CLASSIFICATION OF POINT CLOUDS ACQUIRED THROUGH MOBILE LASER SCANNER IN URBAN AREAS USING GEOMETRIC AND SHAPE FEATURES

KENTSEL ALANLARDA MOBİL LAZER TARAYICI İLE ELDE EDİLEN NOKTA BULUTLARININ GEOMETRİK VE ŞEKİL ÖZELLİKLERİ KULLANILARAK SINIFLANDIRILMASI

SEMANUR SEYFELİ

ASSOC. PROF. DR. ALİ ÖZGÜN OK

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfilment to the Requirements

for the Award of the Degree of Master of Science

in Geomatics Engineering

2021

ABSTRACT

CLASSIFICATION OF POINT CLOUDS ACQUIRED THROUGH MOBILE LASER SCANNER IN URBAN AREAS USING GEOMETRIC AND SHAPE FEATURES

Semanur SEYFELİ

Master of Science, Department of Geomatics Engineering

Supervisor: Assoc. Prof. Dr. Ali Özgün OK

May 2021, 67 pages

The mobile laser scanners (MLS) serve as a high density, high accurate and faster data collection method for urban areas from a street-level surveying perspective. The major processing phases of a MLS point cloud classification can be considered as (*i*) the neighborhood selection, (*ii*) the feature extraction, and finally (*iii*) the classification. Since MLS point clouds are poor in terms of the attributes, the classification phase must be supported by using features derived from the local neighborhood relations between points in a dataset.

This thesis deals with the point-based supervised classification of point clouds acquired through a vehicle-based MLS system in urban areas using local geometric and shape features. The developed approaches are tested using the benchmark dataset representing the Technical University of Munich (TUM) City Campus. The local features of each

point in the point cloud were extracted and evaluated through three different neighborhood definitions, i.e. spherical, cylindrical and the k-nearest neighbor. As the classification strategy, the Random Forest (RF) classifier that has been preferred in quite a few studies dealing with MLS classification is applied, and is successively tested for the point-based supervised classification. A total of 8 classes are involved during the classification: artificial terrain, natural terrain, high vegetation, low vegetation, building, hardscape, artifact and vehicle. The results were evaluated as classification for all three local neighborhood types with different parameters, and for various combinations of features. The results achieved were compared with three previous studies utilizing the same data set in the literature, and a combination of the features from different neighborhood information increased the overall results at least 4% with considerable improvements (up to 40%) for the producer's accuracies of multiple classes.

Keywords: mobile laser scanning, point clouds, classification, feature extraction, geometric features, shape features, random forest, point neighborhood.

ÖZET

KENTSEL ALANLARDA MOBİL LAZER TARAYICI İLE ELDE EDİLEN NOKTA BULUTLARININ GEOMETRİK VE ŞEKİL ÖZELLİKLERİ KULLANILARAK SINIFLANDIRILMASI

Semanur SEYFELİ

Yüksek Lisans, Geomatik Mühendisliği Bölümü

Tez Danışmanı: Doç. Dr. Ali Özgün OK

Mayıs 2021, 67 sayfa

Mobil lazer tarayıcılar (MLS), kentsel alanlar için sokak düzeyinde ölçüm yapan bir perspektifte yüksek yoğunlukla, yüksek doğrulukta ve hızlı veri toplama yöntemi olarak hizmet etmektedir. MLS nokta bulutu sınıflandırmasının ana işlem aşamaları, (i) komşuluk seçimi, (ii) özellik çıkarımı ve son olarak (iii) sınıflandırma olarak düşünülebilir. MLS nokta bulutları öznitelik açısından zayıf olduğundan, veri setindeki noktalar arasındaki yerel komşuluk ilişkilerinden türetilen özellikler kullanılarak sınıflandırma aşaması desteklenmelidir.

Bu tez çalışması, kentsel alanlarda araç tabanlı MLS aracılığıyla elde edilen nokta bulutlarının geometrik ve şekil özellikleri kullanılarak nokta tabanlı kontrollü sınıflandırılmasını ele almaktadır. Geliştirilen yaklaşımlar, Münih Teknik Üniversitesi (TUM) Şehir Kampüsü'nü temsil eden değerlendirme/kıyaslama veri seti kullanılarak test edilmiştir. Nokta bulutundaki her nokta için yerel geometrik özellikler çıkarılmış ve küresel, silindirik ve k-en yakın komşuluk olmak üzere üç farklı komşuluk tanımı ile değerlendirilmiştir. Sınıflandırma yaklaşımı için, MLS sınıflandırması ile ilgili pek çok çalışmada tercih edilen Rastgele Orman (RF) sınıflandırıcı uygulanmış ve nokta tabanlı kontrollü sınıflandırma ile test edilmiştir. Sınıflandırma sırasında yapay arazi, doğal arazi, yüksek bitki örtüsü, düşük bitki örtüsü, bina, sert peyzaj, yapay yapı ve araç olarak toplam 8 sınıf yer almaktadır. Sonuçlar, her üç yerel komşuluk tipi için farklı parametreler ve özellik kombinasyonları için sınıflandırma olarak değerlendirilmiştir. Elde edilen sonuçlar, literatürde aynı veri setini kullanan ve daha önce yapılan üç çalışma ile karşılaştırılmış ve farklı komşuluk bilgilerinden gelen özelliklerin kombinasyonu, birden fazla sınıfın üretici doğruluğu için önemli iyileştirmelerle (%40'a kadar) beraber genel sonuçları en az %4 oranında artırmıştır.

Anahtar Kelimeler: mobil lazer tarama, nokta bulutları, sınıflandırma, özellik çıkarma, geometrik özellikler, şekil özellikleri, rastgele orman, nokta komşuluğu.

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my supervisor Assoc. Prof. Dr. Ali Özgün Ok, for his dedicated attention and his efforts.

I would like to thank other committee members of my thesis, Assoc. Prof. Dr. Nusret Demir, Assoc. Prof. Dr. Sultan Kocaman Gökçeoğlu, Assoc. Prof. Dr. Fatih Nar and Assoc. Prof. Dr. Saygın Abdikan for their valuable comments and contributions.

I would like to thank Dr. Marcus Hebel and Joachim Gehrung of Fraunhofer-IOSB for providing the MLS point clouds.

I am thankful to all my close friends Ezgi Burçin Gazel and Merve Cansu Öztürk for their valuable friendship.

To my family, thanks for your love and support.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LiDAR:…………………………………………………..Light Detection and Ranging

3D:……………………………………………………………..…..…..3-Dimensional

SLS:……………………………………………….....…..Spaceborne Laser Scanning

ALS:……………………………..…………………......Airborne Laser Scanning

TLS:………………………...………………………..…..Terrestrial Laser Scanning

MLS:……………………………………………………….. Mobile Laser Scanning

kNN:……………………………………………..….… k- Nearest Neighborhood

RF:………………………………………………………………….…..Random Forest

GNSS:…………………………………….….....Global Navigation Satellite System

IMU:……………………………………………………..Inertial Measurement Unit

DMI:………………………………………………..…...Distance Measuring Indicator

TOF: ……………………………………………………..……………...Time of Flight

FOV:……………………………………………………………….…Field of View

GPS:…………………………………………………….…Global Positioning System

RANSAC:………………………………………………. Random Sample Consensus

DTM:……………………………………………..……..…Digital Terrain Model

2D:………………………………………………………..………2-Dimensional

PCA: ……………………………………………..…Principal Component Analysis

SVM:………………………………………………………...Support Vector Machine

IoU:……………………………………………………...… Intersection over Union

CSF:……………………………………………………….. Cloth Simulation Filtering

CART:……………………………………………… Classification and Regression Tree

1. INTRODUCTION

1.1. Purpose and Scope

Leading-edge technology of Light Detection and Ranging (LiDAR) is one of the privileged methods in mapping and surveying domain due to its progressive ability to generate dense and highly accurate three-dimensional (3D) point clouds. Such ground-breaking data can be obtained through different platforms/sensors; and therefore, up till now, advanced systems including spaceborne laser scanning (SLS), airborne laser scanning (ALS), terrestrial laser scanning (TLS) and mobile laser scanning (MLS) were successfully developed. Amongst them, the MLS serves as a high density, high accurate and faster data collection method for urban areas from a street-level surveying perspective. For that reason, many leading applications in the fields of 3D city modeling [1], vegetation mapping [2], building information modeling [3], transportation infrastructure mapping [4,5], pedestrian detection [6], change detection [7], autonomous vehicle driving [8] are carried out based on MLS-derived information. For such applications, MLS data sets can be utilized exclusively or be supported with comparable data sets including ALS, TLS, aerial images, terrestrial images and so forth.

In general, a regular MLS system consists of multiple elements involving imaging, scanning, positioning and storage systems placed on a mobile platform such as a vehicle, boat, train, backpack, robot (cf. Chapter 2). Vehicle-based platforms are common in urban areas due to their relative speed in regular traffic; thus, resulting in production of massive datasets at a single day of data collection [9]. Since the manual processing of large amount of MLS data is laborious and time-consuming, timesaving automated techniques for the processing and interpretation of such data are crucial and necessary for the effective handling of point clouds derived from the MLS systems.

Until now, well-designed automated strategies have been proposed for the classification and modeling of MLS point clouds in urban environments. However, even though promising results were reported in a number of research studies, accurate and complete classification of MLS point clouds is of still concern. The ultimate goal is indeed to predict and finally assign a proper class label to each single point existing in the MLS point cloud after a particular classification strategy. The point cloud geometry (i.e. point neighborhood) forms key evidence that should be investigated, and for that reason, a number of segmentation- and point-based classification methods were proposed. The major processing phases of the classification of a MLS point cloud can be considered as (*i*) the neighborhood selection, (*ii*) the feature extraction, and finally (*iii*) the classification. Since MLS point clouds are poor in terms of the attributes, the classification phase can be supported and performed by using features derived from the local neighborhood relations between points in a dataset [10]. For this purpose, different strategies describing and evaluating the local neighborhood of a point, e.g. k-nearest neighborhood (kNN) method based on k closest points, cylindrical and spherical neighborhood designs based on a certain radii, were developed in the past; and employed accordingly during the inference/computation of the geometric- and shape-based features from the MLS point clouds. To accomplish the classification phase, a variety of classification approaches including the machine learning algorithms such as instance-based, probabilistic learning, rule-based learning, max-margin learning and ensemble learning, and deep learning algorithms were tested [10-12]. Besides, a number of benchmark datasets were published by several authors/institutes covering different urban regions while the published datasets differ for a number of criteria including the platform/sensor utilized, the point density collected, the number of classes involved, and the season of data acquired. In this respect, the outputs of the previous approaches developed are still limited due to extensive differences between the input datasets.

This thesis deals with the point-based supervised classification of point clouds acquired through MLS in urban areas using features dealing with the basic geometry and shape of a point neighborhood. The developed approaches are tested using the benchmark dataset representing the Technical University of Munich (TUM) City Campus [13]. The local features for each point in the point cloud were extracted and evaluated through three different neighborhood definitions, i.e. spherical, cylindrical and the k-nearest neighborhoods. For the classification approach, the Random Forest (RF) classifier that has been preferred in quite a few studies dealing with MLS classification is applied, and successively tested for the point-based supervised classification. In this thesis, a total of 8 classes are involved during the classification: artificial terrain, natural terrain, high vegetation, low vegetation, building, hardscape, artifact and vehicle. All implementation and processing of the framework presented in the thesis was developed using the MATLAB environment, and the open source 3D point cloud and mesh processing project, CloudCompare. Figure 1.1 shows a flowchart that briefly summarizes the framework implemented to perform the point-based supervised classification of MLS point clouds using geometric and shape features.

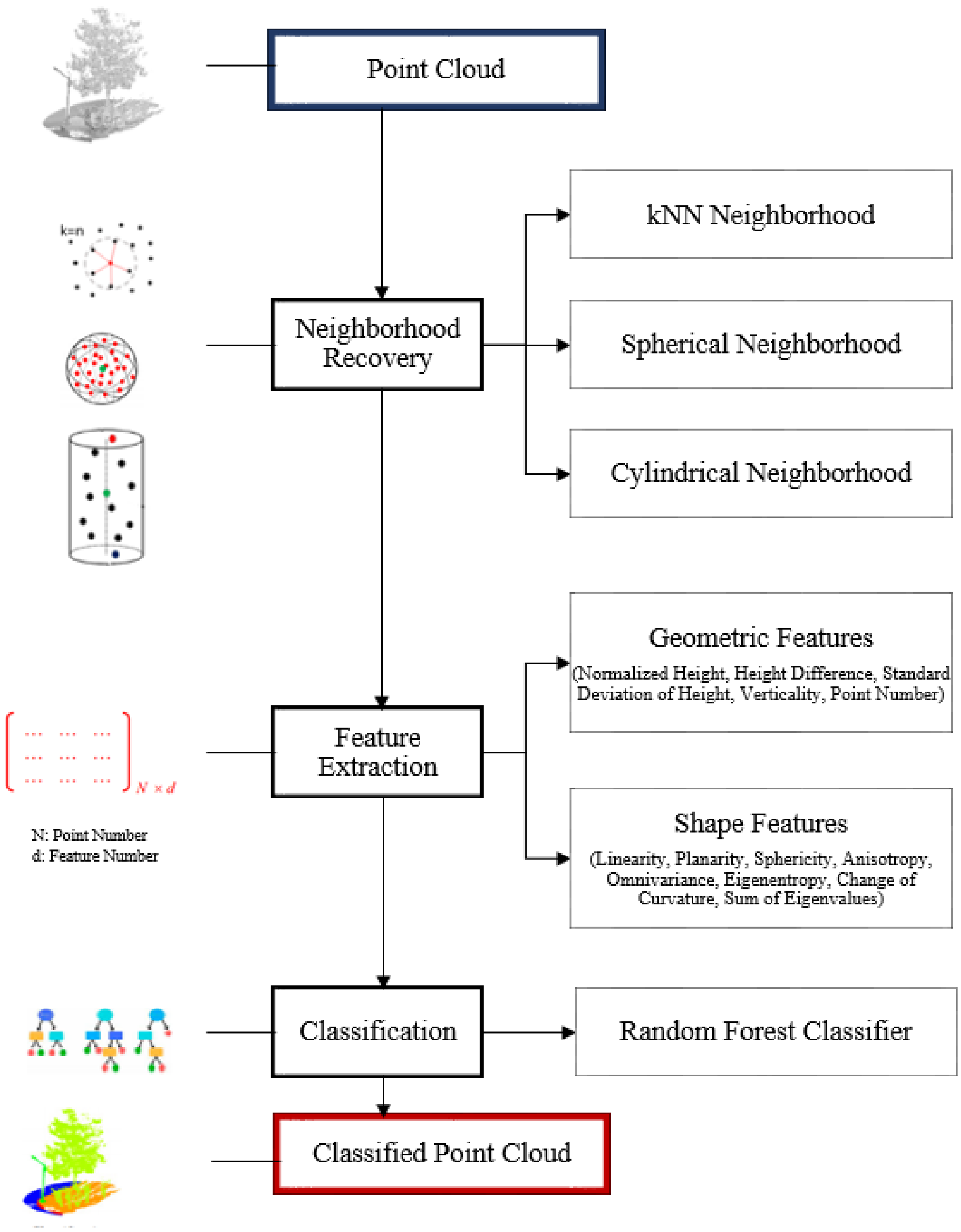


Figure 1.1: Classification workflow of the MLS point clouds.

1.2. Contributions

This thesis contributes to the related literature in the following aspects:

- The effects and importance of the geometric and shape features extracted from three different neighborhood types on the classification results were systematically evaluated. Only a few studies dealing with the assessments of such issue have been found in the literature.
- Combining important features of different neighborhood relations is proposed to improve the classification output while keeping the processing load in a manageable time-frame.
- The proposed framework is implemented using a developed MATLAB script.

1.3. Organization of the Thesis

This thesis is organized as six chapters. Chapter 2 presents the preliminaries of MLS, and also reviews the related work devoted for the segment-based and point-based classification of MLS data. Chapter 3 details the implemented framework as the recovery of the point neighborhood, extraction of features of a point, and lastly the classification of point clouds. The study area, the dataset used, the assessment strategy, and finally the parameter evaluation are all described in Chapter 4. Chapter 5 comprises of the results of this study, and the related discussion. Chapter 6 contains the conclusions reached, and states the recommendations for further research on this topic.

2. BASICS AND STATE-OF-ART

2.1. Mobile Laser Scanner (MLS)

The 3D point cloud data generated based on the LiDAR principles have been a vital input for 3D urban models; and therefore, such data are frequently involved in a variety of application studies such as urban planning, simulation, emergency response, mapping and visualization [14]. MLS is a high-resolution, LiDAR-based, fast and flexible 3D data acquisition method that generates 3D point clouds of objects around its surroundings. In this respect, dense and precise point cloud data can be easily collected by means of a MLS system. The point cloud collected may include objects usually observed in urban areas such as buildings, trees, sidewalks, roads, traffic signs, power cables, poles, and moving objects like cars and pedestrians [15].

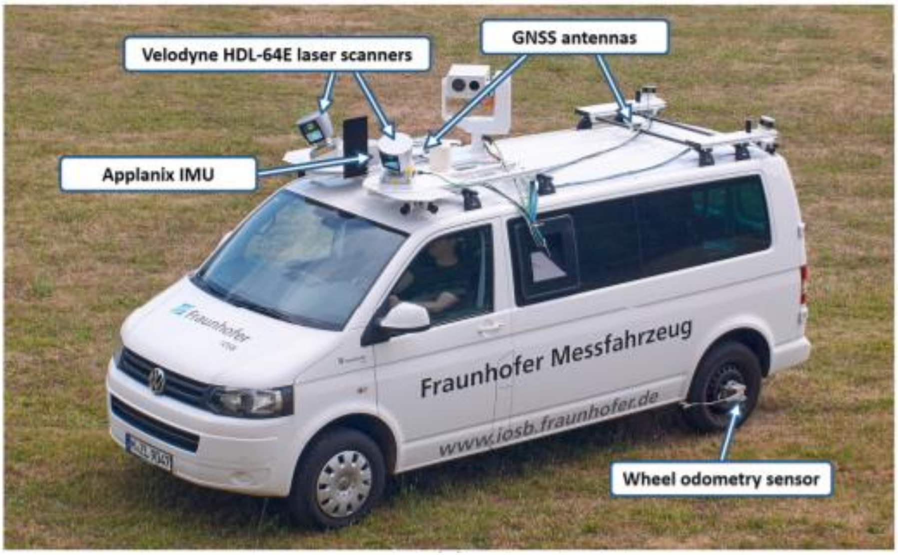
Most MLS systems consist of hardware responsible for the positioning (position and posture) as Global Navigation Satellite System (GNSS) receiver, Inertial Measurement Unit (IMU), Distance Measuring Indicator (DMI), 3D laser scanners, digital cameras, and a computer with data storage. All of this equipment is properly installed on a moving platform, including a vehicle [13], boat [16], train [17], backpack [9], robot [18] to collect 3D surface information along the path making headway (Figure 2.1). The mobile platform is under control, and treats all hardware utilized for the data collection as a single system. All equipment is precisely calibrated to maintain minimal errors between positioning, imagining and scanning systems [19].

(a)

(b)

(c)

(d)



(e)

Figure 2.1: The moving mobile platforms onboard (a) a boat [16], (b) a train [17], (c) a backpack [9], (d) a robot [18], and (e) a vehicle [13].

Laser scanners emit laser pulse within the near-infrared wavelength to the surface and collect back the returning pulse. It measures range as time-of-flight (TOF) or phase shift [15]. The reflected energy is used in both approaches to determine intensity of a point. The intensity is the return strength of the laser beam and represents the reflective properties of the object's surface [20]. The TOF is also known as pulse-based scanners emitting the laser pulse to the objects then receiving the reflected pulse. The distance between the object and the scanner is measured using recorded pulse traveling time. The phase shift and the transmitted number are used for the phase-based system to calculate the distance between the sensor and the related object [15]. Laser scanners with phase shifts are more accurate, but their measurement range is not that much long; hence, pulse-based scanners are generally used in MLS systems as it provides a farther measurement range [21]. Table 2.1 contains the specifications of some laser sensors used in mobile laser scanning.

Table 2.1: Specifications of laser sensors used by MLS systems [22 - 26]

	SICK LMS	**Riegl LMS-Q120i**	**VLP-16**	**Velodyne HDL-32E**	**Velodyne HDL-64E**
Range	30m (max 80m)	150m	1m to 100m	80m -100m	up to 120m
Accuracy	±35mm	20mm	±3cm	±2cm	< 2cm
Data Rate	75000 points/s	10000 points/s	300000 points/s	~1200000 points/s	~2200000 points/s
Vertical FOV	-	80° (-40°,+40°)	30° (-15°,+15°)	40° (+10° to -30°)	26.9° (+2.0° to -24.9°)
Horizontal FOV	100°/180°	-	360°	360°	360°
Wavelength	905 nm	905 nm	905 nm	905nm	905 nm

Raw MLS data collected may lack of texture information. Therefore, color images acquired from optical digital cameras along with the raw point cloud can be organized to visualize the point cloud considering the textural details [15]. As a result, such additional information can also be employed for better understanding of street scenes, and for change detection applications at city-level.

GPS, IMU and DMI provide the trajectory for the direct georeferencing of laser scanning data while the vehicle is in motion [15]. The GNSS gives exact positional information up to centimeter precision, and delivers three basic measurement observations: position, velocity, and time. However, as well known, the positioning accuracy would decline due to multipath phenomenon caused by high-rise obstacles in urban areas like tall trees, buildings, etc. To mitigate multipath effects and overcome the restriction of GNSS signal loss, an IMU is therefore employed to give the instant location, velocity, and attitude of an MLS system. The DMI, a precise odometer, is mounted at the rear wheel of the vehicle, and is used to measure the traveled distance [21]. While GNSS and IMU offer precise position and orientation measurements of a moving vehicle, DMI's supplementary positioning information is employed when GNSS data is unavailable [27]. Figure 2.1e shows the external components of a vehicle based MLS.

The MLS data can be used with other point data sources in urban applications, and in this respect, MLS can be compared with other advanced LiDAR systems (i.e. ALS, TLS, SLS). The main advantages of MLS point clouds are the fast and continuous collection of data with comparatively high point density. However, an ALS system has also capability to provide fast and permanent collection of data in terms of point clouds. Compared to the ALS, small details as well as surfaces that ALS fails to capture (e.g. underneath bridges and in tunnels) can be secured with the MLS systems [28]. Besides, due to the higher altitude of flight and controlled swath width, the point density for ALS systems is more uniform, but the MLS captures data more densely adjacent to the scanner path, and the point density drops farther away from the scanner path. On the one hand, the MLS systems can collect facades of the high structures like buildings from the direct view, but not top of them. On the other hand, the ALS can capture the direct view of the top of all structures including buildings and an oblique view of the vertical façade [19]. Figure 2.2 shows a visual comparison of the viewing geometry of the two popular systems. Apparently, the MLS pulses cannot reach behind certain objects (e.g. buildings, cars), and this fact may increase the occlusion related problems. Compared to ALS, the processing of MLS data may be slow due to ultra-high point density and comparatively large data file size [28].

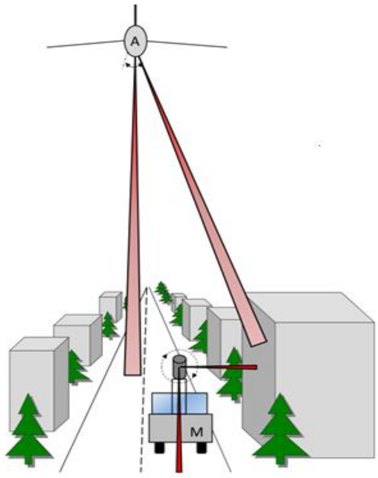


Figure 2.2: Comparison of ALS (A) and MLS (M) systems [19].

The TLS systems can provide some advantages over MLS systems. For the TLS system, scanners are mostly placed on a tripod, and scanning is performed in a static position. They provide additional options for setup locations except for active roads, and have the highest level of the detail for the point cloud. In contrast, the MLS systems measures on navigable roads, and they are preferred on a platform such as a vehicle for urban areas. Although static systems provide higher accuracy, mobile scanning has been widely accepted as an effective method of significantly faster data collection in relatively large urban environments while providing sufficiently accurate datasets [19].

In a different domain, SLS systems (e.g. ICESat, ICESat-2) are utilized to determine the altitude changes and certain topographic features of the Earth. However, because these systems operate on a specific polar orbit, the measurements are only available as a linear measurement of the ground track of the platform, and the laser footprint is quite large and the distance between footprints is quite long compared to any other laser scanning system. Therefore, SLS systems are preferred in specific applications near the poles (e.g. to monitor the elevation of ice sheets, glaciers, sea ice) where dense SLS point cloud is available due to significantly high coverage.

2.2. Related Work

Point clouds obtained through laser scanning contain surface related information of objects. The attributes of each point can simply be described as a 4D vector, namely the location information (X, Y, Z) and the intensity. However, a point in the point cloud does not contain any information about which object on the Earth's surface it belongs to. Consequently, there are two main questions to be answered; (*i*) "which points belong to a certain object?", and (ii) "what is the class label of that particular object?".

Assigning specific label information to every point (or segments of a point cloud) is called *classification of point clouds* [29]. As well known by the community, the methods for the classification can be broadly categorized as *supervised* or *unsupervised* [30]. In addition to this, classification strategies for point clouds can be grouped as *point-based* and *segment-based* [31]. No matter which category or strategy is preferred during the classification process, automated classification of point cloud datasets relies on certain features extracted for each point. A *feature* in this context describes any information regarding to points available in the dataset [29], and the feature sets are mainly constructed on the basis of four different types: geometric, radiometric, color and contextual. Local features (e.g. shape, size, roughness, density) of a point (or segment) are described as broad geometric features. MLS systems are capable of recording intensity and color information of points, and such information can be termed as radiometric and color features, respectively [32]. Finally, contextual features hold the spatial relationship amongst different points in a point cloud dataset [33]. After the features have been extracted from the point cloud, each point in the cloud must be assigned a class label. For that purpose, supervised classification methods (e.g. Random Forest, Support Vector Machines, AdaBoost) are usually implemented [10]. If the previous studies are evaluated, the RF method seems to be the most preferred approach due to its computational advantage and processing speed during labeling of the points surrounded by quite large MLS datasets in terms of the number of points.

Comprehensive summaries and overviews on the topic of point cloud classification can be found in several (recent) review papers, see e.g., [11], [32]. Since the main goal of this thesis is based on the point cloud classification using MLS datasets, this part provides a review about the previous studies that only involve MLS data in their framework (Table 2.2). This chapter explores the previous works, summarizing each study based on essential points such as the algorithm presented, the data used, and the results provided.

Table 2.2: Summary of the previous works using MLS datasets in their framework.

	Previous Work	**Data Source**	**Object of Interest**	**Method Applied**	**Limitation(s)**
Segment-based Classification	Wang et al. [34]	LiDAR DTM	Single tree	Voxelization	Crown might be over split because of the large trees
	Rutzinger et al. [35]	ALS MLS	Building wall	Region Growing Hough	Some of the walls could not be found caused by occlusion.
	Yokoyama et al. [36]	MLS	Pole-like objects	PCA	False segmentation in overcrowded areas
	El-Halawany et.al [37]	MLS	Road curb	PCA, Canny Edge Detection	Contains objects close to the road curb
	Rodríguez-Cuenca et al. [38]	MLS	Pole-like objects	Voxelization, RX Method	Occlusion and shadow
	Xiao et al. [39]	MLS	Vehicle	SVM, RF	Failures caused by under segmentation
	Soilán et al. [40]	MLS	Road markings	K-means Clustering, Neural Network	Misdetection due to fading of paint/occlusion caused by parked vehicles
	Sun et al. [41]	MLS	Artificial terrain, natural terrain, high vegetation, low vegetation, buildings, artifacts, vehicles, hardscape	Supervoxelization, RF	Failed to classify objects that have few samples

Point-based Classification	Demantke et al. [42]	ALS MLS TLS	Ground, building, low vegetation, high vegetation	Cylinder Neighbor	False classification due to scale
	Yang and Dong [43]	MLS	Pole-like objects	Spherical Neighbor Region Growing	Carts were misclassified as trees
	Weinmann et al. [10]	MLS	Wire, ground vegetation, cars facade, pole, motorcycles, pedestrians, traffic signs	k-Nearest Neighbor	Overall accuracy was low on unbalanced datasets
	Zheng et al. [44]	MLS	Facades, cars, motorcycle pedestrians, traffic signs	SVM (features from the cylindrical neighbor)	Overlapping of two or more objects
	Thomas et al. [45]	MLS	Pedestrians,car facade,groud motorcycles, traffic signs, vegetation	RF (features from the multiscale spherical neighbor)	Misclassified classes because of scale
	Wang et al. [30]	ALS MLS	Power line	RF (features from the multiscale neighbor)	Lower ALS classification results considering the point density
	Guo and Feng [46]	MLS TLS	Terrain, artifact building, car, vegetation, hardscape	Neural Network	Misclassified classes because of the small sample and similar appearance
	Atik et al. [12]	ALS MLS	Ground, building, vegetation	Machine Learning Algorithms (features from the multiscale spherical neighbor)	Low accuracy at small scales for high point density data.

2.2.1. Segment-based Classification

In segment-based classification, the point cloud is initially segmented, and afterwards a label is assigned to each segment. As a result, the classification's success is determined by the initial segmentation stage [10]. The segmentation approaches could be divided into four distinct groups: *edge-based*, *region growing*, *model fitting*, *clustering-based*. Besides, hybrid segmentation, the combination of at least two groups' of segmentation approaches, is also presented in the literature.

Edge-based segmentation approaches comprises of two stages: (*i*) edge detection is the process of extracting the boundaries of dissimilar regions, and (*ii*) grouping points within the regions to form the final segment [47]. Because of their simplicity, edge-based algorithms allow for fast point cloud segmentation, but they cannot be maintained when the point clouds have noise and uneven density [48]. *Region growing segmentation approaches* depend on selected seed points, and start segmentation from one or more initial points called seeds. Thereafter, they merge neighboring points with similar characteristics [47]. The region-growing methods are more noise-resistant than edge-based methods, although they are sensitive to seed point position [48]. The segments grow starting from the seed points up to pre-defined similarity criteria in the *bottom-up region growing* segmentation approach. However, for the other *top-down region growing* approach, all points are assigned to a group in the first step, and finally segments are created by splitting. *Model fitting segmentation approaches are* based on the fitting of geometric primitives such as plane, cylinder and sphere to the point cloud. Subsequently, they form the segment by identifying the points supporting the fitting procedure. In this context, two widely employed algorithms are the Hough Transform [49] and the Random Sample Consensus (RANSAC) [50] approach. *Clustering-based segmentation approaches* rely on similarity criteria. The unsupervised clustering algorithm *K*-means divides the point cloud into *K* number of unlabeled segments. However, as in most cases, determining a proper value of *K* is the main problem. Another method applied is based on the Mean-shift clustering algorithm [51]. Unlike the *K*-means [52] clustering algorithm, the number of clusters can be determined automatically with a given user-based parameters. Note that all segmentation approaches may suffer from over- and under-segmentation issues.

2.2.2. Review of Studies Comprising Segment-based Classification

A method to analyze vertical canopy structure and to conduct single tree modeling was presented in [34] for a test site located in the south of Germany, Kuernacher Wald, covered with 95% forest area. They segmented the whole area with the size of 20m×20m grids, and used a raster DTM to compute absolute object height with the help of normalized point cloud. First, a transformation matrix based on the thickness of each layer and the raster resolution of the horizontal surface was used to translate the normalized points in each grid into local voxel space. Considering the voxel as an image pixel, the normalized points in the voxel were defined as the gray value of a pixel. In this way, 2D horizontal projection images were created (Figure 2.3). Projected images at various height levels were subjected to the process of hierarchical morphological opening and closing. 3D prisma were created from 2D crown areas in each layer. As a result, a 3D crown model was reconstructed from all the crown prisms. Lower canopy layers that were overlapped with high canopies can also be found detectable by using the method presented, but the crown might over split because of the large trees.

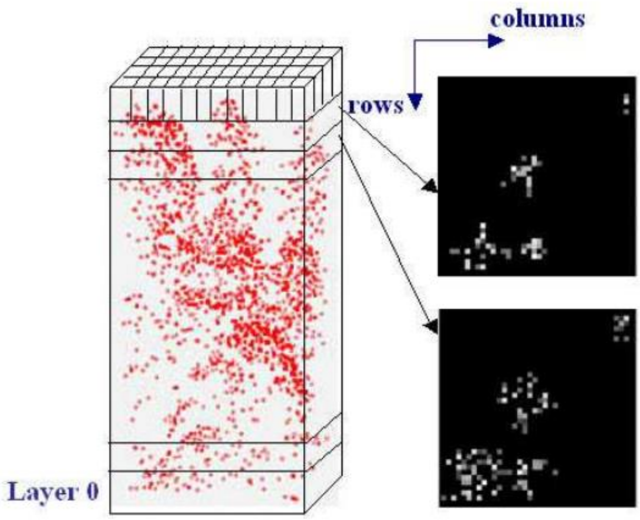


Figure 2.3: Representation of 2D horizontal projection images and local voxel [34].

Automatic wall extraction with ALS and MLS point clouds was examined by [35]. The test data involved 45 buildings (most of them were single story) in Enschede, Netherlands. First, region growing was applied to the point cloud to find planar areas. Seed surfaces were found with the help of 3D Hough transformation. Wall segments were extracted from MLS (165 segments) and ALS (262 segments) datasets. The results were compared with the cadastral map of the test area. Accordingly, the walls missing in the cadastral map were also detected. Some of the walls could not be extracted due to occlusion (between the sensor and the building) caused by trees and cars. While MLS provides more detailed information for the building facades, parts that cannot be collected with MLS were processed with the ALS data.

Pole-like object recognition from MLS data was studied in [36]. For the experiments, three MLS datasets belonging to the same urban area in Japan were utilized. First of all, the ground has been removed from the dataset. The algorithm consists of four stages (Figure 2.4). First, the input point cloud was segmented with the nearest neighbors' graph. Second, to improve the recognition rate, Laplacian smoothing was applied to each segment. With Laplacian smoothing, pole-like objects were transformed into one-dimensional-like thin pole shapes. However, the branching structures were not preserved. The endpoints preserving Laplacian smoothing has been applied to fix this problem, and after that, branching structures remained. Third, each point was classified with Principal Component Analysis (PCA). Finally, thresholding was used to extract segments of the pole-like object. They found that the result depends on the complexity of the measured area, and the dataset containing overcrowded and various objects caused false segmentation.

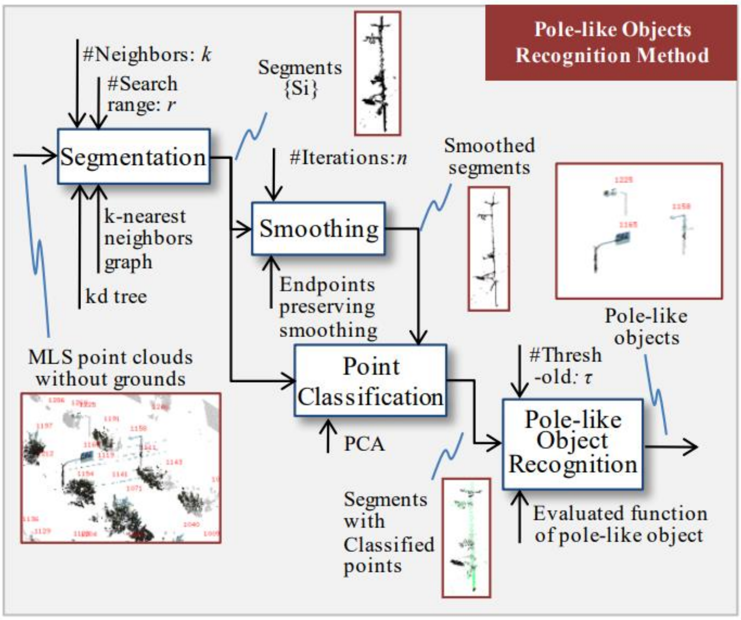


Figure 2.4: The recognition method for pole-like items [36].

A MLS data belonging to Elgin Street in Ottawa city to extract road curbs was used in [37]. Their curb extraction application took place in five steps. Initially, the point cloud was irregular, so the point cloud was organized with using the KD-tree algorithm. Later, all points were searched for spherical neighborhoods, and PCA was performed. The surface normal and normalized eigenvalues were calculated. Next, the point cloud was split and segmented as ground and non-ground based on the direction of the surface normal and relative size of eigenvalue. Parts of objects such as trees, poles and buildings in the separated ground were removed by fitting a plane. In the fourth stage, curb borders, street floor, and sidewalk were extracted. Finally, the curb was separated from the street floor and sidewalk by thresholding different parameters using Canny edge detection [53]. The authors concluded that removing objects close to the curb such as vehicles was found to be difficult.

Detecting vertical urban objects to classifying them as tree and man-made was studied in [38]. Their approach was applied to two different MLS datasets. Accordingly, a geometric index based on geometric features was used to separate vertical and horizontal objects from the datasets. Points belonging to the same surface were segmented. The dataset was divided into 3D vertical pillars to speed up the detection and extraction process. Each point was associated with a pillar. To do that, the point cloud was divided into grids. The pillar was then converted into voxels with fixed heights. The detection was continued starting from the voxel with the lowest height that was containing at least one point until the empty voxel coincided. Each pillar consisted of points between the lowest and the first empty, the others were omitted (Figure 2.5). With the RX anomaly detection algorithm, pole-like objects were automatically detected and unsupervised classification was completed.

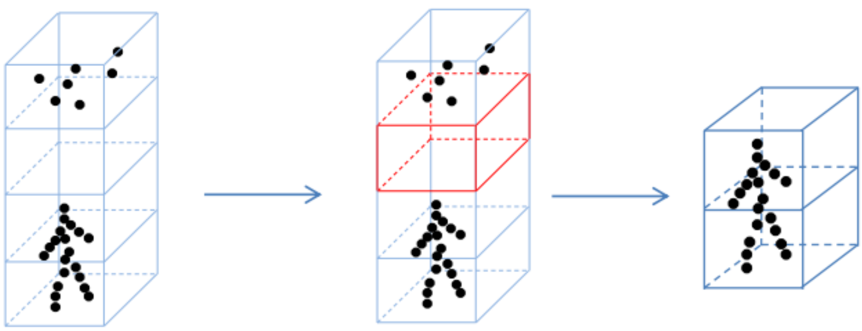


Figure 2.5: Limiting the pillar height up to an empty voxel [38].

Parking monitoring from MLS point clouds was studied in [39]. The dataset was divided into segments. Each segment was considered as an object, and 3D geometric features that describe shape and size were obtained from segments. Vehicle and non-vehicle were separated from each other by making a binary classification. The objects were often found to be incomplete because the scanning of MLS system was conducted from only a single direction. For that reason, a deformable vehicle model has been fitted, and the occluded parts were modeled to redo complete vehicle models for accurate localization. In addition to geometric features, the model parameters were handled as vehicle attributes. Support Vector Machine (SVM) and Random Forest (RF) were preferred as classifiers. The corresponding vehicles were detected and then compared in the feature space to detect vehicle changes in two datasets collected at different times (Figure 2.6). For this purpose, supervised learning was applied. The classifier was trained on the vehicle pairs. In the feature space, the same vehicle pairs and different vehicle pairs were compared, and the difference was used to construct a new training set. Two main issues, under-segmentation and occlusion were found to be the causes of false alarms on the final results.

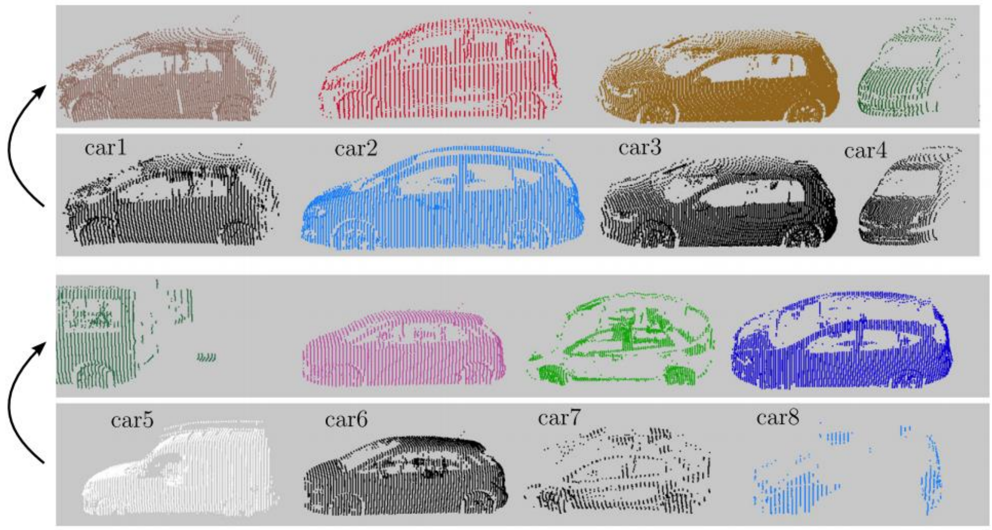


Figure 2.6: Change detection of the vehicles. Vehicles that have not been changed are in black, vehicles with a change are in blue color, and without corresponding vehicles are in white [39].

A road marking extraction and classification method from MLS data was proposed in [40] for smart cities to determine and update the road markings. Reflective points were extracted from the point cloud by segmenting the pavements via K-means clustering, and applying the intensity filtering to separate road markings. Road markings were detected with several image processing steps after producing a binary intensity based image. Two different sets of features were produced for each marking, and pedestrian crossing and arrows were classified through a neural network. As a result, misdetections were observed due to fading of paint or occlusion caused by parked vehicles. The resulting classified road marking were transferred to a Geographic Information System (GIS) for future studies.

The classification of MLS point cloud by extracting geometric features with supervoxel-based local context instead of individual points was presented in [41]. The data was split into 3D cubes for supervoxelization (Figure 2.7). Thus, the MLS point cloud was not affected by the uneven density problem, and the computational time was reduced. Geometric features were extracted based on segments. For each supervoxel, the local geometry tendency was determined using high pass filtering to leverage unique geometric characteristics. Supervised classification through height, density and eigen-based features were performed on the TUM Campus MLS1 dataset, half as training and half as testing, with a Random Forest classifier. As a result, uneven edges, such as the corners between the wall and the ground, were misinterpreted as zigzag connections, and objects with inadequate samples were misclassified.

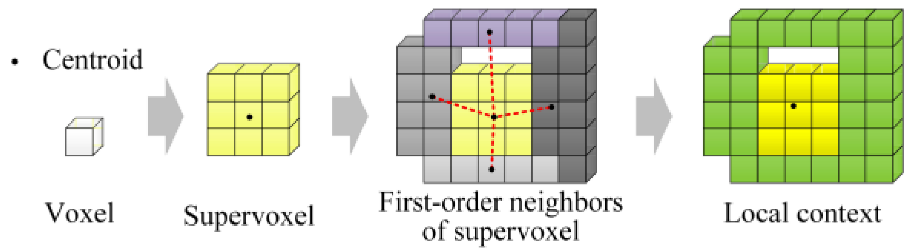


Figure 2.7: Representation of supervoxel [41].

2.2.3. Point-based Classification

In point-based classification, each point in the point cloud is labeled individually without segmenting the point cloud data. The geometric properties of points within a close neighborhood are used to classify them, and the uniqueness of the geometric characteristics depends on the neighborhood. Therefore, different strategies were proposed to define the local neighborhood of a point within the point cloud. However, the most commonly preferred definitions of local neighborhood are based on the spherical neighborhood [54], cylindrical neighborhood [55], and k-nearest neighbors [56]. Since the point density is variable in MLS-based point clouds due to specific reasons (occlusion, varying scanning angle, and varying distance), clarifying the local neighborhood information (length, size, shape etc.) is very important to achieve a satisfactory final result [42]. In most cases, neighborhood definitions can be accomplished with a single parameter (radius or the number of closest points), and prior knowledge about the data can be quite useful for the appropriate selection of that parameter. Besides, there are also approaches proposed for selecting the optimal neighborhood size as per the data, aiming to support different neighborhood sizes for different classes. Multiple scales of the point cloud can be utilized instead of a single scale to characterize the local 3D structure. This can be formed by spherical neighborhood and cylindrical neighborhood of different radii or a combination of these two types [57]. Since the point-based features are generally sensitive to point density, multiscale features might be useful to improve the classification performance.

2.2.4. Review of Studies Comprising Point-based Classification

A methodology to finding the optimal neighborhood radius based on a given point location were presented by [42]. Figure 2.8 represents the linear, planar or volumetric behavior of points and spreading of points as one, two or three dimensions. Dimension-based features are calculated by using cylindrical neighborhoods considering different radii. Two methods were used as entropy feature and similarity index, to finding the optimal neighborhood radius for each point, and evaluated on ALS, TLS and MLS datasets. Considering TLS and MLS evaluations, relatively large objects having low point density can be assumed planar for medium-sized neighborhoods, while objects having small widths can be viewed as planar or volumetric.

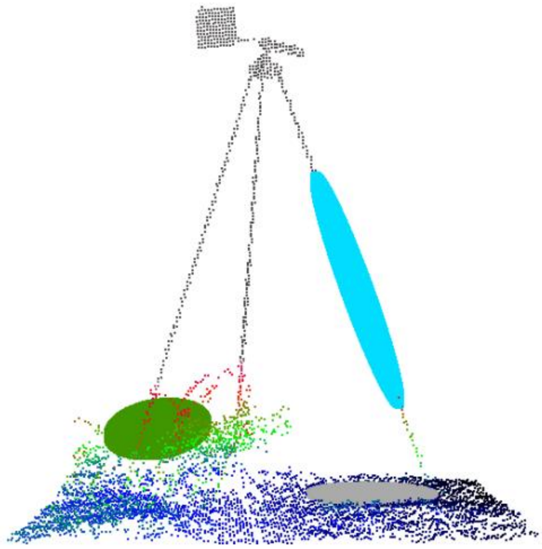


Figure 2.8: Ellipsoid reveals the linear, planar or volumetric behavior if the points are spread in one (blue), two (grey) or three dimensions (green) [42].

A point-based classification have been carried out in [43], and they applied a segmentation method to extract pole-like objects. PCA method was applied to find eigenvalues. Accordingly, dimensionality features like linear, planar and spherical were extracted. The optimal neighborhood selection was found to be important since fixed neighborhoods caused poor estimation results. Therefore, dimensionality features were formulated as an entropy function to finding the optimal neighborhood of a sphere. Point-based classification based on just three shape-based features was carried out by applying the SVM classifier to the MLS point cloud. Finally, a segmentation step was applied to the classified point cloud with region growing method using principal direction, normal vector and intensity of each labeled point. The presented method achieved successful results in classifying pole-like objects in complex areas. However, there were some misclassifications e.g. carts were classified as trees since the extracted shape features were quite similar to each other (Figure 2.9).

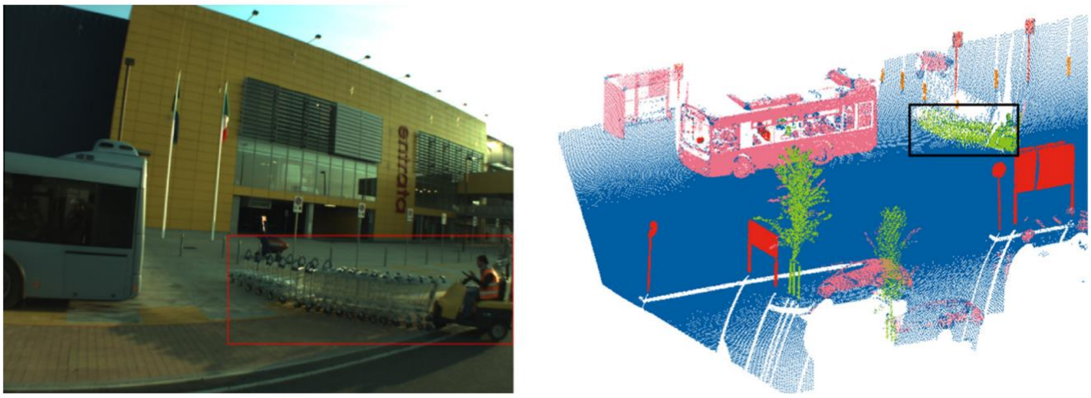


Figure 2.9: Misclassified part of pole-like objects [43].

A framework presented in [10] consisting of components including neighborhood selection, feature extraction, feature selection and classification. They employed the k-nearest neighbors of a given point as the neighborhood definition. First, eigenvalues were obtained by calculating the covariance matrix. Calculated eigenvalues were used to find both local surface variation and other eigen-based features. In total, 21 geometric 3D and 2D features were extracted. 7 different neighborhood definitions were formed by selecting five different k values (10, 25, 50, 75, 100), the optimal neighborhood by dimensionality-based, and eigen entropy-based scale selection. During the training process, the same number of samples was randomly collected for each class. 10 different classification methods were tested on Oakland 3D Point Cloud Dataset and Paris-rue-Madame database. They built a training set with 1000 training examples per class for both, and then utilized the remaining data as the test set. As a result, the Random Forest classifier gave a better result for eigen entropy-based scale selection and with 4 feature selection method. On imbalanced datasets, the ideal neighborhood size was found to be low, and the overall accuracy was shown to be low.

The height of objects was used in addition to the intensity value of the points to classify MLS point cloud in [44]. They assigned the height difference and the density of the points inside the cylinder to each point in the point cloud using the cylindrical neighborhood (Figure 2.10). The cylinder was created vertically, and its height was fixed for the entire dataset. The MLS dataset obtained from Paris-Rue-Madame in 2013 was utilized for the experiments. Although the dataset covers 26 classes in total, the SVM classifier has been selected by applying a point-based classification on just 5 classes out of 26 classes. They separated ground and non-ground points by filtering, and the radius of the cylinder, which is an important parameter, was fixed to 25 cm. Randomly selected 1000 points was chosen for each class for training. They found that the radius of the cylinder and overlapping of two or more objects affects the classification result.

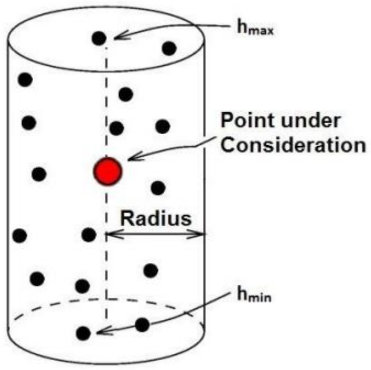


Figure 2.10: Vertical cylindrical neighborhood with a specified radius. The lowest and the highest points determine the height difference of a point [44].

The study proposed in [45] focused on semantic classification with a multiscale spherical neighborhood strategy utilized to extract geometric features. The framework is shown in the Figure 2.11. As argued in the study, scale selection was important as it affected the computational time to extract geometric features. According to their strategy, if the point density was high, the point cloud could be subsampled to create a uniform cloud, and therefore, they divided the point cloud into grids. They evaluated covariance-based features and color features (when available) for the experiments on different datasets such as Paris-Rue-Madame, Paris-Rue-Cassette. The classification was carried on 6 classes (for the Paris-Rue-Cassette dataset). As a training set, randomly distributed 1000 points per class were selected. Besides, three datasets from different environments were assessed. They preferred the Random Forest classifier as a base classification approach, and Intersection over Union (IoU) measure for evaluating the results. They achieved the mean accuracy of all classes as 82.84% and 65.50% for Paris-Rue-Madame and Paris-Rue-Cassette datasets, respectively.

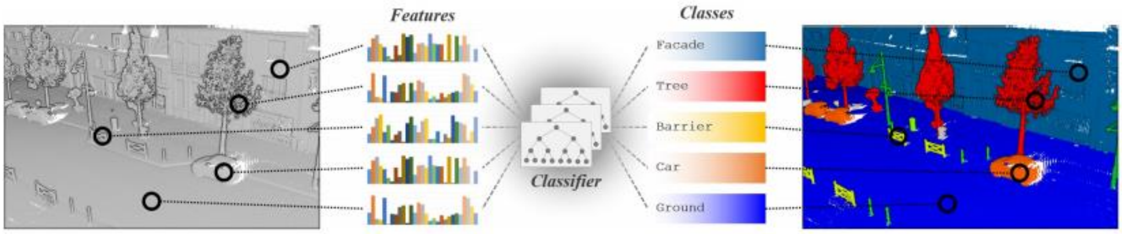


Figure 2.11: Semantic classification framework [45].

A study in the field of power line classification using ALS and MLS datasets was presented in [30]. They carried out supervised classification, and evaluated different neighborhood types, classification methods and feature sets. In total, 5 datasets (3 ALS, 2 MLS) including urban, forest and suburb areas were involved. The ground was filtered from the point cloud, and the powerline point candidate was selected from the points by selecting a height threshold (i.e. 4 m) above the ground. Four types of local neighborhoods were defined (the spherical neighborhood with a fixed radius, vertical cylindrical neighborhood, k nearest neighborhood and optimal k nearest neighborhood whose k value was obtained from eigenentropy-based scale selection). Six different radius parameters were investigated (i.e. 1m, 3m, 5m, 7m, 9m, and 11m). For the powerline classification, eigenvalues were calculated, and accordingly, 3 different feature sets were created. The first feature set included 26 features. PCA was applied to the first set, and those who exceeded 90% of the sum of variance created the second feature set. For the third set, 9 out of 26 features (linearity, scattering, anisotropy, changing of curvature, density, verticality, eigenvalue entropy, radius, and standard deviation of the Z values) were selected. 6 different classification methods were implemented, and RF classification with the multiscale vertical cylinder for the local neighborhood gave the best results. Considering the computation speed and results, the third feature set was preferred, and MLS results were found to be better than ALS results.

A multiscale and hierarchical deep learning approach to classify 3D point clouds (Figure 2.12) was developed in [46]. The Oakland and Europe datasets were preferred as MLS and TLS point cloud, respectively. First, a single-scale neural network was built to learn the deep properties of the point cloud on a single scale, and features were created by combining the features of the convolution layer and the max pool layer. A multiscale point cloud pyramid was created by subsampling the input point cloud to different scales. To build the input point cloud, a set number of n-nearest adjacent points were retrieved by treating each point as the center. The deep features were extracted by using the single-scale deep neural network at every scale and stacked into one feature vector. The classification was applied based on the extracted feature vector to determine the label of each point, and the process was run iteratively until the neural network parameters were optimized. The point labels were compared with the reference labels for the performance evaluation. Some misclassified pixels were encountered for some classes having small-sized samples and having similar appearance. As a result, a classification accuracy of over 90% was reached.

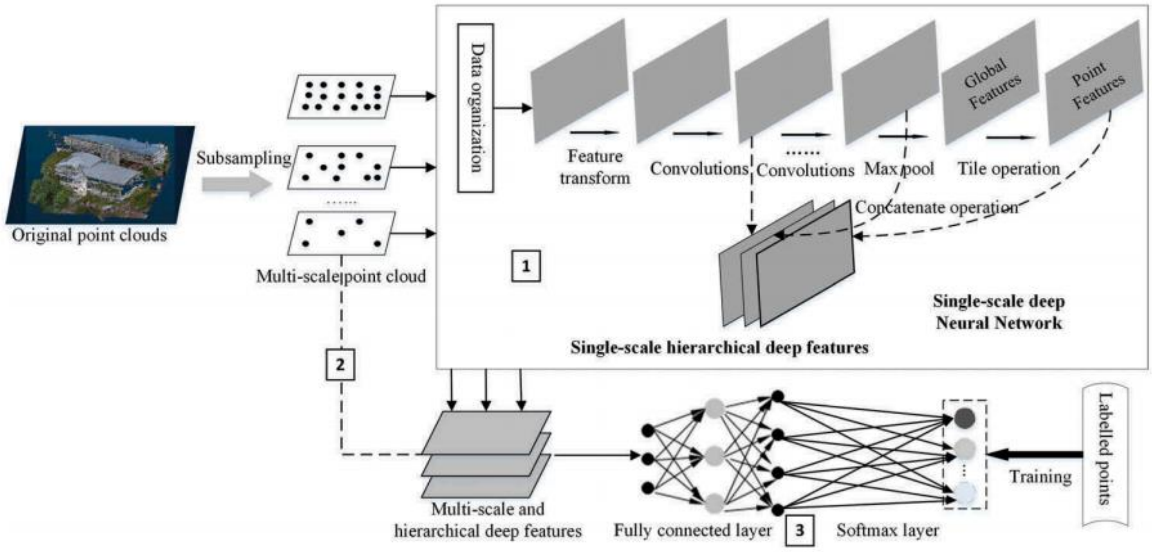


Figure 2.12: Workflow of multiscale and hierarchical deep neural network-based classification framework [46].

The classification performance of several machine learning methods upon several scales was tested in [12]. This process was carried on Dublin city, Vaihingen and Oackland3D datasets, and tested using different classifiers (RF, Linear Discriminant Analysis, Gaussian Naïve Bayes and Logistic Regression, Multi-Layer Perceptron, Decision Tree, SVM and k-nearest neighbors). Three classes (ground surface, building and vegetation) were considered in the classification. For each point, areas with different radii (0.5m, 1m, 1.5m, 2m, and 3m) were created through spherical neighborhood, and 13 features (produced from covariance matrix and height, roughness, normal change rate and volume density) were calculated. Classification accuracy was found to be the best with a 3 m scale for the Dublin city dataset which has the highest point density. However, since the point density is less in Vaihingen and Oakland3D datasets, the highest accuracy was obtained with different scales (1m and 1.5m). Thus, a relatively large radius would be preferred for dense point clouds, and the selected scale parameter (and point density) was found to be important for producing more accurate results.

3. PROPOSED FRAMEWORK

The proposed framework is divided into three main parts: (*i*) the recovery of local neighborhood of each 3D point in the cloud, (*ii*) the extraction of features for all points within the local neighborhood points, and (*iii*) the classification to assigning the labels based on the extracted features. The overall framework is depicted in Figure 1.1. In this chapter, the details of the approaches utilized are provided. Despite the fact that the segment-based approaches provide relatively less computation time and runs on (imperfect) homogeneous regions defined beforehand, the parameterization of the segmentation step (e.g. segmentation method, size, internal parameters) requires critical assessments. For that reason, in this thesis, point-based classification strategy is preferred.

The main algorithm steps are summarized as follows:

Algorithm

1. Let any point in the MLS point cloud be X_i. Each X_i point is taken as the center to create a local 3D structure.
2. Selecting the neighborhood type and defining the neighborhood.
3. Calculating features based on the points in the local neighborhood for each X_i
4. Automatically generating the training, validation and test sets.
5. Training the RF model with the generated training set
6. Validation of the RF model based on the validation dataset
7. Classifying the test set based on the RF model
8. Calculating the accuracy measures (confusion matrix, overall accuracy, kappa).

3.1. The Recovery of a Point Neighborhood

One of the imported factors of the classification of MLS point clouds is determining the local neighborhood points in the cloud, and different approaches can be exploited to collect local neighborhood information from a point cloud [30]. Spherical neighborhoods or cylindrical neighborhoods are formed according to a defined radius parameter, whereas k-nearest neighborhood is determined according to the number of closest points based on Euclidean distance in 3D space [57].

3.1.1. k-Nearest Neighborhood (kNN)

In this method, for a given point, k number of closest points is selected from the point cloud with the help of the Euclidean distance measure. Since the number of k points is fixed, the method adjusts the area of interest according to the point density, and the closest points are selected. Note that, kNN method does not contain a fixed spatial neighborhood size (Figure 3.1). For each point in the point cloud, the k closest points are determined, and the features described in Section 3.2 are extracted according to these points, and assigned as features to the point under evaluation.

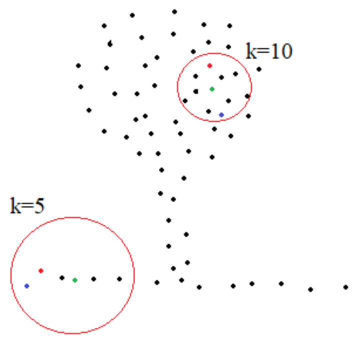


Figure 3.1: Two examples of kNN neighborhood definition (Green: point under consideration, Red: highest point, Blue: lowest point).

3.1.2. Spherical Neighborhood

For each point X_i in the point cloud, X_i is taken as the center of the local neighborhood. A sphere is created over a constant radius r (Figure 3.2).

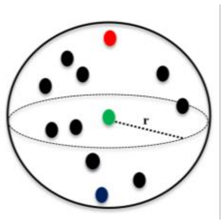


Figure 3.2: Spherical neighborhood with radius r (Green: a point under consideration, Red: highest point, Blue: lowest point).

Based on the points within this sphere, the features described in Section 3.2 are computed. Compared to the kNN, the size of the sphere does not change even the density may change for different points since the radius of the sphere is fixed (Figure 3.3).

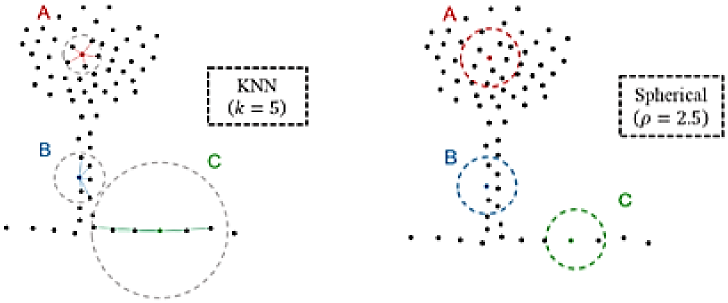


Figure 3.3: The difference between kNN and spherical neighborhood at different points.

3.1.3. Cylindrical Neighborhood

For each point X_i in the point cloud, X_i is taken as the center of the local neighborhood. The point cloud is projected into a 2D horizontal plane (XY-plane) to find the points inside the circle with a fixed radius *r* (Figure 3.4). To extract the features described in Section 3.2, the X, Y, Z coordinates of these points within the circle are used.

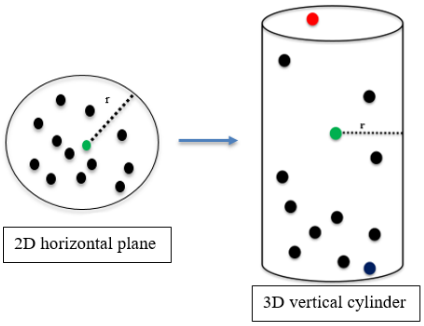


Figure 3.4: Representation of the point cloud in the XY-plane at radius r, and accordingly vertical cylindrical neighborhood with radius r (Green: point under consideration, Red: highest point, Blue: lowest point).

3.2. Feature Extraction

Since MLS point clouds are composed of only points with 3D coordinates, the classification step requires rich features to be extracted from either analyzing the local geometric structure of neighboring points in the cloud or overlaying with optical images (if available) for RGB values. If there is a lack of optical images along with the point cloud, geometric features have to be carefully generated from the close neighborhood of each point in the point cloud. Thereafter, using geometric characteristics collected from the spatial arrangement of 3D points in the immediate vicinity, a classifier can be run. In this thesis, local features around a point neighborhood are grouped as (*i*) 3D geometric properties and (*ii*) 3D shape features (Table 3.1).

Table 3.1: Local geometry and shape information collected from X, Y, and Z coordinates of surrounding points in a point cloud are used to generate features.

Feature Group	**Feature**	**Symbol**	**Formula**	**Reference**
Local geometric features	*Normalized height of a point*	H	$\lvert Z \rvert$	[12]
	Number of points in the neighborhood	N	-	[44]
	Height Difference	ΔH	$h_{max} - h_{min}$	[44]
	Standard Deviation of Height	σH	$\sqrt{\frac{1}{k}\sum_{j=1}^{k}(H_j - \mu H)^2}$	[10]
	Verticality	V	$1 - n_z$	[31]
Local shape features	*Linearity*	L_λ	$\frac{\lambda_1 - \lambda_2}{\lambda_1}$	[56]
	Planarity	P_λ	$\frac{\lambda_2 - \lambda_3}{\lambda_1}$	[10]
	Sphericity	S_λ	$\frac{\lambda_3}{\lambda_1}$	[10]
	Omnivariance	O_λ	$\sqrt[3]{\lambda_1\lambda_2\lambda_3}$	[12]
	Anisotropy	A_λ	$\frac{\lambda_1 - \lambda_3}{\lambda_1}$	[12]
	Eigenentropy	E_λ	$-\sum_{i=1}^{3}\lambda_i \ln(\lambda_i)$	[12]
	Sum of Eigenvalues	$\sum\lambda$	$\lambda_1 + \lambda_2 + \lambda_3$	[56]
	Change of local curvature	c_λ	$\frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$	[10]

3.2.1 Local Geometric Features

The first set of features is defined in terms of the geometric properties of the considered local neighborhood (i.e. normalized height, height difference, the standard deviation of height, number of points within the predefined neighborhood, and verticality). Considering the first geometric feature, the elevation of a point in the MLS point cloud is available w.r.t. to a reference surface, i.e. the ellipsoidal surface (Z). In this thesis, the elevation of each point is normalized ($|Z|$) according to the terrain height. For that purpose, the Clouth Simulation Filtering (CSF) plug-in [58] of the Cloud Compare software is applied to extract the ground surface height. According to the CSF algorithm, the original point cloud is inverted and clouth is passed over it. Looking at the interaction between the input points and the clouth node, the clouth is given its final shape. Produced clouth is used to separate the original point cloud as ground and non-ground. Next, the distance between the points in the original point cloud and the produced clouth mesh is calculated. Finally, the *height of each point* (H) from the mesh is assigned to the relevant points.

The local geometric features calculated other than the heights (H) are

- the *number of points* (N) in the local neighborhood,
- the *height difference* (ΔH), i.e. the difference between the maximum and the minimum height of the points within the local neighborhood,
- the *standard deviation of the heights* of the points within local neighborhood (σH),
- the *verticality* (V), i.e. the difference from vertical of the Z component of the normal vector.

Note that the feature *the number of points* (N) in the local neighborhood is not meaningful for the kNN neighborhood since k is always equal to N for all points. Therefore, this feature is not extracted when the kNN neighborhood is utilized.

3.2.1 Local Shape Features

The other features rely on eigenvalues λ_i with $i = 1,2,3$ which are represented as local shape features. These features are given by

- the *linearity* (L_λ) explains the variance with the only largest eigenvalue,
- the *planarity* (P_λ) explains the variance with the two largest eigenvalue,
- the *sphericity* (S_λ) explains the variance with the smallest eigenvalue,
- the *omnivariance* (O_λ) presents the volumetric point distribution,

- the *anisotropy* (A_λ) is the directional dependence,
- the *eigenentropy* (E_λ) indicates the order/disorder of 3D points within the local neighborhood,
- the *sum of eigenvalues* ($\sum\lambda$),
- the *change of local curvature* (c_λ).

Principal component analysis (PCA) provides the main aspects of the point distribution in 3D, and the magnitudes in the variation of the point distribution based on the center of gravity of the neighborhood [42]. PCA is used to generate a covariance matrix for each point within neighbors, and assessing the eigenvalues generated from the covariance matrix [27]. The coordinates of the neighboring points specify a shape with the eigenvalues obtained from the 3×3 covariance matrix. Eigenvalues are positive and sorted as $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$. The neighborhood defines a 3D line shape when one of the eigenvalues is large and the others are close to zero. If two eigenvalues have almost the same value and the other one is close to zero, the neighborhood forms a 3D plane. Other than this, if all of them are large, it forms a 3D spherical or fuzzy surface [44].

3.3. Random Forest Classification

Once the features are extracted, the next step is to apply a classification to assigning a label to each point in the point cloud. In this study, the RF classifier was employed with geometric features to classify MLS point cloud with a point-based classification point-of-view.

RF [59] is a collection of tree-structured classifiers that consists of a series of classification (or regression) trees generated on random samples of data. The RF uses a subset of randomly constrained and selected predictors to split each node in each classification tree. A tree is grown using random feature selection using a new training data set created from the original data set with replacement [59,60]. Three parameters are required to initialize RF algorithm (cf. Chapter 4). These are the number of trees to grow, the number of variables used to split each node and the minimum number of observations per tree leaf, respectively. A bootstrapped dataset is created by randomly selecting a sample from the original data set, and to perform the method, first, N bootstrap samples are drawn from 2/3 of the training data set. Bootstrapped the dataset and aggregate to make a decision is called bagging. Next, remaining 1/3 of the training data, also called out-of-bag (samples that are not included in the bootstrapped dataset), are utilized to test the error of the predictions. The out-of-bagged error shows how accurate the created random forest model is with out of bagged dataset. Then, from each bootstrap sample, an unpruned tree is created, with m predictors randomly picked as a subset of predictor variables at each node, and lastly, the optimal split among those variables is determined [61]. Classification and Regression Tree (CART) algorithm is utilized when creating trees [59]. Further details of RF method can be obtained from [59].

RF classifier has been utilized in a number of studies involving MLS data classification (see Table 2.2), and instead of relying on a decision based on a single tree, the random forest creates a prediction from each individual decision tree generated, and based on the majority voting of the predictions, and combines the predictions to produce a more accurate result. Figure 3.5 represents diagram of Random Forest classification algorithm.

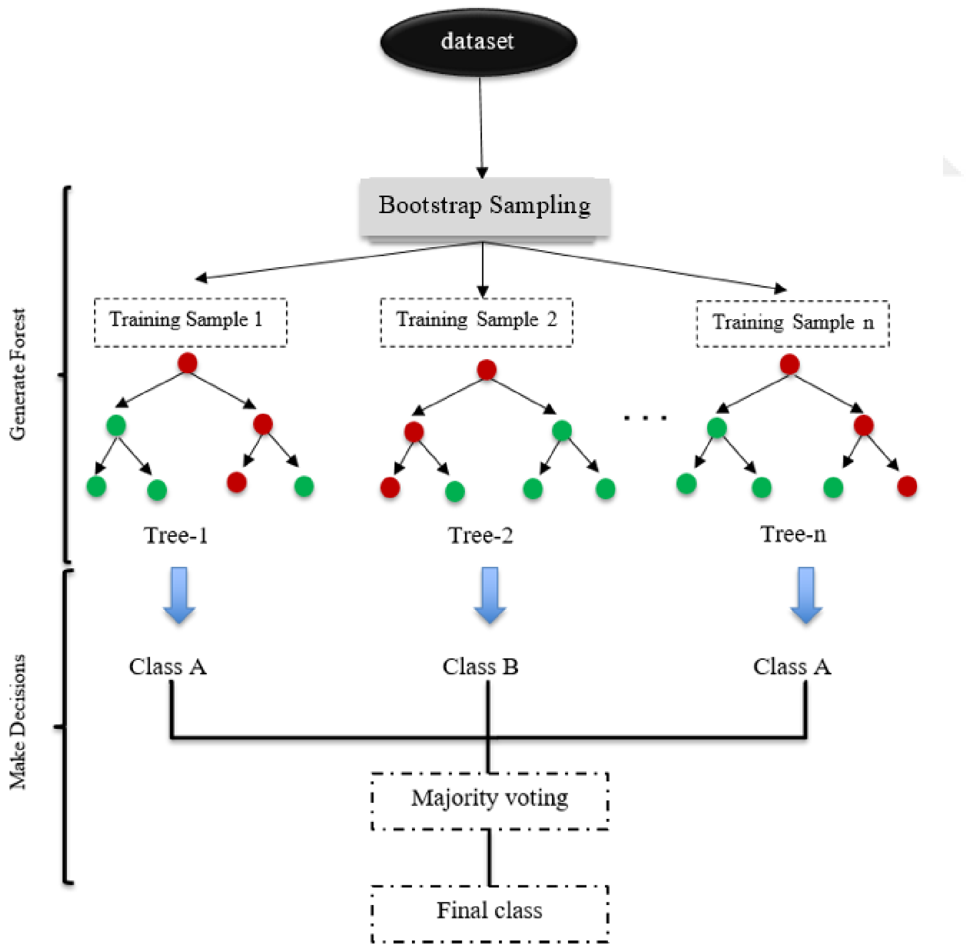


Figure 3.5: Random Forest classification algorithm.

4. DATASET AND EXPERIMENTS

This chapter is devoted to the description of the study area and the dataset, the evaluation strategy, and finally the assessments of the experiments conducted. In this thesis, the dataset utilized belongs to MLS point cloud acquired over a part of Germany, which is a dataset published as a benchmark dataset in 2016 to support the training and validating phases of the classification task of MLS point clouds (Table 4.1).

Table 4.1: Benchmark datasets available for MLS.

Datasets	**Sensors**	**# Points**	**#Classes**	**Year**
Oakland [33]	SICK LMS	1.6M	44	2009
Sydney Urban Objects [62]	Velodyne HDL-64E	-	4	2013
TerraMobilita/iQmulus [63]	Riegl LMS-Q120i	12M	22	2013
Paris-Rue-Madame [64]	Velodyne HDL-32E	20M	17	2014
TUM-MLS1 [13]	Velodyne HDL-64E	41M	8	2016
Paris Lille 3D [65]	Velodyne HD-32E	143.1M	50	2018
SemanticKITTI [66]	Velodyne HDL-64E	4.5 B	28	2019
Toronto-3D [67]	Velodyne HDL-32E	78.3M	8	2020
A2D2 [68]	Velodyne VLP-16	-	38	2020

4.1. Study Area

The study area covers the city campus of the Technical University of Munich (TUM) (Figure 4.1) which is located in Munich, Germany, (48.1493° N, 11.5685° E) and the total area is of about 29000 m^2. In this study, a cooperative benchmark of TUM-PF and Fraunhofer IOSB, TUM-MLS1 dataset is utilized [13]. The research topic not only connects multiple groups (such as computer vision, robotics, and geoinformatics), but also provides a consistent and area-wide representation of the studied urbanized landscape, including high-rise building facades. The test area is around 0.2 km^2 in size, and has highways that are around 1 km long.

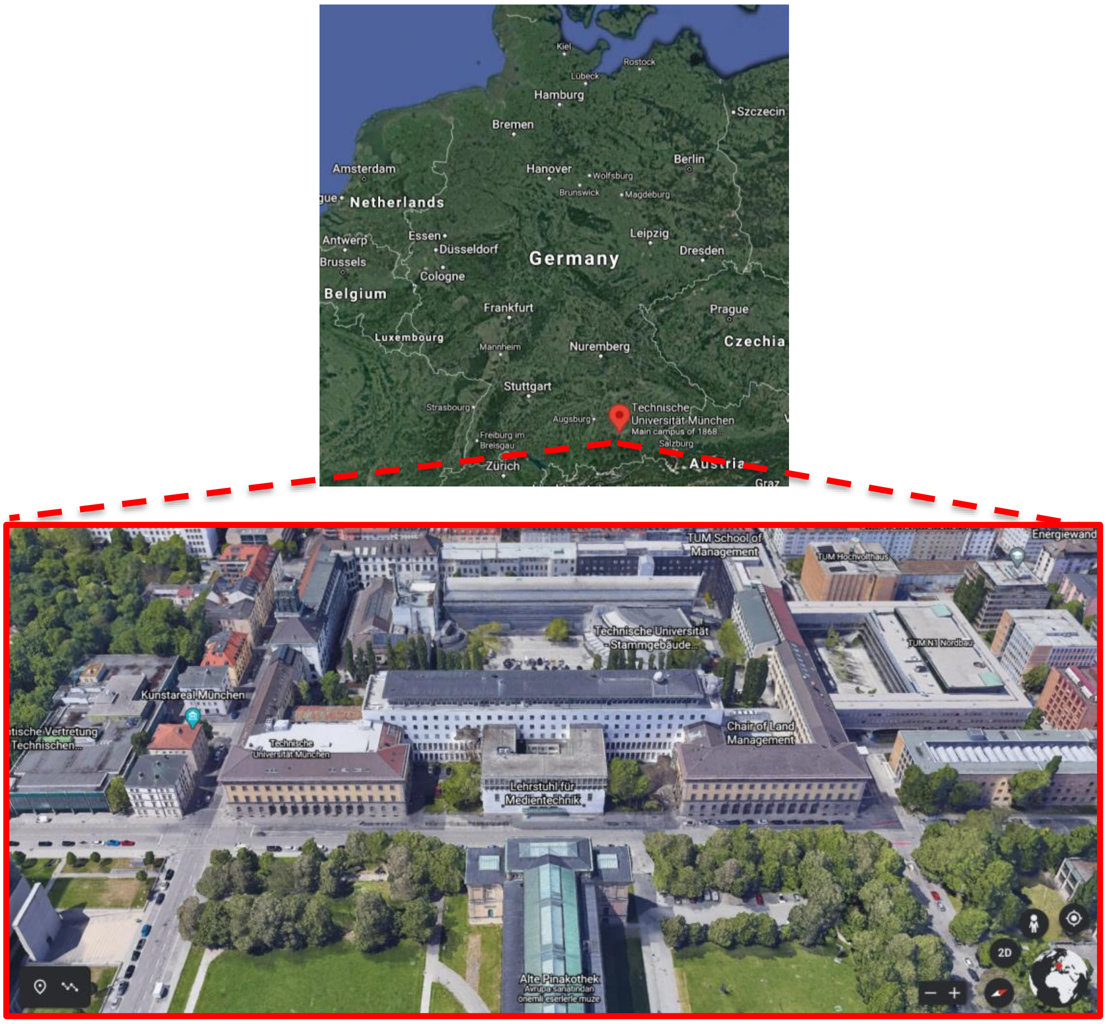


Figure 4.1: Overview and real scene representation of TUM campus from Google Earth.

4.2. Dataset

The test dataset TUM-MLS1 was acquired by MLS platform MODISSA on April 18th, 2016. The point clouds were created using two Velodyne HDL-64E cameras set at a 35° angle on the vehicle's front top [41]. The dataset requires 62 GB of storage, and each point within the dataset has 3D coordinates (X, Y, and Z) and an intensity value. Table 4.2 includes dataset specifications. The parts of the test area have been manually labeled for training and evaluation as ground truth, and it is shown in Figure 4.2.

Table 4.2: Specification of benchmark from MLS

Specification	**Data collection date**	April 2016
	Data type	MLS
	Sensor type	Velodyne HDL-64E
	Sensor angle	35° on the front top of the vehicle
	Scene type	Urban
	File format	pcd
	File size	62 GB

The dataset includes the following classes:

- Artificial Terrain (1) as roads and impervious ground,
- Natural Terrain (2) as grass and bare land,
- High Vegetation (3) as trees,
- Low Vegetation (4) as bushes and flowers,
- Building (5) as building facades and roofs,
- Hardscape (6) as walls and fences,
- Artifact (7) as power cables and other artificial objects, and
- Vehicle (8) as parked cars and busses.

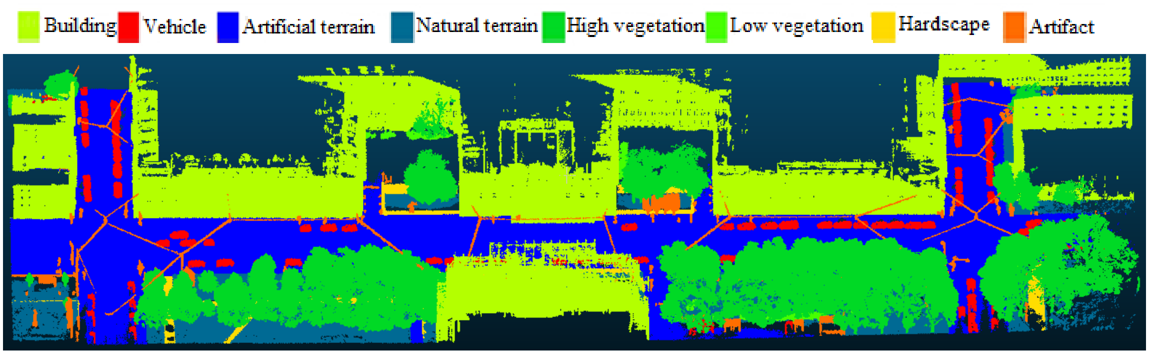


Figure 4.2: Manually classified MLS point cloud of the test scene with eight different semantic labels.

Figure 4.3 shows the distribution of the ground truth (reference data) size of each class for the entire data set, and in total, there are 3,039,327 labeled points. For certain classes like buildings and high vegetation, relatively high numbers of reference points are noticeable. Whereas, the dataset holds relatively less numbers of labeled data especially for two classes, i.e. low vegetation and hardscape.

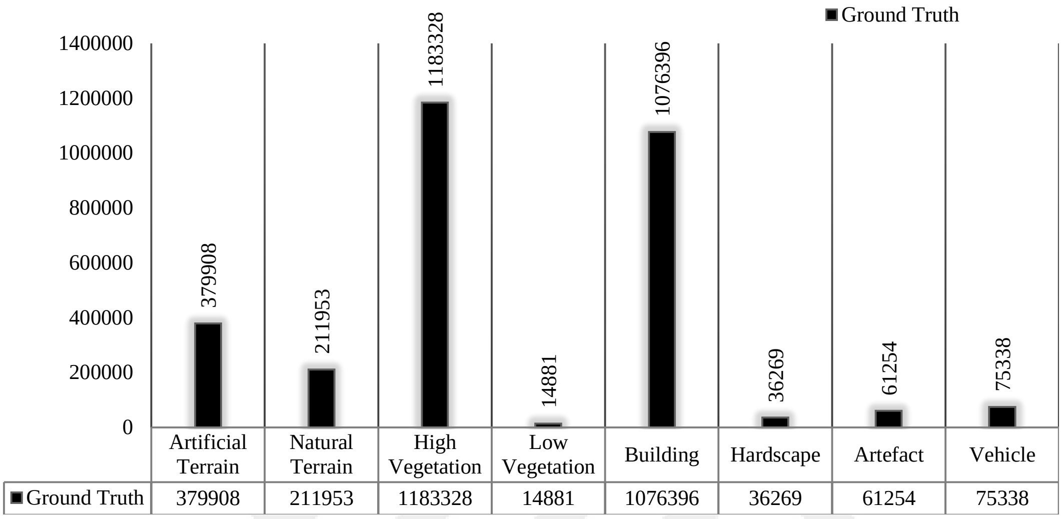


Figure 4.3: Distribution of the ground truth point sizes of each class within the dataset.

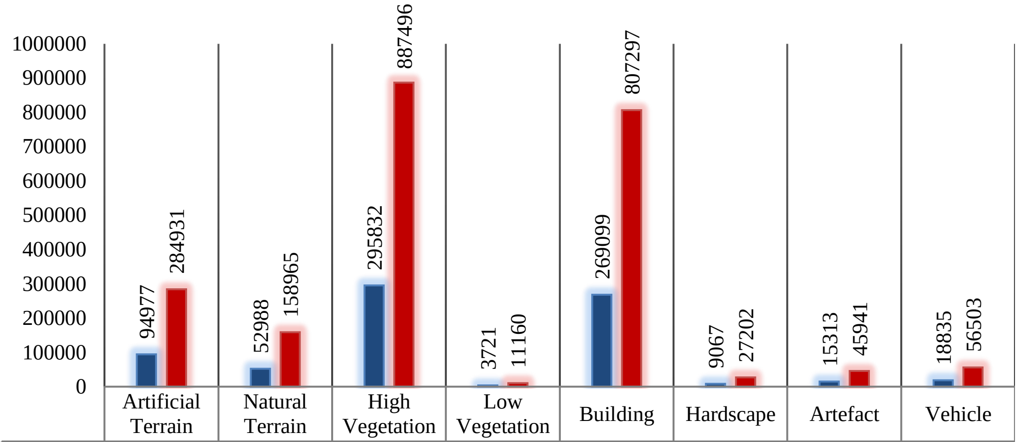
4.3. Assessment Strategy

As the assessment strategy, the *holdout* cross-validation [69] technique is performed. Cross-validation is a popular procedure preferred to evaluate different machine learning models, and designed to figure out how a machine learning model will act for a dataset. The holdout technique is one of the cross validation methods randomly dividing the data set into training and test data.

Initially, the reference data were randomly divided into two groups: one to be used as a training set (about 25%), and the other (about 75%) to be utilized independently for evaluating the classification findings. Figure 4.4 shows those two groups and the distribution of number of points each class within the group.

In the case of holdout cross-validation, the training dataset is also randomly divided into training and validation data. While the classification model is estimated with the training data, the performance of the trained model is internally evaluated with the validation data. The training set is also randomly split into two groups during the RF learning stage, with 80% of the training set serving as training data and the remaining 20% serving as validation data (Figure 4.5).

In addition, during this step, *k-fold* cross validation method (k value was chosen as 5) was also tested. However, since the results computed were found to be the same (difference less than 1%), the results of such cross validation were not reported in this thesis.



	Artificial Terrain	Natural Terrain	High Vegetation	Low Vegetation	Building	Hardscape	Artefact	Vehicle
Training	94977	52988	295832	3721	269099	9067	15313	18835
Testing	284931	158965	887496	11160	807297	27202	45941	56503

Figure 4.4: Distribution of the ground truth sizes of each class for training and testing datasets.

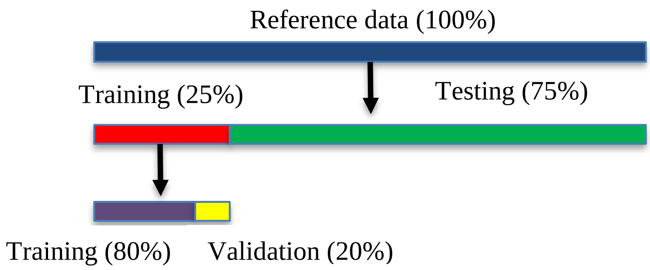


Figure 4.5: Selection strategy of training, validation, and test sets.

During the evaluations of the classified results, a detailed investigation on the classification performance is provided. Well known two measures, overall accuracy (total number of correct labels / total number of reference labels) and kappa index (difference to chance), are calculated to understand how well the RF classifier predict the classes. Confusion matrices as well as results of producer’s and user’s accuracies are also presented for further evaluation.

4.4. Parameter Evaluation

According to the framework described in Chapter 3, the parameters necessary to initialize the proposed method are presented in Table 4.3.

Table 4.3: Parameter test settings defined for the strategies implemented.

Stage	Parameters	{Tested} / "Selected"
3.1. Neighborhood recovery	k Value for kNN neighborhood based on 3D Euclidian distance ($\boldsymbol{N}_{\mathbf{kNN}}$)	{50,75,100,125, "150"}
	Radius for spherical neighborhood ($\boldsymbol{N}_{\boldsymbol{S}}$)	{0.50,0.75, 1.00,1.25, "1.50"}
	Radius for vertical cylindrical neighborhood ($\boldsymbol{N}_{\boldsymbol{C}}$)	{0.50, "0.75"}
3.2. Geometric Feature Extraction	# of features ($\boldsymbol{N}_{\mathbf{kNN}}$)	{in total 12, "in total 9" }
	# of features ($\boldsymbol{N}_{\boldsymbol{S}}$)	{in total 13, "in total 7" }
	# of features ($\boldsymbol{N}_{\boldsymbol{C}}$)	{in total 13, "in total 4" }
3.3. Random Forest Classification	# of trees	{5, 10, 15, 20, "25", 30}
	# of variables (n) to select for each decision split, $\sqrt{n}$	"3" (default)
	Minimum # of observations per tree leaf	"1" (default)

In order to test how point-based classification is affected by neighborhood selection, different neighborhood types along with 13 geometric features described in Section 3.2, were taken into account. The outputs of the classification for different classes in terms of user's and producer's accuracies are summarized in the following tables (Table 4.4, Table 4.5, and Table 4.6).

Table 4.4: User's accuracy (%) and producer's accuracy (%) of the validation set for each class according to kNN (kNN neighborhood: N_{kNN} where k = 50, 75, 100, 125, 150 points) (Best performances are given in bold).

Classes	Type of Accuracy	$\boldsymbol{N}_{\mathbf{kNN:50}}$	$\boldsymbol{N}_{\mathbf{kNN:75}}$	$\boldsymbol{N}_{\mathbf{kNN:100}}$	$\boldsymbol{N}_{\mathbf{kNN:125}}$	$\boldsymbol{N}_{\boldsymbol{kNN}\mathbf{:150}}$
Artificial Terrain	User's	88.8	88.9	89.7	90.3	**90.4**
	Producer's	93.5	94.0	94.1	94.2	**94.3**
Natural Terrain	User's	76.1	78.2	79.3	80.0	**80.3**

	Producer's	79.2	80.1	80.9	82.3	**82.7**
High Vegetation	User's	88.8	94.7	96.3	96.4	**96.5**
	Producer's	95.0	97.5	97.8	98.6	**98.7**
Low Vegetation	User's	76.8	76.8	81.0	83.5	**85.5**
	Producer's	48.3	53.3	59.3	64.0	**67.1**
Building	User's	91.3	95.3	95.9	97.0	**97.3**
	Producer's	86.1	93.1	95.1	95.3	**95.5**
Hardscape	User's	75.3	77.2	79.0	80.0	**81.1**
	Producer's	44.6	49.2	51.2	**53.3**	53.0
Artifact	User's	83.9	87.4	87.9	**90.3**	89.4
	Producer's	45.7	62.1	69.0	71.8	**74.3**
Vehicle	User's	83.6	85.0	85.6	87.1	**88.1**
	Producer's	82.5	83.3	84.0	83.9	**84.3**

Table 4.5: User's accuracy (%) and producer's accuracy (%) of the validation set for each class according to spherical neighborhood type (Spherical Neighborhood: N_S, and radius = 0.50, 0.75, 1, 1.25, 1.50 (in meters)) (Best performances are given in bold).

Classes	**Type of Accuracy**	$\boldsymbol{N_{S:0.50}}$	$\boldsymbol{N_{S:0.75}}$	$\boldsymbol{N_{S:1.00}}$	$\boldsymbol{N_{S:1.25}}$	$\boldsymbol{N_{S:1.50}}$
Artificial Terrain	User's	85.1	86.0	86.7	88.3	**89.0**
	Producer's	90.3	90.9	91.5	92.7	**93.3**
Natural Terrain	User's	73.2	75.9	77.1	79.8	**81.4**
	Producer's	73.7	76.4	77.3	80.2	**81.4**
High Vegetation	User's	83.7	86.4	88.6	89.8	**90.6**
	Producer's	91.1	93.9	95.3	95.9	**96.4**
Low Vegetation	User's	50.5	60.2	70.6	82.9	**83.6**
	Producer's	19.1	34.5	38.9	49.9	**60.3**
Building	User's	83.5	88.1	90.8	92.3	**93.4**

	Producer’s	79.1	82.9	86.1	87.8	**88.9**
Hardscape	User’s	63.5	76.1	84.5	83.5	**87.4**
	Producer’s	19.6	32.1	42.1	49.3	**56.9**
Artifact	User’s	59.5	75.4	81.1	84.9	**87.5**
	Producer’s	19.3	29.4	38.2	44.5	**50.0**
Vehicle	User’s	63.6	71.3	76.2	81.1	**85.4**
	Producer’s	67.2	73.7	78.6	82.3	**84.9**

Table 4.6: User’s accuracy (%) and producer’s accuracy (%) of the validation set for each class according to cylindrical neighborhood type (Cylindrical neighborhood: N_C and radius=0.5, 0.75 (in meters)) (Best performances are given in bold).

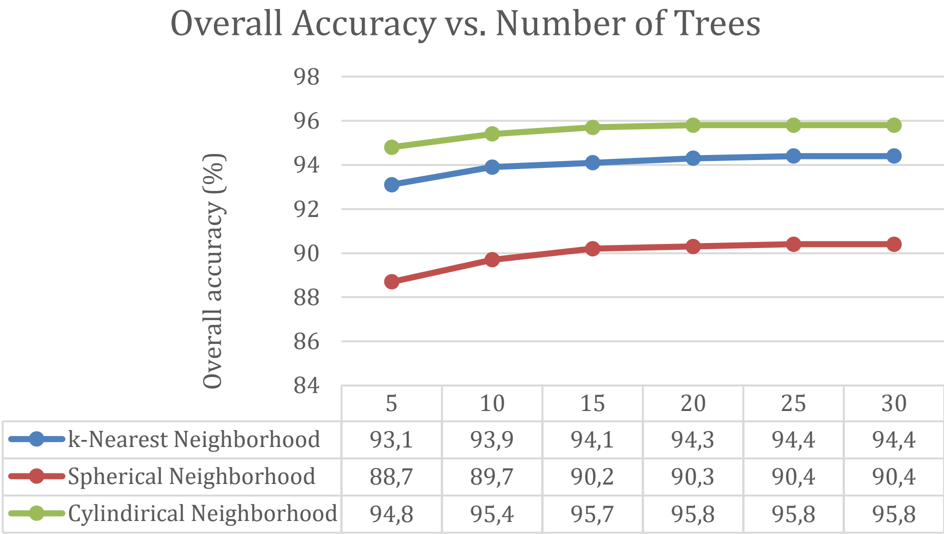
Classes	**Type of Accuracy**	$\boldsymbol{N_{C:0.50}}$	$\boldsymbol{N_{C:0.75}}$
Artificial Terrain	User’s	89.8	**90.9**
	Producer’s	94.0	**94.7**
Natural Terrain	User’s	80.5	**82.7**
	Producer’s	81.7	**84.0**
High Vegetation	User’s	97.9	**98.4**
	Producer’s	99.0	**99.1**
Low Vegetation	User’s	87.0	**90.5**
	Producer’s	65.6	**71.4**
Building	User’s	98.0	**98.1**
	Producer’s	97.3	**97.9**
Hardscape	User’s	81.4	**84.2**
	Producer’s	53.6	**56.0**
Artifact	User’s	90.3	**92.6**
	Producer’s	77.9	**81.0**
Vehicle	User’s	88.4	**89.9**
	Producer’s	86.3	**88.1**

The performance of the classification can be analyzed by inspecting the overall accuracies and kappa values computed for the validation set. Considering the parameters tested for three different local neighborhood types, the best results for the overall accuracies and kappa values are obtained in cases where (*i*) k is set to 150 for kNN neighborhood, (*ii*) the radius is set to 0.75 meters for the vertical cylindrical neighborhood, and (*iii*) the radius is 1.50 meter for the spherical neighborhood (Table 4.7).

Table 4.7: Overall accuracy and Kappa index (in %) of validation set for all three neighborhood and size parameters (best performances are given in bold).

Neighborhood	**Size Parameter**	**Overall Accuracy (%)**	**Kappa Index (%)**
$\boldsymbol{N_{kNN}}$	K=50	88.4	83.4
	K=75	92.5	89.2
	K=100	93.5	90.7
	K=125	94.2	91.6
	K=150	**94.4**	**91.9**
$\boldsymbol{N_S}$	R=0.50	82.3	74.4
	R=0.75	85.6	79.3
	R=1.00	87.8	82.5
	R=1.25	89.4	84.8
	R=1.50	**90.4**	**86.3**
$\boldsymbol{N_C}$	R=0.50	95.1	93.0
	R=0.75	**95.8**	**94.0**

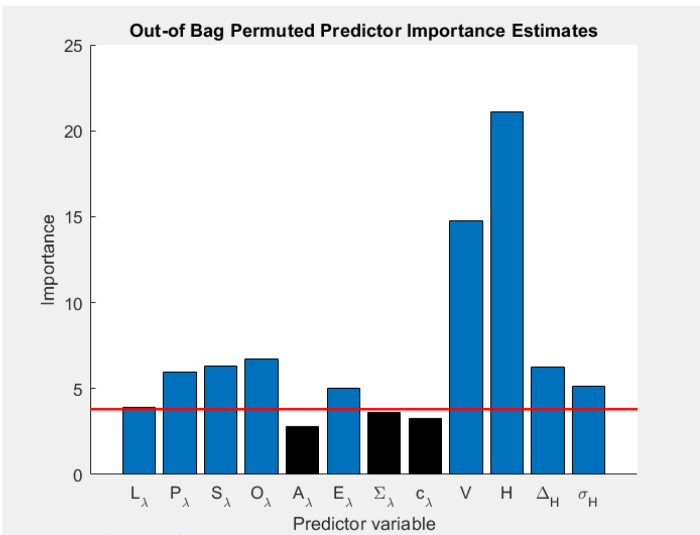
Considering the random forest classification, the effect of the number of trees was evaluated, since the number of trees is an important parameter of the RF classifier affecting the output results of the classification. For that parameter, a number of values (5, 10, 15, 20, 25 and 30) were tested, and the overall results reached for each parameter are illustrated in Figure 4.6. After such evaluation, the number of trees was fixed to 25, when results are found to be stable and consistent after that value. Table 4.3 shows the values that would normally not be modified for the other RF classifier parameters.



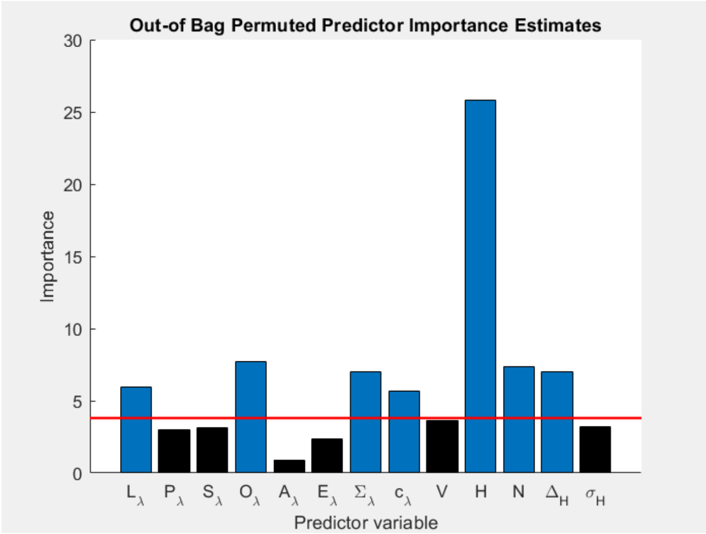
	5	10	15	20	25	30
k-Nearest Neighborhood	93,1	93,9	94,1	94,3	94,4	94,4
Spherical Neighborhood	88,7	89,7	90,2	90,3	90,4	90,4
Cylindirical Neighborhood	94,8	95,4	95,7	95,8	95,8	95,8

Figure 4.6: Overall accuracy based on the number of trees of the Random Forest classifier.

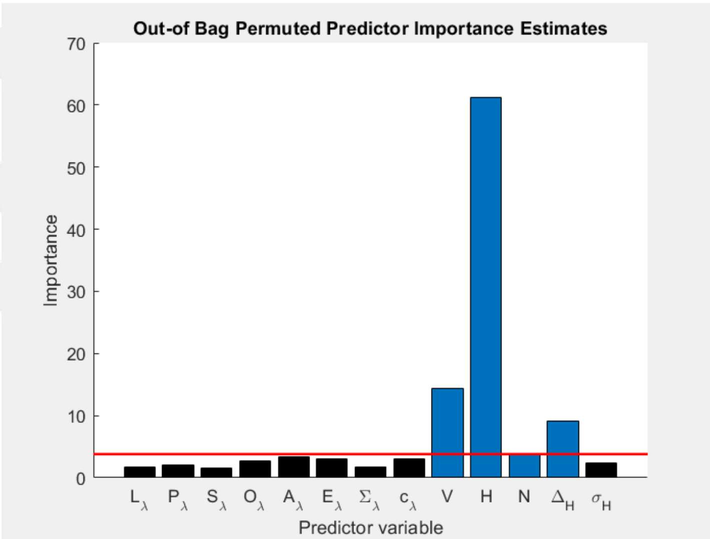
The relative importance of all 13 features used during the classification was tested using "Out of Bag Predictor Importance Estimates" of the RF algorithm of MATLAB (see Figure 4.7). A fixed threshold value was determined and predictors below this threshold were removed. If all of the results were jointly taken into account, the most important parameter for all neighborhood types amongst 13 features was found to be the *normalized height of a point* (*H*) (Figure 4.7).



(a)



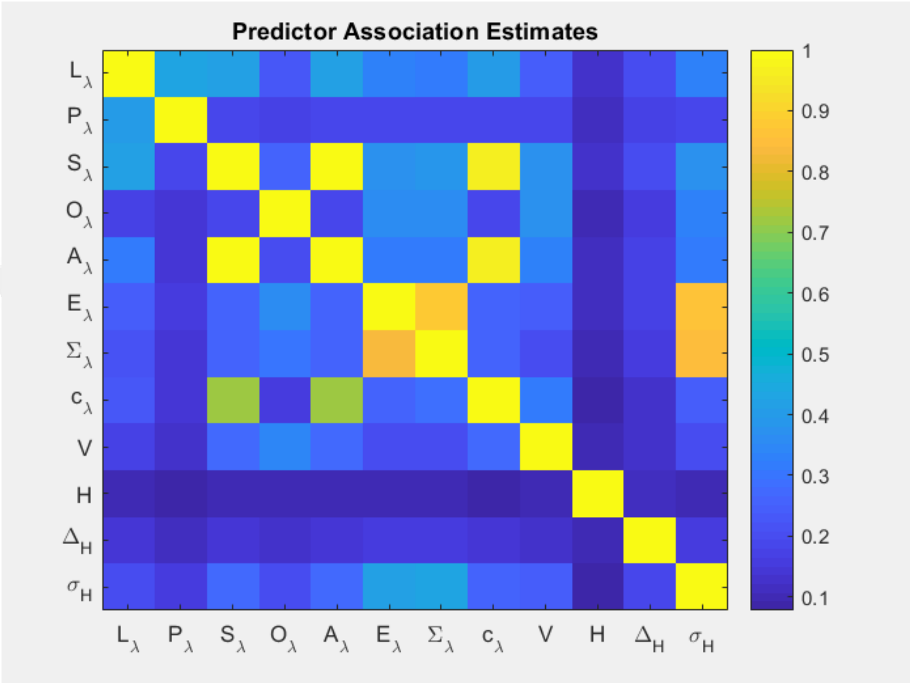
(b)



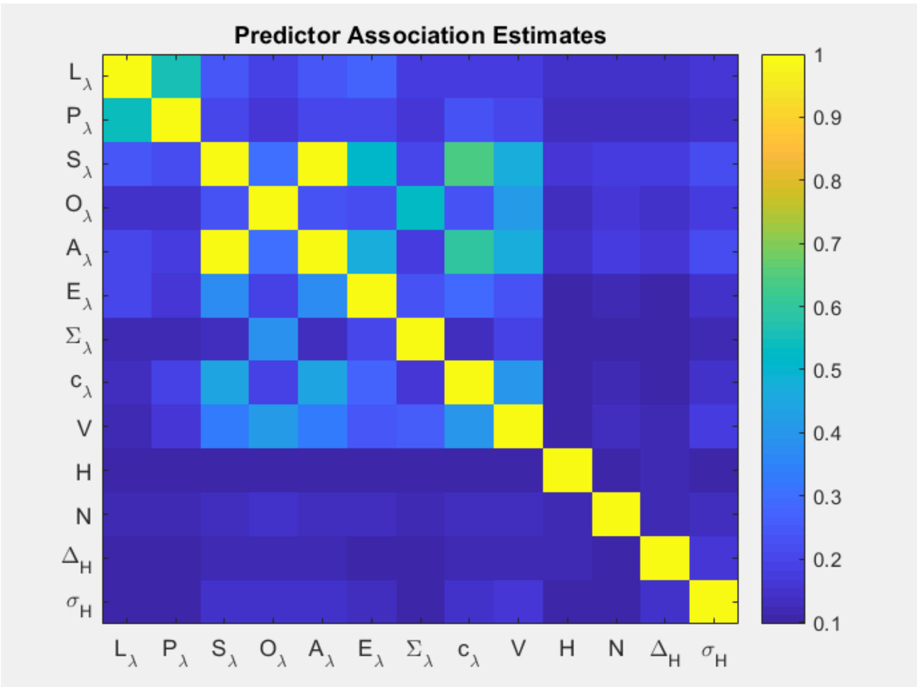
(c)

Figure 4.7: Out-of-bag importance of all parameters for (a) kNN (k = 150), (b) spherical neighborhood (R = 1.50m), and (c) cylindrical neighborhood (R = 0.75m). Note that the red line in all subfigures represent the threshold value selected (i.e. 3.8) to determine predictor importance. Blue and black colored bars represent the features accepted and rejected based on the selected threshold, respectively.

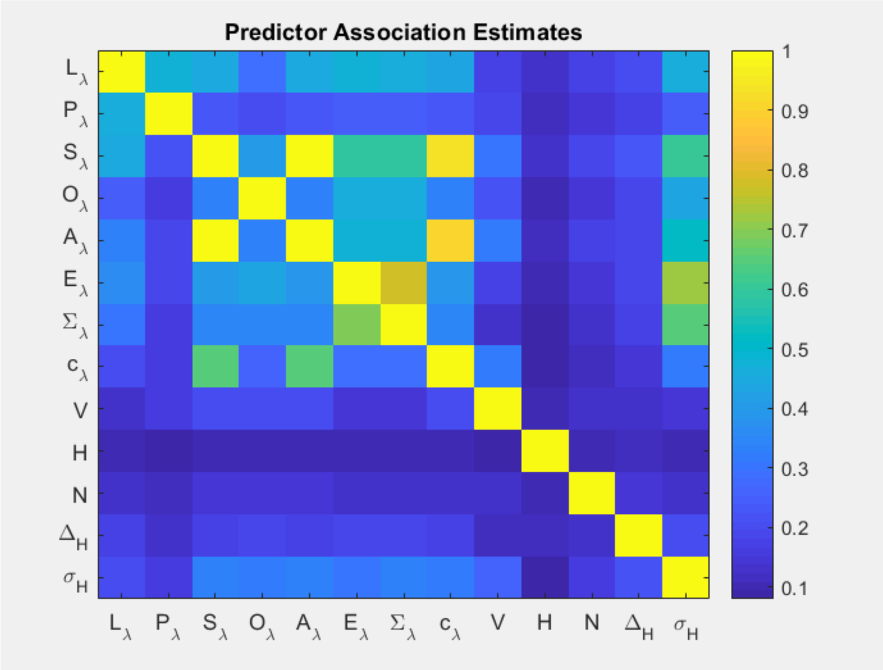
The strength of the relationship between pairs of predictors can be inferred using the elements of the predictor association. Larger values indicate highly correlated pairs of predictors. The tests revealed high correlation between the features *sphericity* and *anisotropy* with almost 100% correlation for all neighborhood types (Figure 4.8). In addition to *sphericity* and *anisotropy*, the feature *curvature* was also highly correlated with these two predictors for the kNN and the cylindrical neighborhood. These three highly correlated parameters were not considered for cylindrical neighborhood, since their importance was found to be relatively low. Depending on out-of-bag importance and predictor association estimates, the parameters given in the Table 4.8 were chosen as final features.



(a)



(b)



(c)

Figure 4.8: Predictor association estimates of using (a) kNN neighborhood, (b) spherical neighborhood, and (c) cylindrical neighborhood.

Table 4.8: Selected features according to out-of-bag importance and predictor association estimates for each neighborhood type.

Neighborhood	Size Parameter	Selected Features
N_{kNN}	K=150	$L_\lambda, P_\lambda, S_\lambda, O_\lambda, E_\lambda, V, H, \Delta H, \sigma H$
N_{S}	R=1.50	$L_\lambda, O_\lambda, \sum\lambda, c_\lambda, H, N, \Delta H$
N_{C}	R=0.75	$V, H, N, \Delta H$

5. RESULTS AND DISCUSSION

This chapter summarizes the point-based classification of the TUM-MLS1 point cloud using 8 classes (i.e. artificial terrain, natural terrain, high vegetation, low vegetation, building, hardscape, artifact and vehicle). First, the classification results regarding the numerical and visual outcomes for each local neighborhood definition are reported, and the reasons related to the false positive matches are discussed. Next, the outputs of RF classifications dealing with combinations of important features extracted from such local neighborhoods are presented. Finally, the classification results achieved in this thesis are compared with the other approaches selected from the literature considering the same benchmark dataset.

5.1. Results for Different Neighborhood Definitions

In order to understand how local neighborhood information affects the classification outcome, three local neighborhood extraction methods are assessed (kNN neighborhood, spherical neighborhood, and vertical cylindrical neighborhood), and all numerical results are reported in Tables 5.1 – 5.4.

Amongst the three local neighborhood information tested, the classification accuracy reached by means of geometric- and shape-based features through a local spherical neighborhood relationship is found to be the lowest. As shown in Section 4.4, the best result for training of the spherical neighborhood is obtained when the radius is set to 1.50 meters. Therefore, the confusion matrix output of the classification carried out using a spherical neighborhood based on 1.50 meter radius is evaluated (see Table 5.1). Main sources of classification errors are confusions between high vegetation and building classes, and artificial terrain and natural terrain classes. In this case, producer’s accuracy of three classes (i.e. low vegetation, hardscape, and artifact) is found to be less satisfactory (≈ 51-61%). Note that for three classes (artificial terrain, high vegetation, and building) pleasing results are achieved for both user’s and producer’s accuracies (≥ 88.3%). In this case, the overall accuracy and kappa index were computed to be 90.1% and 85.8%, respectively (Table 5.4).

Table 5.1: Confusion matrix of the test data using spherical neighborhood (R = 1.5m).

Confusion Matrix of the Test Data (2,279,495 points)									
Classes	***A.T (1)***	***N.T (2)***	***H.V (3)***	***L.V (4)***	***B (5)***	***H (6)***	***A (7)***	***V (8)***	**Total**
A.T (1)	**266616**	14948	0	1	1980	295	78	1013	**284931**
N.T (2)	22344	**130350**	67	71	3288	885	339	1620	**158964**
H.V (3)	0	122	**850667**	239	35163	74	1082	149	**887496**
L.V (4)	27	204	976	**6818**	1462	270	337	1066	**11160**
B (5)	3914	5817	81802	146	**712834**	430	938	1416	**807297**
H (6)	3398	5107	268	175	2107	**15177**	400	570	**27202**
A (7)	237	1168	9614	310	8564	606	**23549**	1893	**45941**
V (8)	1804	1553	459	260	3149	356	595	**48328**	**56504**
Total	**298340**	**159269**	**943853**	**8020**	**768547**	**18093**	**27318**	**56055**	**2279495**

User's Accuracy (%)	89.4	81.8	90.1	85.0	92.8	83.9	86.2	86.2
Producer's Accuracy (%)	93.6	82.0	95.9	61.1	88.3	55.8	51.3	85.5

In the case of kNN local neighborhood, the overall accuracy of the test data when k is set to 150 nearest points is computed as 94.5% (Table 5.4). Although this result is significantly better than the results achieved using the spherical neighborhood, yet a number of points belonging to the classes building and high vegetation are misclassified. The classification results of the low vegetation and hardscape classes are still computed to be the lowest two of all classes (Table 5.2).

Table 5.2: Confusion matrix of the test data using kNN neighborhood (k = 150).

Confusion Matrix of the Test Data (2,279,495 points)									
Classes	***A.T (1)***	***N.T (2)***	***H.V (3)***	***L.V (4)***	***B (5)***	***H (6)***	***A (7)***	***V (8)***	**Total**
A.T (1)	**269178**	12054	5	7	2048	441	67	1131	**284931**
N.T (2)	19786	**132592**	301	154	3257	1119	492	1263	**158964**
H.V (3)	0	408	**875862**	243	9844	113	906	120	**887496**
L.V (4)	278	903	594	**7473**	744	204	386	578	**11160**
B (5)	2641	5898	24925	208	**771658**	286	888	793	**807297**
H (6)	2681	7263	166	103	1293	**14574**	446	676	**27202**
A (7)	249	1822	4025	396	2725	599	**34452**	1673	**45941**
V (8)	1809	3349	137	260	1178	676	724	**48371**	**56504**
Total	**296622**	**164289**	**906015**	**8844**	**792747**	**18012**	**38361**	**54605**	**2279495**

User's Accuracy (%)	90.7	80.7	96.7	84.5	97.3	80.9	89.8	88.6	
Producer's Accuracy (%)	94.5	83.4	98.7	67.0	95.6	53.6	75.0	85.6	

Compared with the other two local neighborhood definitions, the most successful overall result was obtained with the vertical cylindrical neighborhood. As indicated in Table 5.4, the overall accuracy computed is 95.3% with a kappa index value of 93.3% for the cylindrical neighborhood based on radius of 0.75 meters. Two classes, High Vegetation and Building, yielded very successful results with accuracies all over 97.5% despite the confusion within these two classes (Table 5.3). Even in this case, the producer's accuracy of the class hardscape could not reach a satisfactory level (i.e. 56.8%).

Table 5.3: Confusion matrix of the test data using cylindrical neighborhood (R = 0.75m).

Confusion Matrix of the Test Data (2,279,495 points)									
Classes	***A.T (1)***	***N.T (2)***	***H.V (3)***	***L.V (4)***	***B (5)***	***H (6)***	***A (7)***	***V (8)***	**Total**
A.T (1)	**269631**	11341	0	32	2229	504	92	1102	**284931**
N.T (2)	20249	**132354**	197	143	2506	1483	477	1555	**158964**
H.V (3)	4	263	**877185**	309	8389	75	1131	140	**887496**
L.V (4)	342	917	621	**7382**	508	269	394	727	**11160**
B (5)	2651	3045	12295	139	**787128**	312	997	730	**807297**
H (6)	2438	6988	205	170	690	**15387**	515	809	**27202**
A (7)	337	1755	3275	415	2069	648	**35381**	2061	**45941**
V (8)	1946	3352	142	390	1058	688	908	**48020**	**56504**
Total	**297598**	**160015**	**893920**	**8980**	**804577**	**19366**	**39895**	**55144**	**2279495**

User's Accuracy (%)	90.6	82.7	98.1	82.2	97.8	79.5	88.7	87.1
Producer's Accuracy (%)	94.6	83.3	98.8	66.1	97.5	56.6	77.0	85.0

In this thesis, MATLAB was used during the implementation and processing, running on a desktop computer with i7-9700K 3.60GHZ, 32GB RAM. The processing time mainly affected by the size and definition of neighborhood form, and the number of features selected. As can be easily predicted, since the features were calculated using the points detected in the local neighborhood, the processing time increases as the number of points in the neighborhood increases. As indicated in Table 5.4, cylindrical neighborhood provided the best results. However, this comes at the cost of a substantial increase in processing time (approximately 6 hours, see Table. 5.4) especially because of the computation of covariance matrix to calculate the shape features. Although kNN method provided the second best results, its processing has completed within ≈ 8 minutes; thus, provided the shortest processing time. The processing of the spherical neighborhood was found to be nearly two times slower than the kNN method.

Table 5.4: Overall accuracy, kappa index (in %) and processing time (in minutes) for all three neighborhoods (bold values indicate the best values).

Neighborhood Information	Neighborhood Parameter	Time (minutes)	Overall Accuracy (%)	Kappa Index (%)
N_{kNN}	K=150	**7.44**	94.5	92.1
N_S	R=1.50	14.60	90.1	85.8
N_C	R=0.75	355.08	**95.3**	**93.3**

The result of the point-based classification with the RF classifier using the geometric and shape features extracted for each point in the point cloud with the vertical cylindrical neighborhood (R=0.75 meters) is given in the Figure 5.1. If the classification results are examined visually, objects with similar properties like artificial terrain and natural terrain, objects with either overlapping or adjacent (e.g. high vegetation vs. artifact or (building vs. high vegetation) seems to be caused misclassification (Figure 5.2).

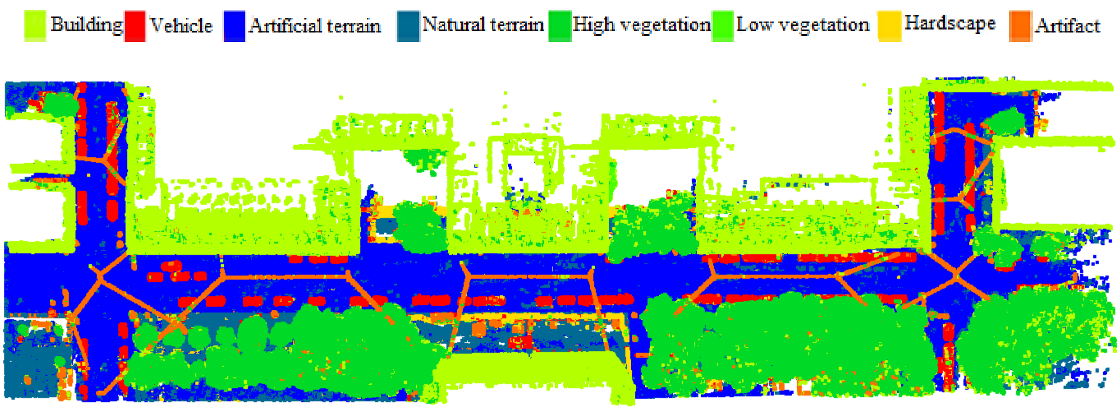


Figure 5.1: Classification output of the test dataset.

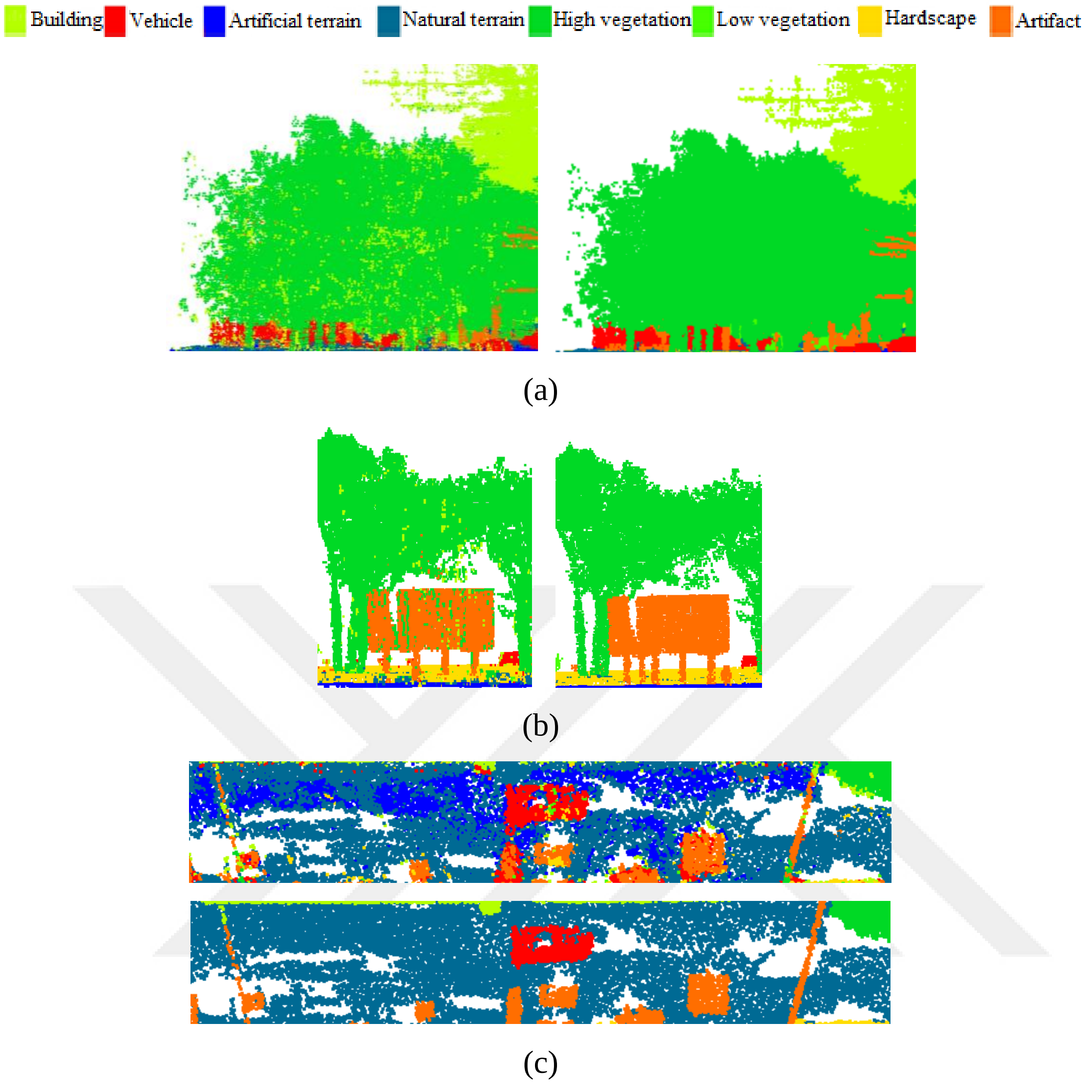


Figure 5.2: Misclassification of points in the (a) high vegetation (left: result vs. right: reference), (b) artifact classes due to overlap with high vegetation (left: result vs. right: reference), (c) natural terrain due to similarities between artificial terrain (top: result vs. bottom: reference).

5.2. Results after the Combination of Features from Different Neighborhood Information

Considering the results presented in Section 5.1, a nice balance between the accuracy and the processing speed was achieved by the kNN method. Accordingly, 9 important features of kNN were selected as base features, and the (important) features from other neighborhood relations are added to the feature set, and thereafter, the results are evaluated in cooperation. Firstly, only the confusion matrix of the test results for kNN neighborhood is given in the Table 5.5 for completeness. Note that the producer's accuracy of the two classes, hardscape and low vegetation, were at an insufficient level. However, overall accuracy and kappa was computed as 94.5% and 92.1%, respectively. This was the shortest process and took ≈ 7.44 minutes (Table 5.9).

Table 5.5: Confusion matrix of the test data using kNN with selected 9 important features (k=150)

Confusion Matrix of the Test Data (2,279,495 points)									
Classes	***A.T (1)***	***N.T (2)***	***H.V (3)***	***L.V (4)***	***B (5)***	***H (6)***	***A (7)***	***V (8)***	**Total**
A.T (1)	**269178**	12054	5	7	2048	441	67	1131	**284931**
N.T (2)	19786	**132592**	301	154	3257	1119	492	1263	**158964**
H.V (3)	0	408	**875862**	243	9844	113	906	120	**887496**
L.V (4)	278	903	594	**7473**	744	204	386	578	**11160**
B (5)	2641	5898	24925	208	**771658**	286	888	793	**807297**
H (6)	2681	7263	166	103	1293	**14574**	446	676	**27202**
A (7)	249	1822	4025	396	2725	599	**34452**	1673	**45941**
V (8)	1809	3349	137	260	1178	676	724	**48371**	**56504**
Total	**296622**	**164289**	**906015**	**8844**	**792747**	**18012**	**38361**	**54605**	**2279495**

User's Accuracy (%)	90.7	80.7	96.7	84.5	97.3	80.9	89.8	88.6
Producer's Accuracy (%)	94.5	83.4	98.7	67.0	95.6	53.6	75.0	85.6

Secondly, important features obtained by kNN (= 9) and spherical neighborhood (= 7) were combined, and RF classification was repeated using all 16 features. Based on the results computed, the overall results and kappa index are increased to 96.1% and 94.4%, respectively (Table 5.9).

Table 5.6: Confusion matrix of the test data using important features of the kNN and spherical neighborhoods (k=150, R=1.5m)

Confusion Matrix of the Test Data (2,279,495 points)									
Classes	***A.T (1)***	***N.T (2)***	***H.V (3)***	***L.V (4)***	***B (5)***	***H (6)***	***A (7)***	***V (8)***	**Total**
A.T (1)	**275331**	6943	0	7	1495	202	72	881	**284931**
N.T (2)	12533	**142589**	77	60	1918	628	349	810	**158964**
H.V (3)	0	121	**876795**	99	9996	28	443	14	**887496**
L.V (4)	65	270	300	**9191**	565	109	237	423	**11160**
B (5)	1402	1996	22998	44	**780164**	50	331	312	**807297**
H (6)	1422	4710	116	49	1252	**19012**	313	328	**27202**
A (7)	209	1033	4443	177	3220	274	**35341**	1244	**45941**
V (8)	1621	952	62	57	613	318	349	**52532**	**56504**
Total	**292583**	**158614**	**904791**	**9684**	**799223**	**20621**	**37435**	**56544**	**2279495**

User's Accuracy (%)	94.1	89.9	96.9	94.9	97.6	92.2	94.4	92.9
Producer's Accuracy (%)	96,.6	89.7	98.8	82.4	96.6	69.9	76.9	93.0

Thirdly, the classification result of the test data was assessed by appending only the 3 important geometric features (H, N and ΔH) through the cylindrical neighborhood to the important features derived from the kNN neighborhood. As a result, surprisingly, the overall accuracy and kappa index are slightly decreased to 95.8% and 93.9%, respectively, when compared to the previous case. This was due to the stunning decrease observed in classification results for the three classes: low vegetation, hardscape and vehicle. In this case, the classification task lasted for around 46 minutes (Table 5.9).

Table 5.7: Confusion matrix of the test data using kNN and cylindrical neighborhoods. Important 9 features of kNN and 3 features of cylindrical neighborhood as H, $N$and ΔH were jointly processed (k=150, R=0.75m).

Confusion Matrix of the Test Data (2,279,495 points)									
Classes	***A.T (1)***	***N.T (2)***	***H.V (3)***	***L.V (4)***	***B (5)***	***H (6)***	***A (7)***	***V (8)***	**Total**
A.T (1)	**272016**	10045	0	5	1433	361	72	999	**284931**
N.T (2)	17288	**136506**	133	117	2385	959	414	1162	**158964**
H.V (3)	0	327	**879552**	228	6400	116	786	87	**887496**
L.V (4)	288	830	484	**8119**	434	152	289	564	**11160**
B (5)	2307	3882	15641	41	**784082**	325	471	548	**807297**
H (6)	2257	6731	154	100	952	**16082**	382	544	**27202**
A (7)	233	1538	3127	338	1963	455	**36786**	1501	**45941**
V (8)	1842	2909	75	168	736	568	598	**49608**	**56504**
Total	**296231**	**162768**	**899166**	**9116**	**798385**	**19018**	**39798**	**55013**	**2279495**

User's Accuracy (%)	91.8	83.9	97.8	89.1	98.2	84.6	92.4	90.2
Producer's Accuracy (%	95.5	85.9	99.1	72.8	97.1	59.1	80.1	87.8

Finally, the RF classification was performed by combining the features obtained from all neighborhood information using a total of 19 features extracted for each point in the point cloud. In this case, the overall accuracy and kappa index were computed to be

96.9% and 95.5%, respectively (Table 5.9). However, as expected, the process took longer than the other combinations tested, as the relatively large number of points in the cylindrical neighborhood significantly increases the total processing time (≈ 66.88 minutes). Figure 5.3 compares visually the reference data and the results of classification performed using each combination for the selected 6 classes.

Table 5.8: Confusion matrix of the test data using all neighborhood information. All important features of kNN and spherical neighborhood, and 3 features of cylindrical neighborhood as H, N and ΔH were jointly processed (k=150 for kNN, R=1.50m for spherical neighborhood and R=0.75m for cylindrical neighborhood).

Confusion Matrix of the Test Data (2,279,495 points)									
Classes	***A.T (1)***	***N.T (2)***	***H.V (3)***	***L.V (4)***	***B (5)***	***H (6)***	***A (7)***	***V (8)***	**Total**
A.T (1)	**276404**	6107	1	5	1309	202	60	843	**284931**
N.T (2)	12117	**143406**	88	54	1561	675	310	753	**158964**
H.V (3)	0	113	**880100**	133	6873	14	261	2	**887496**
L.V (4)	70	240	266	**9472**	461	85	175	391	**11160**
B (5)	1258	1544	15275	20	**788758**	38	132	272	**807297**
H (6)	1404	4461	131	45	969	**19693**	223	276	**27202**
A (7)	220	940	3511	152	2357	257	**37373**	1131	**45941**
V (8)	1640	791	36	59	468	195	248	**53067**	**56504**
Total	**293113**	**157602**	**899408**	**9940**	**802756**	**21159**	**38782**	**56735**	**2279495**

User's Accuracy (%)	94.3	91.0	97.9	95.3	98.3	93.1	96.4	93.5
Producer's Accuracy (%)	97.0	90.2	99.2	84.9	97.7	72.4	81.3	93.9

Table 5.9: Overall accuracy, kappa index (in %) and processing time (in minutes) for combination of all neighborhoods (bold values indicate the best values).

Neighborhood Information	Number of Features	Time (minutes)	Overall Accuracy (%)	Kappa Index (%)
N_{kNN}	9	**7.44**	94.5	92.1
$N_{kNN} + N_S$	9+7 = 16	24.46	96.1	94.4
$N_{kNN} + N_C$	9 + 3 ($H, N, \Delta H$) =12	46.20	95.8	93.9
$N_{kNN} + N_S + N_C$	9+7+3 = 19	66.88	**96.9**	**95.5**

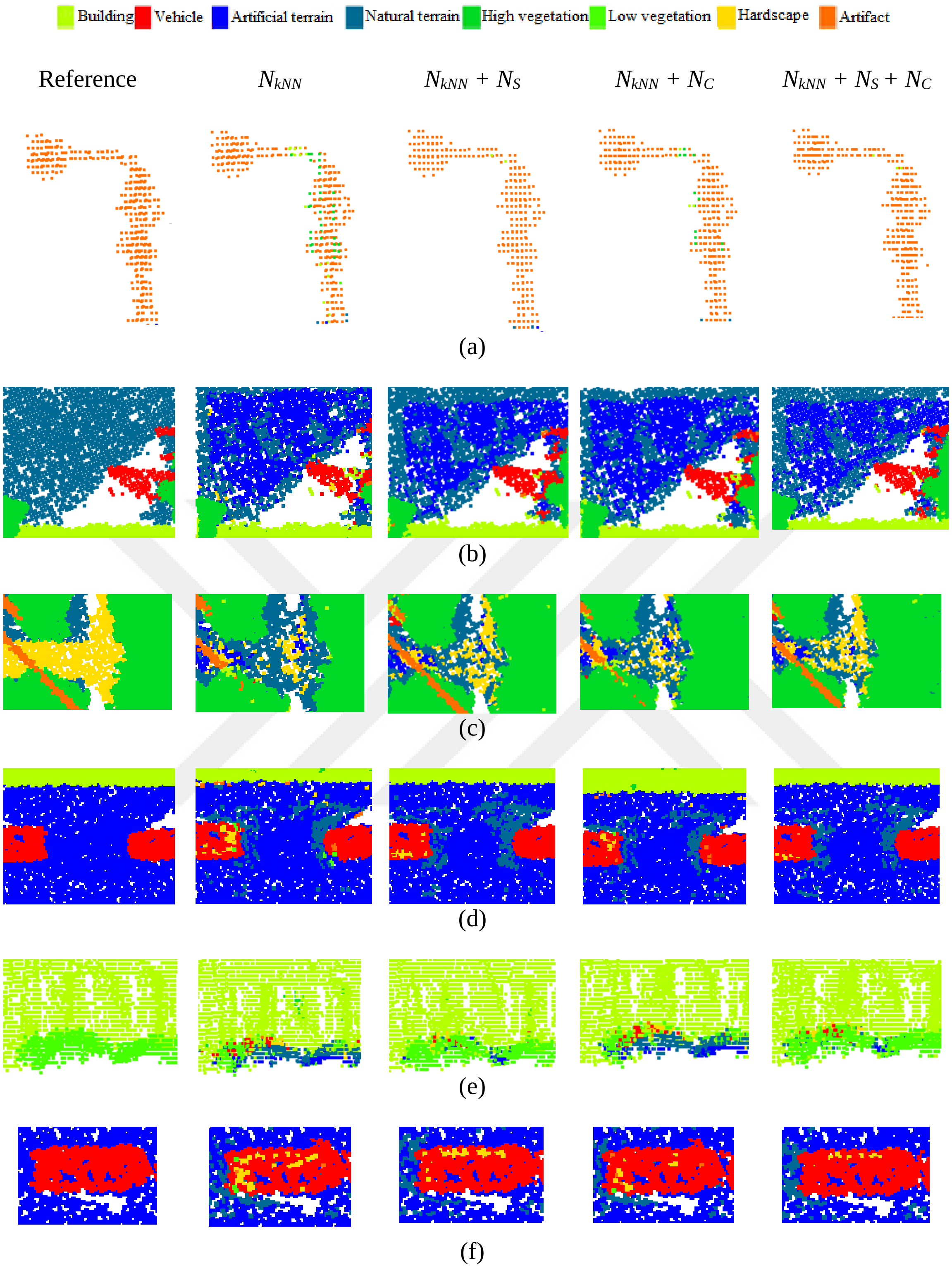
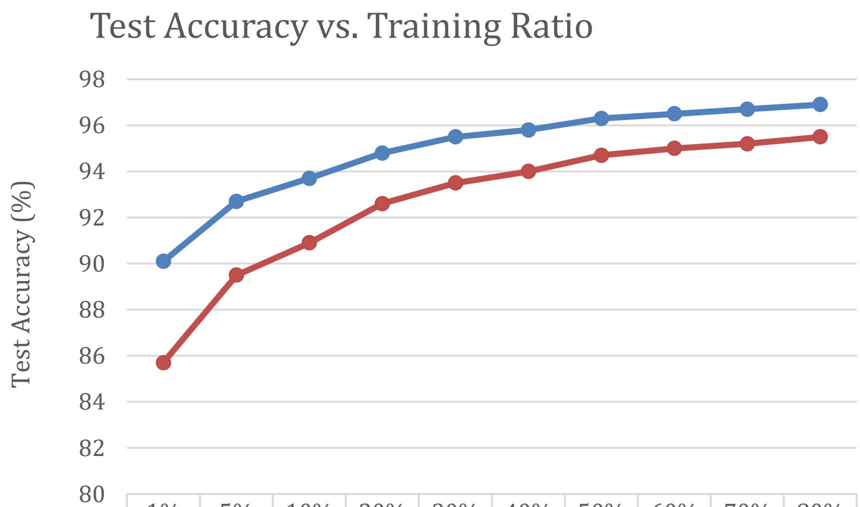


Figure 5.3: Visual classification results based on neighborhood combination of classes (a) artifact, (b) natural terrain, (c) hardscape, (d) artificial terrain, (e) low vegetation, and (f) vehicle.

As described in Section 4.3, all test results were obtained using the training and validation ratio of 80% and 20%, respectively. However, in this thesis, it is also considered how the classification results be affected after reducing the ratio of training set for the approach combining all neighborhood information. For this reason, a set of training ratio was chosen between 1% - 80%, and the overall accuracies and kappa indices of the test dataset (75% of the whole data set, cf. Section 4.3) were computed (Figure 5.4). As seen from Figure 5.4, reducing the training ratio affects the overall accuracy of the test data negatively, as expected. However, this analysis proves that even with a 1% training ratio, it is possible to achieve a successful overall accuracy reaching over 90%.



	1%	5%	10%	20%	30%	40%	50%	60%	70%	80%
Overall Accuracy(%)	90,1	92,7	93,7	94,8	95,5	95,8	96,3	96,5	96,7	96,9
Kappa Index (%)	85,7	89,5	90,9	92,6	93,5	94	94,7	95	95,2	95,5

Figure 5.4: Overall accuracy and kappa index (both in %) of the test dataset when the training ratio is changed.

5.3. Comparison with Previous Studies

The results are evaluated by conducting a comparison to the state-of-the-art approaches reported in [41]. In [41], a total of 13 features; 7 eigen-based features (linearity, planarity, sphericity, omnivariance, anisotropy, eigenentropy and local curvature), 2 height features (mean height and height difference), 3 spatial features (normal vectors), and 1 radiometric feature (intensity), were utilized using the RF classifier. Accordingly, classification of the same benchmark has been performed using this feature set with the methods described in [70] and [71]. In this way, the results achieved in this thesis are compared with the approaches presented in [41], [70], and [71]. Note also that in [41], the number of trees was chosen as 200 for RF classifier, and 50% of the entire dataset was used as training for the results presented in Table 5.10. However, in this thesis, only 20% of the entire dataset was used as training purposes (see Figure. 4.5) and the number of trees was 25.

The comparison of the accuracies of individual classes and the overall accuracies are provided in Table 5.10. As indicated in the results, a combination of the features from different neighborhood information increased the overall results more than 4% compared to the approach presented in [41]. Besides, massive improvements for producer's accuracies of multiple classes are achieved, e.g. low vegetation, hardscape, artifacts.

Table 5.10: Comparison with the other approaches using the TUM-MLS1 dataset (best performances are given in bold).

Class	**Accuracy Measure**	**Approach in [70]**	**Approach in [71]**	**Approach in [41]**	**Approach Presented**
A.T (1)	*User's*	0.951	0.917	**0.952**	0.943
	Producer's	0.932	0.936	0.943	**0.970**
N.T (2)	*User's*	0.781	0.799	0.839	**0.910**
	Producer's	0.905	0.883	**0.911**	0.902
H.V (3)	*User's*	0.858	0.895	0.931	**0.979**
	Producer's	0.934	0.960	0.972	**0.992**
L.V (4)	*User's*	0.697	0.692	0.787	**0.953**
	Producer's	0.333	0.267	0.496	**0.849**
B (5)	*User's*	0.878	0.915	0.933	**0.983**
	Producer's	0.842	0.887	0.931	**0.977**
H (6)	*User's*	0.707	0.864	0.844	**0.931**
	Producer's	0.391	0.365	0.495	**0.724**

A (7)	*User's*	0.489	0.663	0.747	**0.964**
	Producer's	0.190	0.303	0.345	**0.813**
V (8)	*User's*	0.798	0.840	0.827	**0.935**
	Producer's	0.766	0.801	0.848	**0.939**
	Overall	*0.868*	*0.893*	*0.921*	***0.969***

6. CONCLUSION AND RECOMMENDATION

In this chapter, the conclusions from the presented thesis work are summarized and recommendations for future works are presented.

6.1. Conclusion

This thesis has been completed under three main headings: the recovery of a point neighborhood, extraction of geometric and shape features of points within the defined neighborhood, and point-based classification of the TUM-MLS1 point cloud using RF classifier for a total of 8 classes (*Artificial Terrain*, *Natural Terrain*, *High Vegetation*, *Low Vegetation*, *Building*, *Hardscape*, *Artifact*, and *Vehicle*).

The following conclusions are reached from the results achieved for the point based classification framework implemented:

- The type of the local neighborhood significantly affects the results of classification. Considering the neighborhood types evaluated, the worst overall result (90.1%) is computed for the spherical neighborhood. The other two point neighborhood information, kNN and cylindrical neighborhoods, yield similar results. However, the cylindrical neighborhood, reaching 95.3% overall accuracy with a kappa ratio of 93.3%, provided slightly the best results.
- The results for particular classes (i.e. *Low Vegetation*, *Hardscape* and *Artifact*) are computed to be relatively low; and in general, could not reach a satisfactory level (≈ 50-70%). However, since the classification results of the other major classes (*Artificial Terrain*, *Natural Terrain*, *High Vegetation*, and *Building*) dominating the point samples in the dataset with a relatively high number of points, the final overall accuracies are all computed over 90%.
- The results of the neighborhood strategies examined in this thesis are influenced by the size of the local neighborhood. As expected, the larger the neighborhood size parameter, the more time-consuming the processing. Based on the results provided, although the cylindrical neighborhood reached the best results, it consumed the longest processing time amongst the three neighborhood types evaluated.
- Like every supervised classification outline, feature selection step affects the efficiency of processing; and thus, the accuracy of the final classification.

According to the tests carried out in this thesis, the most important feature amongst the selected 13 features is found to be the normalized height of a point (H), and this feature has ranked the first place for all neighborhood types. Therefore, in any case, estimating the terrain height for each point in an accurate manner seems to be a critical issue.

- Considering the geometric and shape features tested, we found that the shape features become more important for kNN and spherical neighborhoods; whereas almost all of the shape features have minimal effect on the final results when the cylindrical neighborhood is utilized. This is most probably due to the large variation of points existing in the cylindrical neighborhood which eventually makes difficult to interpret representative shape information.
- The features obtained with the kNN neighborhood, which has provided a nice balance between the overall accuracy and the processing speed, are combined with the features obtained with other neighborhood types, and thereafter, the classification steps are repeated. As a result, the overall accuracy of 96.9% was reached when a feature set was produced by using all neighborhoods at the same time, and this was the best result achieved for the benchmark dataset TUM-MLS1.
- Although successful results are achieved using the framework implemented, misclassifications are observed for classes, especially the classes with similar geometric features (i.e. artificial terrain and natural terrain), and in classes that are positioned very close to each other (i.e. high vegetation and building), as each point is classified individually without taking into account the neighborhood information during the classification.
- The framework presented in this thesis yielded better results compared to the results of previous studies using the same benchmark dataset. A combination of the features from different neighborhood information increased the overall results at least 4%, and considerable improvements (up to 40%) for producer's accuracies of multiple classes are observed, e.g. *Low Vegetation*, *Hardsca*pe, and *Artifacts*.

6.2. Recommendation

The followings are suggested for further studies:

- The feature set can be enriched (*i*) using the intensity information of the points, and (*ii*) with the RGB images (if available) that may be collected to improve the point cloud classification.
- Depending on the point density of the cloud, fixed radius selection may not be suitable for classes having small number of samples. Therefore, instead of using single neighborhood size parameter for the whole data set, multiple parameters for the size parameter may be preferred.
- The parameters determined in the study may not be effective for a dataset having a large difference in terms of the point density. For this reason, the parameters could be generalized according to the density of points in the cloud.
- The normalization of features is not performed in this thesis; however, feature normalization may positively affect the features which are found to be correlated. Therefore, approaches for the data normalization or dimensionality reduction can be evaluated.
- In this study, for all neighborhood definitions, Kd-tree data structure was utilized to efficiently detecting neighboring points. However, other data structures such as the octree scheme can be evaluated to speed up the processing.
- The receiver operating characteristic (ROC) curve showing the performance of a classification model at all classification thresholds can be drawn. Besides, area under the ROC curve (AUC) can be used to evaluate the classification performance.
- Since the processing load will increase as the data density increases, several approaches (segmentation, voxels etc.) to mitigate this problem can be implemented for better handling of large data sets (over 100 M). For instance, point cloud down-sampling can be utilized by filtering the dataset through a voxel grid.

- The validation of the proposed framework and its parameters must be carried out for other benchmark datasets available to the remote sensing community. In this way, the validity of the parameters and their effects can be better understood.
- Another common way to handle pixel neighborhoods is to incorporate a smoothness prior into the classification framework for point clouds. In this way, similar class labels for the points that are positioned very close to each other can be enforced during the classification step.
- Last but not least, the advances in deep learning may simplify the extraction of feature representations for point cloud classification. In this context, another future work is to test/compare the current framework with deep network models. In addition, how the trained model can be used as a feature extractor and transfer learning (i.e fine tuning strategies) can be investigated.

REFERENCES

[1] A. Gruen, X. Huang, R. Qin, T. Du, and W. Fang, Integration of Mobile Laser Scanning Data with UAV Imagery for Very High-Resolution 3D City Modelling, **2013**.

[2] M. Rutzinger, A. K. Pratihast, S. O. Elberink, and G. Vosselman, Detection and Modelling of 3D Trees from Mobile Laser Scanning Data, **2010**.

[3] C. Chen, L. Tang, C. M. Hancock, J. Yan, H. de Ligt, and P. Zhang, 2D-Based Indoor Mobile Laser Scanning for Construction Digital Mapping Application, presented at the FIG Congress 2018, Istanbul, Turkey, **2018**.

[4] H. Wang, Z. Cai, H. Luo, C. Wang, P. Li, W. Yang, S. Ren and J. Li, Automatic Road Extraction from Mobile Laser Scanning Data, International Conference on Computer Vision in Remote Sensing, pp. 136–139, **2012**.

[5] M. Yadav, A. K. Singh, and B. Lohani, Extraction of Road Surface From Mobile Lidar Data of Complex Road Environment, International Journal of Remote Sensing, vol. 38, no. 16, pp. 4655–4682, **2017**.

[6] B. Borgmann, V. Schatz, H. Kieritz, C. Scherer-Klöckling, M. Hebel, and M. Arens, Data Processing and Recording Using a Versatile Multi-Sensor Vehicle, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV–1, pp. 21–28, **2018**.

[7] R. Qin and A. Gruen, 3D Change Detection at Street Level Using Mobile Laser Scanning Point Clouds and Terrestrial Images, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 90, pp. 23–35, **2014**.

[8] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment, IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 4224–4231, **2018**.

[9] A. Kukko, H. Kaartinen, J. Hyyppä, and Y. Chen, Multiplatform Mobile Laser Scanning: Usability and Performance, Sensors, vol. 12, no. 9, **2012**.

[10] Martin Weinmann, Semantic Point Cloud Interpretation Based on Optimal Neighborhoods, Relevant Features and Efficient Classifiers, ISPRS Journal of Photogrammetry and Remote Sensing, p. 19, **2015**.

[11] Y. Xie, J. Tian, and X. X. Zhu, Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation, IEEE Geoscience and Remote Sensing Magazine, vol. 8, no. 4, pp. 38–59, **2020**.

[12] M. E. Atik, Z. Duran, and D. Z. Seker, Machine Learning-Based Supervised Classification of Point Clouds Using Multiscale Geometric Features, ISPRS International Journal of Geo-Information, vol. 10, no. 3, p. 187, **2021**.

[13] J. Zhu, J. Gehrung, R. Huang, B. Borgmann, Z. Sun, L. Hoegner, M. Hebel, Y. Xu and U. Stilla, TUM-MLS-2016: An Annotated Mobile LiDAR Dataset of the TUM City Campus for Semantic Point Cloud Interpretation in Urban Areas, Remote Sensing, vol. 12, no. 11, **2020**.

[14] R. Wang, J. Peethambaran, and D. Chen, LiDAR Point Clouds to 3-D Urban Models: A Review, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 2, pp. 606–627, **2018**.
[15] H. Zhang, Rapid Inspection of Pavement Markings Using Mobile Laser Scanning Point Clouds, Master thesis, University of Waterloo, Ontario, Canada, **2016**.
[16] M. Vaaja, A. Kukko, H. Kaartinen, M. Kurkela, E. Kasvi, C. Flener, H. Hyyppä, J. Hyyppä, J. Järvelä and P.Alho, Data Processing and Quality Evaluation of a Boat-Based Mobile Laser Scanning System, Sensors, vol. 13, no. 9, **2013**.
[17] Y. Lou, T. Zhang, J. Tang, W. Song, Y. Zhang, and L. Chen, A Fast Algorithm for Rail Extraction Using Mobile Laser Scanning Data, Remote Sensing, vol. 10, no. 12, **2018**.
[18] P. Kim, J. Chen, and Y. K. Cho, Robotic Sensing and Object Recognition From Thermal-Mapped Point Clouds, Int J Intell Robot Appl, vol. 1, no. 3, pp. 243–254, **2017**.
[19] K. Williams, M. J. Olsen, G. V. Roe, and C. Glennie, Synthesis of Transportation Applications of Mobile LIDAR, Remote Sensing, vol. 5, no. 9, **2013**.
[20] A. G. Kashani, M. J. Olsen, C. E. Parrish, and N. Wilson, A Review of LIDAR Radiometric Processing: From Ad Hoc Intensity Correction to Rigorous Radiometric Calibration, Sensors, vol. 15, no. 11, **2015**.
[21] I. Puente, H. González-Jorge, J. Martínez-Sánchez, and P. Arias, Review of Mobile Mapping and Surveying Technologies, Measurement, vol. 46, no. 7, pp. 2127–2145, **2013**.
[22] LMS221-30206 | Detection and ranging solutions SICK, https://www.sick.com/ag/en/detection-and-ranging-solutions/2d-lidar-sensors/lms2xx/lms221-30206/p/p109848 (Accessed: **Jun. 09, 2021**).
[23] Riegl Laser Measurement Systems, Riegl LMS-Q120i 2-D Laser Scanner, Datasheet, **2010**. Available: http://www.riegl.com/uploads/tx_pxpriegldownloads/10_1_DataSheet_LMS-Q120i_20-04-2010.pdf
[24] Puck Lidar Sensor, High-Value Surround Lidar, Velodyne Lidar. https://velodynelidar.com/products/puck/ (Accessed: **Jun. 09, 2021**).
[25] Velodyne’s HDL-32E Surround Lidar Sensor, Velodyne Lidar. https://velodynelidar.com/products/hdl-32e/ (Accessed: J**un. 09, 2021**).
[26] HDL-64E Durable Surround Lidar Sensor, Velodyne Lidar. https://velodynelidar.com/products/hdl-64e/ (Accessed: **Jun. 09, 2021**).
[27] H. Guan, J. Li, S. Cao, and Y. Yu, Use of Mobile LiDAR in Road Information Inventory: a review, International Journal of Image and Data Fusion, vol. 7, no. 3, pp. 219–242, **2016**.
[28] Y. Wang, Q. Chen, Q. Zhu, L. Liu, C. Li, and D. Zheng, A Survey of Mobile Laser Scanning Applications and Key Techniques over Urban Areas, Remote Sensing, vol. 11, no. 13, **2019**.
[29] T. Hemmes, Classification of Large Scale Outdoor Point Clouds Using Convolutional Neural Networks, TU Delft Architecture and the Built Environment, **2018**.

[30] Y. Wang, Q. Chen, L. Liu, X. Li, A. K. Sangaiah, and K. Li, Systematic Comparison of Power Line Classification Methods from ALS and MLS Point Cloud Data, Remote Sensing, vol. 10, no. 8, **2018**.

[31] M. Weinmann, M. Weinmann, C. Mallet, and M. Brédif, A Classification-Segmentation Framework for the Detection of Individual Trees in Dense MMS Point Cloud Data Acquired in Urban Areas, Remote Sensing, vol. 9, no. 3, **2017**.

[32] E. Che, J. Jung, and M. J. Olsen, Object Recognition, Segmentation, and Classification of Mobile Laser Scanning Point Clouds: A State of the Art Review, Sensors, vol. 19, no. 4, **2019**.

[33] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, Contextual classification with functional Max-Margin Markov Networks, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 975–982, **2009**.

[34] Y. Wang, H. Weinacker, and B. Koch, A Lidar Point Cloud Based Procedure for Vertical Canopy Structure Analysis And 3D Single Tree Modelling in Forest, Sensors, vol. 8, no. 6, **2008**.

[35] M. Rutzinger, S. J. O. Elberink, S. Pu, and G. Vosselman, Automatic Extraction of Vertical Walls From Mobile and Airborne Laser Scanning Data, in Proceedings of Laser Scanning, 1-2 September, Paris, France, pp. 7–11, **2009**.

[36] H. Yokoyama, H. Date, S. Kanai, and H. Takeda, Pole-Like Objects Recognition from Mobile Laser Scanning Data Using Smoothing and Principal Component Analysis, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVIII-5-W12, pp. 115–120, **2012**.

[37] S. El-Halawany, A. Moussa, D. D. Lichti, and N. El-Sheimy, Detectıon of Road Curb From Mobile Terrestrial Laser Scanner Point Cloud, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVIII-5-W12, pp. 109–114, **2012**.

[38] B. Rodríguez-Cuenca, S. García-Cortés, C. Ordóñez, and M. C. Alonso, Automatic Detection and Classification of Pole-Like Objects in Urban Point Cloud Data Using an Anomaly Detection Algorithm, Remote Sensing, vol. 7, no. 10, **2015**.

[39] W. Xiao, B. Vallet, K. Schindler, and N. Paparoditis, Street-Side Vehicle Detection, Classification and Change Detection Using Mobile Laser Scanning Data, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 114, pp. 166–178, **2016**.

[40] M. Soilán, B. Riveiro, J. Martínez-Sánchez, and P. Arias, Segmentation and Classification of Road Markings Using MLS Data, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 123, pp. 94–103, **2017**.

[41] Z. Sun, Y. Xu, L. Hoegner, and U. Stilla, Classification of MLS Point Clouds in Urban Scenes Using Detrended Geometric Features from Supervoxel-Based Local Contexts, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV–2, pp. 271–278, **2018**.

[42] J. Demantké, C. Mallet, N. David, and B. Vallet, Dimensionality Based Scale Selectıon in 3D Lidar Point Clouds, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVIII-5-W12, pp. 97–102, **2012**.

[43] B. Yang and Z. Dong, A Shape-Based Segmentation Method for Mobile Laser Scanning Point Clouds, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 81, pp. 19–30, **2013**.
[44] M. Zheng, M. J. P. M. Lemmens, and P. J. M. van Oosterom, Classification of Mobile Laser Scanning Point Clouds from Height Features, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-2/W7, **2017**.
[45] H. Thomas, F. Goulette, J.-E. Deschaud, B. Marcotegui, and Y. LeGall, Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods, International Conference on 3D Vision (3DV), pp. 390–398, **2018**.
[46] Z. Guo and C.-C. Feng, Using Multi-Scale and Hierarchical Deep Convolutional Features for 3D Semantic Classification of TLS Point Clouds, Taylor & Francis, **2018**.
[47] T. Rabbani, F. A. Heuvel, and G. Vosselman, Segmentation of Point Clouds Using Smoothness Constraint, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 36, **2006**.
[48] E. Grilli, F. Menna, and F. Remondino, A Review of Point Clouds Segmentation and Classification Algorithms, ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-2/W3, pp. 339–344, **2017**.
[49] D. H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, Pattern Recognition, vol. 13, no. 2, pp. 111–122, **1981**.
[50] Z. Ximin, W. Wanggen, X. Li, and M. Junxing, Mean Shift Clustering Segmentation and RANSAC Simplification of Color Point Cloud, International Conference on Audio, Language and Image Processing, pp. 837–841, **2014**.
[51] D. Comaniciu and P. Meer, Mean Shift Analysis and Applications, Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1197–1203, **1999**.
[52] J. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297, **1967**.
[53] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679–698, **1986**.
[54] N. Brodu and D. Lague, 3D Terrestrial Lidar Data Classification of Complex Natural Scenes Using a Multi-Scale Dimensionality Criterion: Applications in Geomorphology, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 68, pp. 121–134, **2012**.
[55] S. Filin and N. Pfeifer, Neighborhood Systems for Airborne Laser Data, Photogrammetric Engineering & Remote Sensing, vol. 71, no. 6, pp. 743–755, **2005**.
[56] M. Weinmann, B. Jutzi, and C. Mallet, Feature Relevance Assessment for the Semantic Interpretation of 3D Point Cloud Data, ISPRS Annals of

Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 5, pp. 313–318, **2013**.
[57] M. Weinmann, B. Jutzi, C. Mallet, and M. Weinmann, Geometric Features and Their Relevance for 3D Point Cloud Classification, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 157–164, **2017**.
[58] W. Zhang, J. Qi, P. Wan, H. Wang, D. Xie, X. Wang and G. Yan, An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation, Remote Sensing, vol. 8, no. 6, **2016**.
[59] L. Breiman, Random Forests, Machine Learning, vol. 45, no. 1, pp. 5–32, **2001**.
[60] K. J. Archer and R. V. Kimes, Empirical Characterization of Random Forest Variable Importance Measures, Computational Statistics & Data Analysis, vol. 52, no. 4, pp. 2249–2260, **2008**.
[61] N. Horning, Random Forests : An Algorithm for Image Classification and Generation of Continuous Fields Data Sets, **2010**.
[62] M. D. Deuge, A. Quadros, C. Hung, and B. Douillard, Unsupervised Feature Learning for Classification of Outdoor 3D Scans, p. 9, **2013**.
[63] B. Vallet, M. Brédif, A. Serna, B. Marcotegui, and N. Paparoditis, TerraMobilita/iQmulus Urban Point Cloud Analysis Benchmark, Computers and Graphics, vol. 49, pp. 126–133, **2015**.
[64] A. Serna, B. Marcotegui, F. Goulette, and J.-E. Deschaud, Paris-rue-Madame Database - A 3D Mobile Laser Scanner Dataset for Benchmarking Urban Detection, Segmentation and Classification Methods, **2014**.
[65] X. Roynard, J.-E. Deschaud, and F. Goulette, Paris-Lille-3D: A Point Cloud Dataset for Urban Scene Segmentation and Classification, IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2108–21083, **2018**.
[66] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss and J. Gall, SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences, IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9296–9306, **2019**.
[67] W. Tan, N. Qin, L. Ma, Y. Li, J. Du, G. Cai, K. Yang and J. Li, Toronto-3D: A Large-scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways, **2020**.
[68] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A.S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C.Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis and P. Schuberth, A2D2: Audi Autonomous Driving Dataset, **2020**.
[69] J. Awwalu and F. Ogwueleka, On Holdout and Cross Validation: A Comparison between Neural Network and Support Vector Machine, vol. 6, pp. 2394–9333, **2019**.
[70] A. K. Aijazi, P. Checchin, and L. Trassoudaine, Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach with Evaluation, Remote Sensing, vol. 5, no. 4, **2013**.

[71] Y. Yu, J. Li, C. Wen, H. Guan, H. Luo, and C. Wang, Bag-of-Visual-Phrases and Hierarchical Deep Models for Traffic Sign Detection and Recognition in Mobile Laser Scanning Data, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 113, pp. 106–123, **2016**.