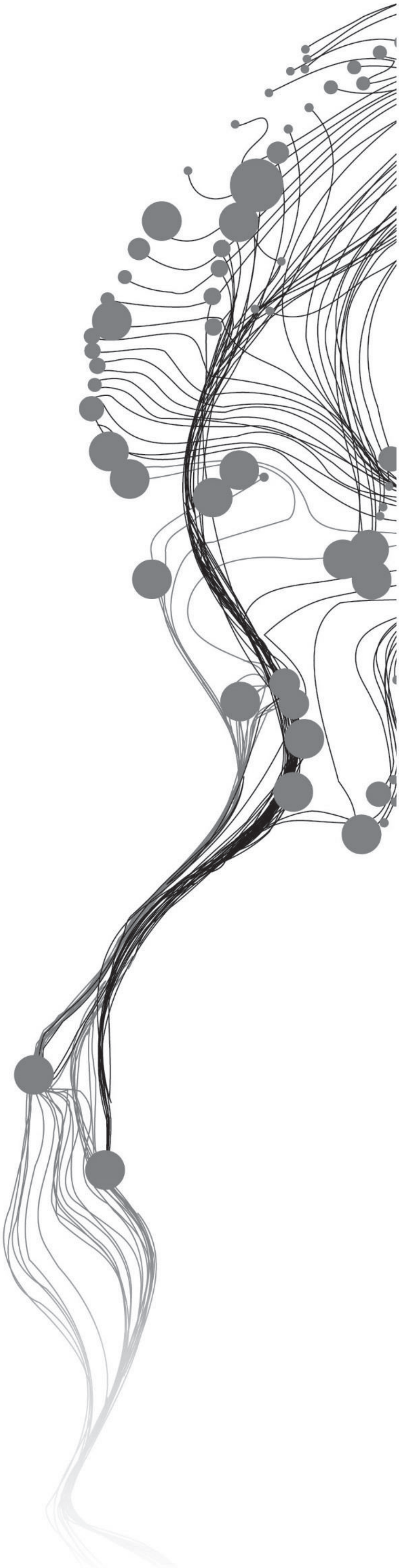
AUTOMATIC DETECTION OF MOVING OBJECTS IN MOBILE LIDAR POINT CLOUDS

XINWEI FANG
February, 2014

SUPERVISORS:
Dr. K. (Kourosh) Khoshelham
Dr. ir. S.J. (Sander) Oude Elberink



AUTOMATIC DETECTION OF MOVING OBJECTS IN MOBILE LIDAR POINT CLOUDS

[XINWEI FANG]
Enschede, The Netherlands, February, 2014

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. (Kourosh) Khoshelham
Dr. ir. S.J. (Sander) Oude Elberink

THESIS ASSESSMENT BOARD:
[Prof. Dr. ir., M.G, Vosselman (Chair)]
[Mag. Dr., M., Rutzinger (External Examiner, University of Innsbruck)]

ABSTRACT

Mobile laser scanner (LiDAR) is a state-of-art sensor which can acquire the data with high flexibility and precision, thus is widely used for urban scene point cloud acquisition. Automated road furniture extractions from those point clouds has been comprehensively studied as it can provide high accurate city model rapidly with low cost. However, the existence of many temporary objects in the urban scene has a negative impact on the extraction algorithm, since those temporary objects are of little use in most of urban applications and, moreover, they hinder the extraction of other permanent objects. Therefore, to detect and remove the temporary objects from the urban scene point cloud is essential for improving algorithm result. Considering data size and data structure of the point cloud, the manual detection of the temporary objects from the point cloud is not applicable because the human vision system is not sensitive to detect sparse point sets and the manual detection from the point cloud is subjective due to different time and operators. Thus, the objective of this thesis is to propose an automatic or a semi-automatic program to detect the temporary moving objects from mobile LiDAR point cloud. Generally, three key procedures were studied in this thesis: 1) data pre-processing, 2) detection of moving object and 3) result evaluation. For data processing, the surface growing and connected component segmentation was applied to remove the ground points and segment the points into object. Then the data trimming procedure was followed to determine and remove the maximum numbers of irrelevant points based on property of the city structure and temporary objects. For the detection of moving objects, recording the difference in the location of a moving object with two sensors is the key point. The detection algorithm firstly calculated the closest point pairs in the two sensors data, and aggregated the corresponding point-pair distances to find the closest segments. In addition, a rule based classification method is designed to determine if the closest pair was the moving object according to the distances of pairs and their orientations. For result evaluation, the ground truth dataset generated by visual examination was used for assessing the result of this method, where the result for the detection of moving objects reached a completeness of 90.0% and correctness of 93.1% respectively.

Keywords: Mobile LiDAR, object detection, moving objects.

ACKNOWLEDGEMENTS

I would like to take this chance to thank for all people who supported me during my Msc study and research in this 18-month period, without whom I will not be able to make it.

To my supervisors, thanks for giving me technique support and scientific suggestion.

To my friends, I will never forget you! Malakas!

To my family, I love you deep in my heart!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation

In recent years, the concept of developing a smart city is of great interest in mapping the urban area, and this stimulates significant demands for producing and developing 3D spatial products (e.g. 3D building Models, DTM, 3D SDI). In order to produce those products, the 3D information has to be extracted from the urban environment properly. Mobile Light Detection and Ranging (LiDAR) is an efficient and powerful tool to provide 3D data within the city area, which has been widely used for many applications(Pratihast, 2010; Ussyshkin, 2009). It emits laser beams and records the time of flight or phase time to estimate the distance between the objects and the LiDAR sensor. Then, a point cloud is generated to represent the 3D environment according to the time and distance information obtained from those laser beams. However, in urban areas, due to the complexity of the scene, the point cloud may contain many points of temporary object such as cars, bikes and pedestrians, and these redundant points will hinder the extraction of objects of interest such as buildings, roads and lamp-post. If those temporary objects were removed, the object (e.g. buildings, roads) extraction process will be much simpler. Moreover, the resulting gaps in the point cloud can be filled with other data sources, which will increase the accuracy and completeness of existing algorithms for 3D information extraction in urban areas. However, in order to remove those temporary objects, they have to be identified first, hence, the detection and extraction of temporary object have to be applied in advance. Generally, temporary objects are normally divided into different sub-categories according to their properties or features. Moving object is always a major category, partly due to the plenty of moving cars, bikes and pedestrians existing all the time in city scenes. One way to detect and remove those objects is manual detection. However, it is not possible to detect all the moving objects by manual selection, because processing the huge amount of data is time consuming and labour costly. In addition, as human vision system is not accustomed to explore sparse point sets, the manual detection from the point cloud is subjective due to different operators. Considering those limitations, it is necessary to develop an automatic or a semi-automatic method to detect the moving objects in the point cloud.

1.2. Problem statement

Several automated algorithms for detection of objects in urban scenes point cloud have been proposed. However, due to the complexity of urban environment, the robustness of those methods and the accuracy of the results still needed to be improved (Sato et al., 2010; Velizhev et al., 2012). Furthermore, as the shape of moving object in the point cloud are always deformed or distorted and they do not follow any properties of common geometry, detecting moving objects in the urban scenes point cloud is more difficult. Most of the algorithms for detecting moving objects require the multi-layer dataset (more than 2 layers) meaning that all data are captured in the same study area in different time and they are all geo-referenced. This method would have a large amount of redundant data to detect the moving objects, but this limits its application because not all LiDAR survey could provide the multi-layer datasets. Moreover, for the point cloud point, the data are captured in the urban area where the complexity of environment leads to a difficulty of separating points of an object from the adjacent objects. Therefore, it is difficult to detect moving objects from urban scene LiDAR point cloud.

1.3. Research Identification

1.3.1. Research objectives

According to the motivation (chapter 1.1) and problem statement (chapter 1.2), if temporary objects were detected and removed from the raw dataset, the accuracy of existing methods for information extraction and the automatic capability of present algorithms would be significantly improved. Therefore, this research will focus on how to effectively and efficiently detect the moving objects from a point cloud. Thus, the objective of this MSc research is to develop an automatic or a semi-automatic method to detect moving objects from vehicle-based LiDAR point cloud (two-layer LiDAR data).

1.3.2. Research questions

1. How efficiently can an automated algorithm detect moving objects?

2. How can the moving objects be correctly extracted in the complex urban scenes point cloud?

3. Moving objects are usually on the road with a certain size/height and moving pattern along the road. Additionally, the scan angle between two scanners and the distances between objects and LiDAR will influence the shapes of the captured object in the point cloud. How to incorporate such information in a method for automatic detection of moving objects?

4. Can the research outcome be used in other applications for LiDAR data pre-processing?

1.3.3. Anticipated result

The final result is expected to be an efficient automatic or semi-automatic program which can identify the location of the moving objects in a point cloud with relatively high completeness and correctness.

1.4. Related work

An object in LiDAR point cloud can be identified if all the points relating to the object are correctly extracted and classified. Extracting objects from LiDAR point cloud in urban area is difficult due to the complexity of the environment(Vosselman & Maas, 2010). However, with the growing interest in digital smart city and the demand of highly accurate 3D building models, many approaches are developed for extracting urban objects from LiDAR point cloud(Brenner, 2009; Vakautawale, 2010). Nevertheless, the studied objects primarily focus on the buildings, roads and pole-like objects (tree trunk, traffic sign, lamp post etc.) (Lehtomaki et al., 2010; Pu et al., 2009; Yang et al., 2013) because these objects shared the common geometric properties (planar, cylinder shape etc.), from which points can be formulated and get separated from adjacent objects. For example, Elberink et al. (2009) applied the roof topology graph in building roof reconstruction by using the property that each face of roof is a large planar and intersection of those faces derive the building roof topology. LI (2013) extracted pole-like objects from MLS point cloud data. Cylinder shape property was used in her work to filter out the uninterested points and extract pole-like objects. After object extraction, the points relating to those objects can be identified easily.

However, the temporary objects, specially moving objects, in the urban area point cloud have been found to have a negative impact on the extraction of urban permanent objects. Yang et al. (2013) pointed out that the moving cars are a major problem for extracting road features in urban areas. Lehtomaki et al. (2010) mentioned that occlusion by moving cars is difficult for extracting pole-like objects in the developed algorithm. In addition, when applying vertical wall extraction to assess solar potential areas, Jochem et al. (2011) concluded that the result might be biased due to the presence of temporary objects, which increased the total shadowed areas in the calculation. Therefore, it is necessary to detect and remove the moving objects in the point cloud.

Studies on detection of moving object have been conducted for many years. Gibson (1954) published a paper about human visual perception of the movement. He set up several experiments to test how human eyes detect and react to objects movement. It can be seen as a very first research in detection of moving object. In late 80s, as personal computer was invented, detection of moving object in computer vision started to develop. Thompson et al. (1990) summarised three basic scenarios describing movements related to camera(sensor), objects and environment (background), these fundamental relations were used in image or video based moving object detection (Betke et al., 2000; Gavrila, 2000). In recent years, with the improvement of computer's processing speed and the implementations of sensors, the detection of moving objects researches have been wildly applied in different fields with various types of sensors(Hansung et al., 2012; Pokrajac et al., 2005). Those research activities are mainly conducted in the robotic automation and automobile industry, and most of these researches rely primarily on the optical sensor to identify the movement. By extracting the object features in images or videos sequence, the methods for detection of moving based on tracking the position changes with respect to the time are exploited. Those algorithms are very handy to perceive a real-time and high accuracy detection.

The detection of moving object in the image or video sequence can be considered as the change detection in point cloud. Because both techniques are seeking location difference of points/objects in respected to time. The changes in airborne LiDAR point cloud are analysed by comparing data from two epochs. Two standard approaches were primarily applied: 1) using two LiDAR datasets, which covered the same area but were obtained in different time, and comparing them in the object domain(Rutzinger et al., 2010; Xu et al., 2013); 2) using one LiDAR data and comparing it with an existing reference data such as a topographic database (Vosselman et al.). The change can be seen after comparing the two datasets. The detection of changes and tracing of moving objects in mobile based LiDAR have also been studied. The multilayer dataset were often used as they provide large amount of redundant information. With help of auxiliary sensors (camera), the motion of temporary objects can be detected (Gavrila, 2000; Schütz et al., 2013).

2. LIDAR TECHNOLOGY

2.1. General laser scanning technology

Multiple technologies had been developed and applied in the past few decades in order to obtain the 3D information from the real world. Nowadays, two modern methods are primarily used for conducting most of 3D measurement, which are 1) based on the triangulation or 2) based on the property of light (laser scanning). The triangulation method, in detail, could be classified into two primary categories, which are passive measurement and active measurement. The tradition application in passive triangulation method is image stereo matching, which use multiple images based on the trigonometry to obtain the 3D scene. The active triangulation methods are less used, however, several classic methods, like color-coded-projection, light sheet projection, single spot projection, are the typical applications in use. Moreover, the method using the property of light to measure the travelling time of single pulse between object and sensor is used in the laser scanning. With the knowledge of the speed of light and the position of the system, the relative position of the object from the system will be obtained. This type of system is named as time-of-flight or LiDAR (Light Detection and Ranging). And the absolute position of the sensor is measured by the GPS. Figure 2-1 describes the standard methods of optical 3D measurements.

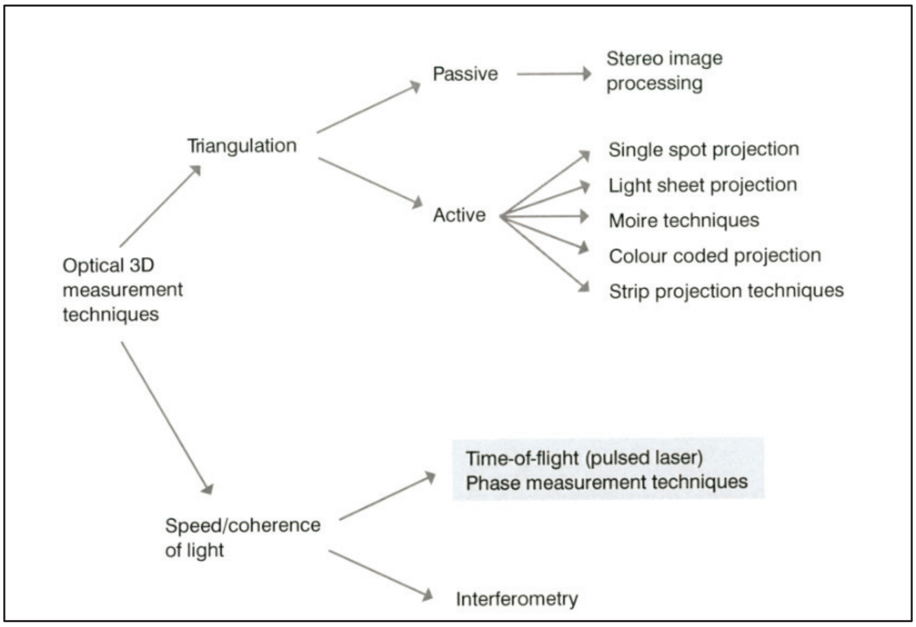


Figure 2-1: Structure of 3D measurement techniques(Vosselman et al., 2010)

The LiDAR system, in detail, could also be further classified into Airborne LiDAR and terrestrial LiDAR. Both of them obtain the texture and 3D geometry remotely by using the property of the light and generate the range map or point cloud in three dimensional coordinate. Time-of-flight (pulse laser) and phase measurement are the two major techniques which are commonly used in the state-of-art LiDAR system.

2.1.1. Time of flight

As the name indicated, the time of flight calculates the distance between the object and sensor by the time that light travels. The laser light generator is placed in a LiDAR system. A mirror is attached on the end of the laser generator to redirect the laser light to the object. And there is a laser beam detector attached next to the mirror to receive the laser pulse reflected from the object as shown in Figure 2-2. It is assumed that the medium between the LiDAR and the object is homogeneous and the velocity of laser beam transmitted in the medium is known. Then, the distance which related to the position of LiDAR can be calculated by knowing a travelling time of laser beam according to the Equation 1.

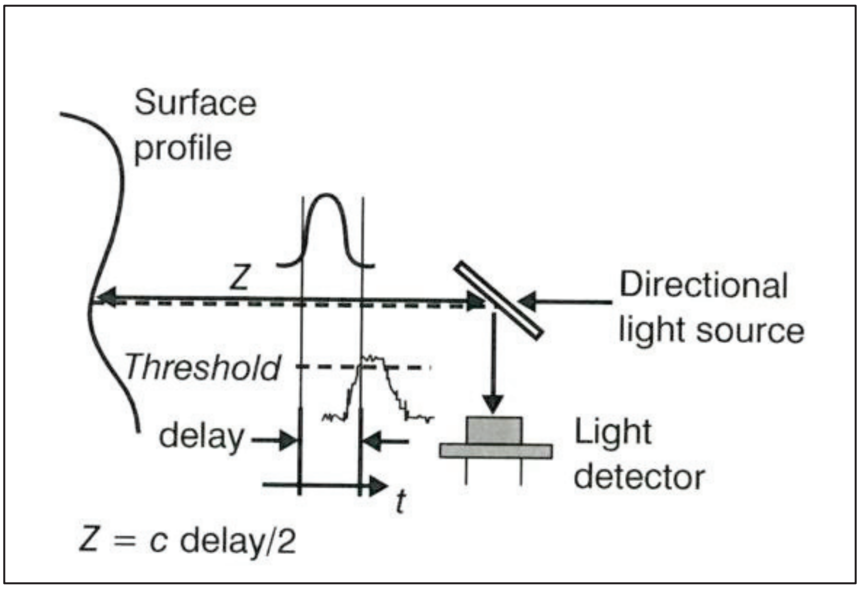


Figure 2-2: Light transit time (NRC Crown copyright) (Vosselman & Maas, 2010)

$$D = c \times \frac{\Delta t}{2}$$ Equation 1

D : The distance between the LiDAR detector and the object
c : The speed of laser beam travelling in the given medium
Δt : The measured time from the moment when laser beam emitted to the moment where the same beam is detected.

2.1.2. Phase measurement

The higher precision measurement can be achieved by using phase measurement technique. Instead of using speed of light to calculate the time delay, a modulated wave is introduced and embedded on the continuous wave to determine the time delay by comparing the phase shift between the out-going and in-coming waves. Both frequency modulation and amplitude modulation can be used as carrying frequency showing in Figure 2-3. However, the measuring range of phase measurement technique is limited by the wave length. If the object is located beyond that range, the accuracy of measured distance is significantly reduced. The distance from the LiDAR system to the object calculated by phase measurement is computed based on equation Equation 1 and Equation 2.

$$\tau = \frac{\Delta\varphi}{2\pi}\frac{\lambda_m}{c} \qquad \text{Equation 2}$$

τ : Time delay
$\Delta\varphi$: Phase difference between two waveforms
c : The speed of laser beam travelling in the given medium
λ_m : Wavelength of the modulation

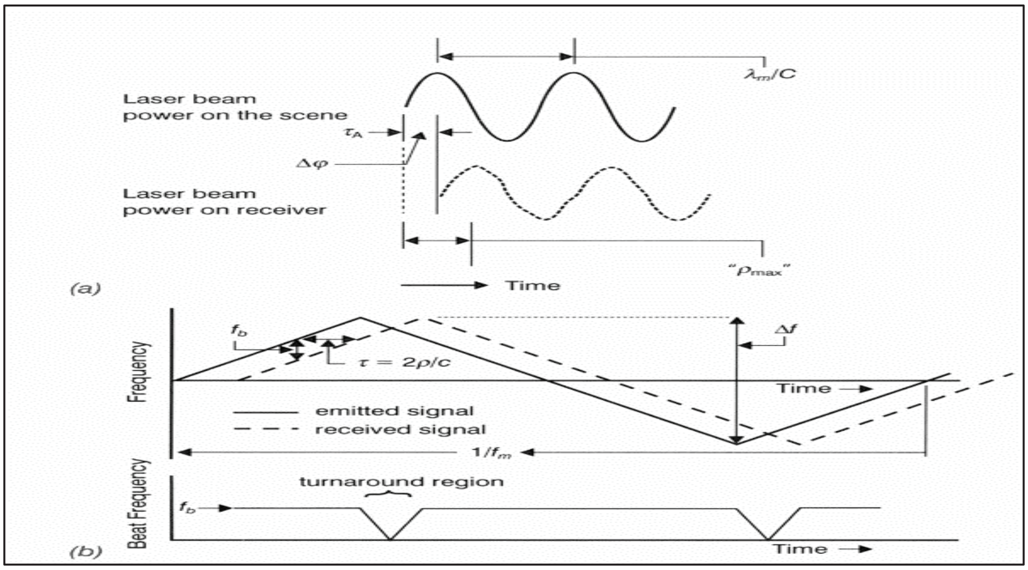


Figure 2-3: Phase measurement (NRC Crown copyright) (Vosselman & Maas, 2010)

2.1.3. Other measurement

To acquire the results with higher precision, the triangulation-based measurement could be applied with position-sensitive detectors. However most of LiDAR systems are designed based on Phase measurement and time-of-flight measurement. More detailed review can be found in (Shan et al., 2008; Vosselman & Maas, 2010)

2.2. Vehicle-based LiDAR

The vehicle-based LiDAR uses the same technology (time of flight or phase measurement) to determine the position of the object. The LiDAR sensors are mounted on the top of the vehicle where the maximum scanning areas can be covered. Inertia Measurement Unit (IMU) and Global Positioning System (GPS) are also on board to maintain the absolute position of the sensor when the platform is moving. Two rotated scanners with 90 degrees of angles perpendicular to each other are commonly used in vehicle-based scanning system. The sensor coverage, orientation and position can be seen in Figure 2-4: Vehicle-based LiDAR (OPTECH.ca & TOPSCAN.de).

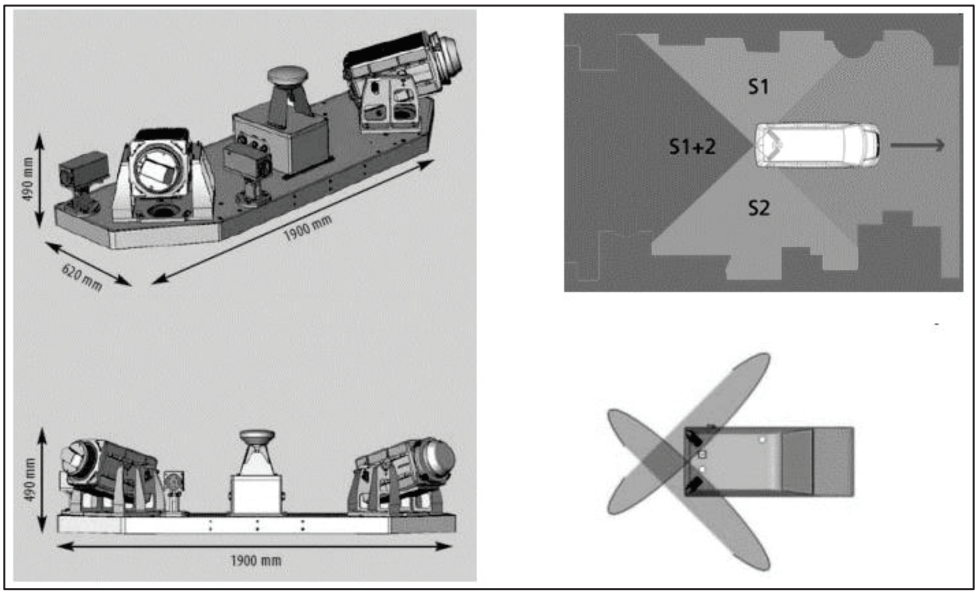


Figure 2-4: Vehide-based LiDAR (OPTECH.ca & TOPSCAN.de)

3. METHODOLOGY

This chapter discusses the methods that were applied for detecting the moving objects. Three procedures are carried out. They are 1) Data pre-processing, 2) Detection of moving objects and 3) Result evaluation method. Segmentation is always the first step when performing the object extraction in point cloud (Vosselman & Maas, 2010), therefore, two segmentation methods and two data trimming strategies are introduced for data pre-processing in chapter 3.2. For detection of moving objects, as the dataset obtained from two sensors with slightly time interval (two-layer), the detection algorithm based on the features of moving objects is applied and performed in chapter 3.3. The evaluation of the method performance is given in chapter 3.4 based on the reference data which is generated by visual examination.

3.1. General workflow

The moving objects normally appear on the road as groups of points, and the size of those moving objects is relatively small compared with other large objects (e.g. ground, building and tree). Hence, the data pre-processing is firstly to remove maximal irrelevant points/objects and preserve maximal interested points/objects. This process reduces the size of the data and it will enhance the efficiency of the detection program.

The reference dataset was generated after the pre-processing process because not all the moving objects can be extracted through the visual extraction from the raw dataset. The reference dataset is evaluated by comparing the raw data and pre-processing result. The labels are assigned to each segment in the reference dataset by visual examination.

According to the property of mobile LiDAR, the two onboard sensors are facing to the different direction (see Figure 2-4). With the moving of the platform, the same object is covered by two sensors from different angles and with different time. Around one second time interval between the coverage of two sensors is expected (described in chapter 3.3.1). With that knowledge, the method for detection of moving objects is developed in both point space domain and object space domain where all points are firstly searching for their closest point in another sensor. After that, all points are aggregated into a segment with their segmentation number to determine the closest segment pair. The method is under the assumption that the point segment relating to a static object from both scanning sensors are the closest pair and stay geometrically closer; on contrary, the point segment relating to a moving object are also the closest pair but the pair followed certain orientation and distance between the two paired segments is below a certain value.

With that assumption, the threshold values and other constrains are used for separating the moving object out of other objects. In the end, the automated detection result is compared with the reference dataset to determine the final completeness and correctness of the application. The frame work is demonstrated at Figure 3-1.

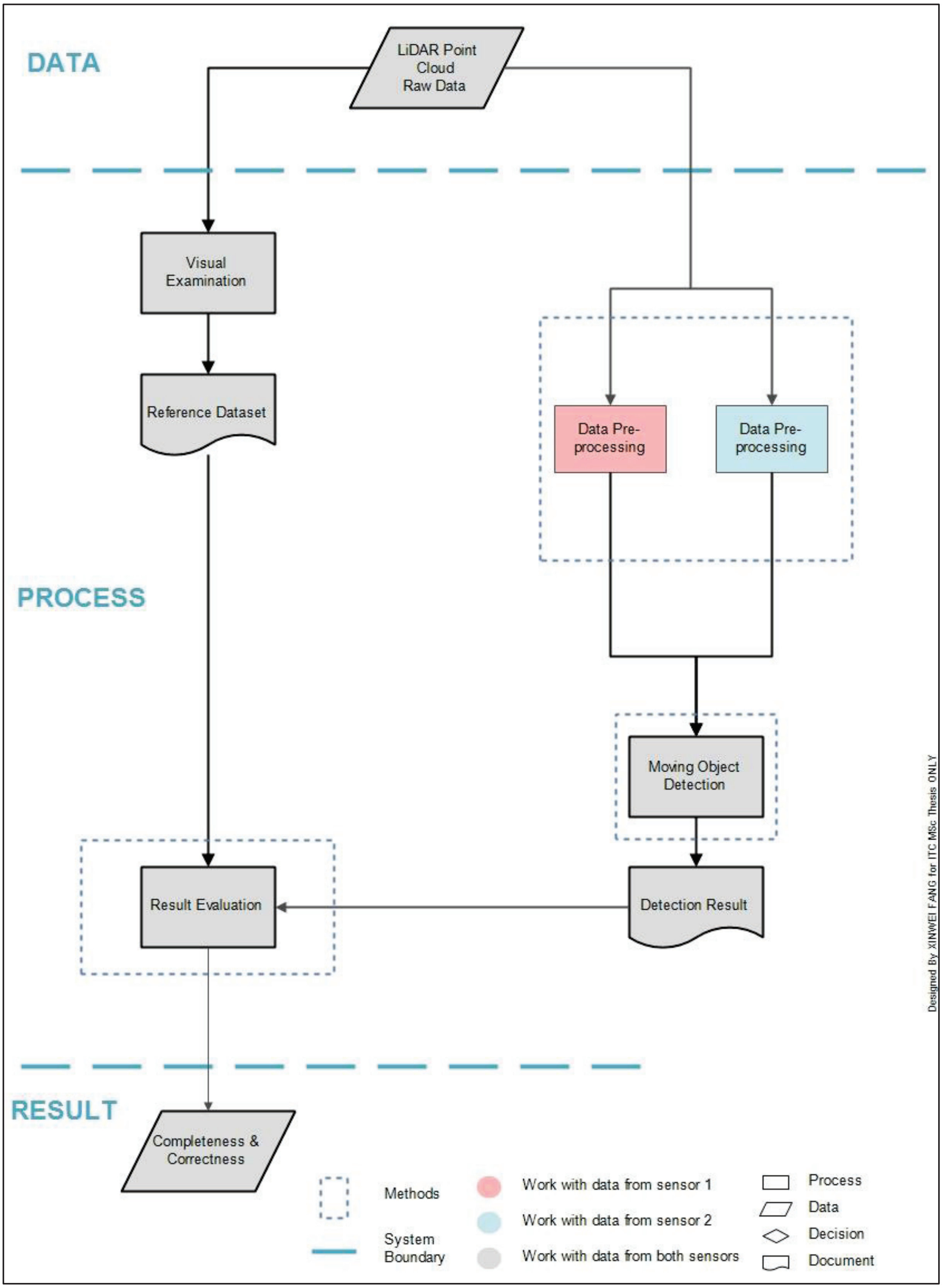


Figure 3-1: General workflow

3.2. Pre-processing

As the LiDAR point cloud always contains enormous numbers of points, running entire dataset in the program are extremely time consuming. Moreover, the desired objects of this study are moving objects, which primarily are cars, bikes and pedestrians. Excluding other objects (ground, building façade, etc.) in the dataset will not only reduce the computational cost but also reduce the complexity of the algorithm for detection of moving objects. Therefore, applying a rough classification before the object detection, removing irrelevant points/objects and segmenting points into objects, is essential. Pu et al. (2011) proposed a method to separate entire dataset into three labeled datasets: ground component, on-the-ground component and off-the-ground component. The ground component is defined by large planar segments which is greater than a certain size and below the 3D trajectory. Once the ground component is generated, the polygon constructed by the outlines of the ground component is also generated. The size of the polygon denotes the areas of ground surface. The on-the-ground component is defined by the component which has points attached on the ground segment and within the areas of the polygon. The off-the-ground component is defined by the segments which fall out of the polygon or not directly connected to the ground segment. As the most of moving objects are considered to be on-the-ground components, Pu et al. (2011)'s method and computer program were applied to extract the objects on the ground. This method included two segmentation methods (i.e., surface growing and connected component) and ground, off-ground segments removal and building facade removal. The result of this procedure was further analyzed in the bounding box based trimming process to remove the irrelevant objects. The workflow of segmentations is shown in Figure 3-2.

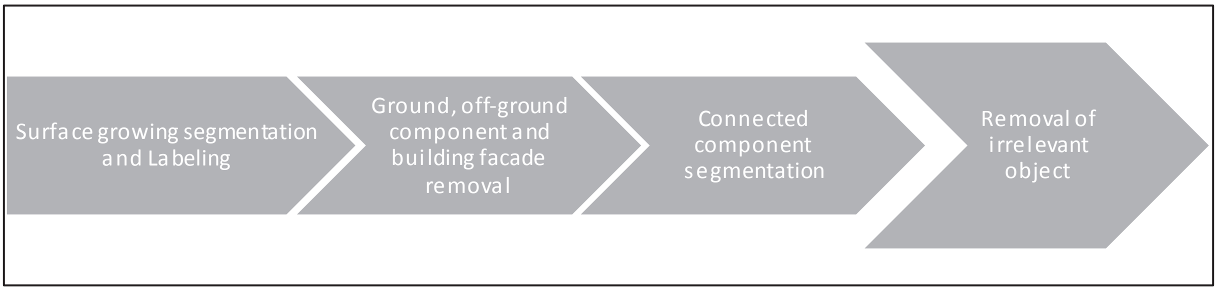


Figure 3-2: Process of segmentation

3.2.1. Surface growing

Surface growing method are the first step of the data processing and it can be seen as an extension of 2D region growing method applied to 3D domain. The 2D region growing method groups the neighboring pixels together only if those pixels satisfying a common criteria. Similarly, surface growing uses the same concept, but applied in the 3D domain. If the nearby points shared the same properties, like planar, they are formulated into a segment. Moreover, like 2D region growing method, the surface growing method also includes two steps: the surface seeds selection and growing of the surface.

The starting seeds are required for formulating the original plane. The seeds are normally selected automatically according to the parameter settings, 1) the seed neighborhood radius and 2) minimum number of seed points. The seed neighborhood radius defines the areas where the initial seeds can be randomly generated. The minimum number of seed points defines the numbers of initial seed points should be generated within the defined areas. Once the seed points are selected, the certain algorithms (RANSAC, 3D Hough transform, robust plane fitting, etc.) are applied to determine if selected points can be fit into a plane. If the plane was generated successfully, it will be used for surface growing process later. If the seed points did not fit a plane, they will be discarded and the program would determine another set of seeds until the required plane is fitted. Once the seeds have been generated, the growing process is ready to start.

In the growing process, two primarily models, namely Planar model and Smooth model, can be selected. In the planar model, the planarity of points is the fundamental key to formulate the plane. By applying the data structure, like K-d tree, the nearby points are added and calculated to determine if the plane will grow or not. In this way, all the original planes are expanded until no more neighboring points can be added. For planar model, re-estimation of the planar parameter to determine the best fit plane is necessary. However, if the plane is too big, several outliers will not influence the fit of the plane. therefore, Vosselman and Maas (2010) suggested that to re-estimate the planar parameters each time as the surface expending 20 to 50 percentages of areas. On the contrary, in the smooth model, the growing depended on the smoothness of local plane. The smoothness of surface is defined by the maximum distance to surface value. The smaller distance indicates the smoother surface. However, as it computes the plane fitting locally, the large angle or sharp edge which normally specifies the boundary of two planes may not be detected.

The surface growing segmentation is applied to the entire dataset where all points are segmented according to the planarity. Once the dataset are segmented into planar objects, the ground, off-ground components and building facades are classified and removed.

3.2.2. Ground, off-ground components and building facade removal

The ground points and building facade points occupied large volume in the dataset and they are not considered to be useful in this study. Moreover, the off-ground components are also not useful in the algorithm for detection of moving object because they have lower chance to be the temporary object. Hence, if that amount of volume can be decreased, the efficiency of the program will be significantly improved and the complexity of the detection algorithm will also be greatly reduced. Therefore, it is essential to identify and remove ground, off-ground components and building facade objects. After surface growing, the planar segments are formulated, several constrains can be set up to filter out those irrelevant segments.

- Ground segments: the ground surface segments are considered as a large horizontal planar objects, which were under certain distance of the 3D trajectory.

Once the ground components are identified, the 2D outline of the ground surface is also generated.

- Off-ground component: Off-ground-components are considered to be segments which fall out of the 2D outline or not connected to the ground surface.

- Building façade: the building façade segments are considered as a large vertical planar objects, which connected to the ground surface segments.

After removing those segments, the on-ground components are remains.

- On-ground components: the segments did not belong to any segments which are listed above and have points connected to the ground segments directly.

The dataset after the rough classification is much smaller than the original one, but all the objects of interest are expected to remain. It improves the processing speed significantly in the following procedures such as connected component segmentation and data trimming.

3.2.3. Connected component segmentation

The detection of object in an image is often done by 2 dimensional grid analysis, where only x and y are considered. Similarly for the airborne LiDAR point cloud, as it captures data from above, most of products from airborne LiDAR is 2.5D. It means each pair of (X,Y) coordinate can only have one height value, which can be formulated as a function Z = (X,Y). Therefore, applying 2.5D airborne LiDAR data to 2D grid is also applicable. However, the LiDAR data captured by mobile or terrestrial scanner could not be applied in the same procedure as the data obtained inside of the environment as real 3D data. If forcedly converted the 3D LiDAR data into 2D grid for processing, the huge amount of information will be lost. Hence, voxels space which is a pixel in 3D space is introduced to adapt the operation from 2D to 3D. The value of the voxel is defined by the numbers of points that stayed inside of each voxel. Furthermore, each voxel has 26 neighbors instead of 8 neighboring pixels in 2D space, indicating 6 of them sharing the face, 12 of them sharing the edge and 8 of them sharing the corner. The connected component segmentation is applied based on the 3D voxel space. In this study, the distance between points will be evaluated to get the most optimal result where it has minimum over-segmentation and under-segmentation and preserves all the moving objects. Moreover, in order to preserve all the moving objects at this stage, the minimum points for formulating one segment is set to be lowest. Once the points are formulated into object segments, the permanent objects (e.g. trees, lamp-post) can be identified and removed as they are not of interest in this study. The irrelevant object trimming is then applied after this process.

3.2.4. Trimming of Irrelevant objects

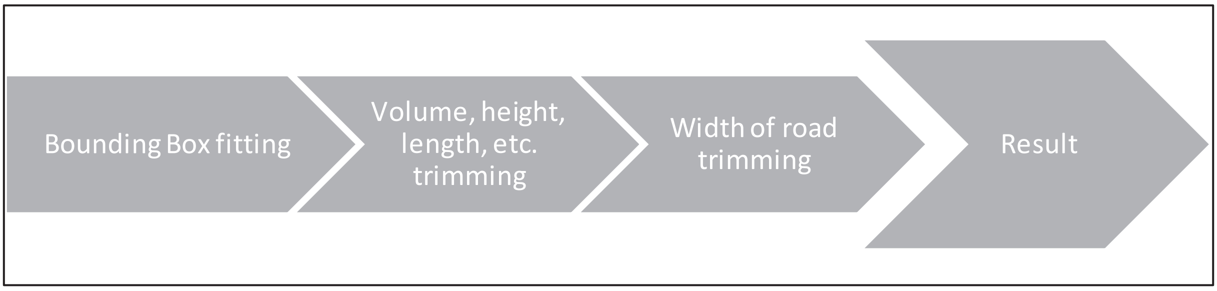


Figure 3-3: Process of object trimming

3.2.4.1. Bounding box

According to the process of object trimming shown in Figure 3-3, the minimum bounding box is applied first. The code of calculating the minimum bounding box is adopted by (Korsawe, 2008). In this program, the minimal volume is calculated to denote the minimal bounding box and all adjacent box edges are perpendicular to each other. The multiple information is given from the bounding box fitting, such as volume, corner points, etc. The rules of bounding box based object trimming are conducted according to the difference between the properties of moving objects and properties of the other objects. After the bounding box based object trimming, more irrelevant points/objects are expected to be removed and all moving object segments are kept. The rules are listed below.

- Volume: if the volume of bounding box was larger than a certain threshold, the segment is removed.

The volume of the objects is expected to be smaller than a certain size because the moving objects have relatively small size of volume when compared with building or tree. The biggest moving object is considered to be a road train. However, as the study area is located within the city, the road train is mostly absent. Hence, the lorry is considered to be the biggest moving object. According to the transportation document which was developed by EU (internationaltransportforum, 2012), the maximal size of lorry allowed in Netherlands is $4\mathrm{m} \times 2.55\mathrm{m} \times 12\mathrm{m} = 122\mathrm{m}^3$ $(\text{height} \times \text{width} \times \text{length} = \text{volume})$. Therefore, the threshold for volume is 122 cubic meters for bounding box based object trimming. If an object has the volume greater than that this threshold, it will be removed.

- Long edge: if the longest edge of the bounding box was longer than a certain threshold, the segment is removed.

The longest edge for moving object is considered to be the length of public transport bus. The overall volume size of the bus is smaller than lorry but the length of the bus is longer. According to the specifications of city buses manufactures (MAN_Bus_EU, 2012; VDL_BUS&COACH, 2014), the average length of the city buses is designed for 15 meters long, at which the bus can obtain the highest efficiency. Hence, the long-edge threshold is determined at 15 meters. The traffic island curbs or long building parts are expected to be removed in this case.

- Maximum height: if the height of the segment was greater than a certain value, the segment is removed.

The most of height limits in city are set around the 3.5m to 4m. Moreover, with consideration of the height of lorry, the height limitation in the program is set to be 4 meters. The trees, lamp-posts and a part of buildings are removed in this case.

- Minimum height: if the height of the segment was below 1 meter, the segment is removed.

The height of the moving objects, pedestrians, cars or bikers, are all greater one meter. Therefore, one meter minimal height is set to remove some lower bushes or small surfaces which were not removed from ground removal process.

- Absolute height: if the lowest point of a segment is above one meter from the ground, the segment is not considered in the following process.

This constrain eliminates the segments which are floating in the air. As the ground elevation changes, there is no absolute value to determine the ground level. Therefore, the 3D scan trajectory is used for estimating the local elevation.

- Minimum numbers of points in one segment: if the numbers of points for a segment is smaller than a threshold, the segment is discarded.

In order to preserve all the moving segment, the minimum value of formulating one segment was set to be minimum in connected component segmentation process. However, at this step, the small segments which not belong to the moving objects are removed by visual analyzing. The threshold value is tested by manual examination and it is based on the rule that all moving objects have to be preserved. In order to get the most optimal result, where the maximum numbers of irrelevant small objects are removed and all moving object are preserved, the threshold value has to be determined by manual examination. However, for automated program, the threshold value has to be estimated according to the data quality.

3.2.4.2. Lamp-post data trimming

In the previous segmentation process, the road width is defined by extending the width in cross road direction in respected to the 3D trajectory. It means that the road area defined in the dataset is symmetric to the 3D trajectory. However, in reality, the trajectory is recorded on the one side of driving lane and it is not the centre of the road. That inaccurate estimation leads to the result that one side of the objects that should belong to the off-road component are mis-included in the dataset. Segments, like building parts, off-road parked cars, are not completely trimmed out in the bounding box trimming process. In order to exclude those segments, the proper width of the road needs to be extracted. Moreover, according to the properties of moving object, the moving objects always appear on the areas of main road, bike path and sidewalk, which are normally within the road boundary. Therefore, using the width of the road to trim the data will not remove the moving objects, and it can remove the maximum numbers of irrelevant objects, which are located far away from the road.

Yang et al. (2013) considered the main road as a planar surface which is bounded by the curb stones. However, the most curb stones are occluded by the parked cars. Therefore, other indication has to be found to determine the width of road. In automobile industry, the research teams detected lamp-post for guiding automatic driving cars instead of using curb stone (Brenner, 2009). LI (2013) detected road furniture which included the lamp-post in the same study area. Thus, her results is used for determining the width of the main road in this study. Moreover, according to (Municipality_Enschde, 2010), for each type of the road, the width of bike path and part of side walk can be estimated by extending the certain width in cross road direction. Therefore, knowing the boundary and type of the main road, the areas where most of moving objects appear including main road, bike path and sidewalk can be derived. The width of main road, in my study, is defined by distance between two sides of lamp-posts. Moreover, the orientation of my study area is consistent, hence, the linear fitting for each side of the lamp-posts are used to determine the two boundary lines of the main road. Two linear functions are generated to represent the each side of the main road boundary. Once the types of road are identified, the width for bike path and sidewalk related is extended and the total areas of road, including sidewalks, bike path and main road are formulated. If the center of the segment is in between two boundary lines, the segment is considered in the following procedure. Otherwise, the segments are discarded. After data pre-processing, the dataset is ready to be performed in the detection process. The dataset after pre-processing is much smaller and contains less irrelevant points, which is expected to enhance the processing speed in the object detection process.

3.3. Detection of moving objects

Once the data was pre-processed, the detection of moving objects is followed. According to the property of vehicle-based LiDAR scanner in chapter 2.2, two-layer LiDAR point cloud which were captured from two sensors with slightly time intervals are used in the method. The detection algorithm is conducted in both point space domain and object space domain as working in a single domain does not give the proper result. The fundamental concept for detection of moving objects is under the assumption that the point segment which belong to one static object from two sensors are located relatively close if compared with moving objects. In addition, the closest distance is used for determining the segments pair because, in theory, the two segments which belong to one moving or static object from two sensors are stay closer than two segments from different objects (chapter 3.3.1). With that assumption, if the closest pair of segments were found, they have higher chance to be the same object. By analyzing those closest pairs of segments, threshold value is designed to exclude the static object in the following process. By further analyzing the properties of moving objects, the constrains, such as moving direction or moving distance, are used for separating moving objects out of other objects. The detailed work flow of the detection of moving objects is depicted below in Figure 3-4.

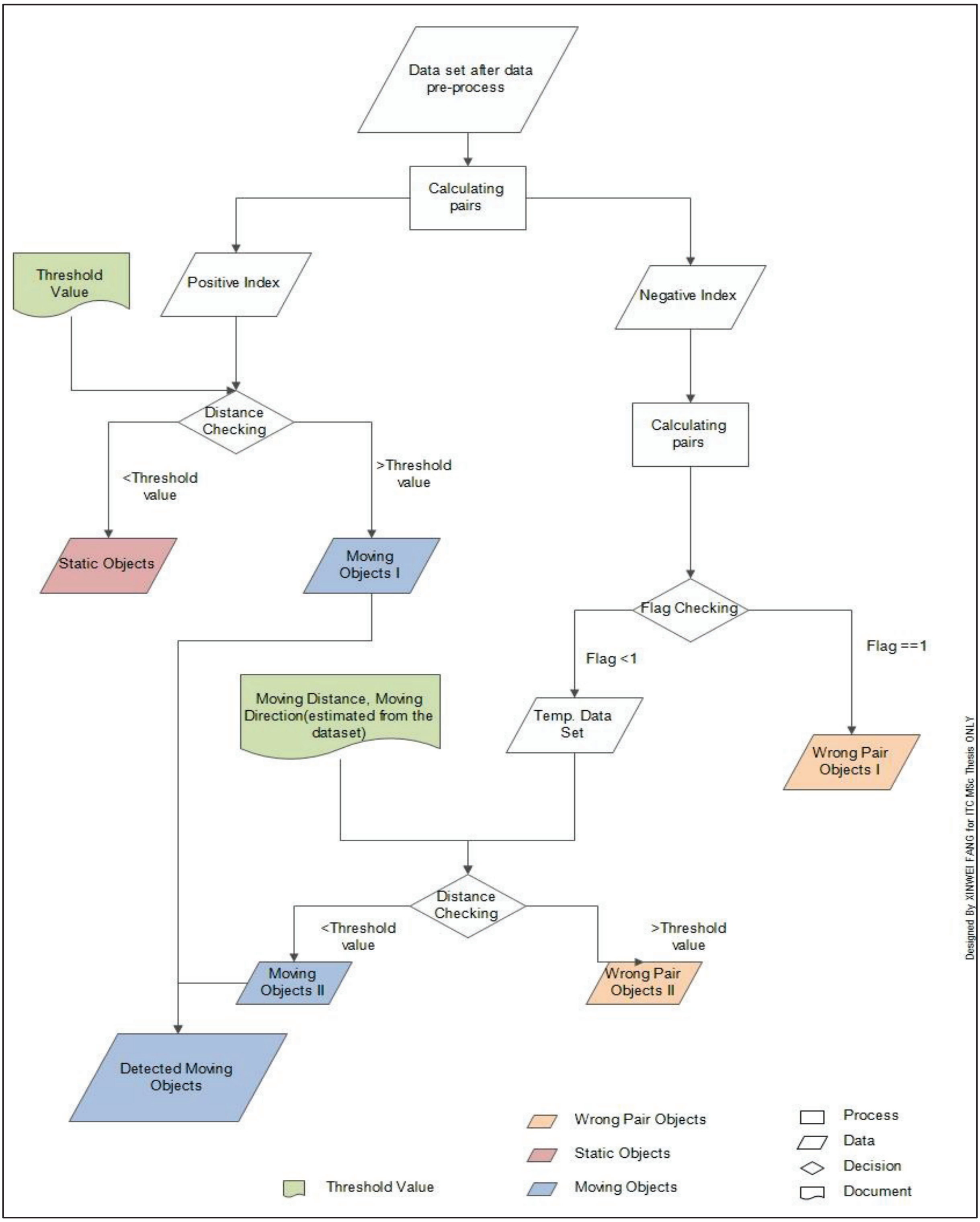


Figure 3-4: A rule-based method for detection of moving objects

The Figure 3-4 describes the rule based method for the detection of moving object detection. The input data are from two sensors which have been geo-referenced and pre-processed. It contains the information of X, Y and Z coordinates and segmentation number of each object segments.

3.3.1. Point and segment pairing

The closest distance between two points is used for finding a pair, under the assumption that the two closest segments from two sensors belong to one object. The assumption was made according to the scan property of mobile-based LiDAR and '2-second driving rules'. Based on the document from Conference of European Directors of Roads (road_safty, 2010), the safe distance between vehicles is suggested. In Netherlands, the 2 seconds rule is applied to all cars to avoid accident. The rule regulates that the rear car should keep the moving distance of 2s at the moving speed from the front car, where 1 second is for reaction time and 1 second is for breaking time. Moreover, with consideration of the property of mobile laser canning (Chapter 2.2), if a static object located far away from the sensor platform in cross road direction, the time interval between the coverage of two sensors would be longer than the time interval of a static object which located close to the sensor platform. If assuming the maximum distance between an object and the sensor is 10m in cross road direction, then 1.5s is expected to be a maximum time interval of static object that is between two sensors coverage (Figure 3-5: Sensors coverage).

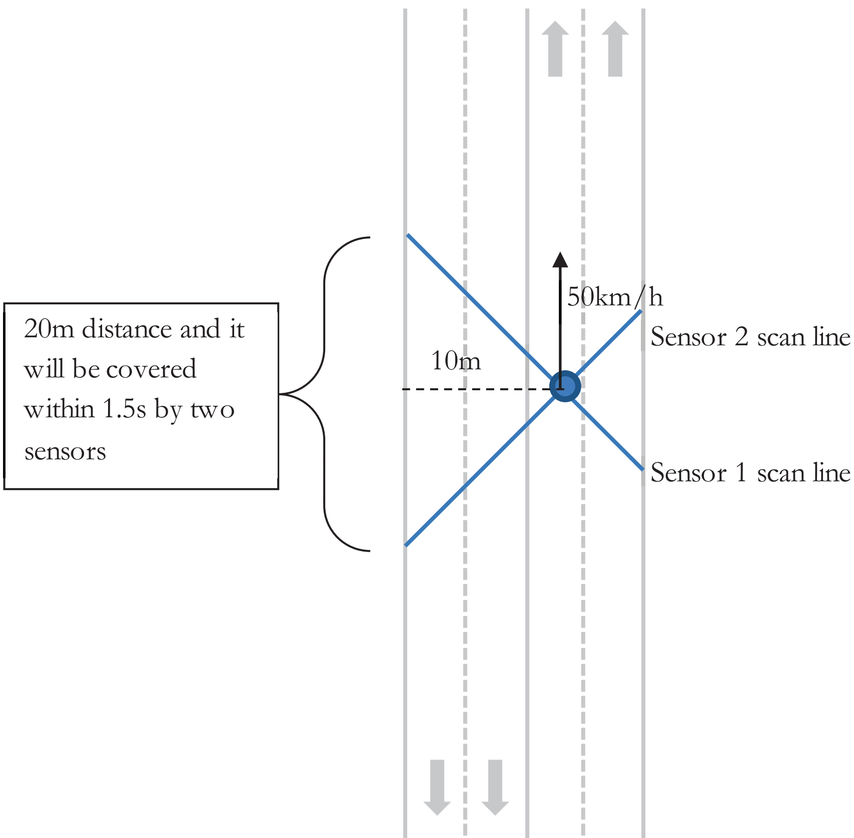


Figure 3-5: Sensors coverage

The maximum distance that a moving object can move between two sensor acquisitions, in theory, is infinite. It depends on the relative speed between the moving object and sensor platform. The distance between two sensors coverage is known according to the property of LiDAR and the platform, if the relative speed between two objects is close to zero, then the time of travelling would be close to infinity. With the infinite traveling time, the distance between two segments would also be infinite. However, in this study area, by visual examination, no extreme distances were found. Therefore, ten meters is selected to be the upper boundary of maximum distance between two moving segments. The value is obtained by analyzing distance of sample pairs for moving objects statistically, which can be found in Figure 4-18.

Under that assumption, the closest pairs are established. The pairs are first conducted in the point space, each point from one dataset is seeking its closest point in another dataset and the distance between two closest points is recorded. When all the points from one sensor have found their closest point in the second sensor, the points are aggregated with their segmentation number into an object. The closest pair of segments is determined according to those closest point pairs aggregated with the segmentation numbers. The distance between the closest pair of segments are calculated by averaging the distances of the closest point pairs. Three types of objects are expected in the dataset after the pairing, which are static object pair, moving object pair and wrong object pair.

- Static object pair: The closest pair of segments are found and distance between two segments is close and smaller than a certain value.

- Moving object pair: The closest pair of segments are found. 1) The distance between two segments are large (greater than a certain value and smaller than a certain value) and they follow a certain moving direction (cars and bikes) or 2) the distance are small but large than a certain value (pedestrains).

- Wrong object pair: The pairs did not belong to either static object pair or moving object pair.

In the point space, when calculating the point pair, the positive and negative index are designed to determine if a point pair can be formulated to be the closest segment pair. Each point from first sensor is looking for the closest point in the second sensor, then, the point found in the second sensor is in turn looking for its closest point in the first sensor. If the returned segmentation number in the first sensor is the same segment which contains the initial point, the positive index is returned. It means the closest point pair is found and no other pairs are found within this distance range. Otherwise, the negative index is returned. After assigning the index to every point, the segment pair can also be checked if two segments are the closest segment pair to each other by aggregating the points with their segmentation numbers.

3.3.2. Positive index

If one segment contains more positive index points than negative one, it means the closest segment pair is determined. On the contrary, if numbers of negative index point are greater than positive one, then, the segment pair is not the closest segment pair for both of segments. According to the analysis in chapter 3.3.1, if the closest segment pair is found, they have high confidence to be the same object or same type of object (moving or static). The segments with positive index are the closest pair which have high confidence to be either static objects or slow moved moving objects because two segments stay close. Moreover, the distance between two paired static segments is smaller than the one of the moving object. Hence, by setting a threshold value of paired distance is able to separate static objects and slow moved objects. Furthermore, as the index value is conducted in the point space domain and the segments are identified in the objects domain, the evaluation of aggregation of points in every segment is needed. Numbers of negative index point and positive index points in each segment are counted and analyzed, as percentage of positive index points in one segment may help to separate the static objects out of other objects. In addition, the threshold value of distance to separate static object and slow moving object are also evaluated and selected. The procedure is discussed in the experiment at chapter 4.4.1.1. The slow moved objects are expected to be detected.

3.3.3. Negative index

Similarly, the segments with negative index are most likely to be either wrong paired objects or fast moved objects as those segments pair are not the closest pair. Therefore, after excluding the points from the static object and slow moved object segments, the points are paired again. If a segment still contains more negative index points in the second point pairing, then, this segment has high confidence to be wrong object pair because no closest pair is determined. However, if the second point pairing gives the more positive index points in one segment, the segment is further analyzed to distinguish if it belongs to moving objects or not.

As the slow moved objects have been identified in the positive index segment, including pedestrians, the fast moving objects are identified here. The fast moved objects are mainly bikes and cars, which follow the moving direction and have high moving speed. Therefore, with consideration of LiDAR sensor geometry and speed limited in the urban area, the following constrains to separating fast moved object in negative index segment are designed.

- Distance of moving: the distance between two segments should be smaller than a certain value.

The distance of upper boundary for moving object in this study area are determined to be 10m by statistical analysis. Therefore, if the distance of two paired segments is greater than 10m, the segment is not considered as moving object.

According to the calculation, there was a one second time interval between two sensors coverage. As the bikes and cars are moving with the certain direction, they are expected to move along the direction of the road when they were captured by two sensors. Hence, the direction of segments orientation is used in detection of moving objects.

- Direction of moving: the orientation of line segment which connected two centers of paired segments should be within certain degrees of the road orientation.

Indicated in the Figure 3-6, if one center of the segment is defined as origin, the center of the paired segment only located in the front or back section are considered to be moving objects.

The cars and bikes, within the time range 1.5s (the time interval between two sensors coverage) are considered to be moving consistent which is along the road direction. The orientation of moving is divided into four sections front, back, left and right showing in Figure 3-6. With the cooperation of the scanning

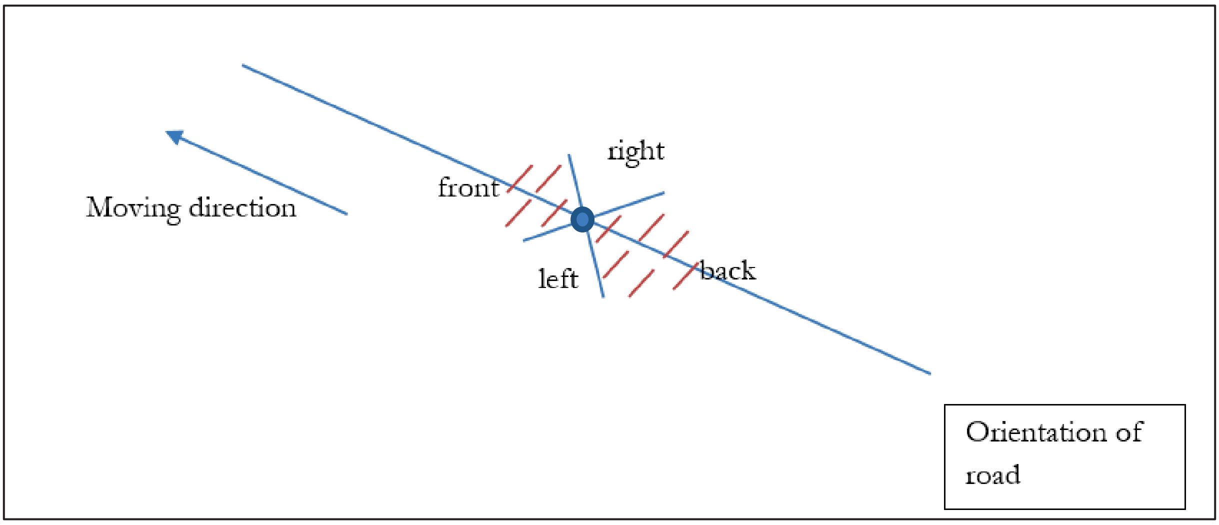


Figure 3-6: Orientation of moving objects

property (moving distance) and moving object property (orientation), the fast moving objects are expected to be separate from the segments which has more negative index points.

The final moving objects are the combinations dataset from moving object in both positive and negative datasets, where both slow moving and fast moving objects are expected to be detected.

3.4. Evaluation method

As no reference data of temporary objects were provided from the survey company, the reference dataset is generated by manual inspection. Therefore, the evaluation can only be carried out in object domain. After data processing procedure, the numbers of moving objects are evaluated by checking if all the moving objects can be identified. If all the moving objects remains, the ground truth is generated based on that dataset. The ground truth dataset is generated by labeling a number on the different types of objects (1- static object, 2 - moving objects and 3-wrong pair objects). The confusion matrix is applied to evaluate the result where the same method were also introduced by (Heipke et al. (1997)). The error assessment table is showing below in Table 1: Error assessment.

Table 1: Error assessment

	Ground truth TRUE	Ground truth FALSE
Result TRUE	True Positive (TP)	False Positive (FP)
Result FALSE	False Negatives (FN)	True Negatives (TN)

- Completeness :

$$\mathbf{Completeness} = \frac{\mathbf{TP}}{\mathbf{TP + FN}} \qquad \text{Equation 3}$$

The completeness is describing the numbers of interested objects in percentage are extracted in respected to the reference dataset. The range is from 0 to 1.

- Correctness :

$$\mathbf{Correctness} = \frac{\mathbf{TP}}{\mathbf{TP + FP}} \qquad \text{Equation 4}$$

The correctness is describing the numbers of objects of interest in percentage are correctly extracted in respect to the reference dataset. The optimum value for correctness is 1 and range is from 0 to 1.

- Accuracy :

$$\mathbf{Accuracy} = \frac{\mathbf{TP}}{\mathbf{TP + FN + FP}} \qquad \text{Equation 5}$$

Accuracy considered both correctness and completeness, and it explains the overall quality of the extraction. The range is from 0 to 1.

4. EXPERIMENTS AND RESULTS

This chapter describes the study area, dataset, experiment approach and the results. In the experiment, the segmentation methods were implemented in C++ and the application program was provided by the Faculty of Earth Observation (ITC), University of Twente. PCM software developed by George Vosselman (ITC, university of Twente) and CloudCompare were used for point cloud labelling and visualization. The process for trimming of irrelevant objects and the detection of moving objects were implemented in the MatLab. The matlab code for calculating the minimal bounding box was adopted from (Korsawe, 2008). The result of LI (2013) Msc research (ITC, university of Twente)were also used for estimating the width of the road. In this chapter, the dataset and study area is introduced firstly in 4.1. Then according to the work flow sequence, each experiment and result is discussed in chapter 4.2 to 4.4.

4.1. Data set and Study areas

The dataset was carried out by a German company TopScan in December, 2008 by using the Optech Lyn Mobile Mapper system Figure 4-1 (Optech, 2009).

LYNX MOBILE MAPPER	V100
Parameter	
Number of lidar sensors	1-2
Camera support	Yes, 2 x 2 Mpixel
Maximum range	100 m, 20%
Range precision	±8 mm, 1 σ
Absolute accuracy	±5 cm (1 σ)[1,2]
Laser measurement rate	100 kHz
Measurements per laser pulse	Up to 4 simultaneous
Scan frequency	150 Hz
Scanner field of view	360° without obscurations
Power requirements	12 VDC, 30 A max. draw
Operating temperature	-20°C to +40°C (extended range available)
Storage temperature	-40°C to +80°C
Laser classification	IEC/CDRH Class 1 eye-safe
Vehicle	Fully adaptable to any vehicle

[1]Assumes good GPS quality.
[2]Accuracy may be improved via post-processing techniques.

Figure 4-1: LYNX Mobile Mapper Specification

The scene was scanned in the city of Enschede, Netherlands. Two rotating scanning sensors were amounted on the top of the vehicle at 45 degrees angles between the central lane where two sensors were perpendicular to each other (see Figure 2-4). The vehicle drove at 50km/h and scanned 20km road. The strip overview is indicating below at Figure 4-2.

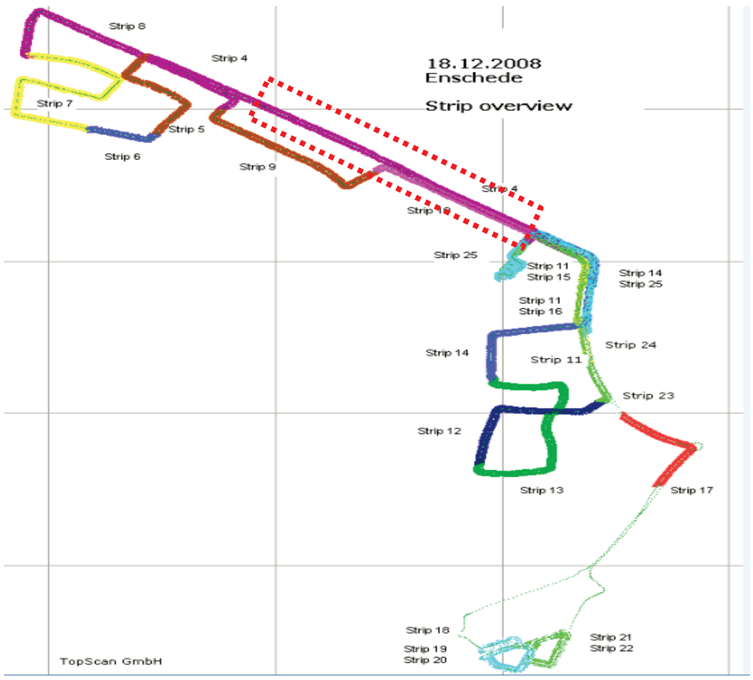


Figure 4-2: Strip overview

The part of the strip_4 which was bounded within the rectangle was used in my study.

4.2. Data process

4.2.1. Surface growing

The parameters selection of surface growing was critical for final segmentation result. In addition, with the consideration of the computational cost, it was impossible to test the parameters for entire dataset. Therefore, the dataset was cut into smaller tiles which was defined by the size $100m \times 38m$ $(length \times width)$ to efficiently test the parameters selection. Three tiles from each sensor, 6 tiles in total, were selected to test the surface growing parameter. The setting value from LI (2013) was used as the initial value. In this study, the surface growing values were fixed by adjusting parameters to achieve the most optimal result with consideration of over-grown, under-grown and preservation of moving objects. The surface growing radius of 0.5m and the maximum distance to surface of 0.3m were selected to segment the entire dataset. The Result showed that all 60 moving object point clusters were able to be visually recognized.

4.2.2. Ground, Off-Ground and building facade removal

Once the dataset was segmented into planar segments, the ground and building facade were detected and removed. By analyzing the planar size, the $20m^2$ and $50m^2$ were selected to be a threshold value to detect the ground surface and building facade surface respectively (LI, 2013). If the size of horizontal planar surface was greater than $20m^2$ and below the 3D trajectory, the segment was considered as the ground segment and removed. Similarly, if the size of vertical planar surface which connected to the ground segments greater than $50m^2$, the surface was recognized as a building facade and removed. The 2D outline of the ground was also generated after the ground surface identified. The segments which fall out of those areas and the segments which did not connect to the ground segment directly were discarded. The segments which did not fulfill above condition were kept and processed in the connected component segmentation. The numbers of moving objects after ground and facade removal had been compared with the number that derived from the raw dataset. Sensor 1: 30 and Sensor 2: 30 moving object segments were all preserved. However, in sensor 1, one moving segment and one static segment were connected together because of the over-growing of the surface. In sensor 2, three segments representing static cars were connected together because of the over-growing of the surface showing. They are shown in Figure 4-3 and Figure 4-4.

Figure 4-3: Surface over-growing in sensor 1

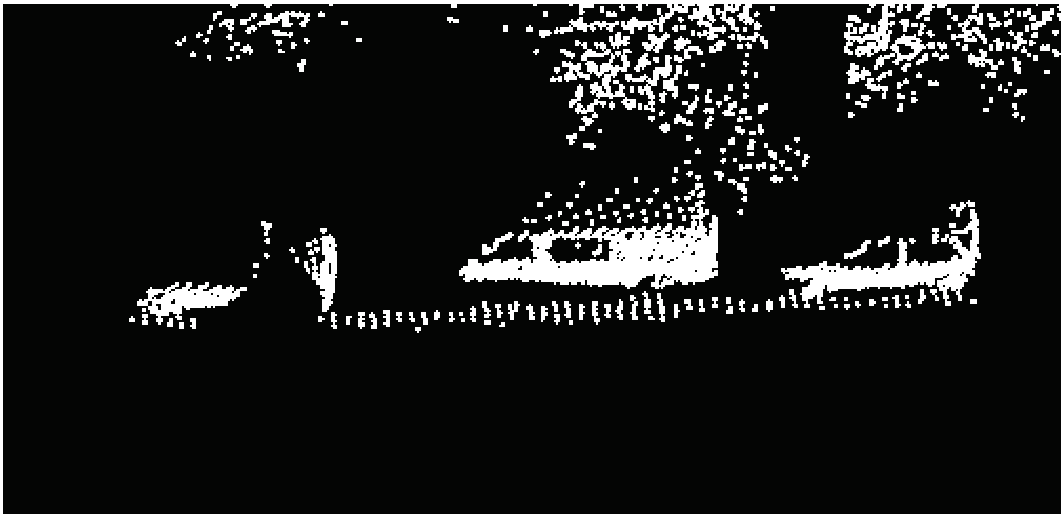


Figure 4-4: Surface over-growing in sensor 2

4.2.3. Connected component Segmentation

After the removal of ground and building facade, the points were segmented into objects by connected component segmentation, since the moving objects are normally considered to be groups of the closest points (Pu et al., 2011). The minimum number of the points that formulate points into one segment was set to minimum as the smallest numbers of point for moving object segment was unknown. The result of threshold value selection was described in Figure 4-5 and Figure 4-6.

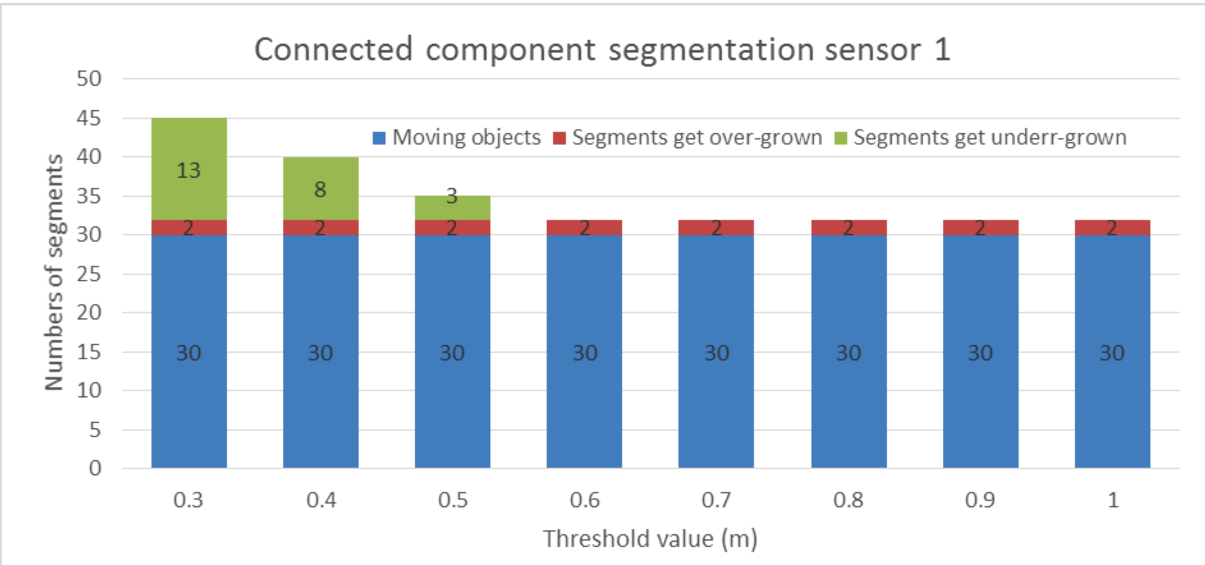


Figure 4-5: Result of connected component segmentation (Sensor 1)

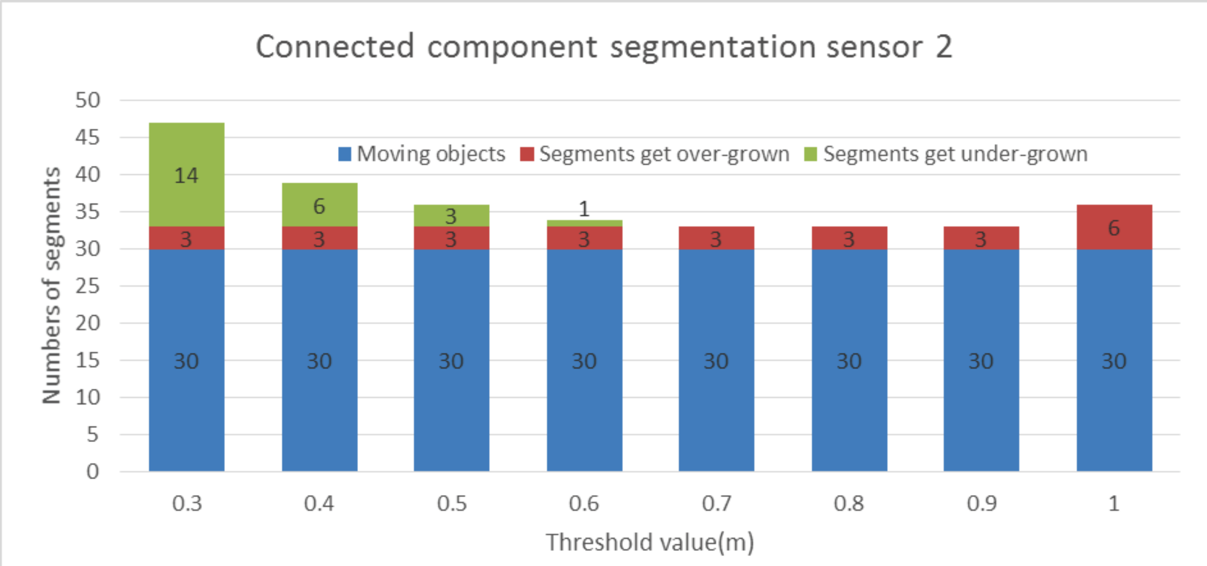


Figure 4-6: Result of connected component segmentation (Sensor 2)

In the Figure 4-5 and Figure 4-6, the value of threshold indicates the distance between the points. The figures showed that all the moving objects segments were successfully segmented in all condition. However, as the increase of the threshold value, the numbers of segments getting over-grown increased. On the contrary, with the decrease of the threshold value, the numbers of objects which were under-grown were increasing. Moreover, as the ground over grown in the surface growing segmentation process, the over-grown segments were always occurred in both dataset. The threshold values between the 0.7 to 0.9m in the dataset were the most optimal values as they have minimum numbers of over-grown segment and under-grown segment occurred. The final decision was set as 0.9m, where the numbers of segmentations are the smallest meaning that small segments were merged, such like the small leave segments were segmented into the tree segment. The result is shown below in Figure 4-7.

Figure 4-7: Result of connected component segmentation

4.2.4. Bounding box data trimming

Once the points were segmented into segments, the minimum bounding box was applied to each segment. The code used for calculating the minimum bounding box in Matlab was developed by (Korsawe, 2008) and covered by the BSD license. The bounding box fitted the smallest cube in respect to the volume to each segment. The multiple values were obtained (volume, corner points, etc.) and they were used in the further process.

- Volume: if the volume of bounding box was larger than 122 cubic meters, the segment was removed.

The volumes of the bounding box were directly returned from the function of the minimum bounding box and it was used for constraining the volume size of the segments. The Figure 4-8 shows the segments which volume were greater than 122 cubic meters. Most of segments were the trees and building parts. And no moving objects were included.

Figure 4-8: The segment volume greater than 122

- Long edge: if the longest edge of the bounding box was longer than 15 meters, the segment was removed.

As the bounding box was defined by the corner points and all adjacent edges of the bounding box were perpendicular to each other. Hence, the edges of bounding box were calculated from the distance between corner points. The value of long edge was determine by looking for the maximal value of the bounding box edges.

Figure 4-9: The segments long edge greater than 15m

In the Figure 4-9, the segments which were detected by long edge constrain were mostly house barriers and no moving objects were wrongly removed.

- Maximum height: if the height of the segment was greater than 4 meter, the segment was removed.

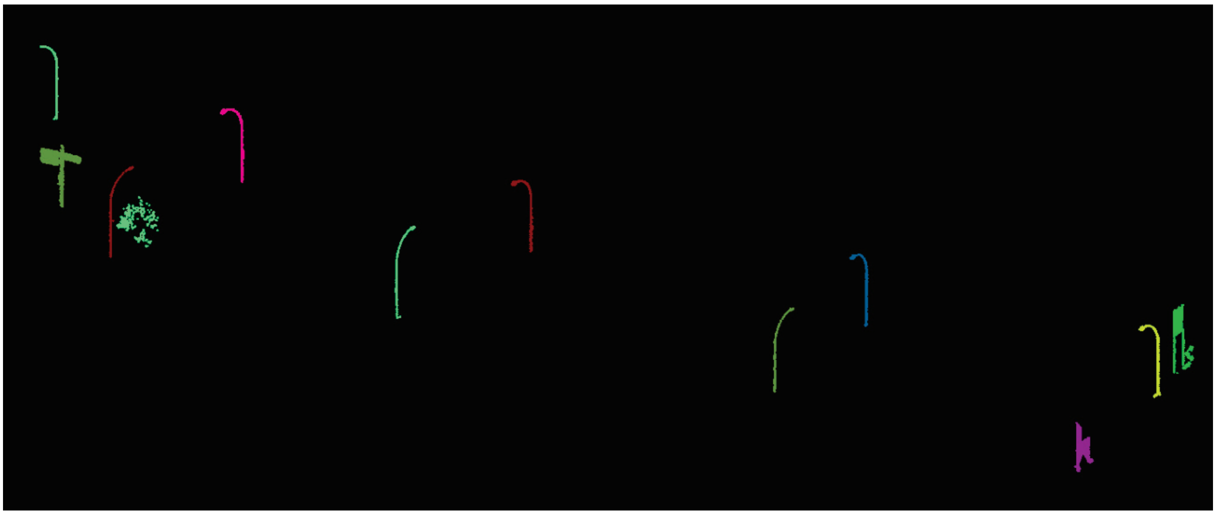


Figure 4-10: The segments height is over 4m

The value was determine by calculating the distance between the maximal Z value and minimal Z value in each segment. For the objects that height were greater than 4 meters were removed. In Figure 4-10, mainly pole-like objects were detected and no moving objects were wrongly identified.

- Minimum height: if the height of the segment was below 1 meter, the segment was removed.

The value was obtained by calculating the difference between the maximum height value and minimum height value of each segment. The minimum height constrain removed small segments which height were less than 1m. No moving objects were removed.

- Absolute height: if the lowest point of a segment was above one meter from the ground, the segment was excluded in the following process.

The value was obtained by comparing the minimum height with the local elevation. The local elevation was estimated from 3D scan trajectory. The center of each segment was looking for the closest point in the 3D trajectory. The Z value of corresponding 3D trajectory subtracted the height of sensor would give the value of local elevation. The absolute height was calculated by minimal Z value of the segment subtracting the local elevation. This constrain removed the objects which were floating in the air. Like higher part of lamp-post, building facade or trees. Moreover, no moving objects were removed.

- Minimum numbers of points in one segment: if the number of points for a segment was smaller than 40, the segment was discarded.

The small segments were removed at this step. By visualizing and finding the smallest numbers of points in a moving object, the threshold was set at 40. All the small segments were removed and moving objects were preserved.

The final result after bounding box data trimming procedure, 395 segments in sensor 1 and 382 segments in sensor 2, were successfully removed and all 60 segments of moving object from two sensors were preserved (Sensor1 : 30, Sensor2: 30). The summary of bounding box data trimming showing that the numbers of segments were removed after each process as depicted in Figure 4-11.

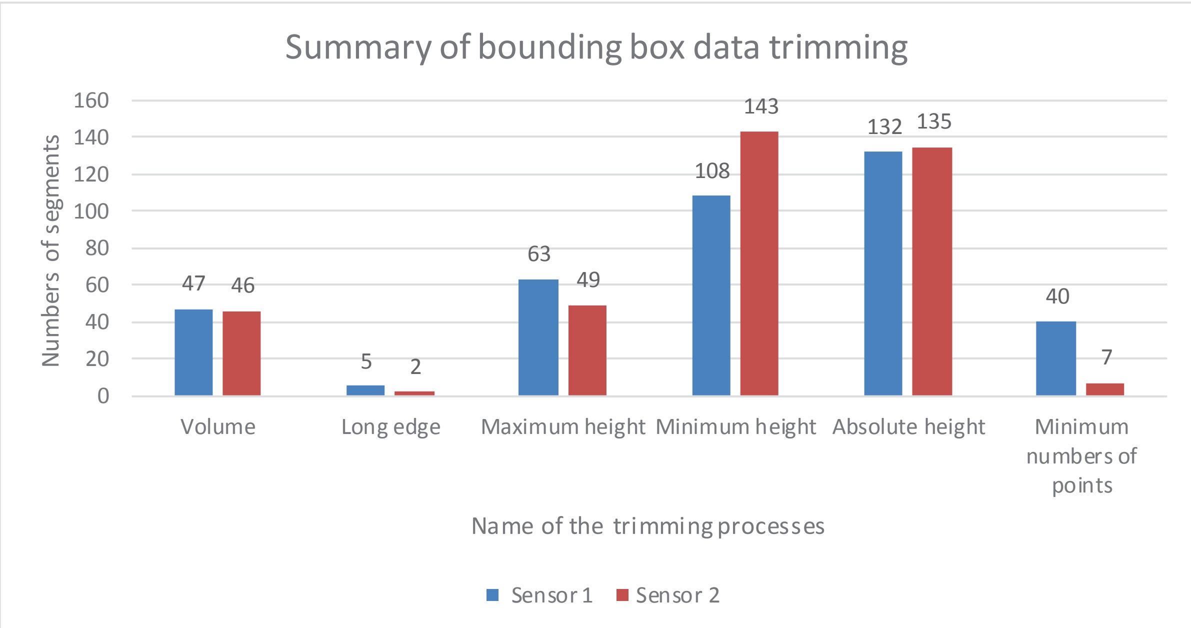


Figure 4-11: Summary of bounding box data trimming

As shown in Figure 4-12, some building facades which appeared on the one side of the main road still remain. That was because when determing the road width in the surface growing segmentation, then it was considered to be symmetric in respected to the scan trajectory. That process resulted in the inaccurate estimation of the road width and the resulted redundant building segments remains. Moreover, those redundant segments increased numbers of wrong paired object segments. Therefore, further data trimming to determine the proper road width and extract the temporary objects is to be explored.

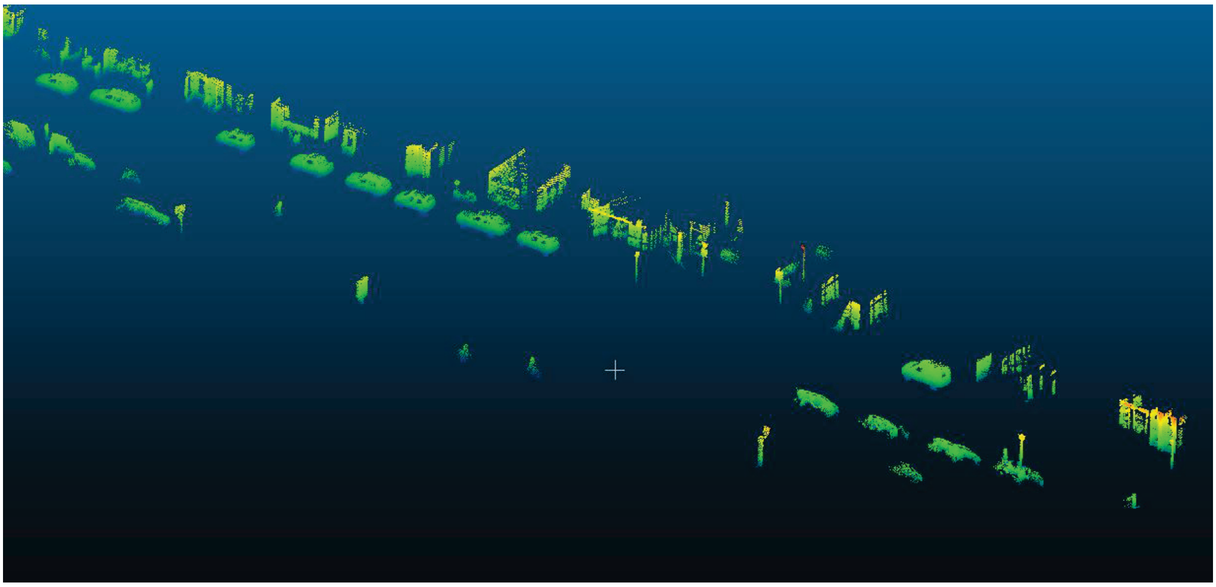


Figure 4-12: Result after bounding box trimming

4.2.5. Lamp-post trimming

The most common moving objects in the dataset were cars and bikes. Those objects usually present on the main road and bike path. Knowing the boundary of the main road, the total road width including bike path and side walk could be roughly estimated and the segments falling in this areas can be temporary objects with high confidence. According the previous research, the lamp-post were a good indication for estimating the boundary of the main road.

Figure 4-13: Lamp-post in study area

The result of pole-like object extraction (LI, 2013) was shown in Figure 4-13. The labels were assigned to each types of pole-like objects, which were used for separating the lamp-post from other pole-like objects in the study area. The accuracy of lamp-post extraction was 84% in completeness and 99% in correctness.

As correctness of the lamp-post extraction was very high, there was no need to exam if the label of lamp-posts were correctly assigned. However, in view of completeness, as only 84% of all lamp-posts were detected, it was better not to connect the lamp-post location sequentially to establish the road boundary. If forcedly connecting all the locations of lamp-posts, the result might be biased. But as the most lamp-post were successfully detected and the orientation of the road segment were consistent, the line fitting to each side of the lamp-posts were applied, for which the road boundary were estimated by the large numbers of lamp-posts observations.

In Matlab program, the locations of those lamp-posts were transferred into 2D as they were only used for constraining the width of the road. Those locations were separated into 2 datasets by the scan trajectory in 2D. According to the moving direction of the sensor, the lamp-post was grouped into left and right side of the scan trajectory. Moreover, the orientation of study areas were consistent and was a straight road segment. Therefore, two boundaries of the road on each side were derived by linear equation fitting of all the locations in each side of the road. The result of equation fitting is shown in Figure 4-14.

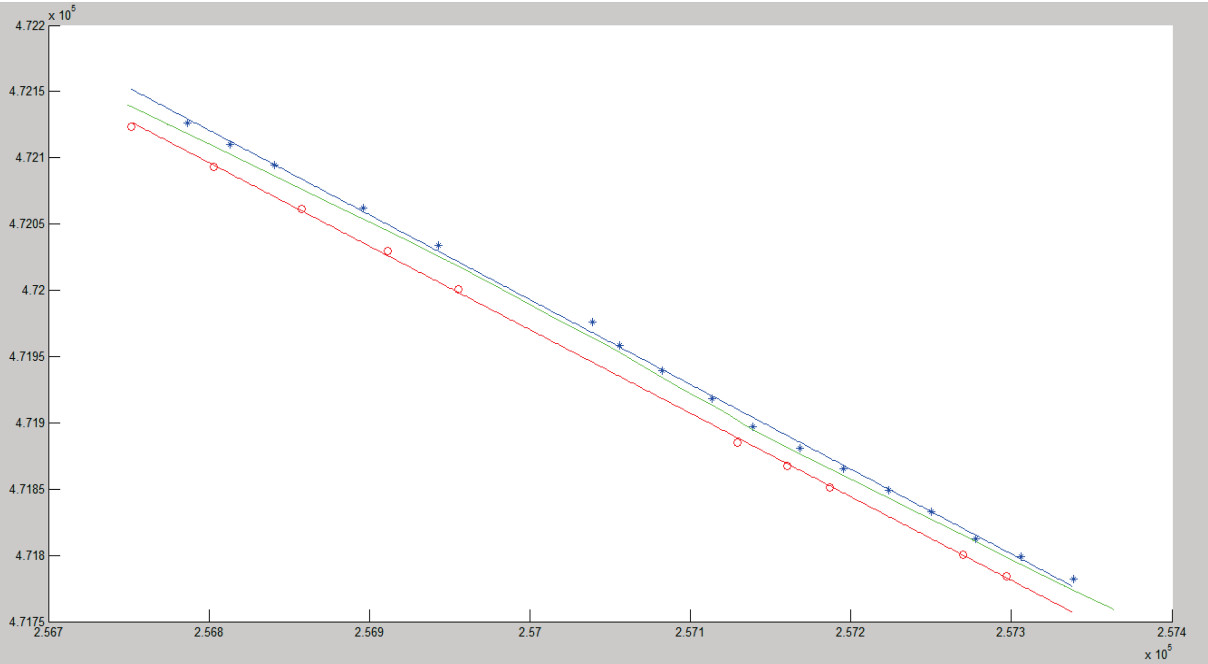


Figure 4-14: Result of lamp-post fitting

The red (left of scan trajectory) and blue (right of scan trajectory) marks indicated the location of detected lamp-posts. The blue and red line were the linear lines fitting to those locations respectively. The green line in the middle was scan trajectory. As the boundary of the main road were represented by the two linear equations, the extension of the road in order to include the bike path and sidewalk was done by shifting the position of the equations. In this case, two meters width was extended to each side of the boundary.

In trimming process, if the center of the segments falling between the boundary lines, the segments were preserved because they were more likely to be temporary objects. If not, the segments were not considered as objects of interest and removed. The segments result after the lam-post trimming is shown in Figure 4-15, with the removal of segments falling out of the road and all the 60 moving segments in two sensors remaining.

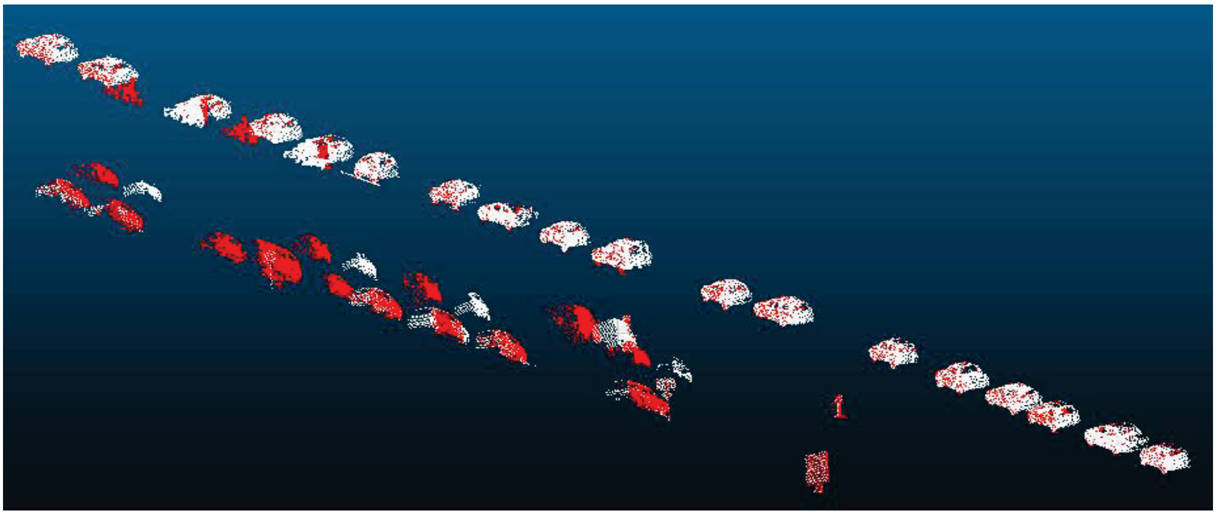


Figure 4-15: Result after lamp-post trimming

4.3. Ground truth dataset

Once the data pre-processing finished, the dataset was evaluated manually by checking if the previous process removed any moving objects. As no moving objects were removed, the ground truth dataset was generated by labeling the dataset into 3 labels, 1-static objects, 2-moving objects and 3-wrong pair objects. The labels were given based on the shape of the segment and other surrounding contexts. According to the property of vehicle-based LiDAR, each object was expected to be found in two sensors. However, due to the occlusion or correctness and completeness of the segmentation method, the object might only be found in one sensor. For the ground truth data labeling, if the segment was temporary object, the segment was labeled as static objects or moving objects even though the object only appeared in one sensor. But for the other permanent objects, like traffic sign, ground segment, if the object only appeared in one sensor, the object in that sensor was labeled as wrong pair.

With the ground truth dataset, the evaluation result for the detection of moving objects was derived. Moreover, every threshold value was tested and selected according to the most optimal results. They are derived by comparing result with ground truth data. Figure 4-16 showed the numbers of objects labeled in the ground truth dataset.

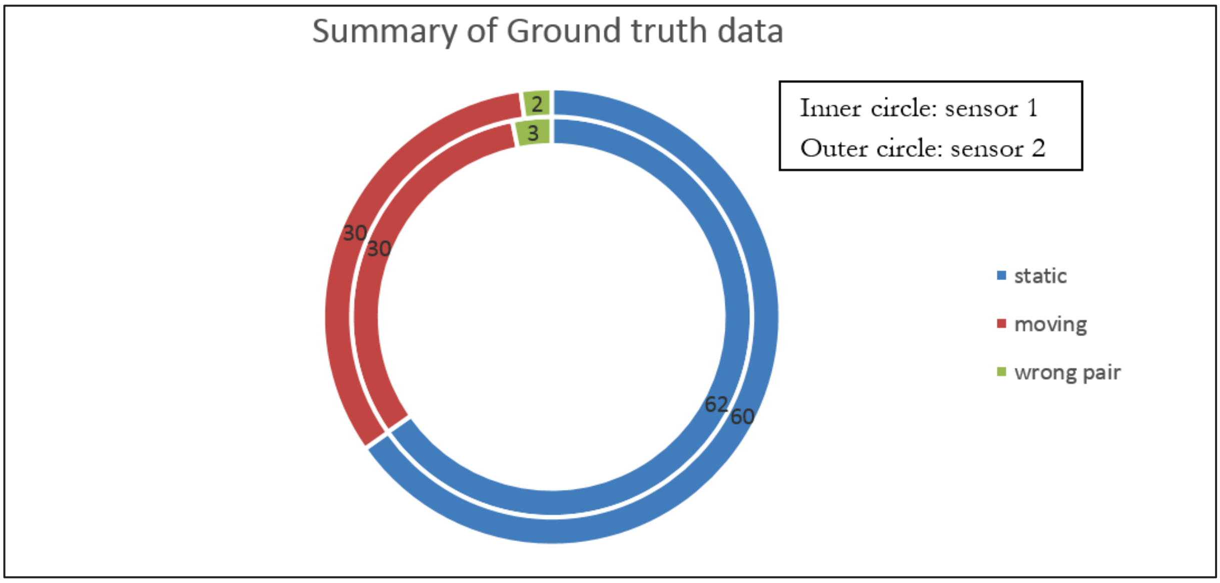


Figure 4-17: Summary of the ground truth data

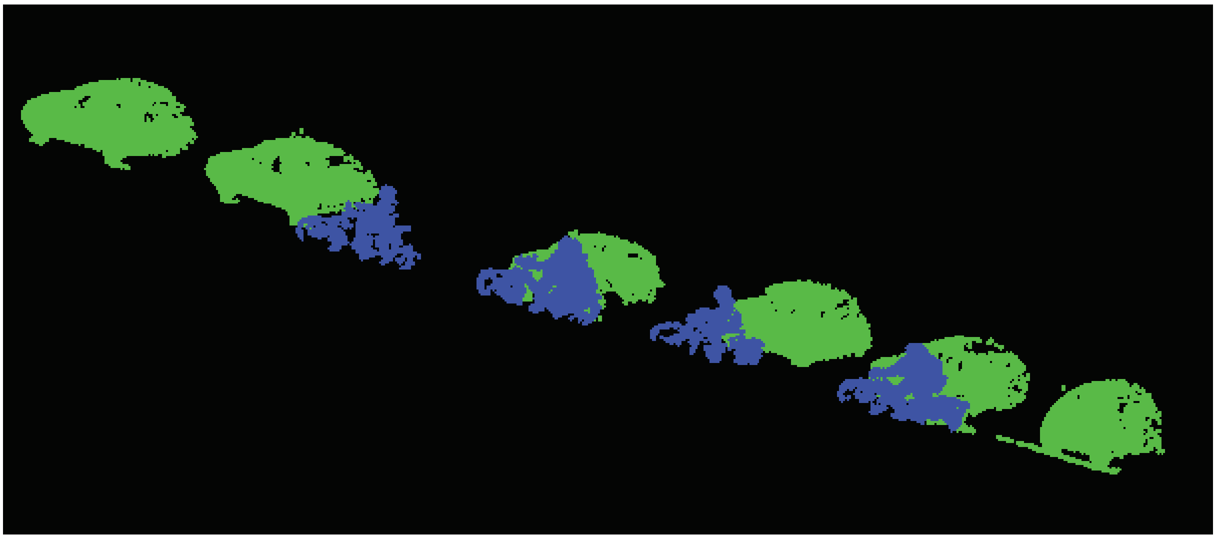


Figure 4-16: Ground truth dataset

The Figure 4-17 showed the ground truth dataset. Three colors were assigned to each object, 1-green (static), 2-blue (Moving) and 3-yellow (Wrong), represent each type of objects respectively. The ground truth was done by manual labelling according to visual comparison of other contexts.

4.4. Result of finding closest point pairs

According to the rule based classification algorithm Figure 3-4, detection and removal static objects were the first and the easiest step because the distance between two segments of one static object from two sensors was relatively small than other two types of objects (moving and wrong pairs object). In the Matlab program, the datasets from both sensors were loaded and saved as two variables which called a base-data and a search-data respectively. By looping all the points in the base-data, each point was looking for the closest point in the search-data, and the distance between two points and index number of search-data were returned. The index number of base-data was known as the value of loop index. Hence, the two points were known being matched. Then, the point found to be the closest point in the search-data was looking for its closest point in the base-data. The '-1' index was defined to make sure the pair was the closest pair. By definition, if the closest point that was found in the base-data belong to the same object which contained the initial point, the distance and index were returned. Those returned index was called positive index. If it was not, then, the distance and '-1' index were returned. The positive value was returned if the correct closest pairs of segments were found, and negative index value indicated the points which may not have the correct closest pair.

4.4.1. Object space domain

4.4.1.1. Positive index points

Once all the points had been assigned an index, the process was applied in segment domain, where all the points belonging to one segmentation number were grouped and analyzed together. In Matlab program, the dataset was looped by segmentation number, where points belonging to the same segmentation number were extracted and processed. In addition, the aggregated distance of all the closest point pairs was calculated, which was the fundamental key to separate static objects and moving objects. The manual selected 45 static objects and 24 moving objects pairs were tested to check if the distance was able to separate the static and moving objects. The result were shown in Figure 4-18.

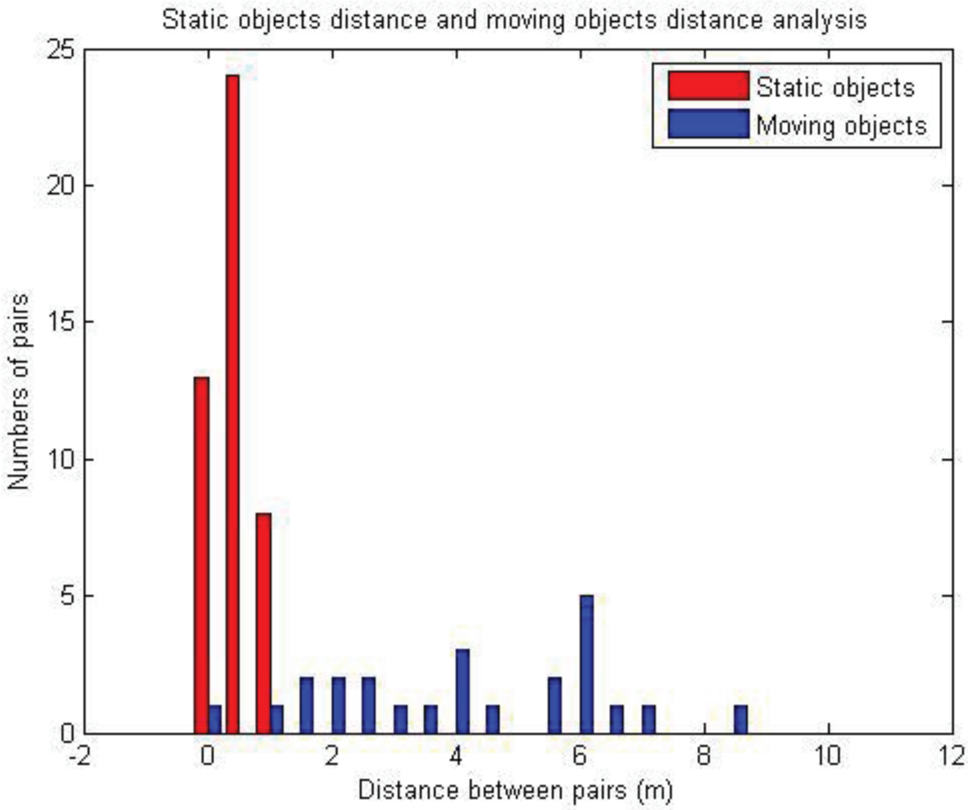


Figure 4-18: Distance difference in static and moving objects

From the Figure 4-18, as the samples of moving objects were too small, no distribution was used to fit the data. Both static and moving object histograms were plotted in the same scaled figure. All the static objects had the distance below 1 meter, where only 2 moving objects were below 1 meter level. Therefore, by using the distance to separate moving and static objects were possible. However, in order to find the most optimal threshold value, more detailed analysis were required.

According to confusion matrix in chapter 3.4, three values, true positive, false positive and false negative of static object extraction, were calculated. The ground truth dataset generated manually was used as reference dataset. As no reference data could be used to assess the ground truth dataset, the true negative value was not considered in this study. Based on the experiment of point index, the distribution of the two indices points in one segment and distance between two segments of one object needed to be analyzed. Therefore, the percentage of positive index point in one segment from 0 to 0.9 (0% to 90%) and the threshold value of average distance from 0.1m to 0.9m were tested. In the automatic evaluated program, three evaluation values (TP, FP and FN) were calculated from all 99 possible combinations and the completeness and correctness were derived from those values. The result is shown below in Figure 4-19, Figure 4-20, Figure 4-21 and Figure 4-22.

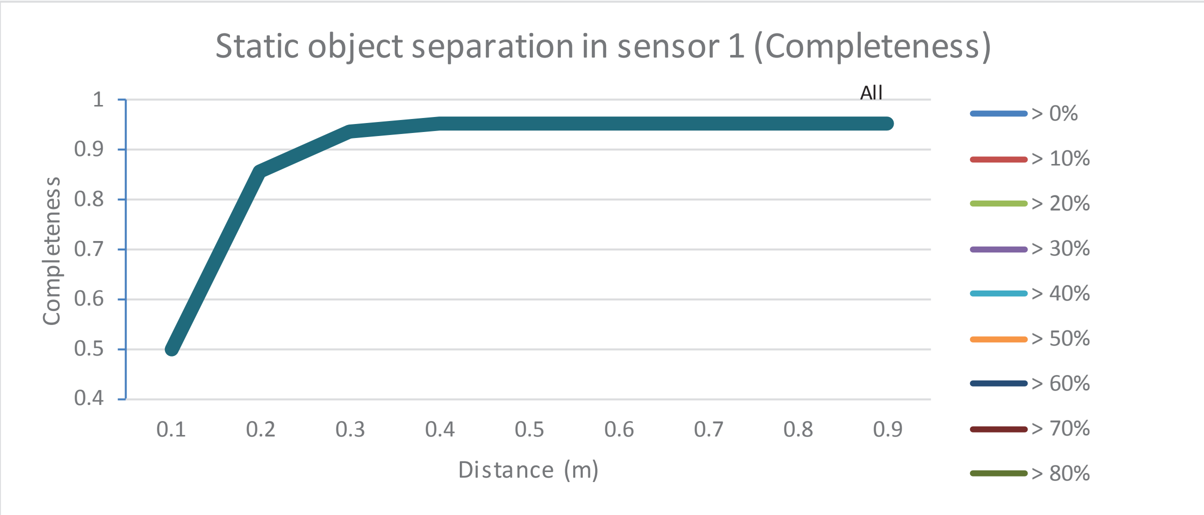


Figure 4-19: Completeness in sensor 1

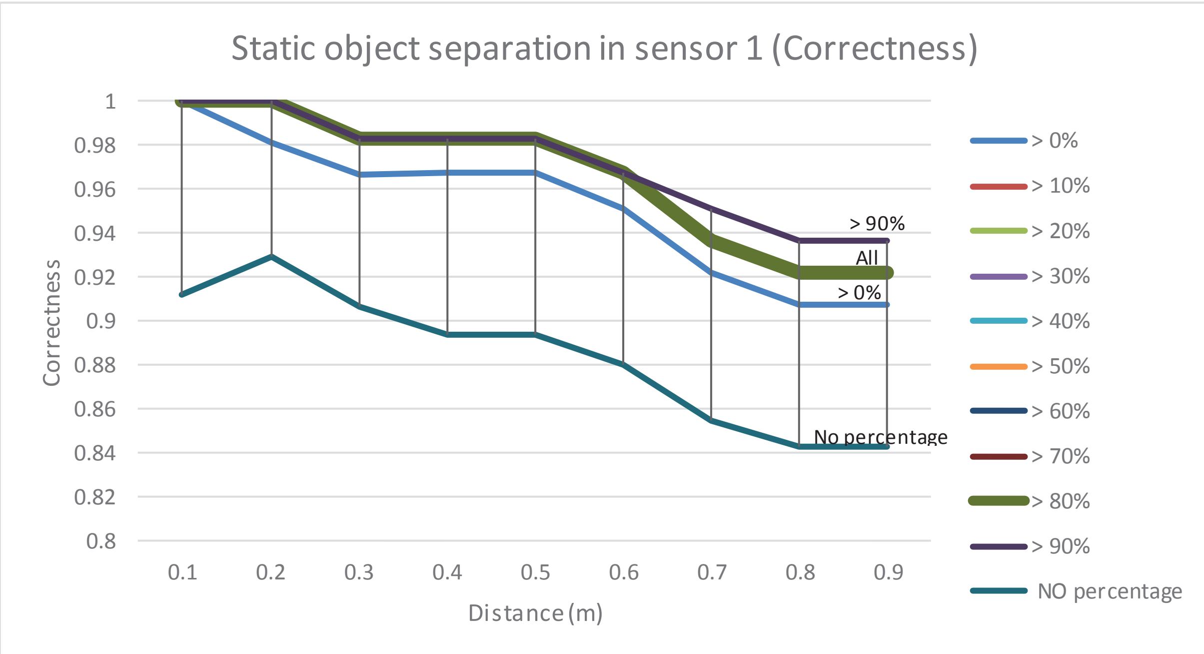


Figure 4-20: Correctness in sensor 1

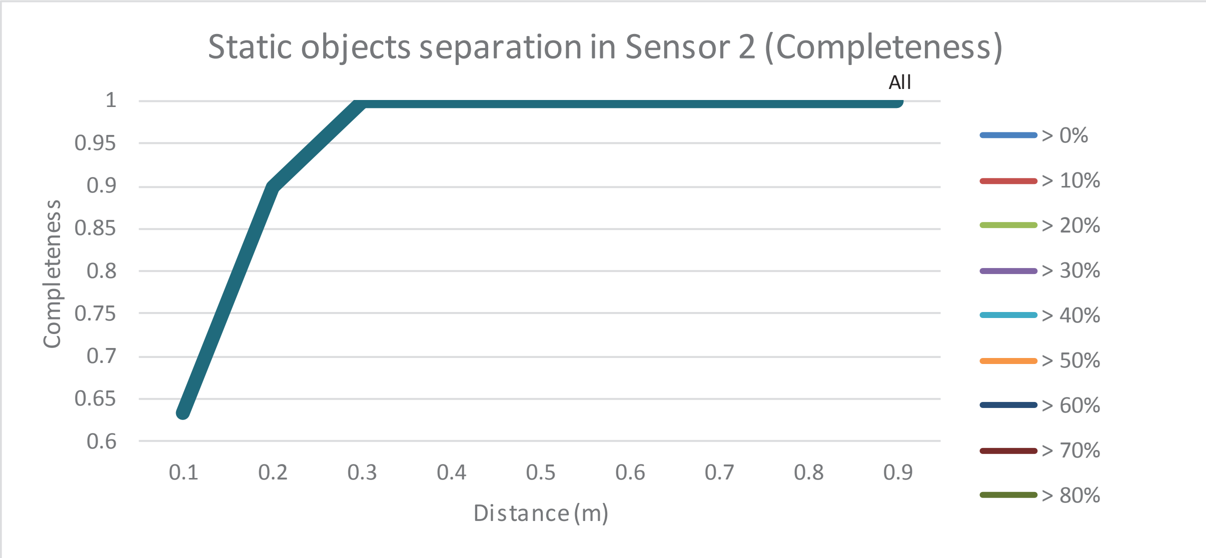


Figure 4-21: Completeness in sensor 2

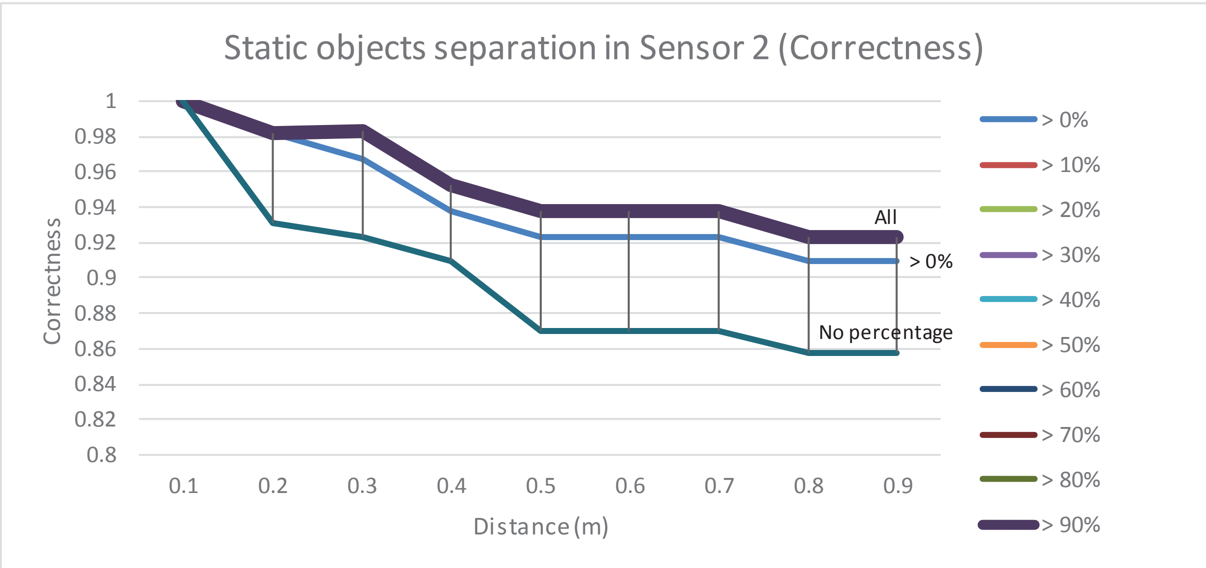


Figure 4-22: Correctness in sensor 2

As can be seen from the Figure 4-19 to Figure 4-22, the percentage of positive index points did not make significant difference in completeness for separating static objects and moving objects in both sensors. However, from the correctness point of view, the inclusion of the factor of positive index made a big difference. The percentage of positive index did not influence correctness if it was above 10%. Hence, we can conclude that the positive index and negative index does help in separating the static objects. Moreover, if value of the percentage of positive index is between >10% and >90%, it did not make a different result. Therefore, the positive and negative index was only used as indication. If a segment contained more than 10% of positive index point, the two segments had high probability to be the closest segment to each other.

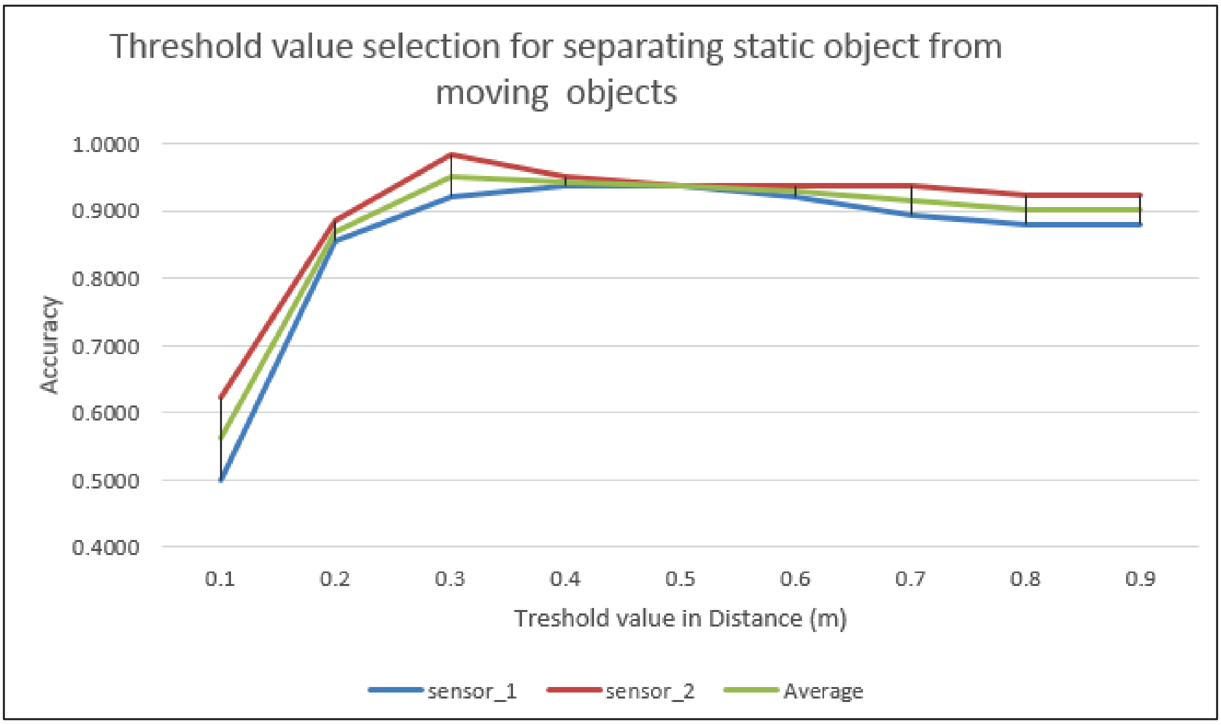


Figure 4-23: Threshold value selection

According to result of experiment, the Figure 4-23 was plotted based on the experiment data. The value of 0.4m in sensor 1 shows the highest accuracy of separating two objects, however, in sensor 2, the most optimal value was 0.3m. Only one threshold value could be applied to both datasets because this value described the tolerant distance in general for separating two objects. Therefore, average accuracy of both sensors were calculated and plotted on the figure in black line. It showed that the maximum overall accuracy was 0.9521 at the distance 0.3m. Hence, 0.3m was selected to be optimal threshold distance for separating the static objects from moving objects (Table 2).

Table 2: Result of extraction static objects

	Completeness	Correctness	Accuracy
Sensor 1	0.93	0.98	0.92
Sensor 2	1.00	0.98	0.98

By using the percentage value and distance threshold value, the static objects and slow moved moving objects were separated in the positive index objects.

4.4.1.2. Negative index points

For the segments which contain majority of negative index were most likely to be wrong paired objects and fast moved moving objects. After excluding the segments which contain more positive index points, the segments which contain majority of negative index were paired again. In the second time of point pair, the distance and positive and negative index were also stored and compared with the first pair. If results of two pairs were identical with two negative index, the segments were assigned to be wrong pairs. On the contrary, if the second pair turned to be positive index, it indicates the correct pair was found. However, those pairs should be analyzed further because the pair may stand for either moving objects pairs or wrong objects pairs. The moving distance and moving direction were considered in the moving detection at this stage (see 3.3.3). The slope value of -0.63, determined from the equation slope of the lamp-post fitting (3.2.4.2 Lamp-post data trimming), was considered to be the orientation of the road. The orientations of each pair were calculated and compared with road orientation. If the segments were located in the front section or back section and distance between two segments was smaller than 20, the segments pairs were recognized to be fast moving objects.

4.4.1.3. Result for detection of moving objects

The final results were the combination of moving objects in positive index dataset and negative index dataset. The result for the detection algorithm was shown at Table 3: Result for the detection of moving objects .

Table 3: Result for the detection of moving objects

	Sensor_1	Sensor_2	Sensor1 + Sensor2
True Positive	28	26	54
False Positive	3	1	4
False Negative	2	4	6
Completeness	0.93	0.86	0.90
Correctness	0.90	0.96	0.93
Accuracy	0.84	0.83	0.84

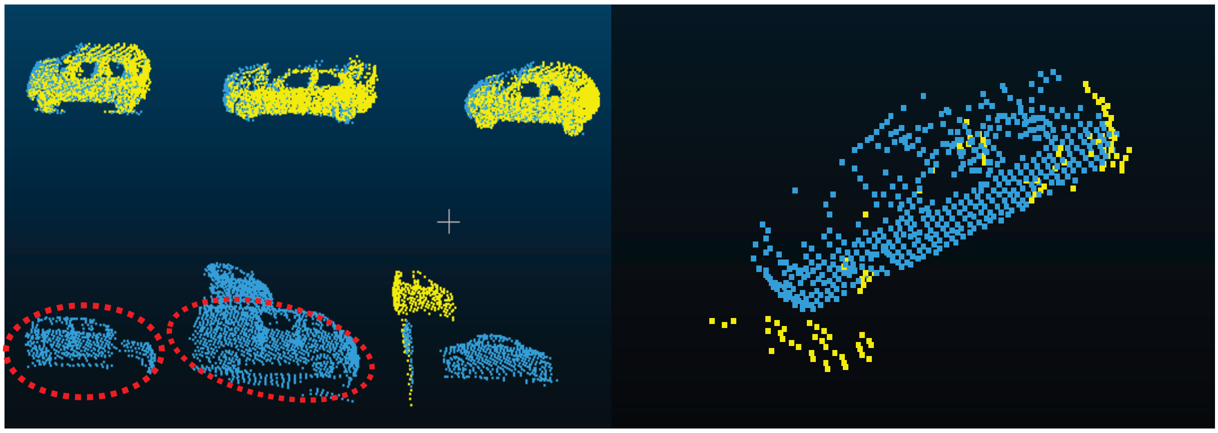
5. ANALYSIS AND DISCUSSION

The detection algorithm can extract the moving object with relatively high value of completeness and correctness. However, in each sensor, there are some false positive and false negative. In this chapter the potential reasons that cause the missing or over detection will be discussed and analyzed. Moreover, the errors which were propagated from the pre-process step may leads inaccurate evaluation of algorithm for the detection of moving objects. Thus, the ideal dataset are generated manually to better evaluate the detection algorithm. The program efficacy and the discussion are also described in this chapter.

5.1. Correctness

5.1.1. Sensor 1 data

In the sensor 1 data, with comparing with other context, three static car segments were wrongly grouped into moving object. They can be seen from Figure 5-1



(a) **(b)**

Figure 5-1: The correctness of sensor 1

The two colors of points indicate the points that were captured from different sensors. The blue and yellow points stand for sensor 1 and sensor 2 points respectively. From the Figure 5-1(a), two cars (with red circle) which belong to static objects were wrongly detected as moving objects because they were found to be the closest segments pair and distance between paired segments was within threshold value. It is because that when applying surface growing segmentation in sensor 2 data, the segments which stand for the same objects in sensor 2 where the segments got over-grown (Figure 4-4) and three segments were connected together.

In the connected component process, the three objects were segmented to be one segment and was removed in the data trimming process.

The Figure 5-1(b) shows another static car which was wrongly identified to be a moving object in sensor 1. From the figure, we can find that the points from sensor 2 (yellow) is very sparse and deformed. It was because that a moving objects was in between the object and the sensor and occluded a part of object. When aggregated the distances of points to distinguish the static objects and moving objects, the value was greater than the threshold value and wrongly identified to be the moving object.

5.1.2. Sensor 2 data

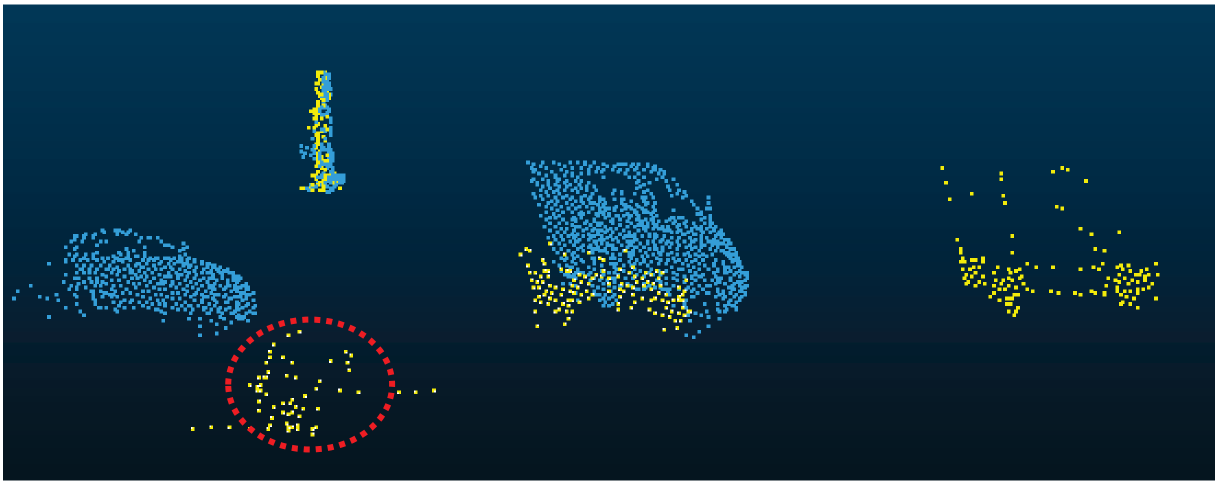


Figure 5-2: The correctness of sensor 2

In Figure 5-2, pairs of two moving cars can be easily distinguished visually. Moreover, with consideration of the shape of the point cloud, two-car pairs can be identified to be a small car and a van. However, as it is indicated in the red circle, a small segment was not successfully removed from the sensor 2 dataset. When applied the pairing, the small segment was found to be the closest pair and located in the front section of the car. Hence, it was wrongly recognized to be a moving pairs and the small segments in sensor 2 was wrongly identified to be a moving object.

5.2. Completeness

5.2.1. Both sensors

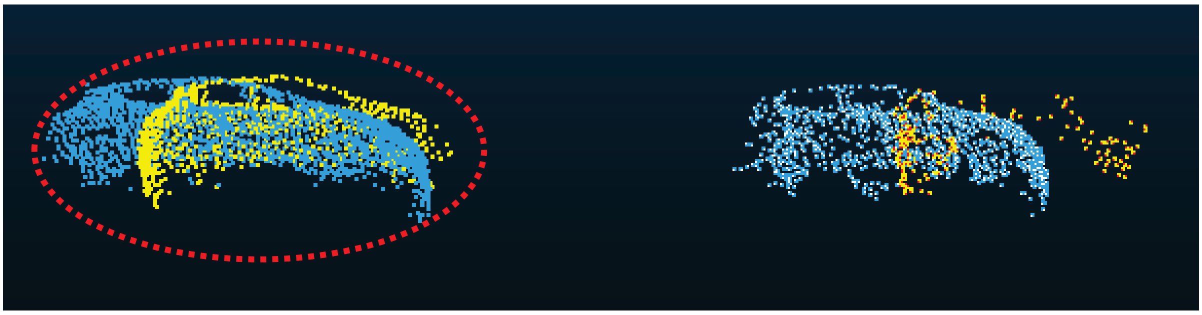


Figure 5-3: Completeness in both sensors

Figure 5-3 shows the example of two moving objects. The object in the red circle was incorrectly identified to be the static object because the aggregated points distance was smaller than a threshold value. It is because that the moving cars were moving slowly when they were captured by LiDAR sensor (according to the other contexts, the cars just started to move from the static state).

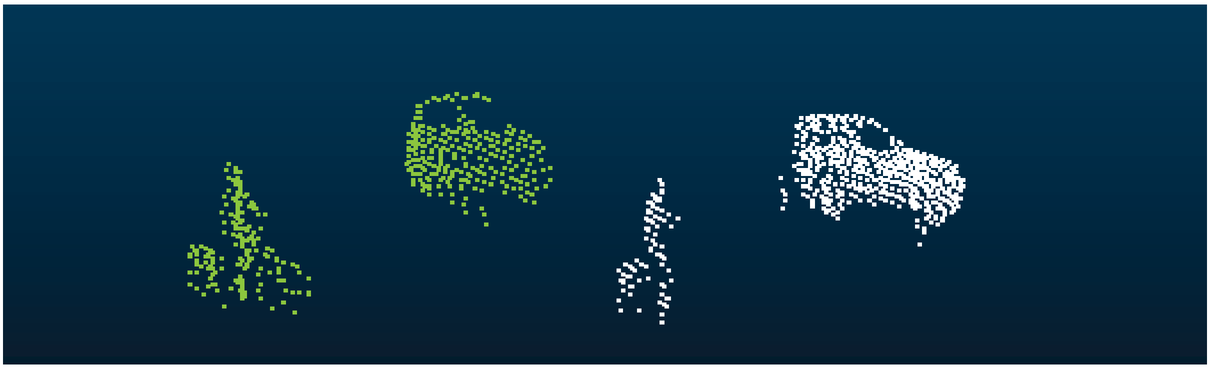


Figure 5-4: Completeness in both sensors

The two colors of segments (Figure 5-4) indicated the wrongly paired segments (bike and car) in the detection algorithm. As the orientation of green pairs was within the left section, it was identified to be the wrong pairs of segments. The white pair was recognized as a moving objects because the orientation was within the range.

5.2.2. Sensor 2 data

Figure 5-5: Completeness in sensor 2

The segments from two objects were paired together in the Figure 5-5 as they were found to be the closest distance segment in the local region. For the segments in the red circle, the white segment found its closest segment as the green object, therefore, the white segment was identified to be a moving object. However, the green color segment was found its closest distance segment with other object, where the orientation of the pairs was fall out of the front and back section. Thus, the segment was not successfully recognized. The similar case that the closest pair presenting the wrong segment pair resulted in the false negative detection in sensor 2 as shown in Figure 5-2.

5.3. Evaluation for detection algorithm with ideal dataset

As analyzed in the chapter 5.1 and 5.2, some of the incompleteness and incorrectness are caused by the errors that propagated from the previous steps. In order to better test the detection algorithm, the ideal dataset are generated manually which considered as only containing the temporary objects. Moreover, every objects in this dataset can be seen in both sensors and there are total 58 segments in each sensor stand for the 29 moving objects. The result for detection of moving objects in ideal dataset is shown at Table 4

Table 4: Result for the detection of moving objects with comparison dataset

	Sensor 1	Sensor 2	Sensor 1 + Sensor 2
True Positive	27	27	54
False Positive	0	1	2
False Negative	2	2	4
Completeness	0.93	0.93	0.93
Correctness	1.00	0.96	0.96
Accuracy	0.96	0.90	0.90

From the table, the both completeness and correctness had been improved, it is because the errors that were propagated from the previous steps were ignored. One false negative value was caused by a slow moving car, where the distance between two segments was smaller than the threshold value. And another one was caused by incorrect pair of the closest segment where two segments from two objects are determined to be the closest pair. The only false positive was caused by the shape of the one segment, which made incorrect estimation of the distance between two segments.

5.4. Program effiency

This section is describing the time used for each process. The time of processing the surface growing segmentation, ground, off-ground segments and building facade removal and connected component segmentation were roughly estimated. The time of the bounding box trimming process was included the time of calculating the bounding box. And all the processes were included the time of loading data. The result is showing at Table 5

Table 5: Time costs of processing each process

	Surface growing segmentation	Ground, off-ground and building facade removal	Connected component segmentation	Bounding box data trimming	Lamp-post data trimming	Moving object detection
Sensor 1	3mins	1min	2mins	77s	80s	660s
Sensor 2	3mins	1min	2mins	55s		570s
Numbers of points in Sensor 1	11 Million	11 Million	1771460	1771460	202753	63651
Numbers of points in Sensor 2	8.9 Million	8.9 Million	1449140	1771460		69020
Processing speed in both sensors (point/s)	55277p/s	165833p/s	13419/s	24398p/s	2534p/s	108p/s

5.5. Discussion

The fundamental concept for the detection of moving object was proven feasible. In the process for detection of moving objects, three types of objects were expected. 1- Static object: the correct segments pair (two segments belong to the same object) that was found from two sensors with small aggregate point distance. 2 – Moving object: the correct segments pairs (two segments belong to the same object) that was found from two sensors with large aggregate point distance and following a certain moving direction. 3 – Wrong pair object: the segment pairs which did not belong to either static object or moving object. The wrong pair object were primarily caused by the occlusion of the dataset or the over, under segmentation in the data process. Moreover, as not every driver follows the 2-second driving rule, the distance between front and rear cars in time are less than 2s, which also reduces the completeness of the algorithm.

In the beginning of the process, the surface growing segmentation was applied to determine the ground segments and remove those segments accordingly. However, by using the surface growing segmentation method, the large planar surface, which will be identified to be ground segments, may be easily over grown or under grown, which may lead temporary object segments got over-grown or under-grown. Those over-grown and under-grown objects will have higher probability to be wrongly identified in the detection algorithm, which increases the complexity of the detection algorithm and reduces the final detection accuracy (see Figure 4-3 and Figure 4-4). Moreover, the result of over growing ground surface will influence the shape and size of objects of interest that a part of the points belonging to the temporary object were identified to be ground segments. Small numbers of points added or subtracted to an object segment will not influence the result significantly, as the evaluation of the moving object was done in object domain. However, if the shape of the original object could be well preserved, the information of the segment shape can be used and worked better with the detection algorithm. The pairing method that applied in the detection algorithm, which was using closest aggregate point distance to establish the pair, not always pairs the same object. Knowing the shape of the object, the searching range can be set. In that case, the highest similarity shape within the search range will be selected as a pair instead of using the closest aggregate point distance. That will produce more reliable pairs for detection algorithm and improve the final accuracy. The static object segments pair will stay closer because of higher precision of the segments shape. Moreover, the orientations of the moving objects pairs could be better obtained as the centers of the segments were calculated from the better shape of the object and, therefore, they are more stable for representing the location of segment.

For the method that adopted in the lamp-post object trimming, it was only working on the road that orientation is consistent because the linear line fitting was applied to obtain the boundary of the road (chapter 4.2.5). In this case, the process of pole-like object trimming has to be done in small block making sure that the orientation of each block is consistent. The advantages of using equation fitting to obtain the boundary is that if several lamp-posts are missing, the boundary line can still be calculated based on the correctly detected lamp-posts. However, this process requires the manual interaction to guarantee the consistency of the road orientation. Another approach for detecting boundary line is connecting all the lamp-post locations sequentially and making a polygon of the road area. This process can be done automatically but requires high quality data of lamp-post extraction.

The reference data was generated manually by visually exploring the point cloud dataset and labeling the objects. Therefore, the evaluation was only applied in the object domain. Moreover, as no other reference was used in this process, the quality of the reference dataset cannot be evaluated. The reference dataset was generated after the data trimming. Cars and bikes, as they appeared in the relatively low noise area (on the road), the numbers of cars and bikes in the ground truth data were compared with original dataset. And the numbers of moving cars and bikes in the ground truth were very close to the true value. However, the numbers of pedestrians in dataset without image was not possible to compare with row data as they appeared far away from sensor (low point coverage) or under the trees or surrendered by other objects (High noise areas) see chapter 4.3. The evaluation accuracy of moving pedestrians were much lower than moving cars and bikes.

For the detection algorithm, the cars and bikes are easier to be detected if compared it with the pedestrians because the cars and bikes shown a clear feature in the form of shape and they move fast and moving direction are consistent. On the contrary, the pedestrians are very hard to be detected because they are moving slowly to the random direction. Moreover, the size of human are relatively small and without clear feature and they mostly appear on the sidewalk where it is far away from sensor and has low point density covered. Those weakness made the pedestrians harder to be detected than detection of moving cars and bikes.

6. CONCLUSION AND RECOMMENDATION

6.1. Conclusion

The objective of this MSc research is to develop an automatic or a semi-automatic program to detect the moving temporary objects from vehicle-based LiDAR point cloud. The method was developed under the assumption that the point segment belonging to one static object from both scanning sensors was paired and stay geometrically closer; on the contrary, the point segment belonging to one moving objects was also paired but separate apart with the certain orientations. The entire workflow contains two parts. The raw data pre-processing was conducted firstly. In that procedure, the maximum irrelevant points/objects were removed while all the temporary objects were preserved. Secondly, the detection of moving object will be conducted after the data being trimmed. A positive index and negative index was designed for testing if the segments pair was the closest pair related to both segments. Then, different detection of moving objects strategies were applied to those two groups of objects respectively according to their index. The final result was combined by the outcome of detection process from those two groups. The proposed algorithm was tested in the area where 30 moving objects were identified in the reference dataset including pedestrians, bikes and cars. The program worked automatically and relatively fast and entire process required little human interactions .The evaluation was done by comparing the result with the reference data and it shown the relatively good result. Several conclusions can be made after the experiment.

- The fundamental concept which using distance, orientation and location of paired segments to detect moving temporary was proven to be feasible.
- As a large amount of points in the dataset was irrelevant to the temporary object, a proper data trimming are better to be performed before the detection. It speeded up the procedure and reduced the complexity of detection.
- The connected component segmentation worked well to group the points into one segment.
- Using surface growing segmentation to segment and remove points which belong to ground was not mandatory, as it may influence the shape of the object.
- The assumption of the closest pair of segments belong to one object did not always hold in practical.
- The proposed algorithm was able to accurately detect the moving temporary object in the study area automatically. Based on the evaluation result, total completeness achieved 90%, correctness achieved 93.10% and overall accuracy was 84.37%.

6.2. Answers to research questions

1. How efficiently can an automated algorithm detect moving objects?

After the removal the ground and other irrelevant points, the process time will be less than 15mins for detecting 800m long road. However, the object trimming process will take longer time in order to accurately trim the irrelevant points out. The time of processing each process was shown in Table 5: Time costs of processing each process.

2. How can the moving objects be correctly extracted in the complex urban scenes point cloud?

Most of the segments of moving objects can be correctly extracted by connected component segmentation after the ground removal process as they appear on the ground with a cluster of high density points. However, for some small moving objects, like pedestrians, they appear on the sidewalk where far away from the sensor with low point density coverage and they may also occluded by other objects. Hence, to extract those moving object are difficult.

3. Moving objects are usually on the road, have a certain size/height and move along the road. Additionally, the scan angle between two scanners and the distances between objects and LiDAR will influence the shape of the captured object in the point cloud. How to incorporate those information in a method for automated detection of moving objects?

The lamp-post road trimming was designed to concentrate the objects on the road. And the size, height and other information were incorporated in the bounding box data trimming. In the detection algorithm, the threshold distance for moving objects was estimated by the scanning angles and speed of sensor platform.

4. Can the outcome of this study be used in other application for LiDAR data pre-processing?

No. surface growing segmentation was applied in the ground removal part, where it added or subtracted the points from the temporary object segments. The detected moving objects may contains fewer points than they actually have. Therefore, the result of this research can only tell the location at which there is a high probability of having points belonging to the moving objects.

6.3. Recommendation

The proposed algorithm could detect the moving temporary moving object with relatively high accuracy. However, some recommendation were given for further study in the related research topic.

- Using segmentation method to remove the ground is not suggested, as it will influence the shape of object of interest. Other ground removal technique, like using LAStools, might work better.

- The road boundary detection can be constrained by polygon, which will be less sensitive to the road orientation. However, higher precision of lamp-posts extraction will also be required.

- The closest segment did not always indicate the same object in practical, searching two or three closest segments and finding the correct pair out of it might be more reliable to find the correct segment pair.

- If the shape of the segments were well preserved, the reflectance and pulse count can be used when seeking for the correct pair from the closest three pairs.

- The independent dataset for evaluating the result are better to be tested as this study only worked in one dataset for testing and evaluation.

LIST OF REFERENCES

Betke, M., Haritaoglu, E., & Davis, L. S. (2000). Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications, 12*(2), 69-83. doi: 10.1007/s001380050126

Brenner, C. (2009). Extraction of Features from Mobile Laser Scanning Data for Future Driver Assistance Systems. In M. Sester, L. Bernard & V. Paelke (Eds.), *Advances in GIScience* (pp. 25-42): Springer Berlin Heidelberg.

Elberink, S. O., & Vosselman, G. (2009). Building Reconstruction by Target Based Graph Matching on Incomplete Laser Data: Analysis and Limitations. *Sensors, 9*(8), 6101-6118.

Gavrila, D. M. (2000). Pedestrian Detection from a Moving Vehicle. In D. Vernon (Ed.), *Computer Vision — ECCV 2000* (Vol. 1843, pp. 37-49): Springer Berlin Heidelberg.

Gibson, J. J. (1954). The visual perception of objective motion and subjective movement. *Psychological Review, 61*(5), 304-314. doi: 10.1037/h0061885

Hansung, K., Guillemaut, J. Y., Takai, T., Sarim, M., & Hilton, A. (2012). Outdoor Dynamic 3-D Scene Reconstruction. *Circuits and Systems for Video Technology, IEEE Transactions on, 22*(11), 1611-1622. doi: 10.1109/TCSVT.2012.2202185

Heipke, C., Mayer, H., Wiedemann, C., & Jamet, O. (1997). Evaluation of automatic road extraction. *International Archives of Photogrammetry and Remote Sensing, 32*(3 SECT 4W2), 151-160.

internationaltransportforum. (2012). Permissible Maximum Dimensions of Trucks in Europe http://www.internationaltransportforum.org/IntOrg/road/pdf/dimensions.pdf

Jochem, A., Höfle, B., & Rutzinger, M. (2011). Extraction of vertical walls from mobile laser scanning data for solar potential assessment. *Remote Sensing, 3*(4), 650-667.

Korsawe, J. (2008). Minimal Bounding Box. Retrieved from http://www.mathworks.com/matlabcentral/fileexchange/18264-minimal-bounding-box/content/minboundbox.m

Lehtomaki, M., Jaakkola, A., Hyyppa, 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

LI, D. (2013). Optimising Detection of Road Furniture (Pole-like Objects) in Mobile Laser Scanner Data.

MAN_Bus_EU. (2012). MAN Truck & BUS. from http://www.bus.man.eu/man/media/en/content_medien/doc/business_website_bus_master_1/Lions_City.pdf

Municipality_Enschde. (2010). Standard street sections (Standaard straatprofielen). from http://www.enschede.nl/wonen/Leefomgeving/TOR/Wegwijs.pdf/

Optech. (2009). The LYNX Mobile Mapper Specification.

Pokrajac, D., Zeljkovic, V., & Latecki, L. J. (2005, 28-30 Sept. 2005). *Spatial-temporal algorithm for moving objects detection in infrared video sequences.* Paper presented at the Telecommunications in Modern Satellite, Cable and Broadcasting Services, 2005. 7th International Conference on.

Pratihast, A. K. (2010). 3D tree modelling using mobile laser scanning data. *University of Twente, Faculty of Geo-Information and Earth Observation ITC, Enschede.*

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, Supplement), S28-S39. doi: http://dx.doi.org/10.1016/j.isprsjprs.2011.08.006

Pu, S., & Vosselman, G. (2009). Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing, 64*(6), 575-584. doi: http://dx.doi.org/10.1016/j.isprsjprs.2009.04.001

road_safty, C. s. T. (2010). Safe distnce between vehicles.

Rutzinger, M., Rüf, B., Höfle, B., & Vetter, M. (2010). *Change Detection of Building Footprints from Airborne Laser Scanning Acquired in Short Time Intervals*.

Sato, S., Hashimoto, M., Takita, M., Takagi, K., & Ogawa, T. (2010). *Multilayer lidar-based pedestrian tracking in urban environments.* Paper presented at the Intelligent Vehicles Symposium (IV), 2010 IEEE.

Schütz, M., & Dietmayer, K. (2013). A Flexible Environment Perception Framework for Advanced Driver Assistance Systems. In J. Fischer-Wolfarth & G. Meyer (Eds.), *Advanced Microsystems for Automotive Applications 2013* (pp. 21-29): Springer International Publishing.

Shan, J., & Toth, C. K. (2008). *Topographic laser ranging and scanning: principles and processing*: CRC Press.

Thompson, W., & Pong, T.-C. (1990). Detecting moving objects. *International Journal of Computer Vision, 4*(1), 39-57. doi: 10.1007/BF00137442

Ussyshkin, V. (2009). Mobile Laser Scanning Technology for Surveying Application: From Data Collection to End-Products. *FIG Working Week, Surveyors Key Role in Accelerated Development Eilat, Israel*, 3-8.

Vakautawale, M. (2010). *Information extraction from mobile laser scanner for road inventory application.* (MSc), University of Twente Faculty of Geo-Information and Earth Observation (ITC), Enschede.

VDL_BUS&COACH. (2014). Technical specifications Euro 6. from http://www.vdlbuscoach.com/Producten/Openbaar-vervoer/Citea/Technische-specificatie.aspx

Velizhev, A., Shapovalov, R., & Schindler, K. (2012). *Implicit shape models for object detection in 3D point clouds.* Paper presented at the International Society of Photogrammetry and Remote Sensing Congress}.

Vosselman, G., Gorte, B., & Sithole, G. *Change detection for updating medium scale maps using laser altimetry*.

Vosselman, G., & Maas, H.-G. (Eds.). (2010). *Airborne and terrestrial laser scanning*: Whittles Publishing.

Xu, S., Vosselman, G., & Oude Elberink, S. (2013). Detection and Classification of Changes in Buildings from Airborne Laser Scanning Data. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-5/W2*, 343-348. doi: 10.5194/isprsannals-II-5-W2-343-2013

Yang, B., Fang, L., & Li, J. (2013). Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing, 79*(0), 80-93. doi: http://dx.doi.org/10.1016/j.isprsjprs.2013.01.016