3	95.03	91.28	93.12
Average	94.92	90.25	92.52
Highway scene			
1	97.08	92.03	94.49
2	95.46	91.62	93.50
3	95.89	89.74	92.71
Average	96.14	91.13	93.57
Underground scene			
1	91.87	89.93	90.89
2	92.06	90.02	91.03
3	92.84	90.55	91.68
Average	91.26	90.17	91.20

Table 4.2 indicates the quantitative accuracy assessment of the road marking extraction results in three different road scenes. Consequently, the proposed U-shaped capsule network delivered an average precision, recall, and F1-score of 94.92%, 90.25%, 92.52% in urban road scenes, and 96.14%, 91.13%, and 93.57% in highways, and 91.26%, 90.17%, and 91.20% in underground garages, respectively. Because of the high point density, few occlusions, and good illumination conditions in highways, the proposed U-shaped capsule network achieved more superior performance for road marking extraction in highway scenes than both urban roads and underground garages. The issue of intensity variation is considerably solved by learning the patches at different locations. The unavoidable errors occurring at the stage of manually annotated label generation could bring in challenges for robust and effective road marking extraction. Furthermore, some road markings are worn and incomplete, resulting in the sizes of such road markings smaller than the manually labeled reference data. Therefore, the road marking extraction performance of the developed model is underestimated in the experimental results. The multiple experiments demonstrated that the proposed U-shaped capsule-based model is able to learn

inherent features (e.g., intensity and shape) for road marking extraction by using different kinds of point clouds. Such data are obtained from complex road scenes, with low point densities, in poor illumination conditions, and with uneven intensity distributions.

4.4.2 Comparative Study for Road Marking Extraction

A comparative study was conducted to evaluate the road marking extraction performance by using the developed models and existing algorithms, including Cheng et al. (2016), Ma et al. (2019a), and Wen et al. (2019b). The test datasets were collected from urban roads, highways, and underground garages with low-intensity contrast and incomplete point clouds, which contain many categories of road markings (e.g., lines, arrows, and texts). These three methods were re-implemented on the testing datasets. Figure 4.6 shows the road markings on the large-scale roadways extracted using four methods.

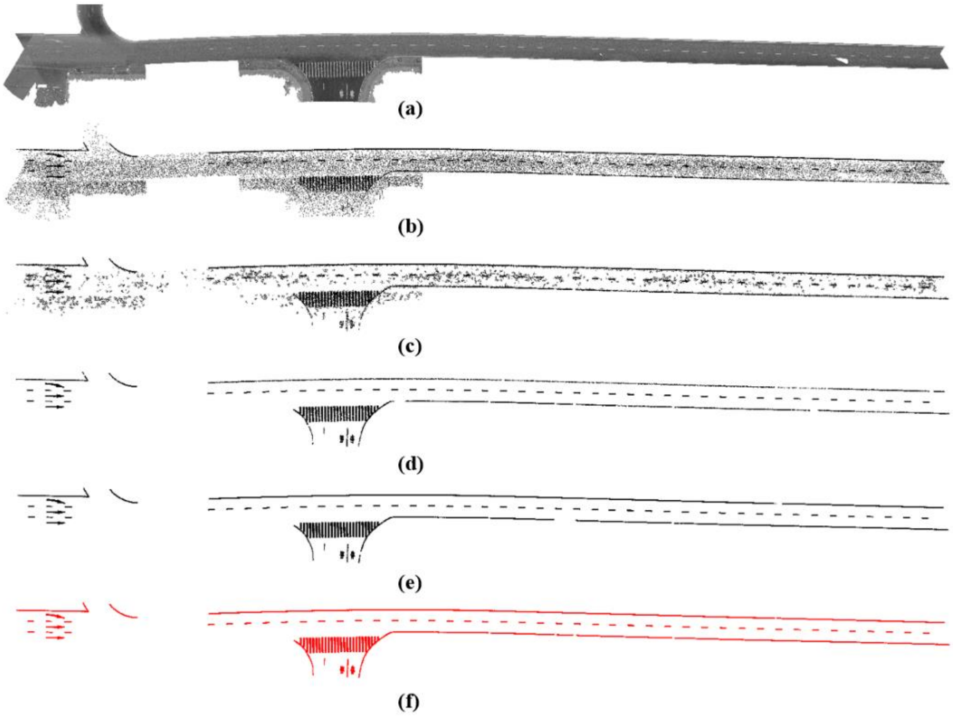


Figure 4. 6 Road marking extraction results using different methods. (a) Raw road surface, (b) Cheng et al. (2016), (c) Ma et al. (2019a), (d) Wen et al. (2019b), (e) the proposed method in this study, and (f) manually labeled reference data.

Accordingly, mobile LiDAR point clouds were transformed from 3D point clouds into 2D georeferenced images at the stage of road marking extraction in both Cheng's (2016) and Wen's (2019b) methods. Cheng's method (2016) employed Otsu's threshold approach (Otsu, 1979) for road marking segmentation according to the discriminant analysis, which requires the generated intensity image should be bimodal with uniform illumination conditions. Thus, it is difficult to completely extract road markings from low-intensity contrast point clouds, especially from underground garage data with poor illuminations. Meanwhile, by projecting 3D point clouds onto a horizontal xy-plane, Wen's method (2019b) performed a revised U-Net neural network to segment different types of road markings. However, the Softmax activation function used at the stage of road marking extraction cannot capture intra-class compactness, thereby resulting in limitations in road marking extraction from low-intensity contrast point clouds. Additionally, Ma's method (2019) mainly concentrated on determining adaptive intensity thresholds on local scales for road marking extraction. Nevertheless, it is quite difficult to define suitable threshold values in different road scenes.

Table 4. 3 Road marking extraction results by using different methods.

Method	Road scene	Precision (%)	Recall (%)	F_1-score (%)
Cheng et al. (2016)	Urban	27.35	33.82	30.24
	Highway	30.57	34.10	32.24
	Underground garage	24.52	29.03	26.59
	Average	27.48	32.32	29.69
Ma et al. (2019a)	Urban	62.63	53.19	57.53
	Highway	70.13	65.54	67.76
	Underground garage	68.73	59.42	63.74
	Average	67.16	59.38	63.01
Wen et al. (2019b)	Urban	92.15	89.33	90.72
	Highway	95.97	87.52	91.55
	Underground garage	91.95	90.07	91.00
	Average	93.36	88.97	91.09
The proposed method	Urban	94.92	90.25	92.52
	Highway	96.14	91.13	93.57
	Underground garage	91.26	90.17	91.20
	Average	94.11	90.52	92.43

Table 4.3 shows the overall performance of different methods for road marking extraction by using precision, recall, and F_1-score evaluation metrics. Cheng's method (2016), Ma's method (2019), and Wen's method (2019b) achieved an average of precision, recall, and F_1-score of 27.48%, 32.32%, and 29.69%, 67.16%, 59.38%, and 63.01%, and 93.36%, 88.97%, and 91.09%,

respectively; while the proposed method in this study delivered an average of precision, recall, and F_1-score of 94.11%, 90.52%, and 92.43%, respectively. As can be seen, the road markings were extracted incompletely or even lost information by using such three comparative methods. In contrast, the proposed U-shaped capsule network is capable of achieving better performance with higher accuracy and less noise in all road scenes.

Moreover, the capsule-based convolutional operations in the proposed model can not only capture the salient features embedded in intensity values but also the shape and position information of the road markings, which makes the proposed model outperform than other methods in terms of correctness and completeness. However, some road markings were not correctly segmented from the generated intensity image patches because of the occlusions of other road users (e.g., vehicles and cyclists) during the data acquisition of MLS systems. Additionally, uneven intensity distribution and varying illumination conditions from different road scenes also make effective and accurate road marking extraction challenging. On the whole, the proposed U-shaped capsule network designs a promising solution for road marking extraction from massive and unstructured 3D MLS point clouds.

4.4.3 Road Marking Classification Results

The experimental results of the hybrid capsule-based road marking classification neural network were evaluated based on the misclassification rate (MCR), which is calculated by:

$$MCR = \frac{\sum_{i=1}^{N} T_i}{N} \tag{4.12}$$

where N is the total number of road marking pixels. Specifically, T_i = 0, if the road marking is correctly classified. Otherwise, T_i = 1. The proposed road marking classification method was evaluated in urban roads, highways, and underground garages.

Accurate and robust road marking classification is essential for fully autonomous driving to design efficient navigation paths and avoid accidents in changing road conditions. Based on the proposed hybrid capsule-based road marking classification method, the extracted road markings were further classified into seven categories, i.e., lane line, dashed line, zebra crossing, straight arrow, turn arrow, diamond, and text. Figure 4.3 details the optimal network configurations through computational complexity and classification accuracy analysis. In the revised dynamic routing algorithm, the number of routing iterations, namely r, plays a significant role to balance between classification performance and computational complexity. Accordingly, multiple experiments were carried out to verify the robustness and convergence of the revised dynamic routing algorithm with different iterations. In fact, more routing iterations usually strengthen the classification performance but results in overfitting problems.

Additionally, to emphasize intra-class compactness and overcome inter-class imbalance, the L-Softmax was adopted to guide weight updates in the hybrid capsule-based classification architecture. By calculating different MCRs after 800 epochs from an urban road scene, Figure 4.7 illustrates the classification performance of different loss functions (i.e., standard Softmax loss and L-Softmax loss) with varying routing iterations. Intuitively, the standard Softmax loss and L-Softmax loss deliver minimal MCRs of 3.67% and 2.33%, respectively, by defining the routing iterations r as 4. Consequently, the L-Softmax loss function can achieve a lower misclassification rate compared with the standard Softmax loss, which demonstrates the L-Softmax loss function can significantly boost the capsule-based classification network performance by using mobile LiDAR point clouds.

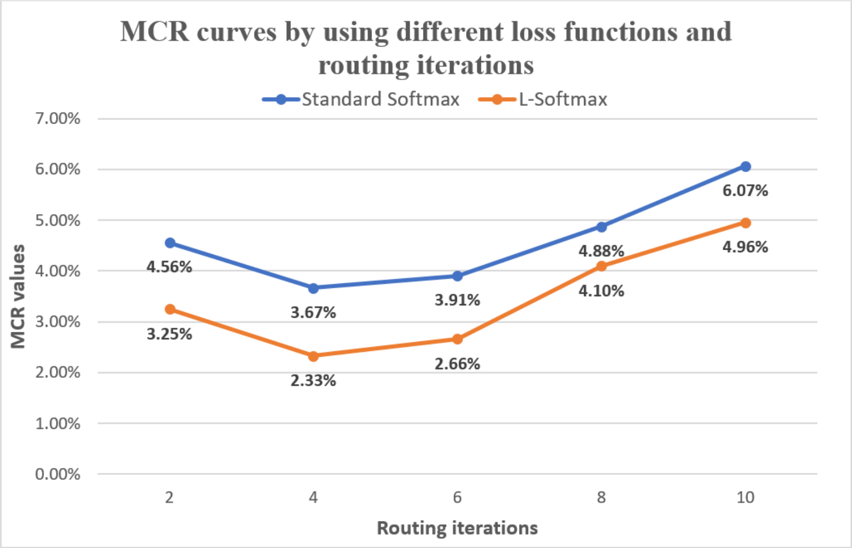


Figure 4. 7 Road marking classification results by using Softmax loss and L-Softmax loss with different routing iterations.

Table 4. 4 Misclassification rates of the proposed classification network in different road scenes.

Evaluation metric	Road scene			Average
	Urban	Highway	Underground garage	
MCR	2.16%	4.87%	3.23%	3.42%

Table 4.4 shows that the proposed hybrid capsule-based network can achieve an average of 3.42% MCR, which demonstrates that most road markings were correctly classified in three road scenes. Figure 4.8 shows the road marking classification results from a complex urban road environment. This scenario is a typical urban road that consists of zebra crossings, lane lines, diamonds, and texts, etc. Various colors denote different road marking types, while the misclassified markings are identified with black boxes. As can be seen, most road markings were correctly classified with a 2.16% MCR. Some lane lines are broken due to moving overloaded trucks and late road maintenance, resulting in broken lane lines similar to dashed lines. Therefore, such lane lines were inaccurately grouped into dashed lines. Additionally, some zebra crossings and straight arrows were misclassified as lane lines due to the erroneous results obtained in the process of road marking extraction. Similarly, Figure 4.9 indicates the classification results from highway point cloud data with a 4.87% MCR. Intuitively, the proposed model is capable of correctly classifying most road markings. However, some lane lines were incorrectly identified as dashed lines, resulting from the incomplete extraction of road markings. Additionally, the shapes of some broken lane lines are very similar to straight arrows, which also leads to false classification results.

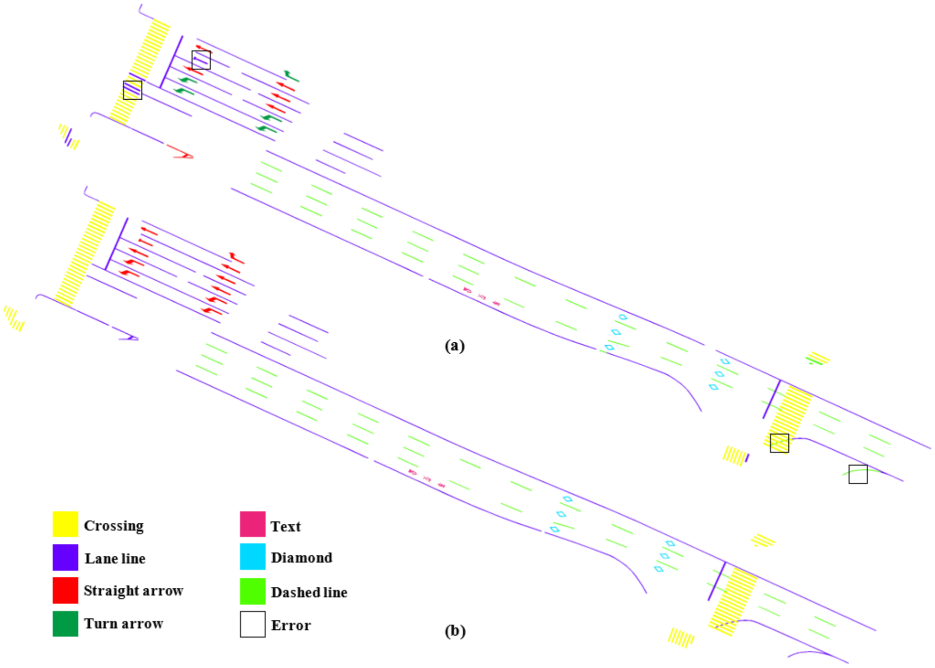


Figure 4. 8 Road marking classification results from an urban road scene. (a) Classification results, and (b) manually labeled reference data.

Moreover, the road marking classification performance of the proposed hybrid capsule-based network was evaluated in underground garage scenes. As shown in Figure 4.10, the proposed classification method can deliver satisfactory classification results from low-intensity contrast point clouds in poor illumination and GNSS-signal denied environments. However, some lane lines were misclassified into dashed lines, because these lane lines were not thoroughly extracted at the stage of road marking extraction. The hybrid capsule road marking classification model regarded them as independent dashed lines and incorrectly trained. Besides, a turn arrow was incorrectly identified as straight arrows since the shape of this turn arrow looks much like straight arrows. The MCR of road marking classification in underground garage environments is 3.23%.

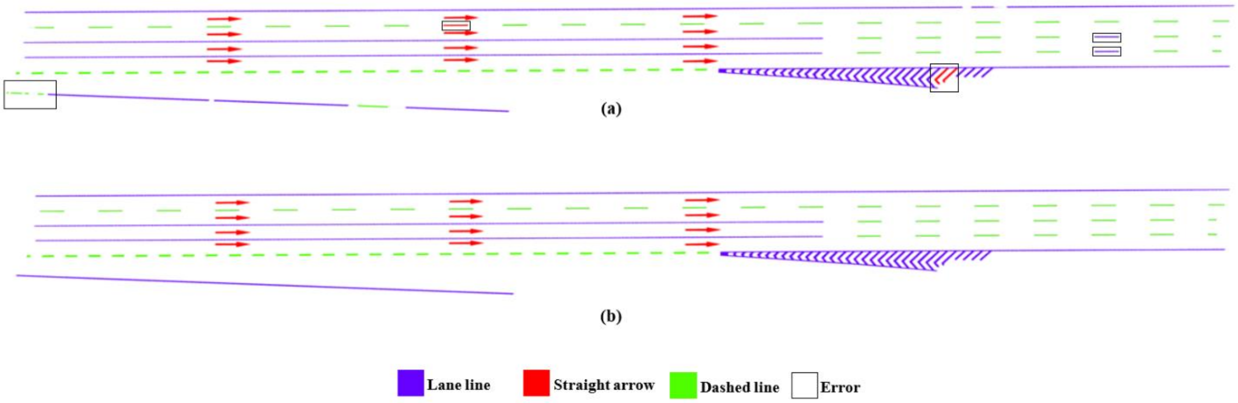


Figure 4. 9 Road marking classification results from a highway road scene. (a) Classification results, and (b) manually labeled reference data.

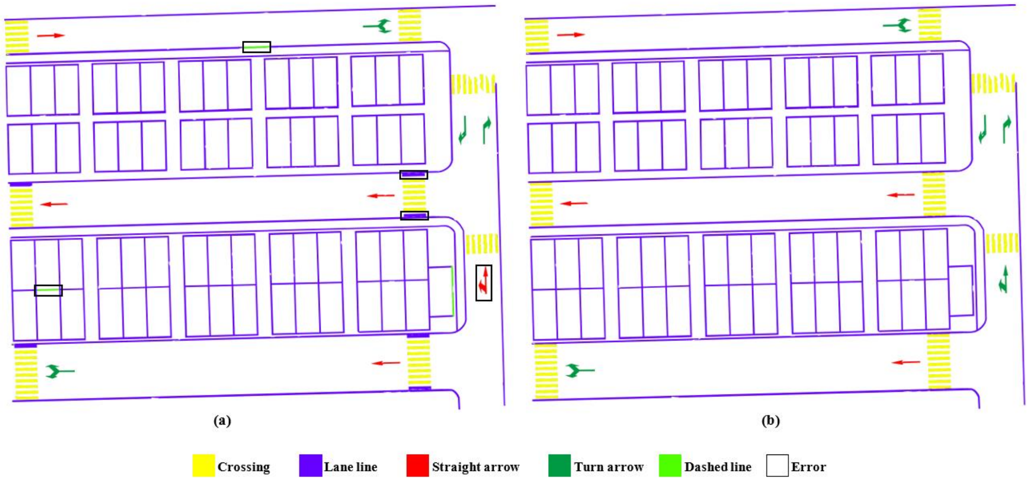


Figure 4. 10 Road marking classification results from an underground garage scene. (a) Classification results, and (b) manually labeled reference data.

To further demonstrate the effectiveness and robustness of the proposed models in this study, the road marking extraction and classification performance was further evaluated by using low-quality point cloud data. Accordingly, Figures 4.11(a)-(c) presents the road surface with low-intensity contrast between road markings and their surrounding environments, the generated intensity image with diverse point densities, and the road surface with worn and incomplete road markings, respectively. Figures 4.11(d)-(f) indicates the corresponding road marking extraction and classification results, respectively. As can be perceived, the proposed capsule-based deep learning networks are capable of effectively extracting and classifying road markings from low-quality input data. On the whole, the proposed capsule-based networks can deliver accurate road marking extraction and classification results on complex road environments, which provides a promising solution in fully autonomous driving and HD map creation.

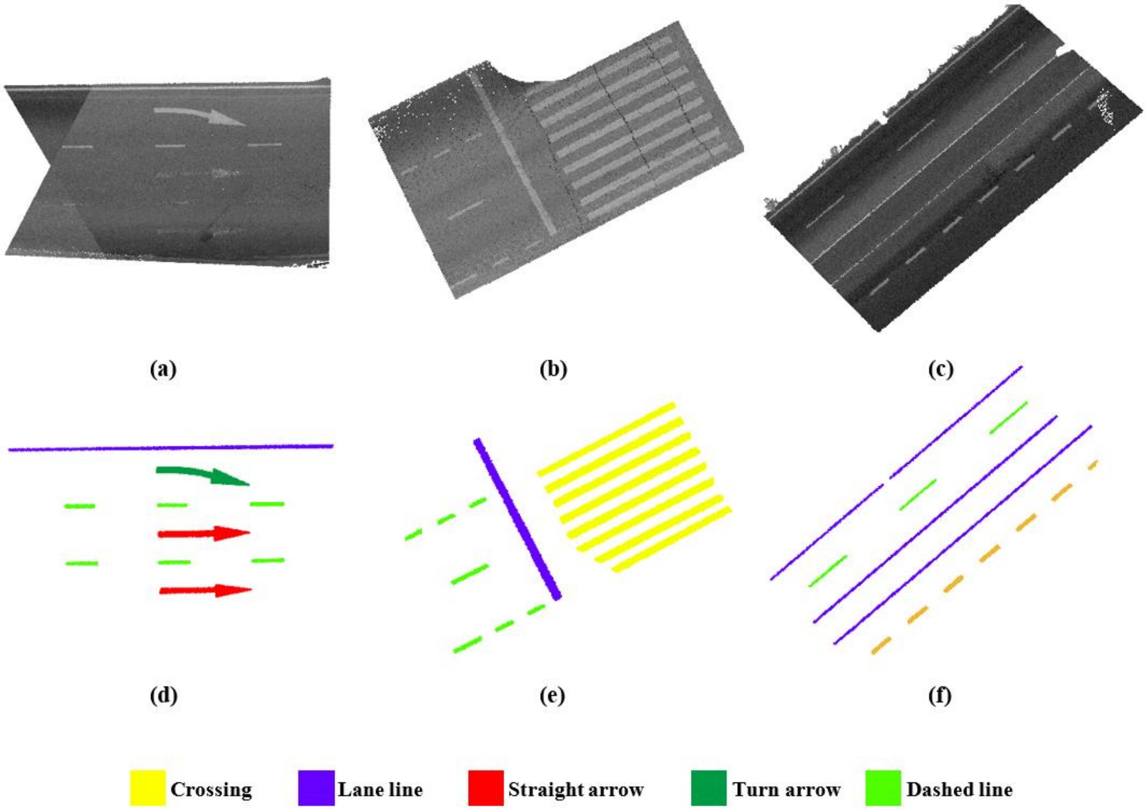


Figure 4. 11 Road marking extraction and classification results on low-quality data. (a) road surface with low-intensity contrast between road markings and their surrounding environments, (b) generated intensity image with varied point densities, (c) road surface with worn and incomplete road markings, and (d)-(f) are corresponding road marking extraction and classification results.

4.5 Computational Efficiency Evaluation

In this study, the proposed framework mainly contains three modules: data-preprocessing, U-shaped capsule network for road marking extraction, and hybrid capsule network for road marking classification. Table 4.5 lists the computational cost for each module, as well as the average time complexity across all over three road scenes. The average processing time of data-preprocessing, road marking extraction, and road marking classification are 36.47s, 3.07s, 2.58s, respectively. In fact, most of the processing time is spent in the data-preprocessing phase. Accordingly, multiple threads and GPU parallel computing techniques can be further applied to not only boost the computational efficiency in the process of 3D point cloud projection but dramatically accelerate capsule-based networks.

Table 4. 5 Computational efficiency of the proposed methods in different road scenes.

Processing module	Road scene			Average
	Urban	Highway	Underground garage	
Data processing (s)	40.25	31.36	37.80	36.47
Road marking extraction (s)	3.37	2.74	3.11	3.07
Road marking classification (s)	2.77	2.35	2.63	2.58

4.6 Chapter Summary

This paper handles the dilemmas related to threshold-based methods for road marking extraction and classification. Such dilemmas result in robustness reduction and computational complexity when dealing with 3D unstructured and high-density point clouds captured by MLS systems, most remarkably due to its varying point density and intensity, as well as low-intensity contrast between road markings and their neighbouring pavements. In this paper, two novel capsule-based network architectures are designed for road marking extraction and classification, respectively, from highly dense MLS point clouds with an irregular data format. Moreover, a road marking dataset containing both 3D point clouds collected by both MLS and BLS systems and manually labeled reference data is created from three types of road environments, including urban roads, highways, and underground garages, while the proposed models were accordingly evaluated by estimating robustness and efficiency using this self-built dataset.

In the extraction process, a U-shaped capsule-based network was designed to extract road markings using 2D georeferenced intensity images. The experimental results demonstrated that the proposed extraction model is capable of effectively encoding high-level features (e.g., changing intensity and pose information) with significantly enhanced road marking extraction performance. The comparative study indicated that the developed method achieved better performance than other threshold-based and U-Net based methods, while delivered an average of precision, recall, and F_1-score of 94.11%, 90.52%, and 92.43%, respectively, in three different road scenes.

In the classification process, a hybrid capsule-based network was proposed to classify seven types of road markings. Compared to those manually defined rule-based classification methods, the proposed method can automatically learn more salient features embedded in intensity values, as well as the shape information of the road markings by using the revised dynamic routing algorithm and powerful L-Softmax loss function. The quantitative evaluation indicated that the hybrid capsule-based network achieved an average of 3.42% MCR in changing road environments.

In conclusion, the multiple experimental results have demonstrated that capsule-based networks are capable of effectively extracting inherent features from massive MLS point clouds and achieving superior performance in road marking extraction and classification tasks. For further research, the feasibility of a point-wise end-to-end deep learning framework should be investigated for robust and effective road marking extraction and classification purposes.

Chapter 5

BoundaryNet: Extraction and Completion of Road Boundaries with Deep Learning

This chapter presents a novel deep learning framework, named BoundaryNet, to extract and complete road boundaries by using both MLS point clouds and high-resolution satellite imagery. In this network, first, road boundaries are extracted by conducting a curb-based extraction method. Such extracted 3D road boundary lines are used as inputs to feed into a U-shaped network for erroneous boundary denoising. Then, a CNN model is proposed to complete the road boundaries. Next, to achieve more complete and accurate road boundaries, a conditional deep convolutional generative adversarial network (c-DCGAN) with the assistance of road centerlines extracted from satellite images is developed. Finally, according to the completed road boundaries, the inherent road geometries are calculated. The experimental results indicate that the BoundaryNet model can provide a promising solution for road boundary completion and road geometry estimation. © [2020] IEEE. Reprinted, with permission, from [Ma L, Li Y, *Li J, Junior J, Gonçalves W, Chapman M. A., 2020. BoundaryNet: Extraction and completion of road boundaries with deep learning using MLS point clouds and satellite imagery. *IEEE Trans. Intell. Transp. Syst*. (under minor revision)].

5.1 Introduction

Urban roads, as one of the essential public infrastructures, provide significant motivations for rapid urban sprawl and create notable economic and social benefits (Wang et al., 2012). Moreover, detailed road inventories are commonly applied to support extensive applications, such as city planning, construction surveying, smart cities, and advanced driver-assistance systems (ADAS) (Pu et al., 2011). Mobile laser scanning (MLS) systems that comprise Light Detection and Ranging (LiDAR) sensors can capture high-density 3D point clouds with mm-level accuracy in large-scale urban environments (Ma et al., 2018). Such point clouds have been widely used for many transportation-related studies, including pavement inspection (Ye et al., 2019), road marking classification (Rastiveis et al., 2020), road object detection and segmentation (Li et al., 2019a), road geometry modeling (Pradhan and Sameen, 2020), and road boundary extraction (Wen et al., 2019). As a crucial component of road topological networks, road boundaries designate allowable driving zones and provide auxiliary road information to promote the development of high-definition (HD) maps and fully autonomous vehicles (AVs). Accordingly, road boundary extraction is generally implemented through extracting road surfaces, followed by road curb detection from MLS point clouds. Although remarkable improvement has been achieved, most of the off-the-shelf methods cannot accurately and completely extract road boundaries, due to occlusions and point density variations during raw MLS data acquisition (Soilán et al., 2019). Thus, in this paper, the theoretical and methodological problems of road boundary extraction and completion are investigated by using MLS point clouds and satellite imagery.

Many studies have been conducted to estimate and recover incomplete road boundaries (Holgado-Barco et al., 2017; Zai et al., 2017; Wen et al., 2019; Ma et al., 2019). One of the most straightforward ways is to collect point clouds multiple times by using MLS systems. However, it is considerably cost-intensive and labour-consuming. Moreover, some interpolation methods, such as linear interpolation, polynomial interpolation, and B-spline curves, cannot deliver the robust solutions for road boundary recovery, especially in complicated crossroads and curved road sections (Ma et al., 2019). Many researchers concentrated on extracting road surfaces or road centerlines from aerial and orbital remotely sensed imagery. In urban environments, roads extracted from satellite and aerial images are usually inaccurate and incomplete because of the complicated imaging conditions, various terrain factors, and partial occlusions caused by roadside trees (Zhang et al., 2019). Furthermore, different image resolutions have significant impacts on road boundary extraction. Specifically, images captured with low resolutions could result in fuzzy ground object features, while high-resolution images also bring in dilemmas and uncertainties regarding various influential factors (e.g., weather sensitivity and limited temporal resolutions).

Compared to camera-based mobile mapping systems, MLS systems are less sensitive to weather and ambient luminance conditions. The point density obtained from high-end MLS systems can reach up to 10,000 pts/m^2 at a wide moving speed range of 40-100 km/h in both urban and highway road scenes, while it poses enormous difficulties for both terrestrial and airborne laser scanning (TLS/ALS) systems to provide such high surveying adaptability and measuring precision (Yan et al., 2015). Roads extracted from ALS point clouds are usually broken lines because of the distortions, point density and intensity variations, and noisy outliers (Kumar et al., 2013; Hu et al., 2014). Moreover, sparse and unevenly distributed TLS point clouds make effective and robust road extraction quite challenging. Global Navigation Satellite System (GNSS) data, which provides spatial trajectory and the full coverage of roads, have been typically employed to extract road boundaries and centerlines (Huang et al., 2015). Nevertheless, both accuracy and completeness of extracted road boundaries using spatial trajectory data and crowd-sourced GNSS data (e.g., taxi GNSS data) are still unsatisfied because of the inevitable system positioning error and multipath effects in urban road scenarios.

Urban roads, segmented from various data sources, indicate various strengths and limitations. There are three key challenges to efficiently and robustly extract and complete road boundaries from MLS point clouds: (1) Road boundary data is incomplete due to occlusions caused by road participants (e.g., pedestrians and cyclists) and roadside infrastructures, or the limited scanning ranges from onboard sensors; (2) Some urban roads with low curbs or worn curbs also lead to incomplete road boundaries by using different extraction methods; (3) Besides, the varieties and uncertainties of missing parts in urban road boundaries bring significant difficulties to ascertain if these gaps should be completed or not. To overcome these challenges, the practicability of road boundary extraction and completion through embedding MLS point clouds with satellite images is investigated. More complete and correct road boundaries with a wealth of road information are obtained if an MLS point cloud related road boundary is generated with the assistance of satellite images. However, these data captured by different sensors at different times, in varying ambient illumination conditions, and with changing point densities, cause the effective data fusion and road boundary recovery challenging (Qin and Gruen, 2014).

Regarding previous road boundary extraction studies showed the potential of multi-sensor combination (Li et al., 2019b; Ravi et al., 2019). However, such methods have significantly improved the accuracy of road boundary extraction by introducing spectral or texture information, they still cannot deal with existing gaps in road boundaries. For road boundary completion, the existing methods have certain limitations (Wen et al., 2019). Considerable challenges still exist to provide promising solutions for road boundary completion, especially in complex curved road scenes. Therefore, it is increasingly necessary to introduce an effective and robust method for road boundary completion.

A novel deep learning based framework is proposed, named BoundaryNet, comprising the following four modules: (1) curb-based road boundary extraction, (2) CNN-based road boundary completion, (3) the D-LinkNet-based road centerline extraction, and (4) the conditional deep convolutional generative adversarial network (c-DCGAN)-based road boundary refinement. Figure 5.1 shows the detailed workflow of BoundaryNet. More specifically, 3D road boundaries are segmented from point clouds as the inputs of the whole framework. The extracted road boundaries that always contain many erroneous lines, are firstly removed using a revised U-shaped encoder-decoder neural network, according to the images projected from 3D point clouds of road boundaries. Next, a CNN-based downsampling and upsampling model enabling to capture more distinctive features of line segments, such as curvature and connectivity, is developed to identify and restore the missing parts. Then, because of the imperfect local details of road boundaries resulting from the CNN-based completion model, such road boundaries are further refined using a c-DCGAN model, with the assistance of the road centerlines obtained from high-resolution satellite images. Finally, according to the completed road boundaries, inherent road geometries, including both horizontal and vertical road alignment parameters, are thus estimated to support road maintenance and traffic safety. Different from 3D point-based methods that suffer from incomplete data collection and point density and intensity variations, a deep learning-based framework combining both MLS point clouds and satellite images is developed, which can more robustly extract and complete road boundaries, and accurately estimates the road characteristics in large-scale urban environments.

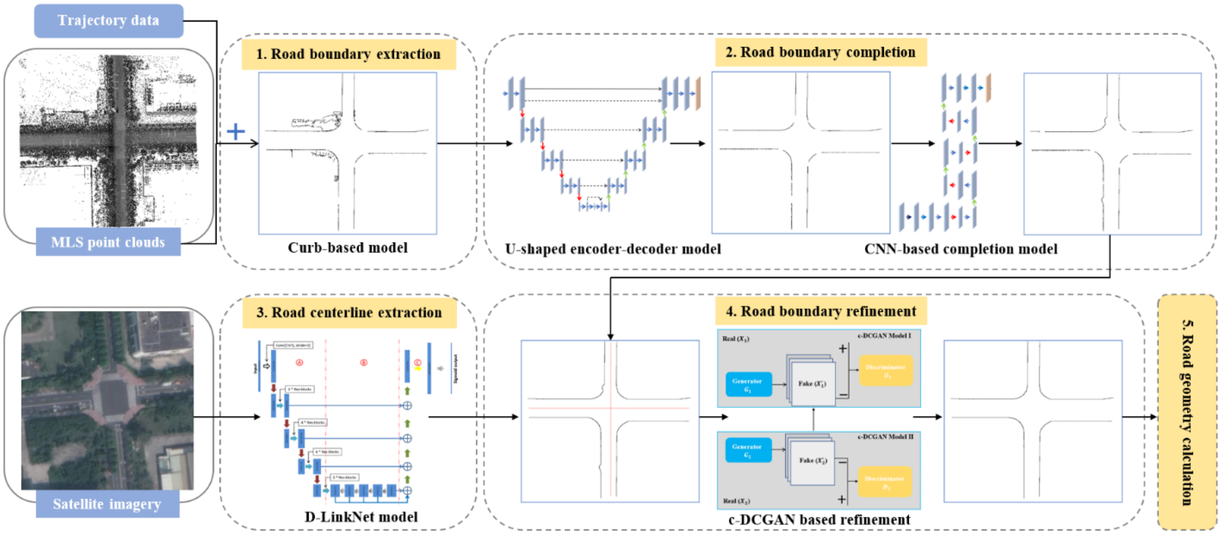


Figure 5. 1 Workflow of the proposed BoundaryNet model.

5.2 Algorithm Description

5.2.1 Road Boundary Extraction

According to the previous work (Ma et al., 2019a), a revised curb-based road boundary extraction method was proposed by using a data slicing structure from 3D point clouds. Firstly, according to the trajectory of the MLS system, the raw mobile LiDAR data were horizontally divided as a series of point cloud blocks, in which corresponding data profiles were created at predefined widths. Then, the point clouds in these data profiles were projected onto the plane that is vertical to the moving direction of the MLS system. Such data profiles were gridded to generate pseudo scan lines, and principal points were accordingly ascertained in grid cells. Next, road curb points were segmented from pseudo scan lines by taking both slope and elevation differences into consideration. Finally, a B-spline interpolation algorithm was used to fit the extracted road curb points into continuous boundary lines. Therefore, this curb-based method is adopted for road boundary extraction.

For complex urban road environments, the extracted road boundaries have multiple objects (e.g., road markings, cracks, and roadside objects) and erroneous lines with occlusions from road users, which brings in considerable difficulties in road boundary extraction. Due to varying road design standards across different regions and the morphological irregularity of erroneous lines, it is quite challenging to completely and effectively extract road boundaries from massive and unstructured 3D point clouds by using the existing rule-based methods (Kumar et al., 2014, Soilán et al., 2017, Jung et al., 2019).

Based on the symmetric encoder-decoder framework and skip connection operations, the U-Net model (Ronneberger et al., 2015) has demonstrated that it enables the delivery of promising solutions for biomedical image segmentation. Herein, a revised U-shaped encoder-decoder deep learning framework is introduced that learns inherent features to separate road boundaries from the erroneous lines (see Figure 5.1). To this end, the extracted 3D road boundaries are first transformed into a 2D image in a horizontal XY plane with a pixel size $S_{\epsilon 1}$. Thus, the complicated erroneous line removal problem is perceived as a straightforward binary image classification task. That is, road boundaries are regarded as foreground and other lines as background. The proposed U-shaped neural network contains encoder and decoder sections. Each encoder layer performs 2D convolutions with a kernel size of 3×3 for spatial feature encodings, while each decoder layer conducts 2D deconvolution (2×2 kernel size) and convolution (3×3 kernel size) operations for segmentation feature map construction. As shown in Figure 5.2, instead of only using skip connection in the U-Net model, the pooling indices connection is employed to feed max-pooling indices (2×2 filter size) derived from the corresponding encoders into decoders for non-linear upsampling operations. Such max-pooling indices can significantly decrease the number of parameters facilitating end-to-end training. To reduce the over-fitting problem, the dropout operation is therefore carried out. The binary cross-entropy (BCE) function is employed to guide the model refinement, which is calculated by:

$$\mathrm{L}(y, \hat{y}) = \frac{1}{N}\sum_{\mathrm{i}=1}^{N}(y \times \log \hat{y}_i + (1-y) \times \log(1-\hat{y}_i)) \tag{5.1}$$

where N indicates the total number of pixels in input images, y is the real value, and $\hat{y}_i$ represents the predicted value. The rasterized 2D images, as training data, are augmented through crop, rotation, scaling, and lighting condition changes, and then resized to 512×512 pixels before feeding into the road boundary denoising network. The training samples are generated by selecting the segmented road boundary pixels as inputs, such pixels in 2D images are manually labeled. In the training stage, the initial learning rate, batch size, dropout rate, and epoch are set as 0.0001, 4, 0.5, and 300, respectively.

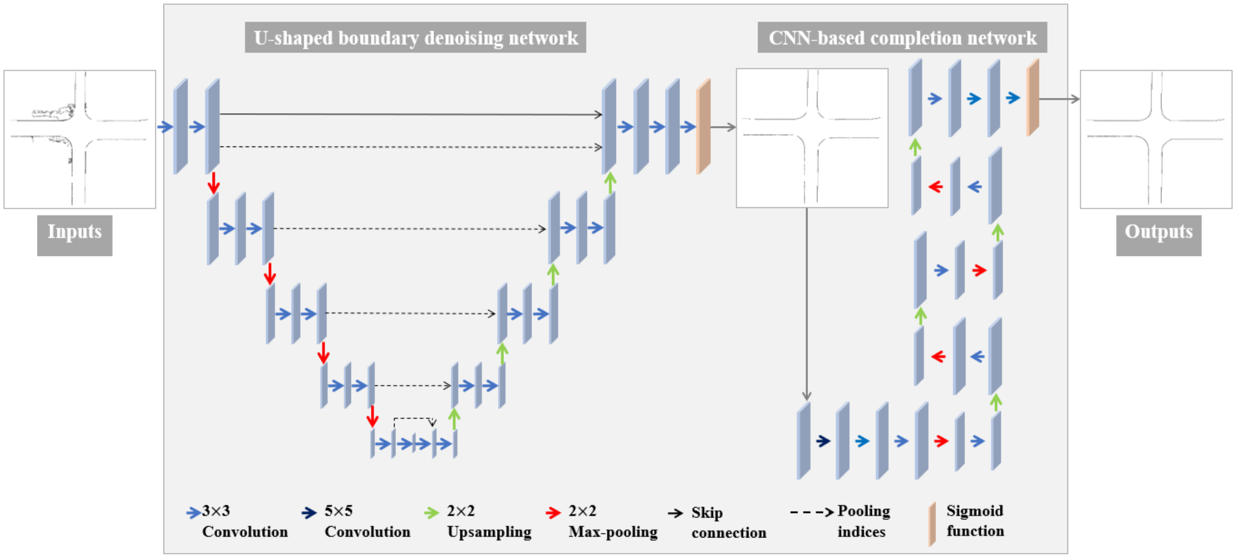


Figure 5. 2 Structures of the U-shaped network and the CNN-based completion network.

5.2.2 CNN-based Network for Road Boundary Completion

Therefore, road boundaries are successfully extracted from MLS point clouds by using the curb-based road boundary extraction method, followed by the U-shaped road boundary denoising neural network. Still, the gaps in extracted road boundaries with varying structures (e.g., lengths and curvatures) cause considerable dilemmas in direct and robust road boundary completion. To date, many researchers are dedicating themselves to image recovery and inpainting based on deep learning-based methods (Isola et al., 2017; Yu et al., 2019; Nazeri et al., 2019), which have achieved high performance. Sasaki et al. (2018) presented an end-to-end deep learning framework to detect gaps in deteriorated line drawings and recover them accordingly. This method has indicated superior performance in line drawing restoration and curvature and thickness conservation. Inspired by line inpainting, a novel CNN-based model is proposed to complete road boundaries in 2D space.

As shown in Figure 5.2, a CNN-based downsampling and upsampling model is developed to identify and fill the missing parts based on the road boundary extraction results obtained after the erroneous line removal. This model contains convolution, max pooling, and upsampling layers. More specifically, all the convolution layers employ a 3×3 kernel size, aside from the first layer, conducting a 5×5 kernel size. Moreover, all convolution layers utilize the Rectified Linear Unit (ReLU) as activation functions, aside from the last one, where the Sigmoid function is employed instead. In the training stage, batch normalization is conducted after each convolution operation, except for the final one. The 2×2 max pooling layers, downsampling by 1/2 size of the feature maps, are utilized for feature encoding and structure recognition of road boundaries from a larger region. Instead of deconvolution layers, the nearest neighbour upsampling method is used to enlarge the image resolution of the outputs. Notably, fully connected (FC) layers are not employed in this process, the image sizes of inputs and outputs are not fixed, resulting in output images of the same size as the inputs. The model is trained by employing the mean squared error (MSE) as the cost function, which calculates the difference between input S and output S'_p as follows:

$$loss\left(S, \hat{S}\,\right) = \frac{1}{M} \sum_{p \in M} (S_p - S'_p\,)^2 \tag{5.2}$$

where M is the total number of pixels in the input image, and S_p and S'_p represent the values at pixel p in the input and the output images, respectively. Because of the limited number of training data, the same training dataset employed in the work of Sasaki et al. (2018) is used. Finally, to obtain smooth and continuous boundary lines, only the completed 2D pixels by the CNN-based completion model are utilized to convert and add back to 3D road boundary point clouds.

5.2.3 Road Boundary Refinement

Because of the complexities and uncertainties of urban roads, the experimental results obtained from the CNN-based completion method cannot fully complete the missing parts in images, particularly for large gaps. As shown in Figure 5.3, the dilemmas are concluded in the following aspects: (1) some large missing parts should be completed (see blue boxes), (2) regular gaps should not be completed (see red boxes), and (3) irregular road structures should be refined (see green boxes). Therefore, the road boundaries that are completed by using the above completion method should be further refined.

Rapid data collection, large area coverage, and detailed feature characteristics on the ground are strengths of remotely sensed imagery. Therefore, the high-resolution satellite images are employed to determine gaps, whether they should be further completed or not. Herein, satellite images are used to extract road centerlines, which represents urban road structures in some ways, for the road boundary refinement in case of complexities and uncertainties.

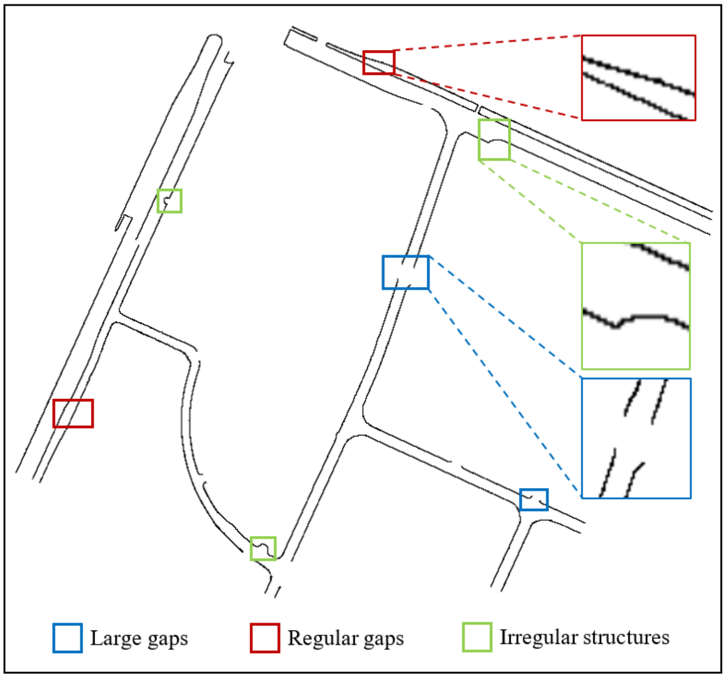


Figure 5. 3 Results of road boundary completion obtained from CNN-based network.

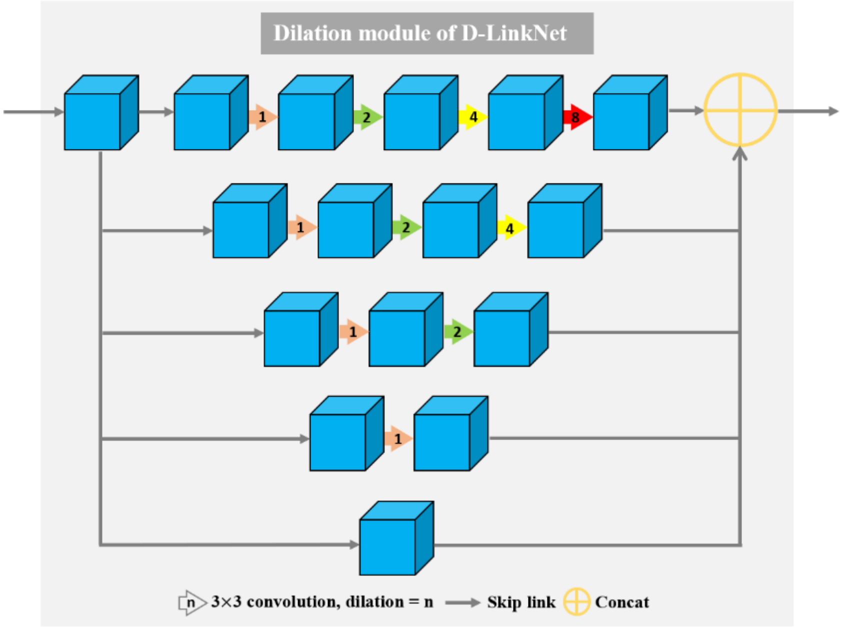


Figure 5. 4 Dilation module of D-LinkNet. It comprises dilated convolution in both parallel mode and cascade mode, and the receptive fields are different in different layers, enabling this model to combine features at various levels. From bottom to top, the receptive field is 1, 3, 7, 15, and 31, respectively.

The high-resolution satellite images are utilized to extract road centerlines that indicate a geometric topology of the road. Such centerlines are therefore used to handle the uncertainties of missing parts and guide the boundary refinement. To this end, the D-LinkNet method (Zhou et al., 2018a) is employed to segment roads from satellite imagery. This network comprises three modules, namely encoder, dilation part, and decoder. Specifically, the encoder module contains five downsampling layers that use ResNet34 (He et al., 2016) pretrained on the ImageNet dataset to learn shallow features and generate feature maps. The 2 × 2 max pooling layers are used to downsample the inputs from the size of 1024 × 1024 to 32 × 32. Then, as shown in Figure 5.4, the dilation module comprises dilated convolution in both parallel mode and cascade mode to increase the receptive field and preserve the detailed spatial information. All the dilated convolution layers employ a 3 × 3 kernel size, with a dilation rate of 1, 2, 4, 8 in the center, respectively. The transposed convolution layers are used to restore the feature maps to the resolution of 1024 × 1024. Each convolution layer uses ReLU as the activation function, except for the last layer, which uses the Sigmoid function. Adam is used as the optimizer. Moreover, this model applies BCE and dice coefficient as the loss function. BCE is calculated by using Eq. 5.1, and the dice coefficient loss is computed as follows:

$$Dice = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i + \sum_i^N g_i} \tag{5.3}$$

where N is the total number of pixels in the input image, and p_i and g_i indicate the values at pixel i in the predicted binary segmentation images and ground truth images, respectively.

To increase the number of training data, the DeepGlobe Road Extraction Dataset (Demir et al., 2018) is used as the training dataset. The testing images were acquired from Google Earth with a pixel size of 50 cm. During the training phase, the initial learning rate, batch size, and epoch are set as 0.0001, 2, and 150, respectively.

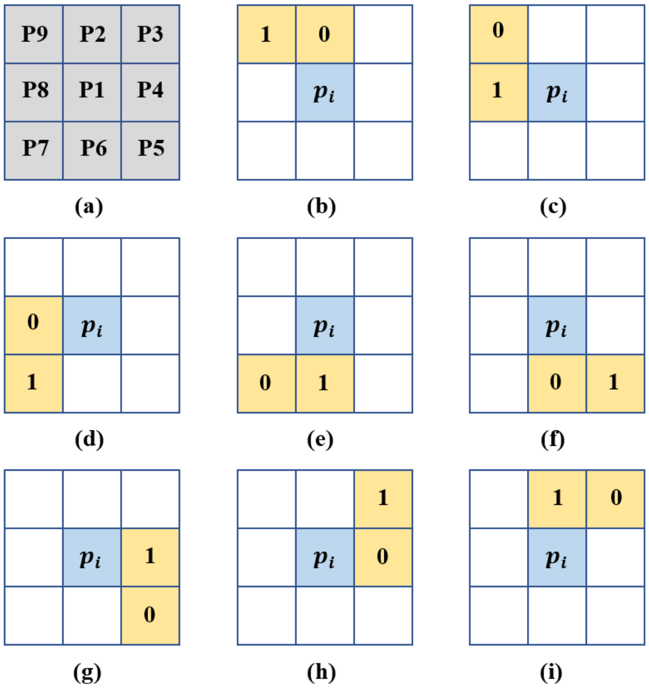


Figure 5. 5 Schema of the eight-connected morphological thinning algorithm. (a) A road pixel p_i and its eight neighbours. (b)-(i) indicate various discriminant conditions, respectively.

Based on the extracted road segments from satellite images, a morphological thinning algorithm (Shi et al., 2013) is employed to generate road centerlines. Given each road pixel p_1 and its eight-connected neighbours, i.e., p_2 to p_9 (see Figure 5.5(a)), the main procedure of this thinning algorithm is to traverse each pixel of the road segment images. In each pass, the road pixels p_i is removed if it meets the following criteria:

i. $2 \leq N(p_1) \leq 6$;

ii. $\delta(p_1) = 1$;

iii. $p_2 \times p_4 \times p_8 = 0$ or $\delta(p_2) \neq 1$;

iv. $p_2 \times p_6 \times p_8 = 0$ or $\delta(p_8) \neq 1$;

where $N(p_1)$ indicates the crossing number of road pixels among pixels p_2 to p_9, and $\delta(p_i)$ represents the discriminant condition, $i = 1$, 2, and 8, respectively. As shown in Figure 5.5(b), the following condition is examined: the upper-left neighbour and the upper neighbour of the pixel p_i is a road pixel (with a pixel value of 1) and an empty pixel (with a pixel value of 0), respectively. Meanwhile, Figures 5.5(c)-(i) illustrate different eight-connected discriminant conditions. Accordingly, $\delta(p_i)$ is set to 1, if only one condition is satisfied. Otherwise, $\delta(p_i)$ is equal to 0. Thus, based on the eight-connected schema, road centerlines are generated from the extracted road segments.

The strengths of this thinning algorithm are both straightforward and fast to conduct. However, the extracted road centerlines usually generate some spurs that decrease the correctness and smoothness of roads. Thus, the least-squares curve fitting (LSCF) algorithm is performed to obtain smooth and accurate road centerlines for irregular road networks. The least-square fitting algorithm takes the following form for each road segment:

$$min \sum_{i=1}^{N} y_i - (p_1 x_i^n + p_2 x_i^{n-1} + \cdots + p_n x_i + p_{n+1}) \tag{5.4}$$

where N denotes the total number of pixels in road segments, x_i and y_i are the row number and the column number of the pixel i, respectively. After solving Eq. 5.4, the estimation is afterward determined by:

$$\widehat{y_i} = round(p_1 x_i^n + p_2 x_i^{n-1} + \cdots + p_n x_i + p_{n+1}) \tag{5.5}$$

where $\widehat{y_i}$ denotes the column number of the pixel i after curve fitting.

Depending on the global coordinate system of satellite images, the generated road centerlines from these images are transformed to 3D point clouds by setting the height values to zero and then merged with road boundaries. Although some urban roads in satellite imagery are inevitably occluded by high-rise objects and roadside trees, the generated road centerlines can still refine the structure of road networks and provide significant guidance for road boundary refinement.

To tackle the complexities of missing parts and restore incomplete road boundaries, it is necessary to decide whether the restored gaps are refined with the assistance of the road centerlines. In this study, the road boundaries are restored to the complete road structure if the road centerline across the gaps. Therefore, the mistaken completions can be handled. Still, it is challenging to solve the dilemmas of incomplete gaps and irregular completion structures.

Generative adversarial network (GAN) models (Goodfellow et al., 2014) has indicated the extensive applications in image translation and image restoration domains. The conditional deep convolutional GAN (c-DCGAN) network is adopted, as an extended work of the DCGAN model, to deal with the issues of irregular completion structures (i.e., an image translation task) and incomplete gaps (i.e., an image restoration task). With the assistance of the road centerline, the c-DCGAN model is capable of restoring the incomplete road boundaries with detailed information, particularly for curved road sections, which makes the rule-based refinement methods challenging.

The DCGAN model consists of two adversarial modules, i.e., generator model G, and discriminator model D. Specifically, a generator G generates outputs, and a discriminator D distinguishes them as "real/fake" samples as much as possible. The adversarial competition between G and D is determined by:

$$\min_{D}\max_{G} L(D,G) = \mathbb{E}_{X\sim P_{data(x)}}[\log D(x)] + \mathbb{E}_{z\sim P_{z(z)}}[\log(1 - D(G(z)))] \qquad (5.6)$$

where the generator endeavors to maximize the loss value, i.e., $D_G^* = arg\max_{G} L(G, D_G^*)$, while the discriminator minimizes it, i.e., $G^* = arg\min_{D} L(G, D)$. In the process of training D, G operates in a feed-forward pattern without backpropagation, and vice versa to process G.

Compared to the standard GAN model, the DCGAN model is more stable to train, resulting in generators that produce reasonable outputs. Briefly, all convolutional net is performed by replacing deterministic spatial pooling operations (e.g., max pooling) with strided convolutions, enabling the model to encode inherent features in both spatial downsampling and upsampling processes. Moreover, batch normalization is adopted to solve the problems of poor initialization and improve gradient flows in deep hidden layers. Instead of ReLU, D uses Leaky ReLU as activation functions. G uses ReLU, except for the output layer, where a $Tanh$ function is used instead.

The c-DCGAN network extends the DCGAN model by generating feature maps from a random noise vector v and a condition c to an output y. For c-DCGAN, road boundary refinement is performed by transforming images with incomplete road boundary lines and centerlines, into refined images with ground-truth road boundary lines. Herein, the condition c defines images with incomplete road boundary lines and centerlines. v is a random noise vector. The result y indicates images with ground-truth road boundary lines. This c-DCGAN model can be easily applied to other road types by using a different condition c. Furthermore, only a small number of training samples are required to feed into the c-DCGAN model for road boundary refinement. Such training samples are: (1) cropped complete road boundaries and centerlines, (2) imperfect boundary lines (i.e., incomplete gaps and irregular completion structures) and centerlines, and (3) manually editing imperfect boundary lines and centerlines by hand drawings.

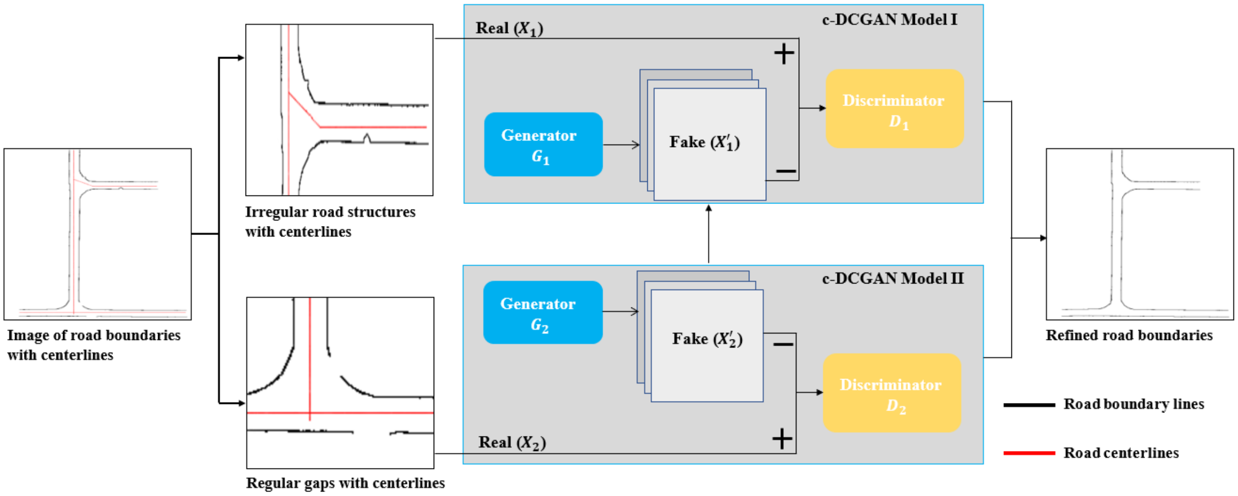


Figure 5. 6 Structure of the c-DCGAN road boundary refinement network.

Since the c-DCGAN model requires 2D images as the inputs, road boundary lines and centerlines in 3D point clouds are converted into 2D images with a pixel size $S_{\epsilon 2}$. The incomplete road boundaries are broken lines, which are different from the road boundaries with irregular completion structures. Thus, the different training data is manually grouped into two categories, and then separately fed into the different c-DCGAN models for refinement (see Figure 5.6). To obtain satisfactory completion results, the refinement results are fed into the model again to solve the problem of irregular completion structures. Finally, the road boundary refinement results are transformed back to 3D point clouds with complete structures.

5.3 Road Geometry Calculation

Geometric inventories of roads have remarkable impacts on road safety. Accurate road characteristics estimation, such as road width, curvature, and slope, is a vital strategy for minimizing traffic hazards and enhancing traffic efficiency. Therefore, the geometric inventory of complex urban roads (especially for curved road corridors) to estimate road characteristics contributes to road maintenance and traffic safety (Holgado-Barco et al., 2017). Based on the completed road boundaries obtained from the BoundaryNet model, some fundamental road geometries, including horizontal and vertical road alignment parameters, are calculated.

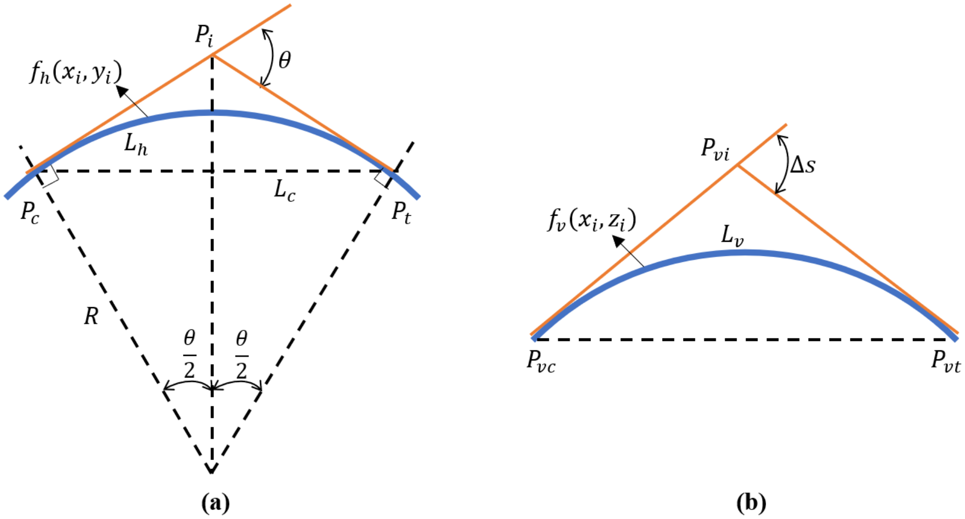


Figure 5. 7 (a) Elements of horizontal road geometry. (b) Elements of vertical road geometry.

Table 5. 1 Descriptions and equations for road geometry estimation.

Road geometry	**Descriptions**	**Equations**
Horizontal curve parameters	Horizontal curve	$f_h(x_i, y_i) = 0$
	Radius of curvature	$R = \frac{5729.58}{\theta}$
	Length of horizontal curve	$L_h = 0.0174533 \times \theta \times \mathrm{R}$
	Curvature of horizontal curve	$C = \left\lvert\frac{\pi-\theta}{L_h}\right\rvert$
Vertical curve parameters	Vertical curve	$f_v(x_i, z_i) = 0$
	Length of vertical curve	$L_v = \int f_v(x_i, z_i)ds$
	Flatness of vertical curve	$K = \frac{L_v}{\Delta s}$

As illustrated in Figure 5.7(a), the horizontal road geometry includes the length of horizontal curves, road width, the radius of curvature, the curvature of horizontal curves, and intersection angles (McCornmac et al., 2013). The length of a horizontal curve L_h denotes the arc length from the point of curvature P_c to the point of tangency P_t, while the radius of curvature R is the radius of the arc. The curvature of a horizontal curve C represents the average curvature value between P_c and P_t. The intersection angle θ is the interior angle at the intersection of the two tangents.

Meanwhile, vertical road geometry (see Figure 5.7(b)), including the length of vertical curves, slope changes, and the flatness of vertical curves, are also estimated (McCornmac et al., 2013). The length of a vertical curve L_v is calculated by using the curve integral of $f_v(x_i, z_i)$. The changes in slope Δs is the algebraic difference of the gradient between two adjacent points. The flatness of a vertical curve K, also called K-value, is the flatness of the vertical arc between P_{vc} and P_{vt}. The road geometry parameters and their corresponding equations are listed in Table 5.1.

5.4 Results and Discussion

5.4.1 Datasets

The experimental data contains MLS point clouds and high-resolution satellite imagery. The MLS point clouds were collected from the HaiCang Industrial Park (HCIP), the Coastal Ring Road (CRR), and the International Conference and Exhibition Center (ICEC) in Xiamen, China, by using a RIEGL VMX-450 MLS system. By integrating two full-view RIEGL VQ-450 laser heads, such VMX-450 MLS system is capable of producing a maximal effective measurement rate of 1.1 million measurements/sec and a maximal scanning range up to 800 m (@150 kHz). The average point density of these data is over 4,600 pts/m^2, and the absolute measurement accuracy and precision can reach 8 mm and 5 mm, respectively. The HCIP consists of complex road corridors and structures, the IECE has complicated urban road conditions with multiple road types, and the CRR contains multi-lane urban expressways with many roadside trees. Such road scenarios result in incomplete road boundary extraction, which makes these datasets suitable for road boundary completion evaluation. Moreover, satellite images with a ground sample distance (GSD) of 50 cm were obtained from Google Maps.

5.4.2 Hyperparameter Optimization

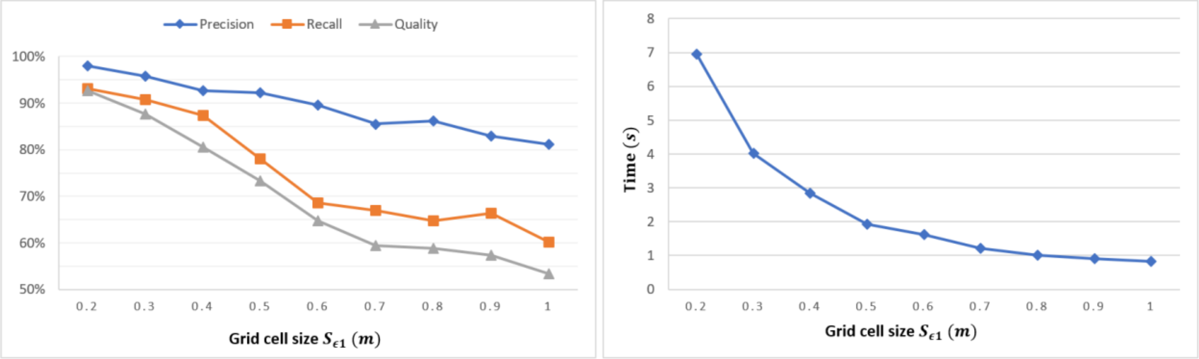
The proposed BoundaryNet has two essential hyperparameters: $S_{\epsilon1}$, the pixel cell size in the revised U-shaped encoder-decoder for road boundary extraction; and $S_{\epsilon2}$, the pixel cell size in the c-DCGAN model for road boundary refinement. To determine the optimal hyperparameter values, the performance of different configurations of these two parameters was evaluated on the CRR dataset during the process of road boundary line denoising and road boundary completion, respectively. The grid cell size was tested in the range of [20, 100] with an interval of 10 cm. The following three evaluation metrics, i.e., precision, recall, and quality, were utilized to quantitatively analyze the performance of road boundary extraction and refinement.

$$\text{Precision: } PR = \frac{TP}{L_{fc}} = \frac{TP}{TP+FP} \tag{5.7}$$

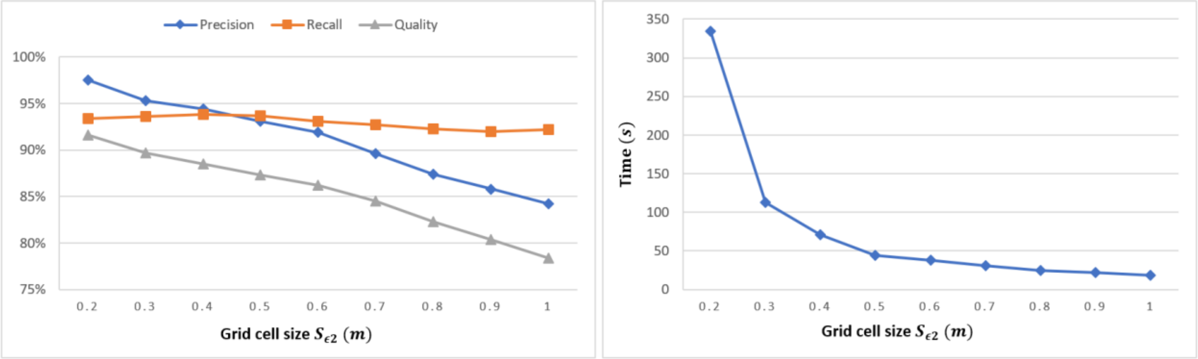
$$\text{Recall: } RE = \frac{TP}{L_{gt}} = \frac{TP}{TP+FN} \tag{5.8}$$

$$\text{Quality: } QU = \frac{TP}{L_{fc}+FN} = \frac{TP}{TP+FP+FN} \tag{5.9}$$

where TP is the length of correctly extracted boundary lines, FP is the length of the extracted boundaries that do not exist in the data, and FN is the length of the ground truth boundaries that are not extracted. L_{gt} indicates the whole length of the ground truth boundary lines, and L_{fc} represents the whole length of completed boundary lines.



(a)



(b)

Figure 5. 8 Performance evaluation and time cost of the proposed models using different grid cell sizes. (a) U-shaped model for road boundary extraction. (b) c-DCGAN model for road boundary refinement.

Figure 5.8 indicates that different grid cell sizes could deliver different performance during the phase of road boundary extraction and refinement. As shown in Figure 5.8(a), all the evaluation metrics, i.e., precision, recall, and quality, achieve better results than others when setting $S_{\epsilon 1}$ at 20 cm. Although it is the most time-consuming, the time cost is still low and reasonable. Therefore, in this study, $S_{\epsilon 1}$ is defined as 20 cm. Figure 5.8(b) shows the recall of the completed road boundaries increases in the range of [0.2m, 0.4m], and then reduces with an increase in grid cell size $S_{\epsilon 2}$. Typically, larger grid cell sizes lead to smaller road boundary gaps. The proposed c-DCGAN model delivers better completion performance on the missing parts of smaller sizes. Still, if the grid cell size is too large, the generated road boundary images are in coarse resolutions, resulting in incorrect gap detection and completion results. With an increase in grid cell size, precision gradually decreases. Because larger grid cell sizes indicate that all lines in an image are thicker and coarser, it leads to the lack of boundary details. Regarding some missing parts in curved roads with small curvature, it is straightforward to restore them with straight lines directly. Moreover, as shown in Figure 5.8(b), completion performance is almost the same when setting the grid cell size $S_{\epsilon 2}$ at 20 cm and 30 cm, respectively. Nevertheless, if $S_{\epsilon 2}$ is set to be 20 cm, the time cost for the c-DCGAN model is much higher than that of setting 30 cm. Hence, to achieve a balance between road boundary completion and computation efficiency, $S_{\epsilon 2}$ is set at 30 cm. Herein, to obtain a high-performance model, the segmented road boundary point clouds are first converted to the 2D images with a grid cell size of 20 cm, in which a U-shaped boundary denoising model is then conducted. Next, the correct road boundary lines are transformed back to 3D point clouds and re-projected onto the 2D images with a 30 cm grid cell size for the c-DCGAN based refinement.

5.4.3 Extraction Results

To quantitatively evaluate the proposed BoundaryNet model, the road boundary points were first manually extracted on the testing datasets. Considering the missing parts of road boundaries, points were manually added based on the road design standards and actual road conditions. To demonstrate the efficiency and robustness of BoundaryNet model for road boundary extraction and completion in complex road environments, road boundaries with varying completeness and curvatures were extracted by using several approaches, i.e., projection-related (Serna and Marcotegui, 2013), saliency-related (Wang et al., 2015a), supervoxel-related (Zai et al., 2017), and curb-related (Ma et al., 2019a). Figure 5.9 shows the road boundary extraction results from the ICEC dataset. Accordingly, the recalls of these methods are 59.25%, 81.30%, 90.88%, and 91.12%, respectively. The higher recall of the curb-based method (Ma et al., 2019a) indicates that it is capable of robustly extracting road boundaries under various curvatures with the assistance of trajectory data in urban road scenes.

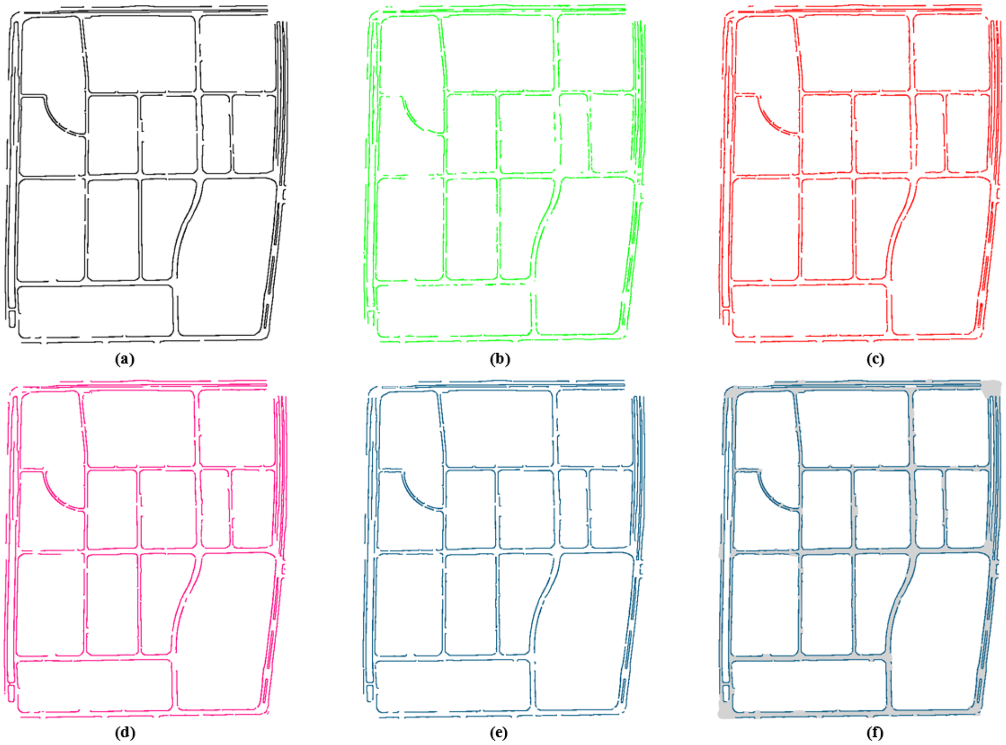


Figure 5. 9 Road boundary extraction results in the ICEC dataset. (a) Ground truth data (in black). (b) Road boundary extraction results derived from Serna and Marcotegui (2013) (in green), (c) Wang et al. (2015a) (in red), (d) Zai et al. (2017) (in pink), and (e) Ma et al. (2019a) (in navy), respectively. (f) Road boundary completion results in the ICEC dataset based on the extracted road boundaries from Ma et al. (2019a).

5.4.4 Completion Results

Moreover, by adopting the same evaluation metrics used in Section 5.4.2, i.e., precision, recall, and quality, the road boundary completion results were then evaluated. Figures 5.10 and 5.11 present the boundary line completion results in the HCIP and CRR, respectively. Several common road scenarios, including straight roads, curved roads, and road intersections, were accordingly presented in zoom-in views. Although the completeness and curvatures vary in these road scenarios, the experimental results indicate that the BoundaryNet obtains high performance in road boundary completion. According to three parts of initial road boundaries with varying completeness and curvatures obtained from four different methods, Table 5.2 indicates the initial quantitative completion results, and Table 5.3 presents the quantitative assessment on the completion results of three evaluation metrics, respectively.

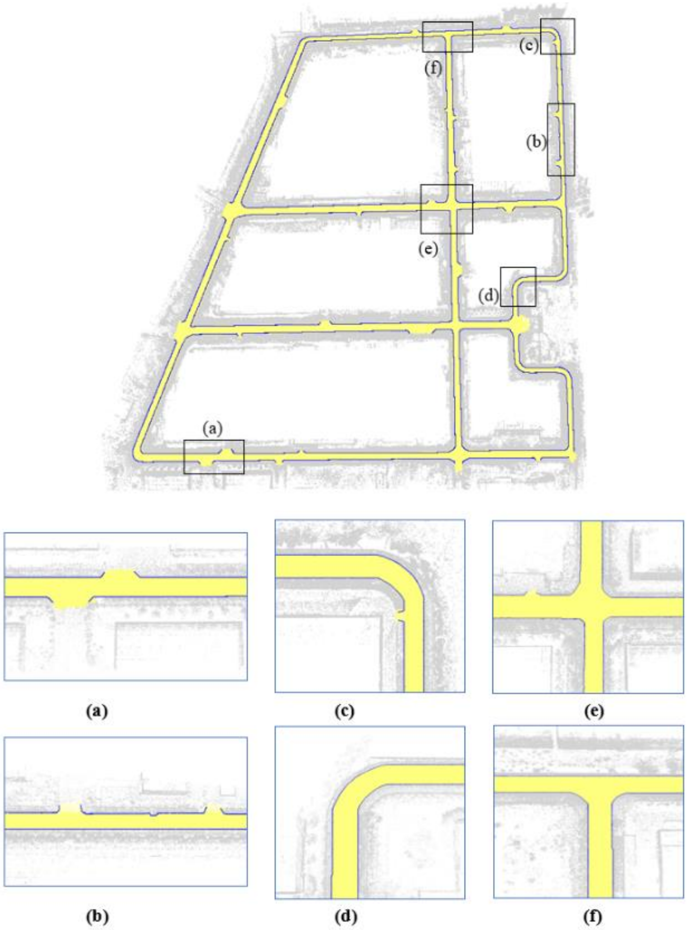


Figure 5. 10 Road boundary completion results using the BoundaryNet model in the HCIP dataset. (a)-(b) Straight line road sections. (c)-(d) Curved road sections. (e)-(f) Road intersections.

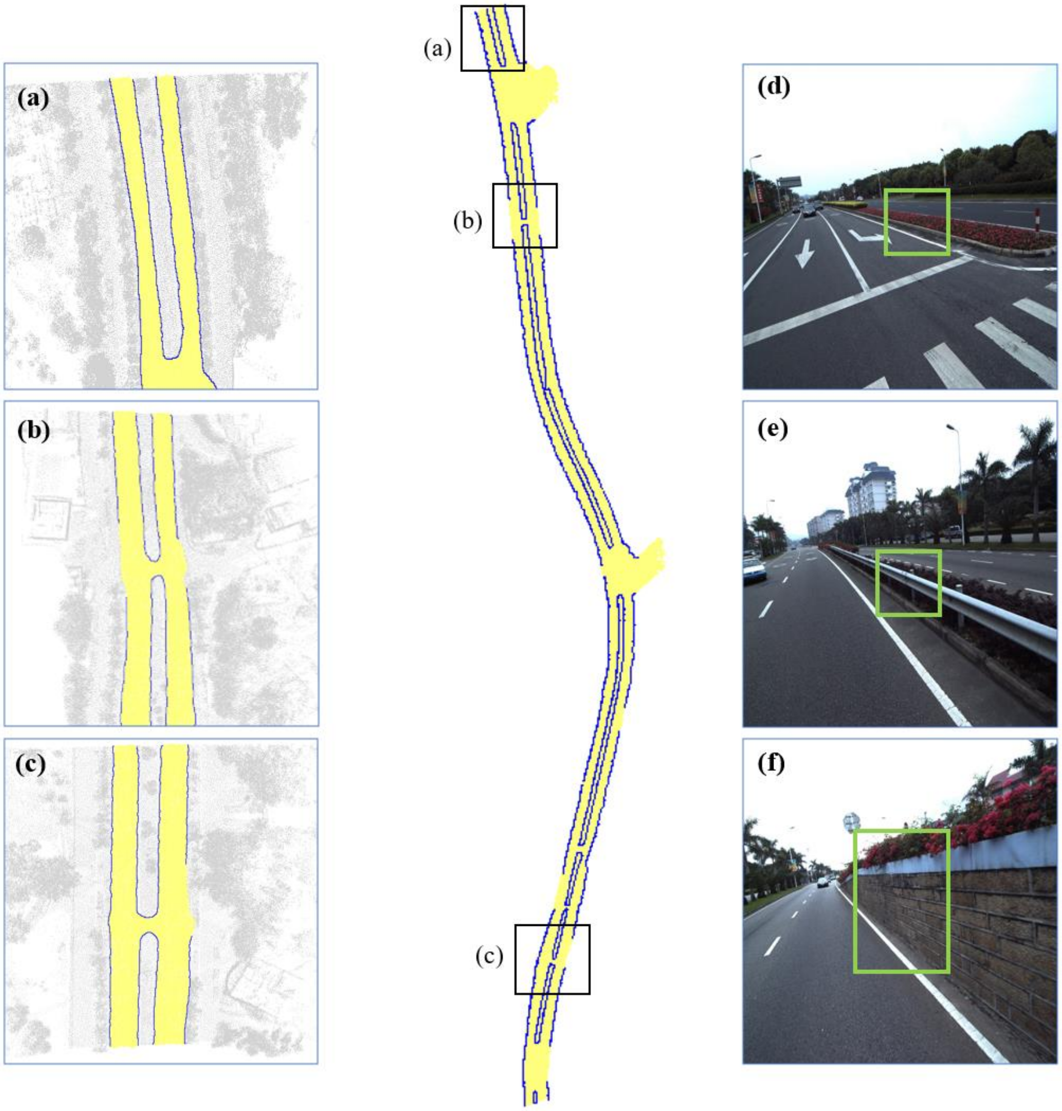
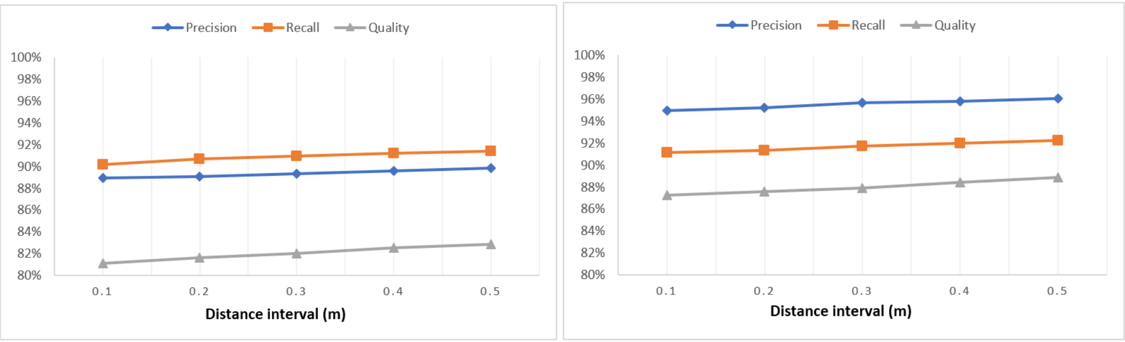


Figure 5. 11 Road boundary completion results using the BoundaryNet model in the CRR dataset. (a)-(c) Completion results in different road scenes. (d)-(f) Different road boundary patterns are shown in green boxes, which indicate road curbs, fences, and parapets, respectively.

Regarding the BoundaryNet tested on the both HCIP and CRR datasets, depending on road boundaries extracted by using the approach of Ma et al. (2019a), the precision, recall, and quality obtained on HCIP and CRR datasets are as follows: 89.89% and 96.06% in precision, respectively; 91.40% and 92.23% in recall, respectively; and 82.88% and 88.86% in quality, respectively. Based on the experimental results, it demonstrates that the road boundary denoising process is remarkably conducive to the completion results. Additionally, the quality of initial road boundary extraction results has a significant impact on the completion performance. The lower recall of the extracted boundary lines shows a more considerable information loss in the structure of these boundaries, which causes increasing uncertainties and difficulties for road boundary completion. That is, if the initial road boundary recall is low, the length of the final boundaries that exclude the ground truth boundaries (i.e., false positive) and the length of the ground truth boundaries that excludes the final completed boundaries (i.e., false negative) are large. Still, despite worn and incomplete road boundaries with a completeness rate of 59%, the BoundaryNet delivers increasing true positive results and significantly enhances recall. Therefore, the BoundaryNet achieves superior performance in road boundary completion, which can deal with incomplete road boundaries with varying rates of completeness. However, some roads have occlusions and various patterns, i.e., road curbs, fences, and parapets (see Figures 5.11(d)-(f)), it is challenging for the revised curb-based boundary extraction approach to extract boundary lines accurately in such places. To address this problem, multiple scans can be conducted at varying scan directions to collect more point clouds of roads, which remarkably increases the strength of the BoundaryNet model.

Furthermore, the influence of using different distance intervals was also evaluated in HCIP and CRR datasets during the process of road boundary extraction. Distance interval is determined depending on the average distance between manually edited points and their neighbouring points from ground truth boundary lines. According to prior knowledge and experimental results, the distance interval defined in the results presented in Tables 5.2 and 5.3 is 50 cm. As illustrated in Figure 5.12, the increasing distance interval values contribute to the substantial improvement in precision, recall, and quality for the HCIP and CRR datasets, which indicates that manually edited points correspond well with the ground truth boundaries.



(a) (b)

Figure 5. 12 Quantitative evaluation results of varying distance intervals in (a) HCIP dataset and (b) CRR dataset.

Table 5. 2 Initial boundary line completion results with varying completeness and curvatures in HCIP and CRR datasets.

Dataset	**Procedure**	**Evaluation metric (m)**	**Method**			
			Projection-related (Serna and Marcotegui, 2013)	Saliency-related (Wang et al., 2015a)	Supervoxel-related (Zai et al., 2017)	Curb-related (Ma et al., 2019a)
HCIP ($\boldsymbol{L_{gt}} =$ 9129.97 m)	Extraction results	TP	5033.91	6721.87	7879.85	8184.45
		FP	807.06	1462.56	2138.73	2039.31
		FN	3852.27	2233.03	1213.82	1238.68
		L_{fc}	5840.97	8184.43	10018.58	10223.76
	Extraction + boundary denoising results	TP	5010.53	6590.36	7570.41	7741.27
		FP	160.83	346.13	444.85	423.13
		FN	4069.45	2487.27	1570.96	1550.19
		L_{fc}	5171.36	6936.49	8015.26	8164.40
	Extraction + boundary denoising + completion results	TP	7924.29	8033.85	8303.14	8399.04
		FP	971.45	985.18	935.92	944.50
		FN	1064.24	995.99	787.22	790.12
		L_{fc}	8895.74	9019.03	9239.06	9343.54
CRR ($\boldsymbol{L_{gt}} =$ 4150.75 m)	Extraction results	TP	2160.01	2862.42	3644.34	3762.16
		FP	129.39	165.06	209.53	142.28
		FN	1884.75	1157.60	380.43	390.42
		L_{fc}	2289.40	3027.48	3853.87	3904.44
	Extraction + boundary denoising results	TP	2126.39	2850.24	3486.77	3558.86
		FP	33.11	46.65	78.34	83.03
		FN	2004.36	1250.50	586.57	599.20
		L_{fc}	2159.50	2896.89	3565.11	3641.89
	Extraction + boundary denoising + completion results	TP	3063.61	3561.38	3820.40	3864.83
		FP	464.05	390.98	167.20	158.70
		FN	564.43	446.94	326.36	325.63
		L_{fc}	3527.66	3952.36	3987.60	4023.53

Table 5. 3 Quantitative assessment on boundary line completion results in HCIP and CRR datasets.

Dataset	**Procedure**	**Evaluation metric (%)**	**Method**			
			Projection-related (Serna and Marcotegui, 2013)	Saliency-related (Wang et al., 2015a)	Supervoxel-related (Zai et al., 2017)	Curb-related (Ma et al., 2019a)
HCIP	Extraction results	PR	86.18	82.13	78.65	80.05
		RE	56.65	75.06	86.65	86.85

		QU	51.93	64.53	70.15	71.40
	Extraction + boundary denoising results	PR	96.89	95.01	94.45	94.81
		RE	55.18	72.60	82.81	83.31
		QU	54.22	69.93	78.97	79.69
	Extraction + boundary denoising + completion results	PR	89.08	89.08	89.87	89.89
		RE	88.16	88.97	91.34	91.40
		QU	79.56	80.22	82.81	82.88
CRR	Extraction results	PR	94.35	94.55	94.56	96.36
		RE	53.40	71.20	90.55	90.60
		QU	51.75	68.40	86.07	87.60
	Extraction + boundary denoising results	PR	98.47	98.38	97.80	97.72
		RE	51.48	69.51	85.60	85.59
		QU	51.07	68.72	83.98	83.91
	Extraction + boundary denoising + completion results	PR	86.85	90.11	95.81	96.06
		RE	84.44	88.85	92.13	92.23
		QU	74.87	80.95	88.56	88.86

5.4.5 Refinement Results

In this study, road centerlines extracted from high-resolution satellite imagery were utilized to enhance the completeness of extracted road boundaries in urban roadways. Figure 5.13 shows the results of road centerline extraction. Due to the missing parts (e.g., large gaps and irregular road structures) in the extracted road boundaries, 1,000 training samples were manually generated depending on road centerlines and boundaries, where each category has 250 positive samples and 250 negative samples. The negative samples were merged with extracted road centerlines as new training inputs to feed into the c-DCGAN boundary refinement model. Specifically, the batch size and epochs were defined as 4 and 300, respectively.

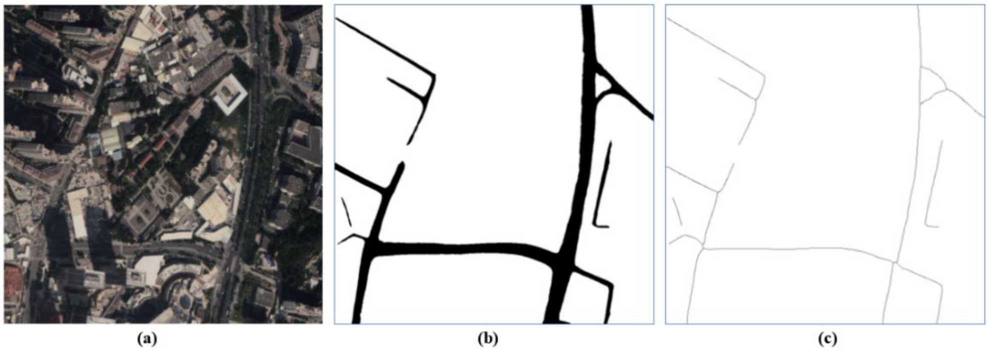


Figure 5. 13 Road centerline extraction from Google Map imagery. (a) High-resolution Google Map image captured in Xiamen, China. (b) Road extraction results using the D-LinkNet model. (c) Road centerline extraction results using the morphological thinning algorithm.

In the testing phase, 400 negative samples were employed to examine the performance of the c-DCGAN boundary refinement model. The generated road boundaries were evaluated by adopting the buffer-overlay-statistics (BOS) approach (Tveite, 1999). By specifying different buffer sizes around the manually annotated road boundaries (i.e., ground truth boundaries), the generated road boundaries after refinement were compared with ground truth boundaries through overlaying and statistics. Accordingly, the accuracy of completed road boundaries was evaluated by the miscoding, which indicates what percentage of the completed road boundaries are located outside of the reference buffers. The miscoding is determined as follows:

$$Miscoding = \frac{Length\ (L \text{ outside } C_iR \text{ in } LC_iR)}{Length\ (L)} \tag{5.10}$$

where L is the completed road boundaries, C_iR is the generated buffer zones, and LC_iR is a mixed dataset by overlaying L and C_iR.

Table 5. 4 Quantitative assessment of road boundaries after refinement in a part of the CRR dataset.

Buffer size	**Miscoding**	
	Large gaps	Irregular road structures
3 cm	3.98 %	4.55 %
5 cm	0	2.35 %
7 cm	0	0

Table 5.4 presents the quantitative assessment in miscoding for different gap categories in a part of the CRR dataset. Consequently, the miscoding in large gaps and irregular road structures is 3.98% and 4.55%, respectively, when setting the buffer size as 3 cm. The miscoding significantly decreases with increased sizes of reference buffers. Notably, for large gaps, the miscoding reduces to 0% by changing the buffer size to 5 cm, which indicates that all road boundaries after refinement have located inside 5 cm reference buffers. Additionally, the refinement model delivers a 0% miscoding for irregular road structures by setting the buffer size as 7 cm. The results demonstrate that the refinement model achieves high performance for straight road boundaries. However, it is challenging to manually annotated ground truth boundary lines in curved road sections, a slight difference between the manually labeled boundary lines and real ground truths can lead to a large miscoding. Thus, the overall performance of the c-DCGAN refinement model is underestimated.

5.4.6 Evaluation on Paris-Lille-3D dataset

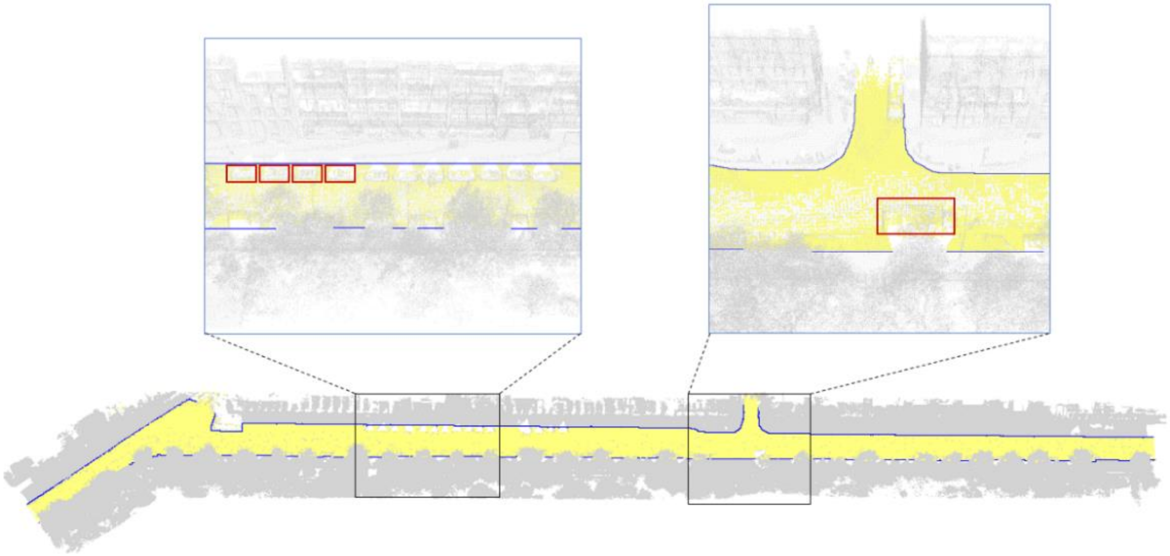


Figure 5. 14 Boundary line completion results using the BoundaryNet model in Paris-Lille-3D dataset.

To further demonstrate the robustness and efficiency of the BoundaryNet in relatively low-quality point clouds with low point densities and measurement accuracy, the BoundaryNet performance was evaluated using the Paris-Lille-3D dataset (Roynard et al., 2018). Paris-Lille-3D point clouds were acquired in the two metropolitan cities in France, i.e., Paris and Lille, by using a vehicle-borne MLS system comprising a Velodyne HDL-32E LiDAR sensor. Compared to the RIEGL VQ-450 laser scanners equipped in VMX-450 MLS systems, the Velodyne HDL-32E laser scanner is capable of delivering a maximal effective measurement rate of 0.7 million measurements per second in a scanning range up to 120 m. The average point density is over 1,500 pts/m^2, and the absolute measurement accuracy can achieve 2-5 cm. This Paris-Lille-3D dataset contains typical urban roadway scenarios with occlusions and various point densities and intensities, resulting in significant challenges for accurate road boundary completion. Moreover, the trajectory data was generated by using the tightly coupled GNSS-RTK/INS extended Kalman filter (EKF) method assembled in the Inertial Explorer software. Thus, the road boundaries were extracted using the curb-based method (Ma et al., 2019a) with fine-tuning parameters. The boundary line completion results are shown in Figure 5.14, in which two samples of typical road corridors with occlusions are enlarged for visual interpretation. Although the initial road boundaries are incomplete due to occlusions caused by street parking cars and moving trucks (see red boxes in Figure 5.14), the proposed completion model can robustly and accurately recover such boundary lines. Consequently, the BoundaryNet achieves 89.31% in precision, 87.65% in recall, and 84.89% in quality in the Paris-Lille-3D dataset, respectively. Remarkably, the recall increases by 29.33% for the Paris-Lille-3D dataset after conducting road boundary denoising and completion operations, which demonstrates that the BoundaryNet can deliver a promising solution for road boundary completion in the low-quality point clouds.

5.4.7 Road Geometry Calculation Results

Due to the lack of intuitive vertical curved road sections in three surveying areas (i.e., ICEC, HCIP, and CRR), only horizontal road alignments were estimated and evaluated. In the study of Wen et al. (2019a), the horizontal road alignments were measured by using a Leica TS-15 total station and a Leica Viva GS-15 base and rover system, which can achieve millimeter-level absolute measurement accuracy. Therefore, by using the same testing MLS point clouds used in Wen's study, such manual field measurements are used as ground truth data to evaluate the performance of the proposed approach quantitatively. Table 5.5 shows the horizontal road geometries calculated from four roads. Consequently, the experimental results delivered a maximum error of $49'04''$ for the intersection angle θ, 0.12 m for the length of horizontal curve L_h, and 0.0004 for the curvature of horizontal curve C, respectively.

Table 5. 5 Horizontal road geometry calculation results using four sample road sections.

Road sections	**θ (°,′ ,″)**			**L_h (m)**			**C**		
	Calculation	GT	Error	Calculation	GT	Error	Calculation	GT	Error
1	120°59′56″	121°35′14″	-35′18″	166.27	166.15	0.12	0.0062	0.0061	0.0001
2	115°18′26″	114°29′22″	49′04″	213.90	214.01	-0.11	0.0053	0.0053	0
3	104′25′04″	104°14′54″	10′10″	26.80	26.69	0.11	0.0492	0.0495	0.0003
4	95°30′42″	96°00′36″	-29′54″	28.69	28.77	-0.08	0.0514	0.0510	0.0004
Max Error			49′04″			0.12			0.0004

5.5 Efficiency Evaluation

Road boundary extraction module in the developed BoundaryNet framework, programmed with C++, was debugged on a Dell desktop with an Intel® i5-7500 CPU (@3.40 GHz) and a 16 GB RAM. All CNN-based road boundary denoising, c-DCGAN based boundary completion, and D-LinkNet based road centerline extraction modules were tested using a workstation equipped with a Nvidia® Geforce 1080Ti graphic card with 12 GB memory and a 32 GB RAM. The detailed discussion about initial road boundary extraction was provided by Ma et al. (2019a). Herein, this study mainly concentrates on the time cost in the process of road boundary denoising and completion. Table 5.6 indicates that the c-DCGAN based road boundary refinement module, running on the GPU, consumes most of the total computational cost. In this study, only the size of input images is considered as the primary factor that determines the time cost. The sizes of the projected road boundary lines onto 2D images in three road scenes, i.e., HCIP, CRR, and Paris-Lille-3D, are 3980 × 3660, 1876 × 5980, and 2540 × 1678, respectively. Accordingly, the HCIP dataset has the highest computational cost. Benefiting from the acceleration of the GPU, this c-DCGAN model has been boosted in a 40× faster fashion, instead of using CPU computation. Consequently, the total processing time for datasets HCIP, CRR, and Paris-Lille-3D are 164.14 s, 95.30 s, and 70.91 s, respectively.

Table 5. 6 Computational cost of different phases for road boundary completion in three datasets.

Dataset	**Boundary completion process**		**Time cost (s)**
	CNN-based boundary line denoising (s)	c-DCGAN based boundary line refinement (s)	
HCIP	30.85	133.29	164.14
CRR	4.97	90.33	95.30
Paris-Lille-3D	3.45	67.46	70.91

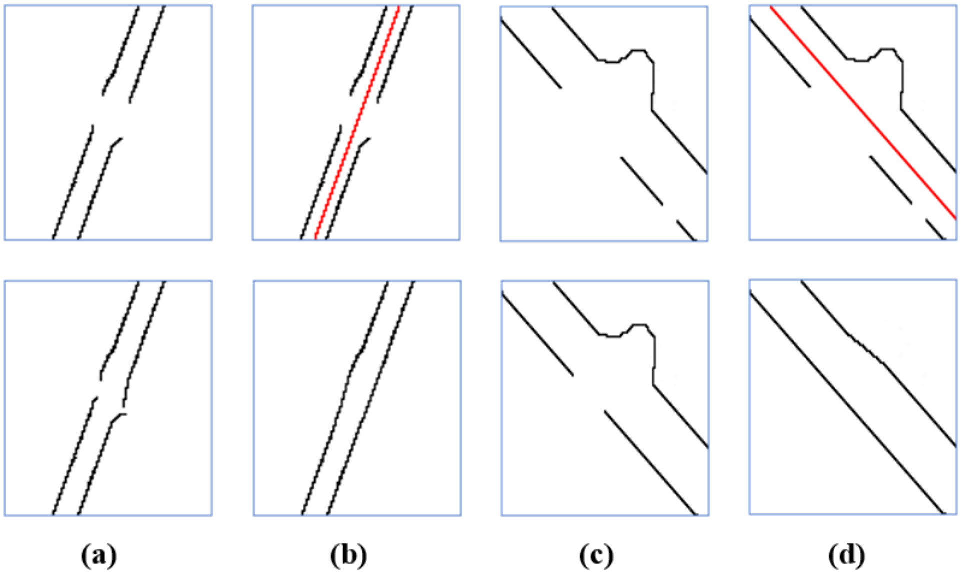


Figure 5. 15 Experimental results on road boundary refinement. (a-b) Training inputs of large gaps, and their boundary refinement results with and without the assistance of the road centerline (in red color), respectively. (c-d) Training inputs of irregular structures, and their boundary refinement results with and without the assistance of the road centerline (in red color), respectively.

To estimate the influence of high-resolution satellite imagery, the BoundaryNet completion performance was further evaluated with or without the assistance of road centerlines. More specifically, 150 epochs were run on the HCIP dataset, the precision, recall, and quality were accordingly recorded in Table 5.7. By introducing road centerlines into BoundaryNet framework for road boundary refinement, the precision, recall, and quality increase by 1.59%, 2.48%, and 2.74%, respectively. Figure 5.15 illustrates some testing results of road boundary refinement with or without using road centerlines. As can be perceived, it is very challenging to complete the missing parts, especially for irregular road structures, without the guidance of road centerlines. With the assistance of road centerlines, the c-DCGAN boundary refinement model can refine the curved roads with gaps based on the geometric characteristics of the road centerlines (e.g., connectivity and curvature). The experimental results demonstrate such road centerlines extracted from high-resolution satellite imagery can significantly improve the completeness of extracted road boundaries in urban roadways.

Table 5. 7 Performance evaluation of the BoundaryNet framework with and without road centerlines on HCIP dataset.

Model	**Evaluation metric**		
	Precision (%)	Recall (%)	Quality (%)
Extraction + boundary denoising + c-DCGAN completion (without centerlines)	88.30	88.92	80.14
Extraction + boundary denoising + c-DCGAN completion (with centerlines)	89.89	91.40	82.88

5.6 Chapter Summary

This paper proposes a deep learning framework to deal with road boundary extraction and completion in complex urban road environments. These problems lead to completeness reduction and curvature loss when processing massive MLS point clouds with many missing parts, most remarkably because of the occlusions caused by road users, background interference, and incomplete data collection. A novel deep learning based framework is developed, named BoundaryNet, to restore 3D road boundaries by employing MLS point clouds and high-resolution satellite imagery. This BoundaryNet model provides a promising solution for 3D road boundary completion by performing a U-shaped neural network for the boundary denoising, a CNN-based network for the boundary completion, and a c-DCGAN based network with the assistance of road centerlines extracted from satellite images for the boundary refinement. Based on the completed road boundaries, the inherent road geometries are calculated.

The proposed methods have been evaluated by determining robustness and efficiency in varying road conditions (e.g., incomplete data collection and worn road curbs), which has demonstrated the superior performance of the BoundaryNet model in road boundary completion. For three highly dense MLS point clouds (i.e., HCIP, ICEC, and CRR) and a relatively low-density dataset (i.e., Paris-Lille-3D), the precision, recall, and quality obtained from HCIP, ICEC, CRR, and Paris-Lille-3D are as follows: precision: 89.89%, 89.64, 96.06%, and 89.31%, respectively; recall: 91.40%, 91.12%, 92.23%, and 87.65%, respectively; and quality: 82.88%, 82.43%, 88.86%, and 84.89%, respectively. Overall, it can be concluded that the developed BoundaryNet model can restore the missing parts of road boundaries under challenging road scenes more robustly, accurately, and efficiently. Further studies will concentrate on boosting the BoundaryNet model performance by developing an end-to-end deep learning framework for 3D road boundary restoration and employing more multi-source data (e.g., open street maps and camera images) to solve the occlusions caused by roadside objects. Moreover, various data distributions obtained from different road environments (e.g., rural roads) should be further used to demonstrate the generalizability of the proposed BoundaryNet model.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

This thesis proposes several deep neural networks for road information extraction by using MLS point clouds. Due to the significant advantages of high flexibility, large-scale data coverage, improved measurement efficiency, less weather sensitivity, and low labour cost, MLS systems are being widely used at an increasing rate in many transportation-related applications. However, because of point density and intensity variations and occlusions caused by road participants, huge challenges remain in completely and efficiently extracting high-level 3D point cloud features, particularly in large-scale and complex road environments. Based on the publicly available point cloud benchmarks with labels and powerful computational resources, deep learning-based methods have demonstrated that they are capable of learning deeper and more distinctive feature representations of different road objects.

This thesis provides a promising solution for intelligent road information extraction and HD map generation with three novel algorithms: (1) road object semantic segmentation, (2) road marking extraction and classification, and (3) road boundary completion. This doctoral thesis intelligently and efficiently extracts road information and explores road inventories from MLS point clouds by using deep neural networks, which further shows the capabilities and extensive applications of advanced MLS systems in road planning and construction surveying.

To deal with the challenges of relatively low feature representativeness and robustness of the most 3D point cloud segmentation methods, an end-to-end feature extraction framework, called MS-PCNN, is developed for 3D point cloud segmentation by using dynamic point-wise convolutional operations in multiple scales. Accordingly, the proposed MS-PCNN network has four main strengths. First, the revised point-wise convolutional filters can learn spatial relationships and extract geometric information of point clouds in local regions, contributing to permutation and translation invariances. Second, MS-PCNN applies a hierarchical PointCONV-based downsampling and DePointCONV-based upsampling architecture so that more high-level features can be extracted in multiple scales. Third, by improving the dynamic graph edge convolution, MS-PCNN can improve feature descriptiveness by learning edge features between a point and its adjacent neighbours. Finally, a CRF post-processing algorithm is used to ensure the consistency of point-wise label prediction and refine segmentation results. The MS-PCNN model is robust to occlusions and diverse point densities for urban scene point clouds. Overall, it is concluded that the MS-PCNN network can achieve dominating performance in 3D point cloud semantic segmentation under large-scale point cloud scenes more effectively and robustly.

Then, to minimize the impacts of intensity and point density variation problems, two novel capsule-based network architectures are proposed for road marking extraction and classification, respectively, from highly dense MLS point clouds with an irregular data format. Moreover, a road marking dataset, containing both 3D point clouds collected by both MLS and BLS systems and manually labeled reference data, is created from three types of road environments, including urban roads, highways, and underground garages. The proposed models are accordingly evaluated by estimating robustness and efficiency using these self-built datasets. In the extraction process, a U-shaped capsule-based network is developed to extract road markings using 2D georeferenced intensity images. The experimental results demonstrated that the proposed model could effectively encode high-level features (e.g., changing intensity and pose information) with significantly enhanced road marking extraction performance. In the classification process, a hybrid capsule-based network is proposed to classify seven types of road markings. Compared to those manually defined rule-based classification methods, the proposed methods can automatically learn more salient features embedded in intensity values, as well as the shape information of the road markings by using a revised dynamic routing algorithm and powerful L-Softmax loss function. The experimental results have indicated that capsule-based networks are capable of effectively extracting inherent features from massive MLS point clouds and achieving superior performance in road marking extraction and classification tasks.

Finally, to deal with the varieties and uncertainties of missing parts in urban road boundaries, a novel deep learning-based framework is developed, named BoundaryNet, to restore 3D road boundaries by employing MLS point clouds and high-resolution satellite imagery. In this network, road boundaries are first extracted by conducting a curb-based extraction method. Such extracted 3D road boundary lines are fed into a U-shaped network for the erroneous boundary denoising purpose. Then, a CNN model is proposed to complete the road boundaries. Next, to achieve more complete and accurate road boundaries, two c-DCGAN models with the assistance of road centerlines extracted from satellite images are developed. Then, according to the completed road boundaries, the inherent road geometries are calculated. More complete and correct road boundaries with a wealth of road information are obtained if an MLS point cloud related road boundary is generated with the assistance of satellite images. Overall, it can be concluded that the developed BoundaryNet model can restore the missing parts of road boundaries under challenging road scenes more robustly, accurately, and efficiently.

In summary, this thesis proposes three novel deep learning frameworks for accurate and robust road information extraction by using MLS point clouds, which effectively address the research gaps of intelligent road information extraction, especially in complex urban road scenarios. Multiple experiments have provided highly accurate multi-object segmentation results and feature extraction results, contributing to road marking extraction, classification, and road boundary recovery tasks, which typically pose enormous difficulties for both threshold-based and rule-based methods in the literature. These three novel deep learning methods indicate a promising solution for intelligent road information extraction by using mobile LiDAR point clouds, which remarkably support the development of high-definition maps and autonomous driving.

6.2 Contributions

Deep learning methods have achieved state-of-the-art performance in the domains of computer vision and remote sensing that both CNNs and discriminative/generative models are capable of exploring feature representation and extracting road information from MLS point clouds. This thesis presents a promising solution for automated and efficient road information extraction. In summary, the main contributions of this thesis are as follows:

- A multi-scale point-wise convolution algorithm, named MS-PCNN, is proposed for road object semantic segmentation, which can directly consume unstructured 3D points and implement a point-wise semantic label assignment network to learn fine-grained layers of feature representations and reduce unnecessary convolutional computations. The MS-PCNN effectively handles the problems of low feature descriptions and low robustness suffered by the most 3D point cloud segmentation algorithms. As demonstrated by the experimental results, the MS-PCNN is superior to other DL-based methods under large-scale point cloud scenes in both segmentation accuracy and computational complexity.

- Two capsule-based neural networks are developed for road marking extraction and classification. The impacts of low-intensity contrast between road markings and their surrounding pavements, as well as varying point densities, are remarkably decreased through encoding the image patches at various locations. The whole road marking extraction and classification framework provides a promising solution for preloaded HD map creation, which further produces an essential road inventory dataset for road marking updates to support the development of AVs.
- A CNN- and c-DCGAN-based method, named BoundaryNet, is proposed for road boundary recovery. Different from 3D point-based methods suffering from incomplete data collection and point density and intensity variations, the BoundaryNet framework combining both MLS point clouds and satellite images can more robustly extract and complete road boundaries, and accurately estimates the road characteristics in large-scale urban environments.
- Furthermore, several training datasets with labels are created. More specifically, a road marking dataset containing both 3D point clouds and manually labeled reference data in three types of road scenes (i.e., urban roads, highways, and underground garages) is constructed. Moreover, a road boundary dataset consisting of 2D intensity imagery and manually edited reference imagery with large gaps and irregular road structures are accordingly built. Such datasets will be publicly released to motivate relevant research.

6.3 Recommendations for Further Research

The proposed MS-PCNN, BoundaryNet, and capsule-based networks have established a solid foundation for intelligent road information extraction in the context of MLS point cloud processing. Such methods have achieved state-of-the-art performance in road object extraction, segmentation, and classification. By taking advantage of such methods developed in this thesis, several further research interests are thus summarized:

(1). **Multi-source data fusion**: Instead of using a single data source, many studies (Liang et al., 2018; Wen et al., 2019a) are employing multi-source data including LiDAR point clouds, drone-based imagery, and airborne or satellite synthetic aperture radar (SAR) imagery to provide significant spectral, spatial, and texture information for the robust and effective road feature extraction in large-scale terrains. However, these data obtained from multiple platforms at several times, in different weather situations, under various lighting conditions, and with diverse sampling densities, make the rapid and robust data calibration and data fusion challenging. Additionally, the point cloud data fusion between high-end LiDAR sensors (e.g., RIEGL VQ-45) and low-end LiDAR sensors (e.g., Velodyne VLP-16) also indicates a valuable research direction for HD map generation and real-time update.

(2). **Graph neural networks (GNNs)**: Recent studies (Zhou et al., 2018b; Landrieu and Simonovsky, 2018; Te et al., 2018) indicate that graph neural networks can deliver a convincing performance and high interpretability for pattern recognition and object extraction from complicated and powerful graphs, including social media networks, physical systems, knowledge graphs, and other related networks. The traditional CNNs applying on irregular 3D point clouds, such as SegCloud (Tchapmi et al., 2017) and SnapNet (Boulch et al., 2017), cannot learn the inherent features, resulting in poor discrimination performance. Graph convolutions have provided an effective method to deal with large-scale 3D point clouds, in which the nodes indicate simple features while the edges represent their adjacent relationship encoded by rich-attributed edge features, as demonstrated by the revised edge convolution operations in the MS-PCNN model.

Moreover, the extended GNN-CRF methods with graph-based constraints also provide powerful functionalities to graph neural networks. Because point clouds are unorganized and point-like graphs, deep graph-based networks that use graphs as inputs can be designed for object extraction, segmentation, and classification purposes. Notably, GNNs can capture not only the pose information but also the spatial relationships among points. Thus, developing GNNs that directly consume 3D point clouds and encode graph features in the fields of road information extraction remains an interesting and challenging task.

(3). **Domain adaption**: To support the advancement of HD maps and AVs, the proposed deep learning methods that focus on intelligent road information extraction and environmental perception are required to adapt different road scenarios in the real world. Domain adaption techniques have been commonly applied by using state-of-the-art deep neural networks (Sun et al., 2013; Silver et al., 2017; Sung et al., 2018). Specifically, deep neural networks embedding with prior knowledge are first trained by using one specific dataset. New deep learning models are then trained using different datasets with various data distributions, which have more learning capacities than those trained from scratch. Due to the limited amount of labelled data and huge data distribution differences in various datasets, adapting the existing knowledge encoded from these labelled samples from one specific dataset to massive unlabelled data from different data distributions poses a considerable challenge. Thus, domain adaption techniques can be further employed to effectively test the generalizability of the proposed neural networks in this thesis, by evaluating the model performance based on different data distributions collected by different LiDAR sensors and in various road scenarios.

(4). **Semi-supervised learning**: Most state-of-the-art deep neural networks (Yang et al., 2018; Li et al., 2019a) are usually designed under supervised methods using labelled training samples, such as point-wise segmentation labels and 3D object bounding boxes. However, because of the limited accessibility of high-quality, large-scale, and massive road object datasets and benchmarks, fully supervised learning models are sensitive to model generalization capability, resulting in non-robust for untrained samples. In contrast, semi-supervised learning methods are capable of improving the generalization capability of different deep neural networks and solving the data absence problem.

(6). **Intelligent processing for other road objects**: Apart from road markings and road boundaries in urban road scenes, both on-ground and off-ground road objects, such as traffic lights, roadside trees, manhole covers, and road cracks, are significant for road participants and autonomous vehicles to provide necessary traffic guidance and improve traffic safety. However, due to the variations of intensity and point density and incompleteness of scanned objects, accurately and robustly detecting and classifying these road objects with detailed context information are still in enormous challenges. AI-based and multi-scale feature fusion methods (Xie et al., 2018; Dong et al., 2018) have delivered a promising solution to extract road objects and context information. Intelligent road object processing and analysis embedded in end-to-end trainable frameworks also present meaningful research directions to promote the development of smart cities and HD maps, such as GNN-based road crack detection and capsule-based manhole cover classification.

References

Armeni, I., Sax, S., Zamir, A. R., Savarese, S., 2017. Joint 2D-3D-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105.

Bai, S., Bai, X., Zhou, Z., Zhang, Z., Jan Latecki, L., 2016. GIFT: A real-time and scalable 3D shape search engine. *In Proc. IEEE CVPR*, pp. 5023-5032.

Bétaille, D., Toledo-Moreo, R., 2010. Creating enhanced maps for lane-level vehicle navigation. *IEEE Trans. Intell. Transp. Syst.*, 11(4), 786-798.

Boulch, A., Le Saux, B., Audebert, N., 2017. Unstructured point cloud semantic labeling using deep segmentation networks. *Proceedings of the Workshop on 3D Object Retrieval*, pp. 17-24.

Boyko, A., Funkhouser, T., 2011. Extracting roads from dense point clouds in large scale urban environment. *ISPRS J. Photogramm. Remote Sens*., 66(6), S2-S12.

Bresson, G., Alsayed, Z., Yu, L., Glaser, S., 2017. Simultaneous localization and mapping: a survey of current trends in autonomous driving. *IEEE Trans. Intell. Transp. Syst.*, 2(3), 194-220.

Cabo, C., Kukko, A., García-Cortés, S., Kaartinen, H., Hyyppä, J., Ordoñez, C., 2016. An algorithm for automatic road asphalt edge delineation from mobile laser scanner data using the line clouds concept. *Remote Sens*., 8(9), 740-760.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Xiao, J., 2015. ShapeNet: an information-rich 3D model repository. arXiv preprint arXiv:1512.03012.

Che, E., Jung, J., Olsen, M. J., 2019. Object recognition, segmentation, and classification of mobile laser scanning point clouds: a state of the art review. *Sensors*, 19(4), 810-852.

Chen, X., Ma, H., Wan, J., Li, B., Xia, T., 2017a. Multi-view 3D object detection network for autonomous driving. *In Proc. IEEE CVPR*, pp. 1907-1915.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., 2017b. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4), 834-848.

Chen, Y., Wang, S., Li, J., Ma, L., Wu, R., Luo, Z., Wang, C. 2019a. Rapid urban roadside tree inventory using a mobile laser scanning system. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*., 12(9), 3690-3700.

Chen, Z., Fan, W., Zhong, B., Li, J., Du, J., Wang, C., 2019b. Corse-to-fine road extraction based on local dirichlet mixture models and multiscale-high-order deep learning. *IEEE Trans. Intell. Transp. Syst*. Doi: 10.1109/TITS.2019.2939536.

Cheng, M., Zhang, H., Wang, C., Li, J., 2016. Extraction and classification of road markings using mobile laser scanning point clouds. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 10(3), 1182-1196.

Chu, H., Guo, L., Gao, B., Chen, H., Bian, N., Zhou, J., 2018. Predictive cruise control using high-definition map and real vehicle implementation. *IEEE Trans. Veh. Technol*., 67(12), 11377-11389.

Chu, H., Li, D., Acuna, D., Kar, A., Shugrina, M., Wei, X., Fidler, S., 2019. Neural turtle graphics for modeling city road layouts. *In Proc. IEEE CVPR*, pp. 4522-4530.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. *In Proc. IEEE CVPR*, pp. 5828-5839.

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Raska, R., 2018. DeepGlobe 2018: A challenge to parse the earth through satellite images. *In Proc. IEEE CVPRW*, pp. 172-181.

Dong, Z., Yang, B., Liang, F., Huang, R., Scherer, S., 2018. Hierarchical registration of unordered TLS point clouds based on binary shape context descriptor. *ISPRS J. Photogramm. Remote Sens.*, 144, 61-79.

Duarte, K., Rawat, Y. S., Shah, M., 2019. CapsuleVOS: Semi-supervised video object segmentation using capsule routing. *In Proc. IEEE CVPR*, pp. 8480-8489.

Engelmann, F., Kontogianni, T., Hermans, A., Leibe, B., 2017. Exploring spatial context for 3D semantic segmentation of point clouds. *In Proc. IEEE ICCVW*, pp. 716-724.

Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors. *In Proc. ECCV*, pp. 224-237.

Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D., 2003. A search engine for 3D models. *ACM Trans. Graph*., 22(1), 83-105.

Geiger, A., Lenz, P., & Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. *In Proc. IEEE CVPR*, pp. 3354-3361.

Girshick, R., 2015. Fast R-CNN. *In Proc. IEEE CVPR*, pp. 1440-1448.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Bengio, Y., 2014. Generative adversarial nets. *In Proc. NIPS*, pp. 2672-2680.

Guan, H., Li, J., Yu, Y., Wang, C., Chapman, M., Yang, B., 2014. Using mobile laser scanning data for automated extraction of road markings. *ISPRS J. Photogramm. Remote Sens.*, 87, 93-107.

Guan, H., Li, J., Yu, Y., Ji, Z., Wang, C., 2015. Using mobile LiDAR data for rapidly updating road markings. *IEEE Trans. Intell. Transp. Syst.*, 16(5), 2457-2466.

Guan, H., Li, J., Cao, S., Yu, Y., 2016. Use of mobile LiDAR in road information inventory: A review. *Int. J. Image Data Fusion*, 7(3), 219-242.

Guo, Y., Sohel, F., Bennamoun, M., Lu, M., Wan, J., 2013. Rotational projection statistics for 3D local surface description and object recognition. *Int. J. Comput. Vis*., 105(1), 63-86.

He, B., Ai, R., Yan, Y., Lang, X., 2016. Lane marking detection based on convolution neural network from point clouds. *In Proc. IEEE ITSC*, pp. 2475-2480.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *In Proc. IEEE CVPR*, pp. 770-778.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. *In Proc. IEEE CVPR*, pp. 2961-2969.

Holgado-Barco, A., Riveiro, B., González-Aguilera, D., Arias, P., 2017. Automatic inventory of road cross sections from mobile laser scanning system. *Comput. Aided Civil Infrastruct. Eng*., 32(1), 3-17.

Homayounfar, N., Ma, W. C., Kowshika Lakshmikanth, S., Urtasun, R., 2018. Hierarchical recurrent attention networks for structured online maps. *In Proc. IEEE CVPR*, pp. 3417-3426.

Isola, P., Zhu, J. Y., Zhou, T., Efros, A. A., 2017. Image-to-image translation with conditional adversarial networks. *In Proc. IEEE CVPR*, pp. 1125-1134.

McCornmac, J., Sarasua, W., Davis, W. *Surveying* 6th Edition. John Wiley, 2013.

Jaiswal, A., AbdAlmageed, W., Wu, Y., Natarajan, P., 2018. CapsuleGAN: Generative adversarial capsule network. *In Proc. ECCV 2018 Workshops*, pp. 526-535.

Jiang, M., Wu, Y., Zhao, T., Zhao, Z., Lu, C., 2018. PointSIFT: A sift-like network module for 3D point cloud semantic segmentation. arXiv preprint arXiv:1807.00652.

Johnson, A. E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell*., 21(5), 433-449.

Jung, H. G., Lee, Y. H., Kim, J., 2009. Uniform user interface for semiautomatic parking slot marking recognition. *IEEE Trans. Veh. Technol.*, 59(2), 616-626.

Jung, J., Che, E., Olsen, M. J., Parrish, C., 2019. Efficient and robust lane marking extraction from mobile lidar point clouds. *ISPRS J. Photogramm. Remote Sens.*, 147, 1-18.

Kang, Y., Roh, C., Suh, S. B., Song, B., 2012. A lidar-based decision-making method for road boundary detection using multiple Kalman filters. *IEEE Trans. Ind. Electron*., 59(11), 4360-4368.

Karpathy, A., Fei-Fei, L., 2015. Deep visual-semantic alignments for generating image descriptions. *In Proc. IEEE CVPR*, pp. 3128-3137.

Klokov, R., Lempitsky, V., 2017. Escape from cells: Deep KD-networks for the recognition of 3D point cloud models. *In Proc. IEEE ICCV*, pp. 863-872.

Knopp, J., Prasad, M., Willems, G., Timofte, R., Van Gool, L., 2010. Hough transform and 3D SURF for robust three dimensional classification. *In Proc. ECCV*, pp. 589-602.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. ImageNet classification with deep convolutional neural networks. *In Proc. NeurIPS*, pp. 1097-1105.

Kumar, P., McElhinney, C. P., Lewis, P., McCarthy, T., 2014. Automated road markings extraction from mobile laser scanning data. *Int. J. Appl. Earth Obs. Geoinf*., 32, 125-137.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. *In Proc. IEEE CVPR*, pp. 4558-4567.

Lee, S., Kim, J., Shin Yoon, J., Shin, S., Bailo, O., Kim, N., So Kweon, I., 2017. VPGNet: Vanishing point guided network for lane and road marking detection and recognition. *In Proc. IEEE CVPR*, pp. 1947-1955.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature*, 521(7553), 436-444.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B. 2018. PointCNN: Convolution on X-transformed points. *In Proc. NeurIPS*, pp. 820-830.

Li, Y., Ma, L., Zhong, Z., Cao, D., Li, J., 2019a. TGNet: Geometric graph CNN on 3D point cloud segmentation. *IEEE Trans. Geosci. Remote Sens*., 58(5), 3588-3600.

Li, Y., Xiang, L., Zhang, C., Wu, H., 2019b. Fusing taxi trajectories and RS images to build road map via DCNN. *IEEE Access*, 7, 161487-161498.

Li, Y., Ma, L., Zhong, Z., Liu, F., Cao, D., Li, J., Chapman, M. A., 2020. Deep learning for LiDAR point clouds in autonomous driving: a review. *IEEE Trans Neural Netw Learn Syst*. doi:10.1109/TNNLS.2020.3015992.

Li, X., Li, J., Hu, X., Yang, J., 2019c. Line-CNN: End-to-End traffic line detection with line proposal unit. *IEEE Trans. Intell. Transp. Syst.*, 21(1), 248-258.

Liang, M., Yang, B., Wang, S., Urtasun, R., 2018. Deep continuous fusion for multi-sensor 3D object detection. *In Proc. ECCV*, pp. 641-656.

Liang, J., Homayounfar, N., Ma, W. C., Wang, S., Urtasun, R., 2019. Convolutional recurrent network for road boundary extraction. *In Proc. IEEE CVPR*, pp. 9512-9521.

Lin, Y., Wang, C., Zhai, D., Li, W., Li, J., 2018. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS J. Photogramm. Remote Sens*., 143, 39-47.

Liu, W., Wen, Y., Yu, Z., Yang, M., 2016. Large-margin softmax loss for convolutional neural networks. *In Proc. ICML*, 2(3), pp. 7-17.

Luo, H., Wang, C., Wen, C., Cai, Z., Chen, Z., Wang, H., Li, J., 2015. Patch-based semantic labeling of road scene using colorized mobile LiDAR point clouds. *IEEE Trans. Intell. Transp. Syst.*, 17(5), 1286-1297.

Luo, R. C., Chiou, M., 2018. Hierarchical semantic mapping using convolutional neural networks for intelligent service robotics. *IEEE Access*, 6, 61287-61294.

Luo, Z., Li, J., Xiao, Z., Mou, Z. G., Cai, X., Wang, C., 2019a. Learning high-level features by fusing multi-view representation of MLS point clouds for 3D object recognition in road environments. *ISPRS J. Photogramm. Remote Sens*., 150, 44-58.

Luo, Z., Mohrenschildt, M. V., Habibi, S., 2019b. A probability occupancy grid based approach for real-time lidar ground segmentation. *IEEE Trans. Intell. Transp. Syst*., 21(3), 998-1010.

Luo, Z., Attari, M., Habibi, S., Von Mohrenschildt, M. 2019c. Online multiple maneuvering vehicle tracking system based on multi-model smooth variable structure filter. *IEEE Trans. Intell. Transp. Syst*., 21(2), 603-616.

Ma, L., Li, Y., Li, J., Wang, C., Wang, R., Chapman, M. A., 2018. Mobile laser scanned point-clouds for road object detection and extraction: A review. *Remote Sens*., 10(10), 1531-1564.

Ma, L., Li, Y., Li, J., Zhong, Z., Chapman, M. A., 2019a. Generation of horizontally curved driving lines in HD maps using mobile laser scanning point clouds. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*., 12(5), 1572-1586.

Ma, L., Li, Y., Li, J., Tan, W., Yu, Y., Chapman, M. A., 2019b. Multi-scale point-wise convolutional neural networks for 3D object segmentation from lidar point clouds in large-scale environments. *IEEE Trans. Intell. Transp. Syst*. doi: 10.1109/TITS.2019.2961060.

Ma, L., Li, Y., Li, J., Yu, Y., Junior, J. M., Gonçalves, W. N., Chapman, M. A., 2020. Capsule-based networks for road marking extraction and classification from mobile lidar point clouds. *IEEE Trans. Intell. Transp. Syst*. doi: 10.1109/TITS.2020.2990120.

Ma, L., Li, Y., Li, J., Junior, J. M., Gonçalves W. N., Chapman M. A., 2020. BoundaryNet: Extraction and completion of road boundaries with deep learning using MLS point clouds and satellite imagery. *IEEE Trans. Intell. Transp. Syst*. (under revision).

Masuda, T., 2009. Log-polar height maps for multiple range image registration. *Comput. Vis. Image Understand*, 113(11), 1158-1169.

Máttyus, G., Wang, S., Fidler, S., Urtasun, R., 2016. HD maps: Fine-grained road segmentation by parsing ground and aerial images. *In Proc. IEEE CVPR*, pp. 3611-3619.

Mathibela, B., Newman, P., Posner, I., 2015. Reading the road: Road marking classification and interpretation. *IEEE Trans. Intell. Transp. Syst*., 16(4), 2072-2081.

Maturana, D., Scherer, S., 2015. VoxNet: A 3D convolutional neural network for real-time object recognition. *In Proc. IEEE IROS*, pp. 922-928.

Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M., 2019. EdgeConnect: Structure guided image inpainting using edge prediction. *In Proc. IEEE CVPR*, pp. 3265-3274.

Nguyen, A., Le, B., 2013. 3D point cloud segmentation: A survey. *In Proc. 6th IEEE Conf. Robot. Automat. Mechtron.,* pp. 225-230.

Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., 2002. Shape distributions. *ACM Trans. Graph*., 21(4), 807-832.

Otsu, N., 1979. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern*., 9(1), 62- 66.

Paquet, E., Rioux, M., Murching, A., Naveen, T., Tabatabai, A., 2000. Description of shape information for 2D and 3D objects. *Signal Process. Image Commun.*, 16(1-2), 103-122.

Powers, D. M., 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learn. Tech*., 2(1), pp. 37–63.

Pu, S., Rutzinger, M., Vosselman, G., Elberink, S. O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS J. Photogramm. Remote Sens*., 66(6), S28-S39.

Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L. J., 2016. Volumetric and multi-view CNNs for object classification on 3D data. *In Proc. IEEE CVPR*, pp. 5648-5656.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. *In Proc. IEEE CVPR*, pp. 652-660.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *In Proc. NeurIPS*, pp. 5099-5108.

Ravi, R., Cheng, Y. T., Lin, Y. C., Lin, Y. J., Hasheminasab, S. M., Zhou, T., Habib, A., 2019. Lane width estimation in work zones using lidar-based mobile mapping systems. *IEEE Trans. Intell. Transp. Syst*. Doi: 10.1109/TITS.2019.2949762.

Rastiveis, H., Shams, A., Sarasua, W. A., Li, J. 2020. Automated extraction of lane markings from mobile LiDAR point clouds based on fuzzy inference. *ISPRS J. Photogramm. Remote Sens.*, 160, 149-166.

Riegler, G., Osman Ulusoy, A., Geiger, A., 2017. OctNet: Learning deep 3D representations at high resolutions. *In Proc. IEEE CVPR*, pp. 3577-3586.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. *In Proc. MICCAI*, pp. 234-241.

Roynard, X., Deschaud, J. E., Goulette, F., 2018. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Int. J. Robot. Res*., 37(6), 545-557.

Rusu, R. B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. *In Proc. IEEE Conf. Robot. Autom. Mechatronics*, pp. 3212-3217.

Sabour, S., Frosst, N., Hinton, G. E., 2017. Dynamic routing between capsules. *In Proc. NeurIPS*, pp. 3856-3866.

Salti, S., Tombari, F., Di Stefano, L., 2014. SHOT: Unique signatures of histograms for surface and texture description. *Comput. Vis. Image Understand*., 125, 251-264.

Sasaki, K., Iizuka, S., Simo-Serra, E., Ishikawa, H., 2018. Learning to restore deteriorated line drawing. *Visual Comput*., 34(6-8), 1077-1085.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural networks*, 85-117.

Seif, H. G., Hu, X., 2016. Autonomous Driving in the iCity—HD maps as a key challenge of the automotive industry. *Engineering*, vol. 2, no. 2, pp. 159-162.

Serna, A., Marcotegui, B., 2013. Urban accessibility diagnosis from mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens*., 84, 23-32.

Shen, Z., Ma, X., Li, Y., 2018. A hybrid 3D descriptor with global structural frames and local signatures of histograms. *IEEE Access*, 6, 39261-39272.

Shi, W., Miao, Z., Debayle, J., 2013. An integrated method for urban main-road centerline extraction from optical remotely sensed imagery. *IEEE Trans. Geosci. Remote Sens*., 52(6), 3359-3372.

Shi, S., Wang, X., Li, H., 2019. PointRCNN: 3D object proposal generation and detection from point cloud. *In Proc. IEEE CVPR*, pp. 770-779.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Chen, Y., 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676), 354-359.

Simonovsky, M., Komodakis, N., 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. *In Proc. IEEE CVPR*, pp. 3693-3702.

Soilán, M., Riveiro, B., Martínez-Sánchez, J., Arias, P., 2017. Segmentation and classification of road markings using MLS data. *ISPRS J. Photogramm. Remote Sens.*, 123, 94-103.

Soilán, M., Sánchez-Rodríguez, A., del Río-Barral, P., Perez-Collazo, C., Arias, P., Riveiro, B., 2019. Review of laser scanning technologies and their applications for road and railway infrastructure monitoring. *Infrastructures*, 4(4), 58.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3D shape recognition. *In Proc. IEEE ICCV*, pp. 945-953.

Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M. H., Kautz, J., 2018. SPLATNet: Sparse lattice networks for point cloud processing. *In Proc. IEEE CVPR*, pp. 2530-2539.

Sun, J., Ovsjanikov, M., Guibas, L., 2009. A concise and provably informative multi-scale signature based on heat diffusion. *Comput. Graph. Forum*, 28(5), pp. 1383-1392.

Sun, Z., Wang, C., Wang, H., Li, J., 2013. Learn multiple-kernel SVMs for domain adaptation in hyperspectral data. *IEEE Geosci. Remote Sens. Lett*., 10(5), 1224-1228.

Sung, J., Jin, S. H., Saxena, A., 2018. Robobarista: Object part based transfer of manipulation trajectories from crowd- sourcing in 3D pointclouds. *In Robotics Research*, pp. 701-720.

Tan, W., Qin, N., Ma, L., Li, Y., Du, J., Cai, G., Li, J., 2020. Toronto-3D: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. *In Proc. IEEE CVPRW*, pp.202-212.

Tatarchenko, M., Dosovitskiy, A., Brox, T., 2017. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. *In Proc. IEEE ICCV*, pp. 2088-2096.

Tchapmi, L., Choy, C., Armeni, I., Gwak, J., Savarese, S., 2017. SEGcloud: Semantic segmentation of 3D point clouds. *In Proc. 3DV*, pp. 537-547.

Te, G., Hu, W., Zheng, A., Guo, Z., 2018. Rgcnn: Regularized graph cnn for point cloud segmentation. *In Proc. ACM International Conference on Multimedia*, pp. 746-754.

Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., Guibas, L. J., 2019. KPConv: Flexible and deformable convolution for point clouds. *In Proc. IEEE CVPR*, pp. 6411-6420.

Tveite, H., 1999. An accuracy assessment method for geographical line data sets based on buffering. *Int. J. Geogr. Inf. Sci.*, 13(1), 27-47.

Wan, R., Huang, Y., Xie, R., Ma, P., 2019. Combined lane mapping using a mobile mapping system. *Remote Sens.*, 11(3), 305-330.

Wang, P., Hunter, T., Bayen, A. M., Schechtner, K., González, M. C., 2012. Understanding road usage patterns in urban areas. *Sci. Rep*., 2, 1001-1007.

Wang, H., Luo, H., Wen, C., Cheng, J., Li, P., Chen, Y., Li, J., 2015a. Road boundaries detection based on local normal saliency from mobile laser scanning data. *IEEE Geosci. Remote Sens. Lett*., 12(10), 2085-2089.

Wang, J., Song, J., Chen, M., Yang, Z., 2015b. Road network extraction: A neural-dynamic framework based on deep learning and a finite state machine. *Int. J. Remote Sens*., 36(12), 3144-3169.

Wang, Y., Xie, Z., Xu, K., Dou, Y., Lei, Y., 2016. An efficient and effective convolutional auto-encoder extreme learning machine network for 3D feature learning. *Neurocomputing*, 174, 988-998.

Wang, W., Yu, R., Huang, Q., Neumann, U., 2018. SGPN: Similarity group proposal network for 3D point cloud instance segmentation. *In Proc. IEEE CVPR*, pp. 2569-2578.

Wang, Z., Jia, K., 2019. Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection. *In Proc. IEEE IROS*, pp. 1742-1749.

Wang, Z., Lu, F., 2019a. VoxSegNet: Volumetric CNNs for semantic part segmentation of 3D shapes. *IEEE Trans.Vis. Comput. Graphics*. 26(9), 2919-2930.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., Solomon, J. M., 2019b. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph*., 38(5), 1-12.

Wen, C., You, C., Wu, H., Wang, C., Fan, X., Li, J., 2019a. Recovery of urban 3D road boundary via multi-source data. *ISPRS J. Photogramm. Remote Sens*., 156, 184-201.

Wen, C., Sun, X., Li, J., Wang, C., Guo, Y., Habib, A. 2019b. A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens*., 147, 178-192.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3d ShapeNets: A deep representation for volumetric shapes. *In Proc. IEEE CVPR*, pp. 1912-1920.

Wu, W., Qi, Z., Fuxin, L., 2019. PointConv: Deep convolutional networks on 3D point clouds. *In Proc. IEEE CVPR*, pp. 9621-9630.

Xie, S., Liu, S., Chen, Z., Tu, Z., 2018. Attentional shape context net for point cloud recognition. *In Proc. IEEE CVPR*, pp. 4606-4615.

Xu, S., Wang, R., Zheng, H., 2016. Road curb extraction from mobile LiDAR point clouds. *IEEE Trans. Geosci. Remote Sens*., 55(2), 996-1009.

Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y., 2018. SpiderCNN: Deep learning on point sets with parameterized convolutional filters. *In Proc. ECCV*, pp. 87-102.

Yang, B., Fang, L., Li, J., 2013. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens*., 79, 80-93.

Yang, B., Liu, Y., Dong, Z., Liang, F., Li, B., Peng, X., 2017. 3D local feature BKD to extract road information from mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.*, 130, 329-343.

Yang, B., Luo, W., & Urtasun, R., 2018. Pixor: Real-time 3D object detection from point clouds. *In Proc. IEEE CVPR*, pp. 7652-7660.

Ye, C., Li, J., Jiang, H., Zhao, H., Ma, L., Chapman, M., 2019. Semi-automated generation of road transition lines using mobile laser scanning data. *IEEE Trans. Intell. Transp. Syst*., 21(5), 1877-1890.

Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L. J., 2019. GSPN: Generative shape proposal network for 3d instance segmentation in point cloud. *In Proc. IEEE CVPR*, pp. 3947-3956.

Yu, Y., Li, J., Guan, H., Jia, F., Wang, C., 2014. Learning hierarchical features for automated extraction of road markings from 3D mobile LiDAR point clouds. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*., 8(2), 709-726.

Yu, Y., Li, J., Guan, H., Wang, C., Wen, C., 2016. Bag of contextual-visual words for road scene object detection from mobile laser scanning data. *IEEE Trans. Intell. Transp. Syst.*, 17(12), 3391-3406.

Yu, Y., Guan, H., Li, D., Gu, T., Wang, L., Ma, L., Li, J., 2019. A hybrid capsule network for land cover classification using multispectral lidar data. *IEEE Trans. Geosci. Remote Sens. Lett.*, 7(7), 1263-1267.

Zai, D., Li, J., Guo, Y., Cheng, M., Lin, Y., Luo, H., Wang, C., 2017. 3-D road boundary extraction from mobile laser scanning data via supervoxels and graph cuts. *IEEE Trans. Intell. Transp. Syst*., 19(3), 802-813.

Zhang, Y., Xiong, Z., Zang, Y., Wang, C., Li, J., Li, X., 2019. Topology-aware road network extraction via multi-supervised generative adversarial networks. *Remote Sens*., 11(9), 1017-1036.

Zhou, L., Zhang, C., Wu, M., 2018a. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. *In Proc. IEEE CVPRW*, pp. 182-186.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Sun, M., 2018b. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv*:1812.08434.

Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. M.*, 5(4), 8-36.

Appendix A

Point-wise Semantic Segmentation in Indoor Environments

To test the robustness and scalability of the developed MS-PCNN model, two highly dense indoor LiDAR datasets, i.e., ScanNet (Dai et al., 2017) and S3DIS (Armeni et al., 2017), were further used. Figure A.1 indicates several sample data, and Table A.1 details the descriptions of different test datasets.

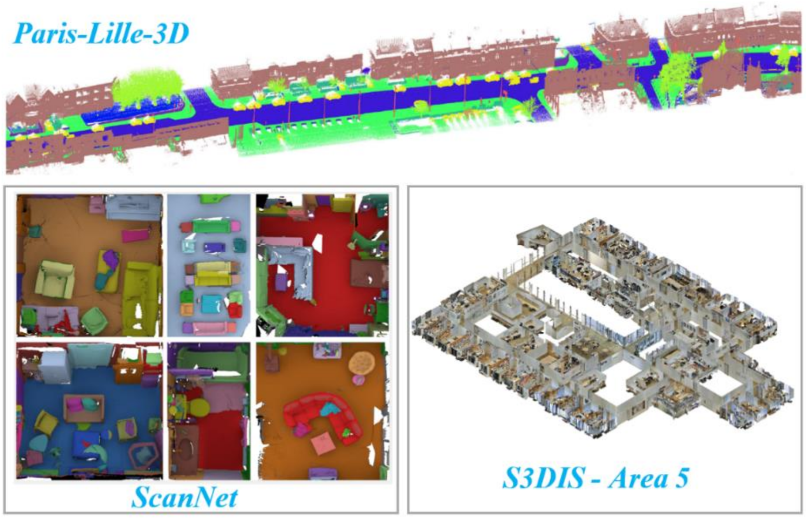


Figure A. 1 Samples of test datasets used in Chapter 3.

Table A. 1 Detailed descriptions of different test datasets used in Chapter 3.

Datasets	Types	No. of classes	No. of objects	No. of points	Scale	Sensor
Paris-Lille-3D	Urban	9	2,479	143.8 M	1,940 m	Velodyne HDL-32E
ScanNet	Indoor	20	>100,000	2.5 M (views)	-	RGB-D sensor
S3DIS	Indoor	13	6,005	215 M	6,020 m^2	RGB-D sensor

A1. ScanNet and S3DIS Datasets

The ScanNet dataset was generated from over 1,500 scans using RGB-D video streaming in indoor environments, including offices, apartments, and conference rooms. There is a total of over 100,000 CAD instances retrieved and placed on the surface reconstructions for semantic voxel labeling. This dataset was manually interpreted and labeled into 20 classes, such as Floor, Desk, Curtains, and Bathtubs.

The S3DIS dataset contains 13 object categories and 6,005 object instances with structural elements including Ceiling, Floor, Wall, Beam, Column, Window, Door, and moveable elements including Table, Chair, Sofa, Bookcase, Board, and others, which were collected in 11 scene categories (e.g., hallways and lobbies) and accordingly labeled. This dataset was generated from 6 different building sections with a total area of 6,020 m^2, 1.2 million of mesh faces, and a total number of 695 million 3D points, respectively. Both ScanNet and S3DIS datasets are commonly applied for object semantic segmentation tasks, which conduces to implement a comparative study between the MS-PCNN model proposed in this study and other existing methods.

A2. Segmentation results on ScanNet and S3DIS

According to the same hyperparameter settings and testing protocols as using the Paris-Lille-3D dataset, the optimal combination was determined as σ = 0.10 and k = 32 on both ScanNet and S3DIS test datasets. Additionally, the initial learning rate, batch size, momentum of Adam, dropout rate and epochs were [0.001, 16, 0.9, 0.5, 200] in ScanNet, and [0.001, 8, 0.9, 0.5, 250] in S3DIS, respectively, which can achieve the superior performance through multiple experiments. Other parameters, such as the radius in point density estimation and dimensions of the output channels, were experimentally determined through trial and error.

Table A. 2 Semantic segmentation results on ScanNet by using different methods.

Methods	**mIoU (%)**	**OA (%)**
ScanNet	30.6	-
PointNet++	38.3	71.4
SPLATNet	39.3	-
PointSIFT	41.5	86.2
PointCNN	52.2	85.1
MS-PCNN	56.8	87.6

Table A.2 shows the segmentation results on ScanNet by using different point-based deep learning methods. Note that, the ScanNet model as the pioneer DL-based method proposed for the ScanNet dataset achieves 30.6% mIoU, which is far from satisfactory in terms of robustness and segmentation accuracy. Although PointNet++ utilized farthest point sampling and multi-scale grouping algorithms to leverage local features from high-density point clouds, it only obtained

38.3% mIoU and 71.4% OA due to the non-uniform distributions and varying point densities in different input scenarios. The proposed MS-PCNN method is superior to both SPLATNet (Su et al., 2018) and PointSIFT even though they capture hierarchical and spatially aware features of input point clouds. Additionally, the MS-PCNN method outperforms PointCNN that ignores edge information among adjacent points in a local region. Compared to other methods, the proposed MS-PCNN model achieves the best performance in the sense of per-object segmentation accuracy.

Furthermore, the semantic segmentation performance on S3DIS by using different deep learning networks is presented in Table A.3. As can be perceived, MS+CU (Engelmann et al., 2017), G+RCU (Engelmann et al., 2017), and SegCloud (Tchapmi et al., 2017) slightly outperform PointNet, while PointNet++ is superior to them. Compared to DGCNN and SPGraph methods, the proposed MS-PCNN method outperforms them by a significant margin. Additionally, the MS-PCNN network achieves competitive performance compared with the PointSIFT (i.e., 67.2% mIoU for PointSIFT and 67.8% mIoU for MS-PCNN) on the S3DIS dataset. The experimental results indicate the strengths and robustness of the proposed MS-PCNN method in semantic segmentation, particularly in complex indoor environments with highly dense point clouds.

Table A. 3 Semantic segmentation performance on S3DIS by using different methods.

Methods	**mIoU (%)**	**OA (%)**
PointNet	47.7	78.6
PointNet++	54.5	81.0
MS+CU	47.8	79.2
G+RCU	49.7	81.1
SegCloud	48.9	-
DGCNN	56.1	84.1
SPGraph	62.1	85.5
PointSIFT	67.2	-
MS-PCNN	67.8	87.3

A3. Segmentation on ShapeNetPart

To evaluate the extensive applicability of the revised PointCONV convolutional operator in small-scale 3D point cloud scenes, the ShapeNetPart (Chang et al., 2015) dataset was further employed. ShapeNetPart datasets consist of 16,881 3D CAD instances, which are classified into

16 categories and 50 part annotations. The majority of 3D objects are labeled with two to five parts. Moreover, ground truths are labeled on down-sampled points on all categories. Thus, the part segmentation task was regarded as a point-wise segmentation problem.

Table A. 4 Segmentation results on ShapeNetPart by using different methods.

Methods	**mIoU (%)**	**Processing time (ms)**
PointNet	83.7	26.4
PointNet++	85.1	168.8
SPLATNet	84.6	260.1
DGCNN	85.1	95.6
SpiderCNN	85.3	184.2
PointCNN	86.1	77.3
MS-PCNN	86.6	69.0

According to the same hyperparameter settings and testing protocols as using the Paris-Lille-3D dataset, the optimal combination was ascertained as σ = 0.10 and k = 16 on the ShapeNetPart dataset. Additionally, the initial learning rate, batch size, momentum of Adam, dropout rate, and epochs were 0.001, 16, 0.9, 0.5, and 200, respectively, which can achieve superior performance through experimental tests. Table A.4 shows the point-wise segmentation results on ShapeNetPart by using various deep learning methods. Notably, MS-PCNN obtains an object instance average mIoU of 86.6%, which is on par with the state-of-the-art methods, e.g., PointNet, DGCNN, SpiderCNN, and PointCNN only considering xyz coordinate information of point clouds as inputs. Although 2D rendered images used in SPLATNet, MS-PCNN could provide more accurate and efficient segmentation results by directly consuming point clouds without data conversion. Furthermore, the processing time presented in Table A.4 indicates the time consuming for both forward and backward propagations through the entire testing dataset, which demonstrates the proposed MS-PCNN can achieve higher computational efficiency and lower time complexity compared to other DL-based methods.

Appendix B

Publications during PhD Study

- **Ma L**, Li Y, *Li J, Wang C, Wang R, Chapman M. A., 2018. Mobile laser scanned point-clouds for road object detection and extraction: A review. *Remote Sens*., 10(10), 1531-1564.

- **Ma L**, Li Y, *Li J, Zhong Z, Chapman M. A., 2019. Generation of horizontally curved driving lines in HD maps using mobile laser scanning point clouds. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*., 12(5), 1572-1586.

- **Ma L**, Li Y, *Li J, Tan W, Yu Y, Chapman M. A., 2019. Multi-scale point-wise convolutional neural networks for 3d object segmentation from lidar point clouds in large-scale environments. *IEEE Trans. Intell. Transp. Syst*. Doi: 10.1109/TITS.2019.2961060.

- **Ma L**, Li Y, *Li J, Yu Y, Junior J, Gonçalves W, Chapman M. A., 2020. Capsule-based networks for road marking extraction and classification from mobile lidar point clouds. *IEEE Trans. Intell. Transp. Syst*. Doi: 10.1109/TITS.2020.2990120.

- **Ma L**, Li Y, *Li J, Junior J, Gonçalves W, Chapman M. A., 2020. BoundaryNet: Extraction and completion of road boundaries with deep learning using MLS point clouds and satellite imagery. *IEEE Trans. Intell. Transp. Syst*. (under minor revision).

- **Ma L**, Chen Z, Li Y, Li J, Zhang D, Chapman M. A., 2019. Multispectral airborne laser scanning point-clouds for land cover classification using convolutional neural networks. *ISPRS Archives*, 42(2), 79-86.

- **Ma L**, Wu T, Li Y, Li J, Chen Y, Chapman M. A., 2019. Automated extraction of driving lines from mobile laser scanning point clouds. *Advances in Cartography and GIScience of the ICA*, 1(12).

- Ye C, Zhao H, **Ma L**, Li H, Chapman M. A., Junior J, Li J., 2020. Curved lane extraction from MLS point clouds towards HD maps. *IEEE Trans. Intell. Transp. Syst*. (under major revision).

- Li Y, **Ma L**, Zhong Z, Cao D, Li J., 2019. TGNet: Geometric graph CNN on 3D point cloud segmentation. *IEEE Trans. Geosci. Remote Sens*., 58(5), 3588-3600.

- Li Y, **Ma L**, Zhong Z, Liu F, Cao D, Li J. Chapman M. A., 2020. Deep learning for LiDAR point clouds in autonomous driving: a review. *IEEE Trans Neural Netw Learn Syst.* doi:10.1109/TNNLS.2020.3015992.

- Li Y, **Ma L**, Tan W, Sun C, Cao D, Li J., 2020. GRNet: Geometric relation network for 3D object detection from point clouds. *ISPRS J. Photogramm. Remote Sens*., 165, pp. 43-53.

- Osco LP, Ramos APM, et al., Li J, **Ma L**, Gonçalves WN, Junior JM, Creste JE, 2020. A machine learning framework to predict nutrient content in valencia-orange leaf hyperspectral measurements. *Remote Sens*., 12(6), pp. 906-927.

- Yu Y, Guan H, Li D, Gu T, Wang L, **Ma L**, Li J., 2019. A hybrid capsule network for land cover classification using multispectral lidar data. *IEEE Trans. Geosci. Remote Sens. Lett*., 7(7), 1263-1267.

- Chen Y, Wang S, Li J, **Ma L**, Wu R, Luo Z, Wang C., 2019. Rapid urban roadside tree inventory using a mobile laser scanning system. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*., 12(9), 3690-3700.

- Ye C, Li J, Jiang H, Zhao H, **Ma L**, Chapman M. A., 2019. Semi-automated generation of road transition lines using mobile laser scanning data. *IEEE Trans. Intell. Transp. Syst*., 21(5), 1877-1890.

- Zhou M, **Ma L**, Li Y, Li J, 2018. Extraction of building windows from mobile laser scanning point clouds. *IEEE IGARSS 2018*, pp. 4308-4311.

- Tan W, Qin N, **Ma L**, Li Y, Du J, Cai G, Li J., 2020. Toronto-3D: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. *In Proc. IEEE CVPRW*, pp.202-212.

- Fatholahi S. N., Gu Y, Liu M, **Ma L**, Chen Y, Li J., 2020. Estimating pm 2.5 concentrations in British Columbia, Canada during wildfire season using satellite optical data. *ISPRS Annals*, 1, 71-79.

- Li Y, **Ma L**, Huang Y, Li J, 2018. Segment-based traffic sign detection from mobile laser scanning data. *IEEE IGARSS 2018*, pp. 4611-4614.

- Zhong Z, Li J, **Ma L**, Jiang H, Zhao H., 2017. Deep residual networks for hyperspectral image classification. *IGARSS 2017*, pp. 1824-1827.

Waiver of Copyright

IEEE, as the publisher of the three manuscripts fully or partly adopted in Chapters 2, 3, 4, and 5, allow the reuse of published papers in the thesis without formal permissions. Thus, the waivers of copyright from IEEE are achieved by the following statement:

Policy Regarding Thesis/Dissertation Reuse, from IEEE Copyright Clearance Center

> "*The IEEE does not require individuals working on a thesis to obtain a formal reuse license; however, you may print out this statement to be used as a permission grant*".