School of Earth and Planetary Sciences
Faculty of Science and Engineering

Plane Segmentation and Registration of Sparse and Heterogeneous Mobile Laser Scanning Point Clouds

HOANG LONG NGUYEN

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

June 2018

DECLARATION

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: . . . .

Date: 26/06/2018

ABSTRACT

Mobile laser scanning (MLS) systems have become the mainstream method of collecting 3D data. The captured point clouds from MLS systems at high speed have sparse and heterogeneous point density. This thesis analyses the potential issues due to these properties of MLS point clouds in different point cloud processing steps, such as local neighbourhood selection, normal vector estimation, plane detection and segmentation, point cloud registration and evaluation of the quality of the registration. The analyses demonstrate that problems may arise when applied to existing methods to those point clouds processing steps to the sparse and heterogeneous point clouds.

This work presents a novel method that utilises the properties of MLS point clouds (e.g. points are captured in order and the scanline patterns) and the fact that points of the same scan profiles belong to the same planar surface to detect and segment planes from MLS point clouds. In order to segment scan profiles from the MLS point clouds, a new saliency feature, namely the direction vector of a point and a novel local neighbouring point selection approach is introduced. The proposed plane detection and segmentation method was verified and compared with respect to other state-of-the-art methods using three different datasets. The results suggest that the proposed method outperforms the state-of-the-art methods in detecting and segmenting planar features in MLS sparse and heterogeneous point clouds.

In addition, the suitability of different point cloud matching techniques when applied to MLS sparse point cloud registration is discussed and analysed. A new error metric is introduced for this evaluation. The experimental results show that among the current point cloud matching technique, the least-squares plane fitting adjustment (LSPFA) is the most suitable matching technique for MLS sparse point cloud registration and the proposed error metrics is significantly more appropriate for evaluation and comparison the point cloud registration quality than the existing error metrics.

Finally, this research conducts a comparative study of automatic plane-based registration (i.e. LSPFA) for MLS sparse point clouds using inputs obtained from different plane detection and segmentation approaches, and investigates the relationship between the quality of the inputs for LSPFA (e.g. mean error of plane fitting and the discrepancies between the plane parameters) and the quality of the outputs of the MLS point cloud registration. The results suggest that the proposed plane detection and segmentation method is the most suitable technique to obtain inputs for LSPFA as well as there are dependencies between the quality of the inputs and the outputs of LSPFA.

ACKNOWLEDGEMENT

First and foremost I would like to express my sincere gratitude and admiration to my supervisors, Dr. David Belton and Dr. Petra Helmholz for all their assistances and guidance throughout my research at the Department of Spatial Sciences. I am thankful to Dr. Geoff West who was my first supervisor for the first two years of my research. He is the one who motivated me to research on Mobile Laser Scanning. Thanks to the constructive criticism and discussion from them my knowledge and communication skills have been significantly improved during my research. We had a beautiful moment at Rottnest Island with other colleagues from the Department of spatial sciences which I will never forget.

I would like to thank to Yousif, my colleague who had a lot of discussions with me not only about research and tutoring problems but also sharing the life issues. Also, tremendous thanks go to Dr. Ting On Chan who shared the Photogrammetry Lab with me for five months. His advices, encouragements and discussions helped me a lot in my research. We also have many unforgettable moments during the times he was in Perth. I also want to thank to my colleagues Cynthia, Richard and Josh who shared the photogrammetry lab with me.

I would also like to thank to Dr. Ulan who is my master thesis supervisor. Surprisingly, we met again at the Department of Spatial Sciences, Curtin University 4 years after I completed my master at ITC. We had beautiful moments when playing soccer together at Curtin. I also want to thank Dr. Viet Nguyen and my Vietnamese friends in Perth for their support during my normal life.

I am grateful to have been awarded MOET (Vietnam Ministry of Education and Training) scholarship and a top up scholarship from Curtin International Postgraduate Research Scholarship (CIPRS). I want to thank to the office and academic staff, colleagues and friends in the Department of Spatial Sciences, Curtin University.

Finally, I want to thank to my wife, my son, my parents, my younger sister, my parent in law, my younger brother in law who always support me during my study. Without their love and supports this thesis would not be completed.

CONTENTS

PUBLICATIONS AND PRESENTATIONS BASED ON THIS THESIS

Some of the work presented in this thesis has already been published or submitted by the thesis author and the paper co-authors.

Journal papers (peer-reviewed and fully refereed)

(1) Nguyen, H., Belton, D., & Helmholz, P. Planar Surface Detection For Sparse Mobile Laser Scanning Point Clouds. ISPRS Journal of Photogrammetry and Remote Sensing, Submitted and under revision.

(2) Nguyen, H., Belton, D., & Helmholz, P. Review Of Mobile Laser Scanning Registration And The Error Metrics For Quality Evaluation. Photogrammetric Records, Submitted and under revision.

Conference paper (peer-reviewed and fully refereed)

(3) Nguyen, H., Belton, D., & Helmholz, P. (2017). Comparative Study of Automatic Plane Fitting Registration for MLS Sparse Point Clouds with Different Plane Segmentation Methods. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-2/W4, 115-122. doi: 10.5194/isprs-annals-IV-2-W4-115-2017

(4) Nguyen, H. L., Belton, D., & Helmholz, P. (2016). Scan Profiles Based Method For Segmentation And Extraction Of Planar Objects In Mobile Laser Scanning Point Clouds. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLI-B3, 351-358. doi: 10.5194/isprs-archives-XLI-B3-351-2016

Conference presentations (peer-reviewed and fully refereed)

(5) Nguyen, H. L., Helmholz, P., Belton, D., & West, G. (2015). New Methods For Segmentation Of Sparse Mobile Laser Scanning Point Clouds. Paper presented at the The 9th International Symposium on Mobile Mapping Technology (MMT 2015), Sydney, Australia.

Thesis attribution of papers published or being submitted

Author: Nguyen, H. L

Paper #	1	2	3	4	5
Conception and design	x	x	x	x	x
Acquisition of the data and method	x	x	x	x	x
Data conditioning & manipulation	x	x	x	x	x
Analysis and statistical method	x	x	x	x	x
Interpretation and discussion	x	x	x	x	x
Final approval	x	x	x	x	x

I acknowledge that these represent my contribution to the above research output:

Signed

Author: Belton, D.

Paper #	1	2	3	4	5
Conception and design					
Acquisition of the data and method	x	x	x	x	x
Data conditioning & manipulation					
Analysis and statistical method					
Interpretation and discussion	x	x	x	x	x
Final approval					

I acknowledge that these represent my contribution to the above research output:

Signed

Author: Helmholz, P.

Paper #	**1**	**2**	**3**	**4**	**5**
Conception and design					
Acquisition of the data and method				x	x
Data conditioning & manipulation					
Analysis and statistical method					
Interpretation and discussion	x	x	x	x	x
Final approval					

I acknowledge that these represent my contribution to the above research output:

Signed

Author: West, G.

Paper #	5
Conception and design	x
Acquisition of the data and method	
Data conditioning & manipulation	
Analysis and statistical method	
Interpretation and discussion	x
Final approval	

I acknowledge that these represent my contribution to the above research output:

Signed G. A. W. West

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

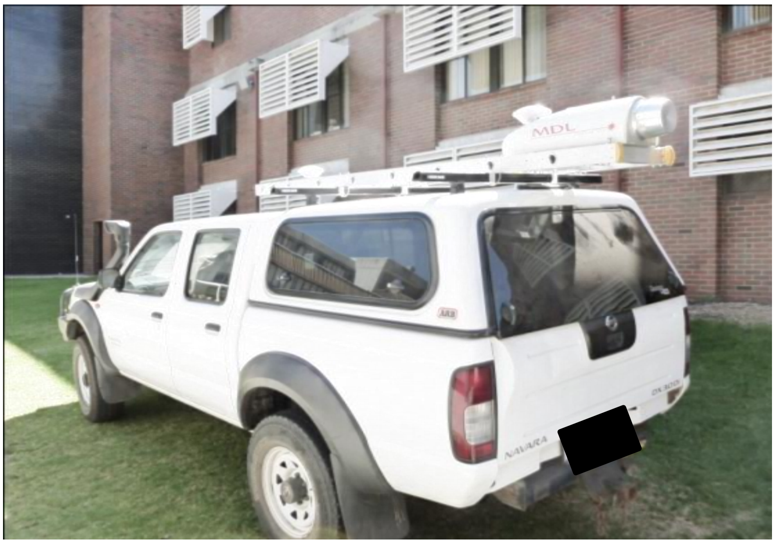
ALS	Aerial Laser Scanning
Cabo	Plane detection based on the line arrangement
CS	Correctly Segmented
IS	Incorrectly Segmented
ICP	Iterative Closest Point
KNN	K-Nearest Neighbourhood
FDN	Fix Distance Neighbourhood
ME	Mean Error
MS	Miss Segmented
MLS	Mobile Laser Scanning
LSPFA	Least-squares Plane Fitting Adjustment
RANSAC	Random Sampling Consensus
mRANSAC	Modified Random Sampling Consensus
RDPCA	Robust and Diagnostic Principal Component Analysis
RANSAC-NV	Random Sampling Consensus considering Normal Vector
RME	Root mean square error
RGPL	Region growing based on the planarity value of neighbouring line segments
SVFPM	Semantic Virtual Feature Points Matching
TLS	Terrestrial Laser Scanning
PCA	Principal Component Analysis
PSPS	Plane detection and segmentation based on the planarity value of neighbouring scan profiles

CHAPTER 1 INTRODUCTION

1.1 Mobile laser scanning system and applications

The demand for accurate spatial data has been increasing rapidly in recent years. Since the late 1980s, mobile laser scanning (MLS) has been developed and driven partly due to this demand. A mobile laser scanning system allows the observations of 3D measurements of objects from a moving platform, hence capturing more details of objects than traditional mapping or surveying practices, and at a much faster rate. As a result, MLS has become a mainstream technology for measuring 3D spatial data. It has been used in a number of applications such as urban surveys, transport infrastructure surveys, highway infrastructure surveys, 3D urban mapping, road and railway construction, coastal surveys applications and many others (Boeder et al., 2011; Shi et al., 2008; Vosselman & Maas, 2010).

A MLS system consists of a moving platform, navigation sensors and laser scanner(s). The moving platform varies for different applications. For instance, it could be a car (Shi et al., 2008), a track maintenance car (Morgan, 2009) or a boat (Boeder et al., 2011). The navigation sensors are normally GNSS receivers in combination with an Inertial Navigation System/Inertial Measurement Unit (INS/IMU), which are used for direct geo-referencing by providing the position and trajectory of the moving platform. Nowadays, there are a number of scanners available on the market including IP-S2, TRIMBLE MX8, LYNX Mobile Mapper, ROAD-SCANNER, STREETMAPPER, VMX-250 - RIEGL, RIEGL VQ-450, MDL Dynascan M250, MDL Dynascan S250. Figure 1.1 shows an example of such a MLS system – the MDL Dynascan S250. Different scanners may have different technical capabilities and specifications, and the systems’ limitations and capabilities must be taken into account when it comes to the applications and processing.



[redacted]

Figure 1.1 An example of a MLS system – MDL Dynacan S250 mounted on a car

The products derived from MLS systems are initially in the form of 3D geo-referenced point clouds. The position of a point in the captured point cloud is defined by its latitude, longitude and elevation, or by 3D Cartesian coordinates. Depending on the MLS system, the captured data may also include the intensity value of each captured point (which describes the strength of the reflected laser beam), and its colours if the captured MLS system is equipped with cameras. An example of a captured MLS point cloud is shown in Figure 1.2. The colours indicate the intensity values of points. The red line indicates the trajectories of the MLS system.

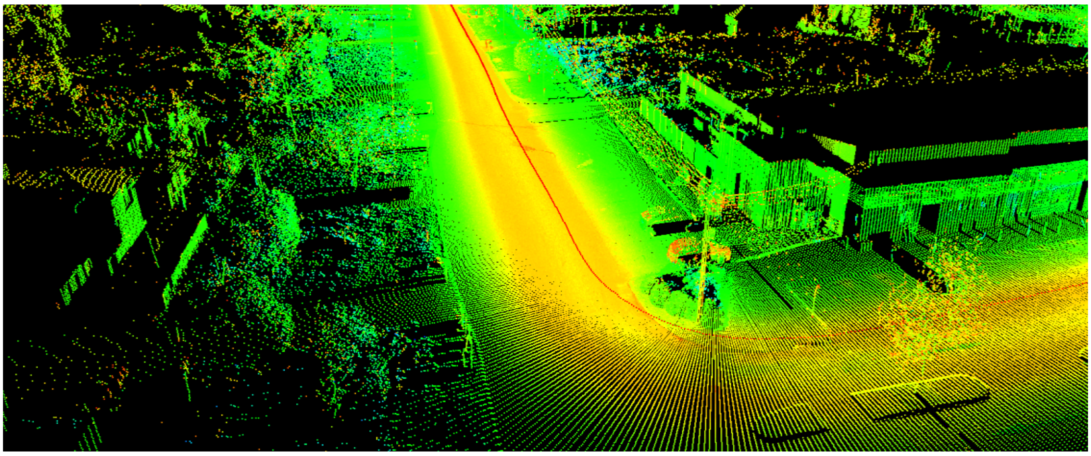


Figure 1.2 An example of a referenced 3D point cloud captured by MDL Dynascan S250

Besides the information that is directly obtained from the scanner(s), local saliency features for each point (e.g. normal vectors and curvatures) can also be estimated based on the nature of the query point-based on its local neighbourhood. These local neighbourhoods are normally determined using K-nearest neighbour (KNN) or fix distance neighbourhood (FDN) algorithms. Then, methods such as principal component analysis (PCA) and least-squares fitting are utilised to provide the estimated values of the local saliency features. The estimated saliency features are widely used in the extraction of features and further processing.

Since the captured point clouds from the MLS are just the point data, is does not provide any information or knowledge about the scanning area. Because of this, there is a need to extract different geometric features such as planes (e.g. building facades, roofs), cylinders (lamp poles), and other such information from the captured MLS point clouds, in order to define and isolate the different elements in the scene. Among those, planar features are the dominant features that exist in most of the survey sites, especially in urban areas due to the presence of building façades. Extraction of planar features is extremely important and is required in several MLS point clouds processing steps such as modelling, registration and calibration (Chan et al., 2013; Previtali et al., 2014; Skaloud & Lichti, 2006).

In many projects, the target areas are required to be scanned several times in order to ensure coverage, good point density, or to obtain the desired objects, which may be occluded due to the presence of obstacles during the scan. Nevertheless, the captured MLS point clouds normally contain errors due to the loss of the GNSS signals and inertial drifts in IMU (Takai et al., 2013). Furthermore, the GNSS used in the MLS system may not have the achievable level of accuracy to position/overlap the scans within the accuracy of the scanner/imagery. These factors lead to discrepancies between different MLS point clouds that are collected, or between the MLS point clouds and pre-existing models (e.g. 3D maps) of the same scanning area. In other words, the derived point clouds of the same area at different runs may not perfectly overlap with each other or with the pre-existing models. These mis-alignments can lead to errors in the modelling or interpretation of the data. Different point cloud matching techniques have been proposed to perform the point cloud registration (i.e. to match a *slave* point cloud to a *master* point cloud (Previtali et al., 2014) or to a *master* model based on existing information (Khoshelham & Gorte, 2009)). These techniques can be categorised into two groups, namely point-based matching (Besl & McKay, 1992; Chen & Medioni, 1991; Gressin et al., 2013; Rusinkiewicz & Levoy, 2001) and feature based matching (e.g. plane-based matching) (Previtali et al., 2014; Rabbani et al., 2007). Both of these have been utilised in MLS point cloud registration.

In order to evaluate the quality of the point cloud registration process, different error metrics have been introduced. Xiao et al. (2012) utilised visual inspection to assess the quality of the point cloud registration outputs. Yang et al. (2016) calculated the distances between targets before and after registration. While other researchers (Grant et al., 2013; Previtali et al., 2014; Yang et al., 2016) measure the registration quality based on the differences between the estimated transformation parameters with benchmark values. Another two error metrics are based on the RMS values of the residuals of the distances between corresponding pairs of points (Gressin et al., 2013; Takai et al., 2013), and the sum of the RMS of the residuals of the distances between points to their corresponding surfaces (Bae, 2006; Chan et al., 2016; Rabbani et al., 2007; Skaloud & Lichti, 2006).

Compared with TLS, the point density of MLS point clouds is often much sparser and more heterogeneous. According to Cahalane et al. (2014), due to the property of MLS point clouds such as points being collected in sequence, the point density of a captured MLS point clouds can be defined based on the point spacing and the distance between the scanlines across the surface (profile spacing) values. Point spacing is the distance between two consecutive points, whereas profile spacing is the distance between two consecutive scan profiles. Point spacing and profile spacing of a MLS point cloud depend on several factors, such as the distance between the scanner(s) to the objects along the trajectory and the velocity of the vehicle. Therefore, point clouds of different objects in the MLS point cloud captured with the same system may have different point densities. If a low-end MLS system (e.g. low scanner rate and low scanner pulse rate) is utilised to capture the data, the point densities of collected MLS Point clouds are normally sparse and heterogeneous in nature. This can lead to issues when processing the data.

1.2 Research Objectives

The sparseness and heterogeneousness (see Chapter 2) of the MLS point clouds lead to special challenges for different tasks in the point cloud processing, such as local neighbourhood selection, local saliency features estimation, features segmentation (e.g. plane detection and segmentation) and point cloud registration. Given the nature of some of the low-end systems, or the limitations in capturing data, this is a common problem faced in point cloud data. The sparseness and heterogeneousness of the captured MLS point clouds lead to special challenges to point cloud post-processing steps. For instance, the current neighbourhood selection methods such as KNN and FND have limitations in determining the neighbourhood of a query point in MLS point clouds in many cases due to sparse and heterogeneous sampling (see sections 2.2 and 3.1.2). This may lead to inaccurate local saliency feature values, and could propagate into further processing steps. Therefore, the first objective is to determine issues caused by the sparseness and heterogeneity of the captured MLS point clouds for different point clouds processing steps. This will focus on planar feature detection and segmentation in particular, as this process plays an important step in further processing steps, such as registration for example. From this, the second objective is to propose a new automatic plane detection and segmentation algorithm that can overcome the disadvantages of the current state-of-the-art methods (see section 2.3) when applied to the sparse and heterogeneous MLS point cloud datasets.

As mentioned previously, different point cloud matching techniques have been utilised to perform MLS point cloud registration. Each of them has their own advantages and drawbacks. Nevertheless, there is limited research that compares the quality of the registration outputs obtained between using these techniques. Hence, the next objective of this work is to discuss and analyse the suitability of these point cloud matching technique when applied to perform registration of sparse and heterogeneous MLS point clouds. This is done in order to determine which point cloud matching approach is the most suitable method to be applied. In order to compare the quality of different registration outputs, a proper error metric for the evaluation is required. Therefore, the fourth objective is to discuss and analyse the limitations of the existing error metrics for the evaluation of the quality of the registration of MLS point clouds. A new error metric is introduced based on this that aims to be not only be suitable in evaluating the quality of the registration outputs but can also be used to compare the quality of the registration outputs obtained using different techniques.

In previous works, the quality of the output from the plane-based matching approaches is believed to have a relation with the quality of the extracted planes. These extracted planes can be derived using different plane segmentation algorithms. Due to the differences in the nature of the applied algorithms themselves and the values of the required parameters, the extracted planes may vary. The final objective is to determine which plane detection and segmentation technique is the most suitable method for providing the inputs for plane-based matching in the presence of sparse and heterogeneous point clouds.

1.3 Contributions and significance

The processing of sparse and heterogeneous point clouds is especially challenging. Most current methods in point cloud processing have limitations in these circumstances, or do not take these properties into account. The significance of this research is that it evaluates the effects of these conditions on existing methods, as well as introduces new methods and metric to overcome these challenges. The outcomes of this thesis are to improve the outputs and utilisation of the results of sparse and heterogeneous point clouds, which are common in low end systems, and some capture campaigns.

The main contributions of this research are as follows:

- This thesis discusses and demonstrates the challenges of the current state-of-the-art local saliency feature estimation methods when applied to the sparse and heterogeneous MLS point clouds. These issues have not been the focus, or addressed in previous research.

- This research introduces a new local saliency feature, namely direction vector, a new local neighbourhood selection method, a new variant of RANSAC for outlier detection and a variant of region growing which are extremely suitable to segment scan profiles from MLS point clouds.

- A new plane detection and segmentation method that is robust and insensitive to the parameters of the method, as well as the properties of the MLS point clouds is developed. This method utilises the proposed local saliency feature, and the planarity values between neighbouring scan profiles to detect and segment different planes in MLS point clouds.

- Traditionally, a segmentation method is normally evaluated based on the number of over-segmented and under-segmented objects. As the segmented planes can be used as inputs for other processes (e.g. the least-squares plane fitting adjustment), this research presents three new evaluation criteria such as evaluation based on the number of correctly segmented and incorrectly segmented points, and evaluation based on the mean error and the plane parameters of the extracted planes.

- A new error metric is proposed that can not only be used to evaluate the quality of the registration, but also to compare the quality of the registration outputs from different matching techniques. This new error metric can overcome the limitations of other error metrics by considering errors in all possible orientations with equal weight.

- Based on experiments using real MLS point cloud datasets, this research shows that point-based matching techniques are not suitable for performing MLS point cloud registration, especially for sparse and heterogeneous MLS point clouds. In contrast, MLS registration should be performed using a feature based matching technique, particularly the least-squares plane fitting adjustment (LSPFA) technique.

- As segmented planes from different plane detection and segmentation methods may be different, a comparative study of automatic plane fitting registration for MLS sparse and heterogeneous point clouds with different plane segmentation methods is conducted. This study not only determined the most suitable segmentation method that should be used to obtain the inputs for the plane-based registration, but also investigates the relation between the quality of the inputs and the final registration outputs.

1.4 Outline of the thesis

In order to achieve the research objectives, this thesis is organised as follows:

In chapter 2 the backgrounds of MLS point clouds is presented, along with the definition of sparse and heterogeneous MLS point clouds being introduced. Next, a literature review about different point clouds processing steps such as local saliency features estimation, plane detection and segmentation is given. This is followed by a discussion of the limitations of recent methods in these point cloud processing steps. Finally, the challenges of the sparseness and heterogeneity of the MLS point clouds to various point cloud processing steps are presented and discussed.

Chapter 3 proposes a novel plane detection and segmentation method that is based on the fact that points on the same scan profile belong to the same features, and utilises the planarity values of different neighbouring scan profiles. A new local saliency feature, namely the direction vector, is introduced in order to extract scanline segments from the point clouds. Next, the limitations of the state-of-the-art local neighbourhood selection algorithms in estimating the direction vectors of points in a sparse and heterogeneous MLS point clouds are presented. To solve these limitations, a new local neighbourhood selection approach is introduced, based on the property of the MLS point clouds (e.g. points are captured in sequence) which is more suitable for the purpose of estimating the direction vectors. Then, a modified RANSAC algorithm is presented to remove "outliers" from the selected local neighbourhoods. A scanline segment extraction technique is proposed to extract scanline segments from the scanlines.

In chapter 4, different state-of-the-art point cloud matching techniques are discussed in detail. Then, the issues of these techniques in respect to the sparseness and heterogeneousness of the MLS point clouds are analysed. Afterwards, the suitability of different error metrics for evaluation of the quality of the MLS point cloud registration is analysed and discussed. Finally, a new error metric is introduced in order to provide a suitable measure to evaluate and compare the quality of the point cloud registration outputs.

In chapter 5, the proposed plane detection and segmentation method is verified using three different datasets including one simulated dataset and two real datasets captured using the MDL Dynascan S250. The outputs from this proposed method are compared with the state-of-the-art methods discussed in chapter 2 using different criteria.

In chapter 6, experiments are conducted using real datasets to determine the most suitable point cloud matching technique for MLS sparse point cloud registration using the proposed new error metric. The MLS point cloud registration outputs obtained using the LSPFA matching approach with inputs from different plane detection and segmentation techniques are presented and discussed. In order to investigate the impacts of the quality of the inputs to the LSPFA matching process and determine which method is the most suitable method for providing inputs for LSPFA, two different experiments are conducted.

Finally, the summary of the achievements of this research, and outlooks for future research are provided in chapter 7.

CHAPTER 2 BACKGROUND

In this chapter, the related background of the MLS point clouds is reviewed. Then, a definition of a MLS sparse and heterogeneous point cloud is introduced. Next, the related principles of point clouds processing such as local saliency features estimation, planar feature detection, and segmentation are summarised. Finally, the limitations of the different state-of-the-art approaches as in relation to the sparseness and heterogeneity of MLS point clouds are reviewed and discussed.

2.1 MLS point cloud density

2.1.1 Point and profile spacing

Most of the MLS system utilise 2D line scanner(s) in combination with the motion of the vehicle in order to capture 3D data. MLS systems can be classified into two types: MLS that has the laser head(s) placed perpendicular to the trajectory of the carrier vehicle (type 1); and MLS that has laser scanner(s) placed obliquely to the trajectory of the carrier vehicle (type 2). Point clouds captured by these two different MLS types have different scanline patterns. As a result, the features that can be extracted from these point clouds are different. For example, the MDL Dynascan M250 X-plane has two laser heads, each laser head is orientated at 45^0 to the vehicle (inclined with respect to the vertical and the direction of travel). This way the scanner can scan objects such as thin poles more comprehensively because of a higher point density and inclined scan angle. In comparison, the MDL Dynascan S250 with a single laser head placed perpendicular to the vehicle trajectory, can theoretical miss such features if they fall between two consecutive scanlines. As a result, the point density and the scanline pattern of point clouds differ for different MLSs. This research is mainly focus on type 1 MLS systems as shown in Figure 2.1, as it poses the most issues when processing such point clouds.

Cahalane et al. (2014) proposed different terminologies for different groups of neighbouring points in MLS point clouds. A full revolution of the scan is called a ‘scanline’. A scanline segment or a ‘scan profile’ is defined as a group of neighbouring points that meet two criteria: 1) they belong to the same surface and 2) they belong to the same scanline. In addition, Cabo et al. (2015) defined a polyline as a group of neighbouring scan profiles of the same scanline.

Commonly, the point density of a point cloud is defined as the number of points per meter squared, or the number of points per meter cubed. According to Cahalane et al. (2014), point cloud density of a feature in a 3D MLS point cloud can be specified as the distance between two adjacent points on the same scanline (i.e. 'point spacing') and distance between two neighbouring scan profiles (i.e. 'profile spacing'). Figure 2.1 illustrates examples of all of the terminology mentioned above. The point spacing and profile spacing of a MLS point clouds are influenced by different factors.

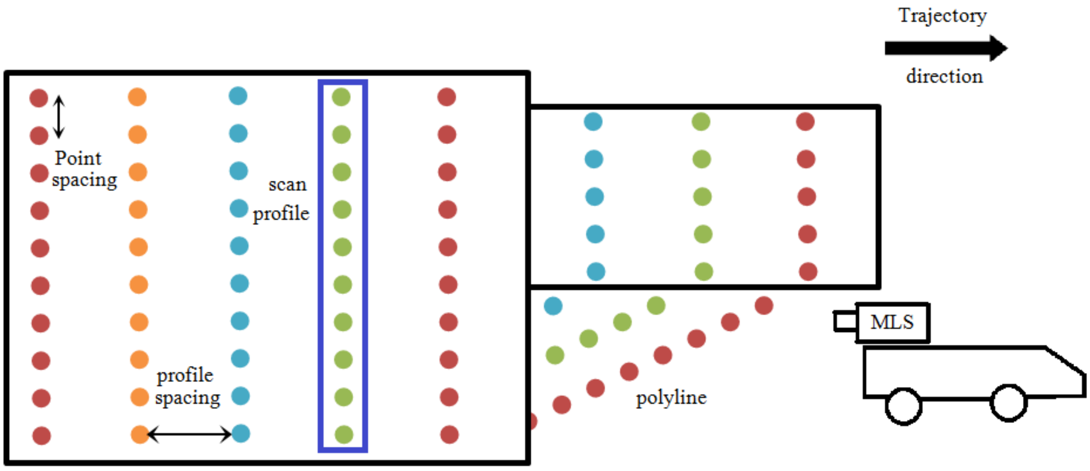


Figure 2.1 Scan profiles patterns of type 1 MLS in a simple environment.

From Cahalane et al. (2014), point spacing of an object is influenced by 1) the orientation of the target and the orientation of the scanner, 2) the height difference between the target and the scanner, 3) the distance from the scanner to the surface and 4) the scanner pulse rate (pulse repetition rate). The scanner pulse rate is the number of points that can be collected by a specific laser scanner in a minute.

The profile spacing is affected by 1) the orientation of the target surface, 2) the speed of the vehicle, 3) the orientation of the scanner and 4) the scanner rate (Cahalane et al. 2014). The scanner rate defines the number of full revolutions of the scanner per minute that a specific laser scanner can perform. In reality, especially for urban scenes, most of the planes are vertical planar surfaces that have normal vectors orthogonal to the vehicle trajectory. Hence, in this research, it is considered that the standard profile spacing D_{sl} as the profile spacing of these planes which can be calculated using eq. 1 as introduced in Cahalane et al. (2010).

$$D_{sl} = \frac{V}{S} \qquad \text{(eq. 1)}$$

Where D_{sl} = standard profile spacing

V = vehicle speed (m/s)

S = scan rate (Hz)

In conclusion, there are two technical specifications (i.e. the scanner rate and the scanner pulse rate) of a MLS scanner that influence the point density of an object in a captured MLS point cloud.

2.1.2 Sparse and heterogeneous point clouds

As previously mentioned, a large number of MLS systems are available on the market. Seven of them, namely ROAD-SCANNER, IP-S2, TRIMBLE MX8, STREETMAPPER, VMX-250 - RIEGL, MDL Dynascan, LYNX Mobile Mapper have been reviewed by Puente et al. (2012). As pointed out earlier, due to different capabilities and technical specifications of different MLS systems, point densities of the same feature captured under the same conditions (e.g. the same distances between scanner and target surface, and the same vehicle speed) will vary using different MLS systems. Table 2.1 compares the four specifications of range, scanner rate (i.e. number of scanlines per second), scanner pulse rate (i.e. number of points per second) and accuracy for the MDL Dynascan S250 (MDL, 2017) and RIEGL VQ-450 (RIEGL, 2015). In this research, a MLS system that has similar specifications (e.g. scanner rate and scanner pulse rate) to MDL Dynascan S250 is considered a low-end MLS system.

	MDL Dyanscan S250	RIEGL VQ-450
Range	up to 250 m	up to 800 m
Scanner rate	up to 20 Hz	up to 200 Hz
Scanner pulse rate	36,000 pps	550,000 pps
Accuracy	1 cm	8 mm
Type	Perpendicular to the scan direction	Oblique to the scan direction

Table 2.1 MDL Dynascan S250's specifications (adopted from (Nguyen et al., 2016))

The profile spacing and the point spacing of the same object captured using MDL Dynascan and RIEGL at different speeds and distances to the target surface is illustrated in Table 2.2 and Table 2.3. In Table 2.2 the standard profile spacing distances are computed using equation 1. As can be seen from Table 2.2 and Table 2.3, the profile spacing and the point spacing captured by the MDL Dynascan S250 are relatively large compare to the RIEGL VQ 450. Since the scanner rate of the RIEGL VQ 450 is 10 times faster than MDL Dynascan S250, the profile spacing of the same object in the same conditions collected by MDL Dynascan S250 is 10 times smaller than data collected by RIEGL VQ 450 (see Table 2.2). Meanwhile, the point spacing values at various distances to the target surface are manually measured from real MLS point clouds, and presented in Table 2.3. The point spacing of a point cloud captured by MDL Dynascan S250 is 15 times larger than captured by the RIEGL VQ 450. In addition, the profile spacing is much bigger than the point spacing.

Standard Profile Spacing [m]		
Speed	MDL Dyanscan S250	RIEGL VQ-450
20 km/h	0.27	0.027
30 km/h	0.41	0.041
40 km/h	0.55	0.055
50 km/h	0.69	0.069

Table 2.2 Profile spacing values of the MDL Dyanscan S250 and the RIGL VQ 450 at speeds of 20, 30, 40 and 50 km/h.

Average point Spacing [m]		
Distance to the object [m]	MDL Dyanscan S250	RIEGL VQ-450 [m]
7	0.03	0.002
27	0.09	0.006
40	0.15	0.010

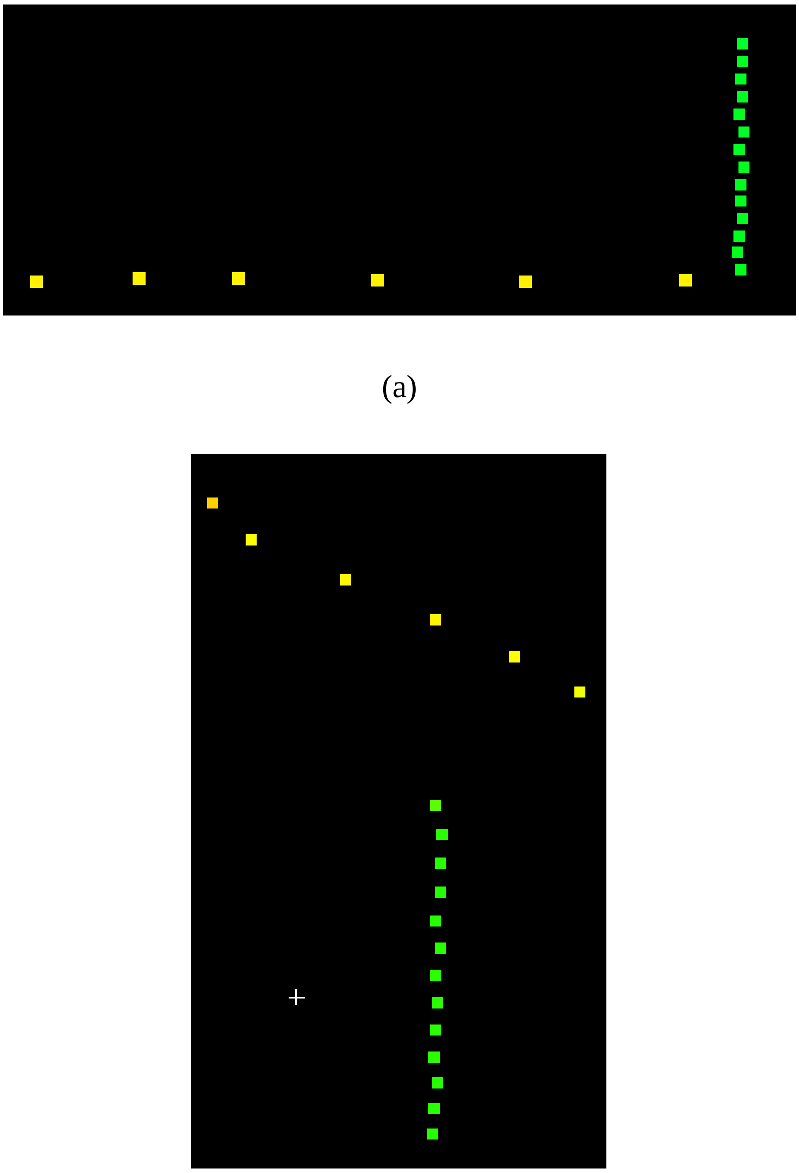
Table 2.3 Point spacing of the MDL Dyanscan S250 collected at different distances between the scanner and the target surface.

It is easy to see from Figure 2.2 that the point spacing values of two adjacent scan profiles (i.e. green and yellow scan profiles) of the same scanline can be significantly different. Specifically, in Figure 2.2 (a) the point spacing of the green scan profile is 69 mm while the point spacing of yellow scan profile is 600 mm. Depending on the orientation of the surfaces, points on the same scanline may have various distances to its neighbouring points (e.g. yellow scan profile in Figure 2.2).

Based on the above, in this research the captured MLS point cloud is defined as a sparse and heterogeneous point cloud if it has the following properties

- Large point spacing and profiles spacing distances (sparse),
- Large discrepancies between the point spacing distances of different scan profiles, and large discrepancies between the point spacing distances and profile spacing distances (heterogeneous).

Large in this context means the number of points is often insufficient to model the small feature or object (e.g. a pole may only have 1 scan profile which is not enough to model itself).



(a)

(b)

Figure 2.2 Examples of heterogeneous point spacing of a scanline of a real captured MLS point cloud; different colours indicate different scan profiles

In conclusion, the point clouds captured by a low-end MLS system commonly have sparse and heterogeneous point densities. These properties of the MLS point clouds may lead to special challenges for different tasks in point clouds processing such as local saliency features estimation and feature segmentation. The challenges for the registration process will be discussed in Chapter 4. Meanwhile, the problem with the local saliency feature and feature segmentation will be discussed in the next sections.

2.2 Local saliency features estimation

2.2.1 Finding local neighbouring points of the query point

Local saliency features such as the normal vector and curvature of a point are widely used in feature segmentation and registration of point cloud data (Gressin et al., 2013; Nurunnabi et al., 2015). They are normally estimated based on the nature of the query point and its local neighbouring points. When processing point clouds, local neighbouring points of the query point are normally determined using the K-nearest neighbourhood (KNN) or fix distance neighbourhood (FDN) algorithm. The KNN determines the local neighbouring points of a query point by setting *k*, where *k* is the number of the closest points to the query point. These points are considered as its local neighbourhood of points, and the local features are calculated based these points. Meanwhile, FDN determines the local neighbouring points of a query point by setting the *fd* parameter. Points that have a distance to the query point smaller than the value of *fd* will be considered as the local neighbouring points. The number of neighbouring points is fixed when using KNN, whereas the number of neighbouring points obtained using FDN depends on the point density around the query point.

Different studies have used either the KNN or FDN algorithms for different applications. KNN guarantees the point neighbourhood sample size, and hence redundancy for feature calculation. In contrast, the size of the local features, or the area that it is calculated on is not constant over the point cloud. In comparison, the FDN helps define the size of local features to be constant, but the redundancy is not consistent throughout and may fluctuate between low and high levels in heterogeneous point clouds, or not be sufficient to calculate the features.

2.2.2 Estimate local saliency features of the query point

After the neighbouring points of the query point are determined, local saliency features are calculated. There are a variety of methods to achieve this, but Principal

Component Analysis (PCA) is a popular method that is applied to estimate local saliency features of the query point. PCA is a popular statistical approach to reduce data dimensionality and for visualisation of the underlying properties (Jolliffe, 1986). In PCA, the nature of the query point and its neighbouring points can be investigated via their covariance matrix. The covariance matrix of these points is defined as:

$$C_{3x3} = \frac{1}{np} \sum_{\mathrm{i}=1}^{np} (p_i - p)^T (p_i - p) \qquad \text{(eq. 2)}$$

where *np* is the number of points, p_i is the point in the group, and *p* is the mean centre of the group.

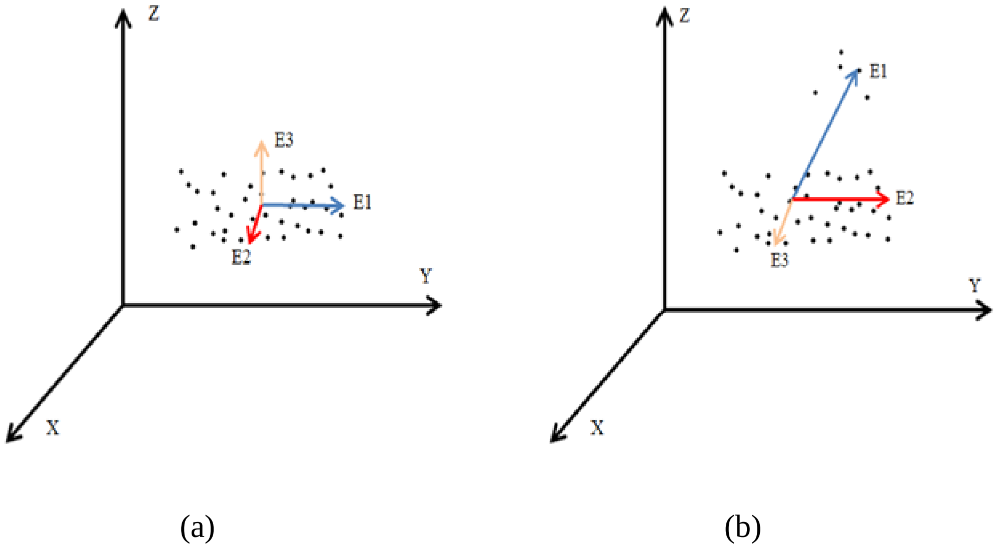
Then, Singular Value Decomposition is applied on to the covariance matrix to solve the eigenvalue equation:

$$\lambda * V = C * V \qquad \text{(eq. 3)}$$

where λ is the matrix that contains the eigenvalues as diagonal elements; *V* is the matrix that contains the corresponding eigenvectors as its columns. Each column of *V* (or the eigenvector) represents a principal component.

The principal components are ranked in a descending order, with the eigenvalues indicating the variation (variance) of the data along the associated eigenvectors. The first eigenvector (vector *E1* in Figure 2.3(a)) denotes the direction of the largest possible variance of the data, and is often used for representing linear features. The second eigenvector (vector *E2*) indicates the direction of the second largest possible variance of the data. In the case of plane fitting or determining planarity, the third eigenvector (vector *E3*) can be considered as the normal vector of the plane formed by the query point and its neighbouring points, and is used as an approximation of the query point’s surface normal vector.

PCA has been shown to be sensitive to outliers (Nurunnabi, 2014) as illustrated in Figure 2.3(b). Due to the presence of outliers, the normal vector of the query point is completely different from the correct vector in this case. These outliers may come from noise in the sampling or from points belonging to other surfaces that are included during the neighbouring selection process.



(a) (b)

Figure 2.3 Normal vectors estimated using PCA a) no outliers b) affected by outliers.

2.2.3 Outlier detection

Since PCA is sensitive to outliers, different outlier detection approaches have been proposed in order to detect and remove outliers from the selected neighbourhood of points. In this research, two of the most popular outliers detection methods in point clouds processing, namely robust model fitting, and Robust and Diagnostic PCA (RDPCA), are reviewed.

2.2.3.1 Robust model fitting for outlier detection

Robust model fitting approaches detect parameterised features and estimate the mathematical parameters of the features of interest from the point clouds using voting techniques. Random sample consensus (RANSAC) (Fischler & Bolles, 1981) and Hough Transform (Hough, 1962) are the two most popular paradigms in this group. Hough Transform detects features by transforming the points into a parameter space, and then performing voting over bins in this space. A high number of votes represent features with a large number of points supporting them. The numbers of dimension of this parameter space depend on the number of parameter of the parameterised features of interest. In the case of detecting planes, depending on the parameterisation, the parameter space has three or four dimensions. Different variants of Hough Transform were compared in Borrmann et al. (2011) in detecting planes in point clouds. Deschaud and Goulette (2010) claimed that compared with

RANSAC, the Hough Transform is too time consuming. Furthermore, the Hough Transform is very sensitive to the segmentation parameter values (Tarsha-Kurdi et al., 2007).

RANSAC was first introduced by Fischler and Bolles (1981). It is widely applied to remove outliers from selected neighbouring points in order to have a “clean” subset to estimate the normal vectors of points in point clouds. It starts by selecting three random points. Then, the plane parameters are calculated from this set of points. Next, any points that have orthogonal distances to the calculated plane smaller than a pre-define distance threshold will be labelled as inliers and form the consensus set. This orthogonal distance (*OD*) between a point p = *[x, y, z]* to a plane PL that is described by its normal vector n = (n_x, n_y, n_z) and the perpendicular distance to the origin ρ can be calculated as follows:

$$OD = n_x x + n_y y + n_z z + \rho \qquad \text{(eq. 4)}$$

This process is iteratively performed a number of times. The number of iterations needs to be large enough to ensure a good outcome based on the size of the feature and the number of predicted inliers. Finally, the plane parameters from the iteration, that has the most number of inliers will be chosen as the final output. The output from RANSAC is considered as a “clean” input for PCA to estimate the local saliency feature of the query point. Numerous variants of RANSAC have been proposed in order to increase the efficiency of RANSAC (Matas & Chum, 2004; Raguram et al., 2008; Torr & Zisserman, 2000). Some of them aim to produce a more useful hypothesis by modifying the sampling step, while others aim to optimize the model verification step. The performances of different variations of RANSAC were evaluated in Choi et al. (2009).

2.2.3.2 Robust and Diagnostic Principal Component Analysis for outlier detection

Nurunnabi et al. (2015) proposed the Robust and Diagnostic PCA (RDPCA) algorithm that aims to detect and remove outliers from data. Figure 2.4 shows the four steps of RDPCA.

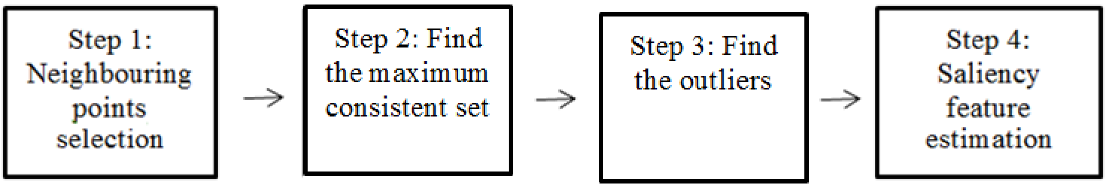

Step 1: Neighbouring points selection
Step 2: Find the maximum consistent set
Step 3: Find the outliers
Step 4: Saliency feature estimation

Figure 2.4 The four steps of RDPCA

In the first step, Nurunnabi et al. (2015) argued that neighbouring points should be selected using KNN so as the local statistics will be estimated from the same number of points, regardless of the point sampling density.

The next step is to find the maximum consistent set, which contains the majority of the inliers. The point to plane orthogonal distance and the variation along the normal vector of the surface are utilised in order to detect and remove outliers. Instead of finding the set that has the maximum number of inliers, RDPCA finds the maximum consistent set based on the value of the least eigenvalue. For all iterations in the algorithm, a fix number of points h is used for the inlier set. This fixed number of points is calculated based on the neighbourhood size parameter (k) and the percentage of outlier parameter (ϵ). Both of these parameters need to be empirically determined. The authors claimed that by fixing the h parameter, the problem of choosing a pre-defined value for the error threshold parameter that may cause problems when using RANSAC is avoided. This process starts by randomly selecting three points from the query point and its neighbourhood. Then, the orthogonal distances of all points to the plane formed by these three points are calculated. Next, h numbers of points that have the least distance to the plane are assigned to the consistent set. This process is iteratively repeated t number of times. The number of iterations t can be determined based on the percentage of outliers (ϵ) and the number of neighbourhood (k). Then, the subset that has the smallest Eigenvalue is chosen as the maximum consistent set.

In order to detect outliers from the local neighbourhood of the query point, Robust z-score based Outlier Detection or Robust MD based Outlier Detection that utilize robust z-score and robust Mahalanobis Distance respectively are used. Finally, PCA is performed on the clean point set to obtain the robust saliency features (e.g. normal vector and curvature) of the query points free from outliers.

RDPCA was proven to outperformed PCA and be more robust and accurate than RANSAC for local saliency features estimation in the case of the dense and homogenous point clouds (Nurunnabi et al., 2015). However, there is no experiment or evaluation of RDPCA on sparse and heterogeneous point clouds.

2.2.4 Issues of recent local saliency feature estimation approaches with sparse and heterogeneous MLS point clouds

Problems may arise when these approaches are applied to estimate the local saliency features for points in a sparse and heterogeneous MLS point cloud data. Previous research has primary been focused on technical solution rather than processing technique to overcome this issue (e.g. faster scanner(s), slower capture velocity).

KNN and FDN algorithms are commonly utilised to determine the neighbouring points of the query point. They may work well in the case of dense point cloud data. However, problems may occur when applying these two algorithms for a sparse and heterogeneous MLS point cloud. Firstly, in many cases the point spacing of a captured point cloud is much smaller than the profile spacing, hence KNN or FND will tend to select neighbouring points in the same scanline as the query point. Therefore, if the *k* or *fd* parameter is not large enough, all the selected neighbouring points may come from the same scanline causing incorrect saliency features of the query points to be estimated.

For instance, as can be seen from Table 2.2 and Table 2.3, the point cloud of a feature which is 7 m away from the scanner and captured at 50 km per hour by MDL Dynascan may have the point spacing 23 times smaller than the profile spacing. In this case, the forty-six closest points of a query point belong to the same scanline. If the *k* parameter is set smaller than 46 or the *fd* parameter is set as less than 0.69 m, all of the selected neighbouring points will be taken from the same scanline. Consequently, although all of the selected points are inliers (i.e. belong to the same plane with the query point) they are not sufficient to properly estimate the normal vector of the query point or represent the underlying surface as they all lie on a line in space. In theory, neighbouring points must be selected from at least two scanline in order to obtain an accurate normal vector.

One may argue that the value of *k* or *fd* can be increased to solve this problem. Nevertheless, this may lead to another problem where points belonging to a small surface may be represented by too few points. In these cases, if the *k* or *fd* is set too big, most of the selected neighbouring points will be outliers. Consequently, the percentile of outliers may be larger than the tolerance threshold that algorithms such as RANSAC and RDPCA can handle. Moreover, in a sparse and heterogeneous MLS point cloud, different surfaces may have different point densities, hence setting a fixed value for *k* with the KNN algorithm or fixed value for *fd* with the FDN algorithm does not seem to be appropriate when applied to the whole point cloud.

Another problem which may arise with the normal vector estimation process is the "erroneous inliers" issue. It occurs when there is an "erroneous" plane formed by inliers and outliers. For instance, as shown in Figure 2.5, a query point is near the boundary between two parallel planes (here coloured in red and blue). Depending on the value of *k* or *fd* parameter, the neighbouring points of the query point (yellow point) will be selected inside the orange circle. In this case, two possible planes can be formed (e.g. green plane – correct plane, and pink plane – incorrect plane). The green plane is correct and consists of just the blue points in the neighbourhood. The pink plane is incorrect and consists of the blue points in the same scan profile as the yellow point, and the red points in the neighbourhood. Both could be considered "valid" in term of calculation. Consequently, RANSAC and RDPCA may consider this "erroneous" plane as the plane that the query point belonged to.

Finally, RANSAC and RDPCA do not consider whether the query point is a part of the maximum consistent set or not. Consequently, the local saliency features of the query point may be estimated from points belonging to another feature instead of the feature the query point belongs to.

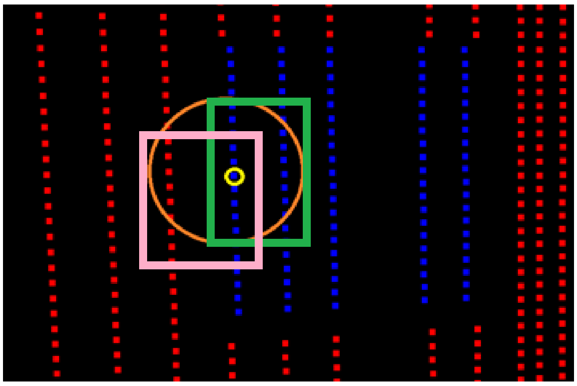


Figure 2.5 Erroneous inlier plane; the circle indicates the neighbouring points radius selection; green and pink rectangles indicates two different planes

2.3 Planar features detection and segmentation

Plane detection and segmentation is one of the most crucial tasks in point clouds processing. The outputs from this process can be utilised as the inputs for further processing steps, such as modelling, registration and calibration. Especially in respect to registration and calibration, the quality of the plane segmentation plays a very important role in the final quality of these two processes. Numerous approaches have been proposed to detect and segment planar objects from the captured point clouds. They can be categorised into four groups: edge based (Huang & Menq, 2001), robust model fitting (Fischler & Bolles, 1981; Hough, 1962; Vosselman & Maas, 2010), Region growing based (Nurunnabi et al., 2012; Rabbani et al., 2006; Vosselman & Maas, 2010) and scanline based (Cabo et al., 2015; Jiang & Bunke, 1994; Sithole & Vosselman, 2003).

These plane detection and segmentation approaches have their own drawbacks (e.g. sensitivity to the parameter values) that have been discussed in the related literature. Further drawbacks may also result when using sparse and heterogeneous point cloud data that have not been focused on in previous research.

2.3.1 Edge based segmentation

Edge based segmentation approaches start by detecting edge and boundary points, followed by grouping the points that are inside the boundaries into a common region (Huang & Menq, 2001). The borders of each region can be defined based on calculating curvature, which can be estimated using PCA on the neighbouring points of each point (Nurunnabi et al., 2012). However, this relies on finding closed boundaries, which is not always possible, especially with sparse and heterogeneous point clouds.

2.3.2 Robust model fitting for the segmentation of planes

Similar to the estimation process for local saliency features, RANSAC is still one of the most popular approaches in detecting and segmenting planes in point clouds. RANSAC was initially developed for robust model fitting in computer vision, however it has been modified for plane detection in point clouds (Bauer et al., 2003; Boulaassal et al., 2007; Previtali et al., 2014; Vosselman & Maas, 2010).

As illustrated previously (section 2.2.3.1), RANSAC works by iteratively finding a minimal sample set, calculating the parameters based on this set, and determining the number of inliers based on the distances of the points to the model based on the calculated parameters. The group with the largest inlier or consensus set is then selected. If the number of the largest consensus set is larger than a pre-defined threshold, this group will be considered as the representation of a planar feature and points in this group are removed from the point clouds. The RANSAC process is repeated multiple times to extract multiple planes. The process is terminated when the likelihood of obtaining a more accurate plane becomes low (Boulaassal et al., 2007).

Bauer et al. (2003) introduced a method based on RANSAC paradigm to detect building façades in dense point clouds. In this case, the 3D point clouds are not captured by laser scanning, but obtained using image matching. Boulaassal et al. (2007) applied RANSAC to detect planar features in TLS point clouds and archived the outputs with 90% accuracy in their case study datasets. Deschaud and Goulette (2010) concluded that large planar features in noisy point clouds can be efficiently detected using RANSAC, but becomes very slow in detecting small planar features. Previtali et al. (2014) proposed another variant of RANSAC that utilises the normal vectors of points as well as the spatial proximity between points in detecting planar features. By taking these factors into account, the influences of the distance threshold parameter may be reduced.

Nevertheless, according to Tarsha-Kurdi et al. (2007), the outputs of the robust model fitting approaches are sensitive to the input parameters. Furthermore, these thresholds can only be empirically determined (Vosselman & Maas, 2010), which means that it requires a lot of testing to determine the ideal parameters for RANSAC. According to Fujiwara et al. (2013), point clouds with different point densities require different distance threshold values in order to obtain accurate outputs. Consequently, problems may arise with MLS point clouds as point densities of different features in the same MLS point clouds vary greatly. In the case where the normal vectors are taken into account in the segmentation process, the under-segmentation problem may arise due to the issues with normal vector estimation (section 2.2.3).

2.3.3 Region growing for the segmentation of planes

Region growing is one of the most popular approaches in detecting and segmenting planar features in MLS point clouds. Region based methods group connected points that have similar characteristics, such as locally planar or smooth normal vectors of a surface (Vosselman & Maas, 2010). Region growing starts by specifying a seed point (or a seed plane). The choice of seed for the new region plays a very important part in the accuracy of the final outputs of the Region growing process.

The seed plane can be determined by analysing the planarity of a random point in the point cloud and its neighbours using RANSAC or Hough transform (Vosselman et al., 2004). When a group of coplanar points has been determined, this group is considered as the seed plane for expansion and the mathematical parameters of the local plane will be estimated. In ideal circumstances, a seed point should not be located near the edge or the boundary of the plane to ensure good results from the region growing. Rabbani (2006) determine seed points based on the residuals of local plane fitting, as points that have small residual value are normally not on the edge between two planar features. In contrast, seed points are chosen based on the curvature of the point in Nurunnabi et al. (2012) as points that have small curvature values are considered as belonging to a planar surface. Nurunnabi et al. (2012) also suggested that edge points need to be removed from the point cloud before performing the region growing process to aid in these criteria.

Rabbani (2006) proposed a region growing technique where a neighbouring point will be considered belonging to the same plane as the seed point if it has a similar normal vector as the seed point, and has a residual for the plane fitting smaller than a pre-define distance threshold (r_{th}). In order to check if a vector is similar to another vector or not, the angular difference or the bias angle needs to be calculated. According to Wang et al. (2001) this value can be calculate using the following equation:

$$A = \arccos\left(n_1^{\mathrm{T}}\, n_2\right) \qquad \text{(eq. 5)}$$

Where A is the angular difference; n_1 is unit vector 1 (i.e. the normal vector of plane 1) and n_2 is unit vector 2 (i.e. the normal vector of plane 2).

If A is smaller than a pre-defined value (θ_{th}), vector 1 and 2 will be considered to have similar orientations. Both of the threshold values (r_{th} and θ_{th}) have to be manually determined. Once all the possible neighbouring points have been assigned to the seed point, a new seed point is selected. This process is repeated until all of the points in the point cloud are assigned into their regions.

Similarly, Nurunnabi et al. (2015) proposed that a neighbouring point will be considered to belong to the same plane as the seed point if it has a similar normal vectors as the seed point, as well as the Euclidian Distance (*ED*) between the seed point and the orthogonal distance (*OD*) of the point to the seed point in the normal direction is smaller than the defined thresholds (i.e. ED_{th} and OD_{th}). These two thresholds can be automatically defined based on the nature of the seed point and its neighbouring points. For instance, the threshold ED_{th} is set equal to the median value of all the distances from each neighbouring point to the seed point, and the threshold OD_{th} is defined based on the median and median absolute deviation of the orthogonal distances between neighbouring points and the best-fit-plane of the seed point as discussed (see Nurunnabi et al. (2015) for more details). Again, the process is repeated until all neighbouring points are assigned to a seed point, and no new seed points can be selected.

The outputs from the Region Growing approaches are heavily dependent on the accuracy of the local saliency feature estimation and the determination of local neighbourhood. The sparseness and heterogeneity of the MLS point clouds leads to special challenges for this process (see section 2.2.3). Consequently, both (over-segmentation and under-segmentation) may occur in the final outputs when applying these approaches to detect and segment planar objects in MLS point clouds.

2.3.4 Scanline based segmentation of planes

Scanline based approaches are based on the principle that points from the same scanline segment belong to the same planar surface. It starts by determining different scan profiles from the data, and then groups neighbouring scan profiles that have similar properties. A fast method for the detection of planar features and segmentation of range images into planar patches was proposed by Jiang and Bunke (1994). For this method, each row in a range image is considered to be a scanline. Then, these scanlines are broken into different smaller scan profiles. Groups of three scan profiles are selected as seed regions for the region growing process if they meet three conditions (Hoover et al., 1996). These are: (1) the length all of three scan profiles must be larger than a pre-defined threshold; (2) these scan profiles must overlap with each other for a predefined percentage; and (3) the distances of every pair of points on any two segments have to be smaller than a specific pre-defined distance. Another scanline based methods that aims to segment planes in ALS point clouds was proposed by Sithole and Vosselman (2003). The captured point cloud is split into two groups of orthogonal profiles running along the X and Y direction. The profiles are formed based on the differences between the height or slope of the neighbouring points. The different surface segments are determined by overlaying all X and Y profiles, and creating segments where profiles overlap. The effectiveness of this approach depends heavily on the density of the point cloud.

Unlike the unorganised point clouds captured using TLS or other devices, point clouds captured from the 2D scanner(s) MLS systems can be considered as partly organised. Thus, scanline based segmentation is a suitable approach to detect and segment planar features. Nevertheless, MLS point clouds have different characteristics compared to TLS point clouds, ALS point clouds and range images data. Cabo et al. (2015) proposed a method to segment plane objects in MLS point clouds. There are two stages in this process: (1) form scan profiles and (2) form planar features. In the first stage, the information about the points belonging to scanlines is utilised to isolate different scanlines from the captured point cloud. Next, different polylines are generated from the determined scanlines based on the distances between two consecutive points of the same scanline (i.e. obtained by setting a polyline gap threshold value to create a continuous section). Finally, different scan profiles are formed by applying the 3D version of the Douglas Peucker algorithm which splits a polyline into different line segments (Ebisch, 2002; Douglas & Peucker, 2011). The Douglas Peucker algorithm simplifies a polyline by introducing a distance threshold parameter d_{dp}. It starts by marking the first and the last point of the polyline as the two end points of the first line. Then, the distance from all of the intermediate points to this line are computed. If all the computed distances are smaller than the distance threshold, this polyline will be considered as a single line formed by the two end points. If there are one or more points that have distances larger than the distance threshold, the point that has the furthest distance to the line will be set as a new node. At this stage, the original line will be divided into two different lines: (1) a line formed by the first node to the new node and (2) a line formed by the new node and the end node. This process is recursively performed until no new lines are formed.

After obtaining all the scan profiles, the scan profile that has that longest length will be selected as the seed. Next, any scan profiles that meet three criteria will be added to the recent region, and later be treated as a new seed. These three specific criteria are: (1) they belong to the scanlines before and after the scanline of the current seed; (2) have similar directions with the seed line; and (3) the distance D between the beginning or the end points of the seed line and the scan profile is smaller than a threshold D_{be}. There are 8 different cases regarding condition 3, as shown in Figure 2.6. This process will be terminated when all of the scan profiles are assigned to their planar features. Similar to RANSAC, all of the required parameters for this algorithm need to be empirically determined.

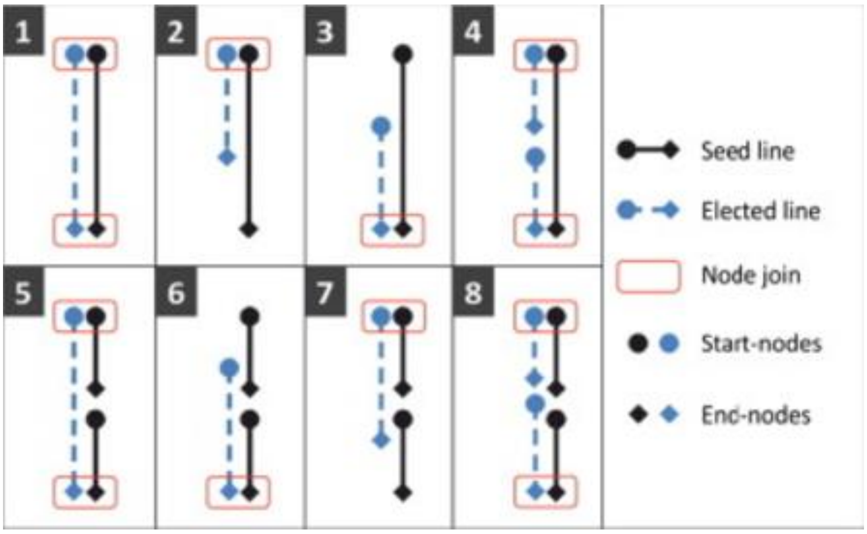
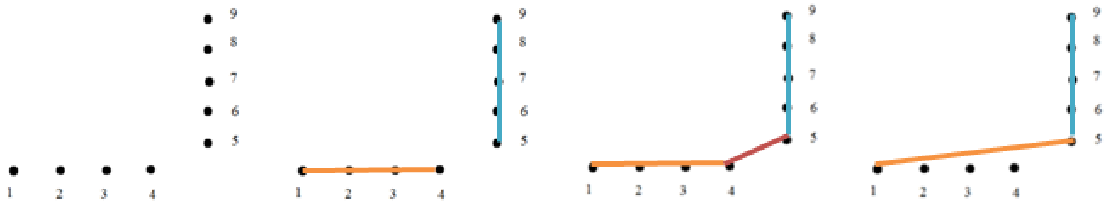


Figure 2.6 Seed lines selections; Adapted from Cabo et al. (2015)

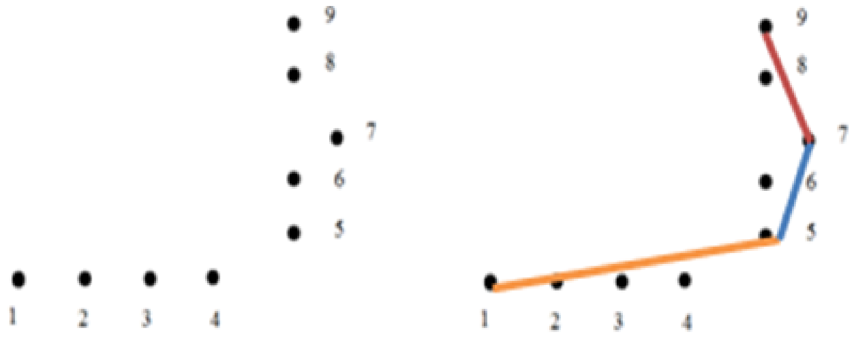
As previously discussed, among all of the scanline based approaches, only the method introduced by Cabo et al. (2015) seems to be suitable for the plane detection of the MLS captured point cloud as it was specifically proposed to detect planes in MLS point clouds. Nevertheless, it has several limitations. First of all, in the scan profile forming step, the 3D version of the Douglas Peucker algorithm is sensitive to the d_{dp} parameters. Figure 2.7 shows an example of having different outputs with different values for d_{dp}. In this example, there is a polyline that has 9 points. Ideally, this polyline should be split into two scan profiles. Scan profile 1 (i.e. the orange line) contains points 1 to 4 and scan profile 2 (i.e. the blue line) contains points 5 to 9 (Figure 2.7b). After applying the 3D version of the Douglas Peucker algorithm, the polyline may be split into two or three scan profiles (Figure 2.7c and Figure 2.7d) depending on the d_{dp}. For the case with the three scan profiles, it is simple to remove the “virtual scan profile” by setting a minimum number of points for a scan profile (e.g. a scan profile must have at least 3 points). For the case of two scan profiles, the 5th point should not be assigned to scan profile 1. This may lead to differences in the calculated plane parameters of the region formed by scan profile 1 with the correct plane parameters or an incorrect equation in the least-squares adjustment models. Similar to other existing methods, there is no guideline for the setting of this parameter for a particular dataset. It needs to be empirically determined.



(a) (b) (c) (d)

Figure 2.7 Effects of d_{dp} parameters on the outputs of 3D version of the Douglas Peucker algorithm; different colours indicate different scan profiles: a) input scanline; b) correct output with colours indicating the two extracted scan profiles; c) incorrect output with colours indicating the three extracted scan profiles and c) incorrect output with colours indicating the two extracted scan profiles

Moreover, the 3D version of the Douglas Peucker algorithm has no resistance to outliers. Indeed, as can be seen from Figure 2.8a, the 7th point is an outlier. Depending on the value of d_{dp} parameters, the blue scan profile maybe split into two different scan profiles as shown in Figure 2.8b.



(a) (b)

Figure 2.8 Outputs from Douglas Peucker algorithm with the present of outliers: a) inputs and b) outputs with colours indicating the three extracted scan profiles.

Finally, the parallelism condition is not enough to detect adjacent planes (i.e. building facades and windows of the same building) that have similar orientations since their scan profiles are parallel to each other (Figure 2.9).

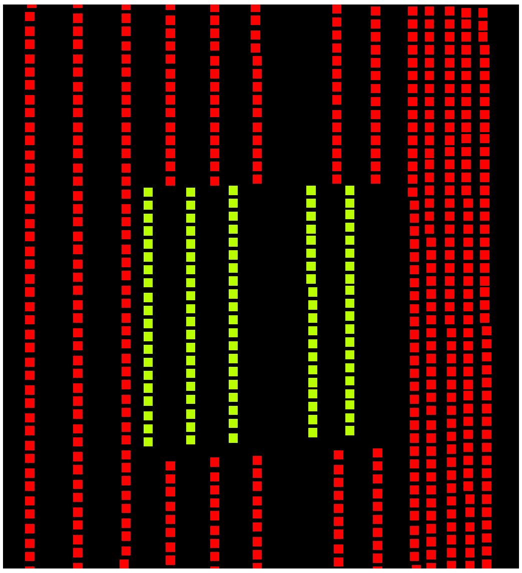


Figure 2.9 Examples of scan profiles of a building façade (red) and a window (yellow)

2.4 Summary

In this chapter, literature about the point density of MLS point cloud were reviewed and discussed. Firstly, a definition of a sparse and heterogeneous point cloud is presented. Then, literature about the local saliency features estimation, planar features extraction and segmentation is reviewed and discussed. As highlighted, the sparseness and heterogeneous of the captured MLS point clouds may lead to special challenges with the discussed point clouds processing processes. As a result, there is a need to improve or propose new approaches to deal with these issues.

CHAPTER 3 DETECTION AND SEGMENTATION OF PLANAR FEATURES IN THE SPARSE AND HETEROGENEOUS MLS POINT CLOUDS

In order to overcome the limitations of the existing local saliency feature estimation and current plane detection and segmentation approaches, a novel plane detection and segmentation method is proposed in this chapter. It is based on the fact that points in the same scan profile belong to the same planar surface. This proposed method utilises the planarity value between neighbouring scan profiles to detect and segment planar surfaces from the MLS point clouds. Moreover, a new local saliency feature, namely a direction vector, is introduced in order to split the different scan profiles within the MLS point clouds. As the parameters of the different steps partly influence each other, the chapter closes with an in-depth discussion of all parameters involved.

3.1 New local saliency feature - direction vector

3.1.1 Definition of a direction vector

Unlike TLS point clouds or point clouds obtained using other methods (e.g. photogrammetry), the captured MLS point clouds data can be considered as partially organised as points are captured in a sequential manner. Due to the fact that, by definition, points on the same scan profile belong to the same planar surface, it can be assumed that a captured point cloud should be segmented into different scan profiles before detecting and segmenting planar features. The existing local saliency features such as normal vectors and curvature seem not to be suitable for this task as often the nearest neighbourhood forms a line, giving an inadequate definition of these saliency features for a surface. Therefore, a new local saliency feature, namely the direction vector, is introduced. The direction vector of a point is considered as the direction vector of the scan profile that this point belongs to. Figure 3.1 shows examples of the direction vectors of points on the same scanlines.

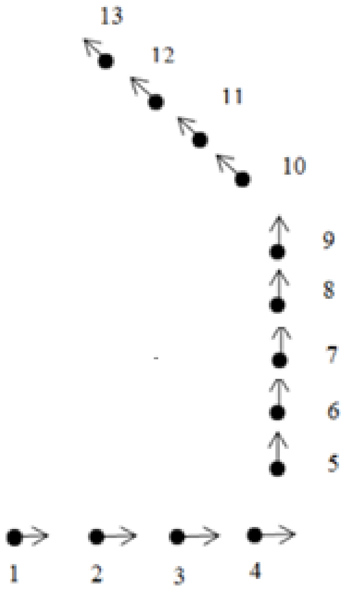


Figure 3.1 Direction vectors of points

3.1.2 Outlier detection using a modified RANSAC algorithm

Based on the discussion in the previous section, although the proposed neighbourhood selection algorithm predicted to outperform KNN and FND algorithm in many cases, it still cannot assure that the neighbourhood is free from outliers. Therefore, an additional processing step needs to be performed to remove outliers from the selected points. Thus, a modified RANSAC (mRANSAC) algorithm that aims to remove outliers from the selected points is introduced. An overview of the workflow for this algorithm is illustrated in Figure 3.2.

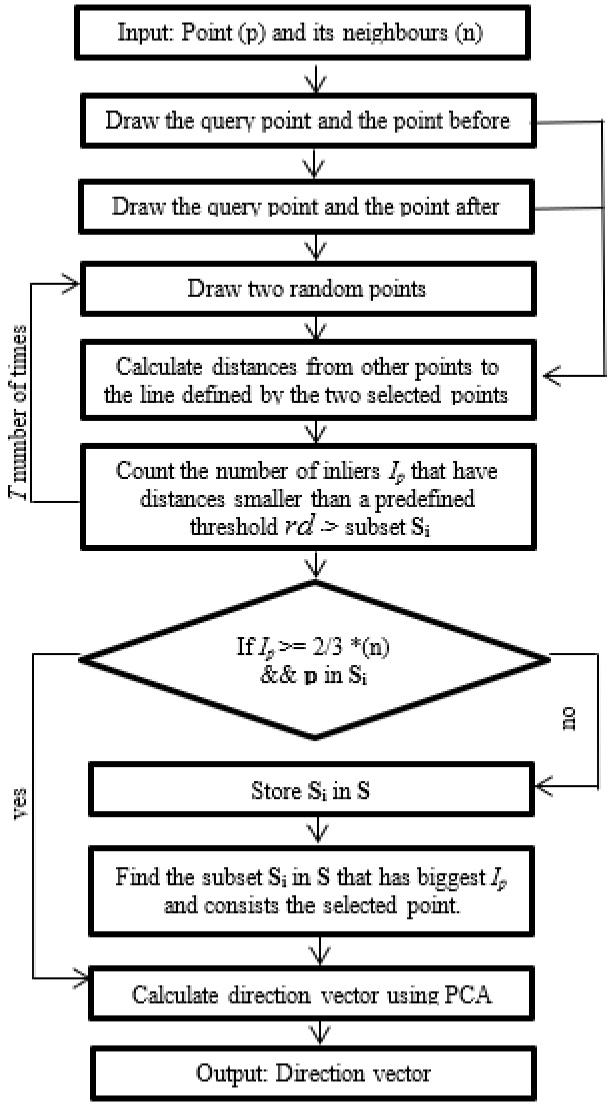

Input: Point (p) and its neighbours (n)
Draw the query point and the point before
Draw the query point and the point after
Draw two random points
Calculate distances from other points to the line defined by the two selected points
Count the number of inliers I_p that have distances smaller than a predefined threshold rd -> subset S_i
T number of times
If I_p >= 2/3 *(n) && p in S_i
yes
no
Store S_i in S
Find the subset S_i in S that has biggest I_p and consists the selected point.
Calculate direction vector using PCA
Output: Direction vector

Figure 3.2 Modified RANSAC algorithm workflows: *p* is the query point, *nnp* is the neighbourhood of *p*, I_p is the number of inliers in each subset $\mathbf{S_i}$ for each iteration *T* and *rd* is the distance threshold

Unlike the original RANSAC, this proposed algorithm does not select samples completely at random. Based on the fact that the two consecutive points of the query points (e.g. captured before and after) have a high probability to be correct inliers, for the first two iterations the selected samples are the query point and one of these two points. Specifically, for the first iteration, the sample is the query point and the point captured before it in the same scanline; for the second iteration, the sample is the query point and the point captured after it in the same scanline. Similar to the original RANSAC the size of the consensus set is the number of points that have distances to the line formed by the selected sampling points smaller than a threshold *rd*. The distance d_i of a point p_i to the line formed by two sampling points p_{s1} and p_{s2} is calculated by:

$$d_i = \frac{|(p_i - p_{s1}) \times (p_{s2} - p_{s1}|}{|p_{s2} - p_{s1}|} \qquad \text{(eq. 6)}$$

where × denotes the cross product and || denotes the length.

If all of the selected points are inliers, the number of inliers will equal the number of points. In this case, the algorithm will terminate, and all of the neighbouring points will be used to estimate the direction vector of the query point. In order to improve the processing time the mRANSAC will terminate at any time when the number of inliers is equal to two thirds of the number of neighbouring points and the query point is in the recent consensus set. This termination condition is based on the fact that in the local neighbourhood selection step, it is believed that normally there are always at least 50% of correct neighbouring points selected. If there is no noise, a maximum 2 iterations are normally needed to get the output.

If this is not the case, e.g. when the query point is located close to an edge of a plane or if the query point is an outlier, further iterations will be performed, and samples will be selected randomly from any neighbouring points captured either before or after the query points. The reason behind this technique is the assumption that points on the same side of the query point have a higher likelihood to belong to the same scan profile. The size of the consensus set for each iteration is determined, and the process is iterated a number of times.

Finally, the subset that has the biggest number of inliers and contains the query point will be assigned as the "clean" subset. The reasoning behind these conditions is expanded on in Figure 3.3. There may exists the case that for the query points (red point), the number of "inlier" points (black points) in the selected local neighbourhood of points is less than the number of "incorrect inlier" points (purple points). Specifically, Figure 3.3 illustrates a query point with its neighbouring points that are selected from three different scan profiles SP1, SP2 and SP3 (with the *nnp* parameter set to 8). In this case, the maximum consensus consists of seven purple points, while the correct consensus only has six points including the query point. Consequently, if this criterion is not taken into account the query point will be considered to have the same direction vector as the purple points, which is incorrect. One may argue that this issue can also be solved by reducing the value for the *nnp* parameter (e.g. to 6 or 7). Nevertheless, it may not be a good solution in similar situations where the number of black points (correct consensus points) is just 5 or less. Furthermore, the aim is to propose a method that is not sensitive to the parameters. In other words, the required parameters for the proposed method can be fixed for any sparse and heterogeneous MLS point cloud dataset. By introducing this criterion, the influence of the value of the *nnp* parameter on the quality of the outputs of the proposed approach is significantly reduced.

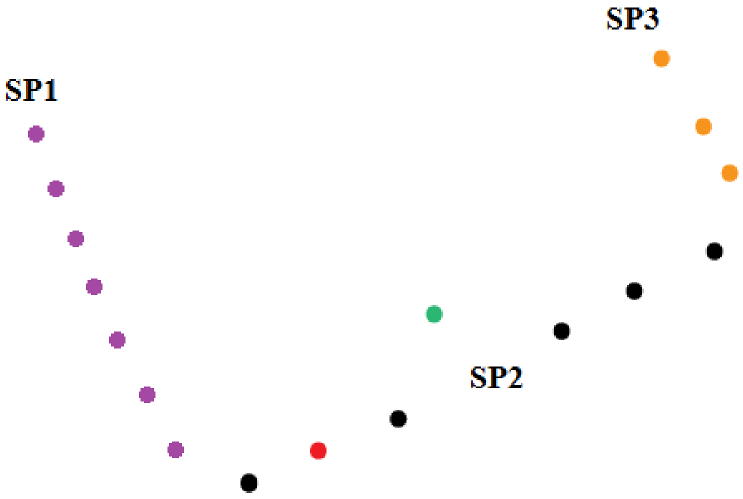


Figure 3.3 Example of direction vector estimation, different colours indicate different scan profiles: purple points are points of scan profile SP1; Black points and the red point are points of scan profile SP2; Orange points are points of scan profile SP3 and green point is outlier points; the red point is the query point.

Practically, the case may also occur that the query point lies very close to the intersection of the two scan profiles as shown in Figure 3.4. In this case, the distance *d* from the query point (i.e. point 9) to both of the scan profiles may be smaller than the *rd* value. Consequently, there will be two subsets which have the same number of inliers, as well as containing the query point. Hence, in order to make an accurate decision, the linearity values of these two subsets will be compared. The linearity value of a group of points is considered as the second smallest eigenvalue of the covariance matrix of those points. The linearity value of a perfectly collinear group of points is zero. The subset that has the smaller linearity value will be assigned as the 'clean' subset. By doing this, the direction vector of point "9" will be estimated as shown in Figure 3.4(b) as the linearity values of group of denoted by "*" points is smaller than group represented by "o" points. This condition not only improves the quality of the estimation outputs, but also compensates for the sensitivity of the *rd* parameter.

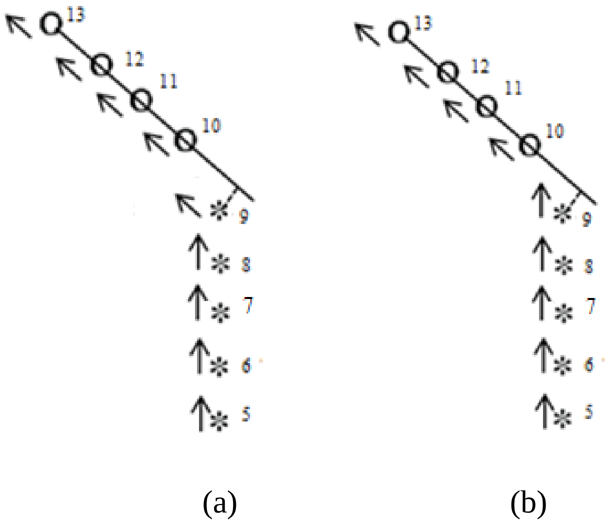


Figure 3.4 Example of direction vectors of points with different *rd* parameters: (a) direction vector when *rd* is equal or bigger than *d*; (b) direction vector when *rd* is smaller than *d* (adopted from Nguyen, Helmholz, Belton, and West (2015))

One of the advantages of the direction vector estimation when compared to normal vector estimation is that the direction vector estimation deals with 2D data instead of 3D data (i.e. the process deals with points on individual scanline instead of the whole 3D point clouds). As a result, the complexity of the algorithm can be reduced. It makes the outputs from this process more reliable and accurate.

3.1.3 Direction vector estimation

Similar to the process for the estimation of a normal vector of a point, the first step is to select the neighbouring points of the query points. These selected points need to belong to the same scan profiles as the query point. PCA is then applied to these groups of points. The direction vector of the query point is considered as the eigenvector corresponding to the largest eigenvalue of the covariance matrix formed by the query point and its neighbours. Vector E1 (i.e. the first eigenvector) in Figure

3.5 denotes the direction of the largest possible variance (the direction vector) of the group of points.

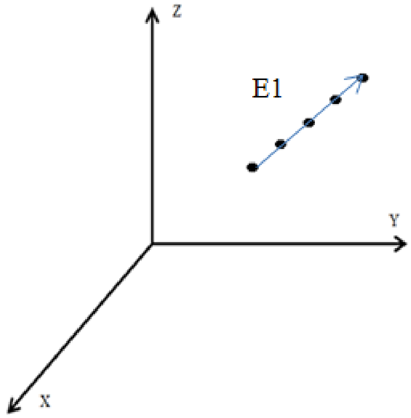


Figure 3.5 Direction vector of a group of points using PCA

In order to obtain the correct direction vector of the query point, the selected neighbouring points need to belong to the same scan profile. Nevertheless, the popular local selection algorithms such as KNN and FDN cannot assure that only the correct neighbouring points are selected. Depending on the predefined parameters for the used algorithm (e.g. *k* parameter in KNN or *fd* parameter in FDN), the neighbouring points are often selected from different scanlines. This may lead to difficulty in removing outliers for the direction vector estimation process. In fact, it is not feasible to automatically select the correct neighbouring points for all the points in just one step. Hence, this step is aimed at selecting the local neighbouring points belonging to the same scanline of the query point (i.e. minimise the number of outliers). Thanks to the property of the MLS point clouds (i.e. points are captured in sequence and many MLS systems also provide the information about the scanline of the captured points) it is easy to select the local neighbouring points of the query points which belong to the same scanline using either KNN or FND on each scanline.

Nevertheless, due to the property of the MLS point clouds such as the particular case as shown in Figure 3.6, the correct neighbouring points of the query point in scanline (i.e. the yellow point inside the red circle) are not just the points that are close in terms of distance, which are the green points. Hence, if the local neighbouring points is determined based on the distance criteria (i.e. using KNN or FDN algorithm), depending on the *k* or *fd* parameter, it may lead to the case that most of the selected points do not belong to the correct neighbourhood of points for the query point. For instance, if the *fd* is inside “the orange circle”, there will only be one correct neighbouring point selected. This circumstance will mainly occur with points near the boundary of two consecutive scan profiles (i.e. two neighbouring scan profiles of the same scanline) that have a large bias angle and different sampling intervals between surfaces. In this case, the selection criteria based on distance is not suitable to maximize the number of correct local neighbourhoods. The aim of any local neighbourhood selection algorithm is to maximize the number of correct neighbouring points.

Therefore, a simple but efficient local neighbourhood selection method that determines the neighbourhoods of the query point-based on the points’ index is introduced. It starts by setting a *nnp* parameter, which defines the number of points that are captured as neighbouring scanline points before and after the query point. The *nnp* parameter in this approach acts similar to the *k* parameter in KNN algorithm.

Using this proposed approach, the issue described in Figure 3.6 can be avoided. Indeed, in the example shown in Figure 3.6, if the *nnp* parameter is set to five, *all* five yellow points and five green points will be selected. Theoretically, if there is no noise in the captured point clouds and the value of *nnp* parameter is smaller than the number of points of the scan profile, then the number of inliers in the selected local neighbouring points is always larger than the number of outliers. In other words, the percentage of outliers is always less than 50%. If the value of *nnp* parameter is larger than the number of points of the scan profile, all of the inliers will be selected. As a result, the number of correct neighbouring points is always maximally selected. These selected inlier points are the ideal inputs for the outlier detection process.

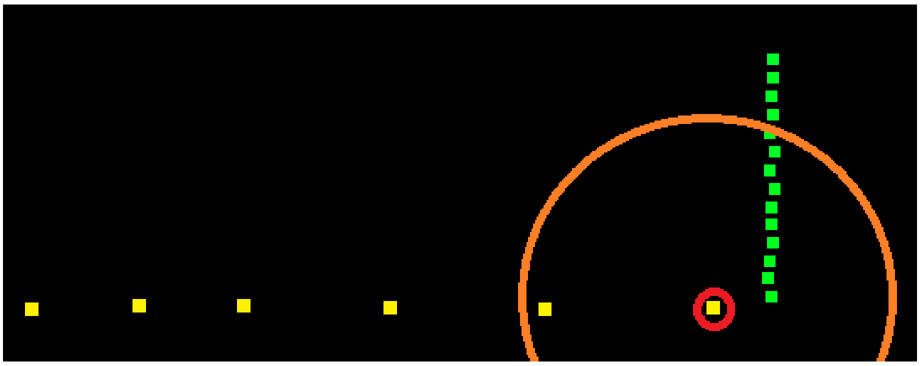


Figure 3.6 Problems in selecting local neighbouring points using KNN or FDN in a particular case with a real captured MLS point cloud; different colours indicates different scan profiles.

Finally, using this approach, no data structures such as a K-D tree or an Octree is required because of the underlying order of the point. The information about the scanline number is normally provided by the MLS system. Alternatively, it can be derived based on the laser positions and the point index.

3.2 The proposed plane detection and segmentation method based on the planarity of groups of scan profiles (PSPS)

Based on the new local saliency feature and the fact that points of the same scan profile belong to the same planar surface by definition, a new planar features detection and segmentation method based on the planarity of groups of scan profiles is introduced. The inputs for this method are the points of the captured MLS point cloud and their estimated direction vectors. This method has three main steps: 1) forming scan profiles; 2) grouping scan profiles and 3) plane detection and segmentation based on the planarity of groups of scan profiles. The workflow of the method is outlined in

Figure 3.7. In step 1, the scan profiles are segmented from the MLS point clouds based on the direction vector of the points. This approach is considered to be more robust against noise than the 3D version of Douglas Peucker algorithm as previously discussed (section 2.3.4). After grouping neighbouring scan profiles that are parallel to each other into the same groups, planar features can be detected and segmented based on the planarity value of the neighbouring scan profiles.

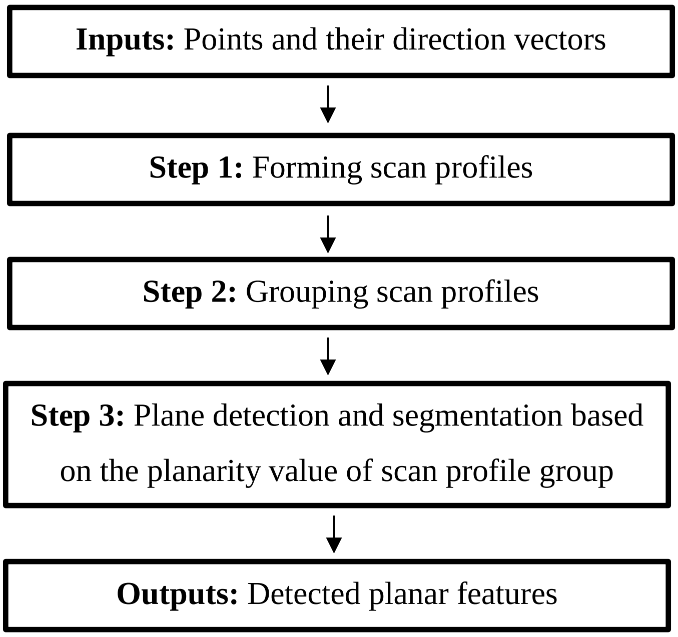

Inputs: Points and their direction vectors
Step 1: Forming scan profiles
Step 2: Grouping scan profiles
Step 3: Plane detection and segmentation based on the planarity value of scan profile group
Outputs: Detected planar features

Figure 3.7 Plane detection and segmentation based on the planarity value of the scan profiles method workflows.

3.2.1 Forming of scan profiles belonging to a plane segment

Based on the definition that points on the same scan profile belong to the same planar object, different scan profiles need to be segmented from the point clouds first. The inputs for this process are the estimated direction vectors of points. In general, neighbouring points have a high likelihood to belong to the same scan profile. Moreover, in practice the case may occur that points of two different scan profiles have similar direction vectors (e.g. different façade elements). Figure 3.8 shows an example of neighbouring points of two different scan profiles (e.g. blue and black) which have similar direction vectors.

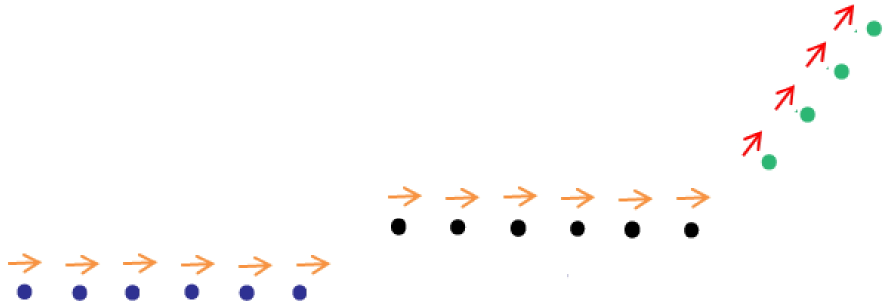


Figure 3.8 points and their direction vector; different colours indicate different scan profiles

Thus, a scan profile growing algorithm that utilises the direction vectors of points, the order of the captured points, and the linearity values of neighbouring points is proposed. This process is as follows: **P** denotes all points in the point cloud and **DR** denotes the direction vectors of points. To begin, the point p_i that has the lowest index is chosen as the seed point for a new scan profile. This point p_i is added to the recent seed group **S**$_\mathbf{c}$ and removed from the point cloud **P**. In this research, it is assumed that the neighbouring point p_{i+1} belongs to the same scan profile with p_i if it satisfies the following conditions: 1) has similar direction vector with p_i (i.e. the value A_i between DR_i and DR_{i+1} is smaller than a threshold A_{th}); 2) has the orthogonal distances DL_i to the line **L** smaller than a threshold DL_{th}, where line **L** is defined as going through p_i and have the same direction vector with p_i.

In this process, the neighbouring points are determined using a similar method that was previously utilised to estimate the direction vector (3.1.2). Value for angle A_i is calculated between the direction vectors of p_i and p_{i+1} which is calculated using

$$A_i = \arccos(DR_i^T DR_{i+1}) \qquad \text{(eq. 7)}$$

the distance DL_i which is calculated by

$$DL_i = \frac{|(p_{i+1} - p_i) \times (DR_i|}{|DR_i|} \qquad \text{(eq. 8)}$$

where × denotes the cross product and || denotes the length.

If A_i and DL_i are smaller than their corresponding threshold, point p_{i+1} will be added to the seed point group $\mathbf{S_c}$ and removed from **P**. Otherwise, the next neighbouring points p_{i+2} will be chosen as the next “potential” point. After no further points are added to the current scan profile, $\mathbf{S_c}$ and **P** are saved and a new seed is chosen. This process is iterated until all points are assigned to their particular scan profiles. The outline of this method is summarised in algorithm 1.

Algorithm 1:

```
Input: X, Y, Z coordinates of points in point clouds P and their direction vectors DR
while P is not empty do
    Find initial seed point $p_i$ from P which has the smallest index
    Insert $p_i$ into $S_c$ and remove form P.
    for each seed point in $S_c$ do
        for each neighbouring point of $p_i$
            Calculate the angle $A_i$ between $DR_i$ and $DR_{i+1}$, the distance $DL_{i+1}$ between
            $p_{i+1}$ and the line L go through $p_i$ and is parallel with $DR_i$
            if ($A_i <= A_{th}$) & ($DL_{i+1} <= DL_{th}$)
                insert $p_{i+1}$ into $S_c$ and remove form P.
            end if
        end for
    end for
    insert $S_c$ to SP
    clear $S_c$
end while
Output: a scan profile list SP
```

3.2.2 Grouping of parallel scan profiles

Mathematically, two lines form a plane if they are parallel or intersect with each other. However, due to the scanning geometry of the MLS point clouds, two intersected scan profiles can lie on the same scanline, but do not belong to the same captured planar surface. In this case, two scan profiles in the MLS point clouds only have the possibility to belong to the same planar features if they are nominally parallel to each other. Hence, after forming the scan profiles, these are assigned into different groups based on two criteria: 1) they belong to adjacent scanlines, and 2) they have similar direction vectors. The scanline line numbers are utilised to check whether two scan profiles meet the first criteria or not. Meanwhile, the second criteria can be checked by calculating the angle between their direction vectors using eq.7. Figure 3.9 shows an example of the grouping of different neighbouring scan profiles into groups based on their direction vectors.

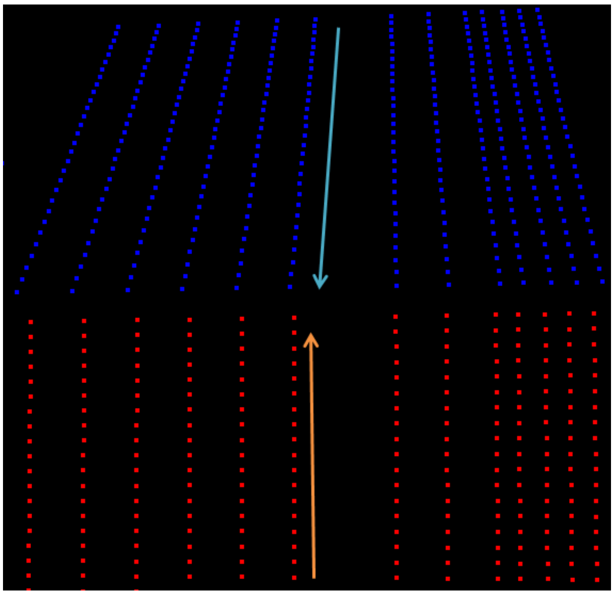


Figure 3.9 Grouping of different scan profiles into groups: the orange arrow indicates the direction vectors of the red scan profiles and the light blue arrow indicates the direction vectors of the blue scan profiles

After the grouping of different scan profiles, each of the groups may contain scan profiles of a single planar object, or each of them may contain more than one planar object if these planar objects are next to each other and have similar orientations. An example of such is given in Figure 3.10, which occurs in cases such as recessed doorways or windows. At this stage, it is impossible to determine how many planar objects are in each group. As a result, another step needs to be perform to accurately detect and segment the correct planar objects from within each group.

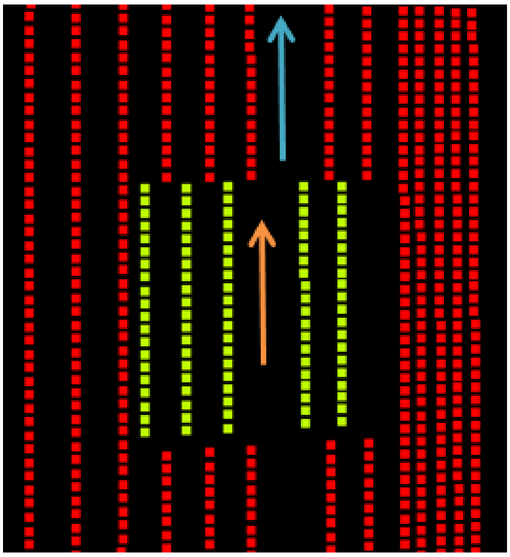


Figure 3.10 Example of two neighbouring planar features having similar orientation: different colours indicate different planes

3.2.3 Plane detection and segmentation based on the planarity value of the scan profiles groups

This method aims to detect and segment planar objects based on the planarity value between different scan profiles. In this research, the planarity value of a group of points is considered as the smallest eigenvalue from the covariance matrix calculated on these points. Under perfect conditions, the planarity value of a group of points representing a planar object will be equal to zero. Nevertheless, these conditions cannot be met in reality since the planar objects are normally not perfectly smooth due to errors and noise existing in any measurements.

In addition to the planarity value, the assumption is made that each plane in the captured MLS point clouds are formed by at least two scan profiles due to the constraint that at least two scan profiles are needed in order to resolve the planar object parameters. Therefore, a plane detection and segmentation method called *plane detection and segmentation based on the planarity of group of scan profiles* (PSPS) is proposed. As discussed in the previous section (3.2.2) each collection of scan profiles which results from the grouping of the scan profiles may contain one or more planar objects. Therefore, the first step in this process is to determine whether a group contains one or more planar object. In order to archive this objective, the planarity value needs to be calculated, and the threshold PL_{th} needs to be set. If the calculated value of the planarity value of a group is smaller than this threshold, all of the points in this group are considered as belonging to the same planar object. If the planarity value is larger than the threshold, further steps are performed in order to determine the different planar objects. This planarity value threshold has the same role as the distance threshold in the RANSAC algorithm.

In practice, different planes may have different planarity values. This depends on many factors, such as the roughness of the plane and the accuracy of the scanner. Consequently, it is not reasonable to set a fix value for the planarity value threshold for the whole MLS point cloud dataset. Hence, an adaptive planarity value is required. In order to achieve this, the assumption is made that the planarity value of a plane is approximately equal to the planarity value calculated from any two scan profiles from this plane. For instance, in Figure 3.11 the planarity values of the green plane and the red plane are assumed to approximate the planarity values calculated from any two of their scan profiles (i.e. $PL(\mathbf{L_i}, \mathbf{L_{i+1}}) \sim PL(\mathbf{L_i}, \mathbf{L_{i+2}}) \sim PL(\mathbf{L_{i+1}}, \mathbf{L_{i+2}}) \sim PL(\mathbf{L_i}, \mathbf{L_{i+1}}, \mathbf{L_{i+2}})$). Based on this assumption, the planarity value threshold of each plane (e.g. green plane) is set dynamically based on the planarity calculated from any of the two scan profiles of this plane (e.g. scan profile $\mathbf{L_i}$ and $\mathbf{L_{i+1}}$).

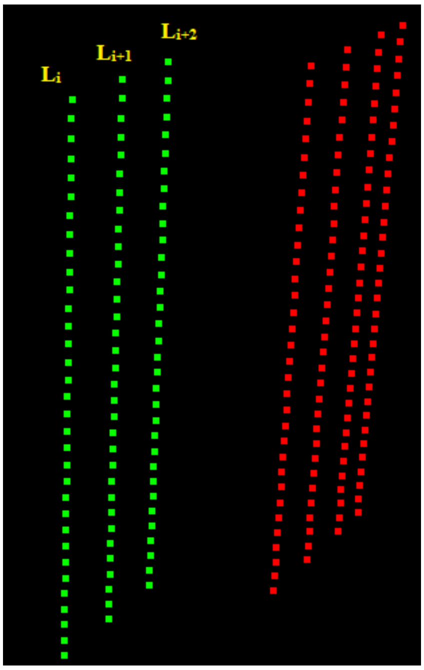


Figure 3.11 Another example of two adjacent planes that have similar orientations

The first step of PSPS is to determine the number of scan profiles in each group. As two scan profiles is the minimum number of profiles required to calculate the plane parameters, it is assumed that if a group has only one or two scan profiles then they represents a single surface. Therefore, any group that has the number of scan profiles less than three will be assigned to a region, and the algorithm will move to another group of scan profiles as input for further processing.

If a scan profile group has at least three scan profiles, the planarity value threshold of each group will be calculated from the first two consecutive scan profiles in the group (i.e. $\mathbf{L_i}$ and $\mathbf{L_{i+1}}$). Based on this value, a planarity threshold is set. After that, the planarity value of all scan profiles for each group is calculated and compare with the planarity threshold. Any group that has the planarity value smaller than its planarity value threshold will be considered as a single planar object. The remaining groups are considered as potentially containing more than one planar feature.

In order to accurately segment the remaining groups, a Region Growing method based on the Planarity of the Line segment (RGPL) algorithm is used. For each of the scan profile groups $\mathbf{G_i}$, RGPL starts by determining the scan profile $\mathbf{L_i}$ that only has neighbours to one side of the scan profile (that is on the edge of the group), and its closest neighbour $\mathbf{L_{i+1}}$ (Figure 3.11). These scan profiles are added to the seed segment group $\mathbf{S_c}$ and the current region $\mathbf{R_c}$ and are removed from the input group $\mathbf{G_i}$. Then, the planarity threshold PL_{th} for this surface will be calculated by applying PCA on the points on $\mathbf{L_i}$ and $\mathbf{L_{i+1}}$ (see section 2.2.2) and multiplying the value by a given scale factor.

Next, the planarity value (*PL*) of points on these scan profiles $\mathbf{L_{i+1}}$ and $\mathbf{L_i}$ and the neighbour on the opposite side of $\mathbf{L_{i+1}}$ ($\mathbf{L_{i+2}}$) will be estimated. If the calculated planarity value is equal to or smaller than the planarity threshold, they will be added to $\mathbf{R_c}$ and $\mathbf{S_c}$, and removed from $\mathbf{G_i}$. The process is repeated similarly for $\mathbf{L}_{i+3}$ and so on until a value is calculated which is larger than the planarity threshold. In this case, $\mathbf{R_c}$ will be saved as a new plane, and $\mathbf{R_c}$ and $\mathbf{S_c}$ will be cleared. The remaining scan profiles will be assign into a new group, and examined to determine whether it represents more than one surface or not. The process is repeated until all scan profiles in all the groups are assigned to different regions. RGPL is illustrated in Algorithm 2 as follows.

Algorithm 2

```
Input: group of scan profiles G1.
while G is not empty do
    while $G_i$ is not empty do
        Find initial seed scan profile $L_i$ from $G_i$ with $L_i$ has neighbours on only one side
        Insert $L_i$ and it nearest neighbour $L_{i+1}$ into $R_c$ and $S_c$, remove form $G_i$. the planarity threshold $PL_{th}$ is calculated from points on $L_i$ and $L_{i+1}$
        for each seed scan profile in $S_c$ do
            if $L_{i+1}$ has neighbour on both side then
                Calculate planarity values $PL$ of points on $L_i$, $L_{i+1}$ and $L_{i+2}$
                if $PL <= PL_{th}$ then
                    insert $L_{i+2}$ into $R_c$ and $S_c$, remove form $G_i$.
                    Remaining scan profiles are put into new group $G_{i+1}$
                end if
            end if
            insert $R_c$ to R
        end for
        clear $R_c$ and $S_c$
    end while
end while
Output: a region list R
```

After RGPL is performed on all of the unsegmented groups, the scan profiles near the boundaries between different planar objects needs to be carefully analysed to determine which surface they belong to in order to have accurate results. Practically, in the scanning area, it may exist the case that there are two planar objects that have similar orientations and are adjacent to each other as shown in Figure 3.12. In this figure, there are two planar objects indicated by different colours. Planar object 1 (i.e. green plane) contains two scan profiles SP1 and SP2. Planar object 2 (blue plane) contains three scan profiles SP3, SP4 and SP5. The middle scan profile SP3 lies very close to the boundary between two surfaces. Both of the planarity values calculated from two green scan profiles SP1 and SP2 with scan profile SP3 and from three blue scan profiles SP3, SP4 and SP5 may be very small, and satisfy the planarity threshold value of each corresponding planes. Hence, depending on the seed scan profile selection order, the middle scan profiles may be assigned to the surface 1 or it may be assigned to surface 2. In order to determine the correct solution, another step is introduced for the scan profile of each planar object which is considered to be an undecided scan profile after processing algorithm 2 (i.e. could belong to either of the adjacent planar object). This would be the case for the middle scan profile in Figure 3.12. The planarity value of the planar object 1 with the undecided scan profile and the planarity value of the planar object 2 with the undecided scan profile are compared. The “undecided” scan profile will be assigned to the planar object that has the smaller planarity value.

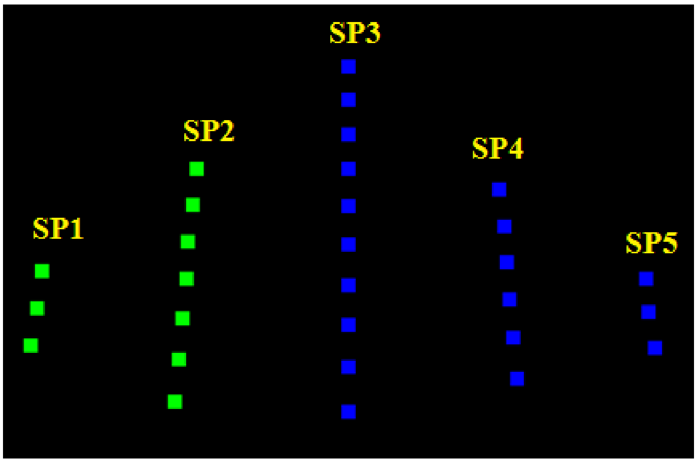


Figure 3.12 An example of two adjacent planar objects that have similar orientations

3.3 Discussions about the required parameters of the proposed method

The required parameters of each step have different impacts to the quality on the plane detection and segmentation outputs. Some of these have insignificant influences, and hence fixed values can be applied for all datasets. On the other hand, some of them can be determined based on the information from the specifications of the MLS system, particularly from the accuracy of the laser scanner(s).

3.3.1 Local neighbourhood selection of neighbours in a scanline

For the local neighbourhood selection, the parameter *nnp* needs to be set. If the value of *nnp* is increased, the probability of having more inliers than outliers can still be preserved. Furthermore, thanks to the proposed mRANSAC algorithm the value of *nnp* parameter will only have a small impact on the outputs of the final results, and as such it can be fixed for any MLS point cloud dataset as previously discussed (section 3.1.2). The only strict condition is that the value for *nnp* must not be set too small (e.g. 1 or 2) as shown in Figure 3.13. In this example, if the *nnp* value is set to 2, the direction of the red point will be considered to be parallel with scan profiles SP4, SP5 or SP7, while the direction of the brown point will either be considered to be parallel with scan profiles SP6 or SP7. It should also not be set to too large (e.g. 10 or larger) to reduce the processing time. Thus, it is suggested to set it equal to 7.

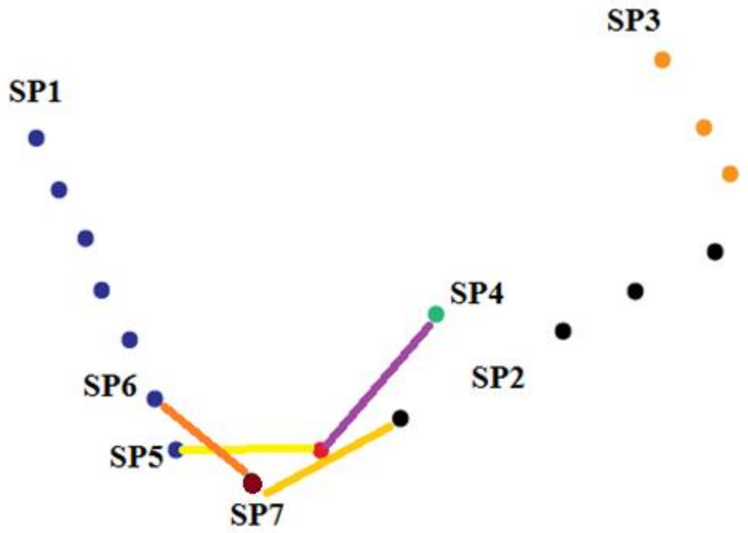


Figure 3.13 Example of *nnp* is set to 2: the direction of the red point can be either considered to be parallel with scan profiles SP4 or SP5, while the direction of the black point can be either considered to be parallel with scan profiles SP6 or SP7

3.3.2 Outlier detection in the estimation of the direction vector

The mRANSAC algorithm used to detect outliers in the estimation of the direction vector require two parameters to be set, the number of iterations *T* and the distance threshold *rd*. In theory, the number of iterations *T* should be set based on the number of selected neighbouring points. Hence, it can be fixed for any MLS point cloud dataset. The aim in choosing the value for *rd* is to determine as many inliers as possible and reduce the processing time. This algorithm will terminate if the number of inliers is larger than two thirds of the number of neighbouring points. Due to imperfect measurements of MLS systems (e.g. the accuracy of the MDL Dynascan S250 and RIEGL VQ-450 is 1 cm and 0.8 cm respectively) the captured points of a scan profile normally do not all lie accurately on a scan profile. They may lie on the same or different side of the scan profile as shown in Figure 3.14. The distances from the captured points to the correct scan profile vary (e.g. points 1, 9 and 12 have the largest distances to the correct scan profile resulting in the worst accuracy value) and are normally equal to or smaller than the accuracy of the MLS system. If there is always at least two points lying exactly on the scan profile, then T_r can be set to the accuracy of the MLS system. Nevertheless, this assumption is not always assured. Therefore, the best candidate points for the process that can be achieved are two points lying on the same side, and have similar distances to the correct scan profile (e.g. point 2 and 3). In this case, the *rd* threshold should not be set equal to the accuracy of the MLS systems as the distance from point 12 to the L3 line (i.e. purple line) is larger than T_r. Hence, the value for *rd* is suggested to be set to twice of the specified standard error for the scanner observations. This is in order to account for the worst case, when all of the neighbouring points lie on one side of the query point (i.e. point 7), as shown in Figure 3.15. In addition, it will also have more inliers in the same iteration (e.g. point 4, 5, 6, 7, 10 and 11 in the case of L3 line in Figure 3.14).

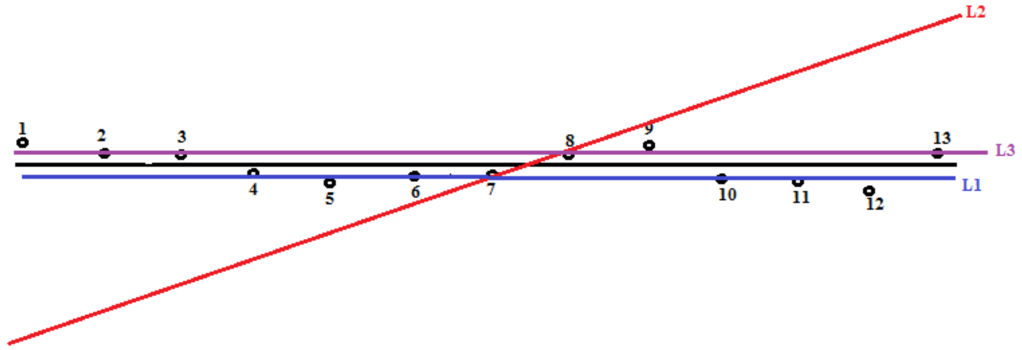


Figure 3.14 Example of different iterations of the mRANSAC to detect outliers of the direction vectors; different colours indicate the line formed by different sampling

points; the black line is the actual line; other lines are formed by different sampling points.

One may argue that by setting the value of *rd* to twice of the accuracy of the scanner, it may lead to outliers being considered as inliers. However, it cannot be assured that a point that has the distance to a scan profile formed by its neighbouring points equal to twice of the accuracy of the scanner is an outlier (e.g. point 7 in Figure 3.15)

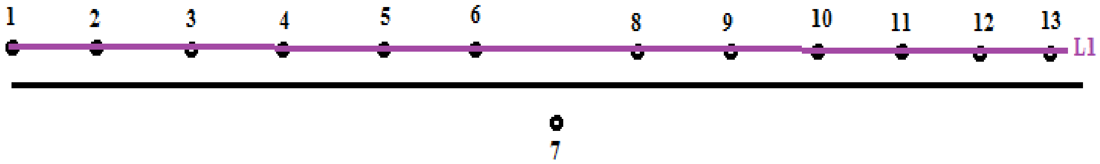


Figure 3.15 Example of the particular case of the query point and its neighbouring points; the black line is the true line; purple line is formed by sampling points

3.3.3 Forming scan profiles and grouping of scan profiles

The step for forming scan profiles requires two parameters, namely the angle threshold A_i and the distance threshold DL_i. The distance threshold DL_i acts similar to the *rd* parameter. As a result, it can also be set to twice of the accuracy of the scanner. Meanwhile, if the dataset is errorless, the A_i parameter can be set to zero. However, due to the roughness of the surfaces and the accuracy of the scanner(s), it can be set to less than 10 degrees as suggested by previous researches for plane detection and segmentation approaches (Cabo et al., 2015; Previtali et al., 2014). This value can also be applied for the angle threshold in grouping scan profiles step.

3.3.4 Plane detection and segmentation

Regarding the scale factor value used to evaluate the planarity threshold, if the scan profile $\mathbf{L_{i+2}}$ does not belong to the same plane as $\mathbf{L_i}$ and $\mathbf{L_{i+1}}$ (Figure 3.11), the planarity value calculated from them will be significantly larger than the planarity value calculated from $\mathbf{L_i}$ and $\mathbf{L_{i+1}}$ (e.g. it is normally more than 10 times larger based on testing which is demonstrated in Chapter 5). On the other hand if the scan profiles $\mathbf{L_{i+2}}$ belongs to the same plane with $\mathbf{L_i}$ and $\mathbf{L_{i+1}}$ the planarity value calculated from them may be slightly larger or smaller than the planarity value calculated from $\mathbf{L_i}$ and $\mathbf{L_{i+1}}$. Hence, the factor value is suggested to be set to 4 for any MLS point cloud dataset based on empirical tests. More analysis about the planarity values are provided and demonstrated concerning a real point cloud dataset in Chapter 5.

Table 3.1 illustrates the standard values for different parameters that are required in this proposed novel plane detection and segmentation method.

Parameters	Values	Comments
nnp	7	Can be fixed for all point clouds
T	10	Can be fixed for all point clouds
A_i	10	Can be fixed for all point clouds
Planarity factor	4	Can be fixed for all point clouds
rd	Twice the specified standard error for scanner observations	Depending on the accuracy of the scanner
DL_i	Twice the accuracy of the scanner	Depending on the accuracy of the scanner

Table 3.1 Standard parameters for PSPS method

3.4 Summary

In this chapter, a new plane detection and segmentation method is presented based on the fact that points on the same scan profile belong to the same plane and utilising the planarity values of different neighbouring scan profiles. Different scan profiles can be extracted from the MLS point clouds based on the proposed local saliency feature – the direction vector of a point. It is discussed how the outputs from the direction vectors estimation are more reliable than the normal vectors estimation outputs. This is due to the fact that the direction vector estimation deals with 2D data rather than 3D in normal vector estimation. By carefully analysing the primitives (e.g. points and scan profiles) near the boundary of different higher primitives (e.g. scan profiles and planes) the impact of the parameter values to the final outputs can be reduced. As a result, one of the significant advantages of the plane detection and segmentation method is most of the required parameters can be fixed or determined based on the specification of the scanner(s).

CHAPTER 4 REVIEW AND DISCUSSION OF MLS POINT CLOUD REGISTRATION APPROACHES INCLUDING MEASURE FOR EVALUATION

Registration is one of the most important tasks in MLS point clouds processing. In this chapter different point cloud matching techniques are reviewed and discussed, especially with regards to analysing potential issues that may be caused by the sparse and heterogeneous nature of MLS point clouds. Until now there has been limited comparisons conducted between the registration results from these point cloud matching techniques dealing with sparse and heterogeneous MLS point clouds. The suitability of current error metrics for measure the quality of MLS registration is also analysed and discussed in this chapter with respect to these point cloud conditions. As a result, a new error metric is proposed, that aims to provide a suitable index to evaluate and compare the quality of the point cloud registration outputs over conventional metrics.

4.1 MLS registration

There are always errors in the absolute position of the MLS captured point clouds due to errors from the loss of GNSS signals and IMU drifts, especially in urban areas. Because of this, registration is one of the most important tasks in MLS point cloud processing. The aim of the registration process is to obtain the transformation parameters in order to align the point clouds to their desire positions. In some cases MLS point clouds (i.e. *slave* point clouds) are required to be registered onto a *master* point cloud (Previtali et al., 2014). In other cases, MLS point clouds are required to be aligned onto predefined or existing *master* models. These models can be obtained from other data source such as 3D maps or geographic databases.

Unlike the registration process for TLS or ALS point clouds, a *slave* dataset should not be rigidly registered to a *master* dataset due to the fact that the registration errors are not constant for the whole dataset. Varying of the transformation parameters caused by drift errors along the trajectory which need to be taken into account (Monnier et al., 2013). Monnier et al. (2013) proposed a MLS registration approach that aims to correct the system's position for the entire trajectory. However, this approach only considers the errors in translation and neglects the rotation errors.

Moreover, as only vertical planar surfaces were taken into account in their experiment, they can only be used to fix the horizontal translation errors. Ridene and Goulette (2009) and Gressin et al. (2012) suggested that point clouds need to be split into different blocks since the registration errors along the scanning trajectories normally increased gradually (or can be considered constant over a small interval for a block). Then, these blocks are treated as different individual static point clouds. The corresponding blocks from different runs are registered by performing a rigid body transformation with six transformation parameters. The six transformation parameters are three rotation angles (ω, ϕ and κ), and three translation parameters (T_x, T_y and T_z). Hence, if they are corresponding blocks **S** and **M** from two MLS point clouds, the mathematical model to align **S** to **M** is formulated as follows:

$$\mathbf{M} = \mathbf{RS} + \mathbf{T} \qquad \text{(eq. 6)}$$

where **R** is the orthogonal rotation matrix formed by three sequential rotation ($\mathbf{R}_1(\omega)$, $\mathbf{R}_2(\phi)$ and $\mathbf{R}_3(\kappa)$). $\mathbf{R}_1(\omega)$ is the rotation matrix around the X axis, $\mathbf{R}_2(\phi)$ is the rotation matrix around the Y axis and $\mathbf{R}_3(\kappa)$ is the rotation matrix around the Z axis. **T** comprises of T_x, T_y and T_z is the translation vector along the X axis, Y axis and Z axis respectively.

Similarly, point $p_s = [x_s, y_s, z_s]^T$ in the *slave* point cloud can be transformed to the coordinate system of the *master* dataset using the equation:

$$p_m = \mathbf{R}\, p_s + \mathbf{T} \qquad \text{(eq. 7)}$$

where p_m is the coordinate of the point p_s in the *master* coordinate system in terms of $[x_m, y_m, z_m]^T$.

While all registration methods essentially aim to solve these six parameters, different methods have been proposed from research to perform the registration of MLS point clouds. These registrations are done based on common points or features in the *slave* and *master* point cloud. They can be categorised into two groups: 1) target based and 2) target free registration. Target based registration requires a lot of manual work (e.g. set up the target or manually determine the target from the captured point clouds). The focus will be on the target free registration methods. Such target free registration methods can be classified into two groups: (1) point-based matching and (2) feature based matching.

4.1.1 Point-based matching

Point-based matching (e.g. the iterative closest points (ICP)) approaches have been the most popular methods for fine registration of laser scanned point clouds. Due to the characteristics of these approaches, they are only suitable to perform pairwise registration between two point clouds. Therefore, they cannot be applied to register a point cloud to a *master* model. The point-based matching can be categorised into two groups: (1) point-to-point matching and (2) point-to-plane (surface) matching. The main difference between them is that point-to-point aims to minimise the distances between corresponding points (Besl & McKay, 1992), while point-to-plane aims to minimise the distances between points to its corresponding tangent plane or surface in the other point clouds (Chen & Medioni, 1991).

4.1.1.1 Point-to-point methods

Theoretically, at least three corresponding point pairs are required in order to perform the registration of two point clouds. Horn (1987) proposed a closed-form solution using unit quaternion to calculate the 3D transformation parameters between two datasets. Practically however, it is a big challenge to determine reliable point pairs in the captured point clouds due to the point density, occlusions and noise. In order to deal with this problem, Besl and McKay (1992) introduced the Iterative Closest Point (ICP) approach. ICP is a pair-wise registration algorithm (i.e. scan-to-scan registration) and aims to find the transformation parameters to match the *slave* point cloud to the *master* point cloud by minimising the squared distances between corresponding point pairs in the overlapping area. The mathematical model of point-to-point matching is expressed as follow:

$$\underset{R,T}{\min} \sum_{i=1}^{C} \lfloor p_{sm} - \mathbf{R}p_s - \mathbf{T} \rfloor^2 \qquad \text{(eq. 8)}$$

where *c* is the number of correspondences and *p_{sm}* is the corresponding point (i.e. the closest point) of point *p_s* in the *master* point cloud.

ICP starts by determining the corresponding point pairs in the *model* and the *slave*. A corresponding point pair contains a point in the *slave* point cloud and its closest point in term of Euclidian distance in the *master* point cloud. The transformation parameters are estimated by minimising the distances of these correspondences. Then, the process is iterated by selecting a new set of corresponding point pairs and the process is repeated until the solution converges to a stable solution.

In order to converge to a solution which is the global minimum, ICP requires a good initial alignment between two scans (Kwang-Ho & Derek, 2008). In other words, the inputs for ICP are two point clouds that are already roughly align to each other. This condition is normally fulfilled in the case of MLS point clouds, as point clouds captured by MLS systems are normally roughly aligned with each other using GNSS and navigation sensors, for coarse georeferencing.

A number of variants of ICP have been proposed in order to improve the performance of ICP in different cases. These variants improved one or more steps of the original ICP algorithm and have been compared in Rusinkiewicz and Levoy (2001). In general, ICP has five steps: (1) point selection, (2) specifying correspondences, (3) weighting of the correspondences, (4) rejecting incorrect correspondences and (5) minimising the remaining correct correspondences.

Step 1 point selection: Points are selected in the *slave* point cloud, or both *master* and *slave* point clouds. Originally, all of the points in the point clouds are used for registration (Besl & McKay, 1992). Masuda et al. (1996) claimed that the points should be selected randomly from the point cloud to preserve the general distribution of the points and reduce the processing time. Godin et al. (1994) suggested that points can also be selected based on the intensity gradient if the information about the colour or intensity for the point are available. Another possibility approach is to select points based on the dimensionality (Gressin et al., 2013). In this approach, the authors concluded that the planar points (i.e. points belong to a planar surface) are likely to be the most reliable points for ICP. Linear points can also play an important part in the registration if their sampling is sufficient, whereas scatter points should either not be used as inputs for ICP, or only in a limited capacity.

Step 2 specifying correspondences: The simplest approach to find the corresponding point for a selected point is to determine the closest point of the query point in the *model* point cloud. However, the problem of an incorrect selection in correspondence may arise due to the presence of noise in the point clouds. A number of researches have been proposed to solve this problem (Pulli, 1999; Takai et al., 2013)). Godin et al. (1994) utilise the colour information to enhance the matching process. Correspondences can be determined based on the angle between normal vectors of the query point and its potential correspondences (Pulli, 1999). Takai et al. (2013) determine the correspondences based on the dimensionality of the points.

Step 3 weighting the correspondences: Correspondences are weighted based on either a defined criteria or given uniform weighting. One method is that corresponding point pairs that have larger distances will have lower weight such that:

$$W_i = 1 - \frac{D(p_{si}, p_{mi})}{Dmax} \qquad \text{(eq. 9)}$$

where W_i is the weight value for correspondence I; $D(p_{si},p_{smi})$ is the distance between the two points of correspondence i and D_{max} is the longest distance between the two points in all of the correspondences. Another approach is by weighing the correspondences based on the angle between the normal vectors:

$$W_i = n_{psi} \cdot n_{psmi} \qquad \text{(eq. 10)}$$

where n_{psi} and n_{psmi} are the two normal vectors of the two points p_{si} and p_{smi} respectively.

Step 4 rejecting incorrect correspondences: Certain corresponding point pairs are removed based on evaluating each pair individually, or by taking into account the whole correspondences (i.e. to find the correspondences that do not conform to the majority of the correspondence between point clouds). The worst corresponding point pairs are generally rejected based on different distance criteria (Masuda et al., 1996; Pulli, 1999; Turk & Levoy, 1994), which include the following:

- A corresponding point pair is rejected if their distance is larger than a pre-defined threshold.

- A pre-defined number of corresponding point pairs that have the largest distances are removed.
- Correspondences whose distances are larger than some multiple of the standard deviation are rejected.
- Correspondences are on the mesh boundaries

Step 5 minimising the distance of the remaining correspondences: With all of the corresponding pairs and the weight values determined, the final step of ICP is to calculate the transformation parameters **R** and **T** that minimises the sum of the squared distances between each corresponding point pair. Several closed form solution have been proposed such as approaches based on singular value decomposition (Arun et al., 1987), and closed form approaches using unit quaternions (Horn, 1987).

ICP has been used widely for the registration of range image data and TLS point clouds. It has also been applied to the registration of MLS point clouds (Gressin et al., 2013; Takai et al., 2013). However, with regards to MLS point clouds, the selection of corresponding point pairs is extremely difficult due to sparseness and heterogeneousness of the MLS point clouds and the way the points are captured. Gressin et al. (2013) pointed out that the point density of the MLS point clouds influences the accuracy of the registration. The higher the point density of the point clouds are, the higher the probability of having correct correspondences, and the closer it will be to the correct correspondence. As a result, the registration outputs can be more accurate. For sparse point clouds, as points of the same objects are captured arbitrary in different point clouds, the accuracy of the registration using point-to-point matching depends on the data and cannot be control by the algorithm itself.

Moreover, Grant et al. (2013) claimed that the quality of the registration output are influenced by the choice of which point cloud acts as the *master*. Indeed, as can be seen from Figure 4.1, the blue point cloud has higher point density than the red point cloud. If the red point cloud is chosen as the *master*, there will be the case that four blue points are considered to have the same correspondence (i.e. one-to-four correspondences) (Figure 4.1). Meanwhile, if the blue point cloud is chosen as the *master*, there will be only one-to-one correspondences.

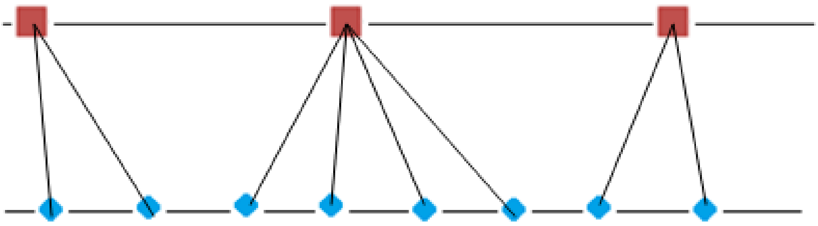


Figure 4.1 Example of specifying correspondences in point-to-point; blue points are points in the *slave* point cloud and red points are points in the *master* point cloud.

As discussed in Chapter 2, the point density of a building façade is significantly different based on the distance of the scanner to the building. If the same area is scanned twice in the opposing directions, as shown in Figure 4.2, the same object will have different point density at different runs. Specifically, in the case of the forward run (run 1) the point density of the object on the left hand side of the scanner will be much denser than the point density of the objects on the right hand side. In contrast, in the case of the backward run (run 2) the point density of the object on the left hand side of the scanner will be much denser than the point density of the objects on the right hand side. This leads to the issues of one-to-many correspondences which may cause errors for the point-to-point matching method.

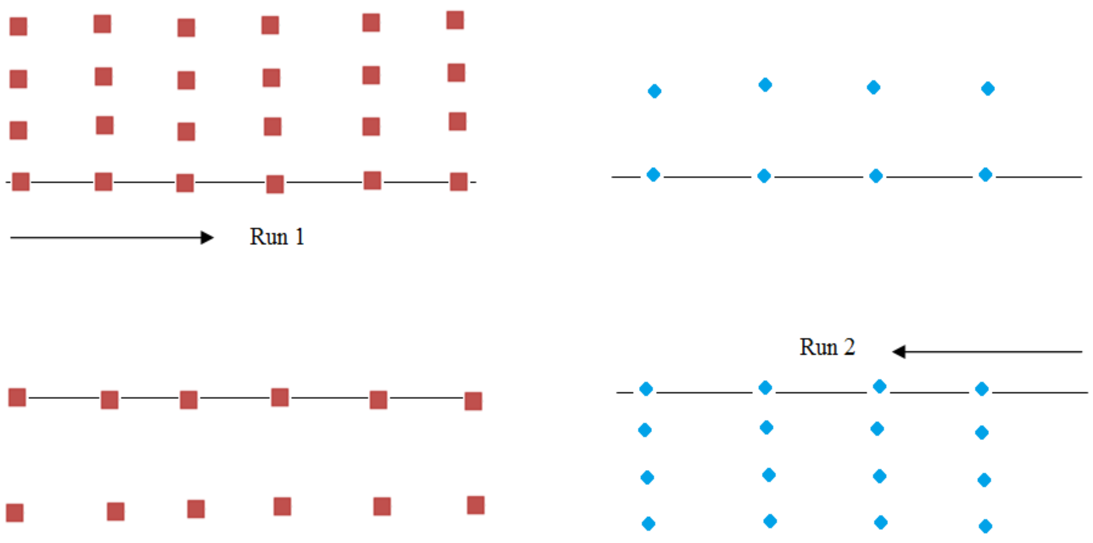


Figure 4.2 Point clouds of the same area captured at two different run (with opposite directions) using the same MLS system; red point cloud dataset is from run 1 and blue point cloud is from run 2.

4.1.1.2 Point-to-plane

The original point-to-plane method was proposed by Chen and Medioni (1991) to perform registration of the range image data. It aims at minimising the distance between points to their corresponding tangent planes in the *master* point cloud instead of minimising the distance between points to their corresponding points in the *master* point cloud. The mathematical model of the point-to-plane technique is expressed as follows:

$$d_{pp} = |(p_m - q_m)^T n_m| \qquad \text{(eq. 11)}$$

where q_m is the point in the *master* point cloud forming the tangent plane and n_m is the normal vector of the tangent plane.

Point-to-plane matching can overcome the issue of the missing correspondence from the point-to-point approach, as it does not require correct correspondences (Figure 4.3). However, it relies on a good definition of the surface, or good neighbourhood of points to define surface. There is a variety of methods for determining the plane.

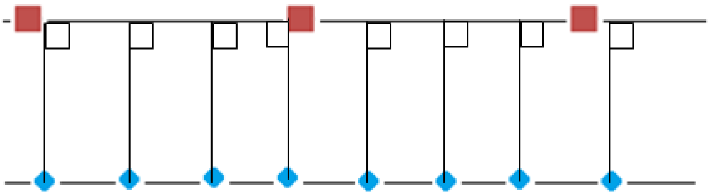


Figure 4.3 Example of point-to-plane matching; blue points are points in the *slave* point cloud and red points are points in the *master* point cloud.

Grant et al. (2012) proposed a registration approach for TLS point cloud data using the corresponding tangent plane of a point defined by the three closest corresponding points in the corresponding point cloud. Furthermore, in order to increase the number of redundancy and to take into account the uncertainty of both scans, the point-to-plane correspondences on both scans are established simultaneously (Figure 4.4).

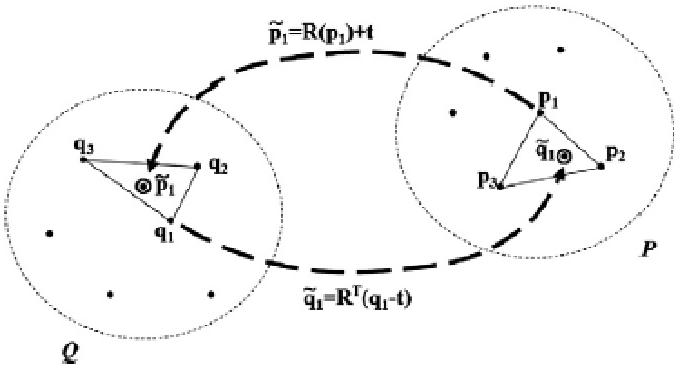


Figure 4.4 Point-to-plane; Adopted from Grant et al. (2012)

However, this approach is not applicable for MLS point clouds as the three nearest corresponding points in the corresponding point cloud normally belong to the same scanline. In other words, they are normally linear. Takai et al. (2013) suggested that the corresponding tangent plane can be obtained using PCA on the query point and its neighbouring points. Nevertheless, the issue of local saliency features estimation makes the point-to-plane method not suitable for sparse and heterogeneous MLS point clouds. The neighbourhood selection can suffer from the effects discussed in section 2.2.1.

4.1.1.3 Other point-based matching methods

Takai et al. (2013) proposed a hybrid point-based matching method – Classification and Combined ICP (CCICP) that combines point-to-point and point-to-plane approaches. This approach also utilise the point classification method using PCA (Demantké et al., 2012). First, points in both point clouds are classified into either (1) linear, (2) planar or (3) scatter points. Only planar points are used for the registration step. The six transformation parameters are obtained by performing a least-squares adjustment process that minimises both point-to-point and point-to-plane distances. Although this approach provides more redundancies than the point-to-point and the point-to-plane matching, it suffers from the limitations from both point-based matching techniques as previously discussed (e.g. the determination of the correspondences or the definition of the tangent planes). Consequently, it seems to be an inappropriate matching technique for MLS sparse and heterogeneous point cloud registration.

Aiger et al. (2008) proposed another point-based matching approach, namely the 4-points congruent sets. This method starts by finding four coplanar points in the *slave* point cloud. Then, the four corresponding points in the *master* point cloud are determined. Finally, those four point pairs are used to calculate the transformation parameter. Nevertheless, according to the authors, the output of this matching technique needs to be refined using ICP and it still relies on the good identification of the points, which can be difficult in sparse and heterogeneous MLS point clouds.

4.1.2 Feature based matching

Unlike point-based matching where matching is restricted between point clouds, feature based matching is able to register a *slave* point cloud to either a *master* point cloud or a *master* model. These pre-defined models can be obtained from other datasets that have higher accuracy than the captured MLS point clouds (e.g. point clouds captured using TLS, as-built documentation, or a 3D map). The feature-based registration is based on the fact that the same features exist in different point clouds/models with the same parameters. The registration is performed by matching the corresponding features in the *slave* point cloud to the *master* point cloud or to the *master* model in order to achieve this.

Numerous research using feature based matching have been proposed in order to perform point cloud registration. Based on their mathematically models, they can be categorised into three groups: (1) Semantic Virtual Feature Points Matching (Chan et al., 2016; Yang et al., 2016); (2) model-to-model (Khoshelham & Gorte, 2009; Rabbani et al., 2007); and (3) point-to-model (Chan & Lichti, 2012; Chan et al., 2013; Rabbani et al., 2007; Skaloud & Lichti, 2006).

4.1.2.1 Semantic Virtual Feature Point Matching

Similar to point-to-point matching techniques, Semantic Virtual Feature Points Matching (SVFPM) also aims to minimise the distances between corresponding points in different point clouds. Nevertheless, unlike point-based matching approaches, in SVFPM the corresponding points are considered as virtual feature points, instead of the actual points in the point clouds. Yang et al. (2016) proposed an automatic registration method based on semantic feature points for large-scale urban scene point clouds. It starts by determining the vertical features lines that belong to pole like objects and vertical planar features as shown in Figure 4.5. The intersections of these vertical feature lines with the ground are considered as the semantic virtual feature points for matching. According to the authors, after executing the feature points matching, ICP needs to be utilised to perform fine registrations between different point clouds. In other words, this approach aims to perform coarse registration between point clouds.

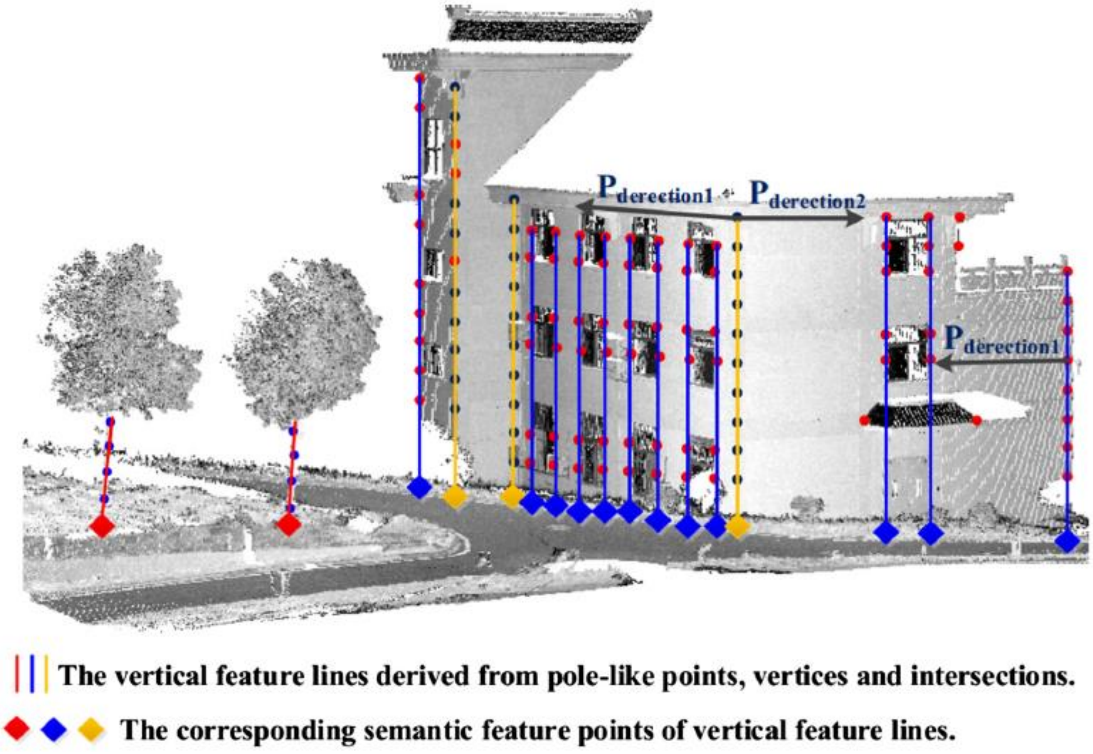
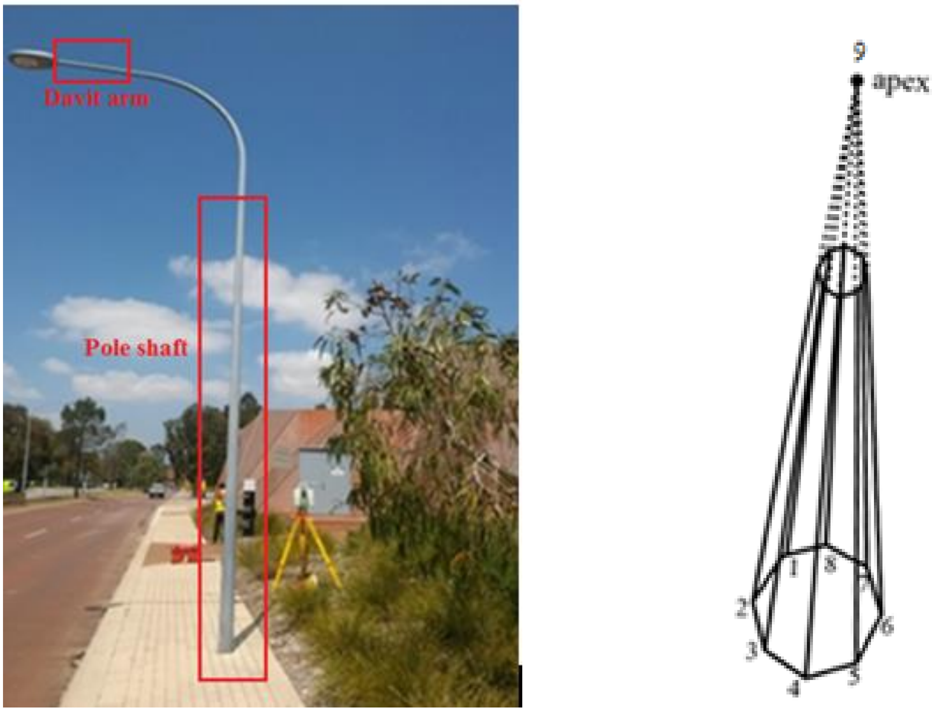

Pderection1
Pderection2
Pderection1
The vertical feature lines derived from pole-like points, vertices and intersections.
The corresponding semantic feature points of vertical feature lines.

Figure 4.5 Semantic feature points of vertical feature lines (adopted from Yang et al. (2016))

The virtual points can also be feature points such as from octagonal lamp poles as shown in Figure 4.6 (Chan et al., 2016). It starts by detecting the lamp poles from the point clouds. The model of the detected lamp pole is then calculated using the least-squares fitting algorithm. Based on the estimated model, nine virtual points are generated, as shown in Figure 4.6(b). Next, the set of eight registration parameters for all possible cases are calculated. The correct parameters are selected based on the alignment of the davit arm. Finally, the registration is refined using a horizontal cylinder fitting process. One of the biggest advantages of this method is that it only requires a single octagonal lamp pole to register two point clouds. Furthermore, there is no requirement on the structure, as long as they can be accurately modelled. However, the quality of this method heavily depends on the accuracy of the parameters of the octagonal lamp pole. Consequently, the poles have to be scanned with high point densities, which are not always applicable in sparse and heterogeneous MLS point clouds. In addition, such structures do not always exist in scanning scenes. The authors also mentioned that this method is more suitable for coarse registration than for fine registration.



(a) (b)

Figure 4.6 a) Octagonal lamp pole in reality and b) Model of an octagonal lamp pole (adopted from Chan et al. (2016))

4.1.2.2 Model-to-model least-squares matching

The goal of the model-to-model matching is to determine the six parameters of the rigid transformation that minimise the root mean square of the differences between plane parameters in the *slave* dataset and their corresponding plane parameters in the *master* dataset. Mathematically, this model can be expressed as follows:

$$\min_{\boldsymbol{R},\boldsymbol{T}} \sum_{i=1}^{c} [\mathbf{H}_{\boldsymbol{R},\boldsymbol{T}}(n_{s,i}, \rho_{s,i}) - \mathbf{H}_{\boldsymbol{R},\boldsymbol{T}}(n_{m,i}, \rho_{m,i})]^2 \qquad \text{(eq. 12)}$$

where c is the number of corresponding planes of different point clouds; **H** is a transformation operator defined by the rotation matrix **R** and the translation vector **T**; $n_{s,i}$ is the normal vector of the plane i in the *slave* point cloud; $\rho_{s,i}$ defines the distance to the origin of plane i in the *slave* point cloud; $n_{m,i}$ is the normal vector of the plane i in the *master* point cloud; and $\rho_{m,i}$ defines the distance to the origin of plane i in the *master* point cloud.

Khoshelham and Gorte (2009) proposed a plane matching method using a linear least-squares model that can be used to register point clouds onto existing 2D maps. According to this research, the transformation of a plane from the *slave* coordinate system to the *master* coordinate system can be expressed as follows:

$$P_m = \mathbf{H}^{\mathrm{T}} P_s \qquad \text{(eq. 13)}$$

where P_m is the vector consists of the four parameters (n_{mx}, n_{my}, n_{mz} and ρ_{m}) of a plane in the *master* dataset; P_s is the vector consists of the four parameters (n_{sx}, n_{sy}, n_{sz} and ρ_{s}) of the plane in the *slave* dataset; and **H** is the transformation matrix for the similarity transformation or the rigid transformation in case of MLS registration.

If there are at least three planes that exist in both, the *slave* and the *master* datasets, a closed-form least-squares solution is able to be used to calculate the transformation parameters to transform the *slave* dataset to the *master* dataset.

Another plane-to-plane matching approach that has been used to register different point clouds was proposed by Previtali et al. (2014). In this research, each pair of corresponding planes provide four observation equations including three equations for the discrepancies of the normal vector and one equation for the difference of the distance from the origin to the least-squares model. These can be expressed as:

$$\delta = \begin{pmatrix} n_{m,i,x} - \boldsymbol{R}n_{s,i,x} \\ n_{m,i,y} - \boldsymbol{R}n_{s,i,y} \\ n_{m,i,z} - \boldsymbol{R}n_{s,i,z} \\ \rho_{m,i} - \rho_{s,i} + \left(\boldsymbol{R}n_{s,i,z}\right) + \boldsymbol{T} \end{pmatrix} \qquad \text{(eq. 14)}$$

where δ is the vector that consists of the discrepancies between the parameters of the corresponding planes of the *slave* and the *master* point clouds; **R** is the rotation matrix; **T** is the translation vector; $n_{s,i,x}$, $n_{s,i,y}$ and $n_{s,i,z}$ are the three components of the normal vector of the plane *i* in the *slave* point cloud; $\rho_{s,i}$ defines the distance to the origin of plane *i* in the *slave* point clouds; $n_{m,i,x}$, $n_{m,i,y}$ and $n_{m,i,z}$ are the three components of the normal vector of the plane *i* in the *master* point cloud; and $\rho_{m,i}$ defines the distance to the origin of plane *i* in the *master* point clouds.

In this model, each pair of corresponding planes only contributes the estimation of **T** in one direction. For instance, T_y (i.e. translation value along the Y axis) can only be estimated using corresponding planes which are parallel to the Y axis. Thus, in order to ensure the success of the method, the planes need to be evenly distributed throughout the scanning scene. In theory, the sparseness and heterogeneity of the captured MLS point clouds will not influence the registration output as the same planar object in different captured point clouds can be extracted regardless of the point density. Nevertheless, the number of redundancies for the least-squares model of model-to-model matching is limited. A correspondence between planes only provides four observation equations. Furthermore, the estimated plane parameters of the same plane in different aligned MLS point clouds (i.e. MLS point clouds that are perfectly registered), are normally not identical. This is due to the accuracy of the laser scanner, the number of points representing the plane, the accuracy of the plane detection and segmentation (e.g. RANSAC), and the method that is used to calculate the plane parameters (e.g. PCA). Especially in the case of sparse and heterogeneous MLS point cloud registration, the quality of the registration using the model-to-model matching approach cannot be assured to be sufficient since the sparseness affects the parameters and extracted values of these models.

4.1.2.3 Point-to-plane least-squares fitting

Point fitting onto a matching plane is also called Least-squares Plane Fitting Adjustment (LSPFA). The goal of this method is to find the six transformation parameters to minimise the sum of the squared distances of points to their corresponding planar surfaces. The first step is to determine point p_s belonging to the plane PL_s in the *slave* point cloud. Plane PL_m, which is the correspondence of plane PL_s in the *master* dataset, can be described by its normal vector n_m (n_{mx}, n_{my}, n_{mz}) and the perpendicular distance to the origin ρ_m. If the registration is completed successfully, point p_m, which is the transformed point of point p_s in the *master* point cloud (eq.7), will satisfy the conditions in the following equation:

$$n_{mx}x_m + n_{my}y_m + n_{mz}z_m + \rho_m = 0 \qquad \text{(eq. 15)}$$

By combining the transformation equation (eq. 7) with the distance to plane equation (eq. 15), the mathematical model F for the least-squares adjustment process can be derived as:

$$F = n_m^{\mathrm{T}} (\mathbf{R} p_s^{\mathrm{T}} + \mathbf{T}) + \rho_m = 0 \qquad \text{(eq. 16)}$$

The mathematical model of this approach shares the same concept with the point-to-plane approach. The main difference between this approach and the point-to-plane matching approach is that the corresponding planes are formed by points representing the actual planes in the point clouds in this approach, or from the planes in the pre-defined models. In contrast, with the point-to-plane matching approach the corresponding tangent plane are locally formed by the neighbouring points of the corresponding point in the *master* point cloud. Equation 16 can be rearranged into the form of the normal equations:

$$\mathbf{AX} + \mathbf{W} = 0 \qquad \text{(eq. 17)}$$

where **X** is the vector of unknowns to be solved. In this case $\mathbf{X} = [\omega\ \phi\ \kappa\ T_x\ T_y\ T_z]^{\mathrm{T}}$ which includes the parameters for the rotation matrix **R** ($\mathbf{R}(\omega)$, $\mathbf{R}(\phi)$, $\mathbf{R}(\kappa)$) and a translation vector **T** (T_x, T_y and T_z) ; **A** is the Jacobian design matrix and **W** is the misclosure vector;

The inputs for the LSPFA registration method can be categorised into two groups: consisting of the point correspondences and the plane parameters. Inputs from the *master* dataset are the mathematical models of the identified planes (i.e. the four plane parameters). These mathematical models can be obtained by applying PCA on the groups of points that have been identified as representing the different planar objects in the captured point clouds (through segmentation for example). Alternatively, they can be the existing models from other data sources, such as from topographic databases (Khoshelham & Gorte, 2009). On the other hand, inputs from the *slave* dataset are different groups of points belonging to the corresponding planes in the *master* model. Each point that belongs to a plane in the *slave* point cloud provides one observation equation for the least-squares adjustment model. As a result, the number of redundancies with this approach is significantly larger than the model-to-model approach. For each pair of corresponding planes the number of observation equations is equal to the number of points representing the planes in the *slave* point cloud. In other words, the number of observation equations provided from a pair of corresponding planes can vary from several to thousands of equations depending on the number of points. These groups of points can be manually or automatically extracted from the captured MLS point clouds. As a result, this technique seems to be the most suitable approach for MLS sparse point cloud registration. The output of this process relies on a good planar model being extracted, as well as good correspondences between the points and the planes. The following discussion will focus on this method only.

4.1.2.4 Requirement for point-to-plane least-squares fitting

The requirements for the success of the least-squares point fitting onto planes adjustment process were discussed in Skaloud and Lichti (2006), Xiao et al. (2012) and Previtali et al. (2014). Xiao et al. (2012) claimed that the transformation parameters can be calculated from three nonparallel planar surfaces. Nevertheless, this requirement does not always assure the success of the transformation calculation. As shown in Figure 4.7, **G1**, **G2** and **G3** are three groups of points representing three different nonparallel surfaces **P1**, **P2** and **P3**. As can be seen from Figure 4.7(b) and (c), there is more than one solution that can preserve the plane condition of the points and the relative relationships between points. According to Skaloud and Lichti (2006), planes need to vary in slopes and orientations to obtain accurate results. In addition, Previtali et al. (2014) stated that planes need to be evenly distributed in the scanning scene. These two statements are too general as they do not specify a specific requirement for the success of point-to-plane least-squares fitting process.

.

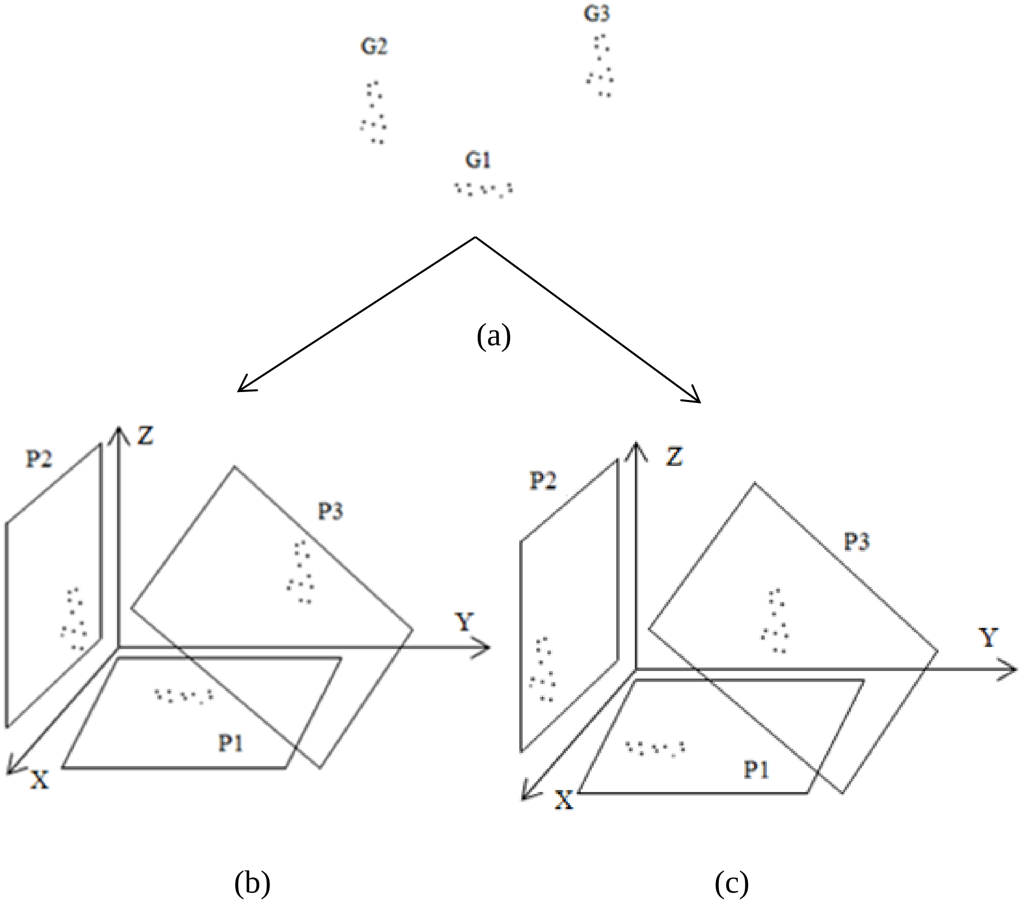


Figure 4.7 Example of translation error in plane-based matching when the minimum requirement is not met: a) three point clouds; b) registration solution 1 and c) registration solution 2

From a geometric point of view, theoretically, the three rotation parameters can be computed from two non-parallel planes. With respect to the calculation of the translation parameters, a pair of corresponding points is required. However, it is almost infeasible to find a pair of correct corresponding points in two different MLS point clouds, especially in sparse and heterogeneous MLS point clouds. Therefore, one of the requirements for the success of the least-squares adjustment is that there are at least three planes that intersect with each other to form a unique point solution. The more triplets present in the least-squares model, the stronger the geometry is. As a result, the least-squares model becomes stronger. However, most of the planar features in the captured MLS point clouds in urban areas are orthogonal with the trajectory of the moving vehicle. Especially, in the case that a type 1 scanner is used, it may occur that there are no triplets present in a captured point cloud (Figure 4.8). This may lead to there is no accurate registration solution for these point clouds.

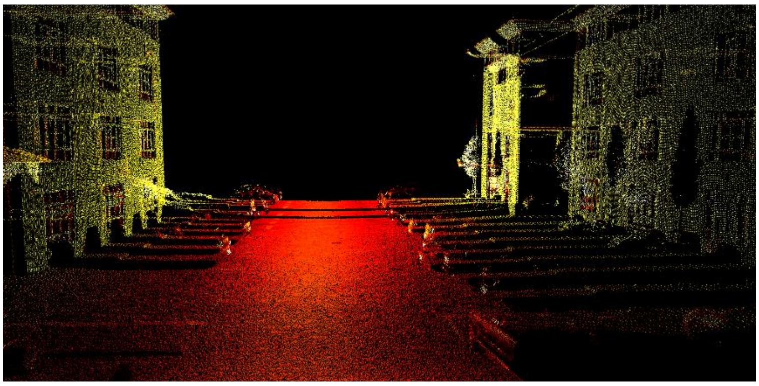


Figure 4.8 Point cloud dataset captured using type 1 scanner; Adopted from Renishaw (2017)

4.1.3 Discussion of the effects of errors in the rotation and the translation parameters

The magnitude of the rotation errors in relation to the quality of the MLS point cloud registration outputs can vary depending on the distances of the *slave* point cloud to the origin. Meanwhile, the magnitude of the translation error in relation to the quality of the MLS registration outputs is the same, regardless of the distance of the *slave* point cloud to the origin. As can be seen from

Figure 4.9, if points P1 and P3 are translated using the same translation vector, the distance from P1 to P2 will be identical to the distance from P3 to P4. Meanwhile, if points P1 and P3 are rotated by α degrees, the distance from P1 to P2 is significantly smaller than the distance from P3 to P4. The larger the distance of the point to the origin is, the larger the effects of the rotation parameter are on the final solution.

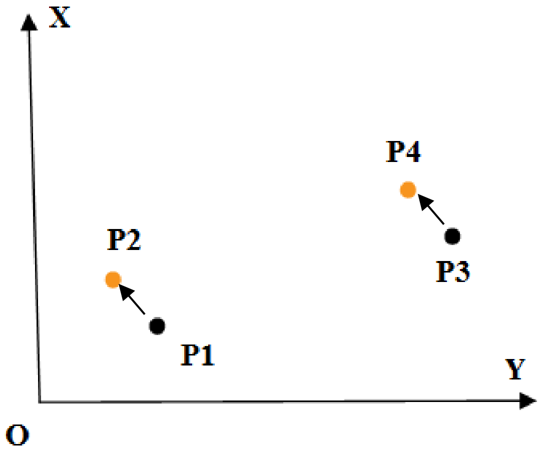


Figure 4.9 Effect of translation parameter at different distances

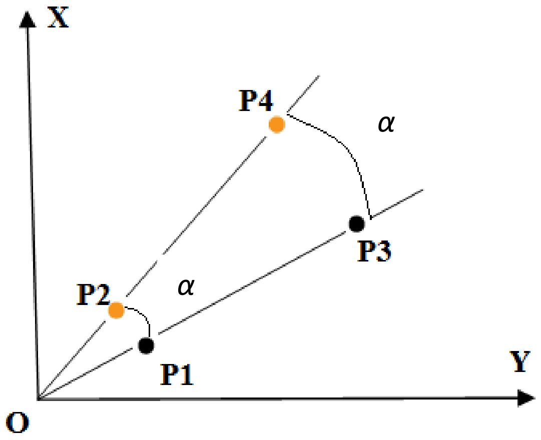


Figure 4.10 Effects of rotation parameter at different distances

As the inputs for the LSPFA cannot be assured to be errorless due to the accuracy of the scanner systems, noise, and the smoothness of the extracted planes, as well as the fact that LSPFA is performed iteratively, both the estimated translation parameters and rotation parameters may not be estimated perfectly. Consequently, if the point cloud datasets are located away from the origin, the least-squares adjustment model may not converge due to the effects of the rotation errors. In the case of MLS systems, the captured point clouds are normally located far away from the origin as they are referenced to a global coordinate system. Furthermore, a MLS project may cover a large area (e.g. several square kilometres). This leads to the capture point clouds being far away from the origin. In addition, it cannot be assured that the estimated plane parameters calculated based on the group of point representing a plane in the *master* point cloud is errorless. The estimated value for the distance to the origin parameter ρ of this plane is significantly affected by the location of the group of point representing this plane. The values of magnitude are larger than those for the normal vector. This again may contribute to the least-squares adjustment model not converging. Therefore, before performing registration, the *slave* point clouds and the *master* point cloud (or the *master* models) need to be translated so that the origins are close to the centre of mass of the points. This way the impacts of errors in the rotation is minimised.

4.1.4 Conclusion of point cloud matching

This analysis was performed in order to determine how sensitive the state-of-the-art point cloud matching methods are in regard to the sparseness and heterogeneousness of MLS point clouds. The scanline patterns and the sparseness of MLS point clouds may have significant impacts on the quality of the registration outputs using point-based matching techniques. In theory, the feature-based matching will not be influenced by the sparseness of the MLS point clouds as the same feature in different captured point clouds can be extracted regardless of the point density. Nevertheless, due to the nature of different feature based matching techniques, the LSPFA matching techniques seems to be the most suitable approach to perform MLS sparse point cloud registration.

4.2 Discussion of evaluation measure

Different point cloud matching techniques use different minimisation criteria (e.g. point-to-point distances or point-to-plane distances). These minimisation criteria have also been used as the error metrics for evaluation of the quality of the registration. Nevertheless, the error metric to test this must be selected carefully to ensure the reliable interpretation of the results and to not be dependent on the method or minimisation criteria. Beside the evaluation of the registration quality based on the minimisation criteria, there are also different error metrics that have been applied in order to evaluate the quality of a registration process. They can be divided into three groups: 1) visual inspection methods; 2) target-based methods; and 3) target-free methods.

4.2.1 Visualisation inspection approach

With visual inspection, the quality of the registration results is estimated based on the visualisations of the final registration outcomes (Xiao et al., 2012). Practically, this approach is usually used as a complement method for other approaches, and only provides a qualitative evaluation instead of objective measurements.

4.2.2 Target-based evaluation approaches

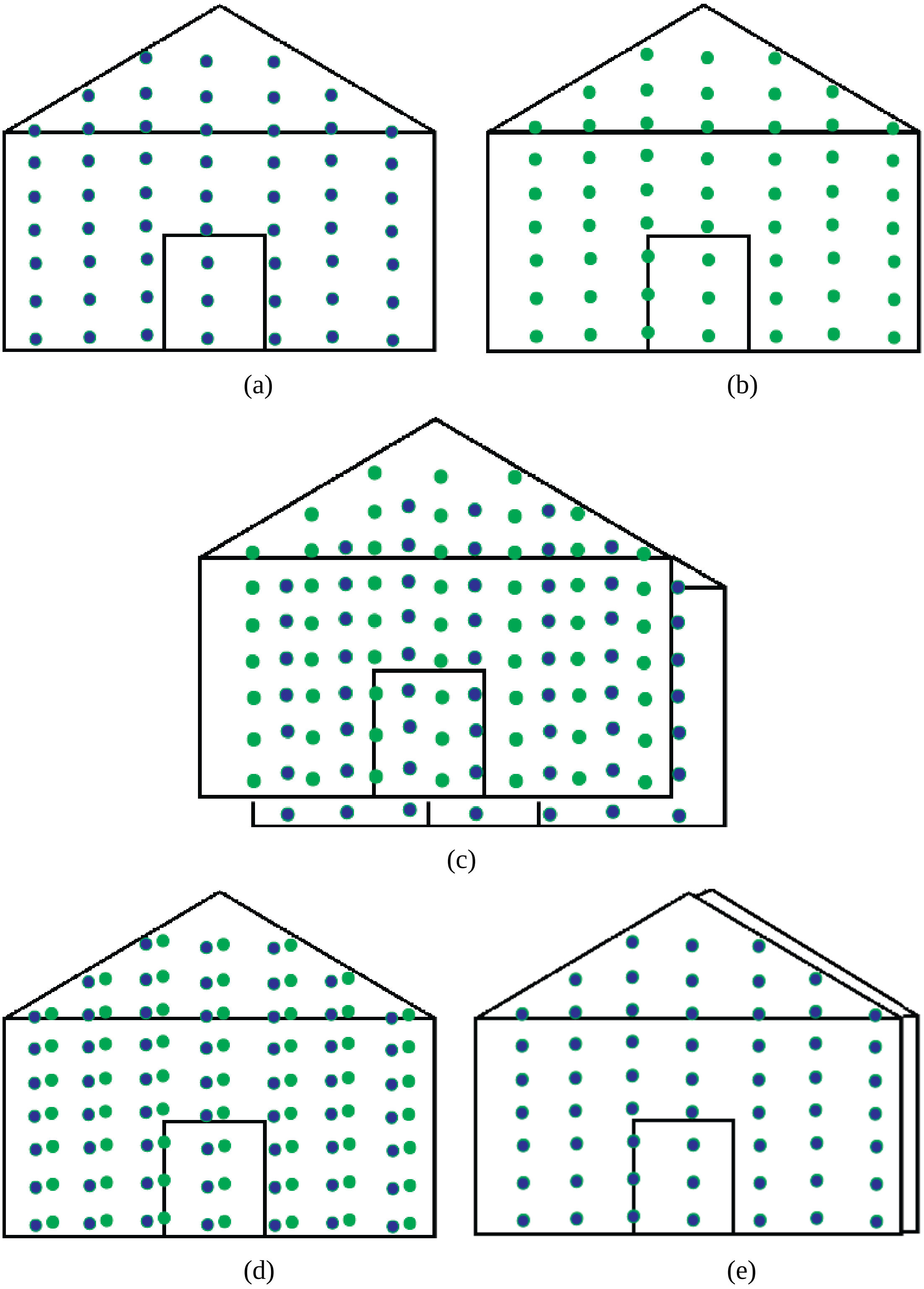
In target based approaches, the quality of the registration can be checked based on the distances between corresponding target points before and after matching (Yang et al., 2016). The other error metric for target based registration approaches are the calculation of the changes in the estimated rotation angles and translation vectors with respect to the benchmark values (Grant et al., 2013; Previtali et al., 2014; Yang et al., 2016). A benefit of this method is that if the targets have not been used in the registration, they provide a good independent validation and quantification of the errors from the registration process. However, they can suffer several limitations.

The biggest limitation of the target-based approach is that they often require a lot of manual work to isolate and extract targets unless they are well defined. Especially for sparse and heterogeneous point clouds, this is not always feasible to perform efficiently. Regarding the examination of errors based on the discrepancies between the estimated transformation parameter values with the benchmarking values, it can be very difficult to interpret the final results in many cases. For example, in the case where there is a need to compare the results of three different registration methods, estimated transformation parameters resulted from a method may have an error of 5 degrees in the roll angle while the estimated parameters from another method may have errors of 4 degrees in roll and 2 cm in the translation parameter along the Y axis and the estimated parameters from a third method may have an error of 4 cm in the translation parameter along the X axis. In this case, while the errors between targets can be calculated, it is difficult to say which method is better as different target configurations in the scene may lead to different errors. Furthermore, the impacts of rotation errors can be significantly different depending on the distances of the point cloud datasets to the origin of the coordinate systems. Consequently, further steps need to be implemented to achieve a better performance in comparisons.

4.2.3 Target-free evaluation approaches

The target-free evaluation approaches can be classified into two categories: 1) the evaluation based on the combined RMS values of the residuals of the distances between corresponding points and 2) the evaluation based on combined RMS of the residuals of the distances between points and their corresponding surfaces. The first approach is mostly used in the evaluations of point-to-point approaches (Gressin et al., 2013; Takai et al., 2013). Meanwhile, the second approach is mainly used in evaluating the registration from feature based approaches (Bae, 2006; Chan et al., 2016; Rabbani et al., 2007; Skaloud & Lichti, 2006).

Due to the sparseness and heterogeneousness of the MLS point clouds, it may not be feasible to have a correct correspondence of points in the corresponding MLS point clouds. This can lead to the problems shown in Figure 4.11. In this figure two point clouds of the same building are captured with different runs (Figure 4.11(a) and Figure 4.11(b)) and have similar point densities. They are roughly aligned to each other, with only mis-alignments in the translation parameters as shown in Figure 4.11(c). Figure 4.11(e) shows the positions of the two point clouds after the registration. As can be seen, a very low value of RMS for the distances between point pairs is achieved in this example despite the fact that the solution is incorrect compared to the correct solution highlighted in Figure 4.11(d). This is due to the way the points are sampled from the surface in sparse and heterogeneous point clouds. In practice, most of the captured points in a MLS point cloud data in urban areas belong to building façades parallel to the MLS trajectories, and the ground (Figure 4.12). Gressin et al. (2013) and Takai et al. (2013) achieved low RMS (e.g. 40 mm) values when applying point-based matching to register MLS point clouds. Nevertheless, as can be seen from their results in Figure 4.13 and Figure 4.14, the point-to-point distances of the correspondences on the façades that are not parallel with the MLS trajectories are much higher than the distances between corresponding points on the facades paralleled to the MLS trajectories. This may indicate an error in the translation and rotation along the MLS trajectories. Nevertheless, it was not addressed in the aforementioned papers.



(a)

(b)

(c)

(d)

(e)

Figure 4.11 Evaluation of the errors when using RMS values of the distances between corresponding points: (a) and (b) show the point clouds of the same building at two different runs; (c) mis-alignment between two points cloud (a) and (b); (d) correct matching solution and (e) incorrect matching solution

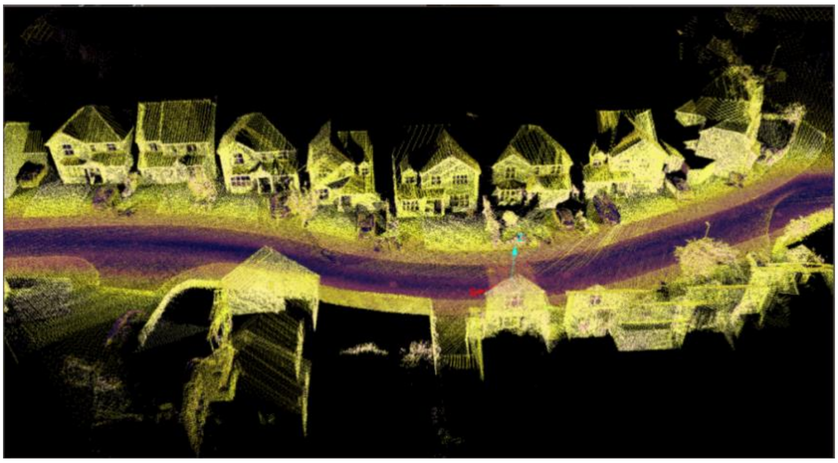


Figure 4.12 Point cloud dataset captured using type 2 scanner; Adopted from Renishaw (2017)

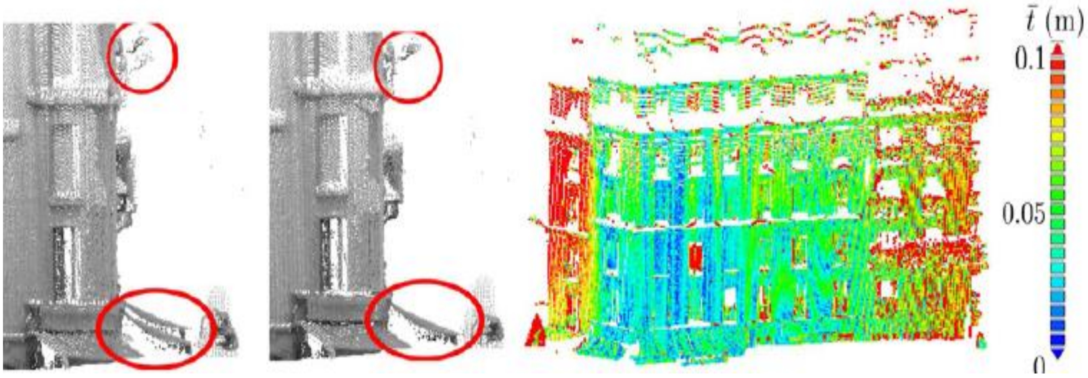


Figure 4.13 The point-to-point distances of the corresponding points after the registration process; Adopted from Gressin et al. (2013)

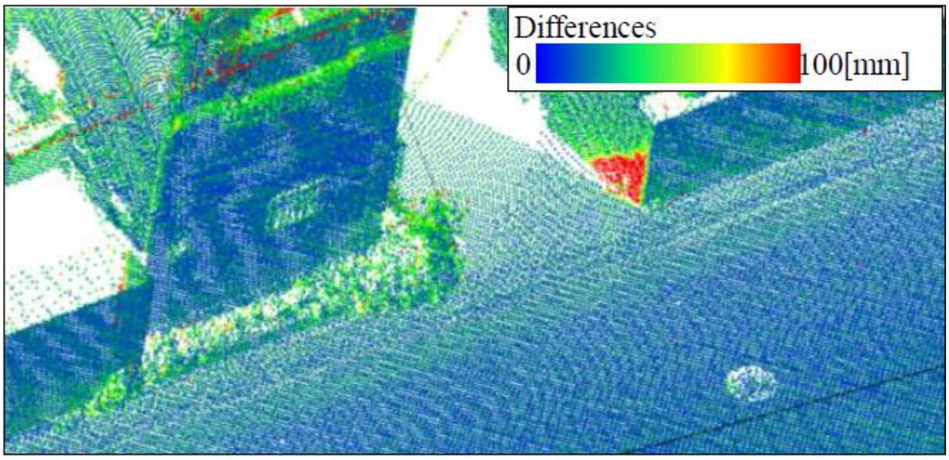
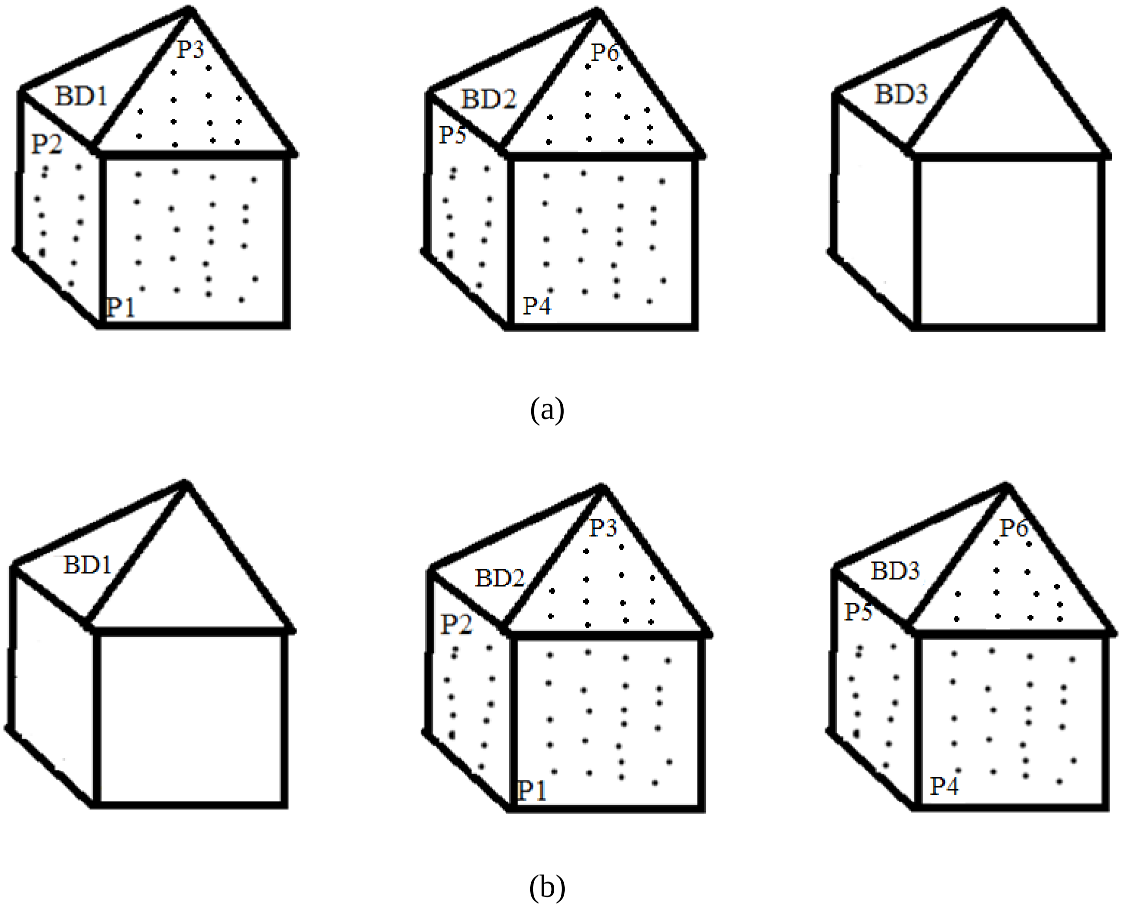


Figure 4.14 The point-to-point distances of the corresponding points after the registration process; Adopted from Takai et al. (2013)

Skaloud and Lichti (2006) and Chan et al. (2016) used the RMS values of the residuals of points fitting the planes to evaluate the quality of the least-squares matching model for the calibration and registration process. This approach seems to be more suitable for the evaluation purpose as, in theory the sparseness and the heterogeneousness of the captured MLS point clouds does not affect those RMS values. In their research, only four planes independent of the registration were used as the check planes. Evaluating the quality of the registration based just on the RMS values of four planes may be limited when the orientations (i.e. normal vectors) of the check planes are not considered. Indeed, theoretically, the residual value of a point fitted to its plane does not change if this point is translated parallel to the plane surface. For instance, as shown in Figure 4.15(a) point clouds 1, 2 and 3 (P1, P2 and P3) belong to building 1 (BD1) and point clouds 3, 4 and 5 belong to building 2 (BD2). Building 3 was not scanned and therefore does not have point clouds belonging to it. However, due to the errors in translation after the registration process, P1, P2 and P3 are incorrectly transformed onto BD2, and P3, P4 and P5 are incorrectly transformed onto BD3 (Figure 4.15(a)). Nevertheless, the RMS of P1 and P4 are still equivalent to the correct solution as the front façades of all the building belonging to the same plane. Similarly, the RMS values of P3 and P6 have not changed as the roofs of all the building belonging to another plane. Meanwhile, the RMS of P2 and P5 are significantly increased as the sides of each building belong to different planes. Consequently, if only P1/P3 and P4/P6 with their similar orientations are used as check planes, the evaluation will be inaccurate. Even given that this example is exaggerated in an urban area with specific setback rules, it is possible for this issue to occur in practice. Similar problems will occurr for the case where point clouds are erroneously rotated around the axis of the normal vector of the plane.



(a)

(b)

Figure 4.15 The RMS values and errors in translations: a) correct registration and b) incorrect registration with translation error in the trajectory direction of the scanner.

4.3 Proposed error metric for the evaluation of MLS registration

Since the mis-alignments between different captured point clouds of the same area may lead to errors in the modelling or interpretation of the data, the evaluation of the outputs of the registration needs to be carefully considered. Target based approaches are not considered in the following discussion as they have been found unsuitable (section 4.2.2). Among the existing target free evaluation approaches (i.e. the RMS of point-to-point distances and the RMS of point-to-plane distances) the error metric that utilises the RMS of the residuals of points fitted onto their planes (Chan et al., 2016; Skaloud & Lichti, 2006) seems to be the most suitable one for sparse and heterogeneous MLS point clouds. The proposed new target free error metric can overcome the limitations of these recent target free approaches, for instance, evaluating the quality of the registration just based on the RMS values of four planes may be limited if the orientations are not taken into account (section 4.2.3). Hence, planes with different orientations need to be considered as check planes to evaluate the quality of the registration process. Then, the average RMS value can be considered as an index to evaluate and compare the quality of the results of different registration approaches. However, when looking at the example presented in Figure 4.15, if the number of check planes that have similar orientations with P1 or P3 (e.g. P4 and P6) outnumber the number of planes that have dissimilar orientations (e.g. P2 and P5), the final average RMS value may be changed insignificantly even given that a large error is present. This will again lead to an inaccurate estimation of the quality of this registration process result. Practically, in MLS point clouds of urban areas, most of the planes have similar orientations which normals are orthogonal with the direction of the scanning vehicle, and hence these may perturb the results in their favour. In addition, if there are errors in estimated transformation parameters, planes which have similar orientations will suffer similar effects. In theory, the quality of the MLS registration outputs using different matching techniques or using the same matching technique but different inputs can be considered to have the same quality if their magnitudes of errors are similar. For instance, if a MLS registration output has an error of 2 cm along the Y axis, and another registration output has an error of 2 cm along the X axis, these two registrations should be considered to have the same quality. In contrast, they will have different quality if the magnitudes of the error are not similar.

Hence, a novel error metric, which takes into account the RMS values of all of the planes in the scanning area instead of using RMS values of a fix number of check planes, and which also considers their orientations, is introduced. The workflow of the proposed error metric is illustrated in Figure 4.16.

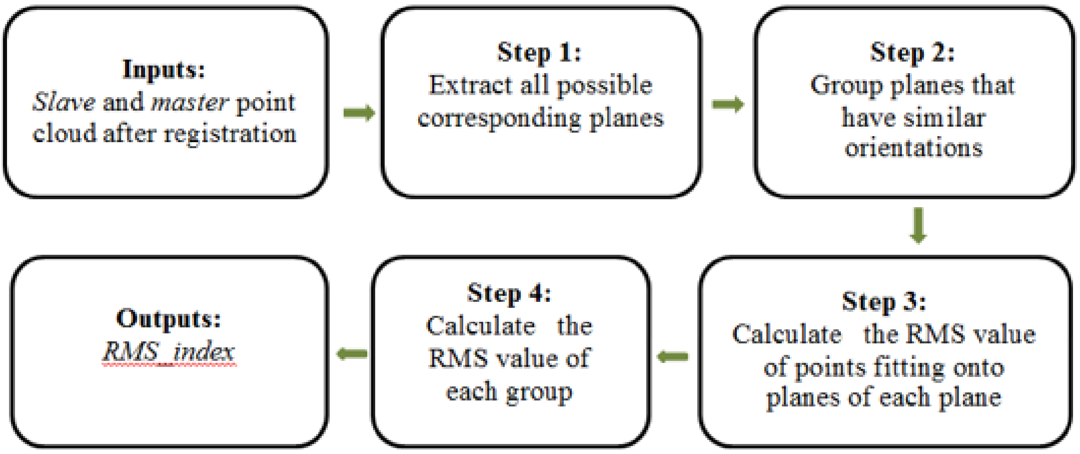

Inputs:
Slave and *master* point cloud after registration
Step 1:
Extract all possible corresponding planes
Step 2:
Group planes that have similar orientations
Step 3:
Calculate the RMS value of points fitting onto planes of each plane
Step 4:
Calculate the RMS value of each group
Outputs:
RMS_index

Figure 4.16 The workflow of the proposed error metric

The evaluation process starts with the accurate extraction of check plane in the *master* dataset. This procedure can be done manually or automatically using an accurate plane detection and segmentation algorithm. The orientation of each plane is then calculated using PCA (as discussed in section 2.2.1). After that, the RMS values for each plane after the registration is calculated and categorised into different groups based on their orientations (i.e. normal vectors). Different check planes are grouped based on the angular differences between them, calculated using eq.5 (in section 2.3.3). In this research, two planes are considered to have similar orientations if the angular difference between them is smaller than 10 degrees. This was suggested and supported by numerous authors for plane detection approaches (Cabo et al., 2015; Nurunnabi et al., 2012; Previtali et al., 2014). By doing this, the new error metric measures the quality of the registration from all possible orientations and each orientation has the same contribution to the final index. As a result, the new proposed method not only evaluates the quality of the registration process but can also be used to compare the quality of the registration process from different techniques.

Finally, the average value of all RMS values of each group is considered as an index to evaluate the registration results, as well as to compare the registration results of the same dataset but using different registration techniques. As a result, the problems with the translation and the rotation parameters along the direction vectors of planes (mentioned in section 2.2.2) can be compensated. The final error metric *RMS_index* can be computed as follows:

$$\text{RMS} = \sum_{i=1}^{n} \frac{\text{RMS}_{\text{gi}}}{n} \tag{eq.18}$$

where RMS_{gi} is the RMS value of each group; n is the number of groups.

4.4 Summary

In this chapter, different point cloud matching techniques were reviewed. The limitations of each technique in performing MLS sparse point cloud registration are analysed and discussed. The analysis shows that the LSPFA seems to be the most suitable approach. However, a set of minimum requirements must be achieved to assure the success of the LSPFA model. Those requirements are: 1) there is at least 1 triplet of planes exists in the scanning area and 2) both the *slave* point cloud and the *master* point cloud (or the *master* model) need to be translated so that the origins are close to the central mass of the points.

In order to evaluate the registration results a novel error metric was proposed. It utilises the RMS values of plane fitting as well as considering the orientations of the check planes. The proposed error metric can overcome the limitations of other error metrics by assessing the quality of the registration regarding all the possible orientations with equal weight.

CHAPTER 5 EVALUATION OF SPARSE AND HETEROGENEOUS MLS POINT CLOUDS PLANE DETECTION AND SEGMENTATION

In this chapter, the proposed plane detection and segmentation method, as presented in chapter 3, are evaluated and compared to state-of-the-art methods as introduced in chapter 2.

5.1 Discussion of the existing plane detection and segmentation methods consider in the comparison process

According to Tarsha-Kurdi et al. (2007), the outputs of the RANSAC is sensitive to the input parameters (e.g. the distance threshold parameter). This threshold can only be empirically determined (Vosselman & Maas, 2010), which means that it requires additional work to determine the ideal parameters for RANSAC. Moreover, according to Fujiwara et al. (2013), different point clouds with different point densities require different distance threshold values in order to obtain accurate outputs. The variant of RANSAC proposed by Previtali et al. (2014) utilises the normal vectors of points as well as the spatial proximity between points in detecting planar features. By taking into account the normal vectors of points and the spatial proximity between points, the influences of the distance threshold parameter may be reduced. In order to extract ‘meaningful’ features (e.g. walls, roofs, etc.) from the point clouds the spatial proximity between points cannot be ignored (Previtali et al., 2014). As a result, this variation is considered as the representative of RANSAC in this research and it is called RANSAC utilising normal vector, or RANSAC-NV. Nevertheless, as discussed in chapter 2, the sparseness of the captured point clouds leads to special challenges for normal vectors estimation process. Hence, the RANSAC algorithm without considering normal vectors of points is also compared and included in this comparison.

Among different techniques using the region growing concept, the robust segmentation method based on RDPCA (Nurunnabi et al., 2015) has been proven to have a very high accuracy. Thus, it is considered as another benchmark method for comparison. Furthermore, the plane detection based on the line arrangement (Cabo et al., 2015) is used for the comparison as it is considered to be the representative of scanline based plane segmentation approach. This approach is termed as Cabo, based on the name of the first author of the paper introducing this method and included in this comparison.

For the evaluation three different datasets were used. These datasets compose of a simulated errorless point cloud dataset and two real point cloud datasets captured using the MDL Dynascan S250 system including a pyramid style feature with three intersection planes, and a point cloud of a road corridor. Different comparison criteria are utilised to evaluate and compare the plane detection and segmentation outputs from all of the state-of-the-art approaches and the novel introduced approach.

5.2 Evaluation using the simulated dataset

A simulated dataset is created based on an errorless point cloud of a simple building comprising of a roof (plane 1), a building façade (plane 2), a window (plane 3) and a door (plane 4), as shown in Figure 5.1. This dataset has the characteristics of a point cloud dataset with heterogeneous and sparse point density with various point spacing distances (i.e. point spacing values vary from 1.0 cm to 1.4 cm) and profile spacing distances (i.e. profile spacing values vary from 0.8 cm to 9.5 cm). In this dataset, three planar features have similar normal vectors (i.e. the window, the door and the house façade surface). The orthogonal distances between the door surface to the building façade and between the window surface and the building façade surface are 6 cm and 5 cm respectively. In reality, these distances can be bigger (e.g. 20 cm), nevertheless, these small distance values are used to test the robustness of the discussed methods in the severe conditions where is may be difficult to separate close parallel planes. The reason behind this simulated dataset is that as the simulated dataset is errorless, the point density is known, and the ideal parameter values for all of the discussed plane detection and segmentation approaches can be determined. The dataset can be used to evaluate the performances of all of the discussed approaches for detecting and segmenting parallel adjacent planar surfaces as the ideal solutions are known. As this dataset is simulated, the scanlines can be easily split, and the scanline detection element of the proposed approach can be easily evaluated.

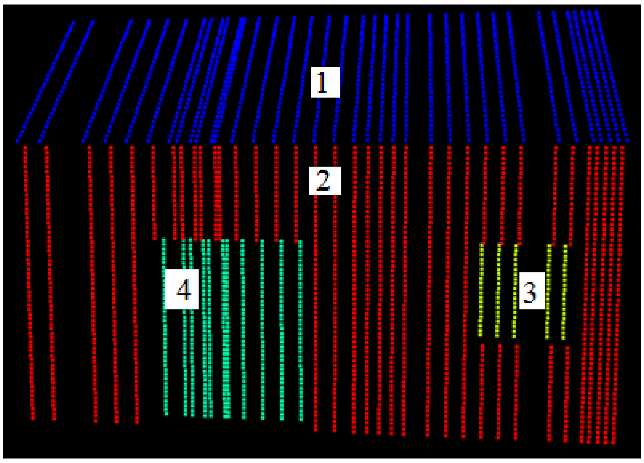


Figure 5.1 The simulated dataset consisting of different planes indicated by different colours; Adopted from Nguyen, Belton, and Helmholz (Submited and under revision-a)

5.2.1 Evaluation of the PSPS method

5.2.1.1 Forming scan profiles

The three steps of the local neighbouring selection, the direction vectors estimation, and the forming of the scan profiles, are grouped into a single procedure which breaks the scanlines into different scan profiles. There are five parameters required for this process, namely the number of neighbourhood points *nnp*, the angle threshold A_i, the number of iterations *T*, the distance parameter for the mRANSAC line fitting approach *rd* and the distance parameter for the forming scan profile process DL_i. All of them are set to equal to the suggested standard setting parameter values (section 3.4) and are summarised in Table 5.1.

Parameter	**Value**
nnp	7 points
T	10 iteration
A_i	0 degree
rd	0 mm
DL_i	0 mm

Table 5.1 setting parameters for the simulated dataset

After this step is performed, all of the scan profiles are perfectly segmented from their scanlines as shown in Figure 5.2, with the process being 100% correct.

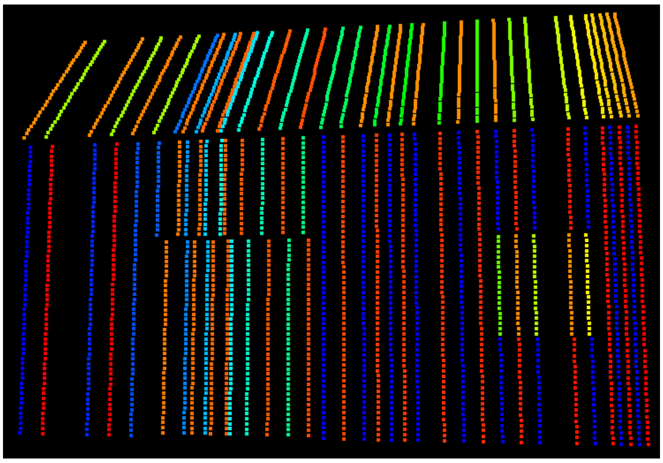


Figure 5.2 Outputs from the process breaking scanlines into different scan profiles process; different colours indicates different scan profiles; Adopted from Nguyen et al. (Submited and under revision-a)

5.2.1.2 Grouping of neighbouring scan profiles

For this simulated dataset, the angular threshold to check the parallelism between adjacent scan profiles is set to zero due to the perfect conditions of the simulation. After grouping, the scan profiles belonging to surface 1, 2 and 3 are automatically grouped into a single group, as their scan profiles are all parallel to each other. The scan profiles of surface 4 are put into a separate group. This is exactly what is expected from the approach, indicating that no problems or incorrect labelling occurred. The results are given in Figure 5.3.

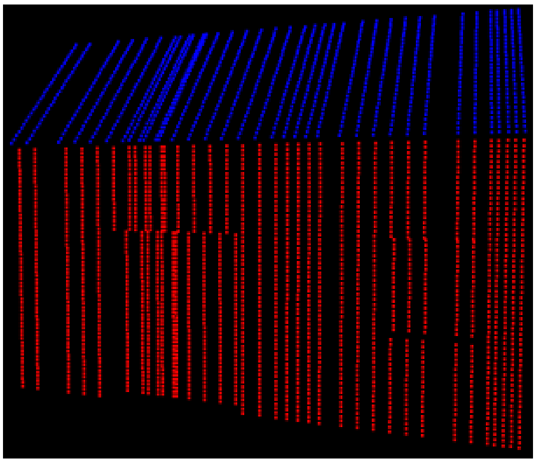


Figure 5.3 Different extracted scan profiles after grouping; Different colours indicate different groups; Adopted from Nguyen et al. (Submited and under revision-a)

5.2.1.3 Plane detection and segmentation based on the planarity value of the scan profiles groups

For this step, no parameter value is required. Using PSPS and the automatic planarity threshold adaption, all of the planes are perfectly detected and segmented. All of the scan profiles are assigned to their correct planar surfaces (Figure 5.4).

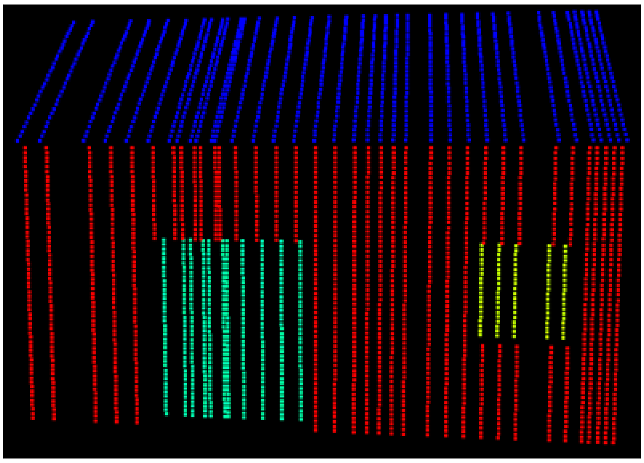


Figure 5.4 Different planes are perfectly extracted; different colours indicate different planes; Adopted from Nguyen et al. (Submited and under revision-a)

5.2.2 Comparison with other state-of-the-art plane detection and segmentation approaches

5.2.2.1 RANSAC and RANSAC-NV

As the simulated dataset has 0 mm accuracy and is error free, the distance threshold for RANSAC and RANSAC-NV can be set to zero. The spatial proximity between points is normally determined using the KNN or FDN algorithm. KNN is used in this research for RANSAC, as it guarantees the required point sample size is given. Regarding the *k* parameter for KNN, it needs to be set large enough to make sure that enough neighbouring points belonging to the same plane are selected and considered to be inliers. In other words, if the *k* parameter is too small, points of the same plane may be considered as belonging to different surfaces because it cannot cross between scanlines. An illustration of this issue is shown in Figure 5.5 with the *k* parameter being set to 10. Consequently, the four planes are segmented into multiple regions. This problem can be solved by introducing a merging process after the RANSAC method is performed. Alternatively, the issue can also be solved by increasing the value for the *k* parameter to ensure that the neighbourhood extends over these discontinuities. In this example, the over-segmentation issue could be avoided by setting the *k* parameter to 30 as shown in Figure 5.6. Nevertheless, some points of the roof are incorrectly labelled as wall points. The reason is that these points are within the distance threshold and will be allocated to the larger surface with the larger consensus set. This is one of the limitations of the RANSAC algorithm when the normal vectors are not taken into account. Similar problems occur if FDN is used when setting the distance threshold. The reason is that these incorrectly assigned points have distances to both the roof surface and the wall surface smaller than the distance threshold, as well as the number of points of the wall being higher than the number of points of the roof.

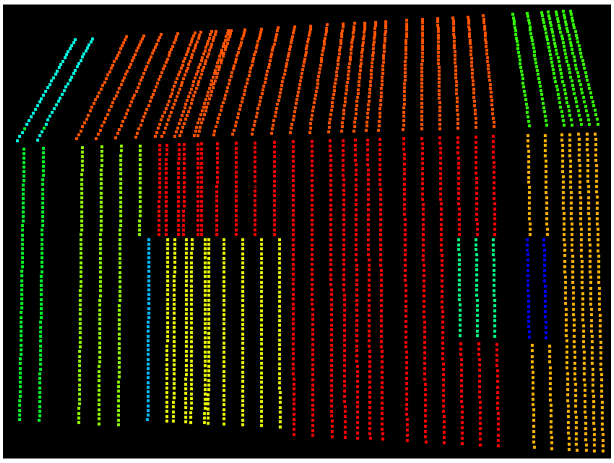
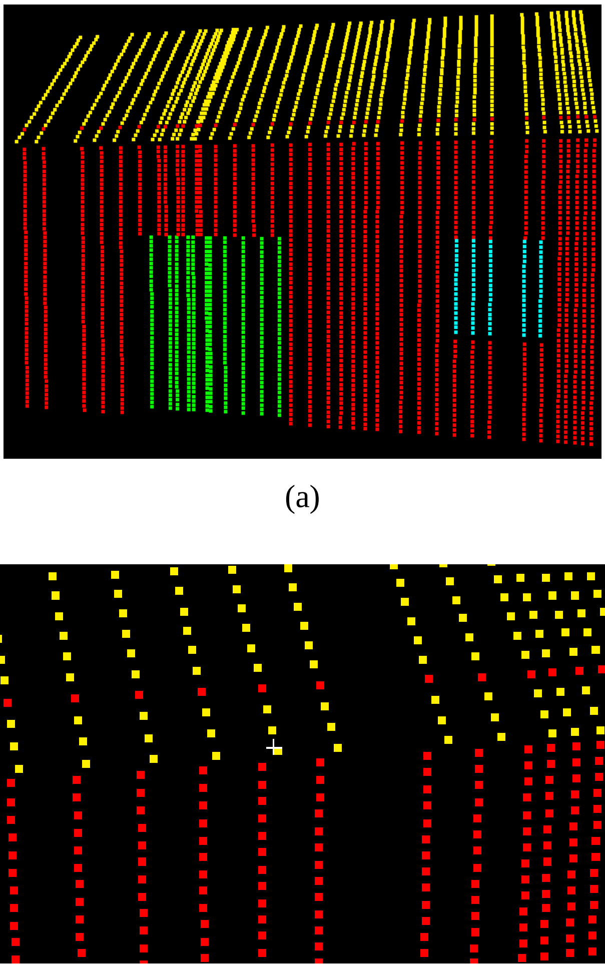


Figure 5.5 Segmentation output using RANSAC (*k*=10) ; Adopted from Nguyen et al. (Submited and under revision-a)

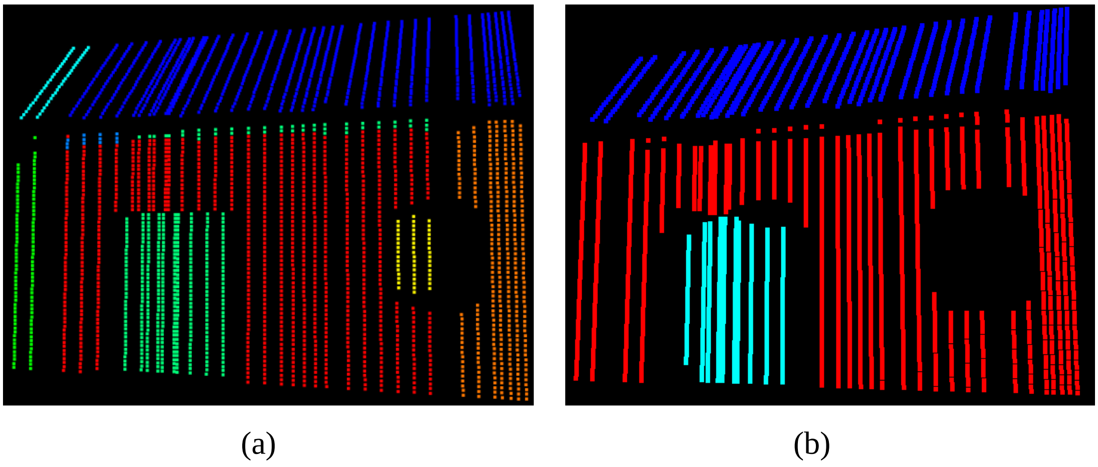


(a)

(b)

Figure 5.6 Segmentation output using RANSAC (*k*=30): (a) overview and (b) closer examination of the roof; different colours indicate different planes; Adopted from Nguyen et al. (Submited and under revision-a)

With regards to the RANSAC-NV, besides the distance threshold parameter, two additional parameters are required. Specifically, the number of neighbouring points *k* to be used to estimate the normal vectors and determine the spatial proximity between points, and the angle threshold is used for region growing process. The angle threshold is set to 0 degree because of the simulated nature of the dataset. Nevertheless, there is no guideline for choosing the values for the number of neighbouring points *k* even for the errorless dataset and for when the point density is known. It can only be ensured that *k* is large enough, so the neighbouring points are selected from at least two scanlines to achieve an adequate surface normal definition. Different values for this parameter are applied to test the effect in order to find the best output for comparison. Explicitly, the *k* parameter is set to 15, 30 and 40. These different *k* values provided different outputs which are presented in Figure 5.7. For *k* = 15, all planes are extracted. However, the outputs are over-segmented (i.e. one plane has been considered as more than one plane). For instance, the building façade surface is segmented into 5 different regions. Increasing the number of neighbourhood points will increase the possibility of having points from other surface in the neighbouring point group. Consequently, the probability of points on the same surfaces having different normal vector directions is higher. Indeed, for *k* = 30, only three planes are detected. In order to provide a good visualisation, only detected planes that have more than 5 points are shown in Figure 5.7.



(a) (b)

Figure 5.7 Segmentation outputs using RANSAC considering the normal vectors; (a) *k* = 15 and (b) *k* = 30; Adopted from Nguyen et al. (Submited and under revision-a)

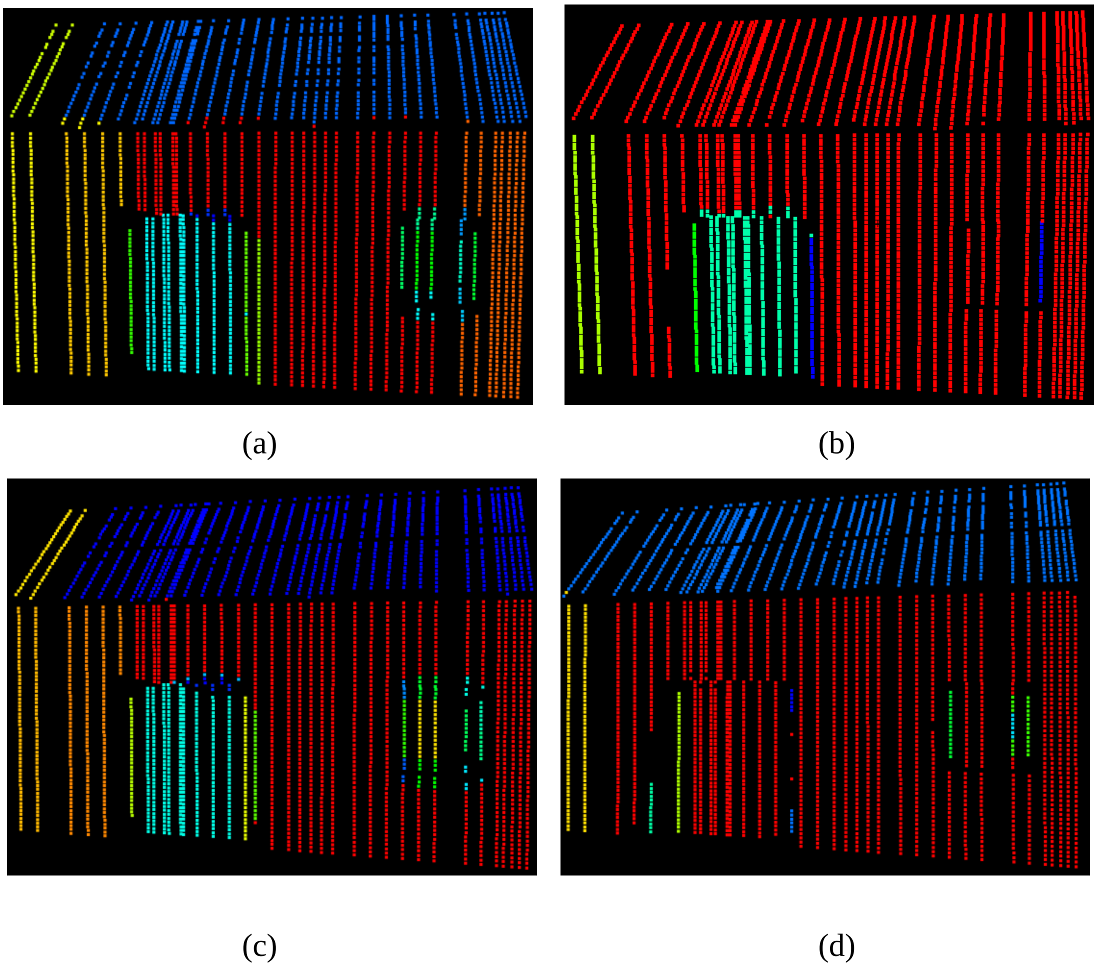
5.2.2.2 RDPCA

Although this dataset is errorless, the percentage of outliers could not be set to zero because in this context the outliers in the selected neighbouring points of each query point include not only noise points, but also points from other adjacent surfaces. Similar to RANSAC-NV, there is also no guideline in determining the two most important parameters of the RDPCA method (i.e. the number of neighbouring points (k) and the percentage of outliers (ϵ)). Therefore, different combinations of these two parameters are applied (Table 5.2). These parameters were found through empirical testing and are set to 15, 20, 30 and 40 for the k parameter and to 15%, 25%, 35% and 50% for the percentage of outliers (ϵ). The angle threshold parameter is again set to 0 degree.

Number of neighbouring Points (k)	Percentage of Outlier (ϵ in %)
15	15
20	25
30	35
40	50

Table 5.2 Different setting parameters of RDPCA for the simulated dataset

Similar to RANSAC-NV, as can be seen from the outputs of RDPCA the RDPCA is also sensitive to the parameter values when applied to sparse and heterogeneous MLS sparse point clouds, with different parameters produce different segmentation outputs (Figure 5.8). For instance, for k = 20 and ϵ = 25%, the roof surface is detected as two separate regions as the two scan profiles on the left are spaced farther away from the rest of the points (Figure 5.8(a)). This can happen when the vehicle with the MLS speeds up or one of the scanlines drops out due to occlusion. The segmented outputs of the other three features are even worse. Each of them are over-segmented into at least 3 regions. The roof can be properly segmented if k = 40 and ϵ = 50% (Figure 5.8(d)). Nevertheless, in this case RDPCA fails in detecting and segmenting the other three surfaces (i.e. they are considered as a single plane). The reasons for the failures of RDPCA for this data comes from the limitations of the KNN and the issues of normal vector estimation (as discussed in Chapter 2 and Chapter 3). Furthermore, another reason for the failures is caused by the process of using KNN for automatically specifying the values for ED_{th} and OD_{th}. As discussed in section 5.1.2.1, the k parameter needs to be large enough to make sure that the points belonging to the same plane are chosen as neighbourhood of points. Nevertheless, this leads to other issues with RDPCA, such as more outliers may be selected as neighbouring points if the k is set to be large. The values for ED_{th} and OD_{th} are automatically determined based on the median values. These will become large due to the ED and OD values from these selected outliers are large. Consequently, over-segmentation occurs.


(a)

(b)

(c)

(d)

Figure 5.8 Segmentation outputs from RDPCA; different colours indicate different planes; (a) k = 20, ϵ = 25%; (b) k = 40, ϵ = 25%; (c) k = 30, ϵ = 50% and (d) k = 40, ϵ = 50%; Adopted from Nguyen et al. (Submited and under revision-a)

5.2.2.3 Cabo

In the case of the Cabo method, the parameters are obtained based on the characteristic of the simulated dataset. For instance, the Douglas Peucker tolerance is set to zero cm, the angular parameter for checking the parallelism is set to zero degrees and the distance threshold between end nodes is set to 10 cm (as the maximum scan profile spacing of this dataset is 9.5 cm). From Figure 5.9, it can be seen that the Cabo method can perfectly extract the roof plane. However, the other planes (i.e. building façade, door and window) are labelled as a single surface (Figure 5.9). The reason for this is that the Cabo method only takes into account the line arrangement condition, and the scan profiles of these three surfaces are parallel to each other.

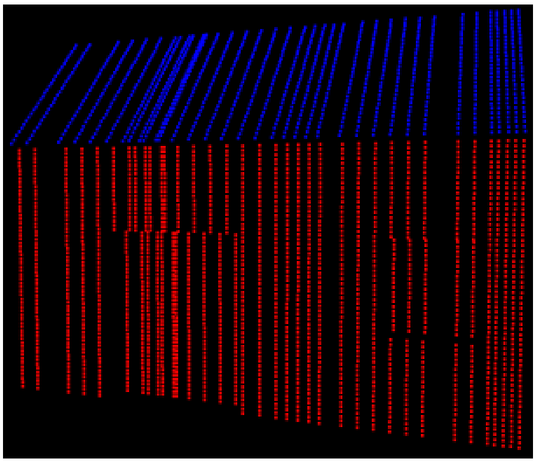


Figure 5.9 Segmentation outputs using Cabo; Adopted from Nguyen et al. (Submited and under revision-a)

5.2.3 Conclusions for the simulated dataset

As the simulated dataset is considered to be collected by a MLS system free from errors and noise, some of the parameter values can be ideally determined for RANSAC, RANSAC–NV, Cabo and PSPS. However, RANSAC-NV and RDPCA still needs certain parameter values to be determined empirically in order to obtain the best outcome. RANSAC performs better than RANSAC-NV in this case. Different parameter values were applied in order to find the best outputs from RDPCA. The outputs from RDPCA are shown to be significantly influenced by the parameter values for the simulated dataset. Cabo fails in detecting and segmenting adjacent planes that have similar orientations. Finally, among these five methods, PSPS provides the best outputs for plane detection and segmentation.

5.3 Evaluation using the target datasets

The target (shown in Figure 5.10) that comprised of planar segments was designed in order to fulfil the requirement for a successful plane-based registration process. This means that it ensures the existence of at least three non-parallel planes that intersect at a unique point (Chapter 4). This allows the target to be used to ensure the success of the least-squares plane fitting adjustment in the case where all of the planar surfaces in the scanning area have normal orientations that are orthogonal with the trajectory of MLS system. For this experiment, the target was placed in a car park, in a controlled, relatively open environment. The target was scanned three times at two different speeds (e.g. 20 km/h and 30 km/h) and with three different distances (e.g. 7 metre, 27 metre and 40 metre) using the Dynascan MDL S250 MLS system. The final point clouds used in this analysis and their properties are presented in Table 5.3.

Due to the specification of the scanning system, the speed of the vehicle for each run, and the distances between the target and the scanning system, the three captured point clouds are sparse. Each of the three captured point clouds has different point densities and a noise level specific to the capture MLS system. Specifically for dataset 1, the average point spacing is 33 cm and the average profile spacing is 32 cm; for dataset 2, the average point spacing is 10 cm and the average profile spacing is 47 cm; and for Dataset 3, the average point spacing is 15 cm and the average profile spacing is 47 cm. Moreover, in order to evaluate the robustness of all of the discussed plane detection and segmentation methods against noise, dataset 4 is created by adding additional points representing noise to dataset 2. Twenty such points with a standard deviation of 10 cm and a minimum error residual of 2.6 cm are randomly added to the dataset. Using these specifications, the additional points should be ranked as outliers defined by the specifications of the MLS. Specifically for the three planes in the dataset, seven noise points were added to plane 1 (T1), five noise points were added to plane 2 (T2) and eight noise points were added to plane 3 (T3). Therefore, the number of noise points for all three planes were approximate 30%. Figure 5.11 shows the visualisation of the 4 point cloud datasets; Table 5.4

summarises the number of scan profiles and number of points for each plane of the target in different datasets.

These four sparse and heterogeneous MLS point clouds datasets are especially challenging for plane detection and segmentation, because the planes are relatively small and represented by only a limited number of points in each of the four datasets. As a result, they are highly suitable test objects for evaluation purposes.

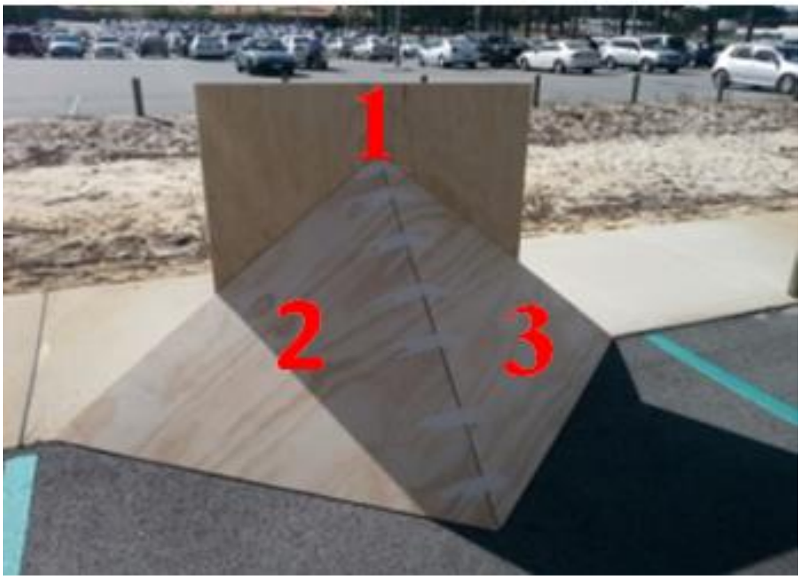
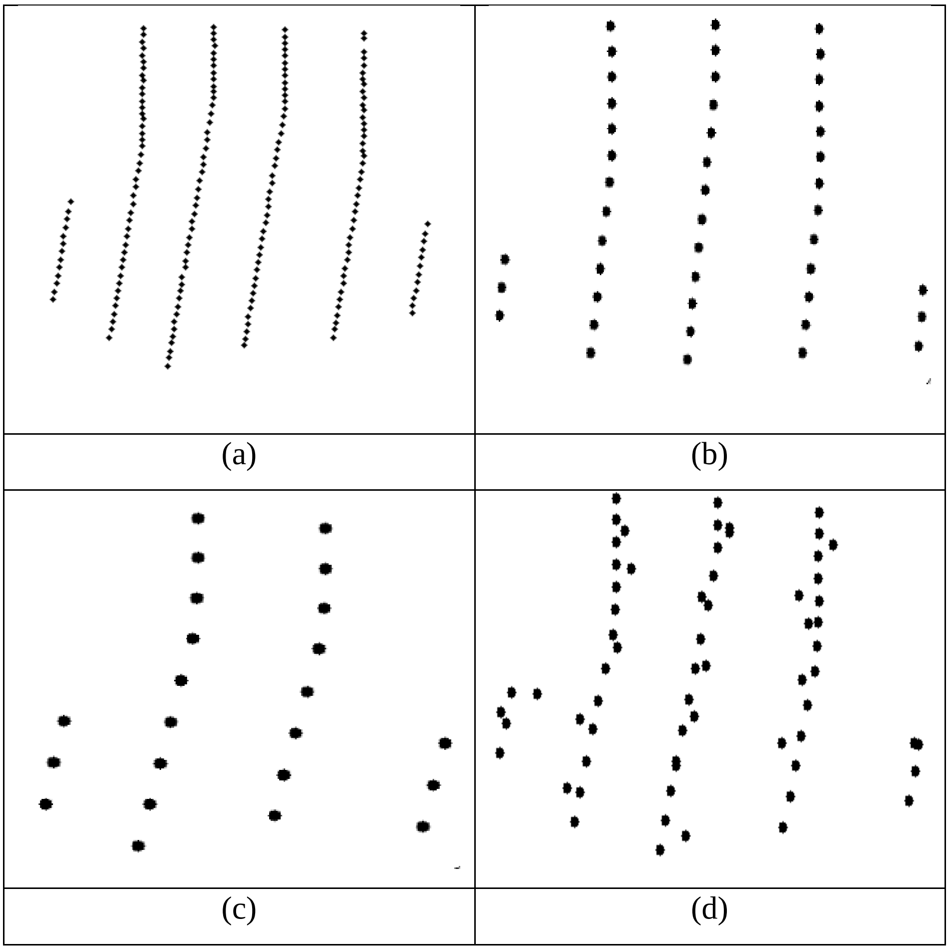


Figure 5.10. The target

Data set	Vehicle speed (km/h)	Distance (metres)
1	~20	~7
2	~30	~27
3	~30	~40
4	~30	~27

Table 5.3 Vehicle speeds and distances from target to scanner of different datasets



(a)

(b)

(c)

(d)

Figure 5.11 The visualisations of datasets 1 to 4: (a) dataset 1; (b) dataset 2; (c) dataset 3 and (d) dataset 4; Adopted from Nguyen et al. (Submited and under revision-a)

Data set	Plane T1	Plane T2	Plane T3
1	4/64	3/71	3/64
2	3/16	2/10	3/19
3	2/7	2/8	2/8
4	3/16	2/10	3/19

Table 5.4 Number of scan profiles/number of points on each plane of the target; Adopted from Nguyen et al. (Submited and under revision-a)

In order to obtain a benchmark for evaluation, the planar surfaces of the target in the four testing datasets are manually extracted based on the known target design. The results are shown in Figure 5.12.

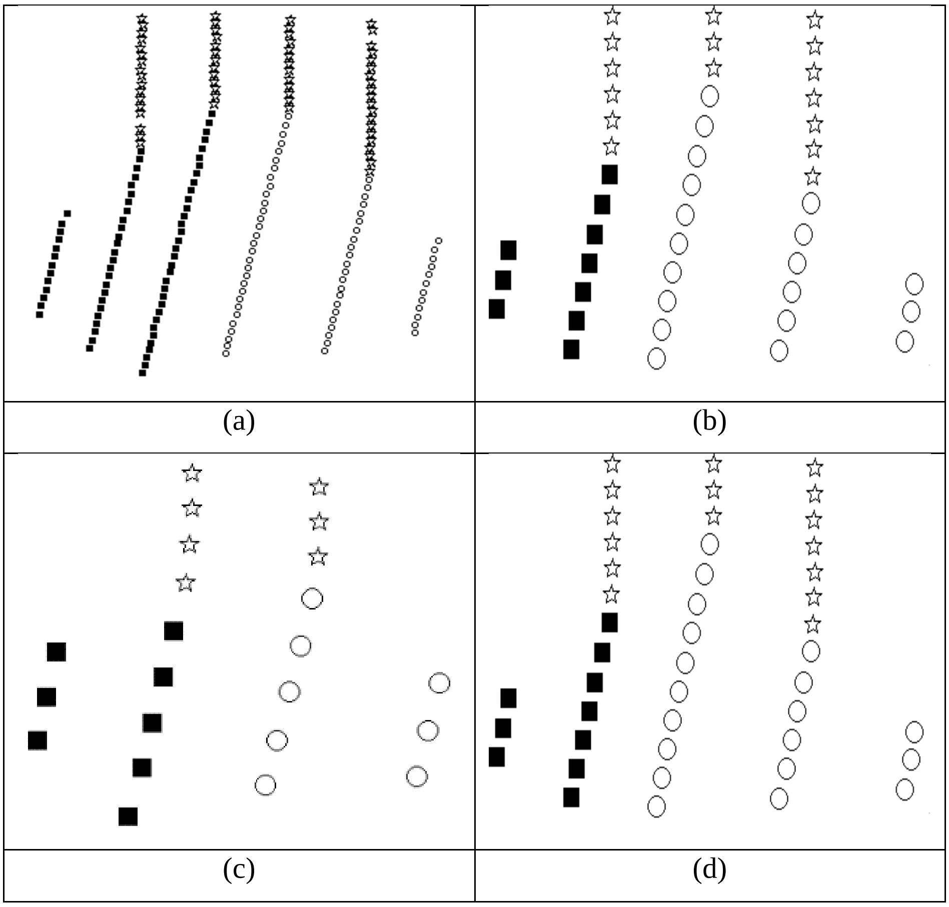
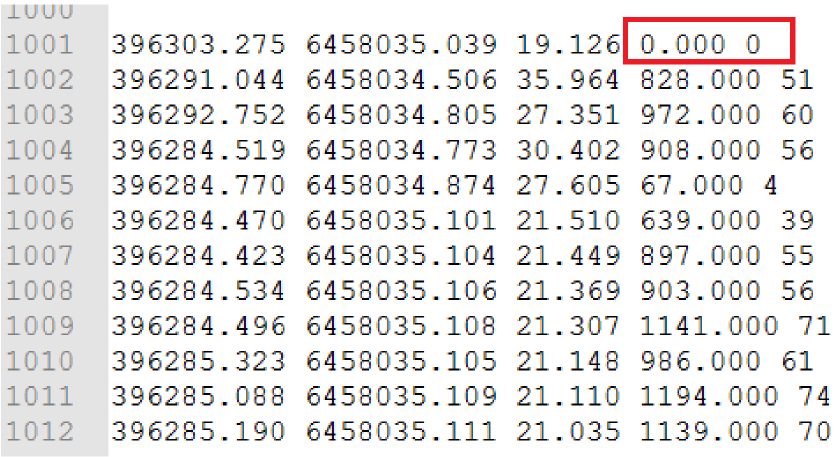


Figure 5.12 Manual segmentation outputs of the four datasets: (a) dataset 1; (b) dataset 2; (c) dataset 3 and (d) dataset 4; different symbols indicate different planes; Adopted from Nguyen et al. (Submited and under revision-a)

5.3.1 Evaluation of the segmentation results using the PSPS method

5.3.1.1 Forming scan profiles

The captured point clouds were split into different scanlines based on the laser positions and the point index. The laser position points of each scanline is defined as “break points”, and therefore captured point clouds can then easily be segmented into different scanlines. In the point cloud data file captured using MDL Dynascan S250 the laser position point has an intensity value equal to zero (Figure 5.13).



```
396303.275 6458035.039 19.126 0.000 0
396291.044 6458034.506 35.964 828.000 51
396292.752 6458034.805 27.351 972.000 60
396284.519 6458034.773 30.402 908.000 56
396284.770 6458034.874 27.605 67.000 4
396284.470 6458035.101 21.510 639.000 39
396284.423 6458035.104 21.449 897.000 55
396284.534 6458035.106 21.369 903.000 56
396284.496 6458035.108 21.307 1141.000 71
396285.323 6458035.105 21.148 986.000 61
396285.088 6458035.109 21.110 1194.000 74
396285.190 6458035.111 21.035 1139.000 70
```

Figure 5.13 Point cloud data captured using MDL Dynascan S250

Next, the scanlines can then be split into scan profiles. The required parameters for this step can be either selected using the standard values, or determined based on the accuracy of the laser scanner of the captured MLS system (as discussed in section 3.4). Specifically, the parameters used for this test is illustrated in Table 3.1.

Parameter	**Value**
nnp	7 points
T	10 iteration
A_i	10 degree
rd	20 mm
DL_i	20 mm
Planarity factor	4 times

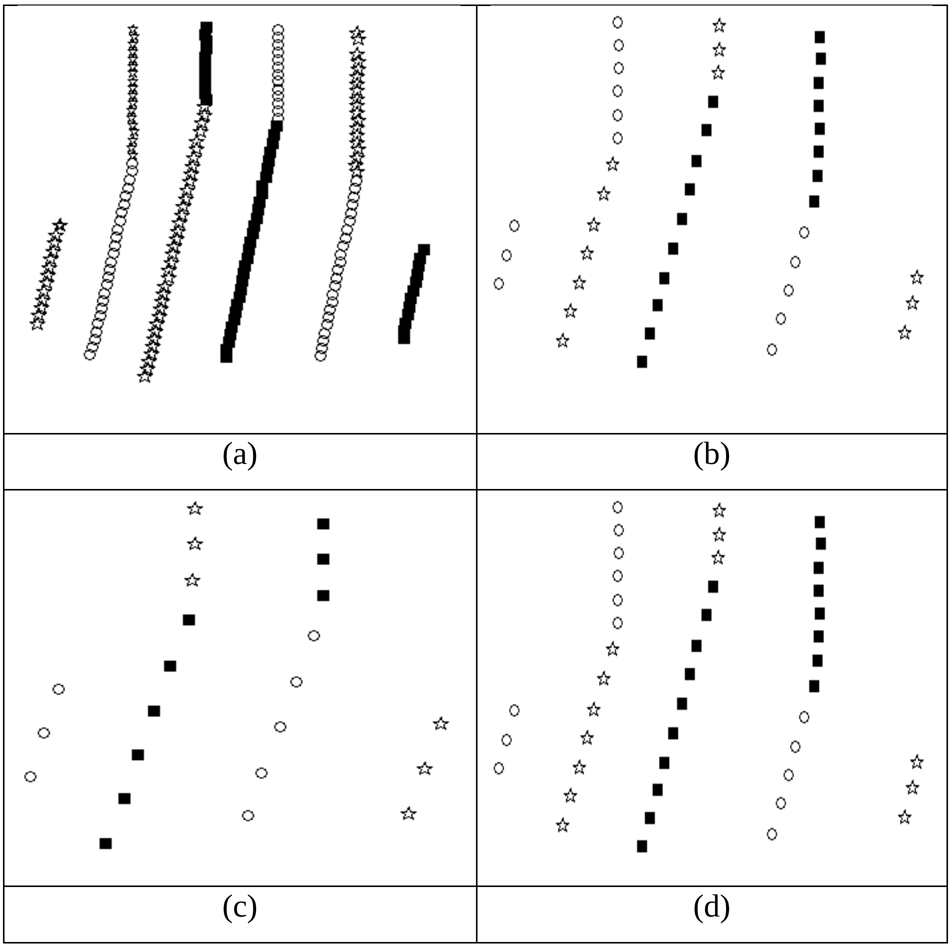
Table 5.5 Standard parameters setting for the target datasets

In order to investigate the influences of these five parameters on the results, different values for these parameters are applied as outlined in Table 5.6. Specifically, the neighbourhood size parameter *nnp* is set to 5 (i.e. 5 points captured before and 5 points captured after the query point), 7 and 10; the angle threshold A_i were set 5, 7 and 10 degrees; and the number of iterations *T* was set to 5, 7 and 10 for both datasets**.** The threshold for the modified RANSAC line fitting, *rd* and DL_i was set to 2 cm and 2.5 cm, which is twice the range accuracy of the MDL Dynascan S250 mobile laser scanning. Furthermore, the planarity factor was set to 3, 4 and 8 times. Different combinations of these parameter values were applied to the datasets.

Parameter	**Value**
nnp	5, 7 and 10 points
T	5, 7 and 10 iteration
A_i	5, 7 and 10 degree
rd	2 and 2.5 cm
DL_i	2 and 2.5 cm
Planarity factor	3, 4 and 8 times

Table 5.6 Different parameters values for the proposed method for the target datasets

The same results were achieved for all the parameter values, including the dataset with additional noise points. These results are shown in Figure 5.14 and illustrates that all the scan profiles were successfully detected and segmented, and all the noise point were removed from dataset 4

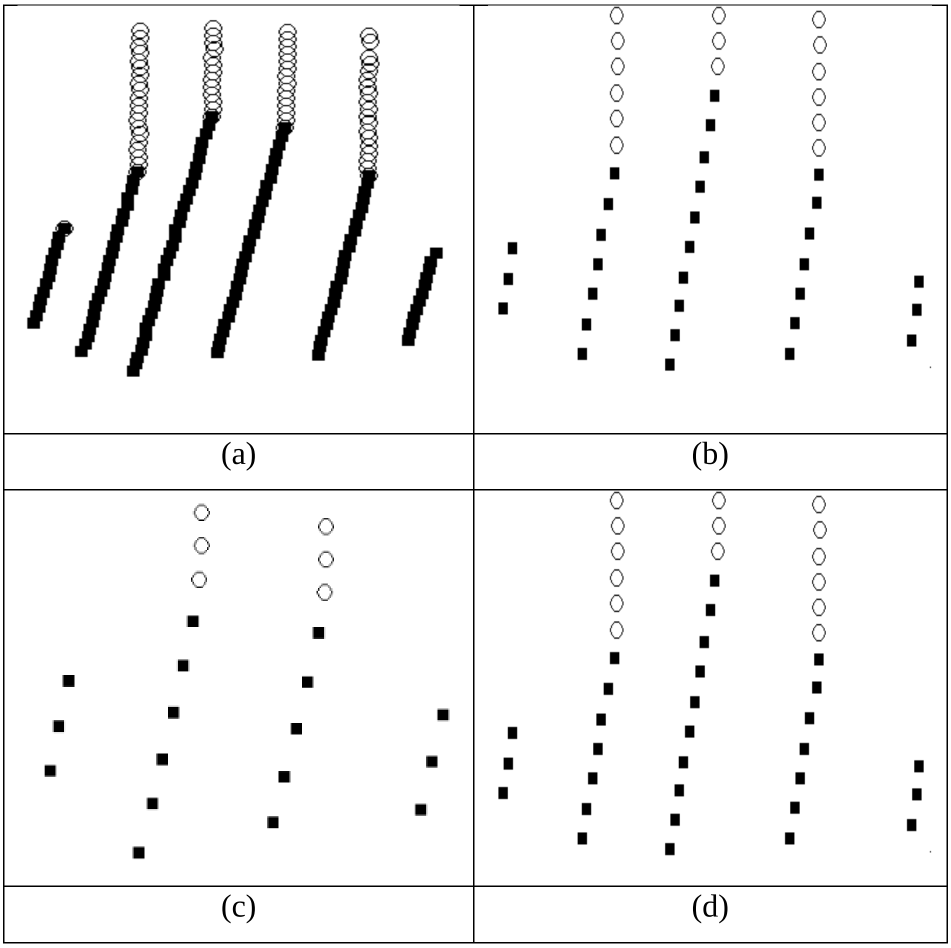


(a) (b) (c) (d)

Figure 5.14 Outputs of the breaking scanlines into different scan profiles process of the target datasets; different symbols indicate different scan profiles; Adopted from Nguyen et al. (Submited and under revision-a)

5.3.1.2 Grouping of neighbouring scan profiles

After the scan profiles are formed, unlike the simulated dataset, the standard value for the angle threshold parameter is set to 10 degrees based on previous discussions in section 3.4. However, different values for angle threshold parameter (e.g. 5, 7 and 10 degree) were applied to investigate its potential impact on the solution. Similar to the previous step, changing the value of the required parameter caused no change to the outputs. For all of the four datasets, scan profiles belonging to plane 1 are assigned to one group (Figure 5.15). Meanwhile, as the angle between plane 2 and the ground surface, and the angle between plane 3 and the ground surface are similar, the scan profiles of these two surfaces are parallel to each other. Hence, they are correctly labelled as belonging to the same group. Overall, no incorrect grouping was present.



(a)

(b)

(c)

(d)

Figure 5.15 Different scan profiles assigned into different groups; different symbols indicate different groups; Adopted from Nguyen et al. (Submited and under revision-a)

5.3.1.3 Plane detection and segmentation based on the planarity value of scan profile groups

In the final step, the planarity threshold parameter is applied to segment the groups into different planar features. The planarity threshold is used to segment adjacent planes assigned to the same group (section 5.2.1.3) e.g. plane 2 and 3 in the target datasets. In other words, it is used to segment adjacent planes that have their scan profiles parallel to each other. In order to investigate and demonstrate the planarity values discussed in section 3.3, the planarity values of different scan profiles of different target planes were calculated. Table 5.7 illustrates the values of the planarity condition using different combination of inliers and outlier for scan profiles. It can be seen that if the planarity value is calculated from three inlier scan profiles, then the value will be approximately the same as if it is calculated from only two of the inlier scan profiles. Meanwhile, the case containing the outlier scan profile is an order of magnitude larger than the other cases. For instance, for plane 3 in data set 1, these three calculated planarity values are 3.3e-05, 9.7e-05 and 1.2e-03.

	2 scan profile (L_i and L_{i+1})	3 scan profile (L_i, L_{i+1} and L_{i+2})	2 scan profile + outlier scan profile (L_i, L_{i+1} and L_{i+2})
Dataset 1 – P1	5.7e-05	4.7e-05	X
Dataset 1 – P2	2.4e-05	3.2e-05	2.1e-03
Dataset 1 – P3	**3.3e-05**	**9.7e-05**	1.2e-03
Dataset 2 – P1	7.1e-06	8.4e-06	X
Dataset 2 – P2	**3.3e-05**	X	**1.5 e-04**
Dataset 2 – P3	3.8e-05	**5.8e-05**	1.4e-02
Dataset 3 – P1	2.7e-05	X	X
Dataset 3 – P2	8.0e-05	X	3.8e-03
Dataset 3 – P3	1.6-05	X	4.1e-03

Table 5.7 The planarity values calculated from scan profiles of plane 2 and 3 in datasets 1 to 3: X indicates no value due to those planes contain only two scan profiles; the highlighted values are discussed in the text

An exception is plane 2 in dataset 2 where there is only a small difference. This is because this scan profile could potential belong to both planes. Specifically, the planarity value calculated from the two “inlier” scan profiles (i.e. SP4 and SP5) and the one “outlier” (i.e. SP3) is only about five times larger (e.g. 1.5e-04 and 3.3e-05) (Figure 5.16). This is because the scan profile SP3 is close to the boundary of two planes. As can be seen from Figure 5.16 and Figure 5.17, the scan profile in the middle (i.e. SP3) can potential belong to both, i.e. plane 2 (containing SP5 and SP4) or plane 3 (containing SP1 and SP2). By performing careful visual inspections (Figure 5.18) it can be seen that SP3 belongs to the same plane as SP1 and SP2 (i.e. plane 3) instead of SP4 and SP5 (e.g. plane 2). Consequently, if the planarity factor is set larger than 5 times (e.g. 6 times in this case) SP3 may be assigned to belong to plane 2, which is incorrect. However, thanks to the extra step that analyses the planarity of these two planes (section 3.2.3), SP3 was finally assigned to plane 3 as the planarity value of SP3 with SP1 and SP2 (e.g. 5.8e-05) is smaller than the planarity value of SP3 with SP4 and SP5 (e.g. 1.5e-04).

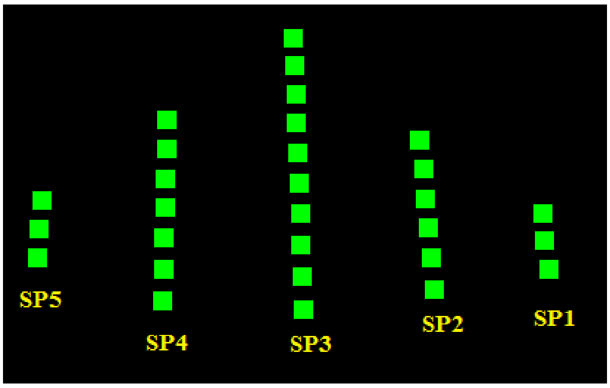


Figure 5.16 Front view of plane 2 and 3 in dataset 2

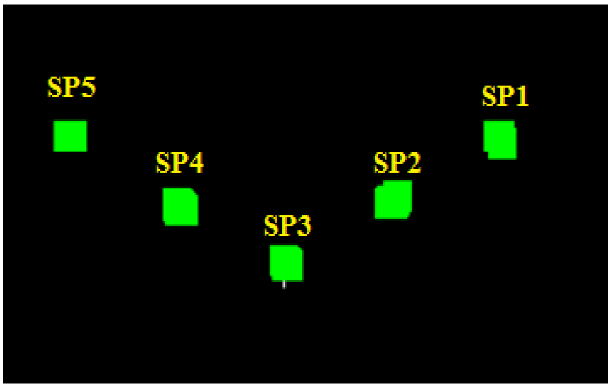


Figure 5.17 Plane 2 and 3 were projected onto the plane that is orthogonal with their scan profiles

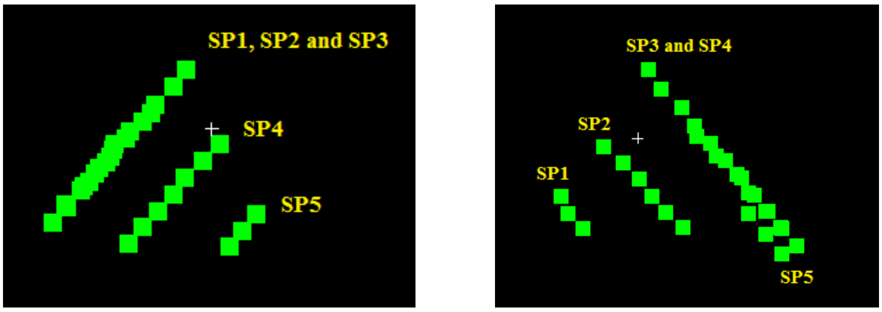


Figure 5.18 Side view of plane 2 and 3

As a result, planes of the target in all of the datasets were properly detected and segmente, without over-segmentation or under segmentation, and the noise points were eliminated as shown in Figure 5.19.

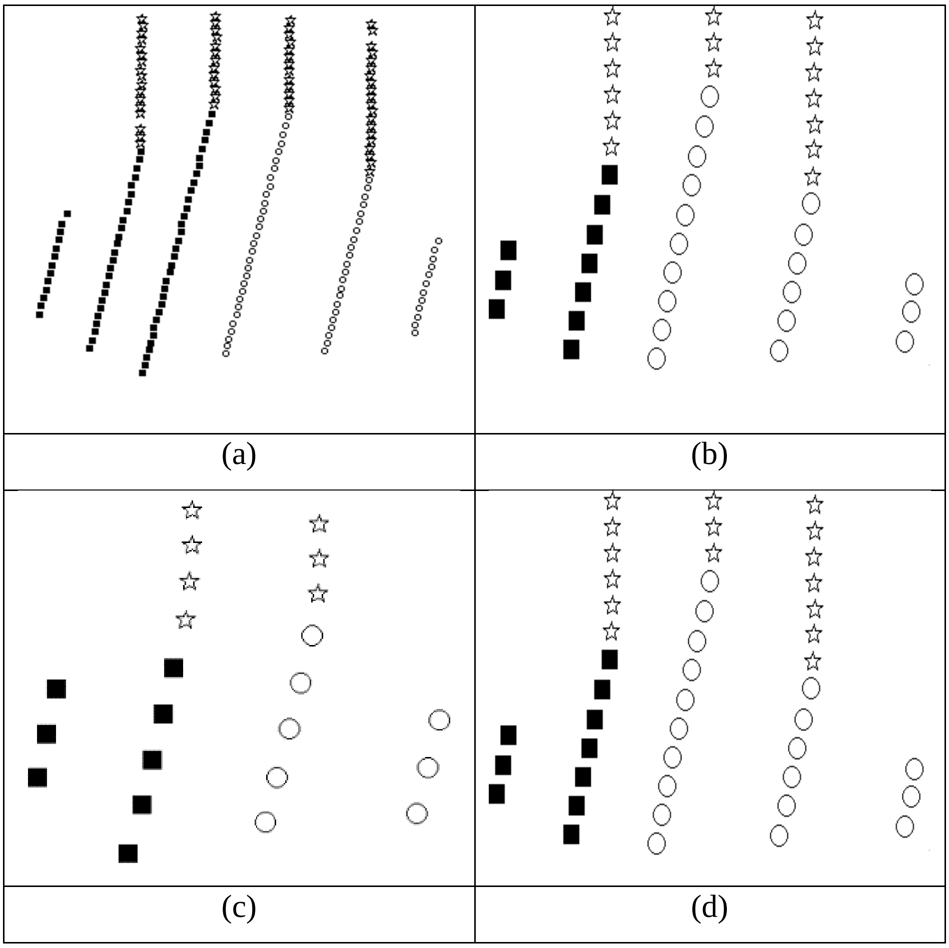


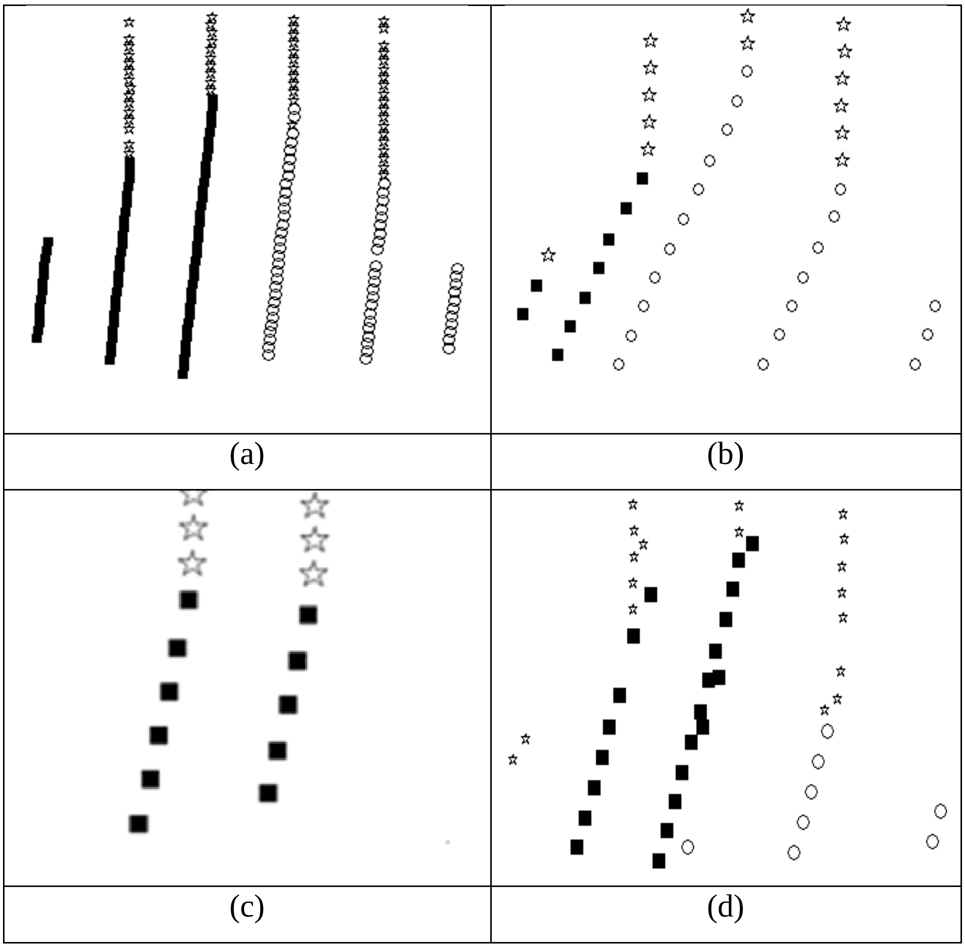
Figure 5.19 Segmentation results of the four datasets using the proposed method: (a) dataset 1; (b) dataset 2, (c) dataset 3, (d) dataset 4; Adopted from Nguyen et al. (Submited and under revision-a)

5.3.2 Comparison of the segmentation results with other state of the approaches

The quality of the current plane detection and segmentation approaches is evaluated based on the number of over-segmentation and under-segmentation features. Under segmentation occurs when more than one real object is merged. Over-segmentation occurs when a single object is labelled as more than one object. This work focuses more on the quality of the plane with respect to the inputs for the registration process. The inputs for the least-squares plane fitting adjustment come from both the *slave* point cloud and the *master* point cloud (Chapter 4). Thus, the comparisons are separated into three parts. In the first part, the segmentation outputs were evaluated regarding the number of over-segmentation and under segmentation features. The segmentation outputs were then compared regarding the number of correctly and incorrectly segmented points. Finally, the segmentation outputs were evaluated in regards to the plane parameters.

5.3.2.1 Evaluation of RANSAC and RANSAC-NV

Unlike the simulated dataset, setting the distance parameter value for the RANSAC algorithm needs to be empirically determined. Therefore, different values for the distance parameter are applied as shown in Table 5.8 in order to find the best outputs. As the sensitivity of the distance threshold parameter was analysed and discussed in other research (Fujiwara et al., 2013; Tarsha-Kurdi et al., 2007) it is not the focus of this research. For this dataset, the impact of the local neighbourhood parameter *k* is only minor as long as it is large enough to cross scanlines. The distance threshold was set to 4 cm and the *k* is set to 20. Figure 5.20 shows the output of RANSAC for the datasets. RANSAC works quite well with datasets 1 and 2, with only a few points incorrectly assigned. As can be seen from Figure 5.20b one point of plane 2 was assigned to plane 1 due to the spatial proximity between points (discussed in 5.1.2.1). RANSAC has problems in detecting plane 2 and 3 of the target in dataset 3. The reason for this failure is that points in the two middle scan profiles form an erroneous planar object that has more points than plane 2 and 3. Regarding dataset 4, if the distance threshold is set too large, noise points will be considered as inliers. However, if it is set too small, RANSAC cannot detect the planes. Furthermore, in some case an erroneous plane can be formed by outliers and a set of inliers (discussed in section 2.2.4). Consequently, RANSAC could not provide accurate segmentation results for dataset 4.



(a)

(b)

(c)

(d)

Figure 5.20 Segmentation outcomes from RANSAC without normal vectors: (a) dataset 1; (b) dataset 2, (c) dataset 3, (d) dataset 4; Adopted from Nguyen et al. (Submited and under revision-a)

The outputs of RANSAC-NV are influenced by three parameters: (1) the number of local neighbouring points for the normal vector estimation and the spatial proximity between points determination; (2) the distance threshold and (3) the angle threshold. While the angle threshold was set to 10 degrees (section 3.1.3.3), various values for neighbourhood size and distance threshold were applied to get the best output (Table 5.8). Due to the number of points of the target in dataset 3 being very small (i.e. there are only seven, eight and eight points on plane 1, 2 and 3 respectively) and the problems this caused in the normal vector estimation, RANSAC-NV completely fails to detect planes in this dataset. With regards to the other three datasets, the distance threshold of 0.05 metres seems to be the most suitable value based on the results shown in Figure 5.21. The neighbourhood size parameter needs to be chosen differently for each dataset because it depends on the point density of the point clouds. Specifically, the desired neighbourhood size parameter needs to be set to 30, 10 and 30 for dataset 1, 2 and 4 respectively in order to get the best outputs. RANSAC-NV again shows poorer performances than RANSAC in the target datasets because of the normal estimations problems due to the sparseness of the point clouds. RANSAC-NV can detect all of the target planes in dataset 1, but with number of missing points on plane 2 and 3 (Figure 5.21d). The segmentation outputs from dataset 2 and 4 are not accurate in terms of selecting the correct points.

Neighbourhood size	Distance threshold (in metre)
5	0.01
10	0.025
15	0.040
20	0.050
30	0.100

Table 5.8 Parameters list for the RANSAC-NV

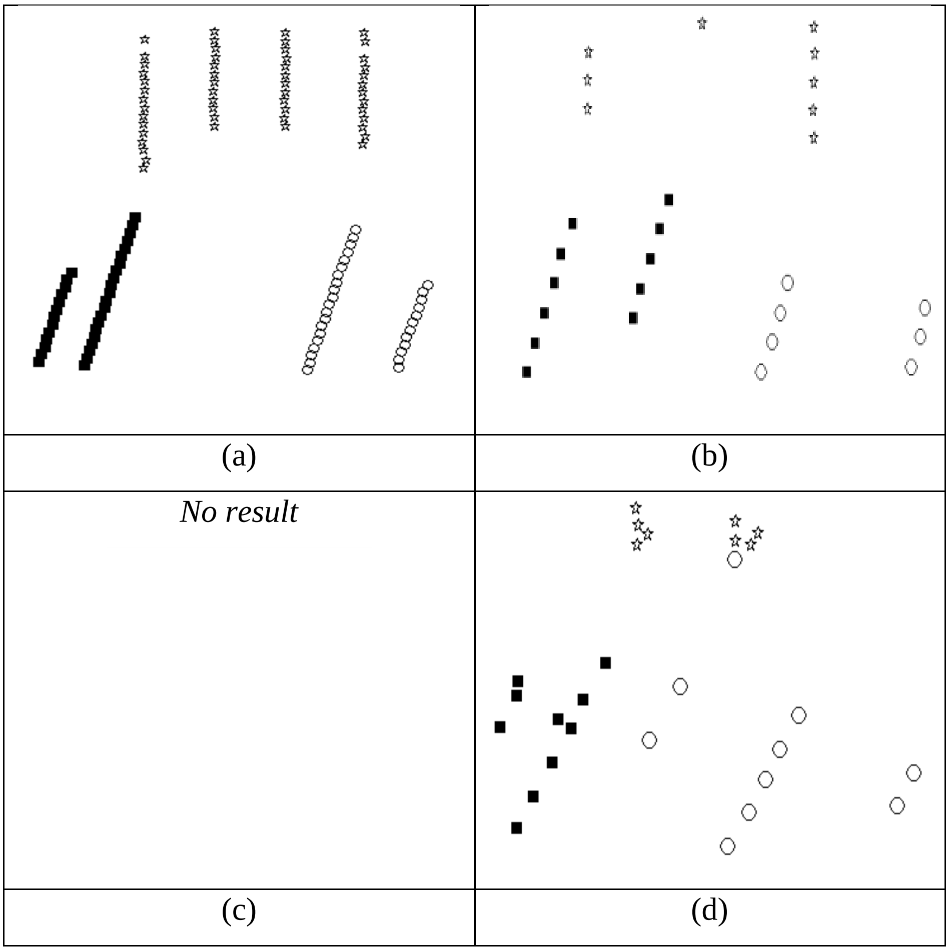
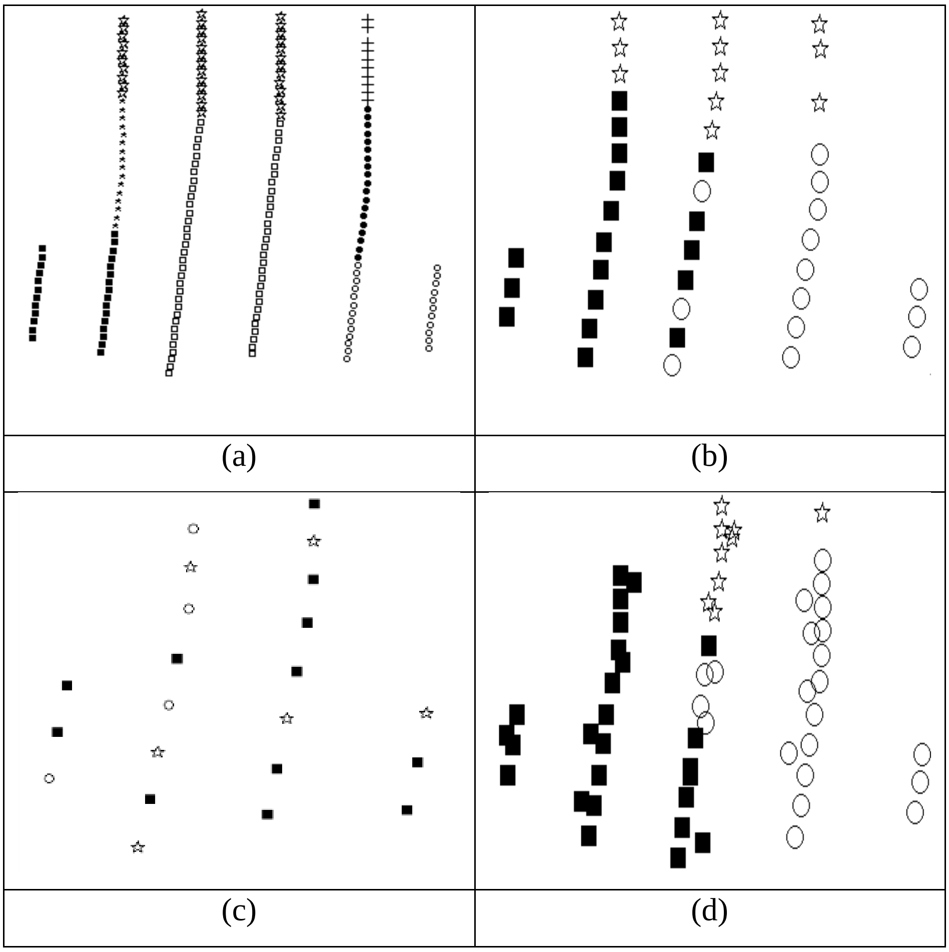


Figure 5.21. Segmentation outcomes from RANSAC: (a) dataset 1; (b) dataset 2, (c) dataset 3, (d) dataset 4; Adopted from Nguyen et al. (Submited and under revision-a)

5.3.2.2 Evaluation of the segmentation using RDPCA

Similar to the experiment with the simulated dataset, different input values are applied in order to evaluate their effects and to find the best outputs that RDPCA can provide. These are described in Table 5.2. The output with the neighbourhood size k = 30 and the percentage of outlier ϵ = 25% are shown in Figure 5.22. It can be seen from Figure 5.22(a) that RDPCA considers that there are seven planes in the point clouds of the target in dataset 1 instead of three. So, while all of the planes of the target are detected in dataset 1, many points are incorrectly assigned into additional, over segmented surfaces, especially with points near the boundaries of plane 2 and 3 (Figure 5.22(a)). RDPCA can detect all of the three planes in dataset 2 with some incorrectly segmented points. Meanwhile, it completely failed when applied to dataset 3 (Figure 5.22(c)). In contrast, all of the three planes are detected in dataset 4

(Figure 5.22(d)). Nevertheless, RDPCA fails in removing the noisy points. Consequently, many of them are labelled as belonging to one of the three planes of the target.



(a)

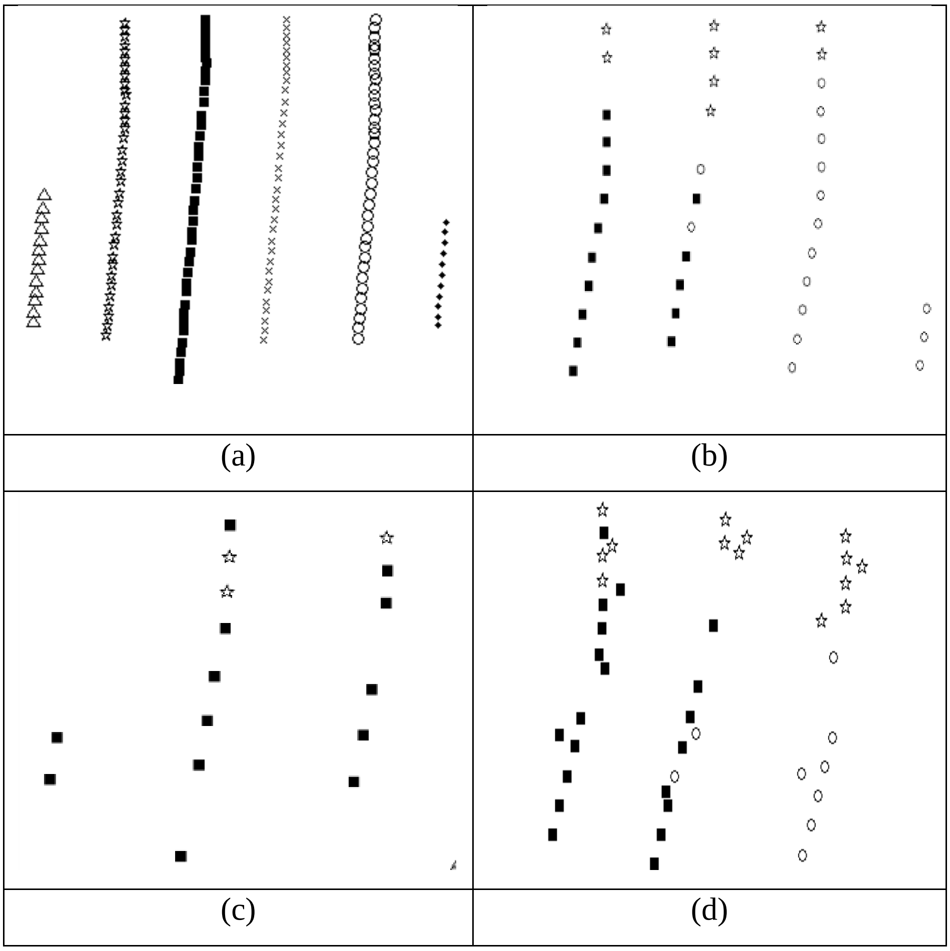
(b)

(c)

(d)

Figure 5.22. Segmentation outcomes from RDPCA k=30; ϵ = 25%):: (a) dataset 1; (b) dataset 2, (c) dataset 3, (d) dataset 4; Adopted from Nguyen et al. (Submited and under revision-a)

Figure 5.23 illustrates the output with k = 20 and ϵ = 25%. The target of dataset 1 was over segmented into six planes (Figure 5.23(a)), in which each scanline was considered as a plane. The explanation for this output is due to the small value for k (e.g. 20). The neighbouring points were selected only from the same scanline as the query points due to the point spacing distance being about ten times smaller than the profile spacing distance. RDPCA can detect plane 1 and 3 in dataset 2. However, it failed in segmenting plane 2 (Figure 5.23(b)). Again, it still completely failed in dataset 3 Figure 5.23(c). Only plane 1 was partly segmented in dataset 4 (Figure 5.23(d)).



(a)

(b)

(c)

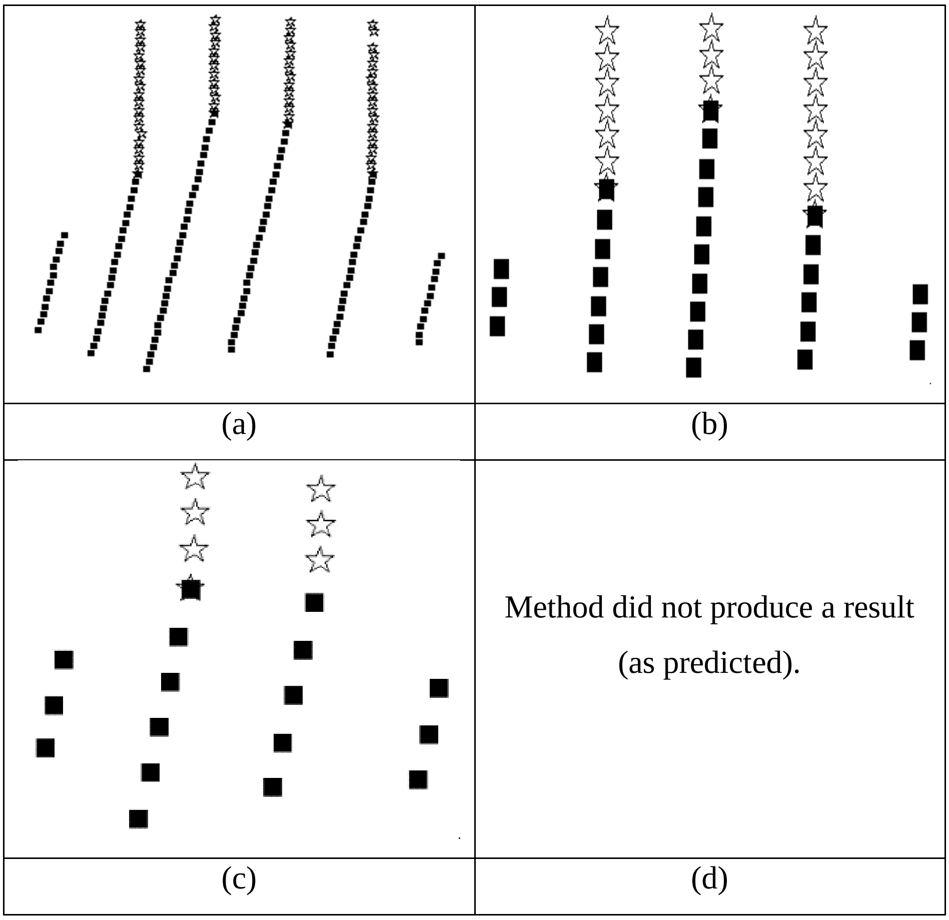
(d)

Figure 5.23 Segmentation outcomes from RDPCA (*k*=20; ϵ = 25%): (a) dataset 1; (b) dataset 2, (c) dataset 3, (d) dataset 4; Adopted from Nguyen et al. (Submited and under revision-a)

The main reasons for the failures of RDPCA in these experiments were caused by the limitation of KNN, the local saliency feature estimation, and the use of KNN for automatically determine ED_{th} and OD_{th} values (as discussed in Chapter 2).

5.3.2.3 Evaluation of the Cabo segmentation method

In this experiment, all of the required parameters are used as suggested in Cabo et al. (2015). For instance, the Douglas Peucker tolerance is set to 5 cm, the angular parameter for checking the parallelism is set to 2 degrees, and the distance threshold between end nodes is set to 70 cm. In addition, only scan profiles that have more than 2 points are kept (see section 2.3.5). As can be seen from the results in Figure 5.24, the 3D version of the Douglas Peucker algorithm shows good performance in segmenting scan profiles from scanlines in the first three datasets, as they are free of outliers. The exception is that some of the points near the boundary of two scan profiles of the same scanline are assigned as belonging to both scan profiles. However, it completely failed when applied to the noisy dataset as the 3D version of the Douglas Peucker algorithm has no resistance against outliers (see section 2.3.5 for more details). After the scan profiles are segmented, Cabo could not detect adjacent planar surfaces that have a similar orientation to the ground. This is because the scan profiles of these planes are parallel to each other, as well as being neighbours with each other. Consequently, for datasets 1 to 3, Cabo can only detect plane 1 and labels plane 2 and 3 as a single plane. The output from the segmentation outputs of Cabo method on the four datasets are illustrated in Figure 5.24.



(a)

(b)

(c)

Method did not produce a result (as predicted).

(d)

Figure 5.24. Segmentation outcomes from Cabo method: (a) dataset 1; (b) dataset 2, (c) dataset 3, (d) dataset 4; Adopted from Nguyen et al. (Submited and under revision-a)

5.3.3 Comparing the results with respect to number of correct and incorrect segmented points

Each extracted point in the *slave* point cloud contributes an equation to the least-squares model (section 4.1.2.3). As a result, each point that is assigned to its correct surface will introduce a correct equation into the registration adjustment. In contrast, each point that is assigned to an incorrect surface will contribute an incorrect equation to the least-squares model. In addition, any point that is not assigned to its surface will reduce the number of correct equations of the model and therefore it will reduce the redundancy. Therefore, after performing the five discussed methods, the quality of the detected planes are assessed based on: (1) the number of points properly labelled to their surfaces (correctly segmented-CS); (2) the number of points which are incorrectly labelled to one of the three surfaces (incorrectly segmented-IS); and (3) the number of points that are not labelled to any of the three surfaces (mis segmented-MS). Table 5.9 shows the number of CS, IS and MS points of different plane detection and segmentation techniques using the best input parameters for each dataset, as determined in the results presented in the previous section.

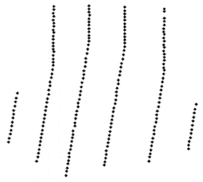
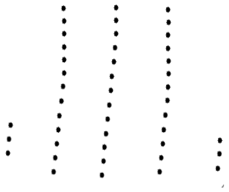
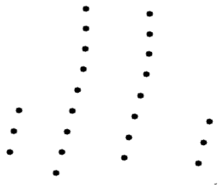
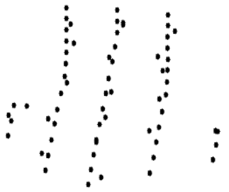
	Methods	CS	MS	IS (point)
Dataset 1 (199 points) 	RANSAC-NV	117 (58.7%)	82 (41.3%)	0
	RANSAC	195 (98.0%)	4 (2.0%)	0
	RDPCA	88 (44.2%)	111 (55.8%)	0
	Cabo	135 (67.8%)	64 (32.2%)	66
	PSPS	199 (100%)	0 (0.0%)	0
Dataset 2 (45 points) 	RANSAC-NV	22 (48.9%)	23 (51.1%)	5
	RANSAC	42 (93.3%)	3 (6.7%)	2
	RDPCA	31 (68.8%)	14 (31.2%)	12
	Cabo	26 (57.7%)	19 (42.3%)	22
	PSPS	45 (100%)	0 (0.0%)	0
Dataset 3 (23 points) 	RANSAC-NV	6 (26.1%)	17 (73.9%)	1
	RANSAC	6 (26.1%)	17 (73.9%)	0
	RDPCA	9 (39.1%)	14 (60.9%)	8
	Cabo	6 (26.1%)	17 (73.9%)	17
	PSPS	23 (100%)	0 (0.0%)	0
Dataset 4 (45 points + 20 noise points) 	RANSAC-NV	20 (44.4%)	25 (55.6%)	6
	RANSAC	37 (82.2%)	8 (17.8%)	11
	RDPCA	25 (55.5%)	16 (43.5%)	31
	Cabo	x	x	x
	PSPS	45 (100%)	0 (0.0%)	0

Table 5.9 Number of points of the final segmentation outcomes from different methods with datasets 1-4: CS – correctly segmented points; IS – incorrectly segmented points and MS – mis-segmented points; x – segmentation failed; Green numbers indicate accurate results and red numbers indicate inaccurate results.

As can be seen from Table 5.9, the quality of the detected planes obtained from RANSAC, RANSAC-NV and RDPCA have an inverse relation with the point density of the point clouds. The sparser the point clouds are, the lower the plane detection and segmentation quality is. Particularly, the percentage of correctly segmented points (CS) detected by RANSAC-NV is 58.7 percent for dataset 1, 48.9 percent for dataset 2, and down to 26.1 percent for dataset 3. Similar to RANSAC-NV, for RANSAC the percentage of correctly segmented points decreases from 98 percent for dataset 1 to 26.1 percent for dataset 3. With RDPCA the percentages of incorrectly segmented point increases from 18.4 percent for dataset 2 to 47.6 percent for dataset 3.

In contrast, the outputs obtained using Cabo and PSPS were not affected by the reduction of the point density of the MLS point clouds. However, Cabo could not separate plane 2 and plane 3 from each other as they have the same orientation with the ground, which is due to the fact that it only considers the parallelism between neighbouring scan profiles (see section 2.3.4).

Finally, all of the four mentioned state-of-the-art plane detection and segmentation methods are significantly affected by the presence of noises, especially applying for dataset 4. The sum of the points of CS, IS and MS is smaller than the number of points in dataset 4 (65 points), because the dataset contains introduced noise points (20 points). Noise points should not be detected as the planar points. The Cabo method could not detect any plane of the target. Meanwhile, PSPS is robust against the presence of the noise in this case, and achieve in all dataset a CS value of 100%.

5.3.4 Comparing the results with respect to plane parameters

As mentioned in section 4.1.2.3, besides points assigned to the planes, the parameters of the planes are also used as inputs for the LSPFA. The four parameters of a plane are the three elements of its normal vector and its distance to the origin. The number of equations that is affected by the plane parameters depends on the number of points assigned to its corresponding plane in the *slave* point cloud. Depending on the number of points of the corresponding plane in the *slave* point cloud, an incorrect plane in the *master* point cloud or *master* model can introduce a few of incorrect equations to thousands of incorrect equations to the least-squares model. Thus, the accuracy of the parameters for the planes plays a very important role in the accuracy of the final registration results. In order to evaluate the quality of the segmented planes from different techniques, the bias angles (in degree) between the normal vectors of the correct planes (i.e. benchmarking planes) with the normal vectors of the extracted planes from all of the discussed methods are compared. The parameters of the benchmarking planes, as well as the detected parameters for the planes from different methods are derived by applying PCA on the extracted groups of points representing different planes of the target. After the plane detection and segmentation process is performed, each group of points representing a specific plane is considered to be free of outliers. The ρ parameter (i.e. the perpendicular distance to the origin of the plane) is not taken into account for the comparisons as it is correlated with the normal vector and it has a heavy dependency with the location of the plane. In other words, the change in the normal vector leads to a change in the value of the ρ parameter, and depending on where the plane is located with regards to the origin, a small change in the orientation of the normal vector (e.g. 0.3 degree) may lead to a significant change in the value of ρ (e.g. thousands of metres) or just a minor change (e.g. few centimetres). As discussed in the previous section, RANSAC, RANSAC – NV, RDPCA and Cabo failed in detecting and segmenting planes in dataset 4, so only the segmentation results of dataset 1, 2 and 3 are compared. The difference between the angles of the detected planes and the benchmarks are shown in Table 5.10.

	RANSAC (°)	RANSAC-NV (°)	RDPCA (°)	Cabo (°)	PSPS (°)
Dataset 1 – plane 1	0.3	0.1	1.2	0.4	0
Dataset 1 – plane 2	0.2	1.1	1.5	x	0
Dataset 1 – plane 3	0.2	3.6	3.3	x	0
Dataset 2 – plane 1	0.1	0.3	**12.5**	2.1	0
Dataset 2 – plane 2	0.2	**5.3**	**5.6**	x	0
Dataset 2 – plane 3	0.6	2.1	2.4	x	0
Dataset 3 – plane 1	0	x	x	0	0
Dataset 3 – plane 2	x	x	x	x	0
Dataset 3 – plane 3	x	x	x	x	0

Table 5.10 Bias angles in degree from different methods (x: undetected); the bold values indicate the worse reuslts

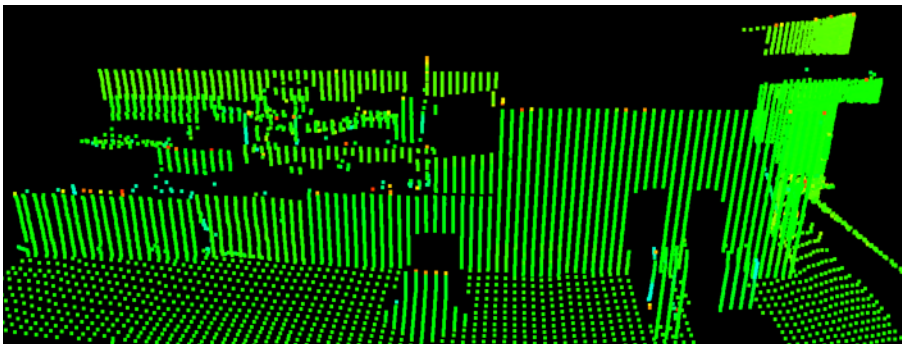
As can be seen from Table 5.10, RDPCA and RANSAC-NV provided the worst quality inputs from the *master* point clouds for the least-squares model, with the maximum difference to the benchmark being 12.5 degree and 5.3 degree respectively. Cabo offers quite good results when detecting plane 1, but it has problems with the adjacent planes that have similar orientation to the grounds. The gaps between segmentation outputs from RANSAC and the benchmarks are relatively small for the first two datasets. However, RANSAC could not distinguish plane 2 and plane 3 in dataset 3. Meanwhile PSPS provides ideal segmentation outputs for *master* inputs due to the perfect plane detection and segmentation outputs for all of the cases.

5.3.5 Conclusions for the case of the target dataset

After considering all of the quality measurement approaches including traditional approaches and the proposed approaches in the target dataset, the proposed plane detection and segmentation method shows a promising potential to be the most accurate method although the experiments are conducted with a small dataset. It can accurately detect and segment planes in the case which all of other state-of-the-art methods fail. It is also shown to be robust against noise. However, it still needs to be demonstrated to work efficiently with a representative MLS dataset.

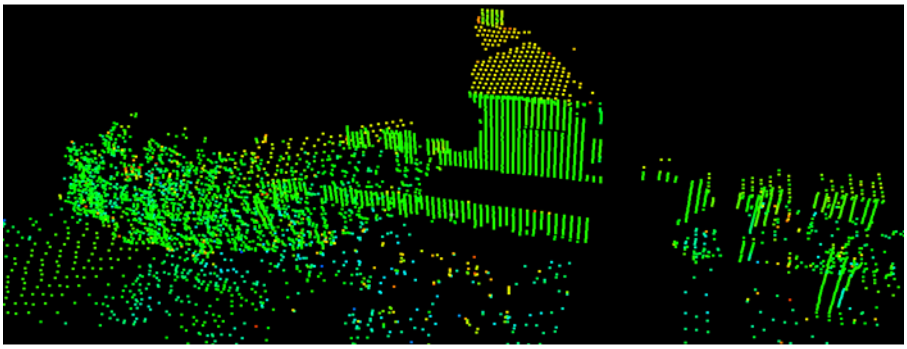
5.4 Evaluation using the road corridor dataset

The third dataset was captured along a road corridor near the Curtin University Bentley campus in Western Australia using a MDL Dynascan S250 system. This dataset has an average point spacing of 7 cm and an average profile spacing of 25 cm. This dataset can be considered to have different planar features, which exist in a normal MLS point cloud dataset. The scanning area has planes on both (the left and the right) sides of the scanner. Many of them are isolated from each other, thus they are easy to detect and to segment. However, some of them are closely aligned and more difficult to isolate, especially in regard to the sparse and heterogeneous nature of the point cloud. Therefore, fifteen planes were chosen as shown in Figure 5.26(a) and Figure 5.26(b) for evaluation and comparison. There are eleven planar features on the left hand side of the road (Figure 5.26(a)). This includes three planes of the target (plane 1, plane 2 and plane 3), a façade of a supermarket building (plane 4), a small planar object (plane 5), and the front surface of an air conditioner (plane 6) near the supermarket, two phone boxes (plane 8 and plane 9), and two façade of a building near the supermarket (plane 10 and plane 11). On the other side of the road (Figure 5.26(b)), there are 4 planes including one façade of a house (plane 14) with a window (plane 13), the roof of this house (plane 15) and a fence (plane 12). In these fifteen planes, excluding the target, there are other planes which may cause challenges for detection and segmentation. For instances, plane 4 and plane 6 are adjacent to each other and have similar normal vectors. Plane 13, plane 14 and plane 15 are similar to the circumstances presented with the simulated dataset with small offsets but having non-simulated noise level. Plane 8 and plane 9 have outliers near their top edges (Figure 5.27).



(a)

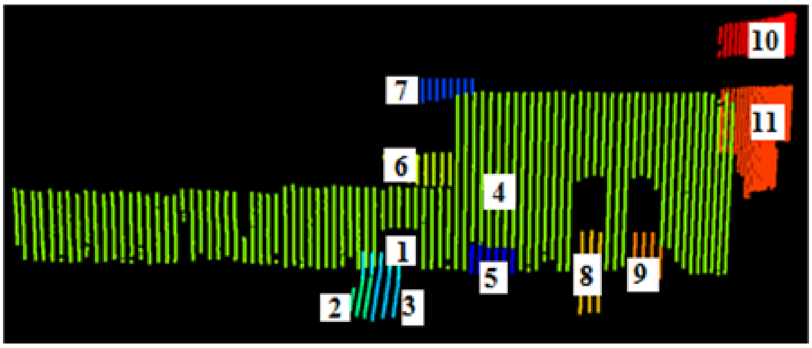
(b)



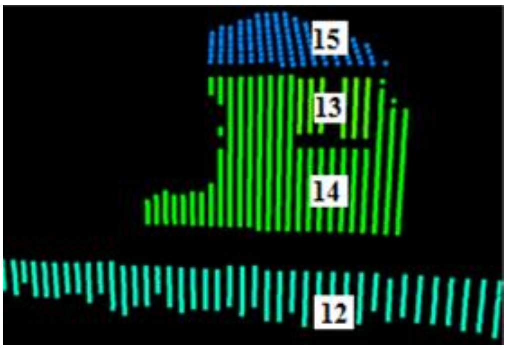
(b)

(d)

Figure 5.25 Visualisation of road corridor dataset; a) The south side; (b) Photo of the south side; c) North side of the road and (d) Photo of the north side; Adopted from Nguyen et al. (Submited and under revision-a)



(a)



(b)

Figure 5.26 Visualisations of fifteen planes in the road corridor dataset; a) the South side and b) the North side of the road; Adopted from Nguyen et al. (Submited and under revision-a)

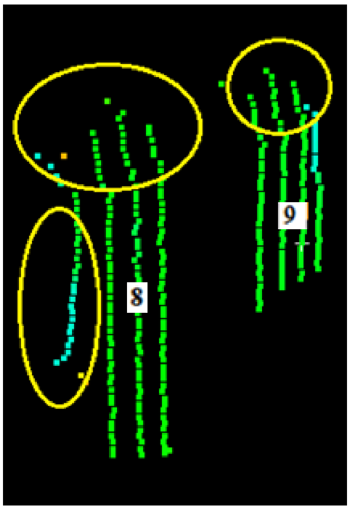
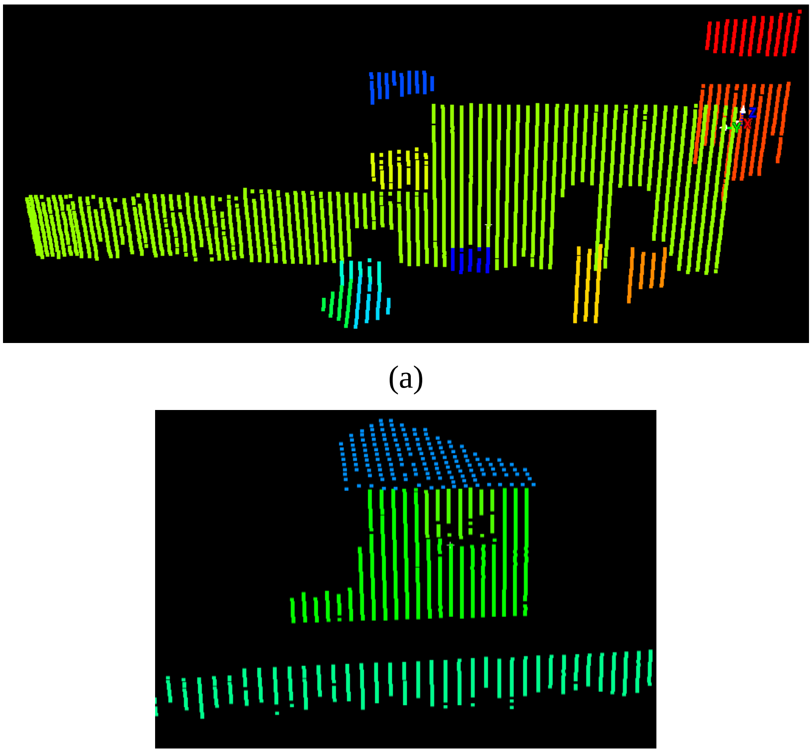


Figure 5.27 Detailed visualisation of plane 8 and plane 9; points inside the yellow ellipse are considered as outliers; Adopted from Nguyen et al. (Submited and under revision-a)

Similar to the target datasets, there is no guidance for determining the required parameters for the state-of-the-art methods for this dataset. Hence, the different parameter values that were used for the target dataset are also used for this dataset in order to find the most accurate outputs from the different methods. Meanwhile, for the proposed method, the standard parameters values setting were utilised for this dataset (Chapter 3). The aims of this section are to evaluate and compare the best outputs from different plane detection and segmentation approaches, and to discuss the issues that may occur when applying them to this dataset.

5.4.1 Evaluation of the PSPS method

For this dataset, only the final plane detection and segmentation results are discussed, i.e. after all of the segmentation steps are performed, and after all of the planes are detected (Figure 5.28). Specifically, all of the planes on the target on the left hand side are properly segmented; outliers near plane 8 and plane 9 were removed and plane 4 and plane 6 are split into two different features. The window, the façade and the roof of the house on the right hand side (i.e. plane 13, plane 14 and plane 5) are detected and segmented properly, similarly with the simulated dataset. Plane 4 and plane 6 on the left hand side are assigned into two separate regions. There are minor discrepancies between the detected planes and the ground truth. Nevertheless, they are insignificant based on visual inspections as shown in Figure 5.28.



(a)

(b)

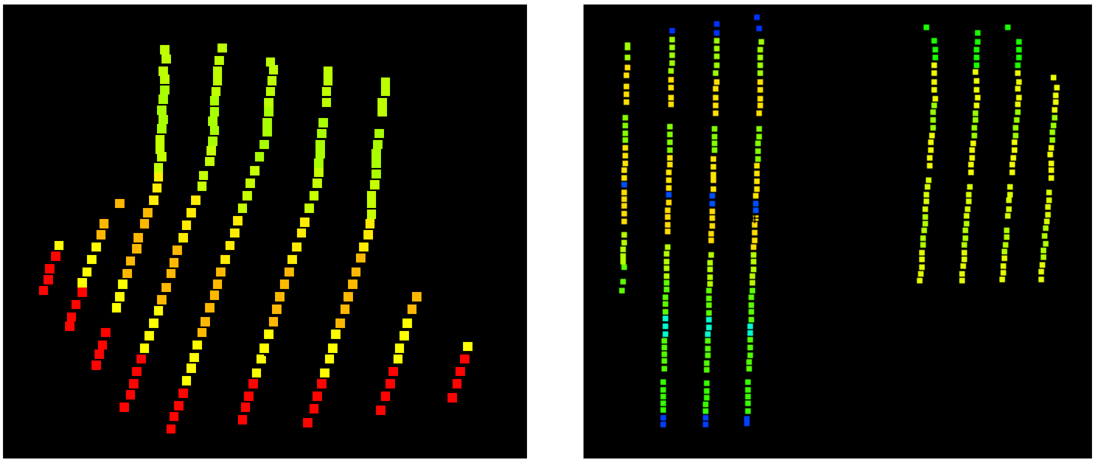
Figure 5.28 Plane detection and segmentation outputs of the road corridor dataset using PSPS; Adopted from Nguyen et al. (Submited and under revision-a)

5.4.2 Comparison with other state-of-the-art plane detection and segmentation approaches

Similar to the evaluation of the target dataset, three different criteria were used to evaluate and compare the quality of different plane segmentation method: a) the number of over-segmentation and under-segmentation features; b) the mean error of the segmented plane; and c) the plane parameters.

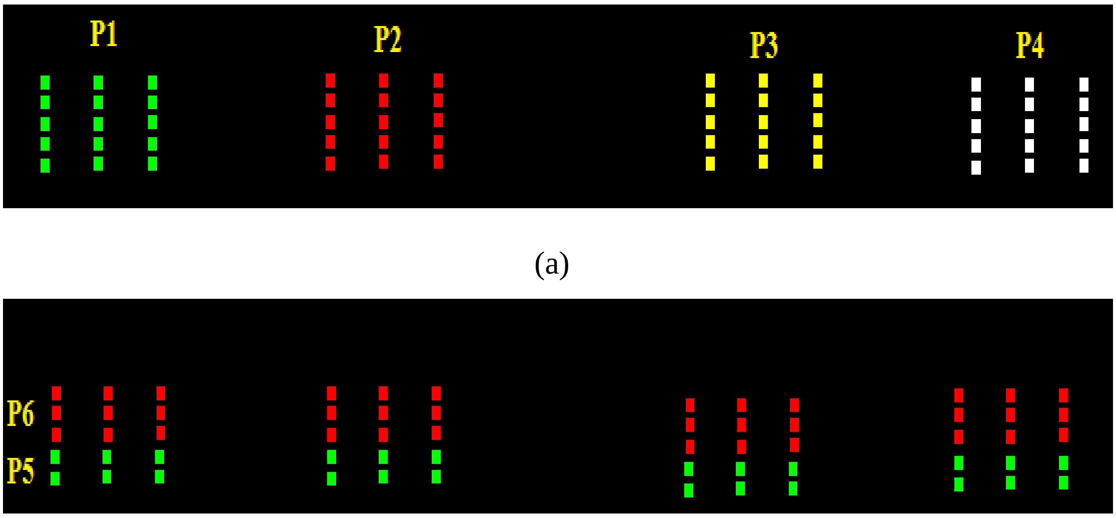
5.4.2.1 Evaluation of RANSAC and RANSAC-NV

RANSAC without considering normal vector has limitations in detecting small planes in a large point cloud dataset. As can be seen from Figure 5.29a and Figure 5.29b, the segmentation of the planes of the target and of the planes 8 and 9 were completely inaccurate. This issue is also demonstrated in Figure 5.30. As can be seen in this figure, there are four planes P1, P2, P3 and P4. Each of them is represented by a small number of points. RANSAC by its nature, may segment this dataset into 2 planes (i.e. P5 and P6) (Figure 5.30b) as they are considered to have larger consensus results than the real planes (i.e. P1, P2, P3 and P4). This problem could be solved by utilising the normal vectors or by improving the existing RANSAC algorithm.



(a) (b)

Figure 5.29 Output of RANSAC for the road corridor dataset: (a) the segmented point cloud of the target and (b) segmented point clouds of the plane 8 and 9; different colours indicate different segmented features

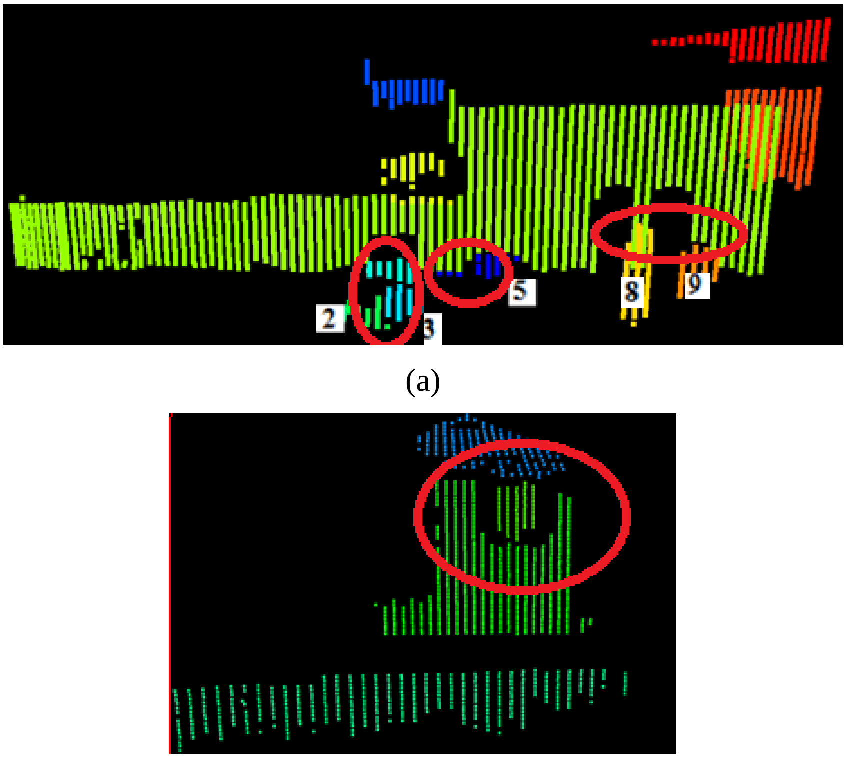


(a)

(b)

Figure 5.30 Example of limitations of RANSAC without considering the normal vectors: a) the correct outputs and b) outputs obtained by using RANSAC

The input parameters for RANSAC-NV were empirically determined from the values in Table 5.8. The distance threshold was chosen as 5 cm and the neighbourhood size *k* was set to 30 as this provided the best outputs based on visual inspections. As can be seen from Figure 5.31, RANSAC-NV detects all the planar features in the dataset. Nevertheless, there are many mis-segmented points in most of the planar objects, except for plane 12 and plane 7, as they are isolated from other objects. There are also a number of incorrectly segmented points on plane 5 and plane 8. The reason for the erroneously segmented points in this case resulted again from the issues of normal vector estimation discussed in Chapter 2.



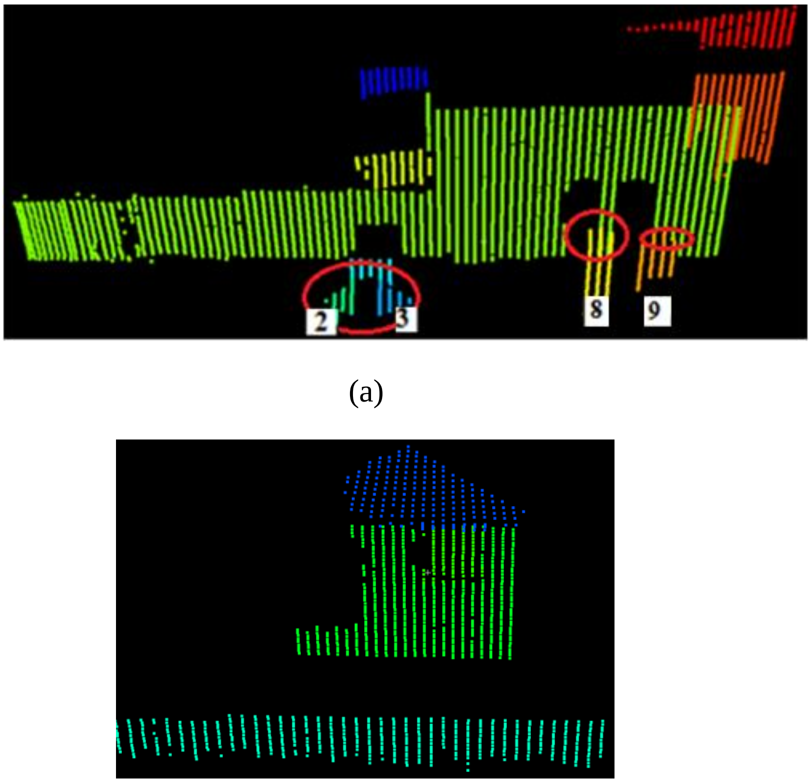
(a)

(b)

Figure 5.31 Plane detection and segmentation outputs of the road corridor dataset using RANSAC-NV; Adopted from Nguyen et al. (Submited and under revision-a)

5.4.2.2 Evaluation of the segmentation method based RDPCA

Unlike RANSAC, RDPCA does not have many mis-segmentation problems. However, it is sensitive to the parameter values in the case where there are adjacent planes that have similar normal vectors. For instance, when the number of neighbourhood of points *k* is set to 40 points and the percentage of outliers (ϵ) is set to 30 percent, RDPCA assigned plane 6 and plane 8 as a single surface. This is due to the limitation of the use of KNN for automatically determining ED_{th} and OD_{th} values as previously discussed. Nevertheless, by changing the values of these two parameters to 30 points and 25 percent, the result is that the method segments them correctly. RDPCA also has issues with points near the boundary of adjacent objects that have different normal vectors due to the limitations of KNN and the normal vector estimation technique (i.e. the RDPCA for local saliency feature estimation). Indeed, in this dataset, points near the middle of plane 2 and plane 3 of the target are not assigned to their corresponding planes.



(a)

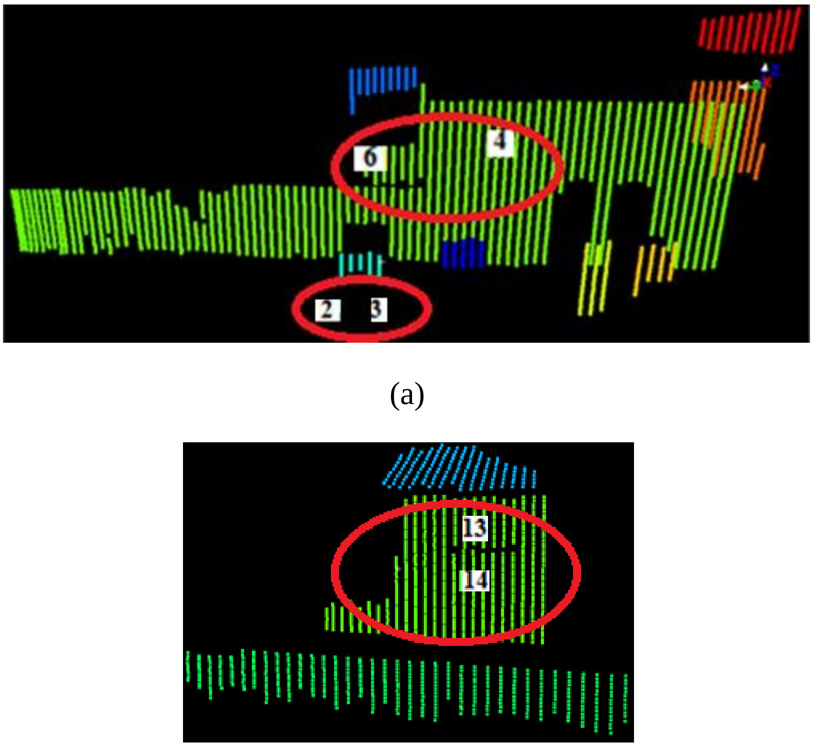
(b)

Figure 5.32 Plane detection and segmentation outputs of the road corridor dataset using RDPCA; Adopted from Nguyen et al. (Submited and under revision-a)

5.4.2.3 Evaluation of Cabo segmentation method

Again the suggested parameter values are taken from Cabo et al. (2015) with the Douglas Peucker tolerance was set to 5 cm, the angular parameter for checking the parallelism was set to 2 degrees, and the distance threshold between end nodes was set to 70 cm. These values were used to detect and segment planes in this dataset.

Similar to the target dataset, Cabo fails in detecting and segmenting planes that are adjacent and have similar orientation with the ground (i.e. plane 2 and 3, plane 4 and 6, and plane 13 and 14). The polyline gap threshold value and distance threshold between end-nodes parameter values needs to be set carefully in order to prevent Cabo labelling plane 10 and 11 as the same feature. Meanwhile, the other planes are accurately detected as they do not have similar orientations with their neighbours.



(a)

(b)

Figure 5.33 Plane detection and segmentation outputs of the road corridor dataset using Cabo; Adopted from Nguyen et al. (Submited and under revision-a)

5.4.3 Comparing the results with respect to the mean error

Unlike the target dataset, for this dataset the quality of the plane detection and segmentation results were evaluated based on the mean errors values of the detected planes instead of the number of CS, IS and MS points. It is assumed that there is a relation between the mean error values and the quality of the LSPFA. The reason for calculating the mean error values instead of other values such as RMS or mean absolute error is based on the assumption that in the least-squares model, an incorrect equation may compensate for another incorrect equation. This statement will be further investigated and demonstrated in Chapter 6. The mean errors value of each planar feature is computed using the following equation:

$$ME = \frac{\sum_{=1}^{n}(x_i n_x + y_i n_y + z_i\, n_z - \rho)}{n} \qquad \text{(eq. 19)}$$

where x_i, y_i and z_i are the coordinates of point i; n_x, n_y, n_z and ρ are the plane parameters (benchmark values) that coincide with point i.

Due to the fact that Cabo fails in detecting several planes, such as plane 2, 3, 4, 6, 13 and 14, the average mean error value from the Cabo method is not calculated and the Cabo method is excluded from this comparison. Nevertheless, the mean error values of each of the individually detected planes obtained using Cabo are still calculated. Table 5.11 shows the mean error values of different planes as well as the average values of each technique. As the mean error values are rounded to the nearest millimetre, 0 value does not mean the planes are perfectly extracted. Plane 14 seems to be difficult for all of the methods with the minimum mean error value of 15 mm. It could be resulting from the imperfect measurements of the system and the roughness of the plane. As can be seen from Table 5.11, most of the detected planes from Cabo have very low mean error values (e.g. close to 1 mm). Among the other three techniques, RANSAC seems to be the worse with the average mean error value of 9 mm, and the maximum value of 61 mm for plane 6 due to many segmentation problems. Meanwhile, PSPS still provides the highest quality plane outputs with most of the mean error values are approximate 1 mm.

	RANSAC-NV (mm)	RDPCA (mm)	Cabo (mm)	PSPS (mm)
Plane 1	1	2	1	1
Plane 2	3	2	**x**	0
Plane 3	1	6	**x**	1
Plane 4	1	0	**x**	1
Plane 5	12	1	0	1
Plane 6	**61**	15	**x**	6
Plane 7	**18**	2	15	1
Plane 8	7	11	0	0
Plane 9	1	13	0	1
Plane 10	0	0	0	0
Plane 11	3	1	0	0
Plane 12	0	0	0	0
Plane 13	2	0	**x**	3
Plane 14	17	**35**	**x**	15
Plane 15	7	3	2	1
Average	9	6	**x**	2

Table 5.11 Mean errors of planes detected and segmented by different approaches compared to the benchmark in millimetres; the highlighted number indicates the low (green) and high (red) value of the mean errors; x indicates the undetected plane

5.4.4 Comparing the results with respect to the plane parameters

The results of the bias angle values between the detected planes and the benchmarks are shown in Table 5.12. Similar to previous section, the average bias angle value from Cabo is not calculated as it could not detect the planes 2, 3, 4, 6, 12 and 13. As can be seen from Table 5.12, RANSAC is still the worse approach to obtaining inputs from the *master* point cloud, with the average bias angle of 1.61 degree compared to the benchmark. Furthermore, the maximum bias angle is larger than 7 degrees (i.e. 7.32 degrees for plane 14). RDPCA provides slightly higher quality results with the average bias angle of around 1 degree and the maximum bias angle of approximately 4 degrees. The plane detection and segmentation outputs obtained from PSPS shows the best inputs for the *master* point cloud for the MLS registration in term of the plane parameters all of the discussed approaches. The average bias angle is less than 0.5 degrees and the maximum value is less than 2 degrees.

	RANSAC-NV (°)	RDPCA (°)	Cabo (°)	PSPS (°)
Plane 1	0.619	0.235	0.580	0.387
Plane 2	0.797	2.352	x	0.218
Plane 3	0.159	0.450	x	0.428
Plane 4	0.021	0.085	x	0.020
Plane 5	1.624	0.380	0.633	0.282
Plane 6	3.168	2.390	x	1.814
Plane 7	3.676	3.561	3.972	0.834
Plane 8	1.077	2.843	1.852	1.746
Plane 9	3.255	1.411	2.598	0.279
Plane 10	0.179	0.027	0.007	0.028
Plane 11	0.204	0.232	0.395	0.243
Plane 12	0.841	0.344	0.072	0.205
Plane 13	0.296	0.134	x	0.057
Plane 14	7.326	0.397	x	0.038
Plane 15	0.978	0.878	0.253	0.354
Average	1.614	1.047	x	0.462

Table 5.12 Bias angles of different planes detected and segmented by different approaches compared to the benchmarks in degrees; x indicates undetected plane

5.4.5 Conclusions of the road corridor dataset

For the dataset that have different planar features that exist in a normal MLS point cloud dataset, the proposed plane detection and segmentation method (PSPS) still provides the most accurate segmentation outputs. Based on the evaluation results, the segmentation outputs of the proposed method are very similar to those extracted manually. It was also proven that the standard setting parameters for PSPS can be applied for the MLS point cloud datasets.

5.5 Summary

In this chapter, the proposed plane detection and segmentation method and other state-of-the-art methods applied to sparse and heterogeneous MLS point clouds dataset are evaluated and compared using three different datasets. Three new evaluation criteria are proposed in order to evaluate the quality of the segmentation outputs with respect to the inputs to the LSPFA process. The results prove the correctness of the discussions about the limitations of the state-of-the-art methods in detecting and segmenting planes in sparse and heterogeneous MLS point clouds. The outputs also showed that the quality of the segmentation outputs becomes lower when the point density of the MLS point clouds gets sparser after a certain level. Furthermore, the state-of-the-art methods for plane detection and segmentation are sensitive to their inputs parameters, especially in the case of sparse point density and do not robust against noise.

The evaluation also showed that among all of the approaches, outputs from the proposed method provided the highest quality inputs for MLS point cloud registration and for other purposes. Furthermore, the proposed approach is not sensitive to the parameter values as they can be either automatically determined or fixed due to carefully analysis of the points and scan profiles near the boundary between planar segments. Although the novel introduced method is only evaluated in detecting and segmenting planes in sparse and heterogeneous MLS point clouds, it is believed that it can be also applied to dense MLS point clouds.

CHAPTER 6 EVALUATION OF MLS POINT CLOUD REGISTRATION

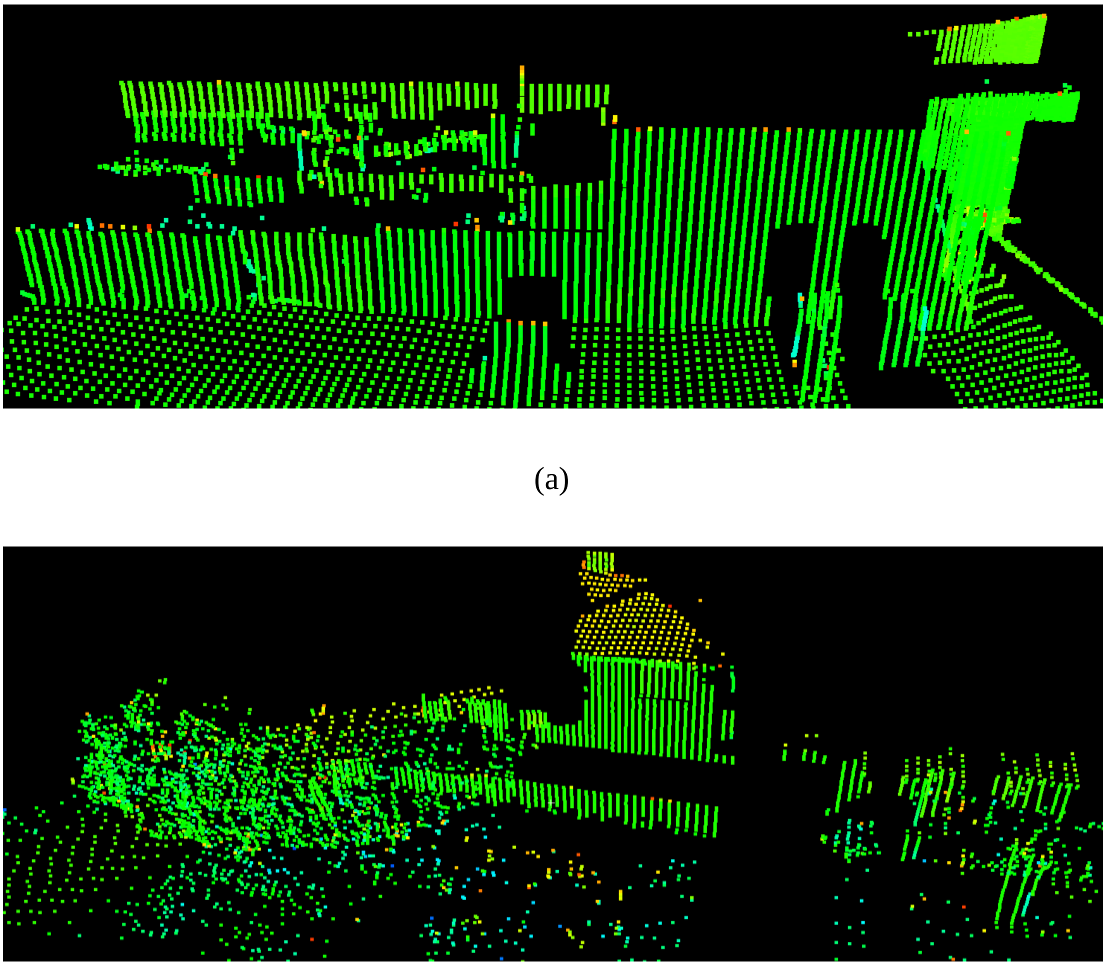
This chapter firstly introduces the test set up (i.e. the datasets used and the error metrics applied), that will be used in the registration evaluation process. Next, it evaluates which registration technique is the most suitable approach for sparse and heterogeneous MLS point clouds. For this evaluation the newly proposed error metric introduced in Chapter 4 is applied. Finally, the dependencies from the inputs obtained by the different plane detection and segmentation techniques are examined in regards to the quality of the most suitable registration method.

6.1 Test set up

6.1.1 Datasets

Two MLS point cloud were used as experimental datasets in this chapter. Dataset 1 is the point cloud dataset used in the previous Chapter. Meanwhile, dataset 2 was captured at the same road corridor, but in the opposite direction at a similar speed. As a result, while the scan profile spacing distances of the two point cloud datasets are similar, their point spacing distances are different. Specifically, the south side of the road of the point cloud dataset 1 has a smaller average point spacing than the south side of the point cloud dataset 2 (e.g. 5.5 cm and 8 cm respectively), while the north side of the road of the point cloud dataset 1 has a larger point spacing (e.g. 8 cm and 5.5 cm respectively).

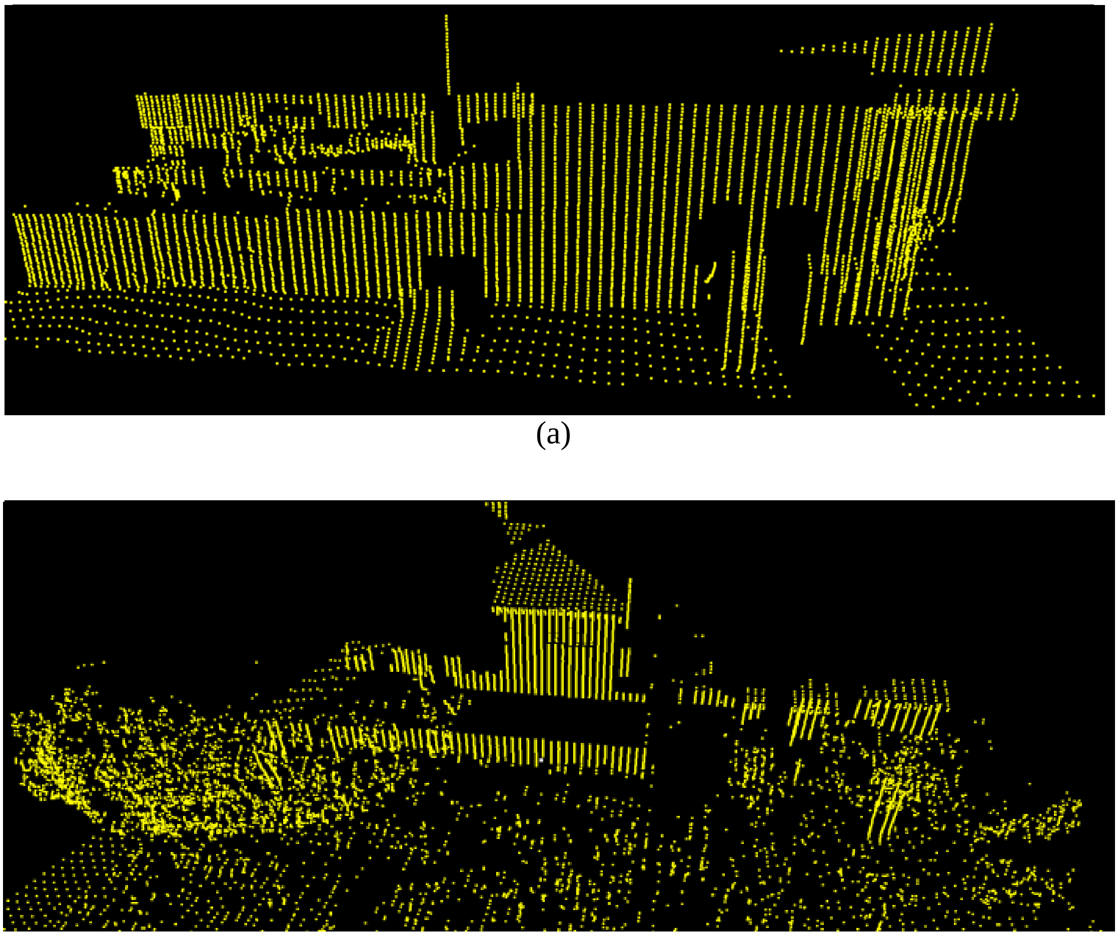
Originally, all planar objects in the two captured point clouds are parallel with the trajectory of the scanning vehicle. Consequently, there are no triplets of planes existing in this dataset. However, these triplets are required for the success of the LSPFA process, as previously discussed in Chapter 4. Hence, a target which is formed by three planes, as shown in section 5.2, was placed in the scanning area. These three planes are used in order to meet the minimum requirement for the success of the plane-based registration process, as well as to enhance the diversity in angular orientation of the planes for the evaluation process.



(a)

(b)

Figure 6.1 Point cloud dataset 1: North (a) and south (b) side of the road; Adopted from Nguyen et al. (Submited and under revision-b)

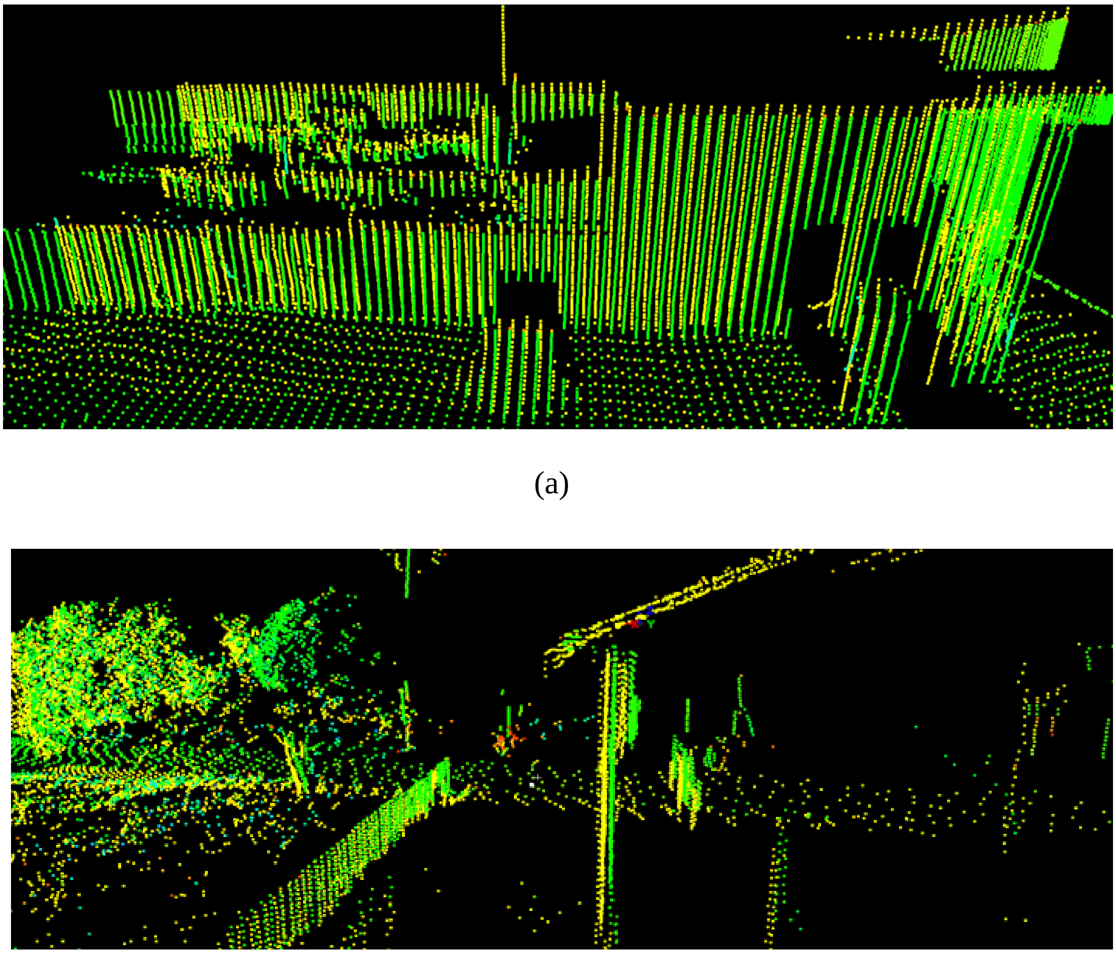


(a)

(b)

Figure 6.2 Point cloud dataset 2: North (a) and south (b) side of the road; Adopted from Nguyen et al. (Submited and under revision-b)

The visualisation of the two datasets are shown in Figure 6.1 and Figure 6.2. The discrepancies between the two datasets are illustrated in Figure 6.3. As discussed in Chapter 4, the point-based hybrid technique suffers from the limitations of both the point-to-point and the point-to-plane matching approaches. According to the authors (Chan et al., 2016; Yang et al., 2016) the semantic virtual feature point matching (SVFPM) techniques are only suitable for coarse registration, while the model-to-model matching techniques have limitations in the number of redundant observations and the inconsistency in the plane parameters calculated from the point clouds as previously discussed. Hence, the SVFPM, model-to-model and the point-based hybrid techniques are not utilised for this comparison. The point-to-point approach (section 4.1.1.1) was performed using Cloud Compare – an open-source software (Cloud Compare, 2017). Cloud-Compared allows users to perform different processing tasks with 3D point clouds data, such as visualisation, registration and segmentation. Meanwhile, the point-to-plane method (section 4.1.1.2) and LSPFA method (section 4.1.2.3) were implemented in C++ using Eigen library (Eigen, 2017) for matrices and vector computations and ANN library (ANN, 2016) for implementing KD tree data structure and local neighbourhood selection such as KNN and FDN.



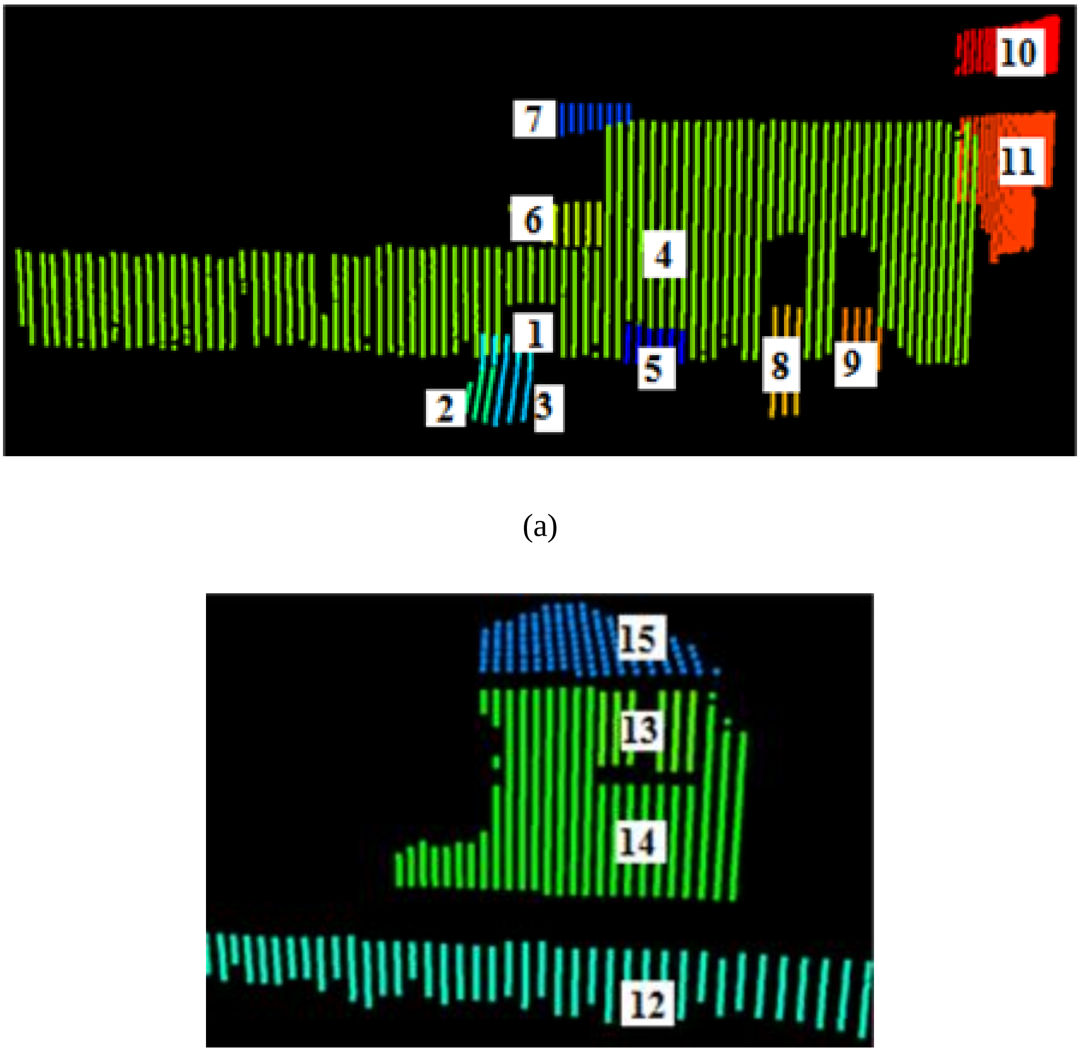
(a)

(b)

Figure 6.3 Mis-alignments between data set 1 and 2; a) North side, b) South side; Adopted from Nguyen et al. (Submited and under revision-b)

6.1.2 Error metrics for comparison and evaluation

As discussed in the section 4.2.3, the RMS value of the residuals for the distances between corresponding point pairs should not be used to evaluate the accuracy of the MLS registration process, instead the RMS values of the residuals of the points fitted to plane should be used instead. In theory, the check planes should be independent from the minimisation equations. Nevertheless, there are a limited number of planes with different orientations that exist in these datasets (e.g. there is only one planes triplet). Moreover, parallel planes suffer similar effects if there are errors in the registration. Hence, the fifteen planes mentioned in the scene (Chapter 5) are extracted manually and are used as check planes. As the check planes have different orientations and are located in different areas in the point clouds, the evaluation can still be considered to be reliable. Afterwards, PCA is used to calculate the plane parameters of the extracted planes in the *master* point cloud. Next, planes are assigned into five groups based on the angular difference between them to compute the RMS values after the registration process. Specifically, each of the groups from 1 to 4 consists of only one plane, namely 2, 3, 5 and 15 respectively. While group 5 consists all of the remaining planes, as they are all normally orthogonal to the ground surface and parallel with each other. The details of each group are shown in Table 6.1. Finally, the *RMS_index* value is calculated using equation 18 in Chapter 4. By performing the calculation on the benchmark, the *RMS_index* value before performing the registration is determined to be 82 mm.



(a)

(b)

Figure 6.4 Visualisations of fifteen planar surfaces manually extracted from dataset 1.

Group ID	Plane(s)
Group 1	2
Group 2	3
Group 3	5
Group 4	15
Group 5	1, 4 and 6 to 14

Table 6.1 Planes are divided into different groups based on their normal vectors.

6.2 Point-based registration vs plane-based registration experiments

Before performing the registration, both of the point clouds' origins were translated to the centre of the test site in order to compensate for the effects of rotation when the datasets are far away from the origin (section 4.1.3).

6.2.1 Evaluation of different registration strategies using the proposed error metric

6.2.1.1 Evaluation using the point-based matching method: Point-to-point

As the registration output of the point-to-point approaches are influenced by the choice of which point cloud acts as the *master* (Grant et al., 2013), both backward (i.e. registration output using point-to-point to register dataset 2 to dataset 1) and forward (i.e. registration output using point-to-point to register dataset 1 to dataset 2) registrations are performed. Grant et al. (2013) claimed that ICP provides the most accurate output when the *slave* point cloud is a subset of the *master* point cloud. This is often not the case in MLS point cloud registration. Table 6.2 illustrates the estimated transformation parameters of both the backward and forward registrations using point-to-point matching technique. Due to the fact that both of the MLS point cloud datasets were translated to be close to the origin, as well as being roughly align to each other, both of the estimated rotation and the translation parameters were found to be relatively small as expected.

As can be seen from Table 6.3, the output of the backward registration significantly differs from the output of forward registration. For instance, using forward registration the RMS value of plane 1 (e.g. 47 mm) is much smaller than using forward registration (e.g. 134 mm), while using backward registration the RMS value of plane 2 (e.g. 80 mm) is larger than using forward registration (e.g. 66 mm). Overall, using the proposed error metric, the forward registration output is considered the representative output of the point-to-point method as it has a low final RMS index value (e.g. 53 mm). Nevertheless, this RMS value demonstrates that there is a large discrepancy between the two point cloud datasets, and the results from the two different methods. The registration needs to be improved based on the evaluation of the results.

	Rotations (°)			Translations (m)		
	ω	ϕ	κ	T_x	T_y	T_z
Dataset 1 to dataset 2	-0.159	-0.566	-0.018	-0.054	-0.098	-0.274
Dataset 2 to dataset 1	-0.160	-0.561	0.097	-0.083	-0.319	-0.297

Table 6.2 Estimated transformation parameters from point-to-point approach

	$\text{RMS}_{\text{registration}}$ before registration (mm)	Point-to-point forward after registration (mm)	Point-to-point backward after registration (mm)
Group 1	155	47	134
Group 2	54	80	66
Group 3	105	83	51
Group 4	30	38	33
Group 5	65	15	7
RMS_index	82	53	58

Table 6.3 RMS values of the points fitted onto their models before and after registration using point to point approach

6.2.1.2 Evaluation of the point-based matching method: Point-to-plane

The point-to-plane matching approach proposed by Grant et al. (2012), which uses the three closest points to defined the plane, is not suitable for MLS point cloud registration. This is due to the fact that the three nearest points of a point in a sparse MLS point cloud normally belong to the same scanline, hence these three points should not be used to represent the corresponding tangent plane. Hence, the point to plane approach proposed by Takai et al. (2013) is considered as the representative of the point-to-plane approaches. This approach uses planes formed by *k* number of local neighbouring points instead of three nearest points. By performing a point-to-plane registration process (section 4.1.1.2), the *RMS_index* value was reduced from

82 mm to 22 mm. Details of the RMS values of each group are described in Table 6.5. The estimated rigid transformation parameters are given in Table 6.4.

Rotations (°)			Translations (m)		
ω	ϕ	κ	T_x	T_y	T_z
-0.122	-0.545	0.119	-0.086	-0.262	-0.020

Table 6.4 Estimated transformation parameters of point to plane

	$\text{RMS}_{\text{registration}}$ before registration (mm)	Point-to-Plane $\text{RMS}_{\text{registration}}$ after registration (mm)
Group 1	155	33
Group 2	54	3
Group 3	105	51
Group 4	30	16
Group 5	65	5
RMS_index	82	22

Table 6.5 RMS values of the points fitted onto their models before and after least-squares adjustment process using point-to-plane; Adopted from Nguyen et al. (Submited and under revision-b)

Although the RMS value using point-to-plane approach is much smaller than the RMS value using point-to-point (i.e. 22 mm compared to 53 mm), the discrepancy between two point clouds datasets is still quite large and needs to be improved.

6.2.1.3 Evaluation using the LSPFA matching method

The rigid transformation parameters were calculated by performing the LSPFA process that fits point of dataset 2 onto their corresponding planar surfaces in dataset 1 (section 4.2.1.3). After performing the registration process, the calculated rigid transformation parameters values (eq. 10) are determined and shown in Table 6.6. As the registration requirements are met, the average value of RMS value was significantly reduced from 82 mm to 3 mm after performing the LSPFA. Details of the RMS values for each group are described in Table 6.7.

Rotations (°)			Translations (m)		
ω	ϕ	κ	T_x	T_y	T_z
-0.190	-0.449	-0.004	-0.087	-0.256	-0.263

Table 6.6 Estimated transformation parameters of the benchmarks

	RMS values before fitting (mm)	RMS values after fitting (mm)
Group 1	155	0
Group 2	54	1
Group 3	105	1
Group 4	30	9
Group 5	65	4
RMS_index	82	3

Table 6.7 RMS values of the points fitted onto their models before and after least-squares adjustment process; Adopted from Nguyen et al. (Submited and under revision-b)

As can be seen from Table 6.7, the final RMS value after the LSPFA registration is only 3mm, which is much better than the RMS values of the point-based matching techniques, which were 53 mm and 22 mm. This value also shows that the registration was accurately performed.

6.2.1.4 The importance of meeting the minimum requirement for plane-based matching

Originally, the planar objects in the scanning area are not sufficient to meet the minimum requirement for the success of the plane-based registration process (section 4.1.2.4). To evaluate this behaviour, an experiment using only a subset of the points of the original dataset (i.e. without the designed target being placed into the scans) is performed by applying LSPFA.

After LSPFA is performed, significant horizontal mis-alignments are visible from the registration results (Figure 6.5). Meanwhile, the registration output using the target (section 6.2.1.3 with RMS value of 3 mm) seems to be the most accurate output (Figure 6.6). For the following discussion, the estimated transformation parameter values from that registration are used as the benchmark. It is easy to see from Table 6.8 that the estimated parameter values are very close to the benchmark values, except the estimated value for T_y, with a difference of 197 mm. This is the reason of the huge horizontal mis-alignment after registration in this experiment that causes the large error in the point cloud registration.

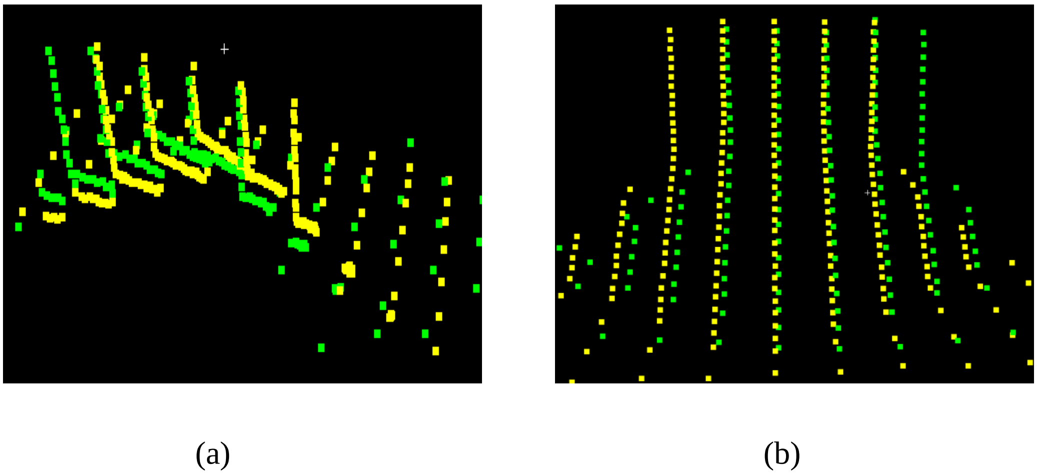


Figure 6.5 Visualisation of registration without the target results: (a) Top-back view of the target after registration; (b) Front view of the target after registration; Adopted from Nguyen et al. (Submited and under revision-b)

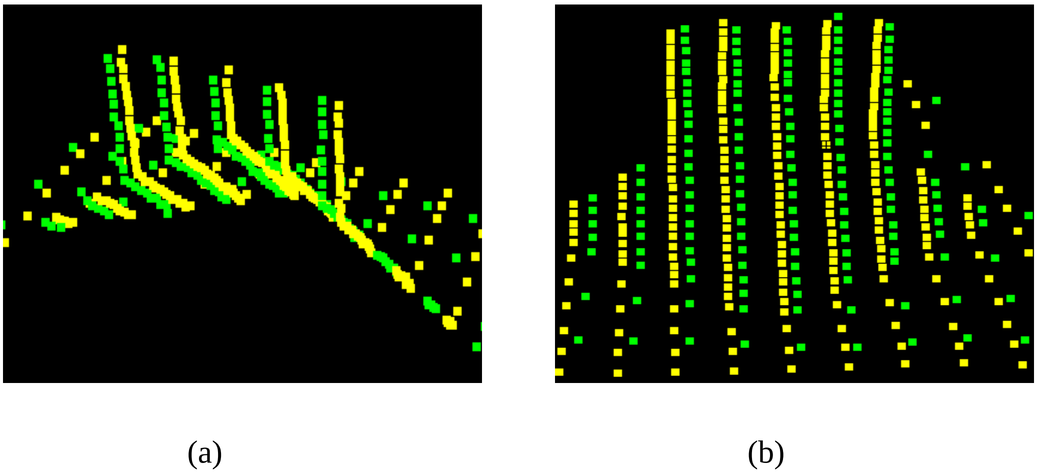


Figure 6.6 Visualisation of registration with the target results: (a) Top-back view of the target after registration; (b) Front view of the target after registration; Adopted from Nguyen et al. (Submited and under revision-b)

	ω (°)	ϕ (°)	κ (°)	T_x (m)	T_y (m)	T_z (m)
Estimated parameter	-0.197	-0.452	-0.003	-0.093	-0.453	-0.268
Differences with the benchmark	-0.007	-0.003	-0.001	-0.006	-0.197	-0.005

Table 6.8 Estimated transformation parameters after registration without considering the target's planes and the differences with the benchmark; Adopted from Nguyen et al. (Submited and under revision-b)

Despite of the large error in the translation along the Y-axis, the calculated RMS values of group 3, 4 and 5 are unchanged (Table 6.9). This is as expected as planes in the groups 3, 4 and 5 have the direction vectors parallel with the Y-axis (see section 4.2.3). Meanwhile, the RMS values of group 1 and 2 are significantly increased as the direction vectors of planes in the groups 1 and 2 are not parallel with the translation error direction.

	Group 1	Group 2	Group 3	Group 4	Group 5	Average
RMS value (mm)	88	76	1	9	4	36
Difference with BM (mm)	88	75	0	0	0	33

Table 6.9 RMS values of the points fitted onto their models after registration using LSPFA without the target and the differences with the benchmark; Adopted from Nguyen et al. (Submited and under revision-b)

6.2.1.5 Demonstrations of the limitations of existing error metrics for the evaluation process

The basis of this discussion is the outputs that were obtained in the previous experiments. Firstly, if the discrepancies between the estimated parameters values and the benchmark are utilised as the error metric to measure the quality of the registration process, it will be very difficult to draw a conclusion on the question of which approach is better. Indeed, looking at the transformation errors from point-to-point and point-to-plane methods in Table 6.10, Figure 6.7 and Figure 6.8, it can be seen that the point-to-point method has the largest sum of errors regarding the rotation parameters. In contrast, LSPFA without the target has the smallest error value. The sum of errors in translation of the point-to-plane method is the smallest error value for the rotation parameters (e.g. 0.187 degrees). In theory, different sets of transformation parameters transform the *slave* point cloud to different locations. Two different sets of transformation parameters can be considered to have the same quality if the magnitudes of their errors are similar. Nevertheless, it is impossible to evaluate the magnitudes of errors of these MLS registration outputs based on the discrepancies between the estimated parameters values and the benchmark. Consequently, further investigation needs to be implemented in order to find which registration technique is best in this case.

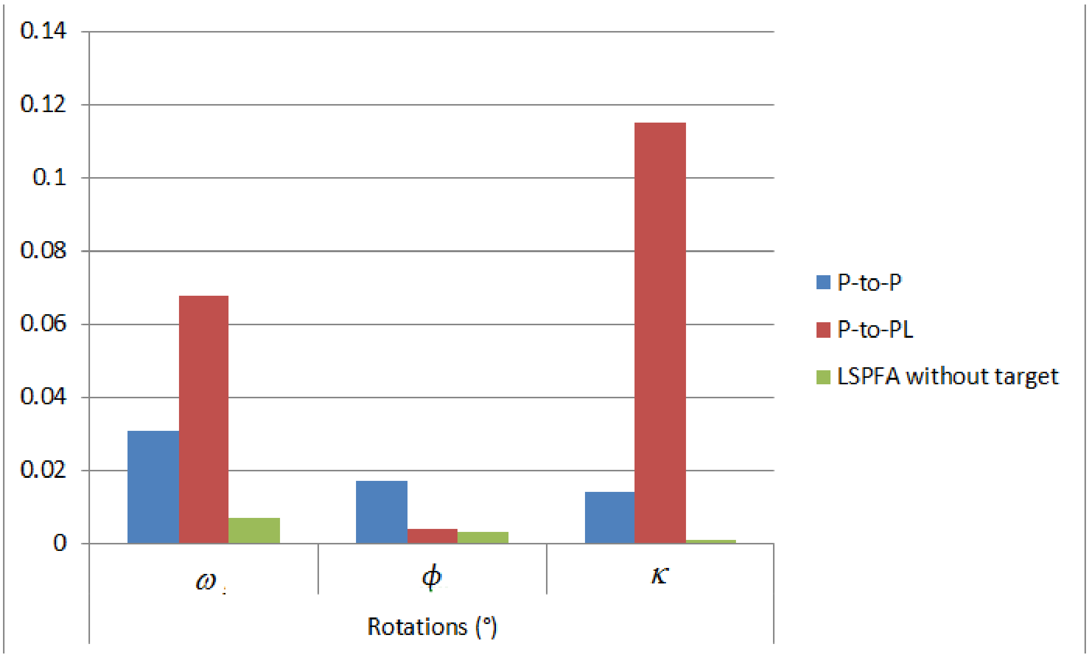


Figure 6.7 Rotation errors from point-to-point (P-to-P), point-to-plane (P-to-PL) and plane-based LSPFA method without target compared to the benchmark; Adopted from Nguyen et al. (Submited and under revision-b)

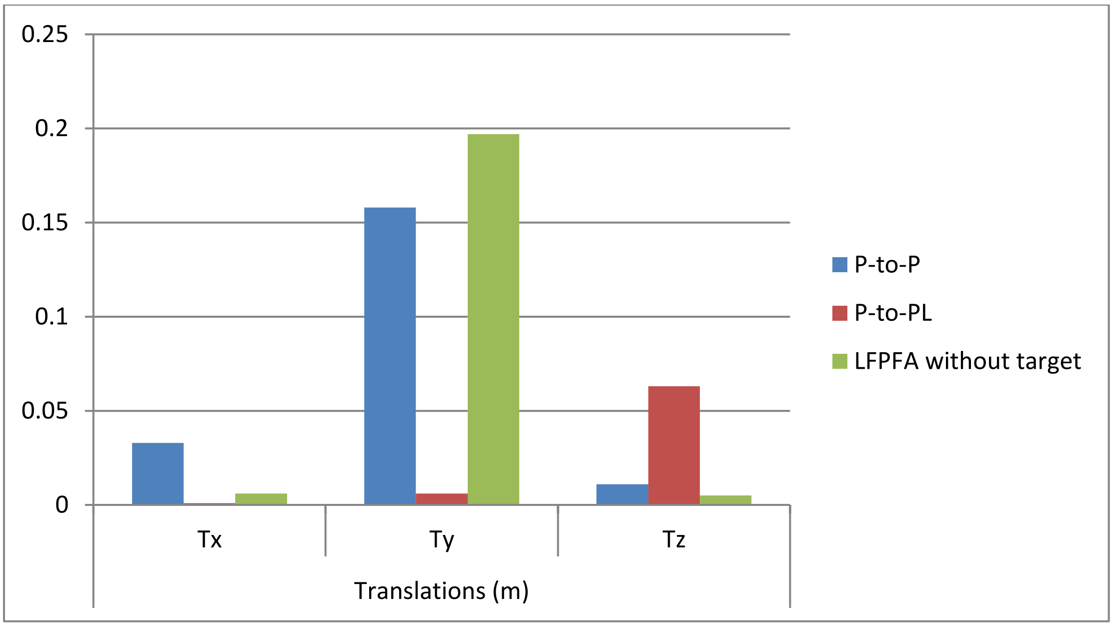
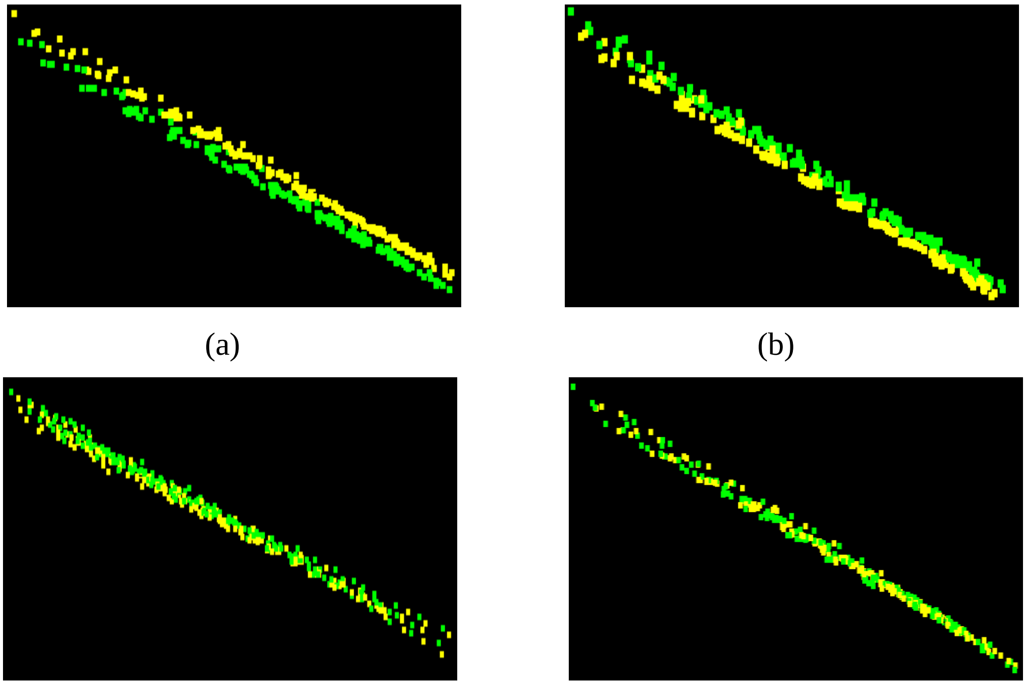


Figure 6.8 Translation errors from point-to-point (P-to-P), point-to-plane (P-to-PL) and plane-based LSPFA method without target compared to the benchmark; Adopted from Nguyen et al. (Submited and under revision-b)

	Rotations (°)			Translations (m)		
	ω	ϕ	κ	T_x	T_y	T_z
Point-to-Point	-0.031	-0.017	-0.014	-0.033	-0.158	0.011
Point-to-Plane	-0.068	-0.004	-0.115	-0.001	0.006	-0.063
Plane-based LSPFA method without target	-0.007	-0.003	-0.001	-0.006	-0.197	-0.005

Table 6.10 Transformation errors from point-to-point, point-to-plane and plane-based without target and the benchmarks; Adopted from Nguyen et al. (Submited and under revision-b)

The point clouds of the planes 3 and 15 after the registration process are visualised in Figure 6.9 and Figure 6.10 respectively for visual inspections. From these two figures, it is clear to see that, LSPFA outperform the other two approaches. Furthermore, the point-to-plane matching method is better than point-to-point matching method. However, it is very difficult to assess the results when the plane-based LSPFA method without the target, the point-to-point matching, and the point-to-plane matching are compared. As can be seen from the Figure 6.9, it can be said that LSPFA without the target provided the best out of the three results. However, from Figure 6.10, it can be concluded that LSPFA is the worse in the case where the minimum requirements are not meet. Again, it is a challenge to draw a conclusion in this case.



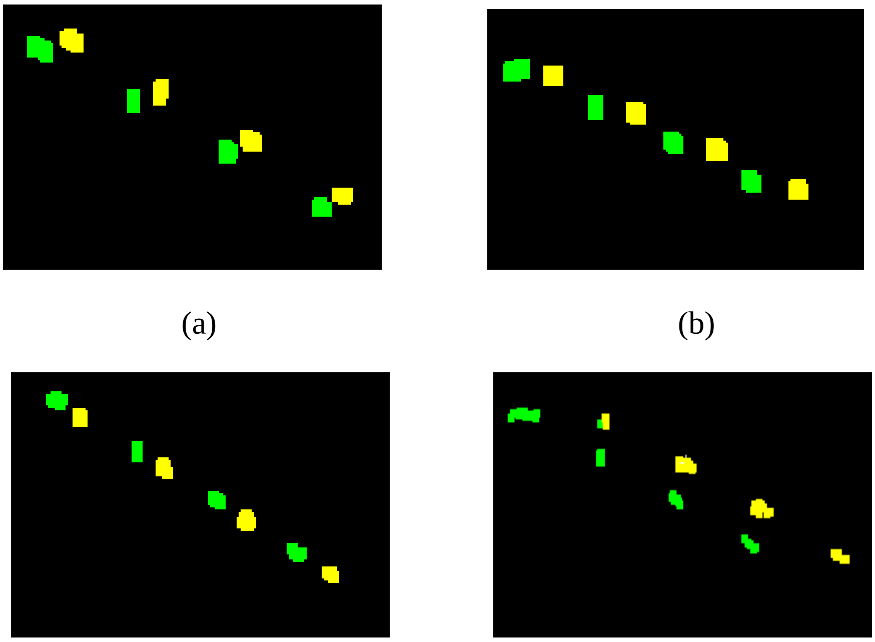
(a)

(b)

(c)

(d)

Figure 6.9 Visualisation of the registration results of plane 15: a) point-to-point; b) point-to-plane; c) least-squares plane fitting and d) plane-based LSPFA method without target; different colours indicate the points from the two input point clouds (i.e. *master* and *slave*) ; Adopted from Nguyen et al. (Submited and under revision-b)



(a) (b)

(c) (d)

Figure 6.10 Visualisation of the registration results of plane 3: a) point-to-point; b) point-to-plane; c) plane-based LSPFA method and d) plane-based LSPFA method without target; different colours indicate the points from the two input point clouds (i.e. *master* and *slave*) ; Adopted from Nguyen et al. (Submited and under revision-b)

Finally, if the check planes are randomly chosen without considering their orientations and they come from group 3, 4 or 5, the *RMS_index* of the plane-based LSPFA method without target will be exactly the same as the plane-based LSPFA method with the target included. This is despite having significant error in the translation parameter along the Y axis (Table 6.9). This means that the translation error may not be detected. Consequently, the quality of the registration output may be over-evaluated, and not reliable in assessing the success of the registration.

6.2.2 Conclusions about the registration outputs using different matching technique

The existing error metrics have limitations in evaluating and comparing the quality, as discussed. Meanwhile, the proposed error metric provides a reliable approach to compare the registration results from different techniques. As can be seen from Table 6.11, LSPFA outperforms both of the point-to-point and point-to-plane approaches as it has the lowest final *RMS_index* (e.g. 3mm). In order to obtain accurate registration results, point cloud datasets need to be translated so that the origin is close to the central mass of points as well as the minimum requirement for LSPFA discussed in section 5.2 must be fulfilled. Otherwise, large error will occur in the registration outputs. Nevertheless, this requirement can be easily satisfied in most of the cases.

	Point-to-point (mm)	Point-to-plane (mm)	LSPFA (mm)	LSPFA without target (mm)
RMS_index	53	22	3	36

Table 6.11 Comparisons of the registration outcomes from different approaches; Adopted from Nguyen et al. (Submited and under revision-b)

6.3 Comparative study of the quality of the plane-based LSPFA method using inputs from different plane segmentation approaches

Theoretically, the inputs for the LSPFA approach are different groups of points representing the planar features in the *slave* point cloud and the plane parameters in the *master* point cloud. In order to perform the registration and the evaluation process automatically, planar features need to be automatically extracted and segmented from the point clouds. The quality of the inputs may affect the final result of the registration process, as well as the evaluation process. Therefore, in this section, a comparative study of automatic plane fitting registration for sparse and heterogeneous MLS point clouds with different plane extraction and segmentation approaches is conducted. The LSPFA output from the previous section is used as the benchmark for comparison as it was shown that it achieved the most accurate results after compared to the manually segmented planes in that experiment.

6.3.1 Overview

The inputs for the least-squares plane fitting adjustment process are obtained from both of the *slave* and the *master* point clouds or dataset. The inputs from the *slave* point cloud are the 3D coordinates of points in different groups representing the corresponding planes in the *master* point cloud. The inputs from the *master* point cloud are the parameters of the planes that have correspondences in the *slave* point cloud. In section 5.4.3 and 5.4.4, two different quality indexes of the segmented planes with regards to the inputs for LSPFA were introduced, and PSPS was shown to be the most suitable techniques based on the results of the experiments. However, these statements are just an assumption and the goal of this section is to validate this. Furthermore, this section also aims to investigate the magnitude of the dependencies between the quality of the inputs and the registration outputs.

Hence, two different experiments are conducted for the two different registration scenarios (section 2.4). In the first experiment, the captured MLS point cloud dataset is required to be registered onto a *master* model. Unfortunately, there is no data from a higher accuracy dataset available for this scanning area. Therefore, dataset 1 is considered as the *master* dataset and dataset 2 is assigned as the *slave* dataset. Hence, planar features in dataset 1 were manually extracted in order to obtain the best segmentation outputs (i.e. to be used as the model). Then, PCA was used to compute the plane parameters. The plane detection and segmentations outputs obtained using different techniques were used as the inputs extracted from the *slave*. In the second case, inputs for both *slave* and *model* point clouds are automatically extracted using all of the discussed plane detection and segmentation methods.

In chapter 5, only RANSAC-NV, RDPCA and PSPS could successfully detect all of the planar features, especially with respect to the three planar target surfaces. In other words, their outputs for the captured dataset satisfy the minimum requirement for the success of the LSPFA method. Meanwhile, Cabo method failed in detect two planes of the target, which will lead to the failure of the LSPFA (section 5.3.5). Therefore, the Cabo method will not be taken into account for the comparison in this chapter.

As the two captured datasets in this case are already roughly aligned, all the initial values for the 6 unknown parameters are set to zero.

6.3.2 Registration of the *slave* point cloud to the *master* model

As presented in chapter 4, it was assumed that there is a relation between the mean errors of the detected planes (i.e. inputs) and the quality of the registration outputs, particularly the final error metric value. This assumption is based on the fact that LSPFA aims to find the six rigid transformation parameters that minimise the sum of the square of the distances between points to their corresponding planar surfaces in the *master* model. In this chapter, the average of the mean error value of each group of planes is shown instead of mean error value of the individual plane. This is because the planes in the same group suffer the same impact with respect to the errors in transformation. Details of mean error values of different groups of planes are shown in Table 6.12.

	RANSAC-NV (mm)	RDPCA (mm)	PSPS (mm)
Group 1	3	2	0
Group 2	1	5	1
Group 3	7	5	1
Group 4	12	1	1
Group 5	10	8	3
Average	6	4	1

Table 6.12 Mean error values of each group based on three different segmentation methods

As shown in Table 6.12, the mean error values of all planar surfaces obtained using PSPS are much smaller than those values of planar surfaces extracted using RANSAC-NV and RDPCA. For instance, the mean error of group 4 of PSPS is only 1 mm, compared with 12 mm of RANSAC-NV. The mean error values of individual group constitute to an average mean error value for PSPS, which is six times smaller than RANSAC-NV and 4 times smaller than RDPCA. Furthermore, the number of redundant observations exists in PSPS, which is also higher than in RANSAC-NV and RDPCA (see Chapter 5 for more details).

	RANSAC-NV (mm)	RDPCA (mm)	PSPS (mm)
Group 1	4	1	0
Group 2	1	2	1
Group 3	1	3	1
Group 4	11	12	9
Group 5	4	4	4
RMS_index	4	4	3

Table 6.13 The RMS values after registration using different inputs obtained using different plane detection and segmentation techniques.

The registration results using the PSPS output is slightly better than using RANSAC-NV and RDPCA (Table 6.13). The correlation coefficient value between the mean error values and the RMS values was calculated. The result shows that they are highly positive correlated to each other with the correlation coefficient value is 0.945. Figure 6.11 shows the correlation between the mean errors and the RMS values from different approaches.

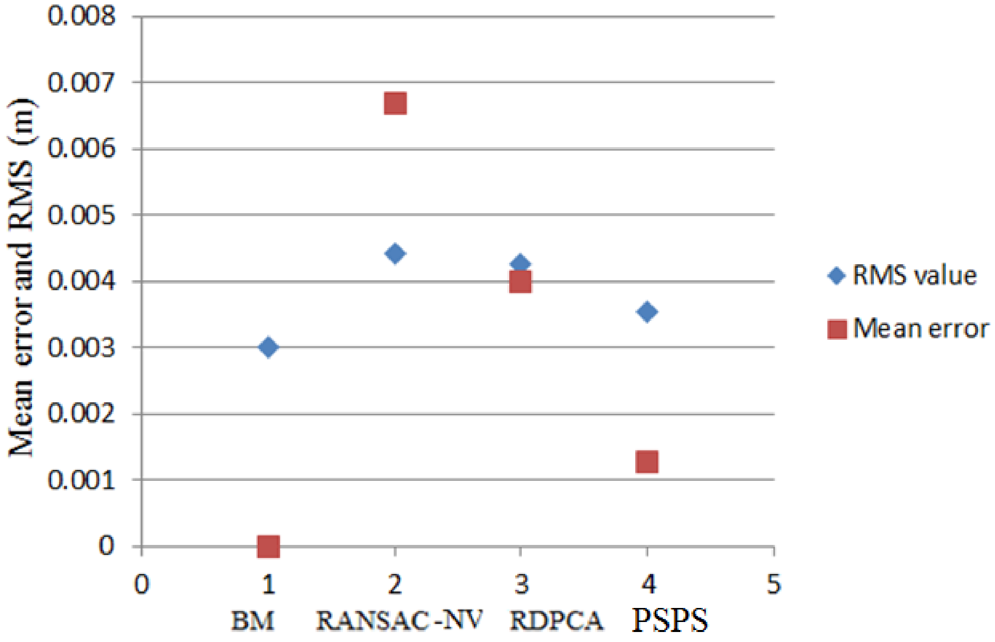


Figure 6.11 Mean errors and RMS values from the segmentations results created using different methods.

In conclusion, the final average RMS values of the plane-based registration process using inputs from the discussed methods are not significantly different. However, PSPS still provides the highest quality data for the registration of the *slave* point cloud onto the *master* model when the LSPFA process is used for the estimation of the transformation parameters values.

6.3.3 Registration of the *slave* point cloud to the *master* point cloud

In many real applications, the captured point clouds from different runs are required to be registered to each other. In this case, one of the MLS captured point clouds is assigned as the *master* and the other point clouds is assigned as the *slave*. Hence, in this part of the experiment, the parameters of the *master* planes are extracted from the point clouds using three of the discussed techniques and used in the registration process.

As discussed in Chapter 4, each inaccurate set of plane parameters contributes a number of imprecise equations to the least-squares model, where this number is equal to the number of points extracted from the *slave* point cloud to the plane. It is believed that there is a relation between quality of the extracted plane and the quality of the MLS registration results. The differences between the plane parameters (i.e. normal vectors and distances to the origin) of the automatically extracted planes of the *master* and the benchmark were also calculated.

As can be seen from Table 6.14, the differences between the planar surfaces from RANSAC-NV and the benchmark is the worse than compared with the other methods. Particularly, the average values in angle and distance to the origin are 1.088 degree and 0.24 m respectively. Meanwhile, those values of PSPS are relatively small (e.g. less than 0.436s degree and 0.049 m).

	RANSAC-NV (°)	RDPCA (°)	PSPS (°)
Group 1	0.798	0.135	0.218
Group 2	0.159	0.235	0.429
Group 3	0.979	0.215	0.355
Group 4	**1.624**	0.380	0.282
Group 5	**1.879**	**1.079**	0.130
Average	**1.088**	**1.024**	**0.283**

Table 6.14 Discrepancies between the angles of the extracted planes and the benchmark when applying different segmentation methods as the pre-step for the LSPFA process

	RANSAC-NV (m)	RDPCA (m)	PSPS (m)
Group 1	0.056	0.015	0.029
Group 2	0.035	0.047	0.089
Group 3	**0.791**	0.054	0.281
Group 4	**0.212**	**0.247**	0.022
Group 5	0.106	0.022	0.004
Average	**0.240**	**0.216**	**0.049**

Table 6.15 Differences between distances to the origin of the surfaces extracted by the selected methods and the benchmark

After the calculation of the average RMS values of the final registration result using all the techniques, the values are insignificantly different compared to the previous registration case (Table 6.16). However, the RMS values of each individual group has changed (Table 6.17). For example, with RANSAC-NV the RMS value of group 1 changes from 4 mm to 1 mm; the RMS value of group 3 increases from 1 mm to 3 mm. Similarly, with RDPCA, the RMS value of group 1 increases from 1 mm to 3m and the RSM value of group 4 decreases from 12 mm to 10 mm. The reason behind these results may be due to the fact that the equations in the LSPFA model are correlated to each other. The LSPFA process aims to minimise the sum of squared distances from points to their surfaces in the *master* point clouds, thus there may be a case that incorrect equations may compensate or mask the effect of other incorrect equations. This may lead to the final average RMS values not changing significantly as has occurred in this case. However, these circumstances occur arbitrarily and cannot be controlled. In other words, in this case the final values of the error metric was unchanged, but it could not be ensure that the same thing happens for all other cases.

	RANSAC (mm)	RDPCA (mm)	PSPS (mm)
Group 1	1	3	1
Group 2	2	2	1
Group 3	3	2	2
Group 4	11	10	8
Group 5	5	5	5
RMS_index	4	4	3

Table 6.16 RMS values after registration using three different approaches

	RANSAC (mm)	RDPCA (mm)	PSPS (mm)
Group 1	-3	2	1
Group 2	1	2	0
Group 3	2	-1	1
Group 4	0	-2	-1
Group 5	-1	1	1
Total	-1	0	2

Table 6.17 Differences between RMS values after the registration using the previous experimental results compared to three different techniques.

Therefore, in order to further explore the impacts between the correctness of the plane parameters and the RMS value of the registration process, another experiment was conducted, in which the inputs for the model were taken from the benchmark, except the planes in group 4. In other words, only planes in group 4 were detected and segmented automatically by the three different plane detection techniques, and the parameters for the planes in the other groups were taken from the benchmark values. As can be seen from Table 6.14, the discrepancies between the extracted normal vectors for the planes and the benchmarking values were 1.624 (RANSAC-NV), 0.380 (RDPCA) and 0.0282 (PSPS) degrees. And the differences between the extracted distances of the planes to the origin with respect to the benchmarking values were 0.212 (RANSAC-NV), 0.247 (RDPCA) and 0.022 (PSPS). The RMS values from each of the techniques after the registration process were performed are presented in Table 6.18. Again, PSPS provided the most accurate registration result among the three techniques with the average RMS value and individual RMS values being similar to the case of matching the *slave* point cloud to a *master* model experiment (section 6.3.2). Meanwhile, due to the large differences between the extracted planes in group 4 from RANSAC-NV, the RMS values of registration outputs increased significantly, especially with group 1 and group 4. It can be seen that a very high correlation of 0.992 between the planes quality and the quality of the registration outputs.

	RANSAC-NV (mm)	RDPCA (mm)	PSPS (mm)
Group 1	7	1	1
Group 2	2	3	1
Group 3	1	3	1
Group 4	15	12	9
Group 5	4	4	4
RMS_index	6	4	3

Table 6.18 RMS values after the registration using three different approaches for group 4, features were automatically extracted

	RANSAC (mm)	RDPCA (mm)	PSPS (mm)
Group 1	3	0	1
Group 2	1	1	0
Group 3	0	0	0
Group 4	4	0	0
Group 5	0	0	0
Total	8	1	1

Table 6.19 Differences between RMS values after the registration using three different techniques with previous experiment result (section 6.3.2)

1.3.4 Conclusions of the quality of the LSPFA method using inputs from different plane detection and segmentation approaches

Although the difference between the *RMS_index* of the registration outputs using inputs from different plane segmentation approaches was not significant, the results from the experiments show that PSPS provided the highest quality inputs to the LSPFA process. The accuracy of the registration process is highly dependent on the mean errors of the extracted planar features from the *slave* point cloud and the plane parameters from the *master* point cloud. In addition, the PSPS has been proven to be not as sensitive to the parameters as other methods are. Hence, it is highly suitable for getting the inputs for the LSPFA process in term of inputs from *slave* point cloud and inputs from *master* point cloud. The new error metric once again has been proven to be more suitable for evaluating and comparing the quality of the registration outputs than the traditional error metrics.

6.4 Summary

In this chapter, an error metric to evaluate and compare the quality of the MLS registration process using different matching techniques was evaluated. Experiments were conducted on the real MLS datasets to evaluate and compare the registration outputs obtained using point-based matching and feature based matching, specifically for the LSPFA method. The results from the experiments confirmed that the LSPFA approach outperforms the point-based matching (e.g. point-to-point and point-to-plane) techniques for registration of sparse MLS point clouds. The limitations of the existing error metrics are demonstrated and discussed. Finally, the new error metric was proven to be suitable in evaluating and comparing the MLS point cloud registration outputs when compared to other error metrics.

Next, a comparative study of MLS registration outputs obtained using different plane detection and segmentation techniques was conducted. Two different experiments were performed based on two different practical scenarios. PSPS was proven to be the most suitable method to provide inputs for LSPFA.

CHAPTER 7 CONCLUSIONS AND OUTLOOKS

This research has shown that sparse and heterogeneous point cloud data is difficult to process. Previous research has focused on the engineering solutions (e.g. increasing the scan rate or capturing data at slow velocity in order to increase the point density) rather than processing techniques to overcome this problem. This research focused on the processing technique to estimate the local saliency feature and extract planar features from the sparse and heterogeneous point cloud data in order to utilise the data captured from low-end laser scanner(s) at normal velocity. The new proposed method has proven that it can accurately detect and segment planes from the sparse and heterogeneous point cloud data. The outputs of the proposed method can be used as the inputs for other further processing steps, such as modelling and MLS point cloud registration. As a result, in many applications (e.g. 3D city mapping), the low-end scanner(s) can be used to scan at the normal road velocity without worrying about the quality of the processing steps.

Furthermore, the propose error metric of this research offers a reliable approach to qualify the MLS point cloud registration process, especially, in the case of sparse and heterogeneous MLS point cloud registration. It can overcome the limitations of the current error metrics for evaluation of the quality of the MLS point cloud registration process. The following are the details of the achievements of this Thesis.

7.1 Achievements

The point density of a captured point cloud has been addressed by many researchers. Nevertheless, there is limited research focused on the sparseness and heterogeneity of MLS point clouds captured using low-end MLS systems up to this point. The properties of the MLS point clouds lead to difficulties for state-of-the-art methods in different point clouds processing steps, such as local saliency features estimation, plane detection and segmentation. For instance, the KNN and FDN are not suitable for local neighbourhood selection since different features at different place in the captured MLS point clouds may have different point densities. It may lead to different issues for normal vector estimation process such as the percentage of outliers may be larger than the tolerance of the state-of-the-art outlier remover methods. Consequently, in many cases the outputs of the normal vector estimation are not accurate, leading to inaccuracy in the plane detection and segmentation as well as the MLS point cloud registration.

Plane detection and segmentation is a crucial process in point cloud processing such as modelling, plane-based matching for registration of MLS point clouds and plane-based calibration. Inspired from the fact that points on the same scan profile belong to the same planar feature, a novel method that utilises the scan profiles patterns and the planarity values between neighbouring scan profiles to detect and segment planar features from the MLS point clouds was proposed. In order to segment different scan profiles from the captured MLS point clouds, a new local saliency feature, namely the direction vector of points that is exclusively applicable for MLS point clouds datasets was introduced. This new local saliency feature was proven to be suitable for MLS point clouds datasets as points in MLS point clouds can be considered as partly ordered (i.e. points are captured in sequences). Then, a new local neighbourhood selection method and the modified RANSAC (mRANSAC) algorithm, which utilises the normal vector information, were proposed. Both methods aim to accurately estimate the direction vectors of points in a MLS point cloud dataset. The process of estimating direction vectors of points was shown to be more reliable than estimating the traditional local saliency features (e.g. normal vectors) as it only has to deal with 2D data (along a scanline) rather than 3D data. Afterwards, a scan profile detection algorithm was proposed to segment scan profiles from the MLS point clouds.

The proposed plane detection and segmentation method was evaluated using three different MLS point cloud datasets including a simulated dataset and two real datasets. The segmentation outputs from the proposed method were compared with three of the most recent state-of-the-art methods using different comparison criteria. The evaluation criteria included traditional comparison measures (e.g. the number of over-segmented and under-segmented regions), as well as the newly introduced measure that focuses on the quality of the plane detection and segmentation as the inputs for the LSPFA matching. Those measures included the number of correctly segmented points, incorrectly segmented points and mis-segmented points, the mean errors of the extracted planes, and the discrepancies between the detected planes and the benchmark. The evaluation results prove that the proposed plane detection and segmentation method outperforms the other methods when applying them to sparse and heterogeneous MLS point cloud datasets. It especially succeeded in the detecting and segmenting planar features in the presence of limitations due to the way the data was captured in which other methods fails, e.g. segments with a similar normal vector direction and small offsets. However, these limitations are often present in scans of urban corridors, e.g. a door and window element of a façade. The new method was shown to be also robust against noise, whose presences are again not uncommon for MLS point clouds. Furthermore, by carefully analysing the primitives (e.g. points and scan profiles) near the boundary of different higher primitives (e.g. scan profiles and planes) the required parameter values in the new proposed method have small impacts on the outputs and they can be either fixed or automatically determined based on the specifications of the scanner(s).

Different error metrics were proposed to evaluate and compare the registration quality from different registration methods. However, they all have their own limitations which leads them to not to be suitable for evaluating the quality of the MLS registration outputs. Furthermore, when comparing the quality of the registration outputs using different techniques, they can be influenced by the technique used, and the distribution of the surfaces through the scene. Hence, a novel error metric that utilised the RMS values of points fitted onto planes residuals, in combination with taking into account the orientation of the check planes after registration was presented. By grouping the RMS values based on their orientations, the quality of the registration can be assessed with equal weights given to the possible orientations present in the scene.

Different point cloud matching techniques were reviewed and discussed. The suitability of these matching techniques to the MLS sparse point cloud registration was analysed. The effects of rotation and translation parameters regarding to the convergent of the least-squares model were also analysed. The quality of the registration results of the MLS sparse point clouds obtained using different state-of-the-art matching techniques were also compared and presented using the proposed error metric. The results showed that the least-squares plane fitting adjustment approach (LSPFA) outperforms other matching approaches in performing MLS sparse point cloud registration.

In order to perform the MLS point cloud registration process automatically, planes in the point clouds need to be detected and segmented automatically from the captured point clouds. The plane detection and segmentation outputs obtained using different approaches may be dissimilar due to different segmentation criteria. The segmented planes are used as inputs for LSPFA. They can be categorised into two groups: 1) the extracted points representing the corresponding planes in the *slave* point cloud and 2) the plane parameters of the corresponding planes in the *master* point cloud. In order to investigate the influences of the quality of different inputs obtained using different plane detection and segmentation approaches as well as to determine the most suitable inputs for LSPFA, experiments were conducted using real datasets. The results show that the quality of the MLS registration outputs were related to the mean errors of the extracted planes and correctness of the plane parameters (i.e. the discrepancies of the plane parameters between the extracted planes and the benchmark). The newly proposed plane detection and segmentation approach has been demonstrated to be the most suitable approach in combination with the LSPFA matching for the registration process.

7.2 Outlook

The new local saliency feature, namely the direction vector works very well to segment scan profiles from scanlines in MLS point cloud datasets. However, it seems unsuitable for detecting scan profiles of other geometric features such as the cylinders or octagonal lamp poles for point clouds captured using type 2 scanner(s). Moreover, the proposed segmentation method is only suitable to detect and segment planar objects in MLS point cloud datasets. There is also a need to detect and segment other geometric primitives in urban areas such as spheres, octagon lamp poles, lamp poles, cylinders and other complex features.

The proposed plane detection and segmentation method could also be extended to ALS (e.g. roofs detection) and TLS point clouds (e.g. building façade detection) as the outputs can be used as input for further processing. For instance, Yu et al. (2017) requires façade segments in order to derive automatically the elements of the façade applying a grammar engine.

Unlike the registration process between two MLS point clouds, in many applications the captured point clouds may be required to register onto TLS point clouds, ALS point clouds, Photogrammetric point clouds or predefined models. In this case, the gaps between MLS point clouds and other datasets may be significant. Meanwhile, the LSPFA method requires a good initial alignment between the processed datasets, which while not an issue with MLS data, might need to be addressed in furthering research.

REFERENCES

Aiger, D., Mitra, N. J., & Cohen-Or, D. (2008). 4-points congruent sets for robust pairwise surface registration. *ACM Trans. Graph., 27*(3), 1-10. http://dx.doi.org/:10.1145/1360612.1360684

ANN. (2016). ANN Library. Retrieved from https://www.cs.umd.edu/~mount/ANN/

Arun, K. S., Huang, T. S., & Blostein, S. D. (1987). Least-Squares Fitting of Two 3-D Point Sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-9*(5), 698-700. http://dx.doi.org/:10.1109/TPAMI.1987.4767965

Bae, K.-H. (2006). *Automated registration of unorganised point clouds from terrestrial laser scanners / Kwang-Ho Bae.* Thesis (Ph. D.)--Curtin University of Technology.

Bea, & Derek, D. L. (2008). A method for automated registration of unorganised point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing, 63*(1), 36 - 54. http://dx.doi.org/:https://doi.org/10.1016/j.isprsjprs.2007.05.012

Bauer, J., Karner, K., Schindler, K., K., A. , & Zach, C. (2003). *Segmentation of building models from dense 3D point clouds.* Paper presented at the In Workshop of the Austrian Association for Pattern Recognition, Laxenburg, Austria

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 14*(2), 239-256. http://dx.doi.org/:10.1109/34.121791

Boeder, V., Kersten, T. P., Thies, T., & Sauer, A. (2011). *Mobile Laser Scanning on Board Hydrographic Survey Vessels - Applications and Accuracy Investigations.* Paper presented at the FIG, Marrakech, Morocco

Borrmann, D., Elseberg, J., Lingemann, K., & Nüchter, A. (2011). The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. *3D Research, 2*(2), 3. http://dx.doi.org/:10.1007/3DRes.02(2011)3

Boulaassal, T. Landes, P. Grussenmeyer, & TarshaKurdi, F. (2007). *Automatic segmentation of building facades using terrestrial laser data.* Paper presented at the ISPRS Workshop on Laser Scanning

Cabo, C., García Cortés, S., & Ordoñez, C. (2015). Mobile Laser Scanner data for automatic surface detection based on line arrangement. *Automation in*

Construction, 58, 28-37. http://dx.doi.org/:http://dx.doi.org/10.1016/j.autcon.2015.07.005

Cahalane, C., McElhinney, C. P., Lewis, P., & McCarthy, T. (2014). Calculation of Target-Specific Point Distribution for 2D Mobile Laser Scanners. *Sensors (Basel, Switzerland), 14*(6), 9471-9488. http://dx.doi.org/:10.3390/s140609471

Cahalane, C., McElhinney, C. P., & McCarthy, T. (2010). Mobile Mapping System Performance: An initial investigation into the effect of vehicle speed on laser scan lines*Remote Sensing & Photogrammetry Society Annual Conference - "From the Sea-Bed to the Cloudtops", held in Cork, Ireland,*

Chan, T. O., & Lichti, D. D. (2012). CYLINDER-BASED SELF-CALIBRATION OF A PANORAMIC TERRESTRIAL LASER SCANNER. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XXXIX-B5*, 169-174. http://dx.doi.org/:10.5194/isprsarchives-XXXIX-B5-169-2012

Chan, T. O., Lichti, D. D., Belton, D., & Nguyen, H. L. (2016). Automatic Point Cloud Registration Using a Single Octagonal Lamp Pole. *Photogrammetric Engineering & Remote Sensing, 82*(4), 257-269. http://dx.doi.org/:http://dx.doi.org/10.14358/PERS.82.4.257

Chan, T. O., Lichti, D. D., & Glennie, C. L. (2013). Multi-feature based boresight self-calibration of a terrestrial mobile mapping system. *ISPRS Journal of Photogrammetry and Remote Sensing, 82*(0), 112-124. http://dx.doi.org/:http://dx.doi.org/10.1016/j.isprsjprs.2013.04.005

Chen, Y., & Medioni, G. (1991, 9-11 Apr 1991). *Object modeling by registration of multiple range images.* Paper presented at the Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on. http://dx.doi.org:10.1109/ROBOT.1991.132043

Choi, S., Kim, T., & Yu, W. (2009). *Performance evaluation of RANSAC family.* Paper presented at the British Machine Vision Conference, London, UK

Cloud Compare, (2017). Cloud Compare. Retrieved from http://www.danielgm.net/cc/

Demantké, J., Mallet, C., David, N., & Vallet, B. (2012). DIMENSIONALITY BASED SCALE SELECTION IN 3D LIDAR POINT CLOUDS. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XXXVIII-5/W12*, 97-102. http://dx.doi.org/:10.5194/isprsarchives-XXXVIII-5-W12-97-2011

Deschaud, J.-E., & Goulette, F. (2010). *A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing.* Paper presented at the Proceedings of the 5th International Symposium 3D Data Processing, Visualization and Transmission, Paris, France

Douglas, D. H., & Peucker, T. K. (2011). Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature *Classics in Cartography* (pp. 15-28): John Wiley & Sons, Ltd. http://dx.doi.org/:10.1002/9780470669488.ch2

Ebisch, K. (2002). A correction to the Douglas–Peucker line generalization algorithm. *Computers & Geosciences, 28*(8), 995-997. http://dx.doi.org/:http://dx.doi.org/10.1016/S0098-3004(02)00009-2

Eigen. (2017). Eigen Library. Retrieved from http://eigen.tuxfamily.org/index.php?title=Main_Page

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

Fujiwara, T., Kamegawa, T., & Gofuku, A. (2013). Evaluation of Plane Detection with {RANSAC} According to Density of 3D Point Clouds. *CoRR, abs/1312.5033*Retrieved from http://arxiv.org/abs/1312.5033

Godin, G., Rioux, M., & Baribeau, R. (1994). *Three-dimensional registration using range and intensity information*Retrieved from http://dx.doi.org/10.1117/12.189139

Grant, D., Bethel, J., & Crawford, M. (2012). Point-to-plane registration of terrestrial laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing, 72*(0), 16-26. http://dx.doi.org/:http://dx.doi.org/10.1016/j.isprsjprs.2012.05.007

Grant, D., Bethel, J., & Crawford, M. (2013). Comparative Study of Two Automatic Registration Algorithms. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-5/W2*, 91-95. http://dx.doi.org/:10.5194/isprsannals-II-5-W2-91-2013

Gressin, A., Cannelle, B., Mallet, C., & Papelard, J. P. (2012). TRAJECTORY-BASED REGISTRATION OF 3D LIDAR POINT CLOUDS ACQUIRED WITH A MOBILE MAPPING SYSTEM. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., I-3*, 117-122. http://dx.doi.org/:10.5194/isprsannals-I-3-117-2012

Gressin, A., Mallet, C., Demantké, J., & David, N. (2013). Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS Journal of Photogrammetry and Remote Sensing, 79*(0), 240-251. http://dx.doi.org/:http://dx.doi.org/10.1016/j.isprsjprs.2013.02.019

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., . . . Fisher, R. B. (1996). An Experimental Comparison of Range Image Segmentation Algorithms. *IEEE Trans. Pattern Anal. Mach. Intell., 18*(7), 673-689. http://dx.doi.org/:10.1109/34.506791

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A, 4*(4), 629-642. http://dx.doi.org/:10.1364/JOSAA.4.000629

Hough, P. V. C. (1962). Method and means for recognizing complex patterns. (Google Patents).

Huang, J., & Menq, C.-H. (2001). Automatic data segmentation for geometric feature extraction from unorganized 3-D coordinate points. *IEEE Transactions on Robotics and Automation, 17*(3), 268-279. http://dx.doi.org/:10.1109/70.938384

Jiang, X., & Bunke, H. (1994). Fast segmentation of range images into planar regions by scan line grouping. *Mach. Vision Appl., 7*(2), 115-122. http://dx.doi.org/:10.1007/bf01215806

Jolliffe, I. T. J. I. T. (1986). *Principal Component Analysis*.

Khoshelham, K., & Gorte, B. (2009). *Registering point clouds of polyhedral buildings to 2D maps.* Paper presented at the Proceedings of the 3rd ISPRS International Workshop

Masuda, T., Sakaue, K., & Yokoya, N. (1996, 25-29 Aug 1996). *Registration and integration of multiple range images for 3-D model construction.* Paper presented at the Proceedings of 13th International Conference on Pattern Recognition. http://dx.doi.org:10.1109/ICPR.1996.546150

Matas, J., & Chum, O. (2004). Randomized RANSAC with Td,d test. *Image and Vision Computing, 22*(10), 837 - 842. http://dx.doi.org/:https://doi.org/10.1016/j.imavis.2004.02.009

MDL. (2017). Dynascan S250 Retrieved from http://www.hydronav.com/pdf/Dynascan%20S250.pdf

Monnier, F., Vallet, B., Paparoditis, N., Papelard, J. P., & David, N. (2013). Registration of terrestrial mobile laser data on 2D or 3D geographic database by use of a non-rigid ICP approach. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-5/W2*, 193-198. http://dx.doi.org/:10.5194/isprsannals-II-5-W2-193-2013

Morgan, D. (2009). *Using mobile LiDAR to survey railway infrastructure. Lynx mobile mapper*. Paper presented at the FIG Commissions 5, 6 and SSGA Workshop -Innovative Technologies for an Efficient Geospatial Management of Earth Resources, Lake Baikal, Russian Federation

Nguyen, H. L., Belton, D., & Helmholz, P. (2016). SCAN PROFILES BASED METHOD FOR SEGMENTATION AND EXTRACTION OF PLANAR OBJECTS IN MOBILE LASER SCANNING POINT CLOUDS. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLI-B3*, 351-358. http://dx.doi.org/:10.5194/isprs-archives-XLI-B3-351-2016

Nguyen, H. L., Helmholz, P., Belton, D., & West, G. (2015). New methods for segmentation of sparse mobile laser scanning point clouds*The 9th International Symposium on Mobile Mapping Technology (MMT 2015), held in Sydney, Australia,*

Nguyen, H. L, Belton, D., & Helmholz, P. (Submited and under revision-a). Planar surface detection for sparse mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing,*

Nguyen, H. L, Belton, D., & Helmholz, P. (Submited and under revision-b). Review Of Mobile Laser Scanning Registration And The Error Metrics For Quality Evaluation. *Photogrammetric Records,*

Nurunnabi, A. (2014). *Robust Statistical Approaches for Features Extraction in Laser Scanning 3D Point Cloud Data*. Doctor of Philosophy Curtin University of Technologies

Nurunnabi, A., Belton, D., & West, G. (2012, 3-5 Dec. 2012). *Robust Segmentation in Laser Scanning 3D Point Cloud Data*. Paper presented at the Digital Image Computing Techniques and Applications (DICTA), 2012 International Conference on. http://dx.doi.org:10.1109/DICTA.2012.6411672

Nurunnabi, A., Belton, D., & West, G. (2015). Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. *Pattern*

Recognition, 48(4), 1404 - 1419. http://dx.doi.org/:https://doi.org/10.1016/j.patcog.2014.10.014

Previtali, M., Barazzetti, L., Brumana, R., & Scaioni, M. (2014). Laser scan registration using planar features. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XL-5*, 501-508. http://dx.doi.org/:10.5194/isprsarchives-XL-5-501-2014

Puente, I., González-Jorge, H., Arias, P., & Armesto, J. (2012). LAND-BASED MOBILE LASER SCANNING SYSTEMS: A REVIEW. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XXXVIII-5/W12*, 163-168. http://dx.doi.org/:10.5194/isprsarchives-XXXVIII-5-W12-163-2011

Pulli, K. (1999, 1999). *Multiview registration for large data sets.* Paper presented at the Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062). http://dx.doi.org:10.1109/IM.1999.805346

Rabbani, T. (2006). *Automatic reconstruction of industrial installations using point clouds and images.* PhD (Dissertation). Nederlandse Commissie voor Geodesie.

Rabbani, T., Dijkman, S., van den Heuvel, F., & Vosselman, G. (2007). An integrated approach for modelling and global registration of point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing, 61*(6), 355-370. http://dx.doi.org/:http://dx.doi.org/10.1016/j.isprsjprs.2006.09.006

Rabbani, T., van den Heuvel, F. A., & Vosselmann, G. (2006). *Segmentation of point clouds using smoothness constraint.* Paper presented at the IEVM06. http://dx.doi.org:citeulike-article-id:10740344

Raguram, R., Frahm, J.-M., & Pollefeys, M. (2008). *A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus*, Berlin, Heidelberg

Renishaw. (2017). Dynascan S250 point clouds. Retrieved from http://www.renishaw.com/en/dynascan-s250--27332

Ridene, T., & Goulette, F. (2009). Co-registration of DSM and 3D points clouds acquired by a Mobile Mapping System. *Special Issue on Mobile Mapping Technology, , 15*, 824-838.

RIEGL. (2015). RIEGL VQ-450. Retrieved from http://www.riegl.com/uploads/tx_pxpriegldownloads/10_DataSheet_VQ-450_rund_2014-09-02.pdf

Rusinkiewicz, S., & Levoy, M. (2001). *Efficient Variants of the ICP Algorithm.* Paper presented at the Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling. http://dx.doi.org:citeulike-article-id:937464

Shi, Y., Shibasaki, R., & Shi, Z. C. (2008). Towards Automatic Road Mapping by Fusing Vehicle-Borne Multi-Sensor Data*International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, held in Beijing, China,* (pp. 867-872).

Sithole, G., & Vosselman, G. (2003, 22-23 May 2003). *Automatic structure detection in a point-cloud of an urban landscape.* Paper presented at the Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on. http://dx.doi.org:10.1109/DFUA.2003.1219959

Skaloud, J., & Lichti, D. (2006). Rigorous approach to bore-sight self-calibration in airborne laser scanning. *ISPRS Journal of Photogrammetry and Remote Sensing, 61*(1), 47-59. http://dx.doi.org/:http://dx.doi.org/10.1016/j.isprsjprs.2006.07.003

Takai, S., Date, H., Kanai, S., Niina, Y., Oda, K., & Ikeda, T. (2013). Accurate registration of MMS point clouds of urban areas using trajectory. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-5/W2,* 277-282. http://dx.doi.org/:10.5194/isprsannals-II-5-W2-277-2013

Tarsha-Kurdi, F., Landes, T., & Grussenmeyer, P. (2007, 2007-09). *Hough-Transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data.* Paper presented at the ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, Espoo, FinlandRetrieved from https://halshs.archives-ouvertes.fr/halshs-00264843

Torr, P. H. S., & Zisserman, A. (2000). MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Computer Vision and Image Understanding, 78*(1), 138 - 156. http://dx.doi.org/:https://doi.org/10.1006/cviu.1999.0832

Turk, G., & Levoy, M. (1994). Zippered polygon meshes from range images*Proceedings of the 21st annual conference on Computer graphics and* *interactive techniques, held in* (pp. 311-318). 192241: ACM. http://dx.doi.org:10.1145/192161.192241

Vosselman, G., Gorte, B. G. H., Sithole, G., & Rabbani, T. (2004). *Recognising structure in laser scanner point clouds.* Paper presented at the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences

Vosselman, G., & Maas, H. G. (2010). *Airborne and Terrestrial Laser Scanning*. Scotland, UK: Whittles Publishing/CRC Press.

Wang, C., Tanahashi, H., & Hirayu, H. (2001). *Comparison of local plane fitting methods for range data.* Paper presented at the In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA

Xiao, J., Adler, B., & Zhang, H. (2012, 13-15 Sept. 2012). *3D point cloud registration based on planar surfaces.* Paper presented at the 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). http://dx.doi.org:10.1109/MFI.2012.6343035

Yang, B., Dong, Z., Liang, F., & Liu, Y. (2016). Automatic registration of large-scale urban scene point clouds based on semantic feature points. *ISPRS Journal of Photogrammetry and Remote Sensing, 113*, 43-58. http://dx.doi.org/:https://doi.org/10.1016/j.isprsjprs.2015.12.005

Yu, Q., Helmholz, P., & Belton, D. (2017). Semantically Enhanced 3D Building Model Reconstruction from Terrestrial Laser-Scanning Data. *Journal of Surveying Engineering, 143*(4), 04017015. http://dx.doi.org/:doi:10.1061/(ASCE)SU.1943-5428.0000232