RECOGNITION OF OBJECT INSTANCES IN MOBILE LASER SCANNING DATA

XIAOXU LI

February, 2015

SUPERVISORS:

Dr. K. Khoshelham

Dr. ing. M. Gerke

RECOGNITION OF OBJECT INSTANCES IN MOBILE LASER SCANNING DATA

XIAOXU LI

Enschede, the Netherlands, February, 2015

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.

Specialization: Geoinformatics

SUPERVISORS:

Dr. K. Khoshelham

Dr. ing. M. Gerke

THESIS ASSESSMENT BOARD:

Prof. Dr. Ir. M.G. Vosselman (Chair)

Dr. R.C. Lindenbergh (External Examiner, Delft University of Technology)

ABSTRACT

The overall objective of this MSc research is to develop a model-based method in order to detect and recognize different types of road furniture in Mobile Laser Scanning data. The proposed method first extracts an explicit model of the object, followed by a center based voting schema. The local maxima indicate the center location and the spatial configuration of the voters. The voters that correspond to the maxima are considered as the detected points of the objects. Generally speaking, model-based approach collects object model from the point cloud data and learns a pre-knowledge about the object shapes. It is robust to occlusion and does not require a fine segmentation beforehand; moreover, the method can deal with intra-class variability issue in object recognition of 3D point cloud.

The methodology is divided into four parts: model extraction and representation, center based voting, peak detection and performance evaluation. Firstly, model representation is based on the 3D Generalized Hough Transform; it uses the concept of the R-table, which stores all the shape descriptors of an object and represents it with a set of mathematical parameters. Secondly, the method adopts a center based voting schema for all the point cloud from the dataset, and the descriptors of every point in the dataset are indexed into the R table in order to find matched edge point of the same object type. Thirdly, the method applies Metropolis-Hastings algorithm for peak searching. The dense areas after voting, also called the local maxima, are considered as the potential centers of the objects. Strategies of selecting the correct candidate peaks are proposed afterwards. Finally, performance is evaluated given precision and recall and overall accuracy. The computational complexity of the modified method is also evaluated.

To sum up, the modification to the 3D GHT is feasible and reaches satisfied results; it turns the discrete Hough space into a continuous space. The input of the method is the object model and point cloud dataset, and the recognized object points are highlighted which have the same shape with input object model. Performance evaluation is done with confusion matrix, precision and recall. The recognized results reach high completeness. Correctness of the results is generally low, except for car object, the correctness of 82%. Thus the method can successfully find most objects from the data, but it also provides large amount of false positives. For most object types, the precision is extremely low, so further improvement need to be adopted in the future work to cope with this problem. In total, 129 out of 160 of the target objects have been successfully detected, which reach the overall accuracy of 80.63%.

Keywords

Mobile Laser Scanning (MLS), 3D Generalized Hough Transform (GHT), Mont Carlo Markov Chain (MCMC) sampling, Metropolis Hastings (MH), object recognition, road furniture detection

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to people who offer support for my MSc study and research during the 18 months staying in ITC, University of Twente. Without whom, I will not be able to make it.

Firstly of all, I would like to delicate my gratitude to my supervisors: first supervisor Dr. K. Khoshelham (Kourosh) and second supervisor Dr. ing. M. Gerke (Markus), for their brilliant suggestions and generous patience. Special thanks to my first supervisor, who has been extremely patient and inspiring. I also would like to thank Dr.ir. S.J. Oude Elberink (Sander) for providing me technique support, and PHD Biao Xiong and Mengmeng Li for giving me guidance and programming guidance. They encouraged me till the accomplishment of my research.

To my beloved family, I would like to express my greatest appreciation to them, who give me selfless love and encouragements; I love you deep from my heart. To my friends, for their loving care and accompany during my study life in the Netherlands: Gaoyan Wu, Chao Mai, Huiming Cai, Chunhui Shen, Mengxi Yang and particularly, Wei Zhang, who brings me luckiness and truly happiness.

In the end, thanks to all staff from Earth Observation Science department of ITC, the study experience is crucial in my life and their constant help and warmness make me feel at home.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
LiDAR	Light Detection and Ranging
MLS	Mobile Laser Scanning
ALS	Airborne Laser Scanning
TLS	Terrestrial Laser Scanning
RANSAC	Random Sample Consensus Algorithm
GHT	Generalized Hough Transform
ISM	Implicit Shape Model
CG	Computer Graphics
SVM	Support Vector Machine
RGB-D	Red Green Blue-Depth
BoW	Bag of Words
SURF	Speeded Up Robust Features
MBB	Minimum Bounding Box
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
A-R	Acceptance-Rejection
MVNPDF	Multivariable Normal Probability Density Function
PCA	Principle Component Analysis
PCM	Point Cloud Mapper
CAD	Computer Aided Design
DBSCAN	Density-based Spatial Clustering of Applications with Noise
KNN	K-nearest Neighborhood
TP	True Positives
FP	False Positives
TN	True Negatives
FN	False negatives

1. INTRODUCTION

1.1. Motivation

During the urbanization process, traffic volume and car ownership have increased rapidly and road safety management becomes a noticeable issue. Although Europe is regarded to be one of the safest road traffic regions in the world, it still suffers from traffic safety problems (Shen, Hermans, Bao, Brijs, & Wets, 2013). Three factors are considered to influence road safety: vehicle, driver behaviour and route environment or infrastructure (Mc Elhinney, Kumar, Cahalane, & Mccarthy, 2010). Among which, road environment safety is regarded to be a key issue, since broken or damage of transportation infrastructures like traffic light, traffic sign and lamp post can greatly lead to potential traffic accidents. Therefore, inspection and maintenance of the roadside facilities is of great importance and necessity.

Currently, many inspection surveys of road furniture are based on visually analysing 2D digital maps, videos or by manual in-situ inspections (Mc Elhinney et al., 2010). Such works are quite time-consuming and subjected to human eyes. In order to ensure a frequent inspection and maintenance of road infrastructures, roadside object extraction and detection with Mobile LiDAR technique is adopted, due to its abilities to acquire detailed, complete and high resolution terrain geometry efficiently and automatically.

Light Detection and Ranging (LiDAR) data is provided by laser scanners captured from different platforms. Compared with image-based 3D data acquisition, it provides more accurate and precise object geometry and it is less restricted to lighting conditions (Pu, Rutzinger, Vosselman, & Oude Elberink, 2011). Mobile laser scanning (MLS), also referred to as Mobile LiDAR, is a vehicle-based mapping system which provides 3D point clouds with high resolution and explicit object geometry (Kukko, Kaartinen, Hyyppä, & Chen, 2012). MLS provides an efficient and flexible solution for fast data collection (Williams, Olsen, Roe, & Glennie, 2013). It also provides dense point clouds with high precision for objects recognition and extraction, especially along road corridors.

In addition to MLS, airborne laser scanning (ALS) and terrestrial laser scanning (TLS) are also two techniques for acquiring LiDAR data. Although both approaches offer high quality point cloud data and detailed geometric features, they have drawbacks compared with MLS: ALS covers larger area but it has insufficient point density for detecting road geometric surface because of the viewing angle; TLS provides high point density within a limited small-scale area, but it is cumbersome for mapping long stretches of roads (Lehtomäki, Jaakkola, Hyyppä, Kukko, & Kaartinen, 2010). Thus, in the aspects of data acquisition, MLS offers dense point clouds with high efficiency and its mobility makes it more suitable for surveying road structures (Kukko et al., 2012).

1.2. Problem statement

Currently, many algorithms for semi-automated and automated road furniture objects identification have been proposed. However, most existing methods are more successful in recognizing objects in category level and less in instance level, e.g. they aim to detect poles rather than distinguish different pole types; they are based on implicit object features given certain global or local properties like shape, size etc. (Brenner, 2009; Lam, Mrstik, Harrap, Greenspan, & Engineering, 2010). Implicit object features are susceptible to inaccuracy and incompleteness of the data, whereas usage of explicit object model can fulfil the requirements of recognizing different instance types provided that each instance model is given. Thus, general approaches with an implicit knowledge of the objects are not discriminative enough to consider intra-class variability, e.g. distinguish and category road furniture into different instances such as different types of street lights, traffic signs, and traffic lights.

Moreover, partial occlusions and gaps are inevitable in point cloud data; they affect the accuracy of object recognition methods which are based on segmentation, clustering and classification. Therefore, a method which can be robust to occlusion is also required.

1.3. Research identification

1.3.1. Research objectives

The main objective of this research is to develop a method to detect and recognize different roadside objects in MLS data, with the capability to identify different types of road furniture. To achieve this overall objective, several sub-objectives need to be addressed:

1. Analyze the existing model-based algorithms for detection and recognition of roadside objects in 3D point clouds.
2. Develop a method according to a set of criteria, refine the existing algorithm with the expectations to increase the outcome quality and reduce the computational cost.
3. Test the results of the method with a ground-truth dataset (roadside scene in MLS data) and evaluate the performance.

1.3.2. Research questions

The following research questions should be addressed in order to achieve the above mentioned research objectives.

1. Among the existing object detection methods in 3D point cloud, which algorithms can be suitably applied or extended to recognize roadside instances in MLS data?
2. What are the limitations that influence the performance of the selected method, and how to optimize or reduce the drawbacks to the selected method?
3. How to obtain and store object models on condition that a model-based approach is used in this research?
4. How well or badly does the method perform in the context of object instances detection in MLS data?
5. What are the factors that influence the performance of the algorithm used in this research?

1.4. Innovation

The innovation is to develop a model-based method that can recognize road furniture in MLS data with the capability to deal with intra-class variation recognition. The method should be less depends on segmentation and can realize the research objective according to selection of parameters.

1.5. Thesis structure

The thesis is documented into 6 main chapters. Chapter 1 delivers the general introduction of the research, namely motivation, problem statement and research objectives and questions. Chapter 2 reviews several previous and current literatures on object recognition in point cloud. In chapter 3, the justification of the selection method and detailed framework of the methodology developed for the research is given. Chapter 4 shows the step-by-step implementations and results. Evaluation and discussion of the results are delivered in chapter 5, followed by specific constraints to refine the quality of recognition process. In the last chapter, final conclusions and recommendations are presented.

2. LITERATURE REVIEW

In this chapter, methods that related to object detection and road furniture detection are reviewed, and in the based on which, the chapter justifies the methodology developing process. Section 2.1 gives working principle of mobile laser scanning, section 2.2 reviews and simple object detection methods, subsequently in section 2.3, detection methods of complex and arbitrary objects are presented. In the following two sections of the chapter, concept and algorithm that has been adopted by the developed method are introduced. At last, summary on the reviewing methods is given.

2.1. Mobile laser scanning

Mobile laser scanning (MLS), also referred to as mobile light detection and ranging (LIDAR) technology, is a rapid, flexible and high resolution 3D data acquisition (Kukko et al., 2012). Mobile LiDAR is mounted on mobile platforms such as boats, vehicles, trains or snow mobile sledges(Puente, González-Jorge, Martínez-Sánchez, & Arias, 2013). In this research, the MLS dataset is collected by the Optech's Lynx Mobile Mapper system. Mobile mapping system contains five necessary components: the mobile platform; positioning hardware (e.g., GNSS, IMU); 3D laser scanners; photographic/video recording; computer and data storage (Williams et al., 2013) as shown in figure 2-1.

The general principle of laser scanning system uses LiDAR technique for range and angle measurements. Currently two techniques are used for MLS range measurements: time-of-flight (TOF) and phase shift (Puente, González-Jorge, Arias, & Armesto, 2012). The time delay between the scanner emit the laser pulse to the target and the received pulse reflected from the target surface can be used to calculate to range (Vosselman & Maas, 2010), as the equation 2-1 shows:

$$\mathrm{d} = \frac{\Delta t}{2} c \tag{2-1}$$

where c is the speed of light and Δt is the time delay. Phase shift measurement is more accurate with shorter range. The range is evaluated between the phase shift of emitted and received signals:

$$\mathrm{d} = \frac{\Delta\varphi}{2\pi}\frac{\lambda}{2} + \frac{\lambda}{2} n \tag{2-2}$$

where λ is the wavelength, φ is phase difference and n is the unknown number of full wavelengths between sensor and the target object (Puente et al., 2013; Vosselman & Maas, 2010).

MLS have many benefits: cost-efficient 3D data acquisition; high point cloud density (high level detail); increase safety of measurement (Puente et al., 2013), etc. Moreover, MLS offers more sufficient point cloud density than ALS because it equips more scanning angles for detecting vertical structures, and it is less affected by occlusion than TLS because of its mobility and the use of multiple sensors at different scanning planes (Cabo, Ordoñez, García-Cortés, & Martínez, 2014; Puente et al., 2012). These advantages make it extremely powerful for mapping navigable corridors for a wide range of applications including infrastructure analysis, utility mapping and inventory, 3D modelling and inspection of urban environment, etc.

Figure 2-1: Example of a mobile laser scanning (MLS) system and its components (Williams et al., 2013).

2.2. Object detection in point clouds

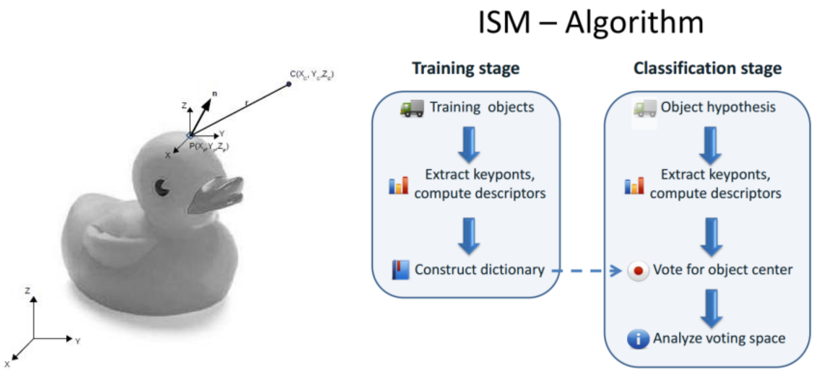
Existing methods of object recognition and detection in point clouds can be divided into two approaches: data-driven and model-driven, also known as bottom-up and top-down (Vosselman & Maas, 2010). Data-driven approach starts by extracting global or local properties (e.g. shape, size and location etc.) from the data, thus they are susceptible to occlusions and gaps in the data, and are not distinguishable enough to recognize intra-class instances and deal with multi-connected objects issue. Different from data-driven approach, model-based approach builds a geometric object model and locates the object from the dataset by verifying the model. Model-driven approach makes use of a beforehand knowledge of the object shape characteristics, while data-driven approach detects object at the point level. A detailed comparison of these two approaches was given by Tarsha-Kurdi, Landes, Grussenmeyer, & Koehl (2007). Model-based recognition is accomplished by matching correspondences of certain features between the scene and the model (Pope, 1994), however, 3D point clouds normally contains neighboring clutter, or occlusion and most objects do not have a full and complete shape. Two ways of describing object appearance are generally adopted in order to solve this problem: global property and local property. A detailed description of the two approaches was given in (Pham et al., 2013). Generally speaking, global feature requires a complete shape of the object whereas local feature needs information form a number of object parts, thus global method has the drawback that it is less successful to handle with partial, deformable shapes and occlusion in the data, and also lack of discrimination in terms with intra-class variations (Knopp, Prasad, Willems, Timofte, & Van Gool, 2010).

Simple geometric shape detection methods in point cloud such as detection of cylinders, planes, cones and spheres have been proposed by many works. Segmentation methods can extract geometric information such as edges, curves and surfaces form the point cloud data based on certain criterion. Mainly used segmentation and filtering methods are scan line segmentation (Sithole & Vosselman, 2005), surface growing (Vosselman, Gorte, Sithole, & Rabbani, 2004), connected components analysis, slope based filtering (Vosselman, 2000) and random sample consensus algorithm (RANSAC) (Fischler & Bolles, 1981). These methods can successfully recognize simple shapes, however, for detecting complex and arbitrary shapes, other object recognition methods need to be combined with or adopted to these segmentation and filtering methods.

Currently, object recognition methods have two categories according to the knowledge been employed: approach with implicit object feature and approach with explicit object model. In the former one, an object is represented with a feature implicitly in terms of the local or global appearances or shapes of the object; then the recognition process is done by searching similar feature vectors (Luo, Liu, Lin, Wang, & Yu, 2005). In the latter case, model-based object recognition attempts to define an explicit object model for all possible shapes of an object and match each object in the data with the given explicit model (Leibe, Leonardis, & Schiele, 2007). To describe an explicit object model, the object model can be mathematically described by a set of model parameters or represented with a set of templates (Khoshelham, 2007).

Hough transform method was an effective recognition method that was originally invented to recognize straight lines in 2D photographs (Hough, 1962). In this original invention, each point (x, y) in the object space correspond to a set of (α, d) in the parameter space. Thus, Hough transform can determine the location of the line by analysing intersections in parameter space with the highest number of curves. Besides detecting lines, the standard Hough transform can also be used to detect other parameterized objects such as planes and cylinders. Ballard (1981) extended this method in order to detect non-analytical object of arbitrary complex shapes. The Generalized Hough Transform (GHT) considered directional information of the boundary, and object model was stored in an R-table, which greatly increased the accuracy and computation speed. R-table contains a set of parameters to define an object shape, respectively reference point, the gradient direction for each boundary point, and the length (radius) as a function of gradient direction angle. An accumulator array is formed, for each edge pixel, increment the corresponding votes in the array. After the voting process, the local maximum (the bin with maximum votes) in the array indicates the reference point and the edge that casts the vote of the bin indicates the shape of the object. The method can also be used to detect objects with rotation and scale variance, in which the accumulator array is four dimensional. However, the drawback of the method to be applied into point cloud data is its huge time and space complexity. Later Khoshelham (2007) extended the GHT method and enabled it to detect more complex and arbitrary shapes in 3D point cloud data as shown in figure 2-2(a). But the main drawback of the GHT still remains.

A robust object recognition method with implicit shape model (ISM) was first proposed by Leibe & Schiele (2003). It was applied for object detection in MLS point cloud by Velizhev, Shapovalov, & Schindler (2012). The method followed the general pipeline by Golovinskiy, Kim, & Funkhouser (2009) and used implicit shape model for the classification phase (Knopp et al., 2010). The algorithm is shown in figure 2-2(b). It builds a model dictionary by extracting key points and computing their descriptors (spin images), and then recognize objects by voting for the object center. In the training stage, descriptors of the key points and their displacement from the object center of the exemplar for each class were clustered to build a geometric dictionary. In the classification stage, a voting-based localization was performed to match descriptors and cast votes for the object center. The method requires small training data, but it does not take density variations into account and fails on objects with badly defined shapes.



(a) (b)

Figure 2-2: (a) Parameters of 3D GHT method (Khoshelham, 2007). (b) ISM algorithm (Velizhev et al., 2012).

In object recognition by template matching method, the object model is represented with a set of voxel templates. Greenspan & Boulanger (1999) proposed an efficient and reliable template matching method that can be applied in range image data. The model was rotated at each pose and its surface was quantized into a voxel space. In such a way, the object was represented by each view of the model in a specific pose. To solve the expensive computational cost, all the templates are composed into a binary decision tree, where each leaf node refers to a number of templates and each internal node refers to a single voxel. Recognition was done by first voxelizing the range image and selecting the seed voxel randomly and iteratively.

The method by Song & Xiao (2014) used depth information for object detection. Computer graphics (CG) CAD models were collected from internet and rendered from hundreds of viewpoints to obtain synthetic depth map. A feature vector was extracted from 3D point cloud projected from depth map and trained an exemplar Support Vector Machine (SVM) classifier for each rendering with both positive ground truth (renderings from CG model) and negative data (from labelled Kinect depth map). Then a 3D detection window was sliced to match the exemplar shape. The method claims to solve several main difficulties in object recognition in RGB-D images: texture, illumination and shape variance, viewpoint variance, noise and sensor error, clutters and occlusion. The method was illustrated in figure 2-3. During testing, the SVMs were used to classify the bounding box in 3D space and decided whether the target shape was inside the box with detection scores. The method achieved 1.7 times improvement average precision compared to the results of using RGB images.

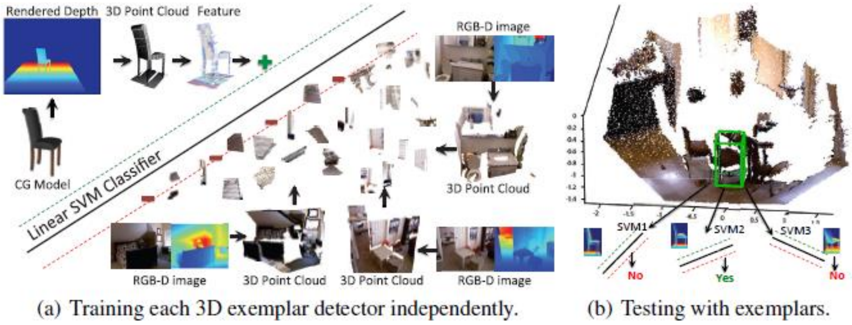


Figure 2-3: Sliding shapes method (Song & Xiao, 2014).

2.3. Road furniture detection in point clouds

Road furniture is the general designation that contains all roadside objects used for traffic safety and control that aims to facilitate and assist drivers. Types of road furniture includes traffic signs, lights and utility poles, road markers, mail boxes, telephone booth or other essential elements.. Among which, infrastructures such as traffic lights, traffic signs, street lights can be curtail to improve traffic safety and control, and are important for car navigation and driver assistance.

In recent years, researches start to address on pole-like road furniture detection and recognition in urban environments from mobile laser scanning data (Cabo et al., 2014). A model-based pole extraction method was mentioned by Brenner (2009), in which pole-like objects can be seen as a special cylinder extraction case. The poles are assumed with an upright characteristic, and a kernel region is used where inside the region are the laser points that represent the pole and outside the region there are less points or no points. The locations of the poles are estimated with respect to the cylinder stacks, and a pole structure is identified by setting a certain minimum number of stacked cylindrical stacks. However, this method cannot extract poles with additional structures attached to the poles and thus it cannot be used to recognize such road furniture as traffic signs or traffic lights with curved tops.

Golovinskiy, Kim, & Funkhouser (2009) investigated a general pipeline for recognizing small objects in urban areas. The pipeline contains four steps, namely localization, segmentation, representation and classification. In figure 2-4, potential objects are first localized by filtering out large parts that do not belong to the objects, followed by a segmentation process to identify object shapes. In the feature extraction phase, objects are described by their shape and context properties with feature vectors. Finally the candidate objects are classified and labelled. In each step, the paper provided several alternative approaches. The performance results show that 65% of the objects are successfully recognized. However, the pipeline has the drawback that it is not discriminative enough for intra-class variability (Velizhev et al., 2012) to recognize different object types.

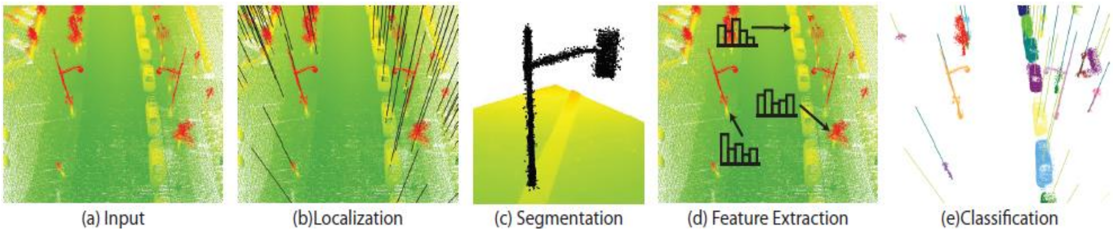


Figure 2-4: Working pipeline in Golovinskiy et al. (2009).

Lehtomäki et al. (2010) proposed another method for pole detection and classification in road environments. The method uses the profile information of the vehicle-based laser scanner, and it assumes that pole is found from a minimum of three sweeps. The phases of the method consists of four phases: scan line segmentation of each profile, clustering of possible sweeps of pole with shorter point segments, merging of clusters, classify candidate clusters into different pole objects based on feature properties like shape, length, point density, etc. However the accuracy of pole detection suffers from shadowing effect between objects, and it is not independent of scanning geometry (i.e. scanning angle and frequency) of the sweeps.

A knowledge-based method was proposed by Pu, Rutzinger, Vosselman, & Oude Elberink (2011) shown in figure 2-5 to recognize roadside structures in MLS data. There are three main steps for feature recognition: rough classification, percentile-based feature recognition and further classification. The purpose of rough classification is to separate ground and non-ground objects and obtain segments that contain interested components. Then percentile-based algorithm was applied: candidate segment is divided into height percentiles, and a certain percentile is divided into horizontal slices (fit a rectangular plane), afterwards, the method computes the diagonal length and center point for each slice. The pole recognition was thus performed by counting the length and center point deviations iteratively between neighboring slices within a certain threshold. Finally, the detected poles are further classified with knowledge-based shape recognition. The performance of the method can be influenced by point cloud density variation and occlusion problem in the presentence of thick trees.

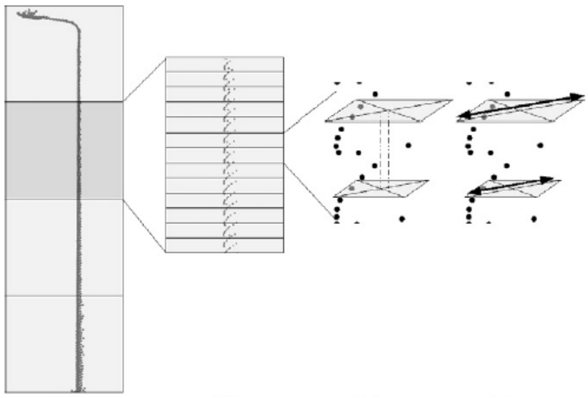


Figure 2-5: Percentile-based algorithm in Pu et al. (2011).

The thesis from Kemboi (2014) used another knowledge-based method to detect and extract road furniture, in which the author described the parameters of the segments by use of histogram and minimum bounding box (MBB). Then computation of the histogram correlation MBB ratios was applied to find similar components with a known sample. The principle behind is that different components of object type tend to have different pattern on the shapes of histogram, and can be described with different set of bounding box parameters. Then the objects are classified by decision tree based on the MBB parameters and correlation thresholds. Further check of the results is by use of iterative closet point (ICP) algorithm. Different from the method by Pu et al. (2011), this method used explicit object sample in the dataset for detecting and recognizing different object types.

More recently, Cabo et al. (2014) presented a method using spatial volumetric elements of the point clouds to detect pole structures, as shown in figure 2-6. The method was divided the space into regular grid. The voxelization step simplified point cloud based on the codification of the XYZ coordinates of the points into a 12-digit code in voxel unit and the values of the code were stored in a vector, where points in the same voxel have the same code. Then the two-dimensional analysis was carried in three stages to find pole candidates based on the properties of the sections: segmentation on the divided horizontal sections from the grid, select the horizontal elements with maximum area, select the element with an isolation criterion. The result of the step was a 2D set segments associated with a Z coordinate. Finally a tridimensional neighborhood analysis was applied to group the voxels, and identified the pole structures with a minimum height threshold. This algorithm can detect poles without assumption of its position, however, it cannot distinguish different pole instances and moreover, it would fail if there are severe occlusions and existence of thick surroundings with respect to a pole.

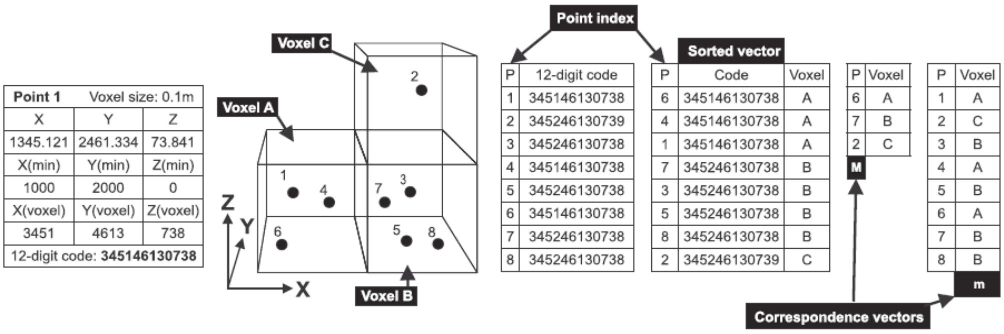


Figure 2-6: Example of codification process (Cabo et al., 2014).

2.4. Bag-of-words concept

Bag-of-words (BoW) concept is popularly used for object categorization. The key idea is to quantize extracted features into visual words, and represent them with a sparse histogram over the vocabulary (Zhang, Jin, & Zhou, 2010). The BoW framework has been proposed for textual document classification and retrieval (Toldo, Castellani, & Fusiello, 2010). Extending the approach to non-textual data requires building the visual vocabulary. Sivic & Zisserman (2003) came up with the bag-of-features (BoF) approach to build a visual vocabulary (a set of all the visual analogue of words) to quantize the shape descriptors into clusters in 2D image. The BoW concept can be adopted with the model based approach in object recognition methods, in which, the feature characteristic of an object model can serve as a codebook. Shapes from the scene can be matched with the correspondences in the codebook during the recognition process. Then objects can be detected by determining the similarity or correlation between the codebook and the matching correspondences in the data.

The BoW concept has been applied by some methods already. In the original paper of 2D ISM method (Leibe & Schiele, 2003), a codebook of local appearance of the object shape was built with training objects, shown in figure 2-7. ISM can be seen as a combination of visual dictionary (Sivic & Zisserman, 2003) and generated Hough transform (Velizhev et al., 2012). ISM method generated object hypotheses with a top-down approach by generating a codebook of the local appearance object shape, followed by a probabilistic voting schema to help recognize object categories. In the recognition process, image patches were extracted around interest points and matched to the correspondences in the codebook. Instead of activate single entry, the patches were firstly clustered based on their similarity, and for every codebook entry, the center position was stored as well. Matching patches cast probabilistic votes in the continuous voting space, and the hypotheses were searched and localized by recognizing the local maxima in the voting space.

In the paper by Knopp et al. (2010), a shape representation with a set of 3D SURF (Speeded Up Robust Features) features were introduced combined with the idea of ISM method applied in 3D mesh models. 3D SURF descriptor was computed around each interest point; it is scale and rotation invariant to reconstruct the object shape. An explicit model that assembles each class was built by SURF feature with the training data. ISM was applied to build a 'visual vocabulary' and to achieve correct classification in the query data. The voting process considered learning and statistical weights.

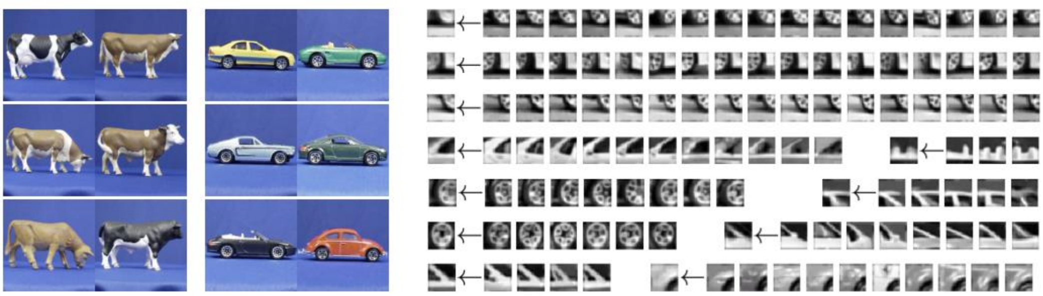


Figure 2-7: Codebook construction process in the original ISM method (Leibe & Schiele, 2003).

2.5. Markov Chain Monte Carlo (MCMC) sampling

Mont Carlo simulation (emphasizes on probabilistic machine learning) can be used to draw samples from target distribution in high dimensional space. Markov chain Monte Carlo (MCMC) sampling is well-known technique that plays a significant role in the application of statistics, physics and commutating science, etc. The algorithm can generate samples from probability distributions using a Markov chain mechanism (Andrieu, Freitas, Doucet, & Jordan, 2003). The mechanism is conducted so that it can spend more time on the important regions of the target distribution $\mathrm{p(x)}$. Given a state space $x^i \in X = \{x1, x2, \ldots, xn\}$, a Markov chain is defined as:

$$\mathrm{p}\left(x^{(i)} \middle| x^{(i-1)}, \ldots, x^{(1)}\right) = \mathrm{T}\left(x^{(i)} \middle| x^{(i-1)}\right) \tag{2-3}$$

where T is the transition matrix. The evolution of the chain only depends on the current state of the chain and a fixed transition matrix. In another word, regardless of the initial state, the chain will stabilizes at a certain value eventually. Thus for any arbitrary start point, the MCMC algorithm can converge to a stable and invariant distribution given the fixed transition matrix.

MCMC method can be used to simulate multivariate density-based distributed samples. It can be used for local maxima filtering, especially when the distribution has large probability mass around the mode (Andrieu et al., 2003). The chain can be defined so that it takes more samples from the around the mode of the simulated distribution so that the mean of the samples show the mode of the distribution. This makes MCMC a suitable solution for finding the mode from a density distribution in object detection filed, thus it overcomes the shortcomings of the high computational complexity of peak detection process in 3D GHT method.

Several implementations of the MCMC method have been proposed, among which, the Metropolis Hastings (MH) algorithm was regarded as one of the ten algorithms that have had greatest influence on the development and practice of science and engineering in the 20th century (Andrieu et al., 2003). MH algorithm is an popular instance of MCMC sampling methods that was first developed by Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller (1953) and generalized by Hastings (1970). The algorithm can obtain random samples from a density distribution; it can approximate the desired distribution based on the probability of acceptance or rejection of a candidate.

2.6. Summary

This chapter mainly reviews some existing methods of object detection and recognition in 3D point cloud data. The existing methods can be divided in to two kinds, data-driven and model-driven. In data-driven methods, object points are grouped or categorized by means of segmentation, classification or clustering. However with the presence of occlusion they are not very robust. Model-based approaches make use of shape information of an explicit object instance, and the objects are represented with global or local shape descriptors, templates, or mathematical parameters. The reviews mainly focus on the road furniture detection where many methods have attempted to classify different road inventories. Among which, the extended 3D Generalized Hough Transform method has the strength of model based approach and the ability to describe an object instance accurately. On the other hand, modifications are necessary to make compensate to the computational complexity issue of the selected method. MCMC sampling is selected for its efficiency in finding the mode from a density distribution. Therefore, by combining the two methods, it is considered can be feasible and promising in realizing the research objective.

3. METHODOLODY

3.1. Introduction of methdolody

3D GHT method has the advantages that it transforms shape detection into a maximum analysis problem in a mathematical way. It is also very robust to partial occlusion issue as it can reconstruct object model explicitly and it uses a center based voting schema to solve unknown parameters in the Hough space. It is capable of finding multiple correspondences in the same voting process (Bevilacqua, Casorio, & Mastronardi, 2008). However, the main drawback of this method is that it requires large computational cost, especially when taking rotation and scale parameters into account, and the voting space is increased to 7 dimensional. Also, as GHT uses discrete voting space, information is lost to some extend for every voting process. To overcome these problems, the idea of modifying the discrete Hough voting space into continuous voting space would be a good solution. Therefore, the method draws the key idea from ISM method (Velizhev et al., 2012), where shape descriptors were collected as a geometric dictionary together with their spatial configuration form the object center, and followed by a voting based localization process to detect maxima.

As the voting results of 3D GHT is large, to reduce computational cost, MCMC method is applied for detecting peaks from the voted data, given the appropriate proposed distribution and parameters of MH algorithm. Such modification avoids the high computational complexity in the Hough space. Moreover, applying MCMC helps directly find target peaks and the mode point in the continuous space, which further improves the efficiency of the method.

3.2. Framework

The research applies a model-driven approach to recognize different road furniture types in MLS data based on explicit models. Individual model is extracted from the dataset as input, and object centers are calculated using a center based voting procedure of 3D GHT. The general process of the research is divided into four phases: data preparation, model representation, peak detection, refinement and evaluation of the final results.

In data preparation phase, object models are manually extracted from the point cloud dataset to serve as inputs for setting up the R-table. A proper pre-examination of the data is required to choose a well-shaped object as the instance model. Then normal vectors are calculated for both the object model and the input dataset. In order to save computational burden, the dataset is been removed of ground points and been cut into subparts of approximately 100 meters along the trajectory.

The second and third phases are main parts of the methodology. Object models are stored in R-table format according to the 3D GHT method. In R-table, the connecting vectors **r** presents the displacement information between the center and the edge point of the object. The object model is denoted with a set of mathematical parameters from R-table. Based on the voting schema, all points vote for their possible centers and thus generate a collection of center data. In the detection phase, MCMC sampling method is selected for finding local maxima. Given specified parameters, MCMC method simulates multivariate samples and its mode point from the center data. Voters that cast votes to the local maxima are considered as the detected points of the object.

To refine the detected results, quality measurement strategies are applied to help eliminate the wrong candidates. Performance of the final results is evaluated and assessed in the last phase by visual inspection into the dataset and by precision and recall examination compared with a labelled ground truth dataset. The framework of the methodology designed is illustrated in the figure 3-1.

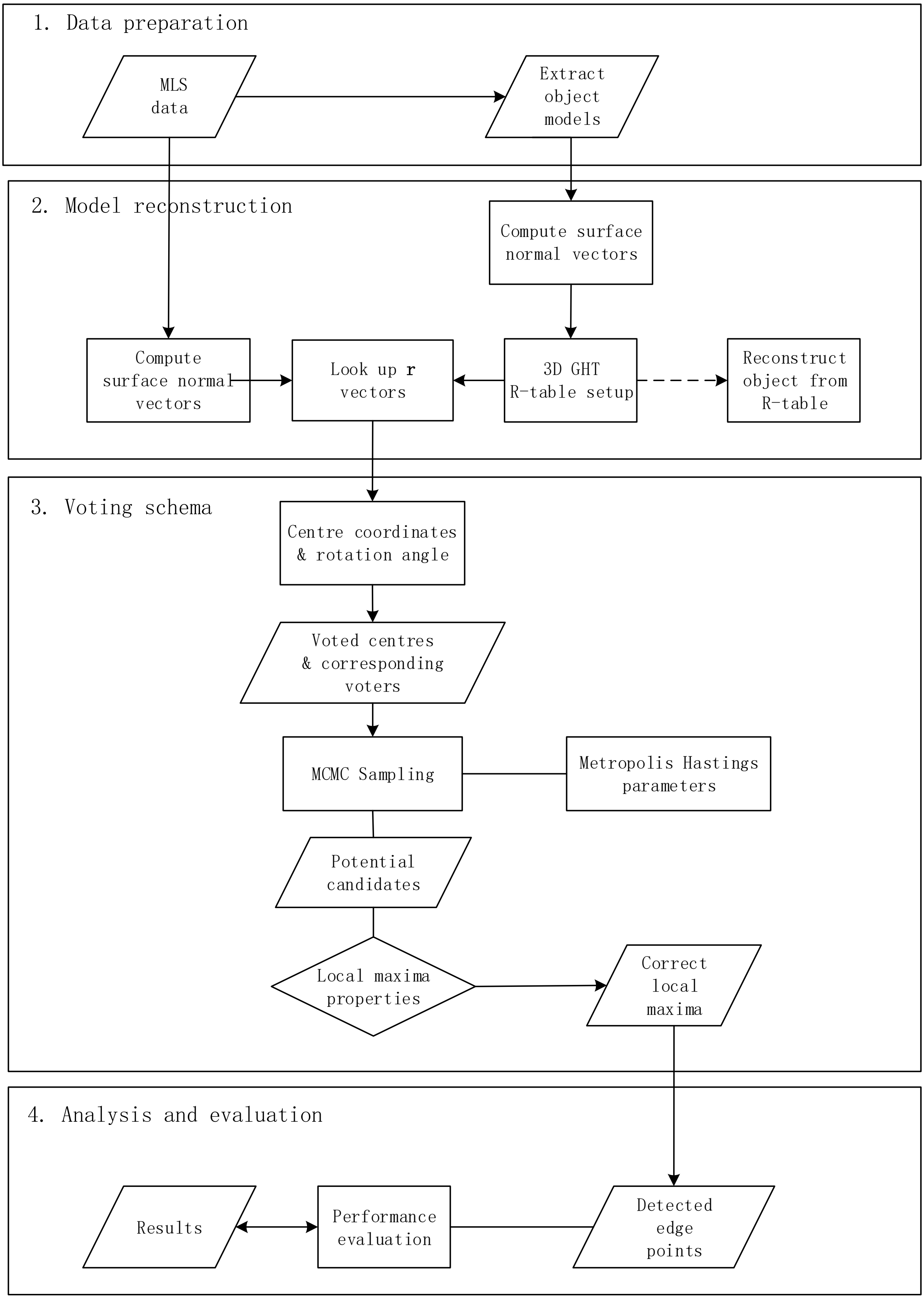


Figure 3-1: Framework of the methodology.

3.3. Data preparation

A simplification and partition of the dataset is necessary to reduce the computing burden. We partition the dataset into subparts so that each dataset contains a certain amount of points and objects, and also for a large input dataset, the results of voting will be too large to fit into the memory. There is no requirement of a perfect segmentation of the point cloud, the dataset only has been removed of ground points and left only the roadside objects.

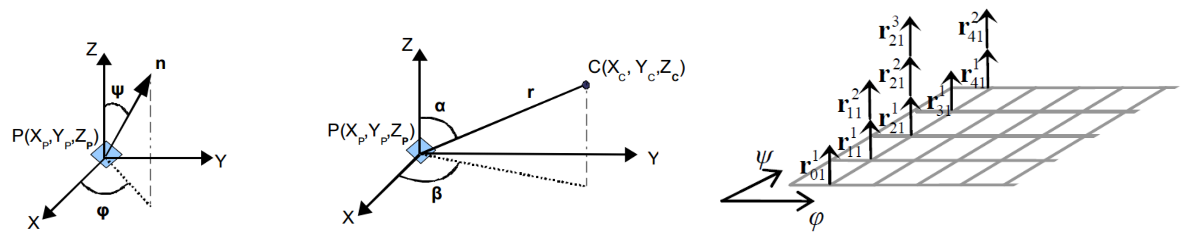
The data preparation process also includes calculating the normal vectors of the points. For an explicit instance, normal vector of each point is used to set up R table with the two orientation angles for model representation; and for points in the dataset, since normal vector is used as index in the later stage of obtaining 3D GHT parameters. Therefore, for every point in both the model and the dataset, their normal vectors need to be calculated. This is done by triangulating the object surface and computing the normal vector of each point. Normal vector (n) is defined by two orientation angles with respect to the normal direction and is of unit length as illustrated in figure 3-2(a).

3.4. Model representation

Firstly, an object model is selected from the dataset. The model is extracted and saved it into a separate file. Then the instance model is represented with parametric equations of 3D GHT with R-table. In 3D GHT method, the connecting vector **r** is used as the shape descriptor of an object. A connecting vector **r** is denoted by the two orientation angles of the normal vector, and r is the length between the reference point and a surface point. For an instance model, parameters of connecting vector can be computed from the coordinates of the reference point and the edge points as in formula 3-1.

$$
\begin{gathered}
r=[(x_p-x_c)^2+(y_p-y_c)^2+(z_p-z_c)^2]^{1/2} \\
\alpha=\arccos\left(\frac{z_c-z_p}{r}\right) \\
\beta=\arccos\left(\frac{x_c-x_p}{r\sin(\alpha)}\right)
\end{gathered}
\tag{3-1}
$$

To construct the R table, an arbitrary reference point need to be selected for the object, and each point of the object is represented by a vector **r** from a surface point to an arbitrary reference point. R-table cells store all the connecting vectors indexed by the two orientation angles of the normal vector. Regardless of rotations and scale, there are 3 parameters involved (figure 3-2(a)). An **r** vector represents the spatial configuration (distances and orientations) between reference point and surface points of the model. It is stored in R table as shown in figure 3-2(b). Object model can be reconstructed by a set of equations as illustrated in formula 3-2, involving the coordinates of reference point x_c y_c z_c and the **r** vector. The shape of an object thus can be described in this R table format.



(a) (b)

Figure 3-2: (a) Parameters involved in 3D GHT method. (b) Storing connecting vectors in the R-table.

The method represents the instance model and describes the connecting information from the reference point c to a surface point p according to formula 3-2. The R-table method is efficient and accurate for object representation in MLS data.

$$\begin{cases} x_p = x_c - r \cdot \sin(\alpha)\cos(\beta) \\ y_p = y_c - r \cdot \sin(\alpha)\sin(\beta) \\ z_p = z_c - r \cdot \cos(\alpha) \end{cases} \tag{3-2}$$

3.5. Generation of center points

For detecting objects from the dataset, the coordinates of reference point is the unknown parameters to be solved. We select the center point as reference point, thus the calculating of the reference point can be carried out by arranging the formula 3-2 into formula 3-3:

$$\begin{cases} x_c = x_p + r \cdot \sin(\alpha)\cos(\beta) \\ y_c = y_p + r \cdot \sin(\alpha)\sin(\beta) \\ z_c = z_p + r \cdot \cos(\alpha) \end{cases} \tag{3-3}$$

However, in reality, objects do not always appear in one pose, so more parameters are needed to add to the formula to deal with arbitrary scale and rotation, and this is also where computational complexity is increased. To consider the rotation and scale, the Hough space is extended into 7 dimensional:

$$\mathrm{c} = \mathrm{p} + \mathrm{s}M_x M_y M_z \cdot r \tag{3-4}$$

where $\mathrm{c} = (x_{c,}y_c, z_c)^T$, $\mathrm{p} = (x_{p,}y_p, z_p)^T$, $r = (r \cdot \sin(\alpha)\cos(\beta), r \cdot \sin(\alpha)\sin(\beta), \cos(\alpha))^T$, s is the scale factor and M_x, M_y, M_z are the rotation matrices around x, y, z respectively.

Since roadside objects only have rotation differences around z axis, which frees 2 orientation parameters. Moreover, in real scene, the fact same kind of traffic instance has the same size, thus lead to the elimination of the scale parameter. Therefore, in this research, to reconstruct different kinds of road furniture instance, the final equations can be formulated as:

$$\mathrm{c} = \mathrm{p} + M_z r \tag{3-5}$$

where $M_z = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$, θ is the rotation angle around z axis. In this way, only 4 unknown parameters remain to be solved. An example of representing an object model is illustrated in figure 3-3.

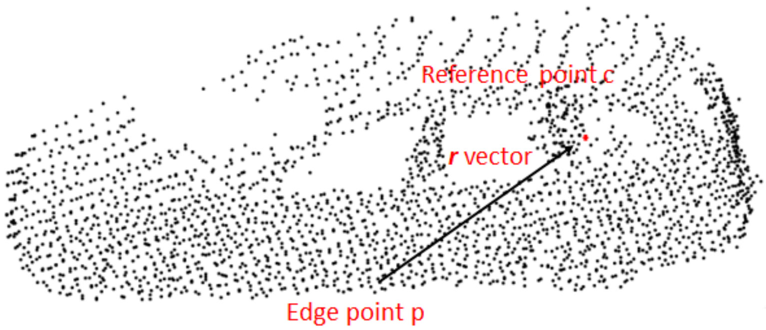


Figure 3-3: Representing a car model with R-table.

3.6. Voting and peak detection

3.6.1. Center-based voting process

The center-based voting process generates voted center points, it follows the idea of 'visual dictionary' concept; the R table of the model serves as the dictionary. To start, the normal vectors of each point in the dataset are computed, the two orientation angles serve as dictionary entry during the voting process. The normal vector is used to look up **r** vectors in the R table at the corresponding angles. For every entry, the matched **r** vector evaluates the possible centers according to formula 3-5, and it is also the process of casting votes for the location of the object center.

As three coordinates of the edge point cannot provide sufficient redundancy to solve 4 unknown parameters, the solution is to give θ in angle intervals in discrete. So for each θ spacing around z axis, the voting process works as follows:

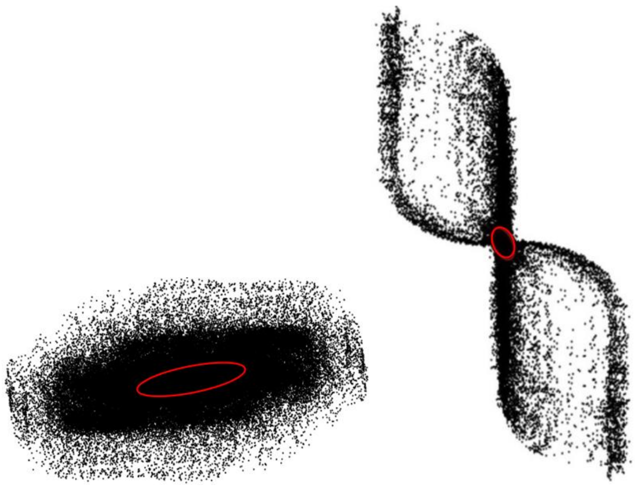
1. Compute the normal vectors for every point in the point cloud.

2. Look up $\mathbf{r}$ vectors in the R-table at the coordinates of (α, β).

3. Evaluate equation $\mathrm{c} = \mathrm{p} + M_z r$ with the corresponding $\mathrm{r}, \alpha, \beta$, and obtain a set of the possible center coordinates xc, yc, zc.

4. Repeat the process for all points in the point cloud.

After the voting process, the center location of the object will receive most votes and is defined as the local maxima. The local maxima are clusters of compact points, and their property differs with different input object models as shown in figure 3-4. Thus, if the property of the correct local maxima can be identified, and the correct object then can be detected.



(a) (b)

Figure 3-4: The local dense area of a car (a) and a lamppost (b).

The voting process generates clusters of voted center points of different objects in the point cloud. The local maxima that has the densest distribution of center points indicates the reference point of the object, and the 3D points that cast votes for the local dense area belong to the instance of the object in the point cloud. However, a problem arises also when generating the maxima in the voting process: all points that satisfy the entry criterion cast votes for their possible centers and thus not only the correct area of the reference point is incremented, but also other areas are been accumulated. Thus it is necessary to propose an efficient method to detect the correct maxima.

3.6.2. Metropolis-Hastings (MH) algorithm

Theoretically, the correct local maxima are the densest area because it receives most votes from the voters. However, the density distribution of generated 3D center points is with arbitrary curves and contains multiple extremes. In order to analyze the distribution, in this section we applies MCMC sampling method to detect local maxima and find the peak of the distribution.

The distribution that needs to be drawn samples from is the 3D distribution of the generated center points. MCMC method is proposed because it can be used to sample complex and non-analytical distributions and find the mode (mean) of the samples. After the center based voting process, we obtain an arbitrarily density based distribution in 3 dimensional, so it is hard to pick out the correct peaks of this distribution directly. To analyze such distributions (e.g. to find their mode) we need to sample them. The densest point, which is the peak of the density based distribution, is found by taking adequate samples from the distribution and calculating the mean value of the samples.

MH algorithm is an implementation of the MCMC sampling, the algorithm takes the samples by running a Markov chain starting from a random initial point. It is based on an Acceptance-Rejection (A-R) sampling method to generate samples from an absolutely continuous target distribution (Zhang et al., 2010). It takes the samples by running a Markov chain starting from a random initial point and the mode of the distribution is found by taking adequate samples using a Markov chain. The principle of MH algorithm is simple: the step involves an invariant target distribution $p(x)$ and a proposal distribution $q(y|x)$, where y is a candidate value given the current value x according to $q(y|x)$. The Markov chain moves towards y according to the probability of move: $A(x, y)$ If the move is not made, the process remains at x as a value of the target distribution. The value of $A(x, y)$ determines how often the movement from y to x, and acceptance probability is define as:

$$A(x^i, y) = \min\left\{1, \frac{p(y)q(x^i|y)}{p(x^i)q(y|x^i)}\right\} \tag{3-6}$$

The pseudo-code is shown in figure 3-5:

1. Initialize x^0
   N = iteration times
2. For i=1:N
   -- Sample u ~ U(0,1).
   -- Sample y ~ q(y|x^i).
   -- if u <= $A(x^i, y) = \min\left\{1, \frac{p(y)q(x^i|y)}{p(x^i)q(y|x^i)}\right\}$
   $x^{i+1} = y$
   else
   $x^{i+1} = x^i$
3. Returns the samples {x1,x2, …,xn}.

Figure 3-5: Pseudo-code of MH algorithm from (Andrieu et al., 2003).

In this method, the MH algorithm finds the local maxima by learning from the information generated from the model. The performance of MH algorithm depends on the choice of parameters, e.g. different proposal distribution lead to different results. Thus, given the appropriate proposal, the MH algorithm will find the maxima from the target distribution. The correct dense area is composed of a set of compact distributed 3D points where the mode of which is the densest.

The proposal distribution is determined to use the multivariable probability density function (mvnpdf) given covariance value along x, y, z axis of the data. Since the probability distribution of the points of local maxima is in arbitrary shapes, so it is hard to analyze by fitting curves or using a function to represent its shape. So we choose mvnpdf as the proposal distribution because it has an analytical form (a simple equation), yet it is sufficient to represent the local maxima shapes, e.g. mimic the density distribution (peaks) of the center points. The detected local maximum of a lamppost is shown in figure 3-6.

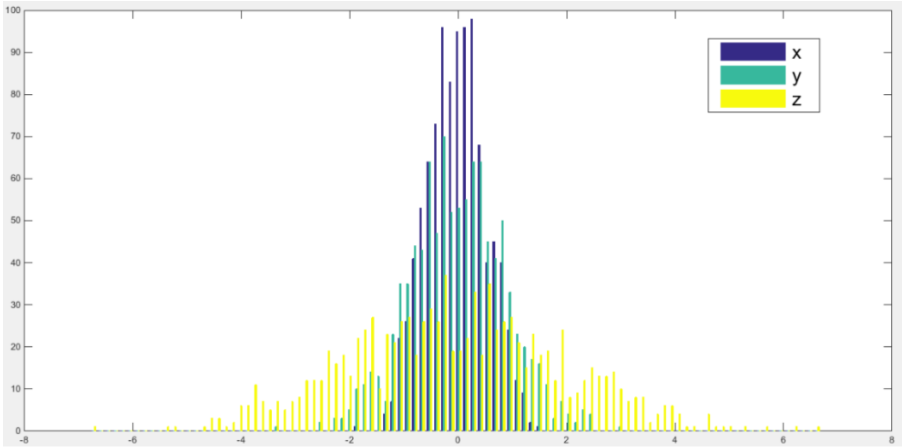
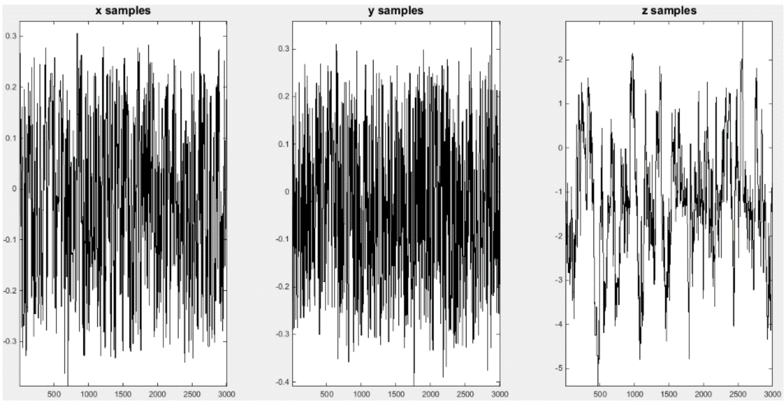
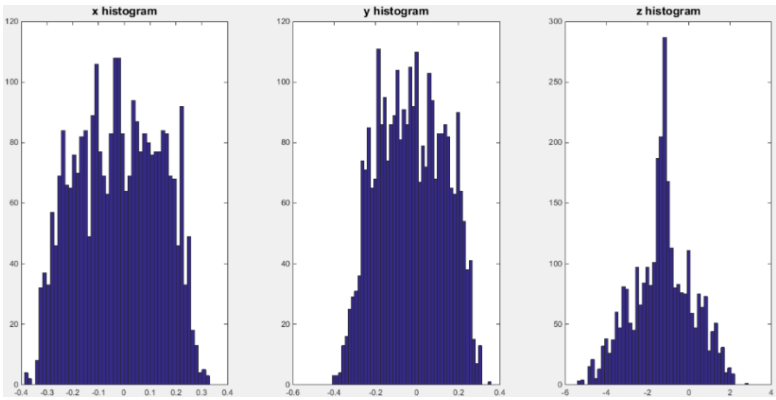


Figure 3-6: Mvnpdf distribution of the lamppost model 1 given the covariance matrix along x, y, z.

Figure 3-7 shows that the MH algorithm randomly initializes one point from the data, following the A-R rule, the Markov chain moves towards to the satisfied candidate, and the sampling process will come to the convergence eventually.



(a)

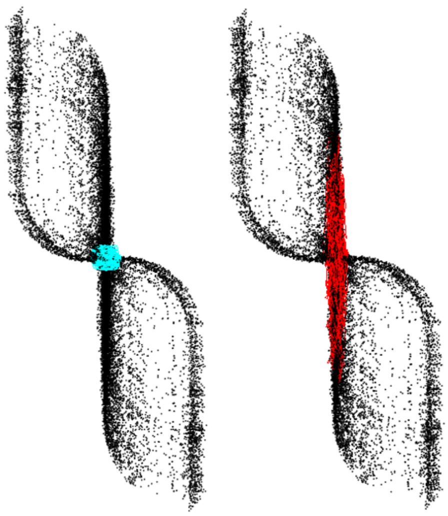


(b)

Figure 3-7: Convergence (a) and histogram (b) plot of the sampling points along x, y, z.

The covariance value (sigma) of the proposal function (mvnpdf) is an important parameter of MH algorithm. It affects the behavior of Markov chain in two ways since it defines the range of search. Firstly, it affects the acceptance rate of a candidate; secondly, it affects the region of sampling space (Chib & Greenberg, 2012). Assume the situation that the sampling process is already come to a convergence. If the moving of Markov chain is too large, the sampling point generated will be very far away from the previous value so that the acceptance chance is low. If the search range is too small, then the chain takes more times to traverse the data to find qualified density area, thus there will be under-sampled problems.

The sigma value of proposal distribution (mvnpdf) is determined according to information generated from the model. Let the center based voting procedure apply on the object model itself to obtain the voted center points of the model. Find the densest area (around the mode point) of the center points of the model and calculate the covariance value in three dimensional x, y, z. When the sigma value is properly set, the sampling result of the MH algorithm is shown as in figure 3-8. In such a way, the peak can be detected and obtained by calculating the mean value of the samples and then define a neighboring area around the mean point.



(a) (b)

Figure 3-8: (a) Neighboring points around the mode. (b) The Markov chain in points.

3.7. Strategy of quality measurement

Not all the sampling results obtained from the MH algorithm are correct maxima, because this method not only finds out the corresponding same objects from the point cloud, it also discovers the object parts that have similar shape with the input model. In order to select the correct local dense area from all the candidates, restrictions and thresholds need to apply to differentiate the sampling results. This process is carried out from three aspects: shape, location and quantity of the detected local maxima.

1. Number of neighboring center points around the sampling mode

   For those wrong local maxima, their voters have the same entries with the input model, or because the algorithm discovers dense area that meet the acceptance criteria in equation 3-6. MH algorithm provides the mode point of these potential local dense areas, as well as the neighboring points around the mode. Theoretically, the correct local maximum receives most votes from points of the object, thus make it a local dense area. Correct results are denser than the wrong ones, thus the number of neighboring points around the sampling mode is larger. According to this, by determining threshold of the number of neighboring center points around the sampling mode, most of the wrong candidates can be eliminated.

2. Mean height of neighboring center points around the sampling mode

   Different objects have different location information, e.g. cars are located on the ground and have lower mean height value, and lampposts have a larger mean. This kind of information can also be obtained from the model. Among the detected objects, some have similar part with the model, however they are wrongly located in the point cloud data, e.g. a car shape can be contained on the wall, and a part of lamppost shape can be contained within other pole-like shapes. Thus, provided the mean height information, some cognitive faults can be avoided. An example is shown in figure 3-9.

Figure 3-9: Wrongly detected points in similar shapes.

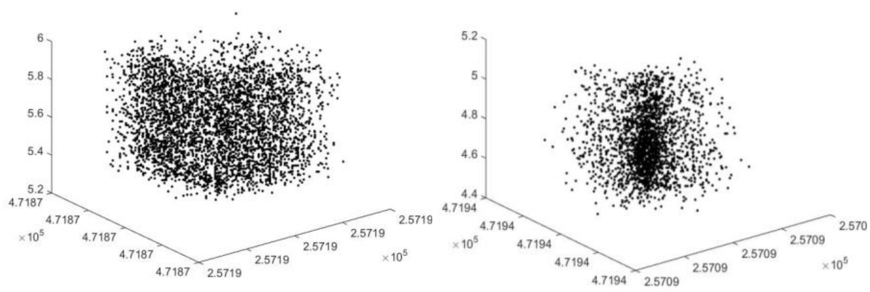
3. Sphericity of neighboring center points around the sampling mode

PCA (principle component analysis) is a statistical procedure that reduces the dimension and analysis the covariance structure of a dataset. It is the description of which direction or where the most the data is distributed. The principle components are found by calculating the eigenvalues and eigenvectors in pairs (Ringnér, 2008). The eigenvector with the largest eigenvalue along each direction represent the largest variation.

Sphericity is used to describe how jet-like an object shape is (Hanson et al., 1975). Sphericity is an eigenvalue-based feature, for $\lambda_1 > \lambda_2 > \lambda_3$, it is denoted as in (Weinmann, Jutzi, & Mallet, 2014):

$$sphericity = \frac{\lambda_3}{\lambda_1} \tag{3-7}$$

Different local maxima of object instances have different sphericity property for the neighboring points around its density mode, although they are in general all in spherical shapes. We can distinguish the correct object local maxima from those wrong candidates use sphericity constraint threshold. For example, lampposts can be discriminated from most of the trees, because normally correct candidates have more compact and denser maximum area. So within same neighboring radius, there will be a more linear-like distribution of a lamppost than that of a tree. Therefore, the sphericity value of sampling points of trees is bigger than that of lampposts. The figure 3-10 shows the sphericity difference between selected center points around the detected peak of a tree and selected points around the detected peak of a lamppost.



(a) Sephericity=0.543 (b) Sephericity=0.295

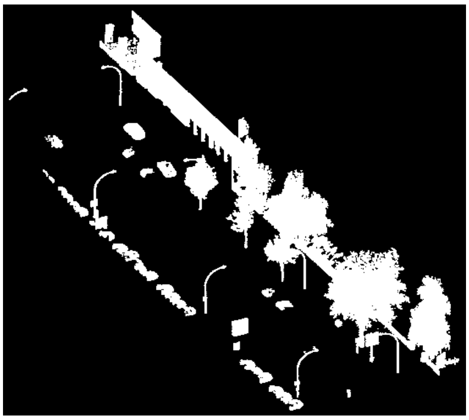
Figure 3-10: (a) Selected center points of a tree. (b) Selected center points of a lamppost.

4. IMPLEMENTATION AND RESULTS

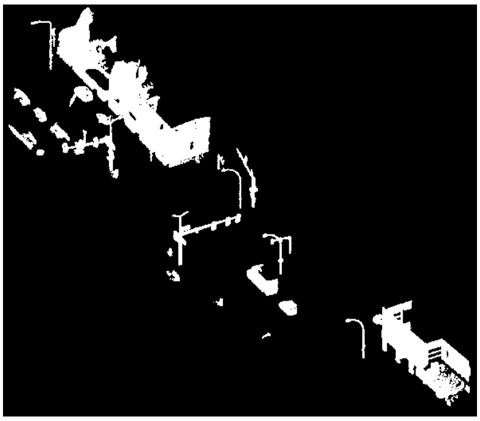
4.1. Study area and the dataset

The MLS dataset used in this research is obtained by TopScan GmbH in December, 2008 using Optech's Lynx Mobile Mapper system. The study area is located in Enschede, east of the Netherlands.

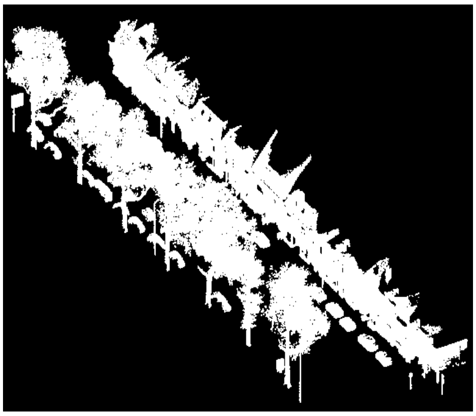
The raw dataset contains large amount of irrelevant data that add burden to the execution, thus simplification of the point cloud data is essential. The dataset used in this research was already been filtered out large horizontal planes: the ground. For computing convenience, the dataset is divided into six parts of approximately 100 meters along the trajectory (see figure 4-1).



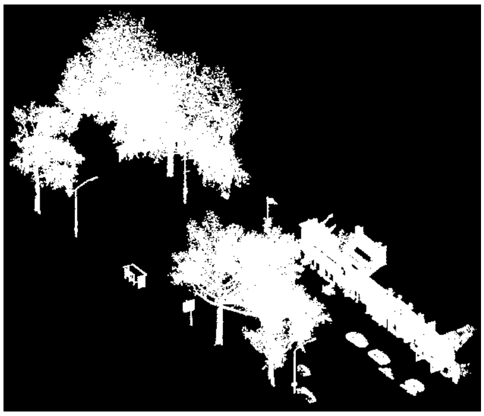
(a) Dataset: part1.



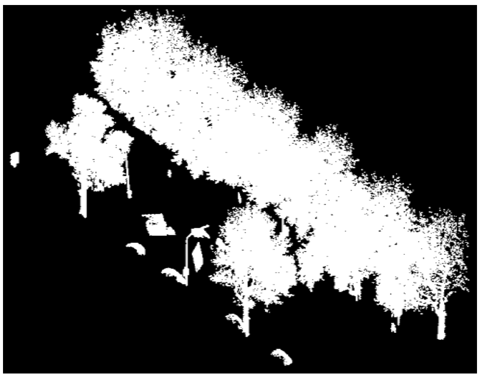
(b) Dataset: part2.



(c) Dataset: part3.



(d) Dataset: part4.



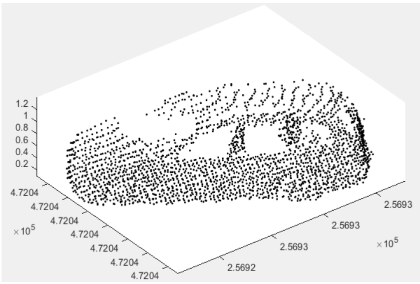
(e) Dataset: part5.

(f) Dataset: part6.

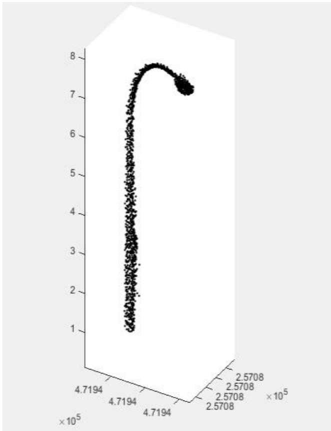
Figure 4-1: Dataset in six parts.

4.2. Model gengeration

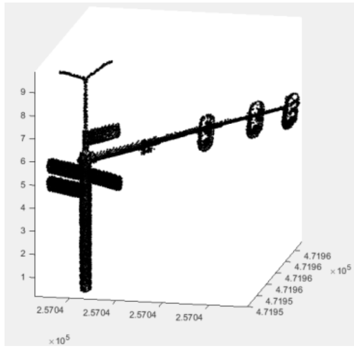
As the method uses a model-based approach, instance of each object type need to be extracted from the point cloud first, and stored for further model representation process. Model instances are extracted from the dataset using PCM, this step is done by simply cut out the model from the dataset and saved into a separate file. While selecting the model, the rule is to select an object as well-shaped as possible, so that the model can provide sufficient and accurate information for the later sampling and recognition stage. The reference point in each model is chosen as the center point of the model. Totally, 6 road furniture models were extracted from the point cloud as in figure 4-2.



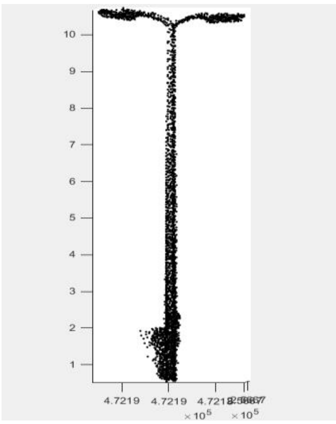
(a) Car model.



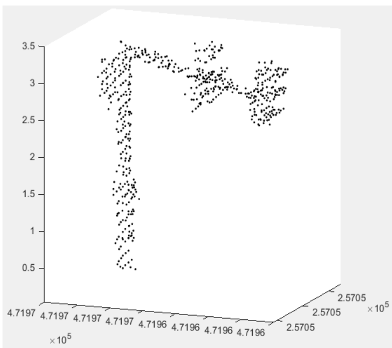
(b) Lamppost model 1.



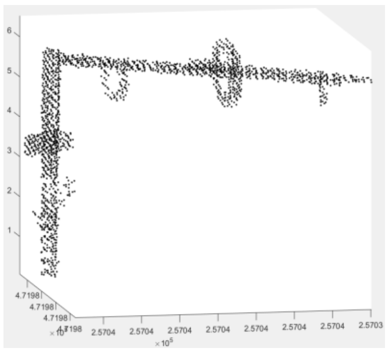
(c) Taffic light model 1



(d) Lamppost model 2.



(e) Traffic light model 2.



(f) Traffic light model 3.

Figure 4-2: Extract object models.

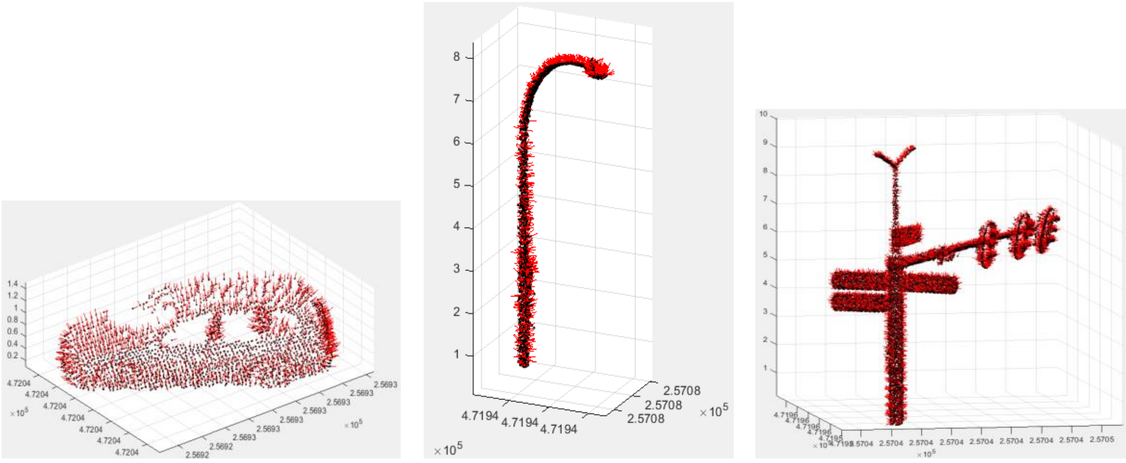
4.3. Center based voting

4.3.1. Surface normal vectors and R table

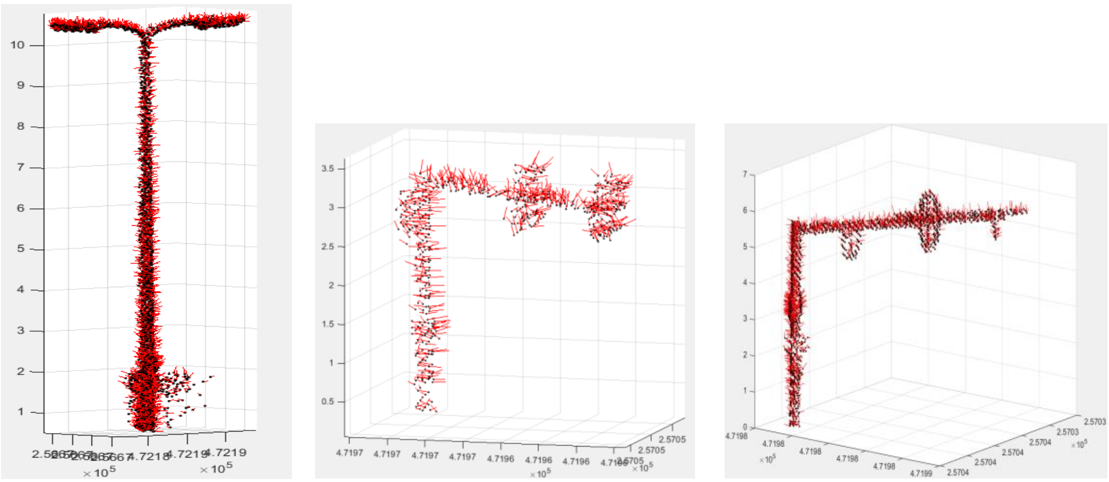
3D GHT method uses **r** vectors to describe the displacement information between the center point (as the reference point) and edge points of the object. The **r** vector is defined by two orientation angle of the normal vector as well as the length r. The normal vector can be obtained by triangulating the surface of the object or fitting planar surface to a small set of points within the neighborhood (Khoshelham, 2007) . In this research, normal vector is calculated in CloudCompare using triangulation method. In CloudCompare, the point cloud data is structured with Octree, and default orientation is around z axis. The resulting file contains the original x, y, z coordinates as well as the components of the normal vector, namely n_x, n_y, n_z. Thus the two angles of the normal can be calculated as:

$$\begin{cases} \alpha = \arccos(n_z) \\ \beta = \arccos\left(\frac{n_x}{\sqrt{{n_x}^2 + {n_y}^2}}\right) \end{cases} \tag{4-1}$$

For each point in the point cloud, the normal vector indexed the corresponding location in the 2D R table. Having obtained the two angles and coordinates of center point of the model, we can recalculate the object points according to formula 3-2. The 3D GHT can represent the instance model correctly and accurately (illustrated in figure 4-3, normal vectors in red).



(a) Car model with normal vectors. (b) Lamppost model 1 with normal vectors. (c) Traffic light model with normal vectors.



(d) Lamppost model 2 with normal vectors. (e) Traffic light model 2 with normal vectors. (f) Traffic light model 3 with normal vectors.

Figure 4-3: Object models reconstructed from R-table with the normal vectors.

4.3.2. Local maxima property of the object model

According to the voting schema, all the points in the point cloud vote for their possible centers and we can obtain a density based distribution of center data which contains a large number of local maxima candidates. Some peaks are easy to differentiate according to their different properties, while others are hard to recognize and distinguish because they are similar with the correct maxima, especially for whose points that in the same shape with the object model. In order to select the correct local maxima from the generated center points, the information about property of the local maxima need to be extracted from the model, and by specifying the properties, correct local maxima can be identified from all the potential candidates.

Table 1: Information generated from the model.

Local maxima	Covariance	Mean height (m)	Sphericity
Lamppost 1	[0.30, 0.70, 4.00]	4.75	0.242
Lamppost 2	[0.06, 0.01, 4.00]	5.20	0.423
Car	[0.70, 0.30, 0.05]	0.72	0.257
Traffic light 1	[5.10, 10.50, 2.30]	5.15	0.473
Traffic light 2	[0.66, 1.43, 0.75]	2.50	0.285
Traffic light 3	[3.60, 4.20, 2.30]	4.50	0.373

Table 1 shows the local maximum information generated from the model. The information guides the choice on the threshold range of the parameters for restricting properties of the local maxima. The covariance value are used as the sigma of the proposal distribution, ensuring that the Markov chain can take sufficient big steps to converge from a far point to the correct center point; the height value is the mean value of object objects since we take the mode point of the sampling results as the reference point; sphericity value describes the compactness and distribution variation of the neighboring points around the sampling mode. The information value is obtained by applying the center based voting process to the model itself and estimating the local maximum of the voted center points. For 10 times acceptable estimation we calculate its mean value of the three property information and take it as the reference information as shown in table 1.

4.4. Local density detection

4.4.1. Metropolis Hastings algorithm

In the detection stage, MH algorithm is applied to find peaks from the voted center points. MH algorithm performs a Markov chain to the density distributed center points and evaluates the probability at a given point from the target distribution. The function to evaluate the probability is shown in formula 4-2. For a given point (e.g. the current chain position), we calculate its number (m) of neighboring points. We also define the constant parameter k.

$$p = e^{(-k^2/m^2)} \tag{4-2}$$

p is the function to evaluate (calculate the probability at a given point) the target distribution composed of the generated center points, k is a constant specifying at least how many points within a neighboring radius around the correct mode, and m is the number of neighboring points at the given point. If the chain is at a very dense point then m is large and the probability is close to 1. If the chain is at a low density point then n is small and the probability is close to 0, as illustrated in figure 4-4.

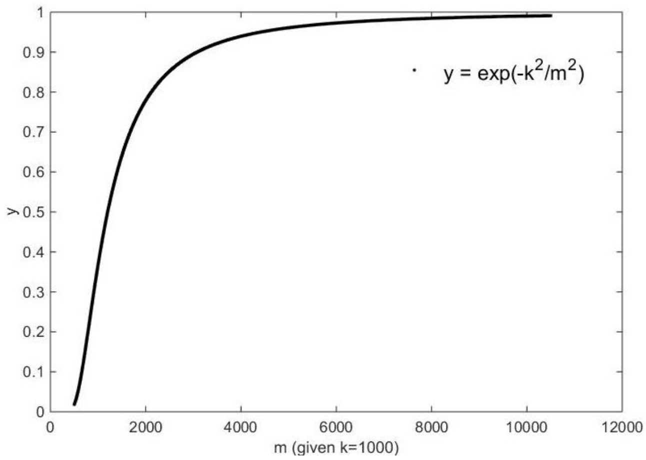


Figure 4-4: Probability function for evaluating the target distribution.

The outcome of the MH algorithm gives a set of sampling points; the mode (peak) is the mean value of the sampling points. Around the mode, we select a set of neighboring point as the local maximum area and further identify them with the three property constraints.

The neighboring area is defined as a cube, for the purpose of fast computing process; the radius of the cube is 0.3m, for the purpose that it fits the smallest local maxima, e.g. a lamppost (see figure 3-8). The constant parameter k is different according to different input model, and it is determined based on pre-analyses and well-tried experiments on the correct local maxima of the object model.

4.4.2. Recognition

Having obtain the candidate local maxima, according to information generated from the object model, the property constrains of the local maxima is adopted. Considering the variations of the sampling results, the parameter is not a threshold value; instead, it is a certain value range in order to identify as more candidates as possible.

The θ angle spacing is specified according to the pose of existed objects (through a rough inspection) in the point cloud dataset. The number (of neighboring points) range is decided by collecting 10 acceptable candidates with MH algorithm on a simple test dataset (containing the target objects), and taking the maximum and minimum value as the range. The sphericity value range is also determined by trying the MH algorithm on the test dataset and optimizes it according to the information in table 1. The sigma value and mean height value range is decided by borrowing the covariance information from table 1. But for some objects like lamppost, they have a big covariance value of z, which influence the moving range of the chain as discussed in section 3.6.2, so the strategy is to reduce the factor of order of the sigma value in table 1. The recognition process ends when no more satisfied candidates can be found.

Table 2: Parameters defined for each kind of object instance.

Input models	Radius and k	Sigma of proposal distribution	Number of neighboring center points	Sphericity of neighboring center points	Mean height of neighboring center points (m)	θ spacing
Car_1	[0.3, 1000]	[0.70, 0.30, 0.05]	$3000 \leq n \leq 20000$	$0.15 \leq s \leq 0.55$	$0 \leq h \leq 1.3$	90
Car_2	[0.3, 800]	[0.70, 0.30, 0.05]	$400 \leq n \leq 4000$	$0.40 \leq s \leq 0.80$	$0 \leq h \leq 1.3$	90
Traffic light 1	[0.3, 7000]	[0.05, 0.10, 0.02]	$6000 \leq n \leq 30000$	$0.35 \leq s \leq 0.85$	$4.5 \leq h \leq 6.5$	180
Traffic light 2	[0.3, 1000]	[0.66, 1.43, 0.75]	$200 \leq n \leq 800$	$0.20 \leq s \leq 0.50$	$2.0 \leq h \leq 3.0$	90
Traffic light 3	[0.3, 1000]	[0.35, 0.42, 0.23]	$1000 \leq n \leq 3000$	$0.25 \leq s \leq 0.45$	$3.5 \leq h \leq 5.5$	0
Lamppost 1	[0.3, 1000]	[0.03, 0.07, 0.40]	$500 \leq n \leq 3000$	$0.20 \leq s \leq 0.40$	$4.0 \leq h \leq 7.0$	180
Lamppost 2	[0.3, 1200]	[0.006, 0.001, 0.40]	$1000 \leq n \leq 5000$	$0.30 \leq s \leq 0.60$	$4.5 \leq h \leq 7.5$	180

** Car_2 is cars that are badly shaped because of the scanning angle of MLS.*

4.5. Recognition results

Recognized objects are highlighted in each of the 6 dataset from figure 4-5 to figure 4-10.

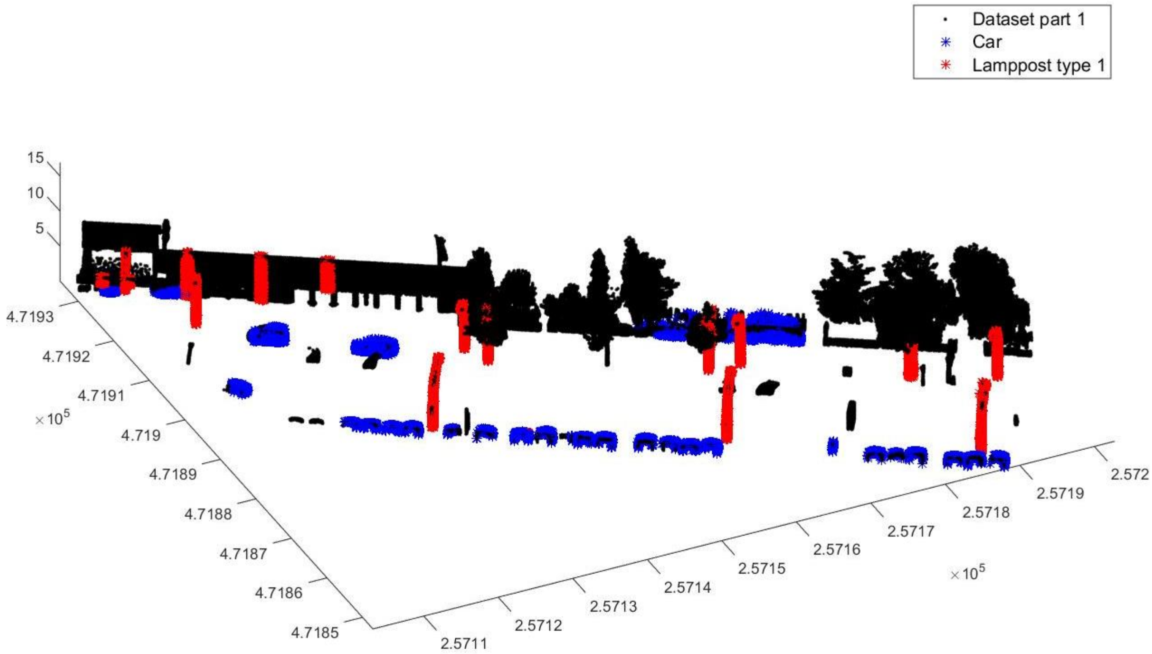


Figure 4-5: Recognition results of dataset part 1.

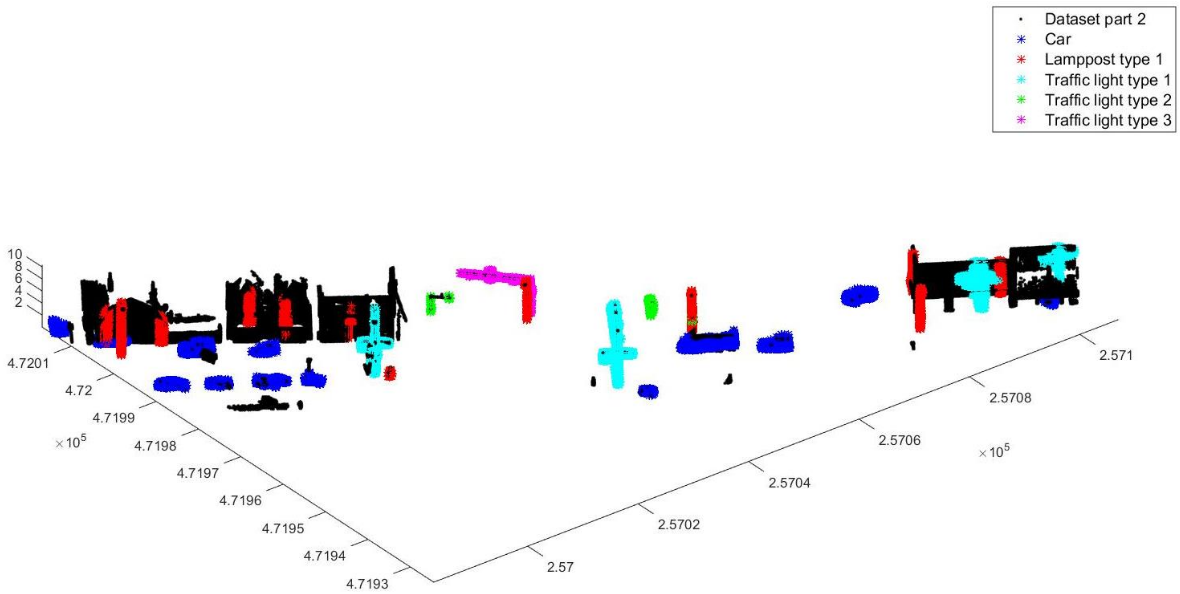


Figure 4-6: Recognition results of dataset part 2.

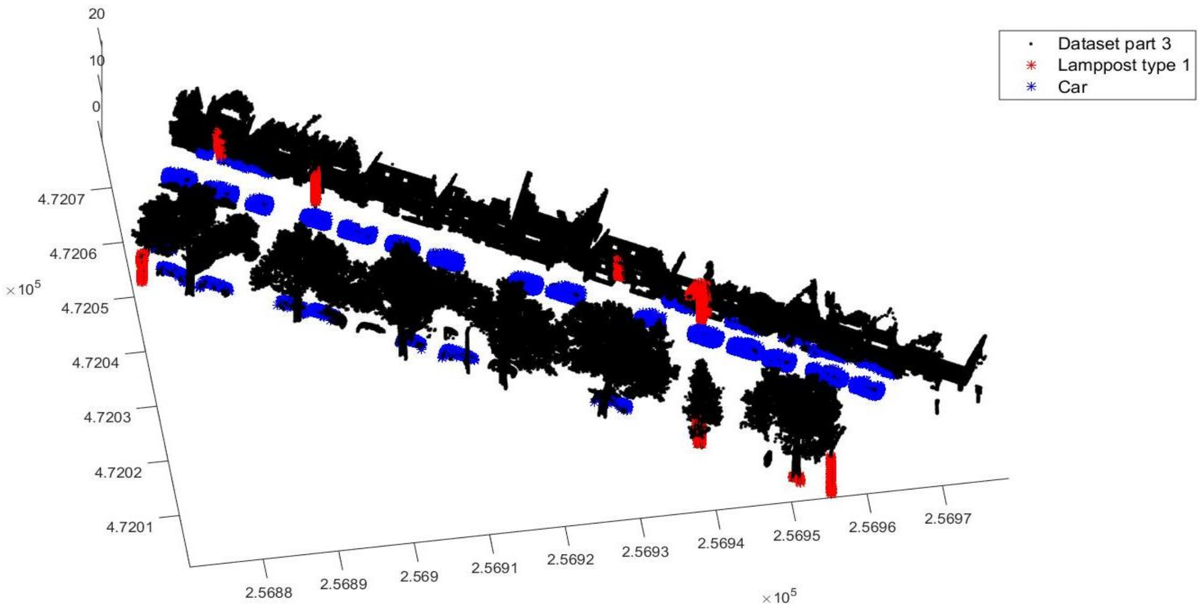


Figure 4-7: Recognition results of dataset part 3.

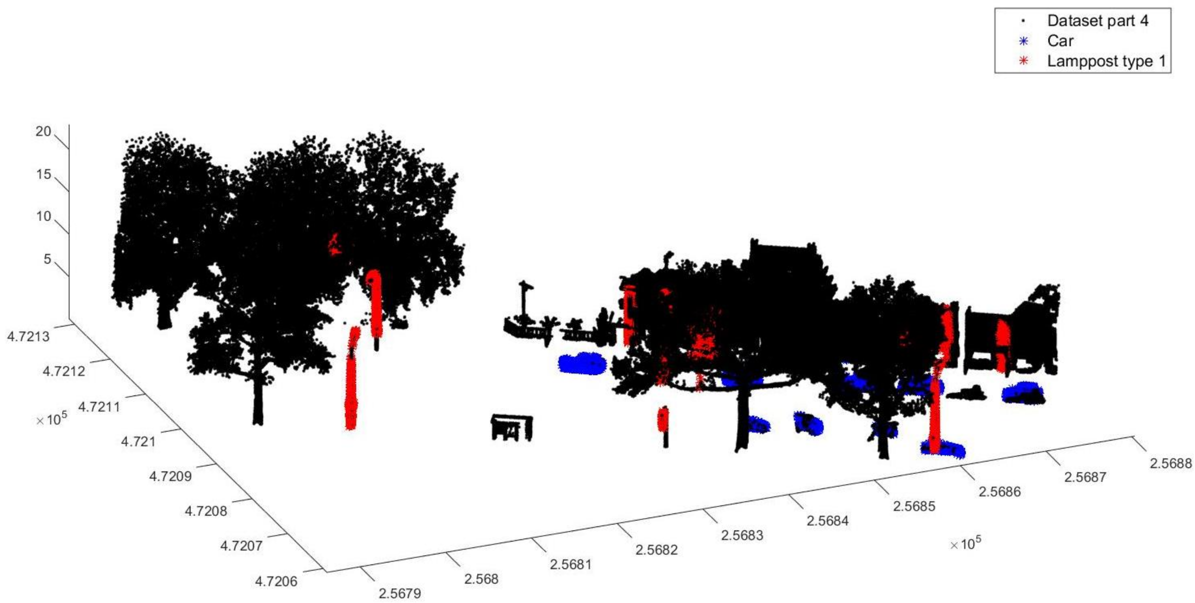


Figure 4-8: Recognition results of dataset part 4.

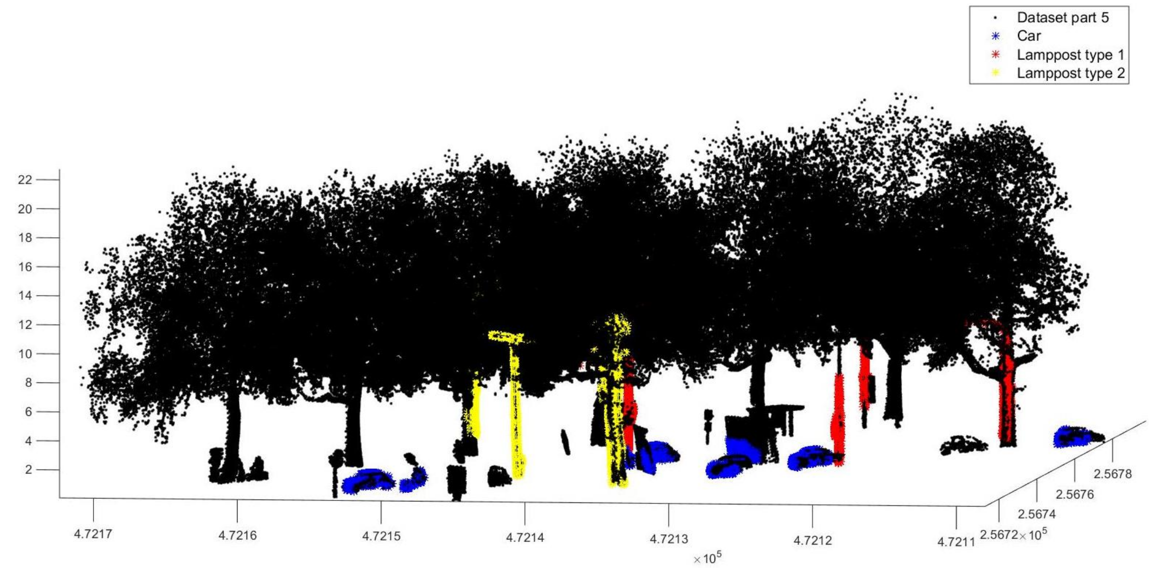


Figure 4-9: Recognition results of dataset part 5.

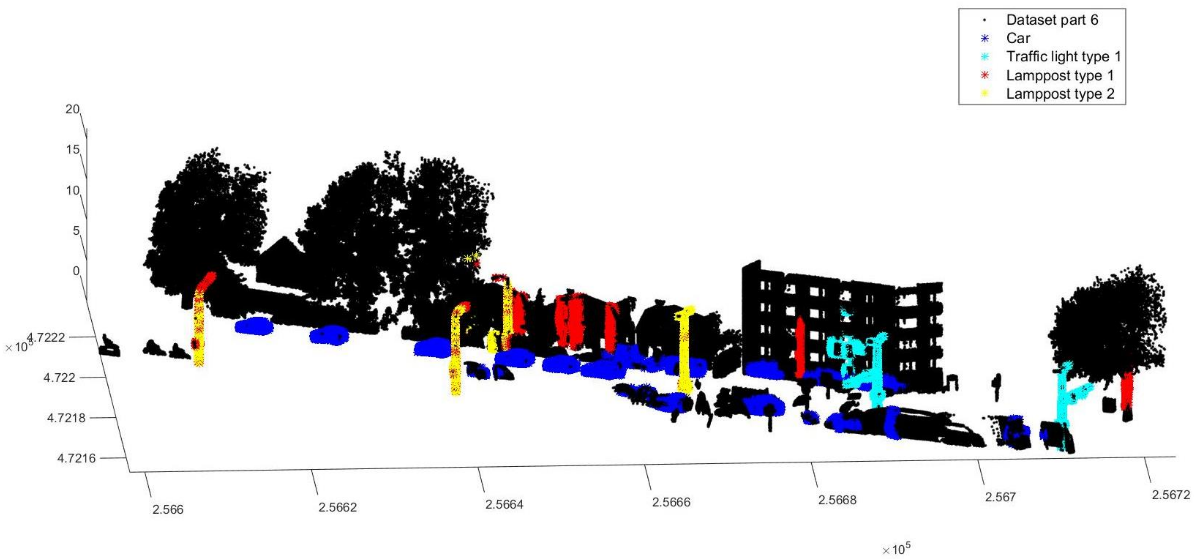


Figure 4-10: Recognition results of dataset part 6.

5. EVALUATION AND DISCUSSION

5.1. Evaluation of the results

5.1.1. Precision, recall and overall accuracy

The evaluation of the results is implemented at the instance level. The reference dataset has already been labelled. The visual check of the reference dataset to evaluate the recognition results and the photograph of the ground truth are also important. The accuracy is defined in terms with precision (positive predictive value) and recall (sensitivity). The True Positives (TP), False Positives (FP), True Negatives (TN) and False negatives (FN) are defined in table 3 (Olson & Dursun Delen, 2008):

Table 3: Descriptions of the confusion matrix.

	Reference positive	Reference negative
Results positive	TP: The number of objects found both in the recognized results and the reference data (correct).	FP: The recognized instances are not the expected objects in the reference data (unexpected).
Results negative	FN: The number of objects in the reference data but not in the recognized results (missed).	

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP}{TP + FP + FN}$$

Precision and recall are also considered as the correctness and completeness of results. Precision is the fraction of the retrieved instances that are correctly relevant to objects in the reference data. Recall is the fraction of relevant instances that are retrieved by the method. Accuracy provides the overall quality of the results in recognizing corresponding objects from the reference data.

Table 4: P-R evaluation and the overall accuracy.

Objects	TP	FP	FN	Precision	Recall	Accuracy
Car	96	20	28	82.76%	77.42%	62.34%
Lamppost 1	24	38	3	38.71%	88.89%	36.92%
Lamppost 2	2	8	0	20%	100%	20%
Traffic light 1	4	3	0	57.14%	100%	57.14%
Traffic light 2	2	2	0	50%	100%	50%
Traffic light 3	1	0	0	100%	100%	100%

The P-R evaluation does not contain TN. TN values does not exist in the error analysis of the recognized results because in the recognition method, the accuracy gives the measure of how good or bad the detected quality is, and there is only one situation of negative results, which is the missing objects that are not been found by the method.

The evaluation of the recognized results is also carried out using the confusion matrix.

Table 5: Confusion matrix of the recognized results.

Objects	Car	Lamppost 1	Lamppost 2	Traffic light 1	Traffic light 2	Traffic light 3	Others	Recognized
Car	96	0	0	0	0	0	20	116
Lamppost 1	0	24	1	0	0	0	37	63
Lamppost 2	0	2	2	0	0	0	6	10
Traffic light 1	0	0	0	4	0	0	3	7
Traffic light 2	0	0	0	0	2	0	2	4
Traffic light 3	0	0	0	0	0	1	0	1
Missed	28	3	0	0	0	0		
Total	124	27	3	4	2	1		
Accuracy	77.42%	88.89%	66.67%	100%	100%	100%		

Table 4 shows that the method recognized not only the correct instances but also detected many false positives (the similar shapes). Table 5 shows that there are overlapping recognitions between lamppost type 1 and lamppost type 2.

5.1.2. Computational complexity

In this research the dominant computational burden is the peak seeking process, since Markov chain runs in parallel in order to find all the satisfied candidates.

In the 3D GHT method, the computational complexity of casting votes to the accumulator bins is a main problem to tackle with, when the voting space is expanded to 7 dimensions. Thus, cost-reduction strategy such as randomized, hierarchical or adaptive voting schemas are recommended to refine the execution process of the method (Khoshelham, 2007). In this modified method, the discrete voting process to find peaks is replaced by the MH algorithm. In implementation, the computational complexity of voting process in the continuous Hough space is $O(M * N)$, where M is the number of points of the input point cloud dataset is and N is θ angle intervals along z axis. In the MH algorithm, for each iteration time, the target distribution is evaluated of the local density at the given point, so the computational cost of sampling process is $O(M' * i)$, where M' is the number of the generated center points (target distribution) and i is the sampling steps of the MH algorithm. In addition, the efficiency improvement strategies are carried out as follows in two aspects:

1. Parallel computing

Another computational reduction strategy on execution time is to make use of the parallel running. In Matlab, parallel computing can improve the efficiency by 1.5 times in general, and the more input the points, the more obvious the efficiency improvement can be seen. For each input dataset, the method uses 4 cores for computing and runs 3 Markov chains at the same time. Subsequently, the wrong candidates and the corresponding center points are removed from the computing process for the next time to prevent duplicate detection. Thus, as the loop time increases, the number of input center points will decrease, and thus the execution time for each loop will decrease.

2. Partitioning the point cloud dataset

   The partitioning the raw point cloud dataset into overlapping subparts ensures that during the execution process, there will not come across memory overflow problems. However, it may not be a problem because the occurrence of memory overflow mainly depends on the hardware and software in use.

5.2. Discussions

5.2.1. Discussions of the 3D GHT method

1. The method cannot work well on objects whose normal vectors are not well distributed, because it makes the local maxima scattered instead of compactly distributed. For example, a road sign in figure 5-1, the normals are not well defined because the point spacing relative to the object surface area is not small enough. Here the object surface area (the pole part) is very small so the normals are very noisy. This leads to a bad center points and the peak of which are hard to be detect by the MH algorithm.

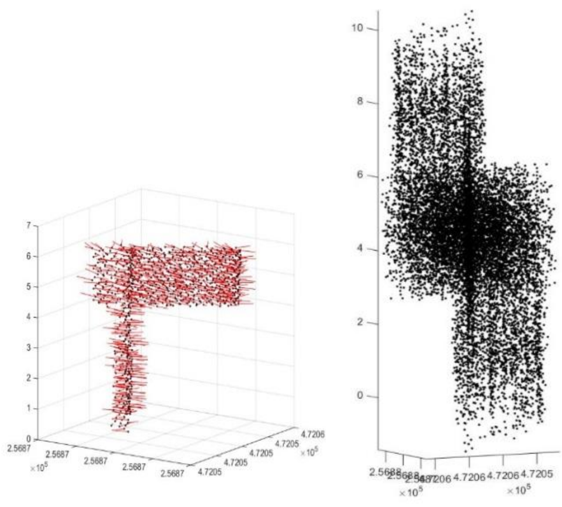


Figure 5-1: Normal vectors and the voted center points of a road sign.

2. The method uses a set of mathematical parameters to define an explicit object. The method is very robust to occlusions as shown in figure 5-2: two cars are blocked of some parts by the motorcycles and the method successfully recognized and separated them; the method also successfully discovered cars that lie in the car garage under the building. However in MLS data, most of the roadside objects cannot be integrally scanned and deformation is inevitable, e.g. cars that are scanned from the front are badly detected by the method.

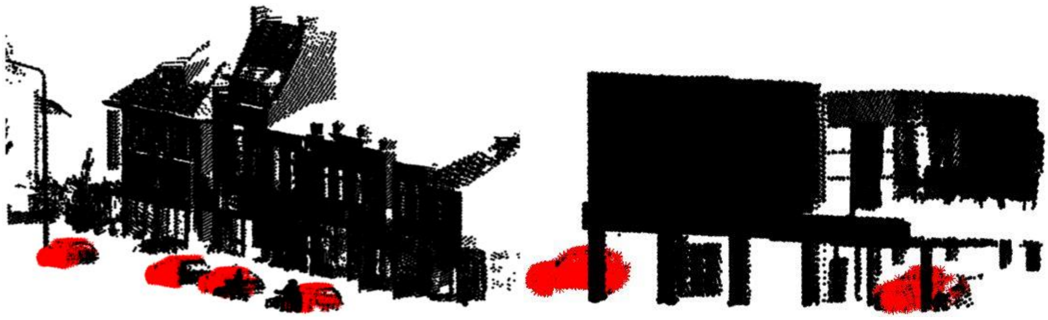


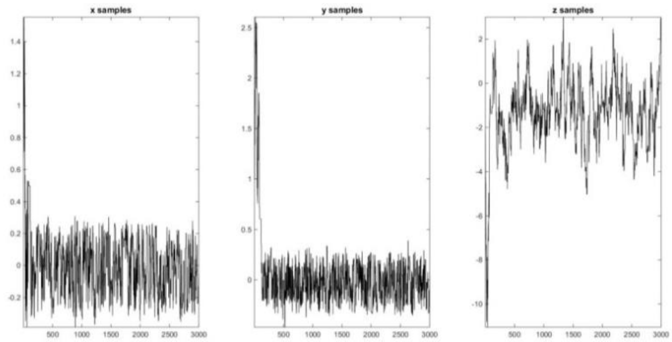
Figure 5-2: The robustness to occlusions.

3. The 2D R table is denoted by the two orientation angles, thus the bin size of the angle spacing needs to be decided. Small angle spacing leads to more **r** vectors in a bin and vice versa. Thus the number of generated center points is different. However, a rough design of the R table (but within the certain range) doesn't influence a lot on the general distinguished property of each object type. In implementation, the angle bin is recommended to be between 10 to 20 degrees.

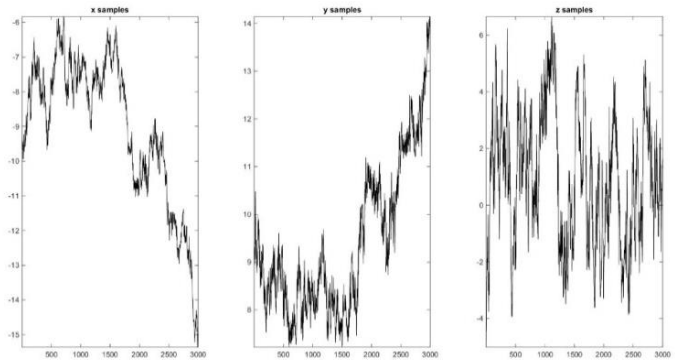
5.2.2. Discussions of the Metropolis Hasting parameters

1. The performance of finding the correct peak in a density distribution depends on the selection of parameters of MH algorithm and the threshold range of the property for the candidates. In this method we choose mvnpdf as the proposal distribution because it is appropriate and the simplest. A proper setting on the A-R criteria also guarantees good quality of candidates. Some properties of the MH algorithm is remarked in (Andrieu et al., 2003) and (Chib & Greenberg, 2012): Firstly, the MH algorithm requires a careful design of the proposal distribution. Secondly, the normalising of the target distribution is not required. Thirdly, although the pseudo-code uses only one Markov chain, several chains can be applied independently in parallel.

2. The idea of MCMC sampling is that the approximation of target distribution process will come to convergence in the end regardless of initial starting point. Sufficient iteration times make sure that the estimation is unbiased. In this research, iteration times around 3000 – 4000 is enough. Small iteration times cause under-estimation and biased results (sampling process stops before the convergence), while large iteration times cause duplications and make the process inefficient. Good convergence process is shown as in figure 5-3(a), the process starts with a random point, and all the sampling points are systematically distributed around the mode point. For the bad convergence process, sampling points jump across the invariant distribution as shown in figure 5-3(b), thus the detected local maxima is incorrect.



(a)



(b)

Figure 5-3: Good (a) and bad (b) convergence process.

3. As for the efficiency of the MH algorithm, the computational complexity of MH algorithm in high dimensions is discussed in the research of Beskos & Stuart (2008). Given target distribution $p(\mathrm{x})$ in dimension space d, the computational complexity is O(d), which lies at the number of steps to run the Markov chain to explore the $p^d(x)$. In general, the computational complexity of MH algorithm depends on the optimization of proposal parameters, the probability. It is possible for the number of MCMC sampling steps to achieve O(1) given the developed random walk of Markov chain in (Pillai, Stuart, & Thiery, 2011).

5.2.3. Discussions of the local maxima property

1. The three properties of local maxima for each object type are decided from three aspects: shape, location and quantity, respectively the sphericity, mean height value and number of neighboring points around the sampling mode. The determination of the threshold range is important to the performance of the method. In this research, the priority is set as sphericity < height < number. Number is considered as the most prioritized constraint because it ensures not missing any possible correct candidates. This can be regarded as the process of generating hypothesis. Subsequently, the mean height threshold makes sure that wrongly located candidates are eliminated. Sphericity is the least prioritized criterion; it helps separate the rest wrong candidates from the correct ones, but still cannot distinguish similar local maxima.
2. During implementation, bias of the local maxima is inevitable. If the detected local maximum is biased, then not all points (voters) of the object can be detected. In another word, the method only catches the voters that cast votes for the local maximum area. However, a small bias isn't against the objective of the research to recognize and detect object from the point cloud, because the detected points still represent the main part of the object and as the detected object is highlighted, it is easy to recognize the object shape form the dataset.

3. The method is less effective to distinguish different types of objects when their local maxima properties are very similar. The method cannot separate them although two different models are used as input, the recognition results will be highly overlapped. For objects like big vehicles (e.g. trucks in figure 5-4), they are bigger sized than small cars and they only have small difference of mean height difference with respect to the local maxima, so they are difficult to be separated from small vehicles when the constraint mainly depends on the height difference. There are also problems to distinguish static cars from the moving cars as in figure 5-5, because the moving cars have strongly varied center location.

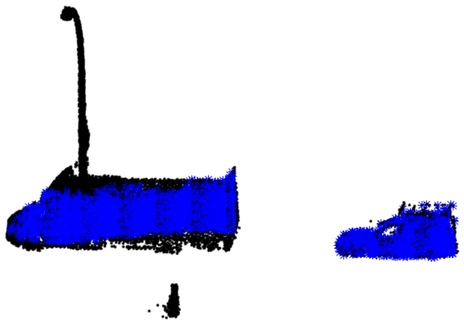


Figure 5-4: Vehicles of different sizes.

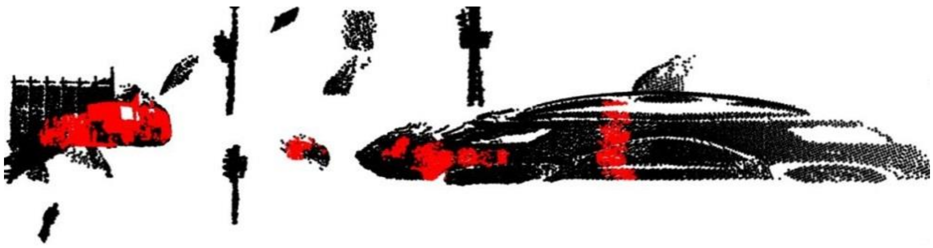


Figure 5-5: Problem situations of recognizing different car types.

For two kinds of lamppost object that only have slight difference in height and the upper part (see figure 5-6), the method cannot separate them because their local maxima properties are highly overlapped. So the method regards these two kinds of instances as one lamppost type. However, due to the slight difference still exist, the occurrence of biased local maxima increase and there can overlapping detection (one object being detected by two input models).

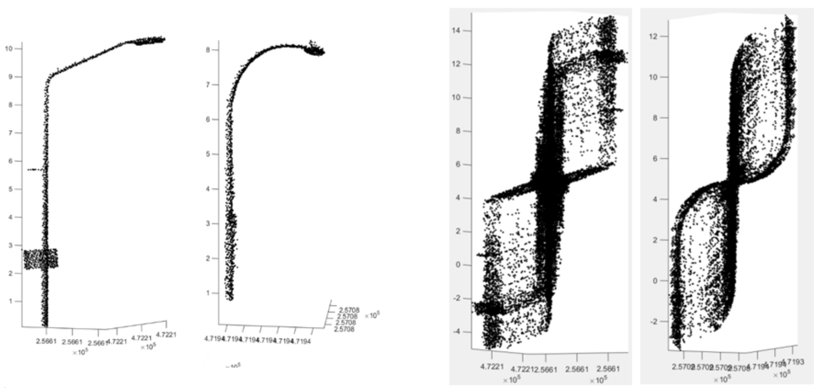


Figure 5-6: Three instances of lamppost with similar shapes and local maxima property.

5.2.4. Discussions of the recognition results

1. The method is based on center based voting schema to find peaks of the voters, and all candidates that satisfy the criteria are regarded as the correct local maxima. Thus the results of

the method not only give correct objects but also recognize points from other objects that have similar shapes with the input model. For example in figure 5-7, it is inevitable that thin trees are recognized as lamppost, and some parts of the wall influences the results of car recognition.

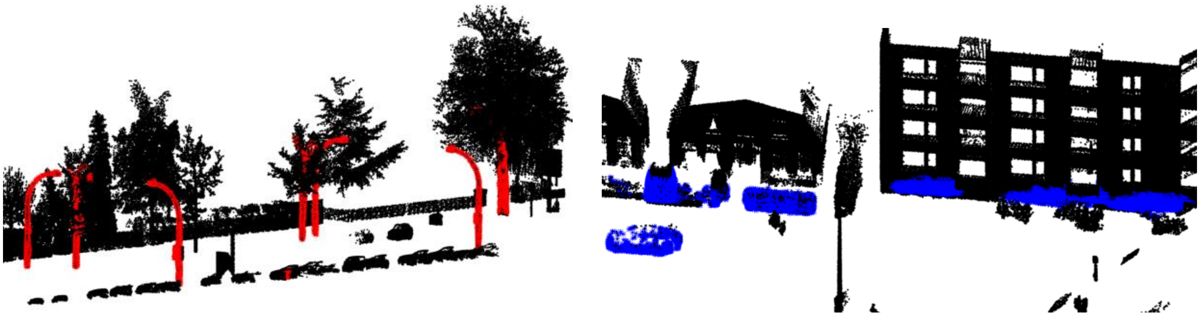
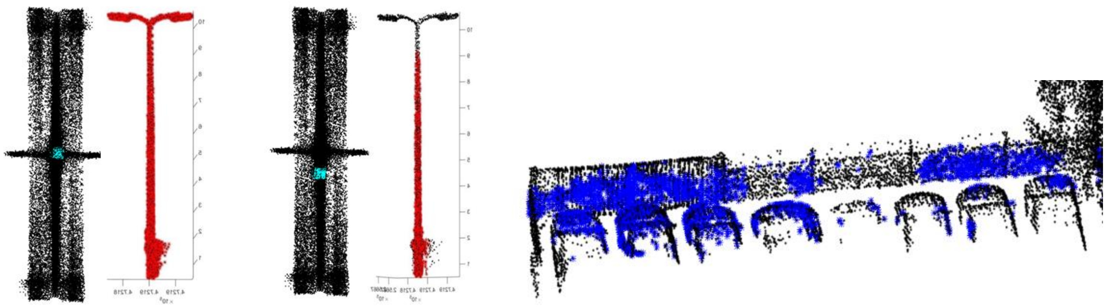


Figure 5-7: Recognized true positives and false positives.

2. If the detected local maxima are biased, the corresponding detected voters are inaccurate. This leads to a bad recognized shape of the object. Another situation is that a biased local maxima detection result not due to bad performance of the algorithm, but because of the bad shape of the objects in point clouds (e.g. in figure 5-8).



(a) (b)

Figure 5-8: (a) Badly detected objects because of biased detected local maxima. (b) Badly shaped objects in point clouds.

3. In two circumstances the method fails to find the target object. In the first situation, the method cannot work well on object that is very badly shaped like figure 5-9. For object that has too few points, for example, the road sign, the spatial property of its local maxima is hard to define and distinguished from others; Secondly, for objects that are badly scanned as shown in figure 5-8(b), its local maxima information is not integral thus cannot be recognized by the method.

Figure 5-9: Badly shaped object.

4. From table 4 we can see that most objects have been correctly recognized by the method and only few target objects are missed. The method generates few wrong results (true negatives) in general, which means the method can successfully separate different object categories. However, the drawback is that the method generates many unexpected objects (false positives), because they

share similar shapes with the input object model. False positives of car mainly come from building walls, false positives of lampposts mainly come from thin trees and building walls. True negatives of lampposts are mainly road signs. False negatives of cars are high because that some of car objects are not well scanned thus the method missed them. Some extreme high or low accuracy results remain discussable: firstly of all, there are too few object instances exist in the dataset, so although the method managed to recognize all the matched objects, it is still hard to judge whether it can be effective enough if more instances exist; secondly, to save computational cost, the method is only applied to dataset part 2 to detect traffic lights. In table 5, there are overlapping recognition results between lamppost type 1 and lamppost type 2 because their local maxima property is difficult to be completely distinguished.

5. Some models like traffic lights and lamppost type 2 are not implemented on the whole dataset; they are only implemented in their existing sub-datasets because of their limited number in the whole dataset. Because of too few quantities of these objects in the dataset, although the method has successfully detected all the objects from the dataset, the number of false positives is more than the true positives.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1. Conclusions

In conclusion, the method is proved to be feasible and effective to achieve the research objectives proposed at the beginning. The developed pipeline of the method can be concluded into three parts. Firstly, select appropriate shape descriptors to the object and generate center points for all points; secondly, apply peak detection method to identify local maxima candidates; thirdly, distinguish the correct peaks from all the possible candidates and relate them to the voters.

1. The developed method has improved efficiency compared with the 3D GHT method. 3D GHT is difficult to be implemented due to its high computational cost, this research successfully implement the modified method and achieved fine results. The method turns the discrete Hough voting into a continuous space (not all the parameters are in the continuous space, the θ angle spacing is in the discrete space), and is capable of finding multiple local maxima per peak detection process. Thus the computational complexity is greatly reduced. Efficiency also been improved by parallel computing and dividing the dataset into smaller subparts.
2. The method is robust to occlusion in MLS data. Given a proper instance model, the method can recognize and detect objects that in same shape with the input model regardless of partial occlusion. However, deformation problem cannot be fully solved since the method is limited by density variation of objects in point cloud data. Since the method follows the center based voting schema, it cannot perform well on objects with badly scanned shapes or objects with strongly varying centers.
3. The method does not require large training data as input, for each object type, only a good explicit model is required. The recognition process is automatic and the program stops until no more satisfied candidates exist. However, the method still requires manually extraction of object models from the data and the local maxima properties need to be specified beforehand for each type of object.
4. The method fails to distinguish objects when they have very similar local maxima properties, and the recognition process is also influenced by false candidates when they are more dominant than the correct one within a searching area. The method is less dependent on segmentation, but more dependent on input parameters of the parameters.
5. The dataset in use contains few object types, but in reality a huge dataset may contain a large number of different instance types. In that case, for each object type a model is needed to be extracted from the dataset, and specification of the local maxima property of each model will make the work more complex and less efficient.
6. In general, the recognition results are influenced by large building walls and thin trees. The recognition results of car are influenced by building walls; the recognition results of lampposts are be influenced by thin trees and building facades; the recognition results of traffic lights are also influenced by parts of the building. Because of few numbers of traffic lights in the dataset, the recognition process is relatively slow since the method keeps rejecting candidates until it finds the correct local maxima.
7. Recall results of all the objects in the dataset are high in general, which means the method reaches high completeness. Precision result is because of many false positives. There are also overlapping detections between objects that with similar shapes, e.g. lamppost type 1 and lamppost type 2. In total, 129 out of 160 of the target objects have been successfully detected.

6.2. Answers to the research questions

1. Among the existing object detection methods in 3D point cloud, which algorithms can be suitably applied or extended to recognize roadside instances in MLS data?

Model based approaches use CAD models, normal vectors in 3D GHT method (Khoshelham, 2007) and template set to represent an object and they are also robust to partial occlusion problems. Other existing descriptors to achieve this goad are spin images used in ISM method (Velizhev et al., 2012) and shape descriptors like 3D SURF (Knopp et al., 2010) etc. The research proposed a model based approach for object recognition in point clouds. By model based approach, it means the method uses an explicit object model as input and extracts its global property of a whole object instead of using local property of the data point. The justification on 3D GHT has been given in section 2.6 and section 3.1.

2. What are the limitations that influence the performance of the selected method, and how to optimize it or reduce the drawbacks on the selected method?

   The selected 3D GHT method has the drawback that it requires large computational cost, yet it has not been taken into implementation in the real 3D point cloud dataset. Thus the research reduces the computational cost by replacing peak detection phase with a continuous peak searching method. The efficiency also been improved by partitioning the dataset and by using parallel computing.

3. How to obtain and store the object models on condition that a model-based approach is used in this research?

   The object models are extracted manually from the dataset. The store of the object models is by constructing an R table. Generally speaking, the advantage is that manually pick a well-shaped and scanned object model ensures that the local maxima property is sufficient and accurate, and one can decide which object model is most the suitable depends on the quality of the obtained point cloud dataset. Because it is an explicit representation which allows exact reconstruction of the model from the R-table. Other model representations, e.g. spin images, are not explicit but implicit.

4. How well or badly does the method perform in the context of object instances detection in MLS data?

   The method is considered feasible and efficient; it reduces computational complexity compared with the original 3D GHT. The stochastic sampling method is reliable in terms of recognition of cars and lamppost type 1; however, it remains discussable of the extreme recognition results in the contest of lamppost type 2 and traffic lights type 3, a summary of the recognition performance is given in section 6.1. All in all, the method achieves the overall objective of the research; it improves the efficiency, and reaches reasonable accuracy.

5. What are the factors that influence the performance of the algorithm used in this research?

   The factors that influence the method can be considered from two aspects, the raw dataset and the method itself. Firstly, the method still has a high demand on the computational burden; a raw input point cloud dataset that contains many false hypotheses will decrease the efficiency of the method. Secondly, the method is sensitive to the parameters of MH algorithm and the defined properties of the local maxima. Thus, the performance may differ if the parameters change.

6.3. Recommendations

Three procedures are important to the method, shape representation, center based voting and peak detection. These procedures can be tried out with different approaches. In order to achieve higher accuracy results, several aspects of the method can be modified in future.

1. Dataset pre-processing

   Partitioning the point cloud into overlapping subparts is a way of reducing computational burden. However, other pre-processing of the dataset can also be applied. Although the method proposed model-based approach, which initially does not require any segmentation process, it is recommended that apply a 'hypotheses generation' process before the recognition phase as in the method by Velizhev et al. (2012). Because unnecessary part or segments will lead to large number of possible candidates and this adds many extra computational cost.

2. Model representation

   Normal vector computing is an important part as it serves as the index during the voting process. In this research the normal vectors are computed with CloudCompare tool. Other approaches to compute normal vectors can be tried out and different parameters of neighboring radius can be adopted when triangulating the object surface. Also, a complete scanned object model can help further improve the recognition accuracy, thus it is recommended to use complete object instance as model to guarantee more accurate learning information for the voting and recognition process.

3. Optimization of the MH algorithm

   In the recognition phase, or the peak seeking phase, MH algorithm has been applied together with three specified properties of the found peaks. MH algorithm is an efficient approach to detect peaks from a density distribution. It mimics arbitrary multivariate distribution based on the proposal distribution. The adopted algorithm in this research is based on the basic principle of the MH algorithm. The mvnpdf is used as the proposal function; however, more accurate and complex function can be proposed such Gaussian distribution. Besides, many modifications to the methods have been proposed since it is regarded as one of the most significant algorithms in 20^{th} century. Algorithms such as Sequential Monte Carlo, Adaptive MCMC, and Hybrid Monte Carlo algorithms etc. (Andrieu et al., 2003) are expected to reach better performance.

4. Optimization of peak detection

- The definition of the sampling mode with its neighboring area is defined as a cube for the purpose to fast the computing process in Matlab. However, it is recommended to define a sphere instead of a cube because it fits the local maxima distribution better, and by doing this, it is possible that the sphericity property can be no longer necessary. Besides, k nearest neighborhood (KNN) method can also be adopted. After obtaining the mode point, search k nearest neighboring points around the mode, thus the number of sampling points' property can be unnecessary.

- Except for using MH algorithm, other methods that with the ability to detect mode area in a density distribution are also recommended. Among which, density-based clustering algorithms such as Mean Shift (Comaniciu & Meer, 2002), and KNN method are helpful. However, the drawbacks of applying these methods are that they cause more computation burden compared with MH algorithm.

- In order to make the local maxima more distinguishable among the candidates, a probabilistic voting schema is recommended to adopt. Instead of letting all the voters cast equal votes, the method can use a probabilistic voting procedure as in (Velizhev et al., 2012), assign different weights to the votes according to the importance of the feature property.

LIST OF REFERENCE

Andrieu, C., Freitas, N. de, Doucet, A., & Jordan, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine Learning*, *50*(1-2), 5–43. doi:10.1023/A:1020281327116

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, *13*(2), 111–122. doi:10.1016/0031-3203(81)90009-1

Beskos, A., & Stuart, A. (2008). Computational Complexity of Metropolis-Hastings Methods in High Dimensions. *Monte Carlo and Quasi-Monte Carlo Mrthods*, 61–71. doi:10.1007/978-3-642-04107-5

Bevilacqua, V., Casorio, P., & Mastronardi, G. (2008). Extending Hough Transform to a Points' Cloud for 3D-Face Nose-Tip Detection. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence* (pp. 1200–1209). Springer Berlin Heidelberg. doi:10.1007/978-3-540-85984-0_144

Brenner, C. (2009). Extraction of Features from Mobile Laser Scanning Data for Future Driver Assistance Systems. In *Advances in GIScience* (p. pp 25–42). Springer Berlin Heidelberg. doi:10.1007/978-3-642-00318-9_2

Cabo, C., Ordoñez, C., García-Cortés, S., & Martínez, J. (2014). An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *87*, 47–56. doi:10.1016/j.isprsjprs.2013.10.008

Chib, S., & Greenberg, E. (2012). Understanding the Metropolis-Hastings Algorithm. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/00031305.1995.10476177#.VJn4wXlKA

Comaniciu, D., & Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603–619. doi:10.1109/34.1000236

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395. doi:10.1145/358669.358692

Golovinskiy, A., Kim, V. G., & Funkhouser, T. (2009). Shape-based recognition of 3D point clouds in urban environments. In *2009 IEEE 12th International Conference on Computer Vision* (pp. 2154–2161). IEEE. doi:10.1109/ICCV.2009.5459471

Greenspan, M., & Boulanger, P. (1999). Efficient and reliable template set matching for 3D object recognition. In *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062)* (pp. 230–239). IEEE Comput. Soc. doi:10.1109/IM.1999.805353

Hanson, G., Abrams, G. S., A.M. Boyarski, M. B., Bulos, F., Chinowsky, W., Feldman, G. J., … F.C. Winkelmann, and J. E. W. (1975). Evidence for Jet Structure in Hadron Production by e+e- Annihilation, *35*(24). doi:http://dx.doi.org/10.1103/PhysRevLett.35.1609

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*(1), 97–109. doi:10.1093/biomet/57.1.97

Hough, P. V. C. (1962, December 18). Method and means for recognizing complex patterns. U.S.: patent 3069654.

Kemboi, B. J. (2014). *Knowledge-based Detection of Road Furniture Objects in Mobile Laser Scanner (MLS) Data*. University of Twente Faculty of Geo-Information and Earth Observation (ITC). Retrieved from http://ezproxy.utwente.nl:2048/login?url=http://intranet.itc.nl/papers/2014/msc/gfm/kemboi.pdf

Khoshelham, K. (2007). Extending Generalized Hough Transform to Detect 3D Objects in Laser Range Data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *3*(Part 3/W52), 206 – 210.

Knopp, J., Prasad, M., Willems, G., Timofte, R., & Van Gool, L. (2010). Hough transform and 3D SURF for robust three dimensional classification, 589–602. Retrieved from http://dl.acm.org/citation.cfm?id=1888212.1888257

Kukko, A., Kaartinen, H., Hyyppä, J., & Chen, Y. (2012). Multiplatform Mobile Laser Scanning: Usability and Performance. *Sensors*, *12*(12), 11712–11733. doi:10.3390/s120911712

Lam, J., Mrstik, P., Harrap, R., Greenspan, M., & Engineering, C. (2010). Urban Scene Extraction from Mobile Ground Based LiDAR Data. *Proceedings of 3DPVT*, 1–8.

Lehtomäki, M., Jaakkola, A., Hyyppä, J., Kukko, A., & Kaartinen, H. (2010). Detection of Vertical Pole-Like Objects in a Road Environment Using Vehicle-Based Laser Scanning Data. *Remote Sensing*, *2*(3), 641–664. doi:10.3390/rs2030641

Leibe, B., Leonardis, A., & Schiele, B. (2007). Robust Object Detection with Interleaved Categorization and Segmentation. *International Journal of Computer Vision*, *77*(1-3), 259–289. doi:10.1007/s11263-007-0095-3

Leibe, B., & Schiele, B. (2003). Interleaved Object Categorization and Segmentation. In *Procedings of the British Machine Vision Conference 2003* (pp. 78.1–78.10). British Machine Vision Association. doi:10.5244/C.17.78

Luo, Y., Liu, Z., Lin, X., Wang, W., & Yu, J. (2005). Similarity search with implicit object features. *Advances in Web-Age Information …*, 1–12. Retrieved from http://link.springer.com/chapter/10.1007/11563952_14

Mc Elhinney, C., Kumar, P., Cahalane, C., & Mccarthy, T. (2010). Initial Resutls From European Road Safety Inspection (EURSI) Mobile Mapping Project. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XXXVIII, P*, 440–445. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.222.5409

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, *21*(6), 1087. doi:10.1063/1.1699114

Olson, D. L., & Dursun Delen. (2008). Performance Evaluation for Predictive Modeling. In *Advanced Data Mining Techique* (pp. 137–147). Springer Berlin Heidelberg. doi:10.1007/978-3-540-76917-0_9

Pham, M., Woodford, O. J., Perbet, F., Maki, A., Gherardi, R., Stenger, B., & Cipolla, R. (2013). Scale-Invariant Vote-Based 3D Recognition and Registration from Point Clouds. In *Machine Learning for Computer Vision* (pp. 137–162). Springer Berlin Heidelberg. doi:10.1007/978-3-642-28661-2_6

Pillai, N. S., Stuart, A. M., & Thiery, A. H. (2011). Optimal Proposal Design for Random Walk Type Metropolis Algorithms with Gaussian Random Field Priors. Statistics; Probability; Computation; Theory. Retrieved from http://xxx.tau.ac.il/abs/1108.1494v2

Pope, A. R. (1994). Model-based Object Recognition - A Survey of Recent Research. doi:10.1.1.42.9021

Pu, S., Rutzinger, M., Vosselman, G., & Oude Elberink, S. (2011). Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing*, *66*(6), S28–S39. doi:10.1016/j.isprsjprs.2011.08.006

Puente, I., González-Jorge, H., Arias, P., & Armesto, J. (2012). LAND-BASED MOBILE LASER SCANNING SYSTEMS: A REVIEW. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XXXVIII-5/*(1), 163–168. doi:10.5194/isprsarchives-XXXVIII-5-W12-163-2011

Puente, I., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2013). Review of mobile mapping and surveying technologies. *Measurement*, *46*(7), 2127–2145. doi:10.1016/j.measurement.2013.03.006

Ringnér, M. (2008). What is principal component analysis? *Nature Biotechnology*, *26*(3), 303–4. doi:10.1038/nbt0308-303

Shen, Y., Hermans, E., Bao, Q., Brijs, T., & Wets, G. (2013). Road safety development in Europe: a decade of changes (2001-2010). *Accident; Analysis and Prevention*, *60*, 85–94. doi:10.1016/j.aap.2013.08.013

Sithole, G., & Vosselman, G. (2005). Filtering of airborne laser scanner data based on segmented point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36(part 3)*, 66–71. Retrieved from http://www.isprs.org/proceedings/XXXVI/3-W19/papers/066.pdf

Sivic, J., & Zisserman, A. (2003). Video Google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision* (pp. 1470–1477 vol.2). IEEE. doi:10.1109/ICCV.2003.1238663

Song, S., & Xiao, J. (2014). Sliding Shapes for 3D Object Detection in Depth Images. In *Computer Vision – ECCV 2014* (p. pp 634–651). Springer International Publishing. doi:10.1007/978-3-319-10599-4_41

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., & Koehl, M. (2007). Model-driven and data-driven approaches using LIDAR data : analysis and comparison. In *ISPRS Workshop, Photogrammetric Image Analysis (PIA07)* (pp. 87–92). Retrieved from http://halshs.archives-ouvertes.fr/halshs-00264846

Toldo, R., Castellani, U., & Fusiello, A. (2010). The bag of words approach for retrieval and categorization of 3D objects. *The Visual Computer*, *26*(10), 1257–1268. doi:10.1007/s00371-010-0519-x

Velizhev, A., Shapovalov, R., & Schindler, K. (2012). Implicit Shape Models For Object Detection In 3D Point Clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *I-3*(1), 179–184. doi:10.5194/isprsannals-I-3-179-2012

Vosselman, G. (2000). Slope Based Filtering of Laser Altimetry Data. … *Archives of Photogrammetry and Remote Sensing*, *XXXIII*. Retrieved from http://www.researchgate.net/publication/228719860_Slope_based_filtering_of_laser_altimetry_data/file/50463524c7061a1220.pdf

Vosselman, G., Gorte, B., Sithole, G., & Rabbani, T. (2004). Recognising structure in laser scanner point clouds. *International Archives of …*, *XXXVI-8/W2*, 33–38. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:RECOGNISING+STRUCTURE+IN+LASER+SCANNER+POINT+CLOUDS+1#0

Vosselman, G., & Maas, H.-G. (2010). *Airborne and Terrestrial Laser Scanning* (p. 336). Whittles Publishing. Retrieved from http://www.whittlespublishing.com/Airborne_and_Terrestrial_Laser_Scanning

Weinmann, M., Jutzi, B., & Mallet, C. (2014). Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *II-3*, 181–188. doi:10.5194/isprsannals-II-3-181-2014

Williams, K., Olsen, M., Roe, G., & Glennie, C. (2013). *Synthesis of Transportation Applications of Mobile LIDAR. Remote Sensing* (Vol. 5, pp. 4652–4692). doi:10.3390/rs5094652

Zhang, Y., Jin, R., & Zhou, Z.-H. (2010). Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, *1*(1-4), 43–52. doi:10.1007/s13042-010-0001-0

APPENDIX

Table 6: Information of dataset subparts.

Sub datasets	Number of points	Length (m)	Width (m)
Dataset part 1	320999	99.50	88.03
Dataset part 2	240447	123.92	92.41
Dataset part 3	384098	108.43	78.19
Dataset part 4	282748	94.16	72.63
Dataset part 5	530861	84.84	64.38
Dataset part 6	562495	125.53	84.56

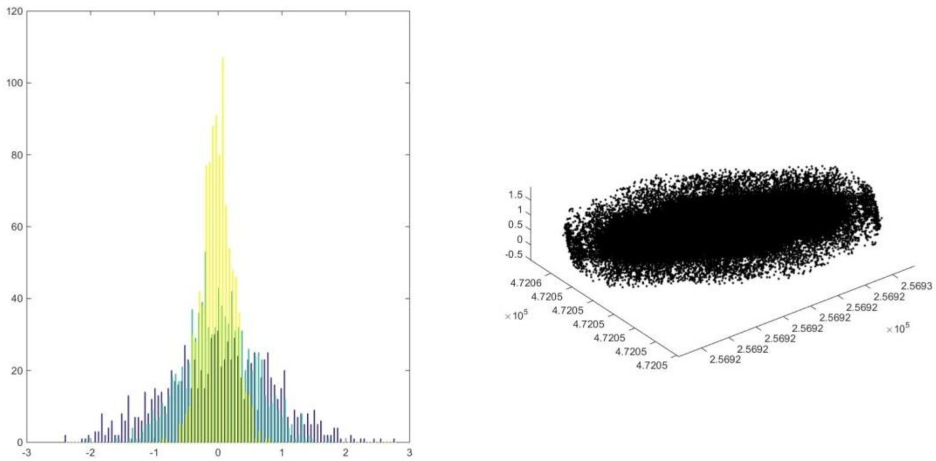


Figure 6-1: Mvnpdf distribution of car model.

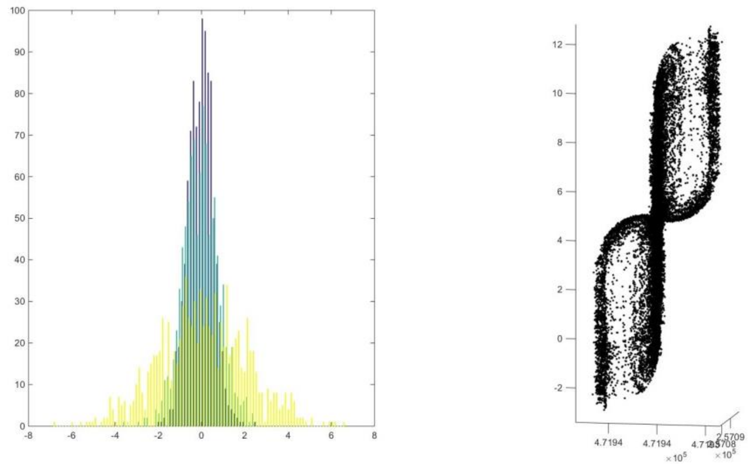


Figure 6-2: Mvnpdf distribution of lamppost type 1.

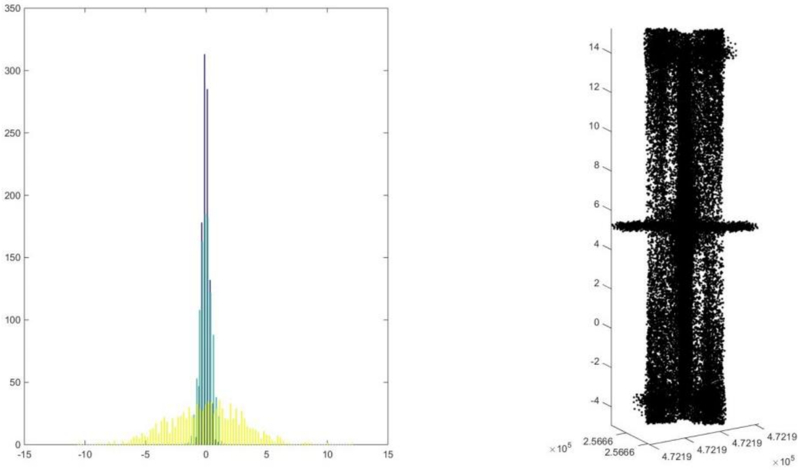


Figure 6-3: Mvnpdf distribution of lamppost type 2.

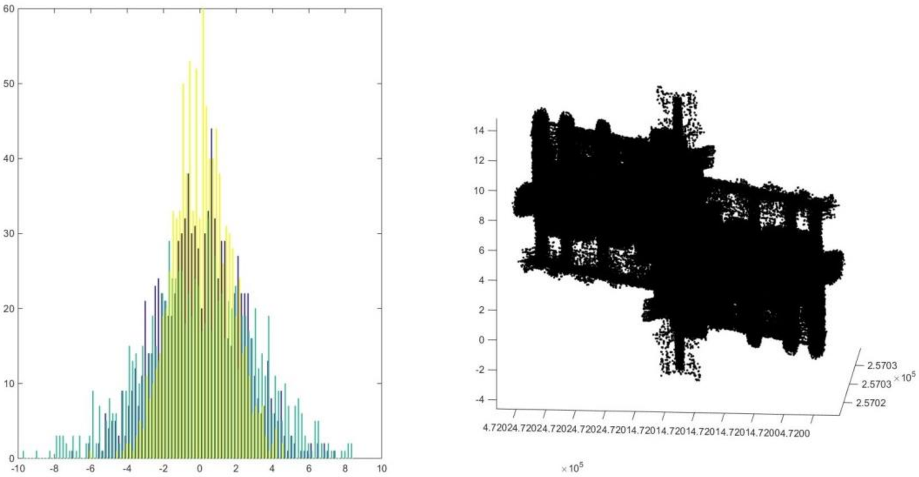


Figure 6-4: Mvnpdf distribution of traffic light type 1.

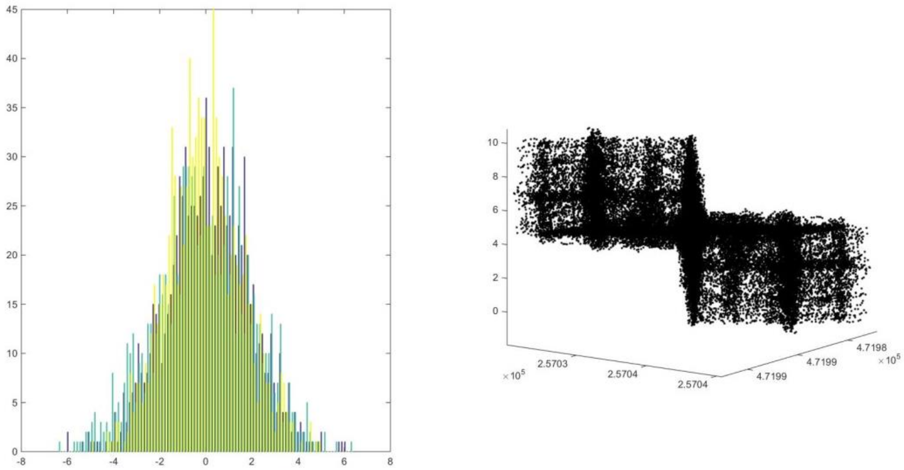


Figure 6-5: Mvnpdf distribution of traffic light type 2.

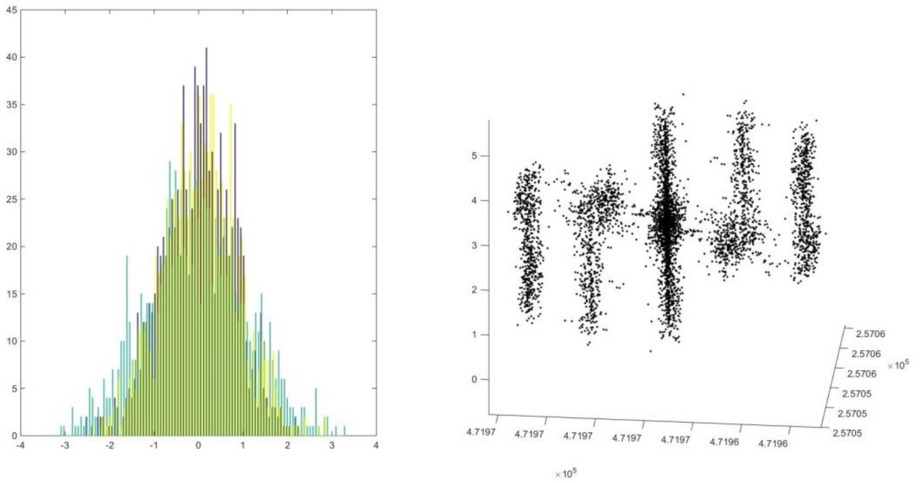


Figure 6-6: Mvnpdf distribution of traffic light type 3.