Department of Spatial Sciences
Faculty of Science and Engineering

Robust Statistical Approaches for Feature Extraction in Laser Scanning 3D Point Cloud Data

Abdul Awal Md. Nurunnabi

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

Sempember 2014

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: 30/09/2014

To those who sacrifice their lives
for
knowledge,
civilization,
humanity,
peace, and
against war.

Abstract

Laser scanning has spawned a renewed interest in automatic robust feature extraction. Three dimensional point cloud data obtained from laser scanner based mobile mapping systems commonly contain outliers and/or noise. The presence of outliers and noise means that most of the frequently used methods for point cloud processing and feature extraction produce inaccurate and unreliable results i.e. are termed non-robust. Dealing with the problems of outliers and noise for automatic robust feature extraction in mobile laser scanning 3D point cloud data has been the subject of this research.

This thesis develops algorithms for statistically robust planar surface fitting based on robust and/or diagnostic statistical approaches. The algorithms outperform classical methods such as least squares and principal component analysis and show distinct advantages over current robust methods including RANSAC and its derivations in terms of computational speed, sensitivity to the percentage of outliers or noise, number of points in the data and surface thickness. Two highly robust outlier detection algorithms have been developed for accurate and robust estimation of local saliency features such as normal and curvature. Results for artificial and real 3D point cloud data experiments show that the methods have advantages over other existing popular techniques in that they (i) are computationally simpler, (ii) can successfully identify high percentages of uniform and clustered outliers, (iii) are more accurate, robust and faster than existing robust and diagnostic methods developed in disciplines including computer vision (RANSAC), machine learning (uLSIF) and data mining (LOF), and (iv) have the ability to denoise point cloud data. Robust segmentation algorithms have been developed for multiple planar and/or non-planar complex surfaces e.g. long cylindrical and approximately cylindrical surfaces (poles), lamps and sign posts extraction. A region growing approach has been developed for segmentation algorithms and the results demonstrate that the proposed methods reduce segmentation errors and provide more robust feature extraction. The developed methods are promising for surface edge detection, surface reconstruction and fitting, sharp feature preservation, covariance statistics based point cloud processing and registration. An algorithm has also been introduced for merging several sliced segments to allow large volumes of laser scanned data to be processed seamlessly. In addition, the thesis presents a robust ground surface extraction method that has the potential for being used as a pre-processing step for large point cloud data processing tasks such as segmentation, feature extraction, classification of surface points, object detection and modelling. Identifying and removing the ground then allows more efficiency in the segmentation of above ground objects.

Acknowledgement

"What soon grows old? Gratitude."
Aristotle

The passage through PhD research was an exciting and wonderful experience in my life which would not have been possible without the help of numerous people, all of which I can remember with appreciation. First and most importantly I would like to express my heartfelt gratitude to my thesis supervisor, Professor Geoff West for his kind help, advice, encouragement, spontaneous guidance and cordial support from the first hour when I arrived at the Perth Airport and then through every moment and stage in this long journey. He spent many hours editing and improving the readability and presentation of my thesis as well as my academic papers. He provided me research freedom which made my study relaxing, pleasing and more fruitful.

I am thankful to my co-supervisor Dr David Belton who provided me guidance; positive thoughts and kind help with many clear ideas and fruitful discussion to make my research smooth. I had the pleasure of sharing the Photogrammetry lab with him and with my fellow colleagues: Richard Palmer, Michael Borck and others. I am also thankful to my associate supervisors Dr Stuart Gordon and Dr Ireneusz Baran, AAM for their kind acceptance to supervise me, Associate Professor Jonathan Kirby, Thesis Chairperson, and Professor Bert Veenendaal, Head of Department of Spatial Sciences for their kind support. I am thankful to the staff at Kurrajong Village, Curtin University housing accommodation especially Mr Norman Johnson and Mrs Amanda Tan for their cordial support.

I am grateful to have been awarded an International Postgraduate Research Scholarship (IPRS) award at Curtin University and a top up scholarship from the Cooperative Research Centre for Spatial Information (CRCSI), whose activities are funded by the Australian Commonwealth's CRC programme. I am thankful to the office and academic staff, colleagues and friends in the Department of Spatial Sciences, Curtin University and in the CRCSI who contributed in many ways to my research. My sincere thanks to the CRCSI industry partners who provided me laser scanning data. Love to Australia.

I am highly appreciative to my MPhil supervisor Professor Mohammed Nasser, University of Rajshahi who has invaluable contribution to build my research career. I am thankful to my MPhil co-supervisor Dr A.H.M. Rahmatullah Imon, Professor Ball State University, USA from whom I learnt much. I would like to thank Dr Ali S. Hadi, Distinguished Professor, American University in Cairo and Emeritus Professor of the Cornell University, USA; Dr Mir Masoom Ali, Distinguished University Professor and Emeritus Professor, Ball State University, Indiana, USA; Dr Arun Kumar Basak, Emeritus Professor, University of Rajshahi, Bangladesh; the late Professor Randy Sitter, University of Simon Fraser, Canada; Professor David Suter, University of Adelaide, Dr Honghua Dai, Deakin University, Dr Shawkat Ali, Central Queensland University, and Dr Ashraf Dewan, Curtin University, Australia for their valuable time, help, inspiration and advice. I would also like to thank the anonymous reviewers and the participants of the conferences who made constructive comments to improve my papers, presentations and research.

I thank all of my teachers, students, friends, relatives, my colleagues Professor Salahuddin Ahmad, Kazi Tareq Ullah, Mahmudul Hasan, Mamun A. Kayeum, Abdur Rahim, Jahangir Alam, and M. Sadek, and all others who helped and inspired me directly or indirectly, and all who have wished me well in my study, research and other purposes.

My eternal gratitude goes to my parents for their affection, and who have always tried to make my life comfortable, put my education above their personal happiness and were always praying for me for success in this life and the hereafter. My thanks to my beloved younger brother, sister, mother-in-law and their family for their constant inspiration, contribution and love. In the same regard, I am also indebted to my wife, whose companionship and continuous sacrifice have been a source of strength; we have shared many discussions about my research and her patience kept me tranquil in my personal and professional life.

At last, but far from the least, my unconditional and unlimited prostration to the almighty Allah for allowing me the joy of doing my PhD, such a colourful event for any educated person, for completing this research in time and for giving me the opportunity to contribute a little for the betterment of human civilization with his endless blessings.

Abdul Awal Md. Nurunnabi

Contents

List of Figures

List of Tables

List of Algorithms

Publications and Presentations based on this Thesis

Journal Publications (peer-reviewed and fully refereed)

1. **Abdul Nurunnabi**, David Belton, and Geoff West, 2014. Robust Statistical Approaches for Local Planar Surface Fitting in 3D Laser Scanning Data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 96, pp. 106–122.

2. **Abdul Nurunnabi**, Geoff West, and David Belton, 2014. Outlier Detection and Robust Normal-Curvature Estimation in Mobile Laser Scanning 3D Point Cloud Data. *Pattern Recognition*, In press, http://dx.doi.org/10.1016/j.patcog.2014.10.014.

3. **Abdul Nurunnabi**, David Belton, and Geoff West. Robust Segmentation for Large Volumes of Laser Scanner 3D Point Cloud Data. Under revision, *ISPRS Journal of Photogrammetry and Remote Sensing*.

4. **Abdul Nurunnabi**, Geoff West, and David Belton. Robust Ground Surface Extraction in Laser Scanning Point Cloud Data. To be submitted soon to the *IEEE Transactions on Geoscience and Remote Sensing*.

5. **Abdul Nurunnabi**, Geoff West, and David Belton. Robust Feature Extraction in Laser Scanning 3D Point Cloud Data. Under preparation.

Conference Publications (peer-reviewed and fully refereed)

6. **Abdul Nurunnabi**, David Belton, and Geoff West, 2013. Diagnostics based Principal Component Analysis for Robust Plane Fitting in Laser Data. In: *Proceedings of the 16th International Conference on Computer and Information Technology* (ICCIT), pp. 484–489, Khulna, Bangladesh, 21–23 December.

7. **Abdul Nurunnabi**, Geoff West, and David Belton, 2013. Robust Locally Weighted Regression for Ground Surface Extraction in Mobile Laser Scanning 3D Data. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume II-5/W2, pp. 217–222. Presented in the ISPRS Workshop on Laser Scanning, Antalya, Turkey, 11–13 November.

8. **Abdul Nurunnabi**, Geoff West, and David Belton, 2013. Robust Outlier Detection and Saliency Features Estimation in Point Cloud Data. In: *Proceedings of the 10th Canadian Conference on Computer and Robot Vision* (CRV), pp. 98–105, Regina, Saskatchewan, Canada, 28–31 May.

9. **Abdul Nurunnabi**, David Belton, and Geoff West, 2012. Robust and Diagnostic Statistics: A Few Basic Concepts in Mobile Mapping Point Cloud Data Analysis. In: *Proceedings of the International Conference on Statistical Data Mining for Bioinformatics, Health, Agriculture and Environment*, pp. 591–602, Rajshahi, Bangladesh, 22–24 December.

10. **Abdul Nurunnabi**, David Belton, and Geoff West, 2012. Robust Segmentation in Laser Scanning 3D Point Cloud Data. In: *Proceedings of the 14th International Conference on Digital Image Computing: Techniques and Applications* (DICTA), pp.1–8, Fremantle, Australia, 3–5 December.

11. **Abdul Nurunnabi**, David Belton, and Geoff West, 2012. Robust Segmentation for Multiple Planar Surface Extraction in Laser Scanning 3D Point Cloud Data. In: *Proceedings of the 21st International Conference on Pattern Recognition* (ICPR), pp. 1367–1370, Tsukuba Science City, Japan, 11–15 November.

12. **Abdul Nurunnabi**, David Belton, and Geoff West, 2012. Diagnostic-Robust Statistical Analysis for Local Surface Fitting in 3D Point Cloud Data. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 1–3, pp. 269–274. Presented in the XXII Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS), Melbourne, Australia, 25 August – 01 September.

13. **Abdul Nurunnabi**, Geoff West, and David Belton, 2015. Robust methods for feature extraction from mobile laser scanning 3D point clouds. To be submitted In: Research@Locate 15, Brisbane, Australia, 9–12 March.

Conference Presentation (abstract based)

Abdul Nurunnabi, David Belton, and Geoff West, 2013. Robust Feature Extraction for 3D Point Clouds from Vehicle-based Laser Scanner. In: *WALIS Forum Conference*, Crown Perth, Western Australia, Australia, 7–8 November.

Invited Seminar Presentation

Abdul Nurunnabi, Robust Outlier Detection and Saliency Features Estimation in Point Cloud Data. *Department of Geomatics Engineering, University of Calgary*, Calgary, Canada. 26 May, 2013.

Acronyms

ABOF	Angle Based Outlier Factor
BP	Breakdown Point
CIN	Correctly Identified Noise
CIIR	Correct Inlier Identification Rate
CIR	Correctly Identified Regular (inlier) points
COIR	Correct Outlier Identification Rate
DB	Distance Based
DDR	Direct Density Ratio
DEM	Digital Elevation Model
DetMCD	Deterministic MCD
DetDRPCA	Deterministic Diagnostic Robust PCA/DetMCD based Diagnostic Robust PCA
DetRD	Deterministic MCD based Robust Distance
DetRPCA	DetMCD based Robust PCA
DetRD-PCA	DetRD based diagnostic PCA/Deterministic MCD based diagnostic PCA
DMI	Distance Measurement Instrument
DMSE	Difference in MSE
DRPCA	Diagnostic Robust PCA
DSM	Digital Surface Model
DTM	Digital Terrain Model
ED	Euclidean Distance
FDN	Fixed Distance Neighbourhood
FMCD	Fast-MCD
FNR	False Negative Rate
FPR	False Positive Rate

FRD	Fast-MCD based Robust Distance
FRD-PCA	FMCD based diagnostic PCA/FRD based diagnostic PCA
FRPCA	FMCD based RPCA
FDRPCA	Fast-MCD based Diagnostic Robust PCA
GPS	Geographic Positioning System
GNSS	Global Navigation Satellite System
HT	Hough Transform
IMU	Inertial Measurement Unit
IF	Influence Function
IQR	Inter Quartile Range
kNN	k Nearest Neighbourhood
LiDAR	Light Detection and Ranging
LMS	Least Median of Squares
lowess/loess	LOcally WEighted Scatterplot Smoother
LOF	Local Outlier Factor
LS	Least Squares
LTS	Least Trimmed Square
LWLS	Locally Weighted Least Squares
LWLMS	Locally Weighted LMS
LWLTS	Locally Weighted LTS
LWR	Locally Weighted Regression
MCMD	Maximum Consistency with Minimum Distance
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimator
MAD	Median Absolute Deviation
MCD	Minimum Covariance Determinant
MCS	Maximum Consistent Set
MVE	Minimum Volume Ellipsoid
MLESAC	Maximum Likelihood Estimation SAmple Consensus
MLS	Mobile Laser Scanning
MD	Mahalanobis Distance
MMS	Mobile Mapping System
MR	Masking Rate

MS	Minimal Subset
MSAC	M-estimator SAmple Consensus
MSE	Mean Squared Error
OD	Orthogonal Distance
OGK	Orthogonalized Gnanadesikan and Kettenring
OP	Outlier Percentage
OS	Over Segment
OSVM	One-class Support Vector Machine
PCA	Principal Component Analysis
PC	Principal Components
PP	Projection Pursuit
PS	Proper Segment
RANSAC	RANdom SAmple Consensus
RD	Robust Distance
Rz-score	Robust z-score
RDPCA	MCMD_Z based Robust Diagnostic PCA
RPCA	Robust PCA
RLS	Reweighted Least Squares
RLWLS	Robust Locally Weighted Least Squares
RLWR	Robust Locally Weighted Regression
RMD	Robust Mahalanobis Distance
SD	Score Distance
SVD	Singular Value Decomposition
SVDD	Support Vector Data Description
StD	Standard Deviation
SR	Swamping Rate
TIN	Triangular Irregular Network
TLS	Total Least Squares
TPR	True Positive Rate
TNR	True Negative Rate
uLSIF	unconstrained Least Squares Importance Fitting
US	Under Segment
WLS	Weighted Least Squares

Chapter 1

"A journey of a thousand miles must begin with a single step."
Lao Tzu

"The beginning is the most important part of the work."
Plato

Introduction

Vehicle based laser scanning otherwise known as Mobile Laser Scanning (MLS) has been proposed as a technology for acquiring a three-Dimensional (3D) survey of the environment and objects in the vicinity of the mapping vehicle accurately, quickly and safely (Tao and Li, 2007; Beraldin et al., 2010). The laser scanning provides explicit and dense 3D measurements (Rabbani, 2006), unlike competing photogrammetric methods that generate point clouds from image matching. Although this technology is still under development, due to its cost effectiveness and reasonable data accuracy it has been used in many applications and research areas include 3D city modelling, corridor (e.g. road and rail) asset and inventory maintenance and management, abnormal load route determination, environmental monitoring, accidental investigation, industrial control, construction management, digital terrain modelling, archaeological studies, marine and coastal surveying, telegeoinformatics, change detection for military and security forces, man-induced and natural disaster management, geographical information systems and simulation (Tao and Li, 2007; Graham, 2010; Kutterer, 2010; Petrie, 2010).

Visualization, analysis, understanding and modelling from MLS data mainly rely on laser scanning point cloud data processing. Feature extraction is a fundamental step for point cloud processing and product creation that is closely related to surface reconstruction, fitting and object modelling. Point cloud segmentation is a prior task for surface reconstruction, feature extraction, object recognition and modelling that comprises the majority of the research in point cloud analysis (Pfeifer and Briese, 2007). In surface reconstruction, the quality of the output surface depends on how well the estimated normals approximate the true normals of the sampled surface (Dey et al., 2005). The presence of outliers is common in point cloud data, which can make the estimates of the normals, curvatures, etc. erroneous, leads to misleading analysis, and produces inconsistent (non-robust) results. We investigate outlier problems in classical existing methods that have been widely used for feature extraction and employ diagnostic and robust statistical techniques as a solution to outlier influence. Since, diagnostic and robust statistical approaches are complementary and have the same goal of robust estimation, in general, we can use the term 'robust approaches' to group robust and diagnostic approaches together. In this thesis, the feature extraction will be performed in two stages: (i) robust saliency feature, e.g. normal and curvature estimation, and (ii) segmentation. The estimated robust saliency features will be used for robust segmentation. Therefore the ultimate goal of this thesis is to develop robust and diagnostic statistics based algorithms to estimate robust local saliency features: normal and curvature for robust segmentation. This thesis also introduces robust algorithms for ground surface extraction in laser scanning point cloud data that has the potential for separating ground and non-ground objects. This is advantageous because the ground can be eliminated and speed up the many complex methods needed for non-ground features analysis.

1.1 Mobile Laser Scanning and Point Cloud Data

MLS is a non-invasive, state-of-the-art solution that incorporates various navigation and remote sensing technologies on a common moving platform. Mapping from moving vehicles (Figure 1.1) has been around for at least as long as photogrammetry has been practiced (Schwarz and EI-Sheimy, 2007). In the early 1990s, a big advance was the availability of the Geographic Positioning

System (GPS) that was able to provide reasonably accurate positioning of the mobile sensor platforms (Novak, 1993). On board the mobile vehicle are advanced imaging and ranging devices, such as cameras, laser scanners or Light Detection and Ranging (LiDAR) systems, and navigation/positioning/ geo-referencing devices such as a Global Navigation Satellite System (GNSS) for the determination of the position of the moving platform, Inertial Measurement Unit (IMU) that contains sensors to detect rotation and acceleration are used for determining the local orientation of the platform. A Distance Measurement Instrument (DMI) or odometer, which is usually connected to a wheel of the vehicle is used to provide linear distance in case of GNSS outage. The constant integration between GPS and IMU deals with a possible loss of signal sent by the satellites and to constantly maintain the high accuracy of data acquisition. The two main components of a Mobile Mapping System (MMS) are geo-referencing based on navigation sensors, and kinematic modelling of imaging sensors. The reader is referred to El-Sheimy (2005), Ip et al. (2007), and Lichti and Skaloud (2010) for details about geo-referencing. The sensor arrangement is necessary to maintain the alignment and accuracy between the navigation equipment and the sensors. Inside the vehicle there is a computer with storage and operational software to control the data acquisition mission. Typically a two-person crew performs the mission, one for driving the vehicle and the other for operating and managing the sensors. Mid- and long-range laser scanners are usually based on time-of-flight technology, which measures time delays created by light waves travelling in a medium from a source to a target surface and back to the source. High-speed counters are used to measure the time-of-flight of the high energy and short length light waves. Laser scanners mounted on the platform usually at a 45° angle to the vehicle track swing the laser beam through 360°. The unit can rotate at 20 to 200 revolutions per second and the laser is pulsed with frequencies of up to 200 kHz. Performance includes a spatial resolution of up to 1 cm at 50 km/hour, range > 100 metres (with 20% reflectivity), measurement precision ± 7 mm (1σ), at operating temperatures $-20°$ C to $+40°$ C (Arditi et al. (2010); Optech[1]; McMullen Nolan[2]). These configurations and advantages vary for different systems. MMS are now able to collect more than 600,000 points per second. According to Graham (2010), the

[1] http://www.optech.com/index.php/products/mobile-survey/, Accessed: 28-08-2014

[2] http://mcmullennolan.com.au/documents/laser-scanning-for-road-corridors.pdf, Accessed:28-08-2014

achievable absolute accuracy of the 3D data can be as good as 1.5–2 cm (following adjustment to control). MLS means that, for example, the two weeks of good weather needed to collect data for a 30-mile highway corridor using other technologies can be replaced with a MMS mission in 40 to 60 minutes. Once the data has been acquired, all the data processing can be performed in the back office. MMS significantly improves safety for data collectors, which is a major concern in highway works. To get the desired results and to extract 3D (x, y, z) coordinates of mapping objects from the geo-referenced images, modelling and data fusion is required. Data fusion is necessary for merging data from the various sources or sensors.

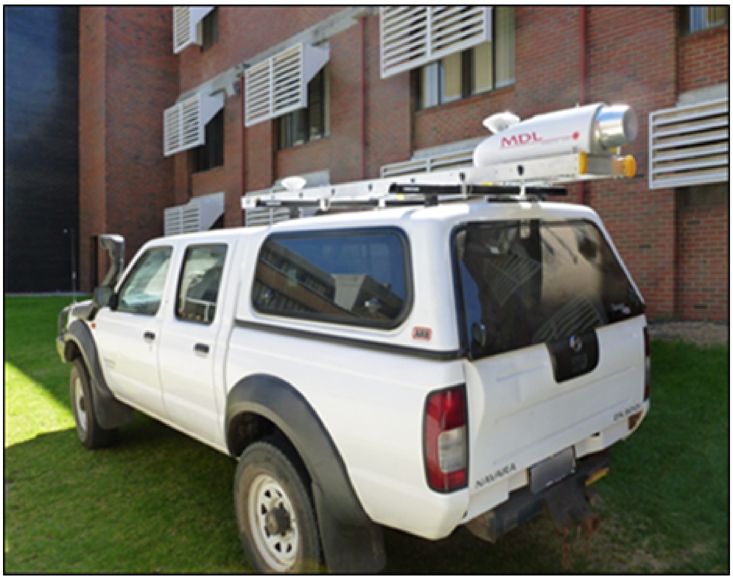


Figure 1.1 Mobile mapping vehicle with onboard sensors (Courtesy, Department of Spatial Sciences, Curtin University).

MLS can produce terabytes of 3D geospatial point cloud data (Figure 1.2) defined by their x, y, and z coordinates (latitude, longitude and elevation). Point cloud data may have colour (r, g, b) information from co-registered cameras and intensity from the reflected laser beam. The output point cloud data is generally stored in an industry standard format called ‘LAS’, which encodes the data into a point based binary or text file. The reader is referred to Tao and Li (2007); Shan and Toth (2009); Toth (2009); Graham (2010); Kutterer (2010); Petrie (2010) and Vosselman and Maas (2010) for more information about MMS. The various stages in the workflow from MMS data collection to output is sketched in Figure 1.3.

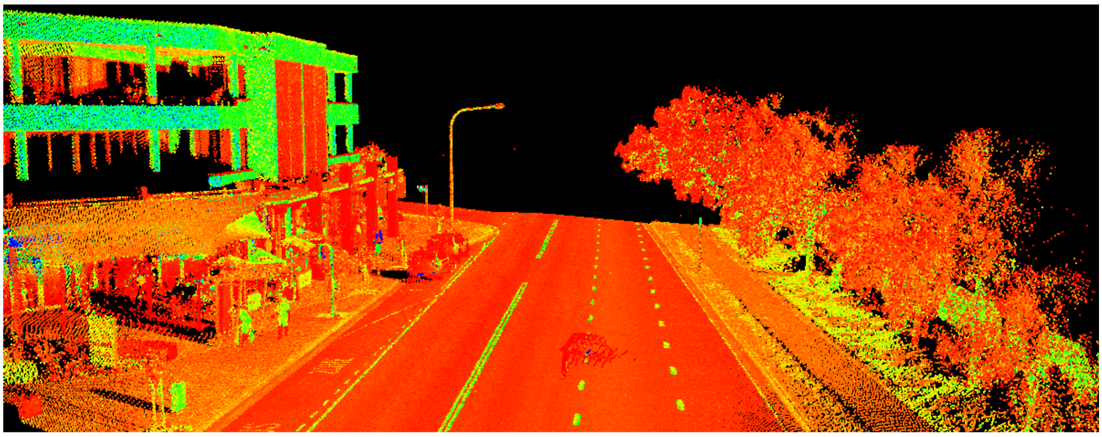


Figure 1.2 Point cloud data with laser intensity (collected by the AAM Group[3]).

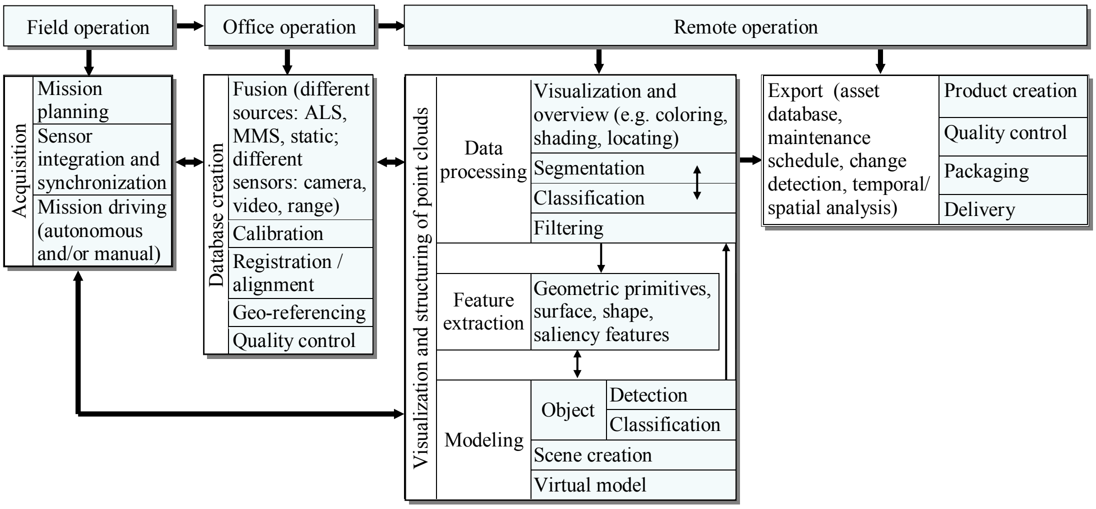

Field operation
Office operation
Remote operation
Acquisition
Mission planning
Sensor integration and synchronization
Mission driving (autonomous and/or manual)
Database creation
Fusion (different sources: ALS, MMS, static; different sensors: camera, video, range)
Calibration
Registration / alignment
Geo-referencing
Quality control
Visualization and structuring of point clouds
Data processing
Visualization and overview (e.g. coloring, shading, locating)
Segmentation
Classification
Filtering
Feature extraction
Geometric primitives, surface, shape, saliency features
Modeling
Object
Detection
Classification
Scene creation
Virtual model
Export (asset database, maintenance schedule, change detection, temporal/ spatial analysis)
Product creation
Quality control
Packaging
Delivery

Figure 1.3 MMS data collection to output workflow.

1.2 Motivation

The data acquired from MLS is just data. It has no knowledge about its statistical distribution and specific surface shape, has complex topology and geometrical discontinuities, has inconsistent point density, may have sharp features and may even be missing pieces e.g. holes. It consists of limited and/or complete multiple structures that may contain features varying in size, density and complexity. In addition to the above, it is impractical to think of point cloud data without

[3] http://www.aamgroup.com/

outliers. Outliers occur because of noise, objects getting in the way including rain, birds and other possibly unimportant features e.g. poles in front of a building. Inclusion of outliers in point cloud data exacerbates the problems for reliable and robust point cloud processing and feature extraction. However, due to the large volumes of point cloud data automatic or semi-automatic approaches are necessary for point classification, segmentation and feature extraction (Sotoodeh, 2006; Belton, 2008; Vosselman and Maas, 2010). Searching through the literature revealed that in spite of the recognition and inevitability of outlier problems, many authors frequently use classical (non-robust) techniques including Least Squares (LS) and Principal Component Analysis (PCA) for point cloud processing tasks without any treatment of outliers (Rabbani, 2006; Belton, 2008). It is known that most of the classical techniques work well only for high-quality data and fail to perform adequately in the presence of outliers. For example, LS and PCA are sensitive to outliers and give inconsistent and misleading estimates of the model parameters (Nurunnabi et al., 2012a, 2013a,c, 2014a). Therefore, automatic, robust and fast methods are necessary for accurate feature extraction that can deal with outliers.

The detection of outliers and parameter estimation without the influence of outliers is a fundamental task in statistics. Robust and diagnostic statistics are two interrelated and complementary branches of statistics that deal with outliers. Stewart (1999) stated that it is important for the reader to note that robust estimators are not necessarily the only or even the best technique that can be used to solve the problems caused by outliers and multiple populations (structures) in all contexts. The necessities of robust and diagnostic methods in statistics, computer vision, machine learning, pattern recognition, photogrammetry and remote sensing have been well described in the literature (Huber, 1981; Hampel et al., 1986; Rousseeuw and Leroy, 2003; Stewart, 1999; Meer, 2004; Wang et al., 2012a,b). To our knowledge, there is nothing in the literature in the context of robust PCA and diagnostics PCA for feature extraction and segmentation in MLS point cloud data. This research adopts robust and diagnostic statistical methods and develops new methods that contribute to robust feature extraction in laser scanning 3D point cloud data.

1.3 Goals

The main goal of the thesis is robust feature extraction and segmentation for laser scanning point cloud data. The resultant segments are the subsets of the point cloud, which are homogeneous within the subsets/groups and represent different features in the same object or different objects. The main concern for feature extraction is: the segmentation results and the extracted features should be statistically robust and accurate in the presence of outliers and/or noise. To get the segmentation results to be resilient to outliers, this thesis presents robust segmentation algorithms, which are based on robust local saliency features: namely normal and curvature. Therefore, to achieve the final goal of robust feature extraction, this thesis addresses the following objectives:

- Investigating appropriate robust and outlier detection methods from statistics, computer vision, data mining, pattern recognition and machine learning to fit local planar surfaces, to get robust estimates for necessary model parameters, and for robust segmentation and other point cloud processing tasks in the presence of outliers and/or noise in the data.

- Developing methods for outlier detection and denoising in point cloud data.

- Developing robust segmentation algorithms for planar and non-planar complex objects.

- Developing an algorithm for seamlessly merging several pieces of segmented slices to process large volumes of point cloud data.

- Developing an efficient and robust ground surface extraction algorithm that can classify ground and off-ground surface points in point cloud data.

1.4 Thesis Organization

To achieve the objectives mentioned in the previous section, we outline the thesis and state the main contributions in each chapter. The structure of the thesis is sketched in Figure 1.4. Each chapter starts with an introduction and contains some common sections: literature review, related principles, proposed methods/algorithms, experiments and conclusions.

Chapter 2 presents the basic principles, ideas, methods and a short review of the literature related to the proposed algorithms.

Chapter 3 proposes robust and diagnostic statistical algorithms for local planar surface fitting. The algorithms use Fast-MCD (Hubert et al., 2005) and Deterministic MCD (Hubert et al., 2012) based robust and diagnostic Principal Component Analysis (PCA) approaches. The performance of the diagnostic and robust statistical methods is demonstrated through several simulated and real MLS datasets. Comparison with LS, PCA, RANSAC, and MSAC demonstrate that the proposed algorithms are significantly more efficient, faster and produce more accurate estimates of the plane parameters and robust local saliency features: normal and curvature.

Chapter 4 introduces two robust distance based statistical techniques for outlier detection, point cloud denoising, and for robust local saliency feature estimation in laser scanning point cloud data. The new algorithms couple the ideas of point to plane orthogonal distance and consistency among the local surface points to get Maximum Consistency with Minimum Distance (MCMD). The new techniques find outliers using: (i) a univariate robust z-score, and (ii) a Mahalanobis type robust distance. Finally, the algorithms fit planes by using PCA to the cleaned data and estimate saliency features. The algorithms are significantly faster than the existing robust and diagnostic statistical procedures proposed in Chapter 3.

Chapter 5 proposes region growing approach based two robust segmentation algorithms, one is especially for multiple planar surface extraction and the other is able to extract planar and non-planar surfaces of complex objects. The algorithms use robust saliency features in the region growing process that are computed using the methods from Chapter 4. We demonstrate and evaluate the proposed algorithms through simulated and real laser scanning data. The algorithms are able to reduce over and under segmentation. Results show the proposed algorithms outperform non-robust classical methods and are significantly better than other robust methods such as RANSAC and MSAC.

Chapter 6 introduces robust algorithms for ground surface extraction in 3D point cloud data. The new algorithms use robust locally weighted regression to get the ground level and to extract the ground surface. Essentially it segments the ground from non-ground objects allowing focussed processing on either.

Finally, conclusions for the thesis are presented in Chapter 7. We summarise the achievements and suggest some directions of future research. It details the significant advances of the research as well as advantages and limitations of the proposed algorithms in the thesis.

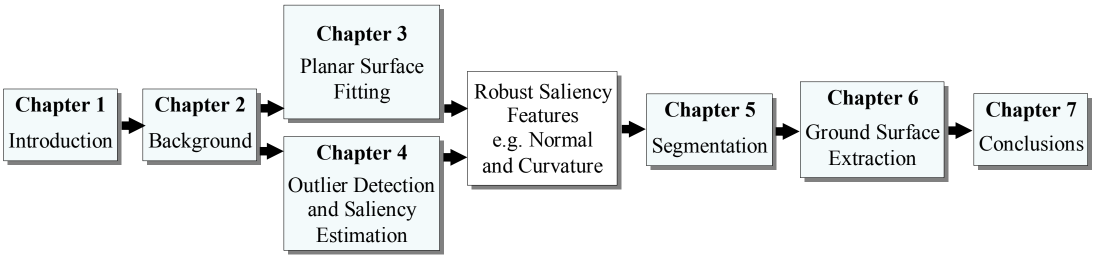


Figure 1.4 Thesis outline.

Chapter 2

"There is nothing so practical as a good theory."
Kurt Lewin

"A theory can be proved by experiment; but no path leads from experiment to the birth of a theory."
Albert Einstein

Basic Ideas, Related Principles, Methods, and a Brief Literature Review for the Proposed Algorithms

In this chapter, we summarize fundamental ideas, related principles and methods used in the proposed algorithms and/or for comparison. Classical approaches such as LS and PCA, and their limitations in the presence of outliers are discussed as these are the two most common statistical approaches that have been widely applied in point cloud processing. A brief introduction is presented on issues concerning outliers as well as on approaches to deal with such outliers. These approaches are robust and diagnostic statistics. The algorithms developed in this thesis are based on these two approaches individually or in combination. We discuss outlier detection, robust location and scatter, robust distance, and robust PCA. In particular, RANSAC is considered because it is one of the most popular robust methods used in many subjects including computer vision, computer graphics, pattern recognition, robotics, photogrammetry and remote sensing for model parameter estimation in the presence of outliers. A variant of RANSAC, MSAC also is considered for comparison because it uses a robust M-estimator and has been recognized as a most competitive one. Several outlier detection methods from data mining, machine learning and pattern recognition literature are presented before regression analysis is discussed. We present basic ideas about boxplot as a robust visualization tool used for performance comparison between the different methods. This chapter also presents a short literature review for the proposed algorithms.

2.1 Classical Statistical Approaches

In classical statistics, methods try to fit all the data points as well as possible, but they rely heavily on certain assumptions (e.g. normality, linearity and independence), which are infrequently met in practice. The problem with classical parametric statistics is that they derive optimal procedures under the assumption that an exact parametric model is the best representation for the real world situation. We discuss LS and PCA in the context of planar surface fitting.

2.1.1 Least Squares

Minimizing the sum of the squared residuals, namely Least Squares (LS) has been used in different ways for plane fitting in many applications (Wang et al., 2001; Klasing et al., 2009). For a set of 3D data points $\{p_i(x_i, y_i, z_i); i = 1, \ldots, n\}$, a plane equation can be defined as:

$$ax + by + cz + d = 0, \tag{2.1}$$

where a, b and c are the slope parameters, and d is proportional to the distance of the plane to the origin. In classical LS, the data points are expressed by a functional relation, $z = f(x, y)$, and the sum of the squared residuals in the z direction is minimized using:

$$\min \sum_{i=1}^{n} (z_i - \hat{z}_i)^2 = \min \sum_{i=1}^{n} r_i^2 = \min \sum_{i=1}^{n} d_{vi}^2, \tag{2.2}$$

where the i^{th} residual r_i or d_{vi} is the vertical distance between the i^{th} point z_i and its fit $\hat{z}_i$, as shown in Figure 2.1a. Minimization of vertical squared errors is not ideal, because it considers errors only in the vertical or z direction (Kwon et al., 2004). To overcome the bias in this one direction, the Total Least Squares (TLS; Huffel and Vandewalle, 1991) approach is used that minimizes the squared sum of the orthogonal distances d_{oi} between the points and the fitted plane, shown in Figure 2.1b:

$$\min \sum_{i=1}^{n} r_i^2 = \min \sum_{i=1}^{n} d_{oi}^2. \tag{2.3}$$

The parameters of the fitted plane can be determined by solving:

$$\min \sum_{i=1}^{n} ((p_i - \bar{p})^T \cdot \hat{n})^2, \tag{2.4}$$

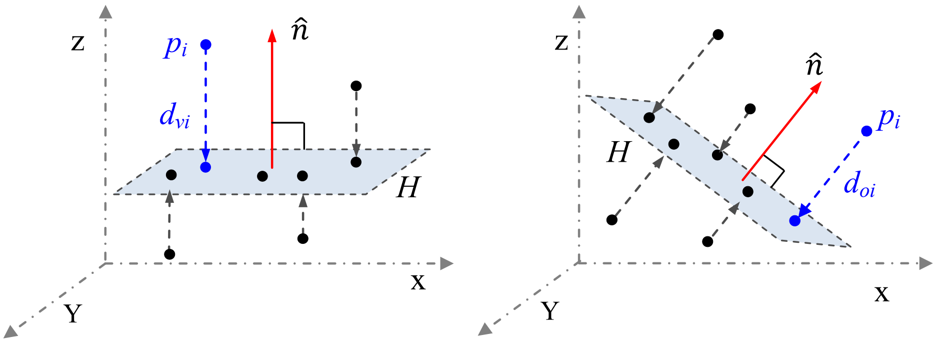
where $\bar{p}$ is the centre of the data:

$$\bar{p} = \frac{1}{n} \sum_{i=1}^{n} (x_i, y_i, z_i), \tag{2.5}$$

where $\hat{n}$ is the unit normal to the fitted plane, and $(p_i - \bar{p})^T \cdot \hat{n}$ is the orthogonal distance between the plane and the point p_i. One of the most common approaches for plane parameter estimation is the eigenvalue method, which minimizes $\sum_{ij}(ax_{ij} + by_{ij} + cz_{ij} + d)^2$ under the constraint: $a^2 + b^2 + c^2 + d^2 = 1$. This minimization is equivalent to finding the eigenvector that corresponds to the least eigenvalue of the matrix:

$$M = \frac{1}{n} \sum_{ij} (x_{ij}, y_{ij}, z_{ij}, 1)^T (x_{ij}, y_{ij}, z_{ij}, 1). \tag{2.6}$$

This method is also known as *PlaneSVD* (Klasing et al., 2009).



(a)

(b)

Figure 2.1 Fitted planes and estimated normals (red arrows): (a) least squares method, and (ii) total least squares method.

2.1.2 Principal Component Analysis

Principal Component Analysis (PCA) is one of the most widely used multivariate/multidimensional statistical techniques for dimension reduction and data visualization (Jolliffe, 1986). Research on PCA dates back to Pearson (1901), and has been recognized as one of the most important techniques for data compression and feature extraction. Its purpose is to find a small number d of linear combinations of the m observed variables that can represent most of the variability of the data. Geometrically, this is equivalent to finding a d (where $d < m$) dimensional linear manifold minimizing the mean squared orthogonal distances of the data points to the manifold (Maronna, 2005). It works as a basis transformation to diagonalize an estimate of the covariance matrix of the data (Schölkopf et al., 1997). PCA proceeds by finding directions of maximum or minimum variability in the data space, and tries to characterize the data by determining orthonormal axes which maximally decorrelate the dataset. By transformation it generates a new set of uncorrelated and orthogonal variables that can explain the underlying covariance structure of the data. The new set of variables, Principal Components (PCs), are the linear combinations of the mean centered original variables that rank the variability in the data through the variances, and produces the corresponding directions using the eigenvectors of the covariance matrix (Johnson and Wichern, 2002; Lay, 2012). In the case of point cloud processing, we often study the nature of the data within a local neighbourhood of a point of interest p_i that can be investigated through the study of the covariance matrix of the neighbourhood. It is also called the local covariance analysis. For 3D point cloud data, we can define the covariance matrix of k points in a local neighbourhood as:

$$C_{3\times 3} = \frac{1}{k}\sum_{i=1}^{k}(p_i - \bar{p})(p_i - \bar{p})^T, \tag{2.7}$$

where $\bar{p}$ is the centre of the data. The denominator k can be replaced by $k-1$ for an unbiased estimator depending on whether the neighbourhood is considered as a sample of a point cloud (Walpole et al., 1998; Kamberov and Kamberova, 2004), or whether the neighbourhood is considered as a unique population since it does not necessarily reflect the properties of the single surface structure or point cloud (Berkmann and Caelli, 1994). Performing Singular Value Decomposition (SVD;

Golub and Reinsch, 1970; Searle, 2006) on the covariance matrix, i.e. solving the eigenvalue equation:

$$\lambda V = CV, \tag{2.8}$$

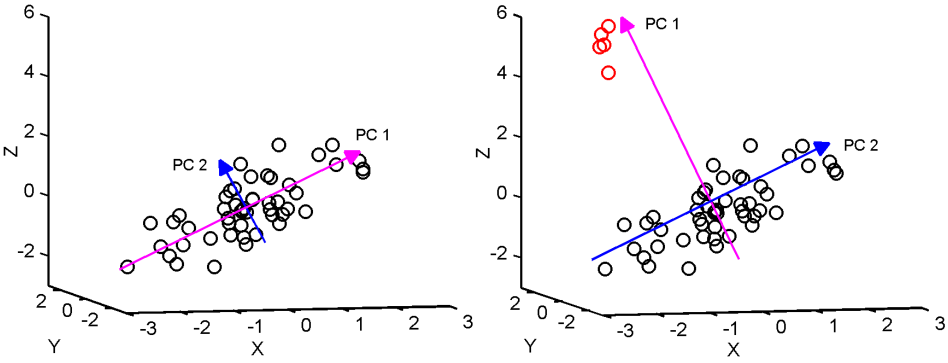
produces λ, the matrix consisting of eigenvalues as its diagonal elements, and V, the eigenvector matrix that contains eigenvectors or PCs as its columns. Given the required eigenvectors and the corresponding eigenvalues, C can be rewritten as:

$$C = \sum_{i=0}^{2} \lambda_i v_i v_i^T, \qquad 0 \leq \lambda_0 \leq \lambda_1 \leq \lambda_2 \tag{2.9}$$

where λ_i and v_i are the i^{th} eigenvalue and eigenvector, respectively. The eigenvalues denote the variances along the associated eigenvectors (Johnson and Wichern, 2002). The PCs are usually ranked in descending order of explaining the underlying data variability according to the descending order of the corresponding eigenvalues, so the first PC is the eigenvector corresponding to the largest eigenvalue (Figure 2.2a). For plane fitting, the first two PCs form an orthogonal basis for the plane, and the third PC is orthogonal to the first two and approximates the normal of the fitted plane. Since the first two PCs explain the variability as much as possible with two dimensions, the fitted plane is the best 2D linear approximation to the data, which is known as the best-fit-plane. The third PC corresponding with the least eigenvalue expresses the least amount of variation and is used to get the estimate of the plane parameters. Figure 2.2a shows two PCs that indicate data variations in their respective directions. To see the outlier influence on PCs, Figure 2.2b adds some outliers (red points) that do not follow the pattern of the majority points. The first PC (magenta arrow) now wrongly indicates more data variability because of the outliers in the wrong direction. The classical PCA is also known as *PlanePCA* (Klasing et al., 2009). By solving the following equation,

$$((x \quad y \quad z)^T - \bar{p}) \cdot v_0 = 0, \tag{2.10}$$

it can be shown that the fitted plane determined by LS and SVD are equivalent (Shakarji, 1998). The reader is referred to Jolliffe (1986), Diamataras and Kung (1996), and Johnson and Wichern (2002) for more details on PCA.



(a) (b)

Figure 2.2 (a) First two PCs are showing the directions of the data variations to their respective directions, and (b) influence of outliers on PCs.

2.2 Outlier, Robust and Diagnostic Statistics

There are two complementary approaches in statistics with the same objective of dealing with outliers: one is robust statistics and the other is diagnostic statistics.

2.2.1 Outliers

People in different scientific disciplines including statistics, computer vision, data mining, pattern recognition, machine learning, photogrammetry and remote sensing define the term 'outlier detection' in many ways using many names e.g. anomaly detection, fault detection, novelty detection, exception mining and one-class classification (Hawkins, 1980; Worden, 1997; Knorr and Ng, 1998; Breuning et al., 2000; Rousseeuw and Leroy, 2003; Hodges and Austin, 2004; Meer, 2004; Sotoodeh, 2006; Kanamori et al., 2009; Schubert et al., 2014). The majority of published research dealing with outliers is in statistics (Hawkins, 1980; Chatterjee and Hadi, 1988; Barnett and Lewis, 1995). Although, there is no general definition of outliers, a good answer of what are outliers and what are the problems of outliers can be found in statistics:

> "In almost every true series of observations, some are found, which differ so much from the others as to indicate some abnormal source of error not contemplated in the theoretical discussions, and the introduction of which into the investigations can only serve ... to perplex and mislead the enquirer" (Barnett and Lewis, 1995).

It is impractical to imagine point cloud data without outliers. Outliers in point cloud data occur for various reasons. The physical limitations of the sensors, boundaries between 3D features, occlusions, moving objects which pass through the scan area faster than they can be captured, multiple reflectance and noise can produce off-surface points that appear to be outliers (Sithole and Vosselman, 2004; Sotoodeh, 2006; Leslar et al., 2010).

2.2.2 Robustness and Robust Statistics

The field of mathematical statistics called robust statistics (Ševljakov and Vilčevskij, 2002). Box (1953) first introduced the technical terms 'robustness' and 'robust' (strong, sturdy), and the subject matter was recognised as a legitimate topic in the mid-sixties, due to the pioneering work of Tukey (1960), Huber (1964) and Hampel (1968). The first monograph is 'Robust Statistics' (Huber, 1981). It is true that the assumptions such as normality, linearity and independence that are commonly made in statistics do not always hold. Some of the most common statistical procedures are excessively sensitivity to seemingly minor deviations from the assumptions. One reason is the occurrence of gross errors, which usually show up as outliers and are dangerous for many statistical procedures. Other reasons behind deviations from initialized model assumptions include the empirical character of many models and the approximate character of many theoretical models. The classical parametric statistics can derive optimal procedures under exact parametric models, but say nothing about their behaviours (e.g. stability) when the models are only approximately valid. In this regards, robust statistics tries to investigate the two complementary characteristics 'optimality' and 'stability' in the same study. The basic philosophy of robust statistics is to produce statistical procedures which are stable w.r.t. small changes in the data or model and even large changes should not cause a complete breakdown of the procedures (Davies and Gather, 2004). Usually, when the sample size increases the sampling variability decreases. Large datasets such as mobile mapping point clouds may have very small variance and error may occur due to systematic bias of model misspecification. Agostinelli et al. (2007) mentioned that two problems may arise when dealing with large and complex data by classical statistical methods: (i) it may not be easy to fit simple and parsimonious models that reflect equally well all the data, and (ii) the sampling variability for such large datasets can be very small, to the extent that, the possible model misspecification bias dominates the statistical error, and may put into question the validity of the analysis. Robust statistical methods can deal with the above two challenges in the presence of outliers in a dataset. That means robust methods are able to manage the situations when outliers and regular observation do not follow the same model.

In robust statistics, first a model is fitted that considers the underlying pattern of the majority of the data and then outliers are determined having the largest deviations (e.g. residuals) from the fit of the majority. Therefore, in robust statistics, methods are developed that should be resilient or not much influenced by outliers. Robustness of an estimator is generally measured by the Breakdown Point (BP), Influence Function (IF) and continuity of the descriptive measure.

The breakdown point of an estimator is the smallest fraction of outlier contamination that can cause an estimator to be arbitrarily far from the real estimate. It is a global measure of robustness. The BP characterizes the maximal deviation (in the sense of metric chosen) from the ideal model F_0 that provides the boundness of the estimator bias. The BP of a functional T at a distribution F as applied to the Huber (1981) supermodel or gross-error model is defined as:

$$\varepsilon^*(T, F) = \sup_{\varepsilon<1}\{\varepsilon : \ \sup_{F:F=(1-\varepsilon)F_0+\varepsilon H}|T(F) - T(F_0)| < \propto\}, \tag{2.11}$$

where H is an arbitrary continuous distribution, ε is the probability of gross errors in the data, and $|T((1-\varepsilon)F_0+\varepsilon H)-T(F_0)|$ is the maximum bias (Hampel et al., 1986; Ševljakov and Vilčevskij, 2002). This notion defines the largest fraction of gross errors that still keeps the bias bounded. The two most important sample based classifications of breakdown point are: ‘additional breakdown point’ and ‘replacement breakdown point’ (Hampel et al., 1986). Care is needed concerning the usual criteria such as consistency of the estimator when a breakdown point is considered. Any location estimator should be equivariant when the data are multiplied by a constant and when a constant is added to them (Rousseeuw, 1991):

$$T(\{cx_1 + d, \ldots, cx_n + d\}) = cT(\{x_1, \ldots, x_n\}) + d. \tag{2.12}$$

In addition, a scatter estimator should be equivariant in the sense that (Rousseeuw, 1991):

$$S(\{cx_1 + d, \ldots, cx_n + d\}) = |c|S(\{x_1, \ldots, x_n\}). \tag{2.13}$$

The absolute value is taken in Eq. (2.13) because a scale estimate is always positive. The BP of the average or mean of a sample $\{x_1, x_2, \ldots, x_n\}$ is $1/n$ because a single observation changed by a large value can make the average value arbitrarily large. However, the sample median contains the possible BP at 50%. The median value may be changed if at least half of the observations are changed by the outliers in order to be certain that the middle observation is among them.

In contrast to the BP, the IF (Figure 2.3) is a local measure which measures the effect of one outlier on the estimator (Donoho and Huber, 1993; Hampel et al., 1986). The IF of a functional T at a distribution F measures the relative changes in the value of the functional caused by the addition of a small proportion ε of spurious observations at x. The IF of T at F can be defined as:

$$\text{IF}(x, T, F) = lim_{\varepsilon \downarrow 0} \frac{T((1-\varepsilon)F + \varepsilon\Delta_x) - T(F)}{\varepsilon}, \tag{2.14}$$

for those x, where the limit exists, and Δ_x denotes the point mass at the point x (Hampel et al., 1986; Croux and Dehon, 2013). A robust estimator should have a bounded IF (Figure 2.3). The reader is referred to (Huber, 1981), (Hampel et al., 1986), (Ševljakov and Vilčevskij, 2002), (Rousseeuw and Leroy, 2003), (Maronna et al., 2006) and (Becker et al., 2013) for more about robustness measures and robust statistics.

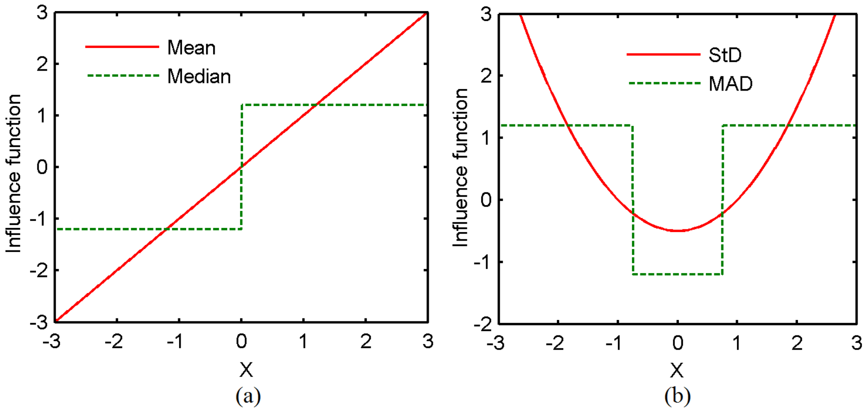


Figure 2.3 Influence functions for: (a) location: mean and median, (b) scatter: StD and MAD.

To explore the influence of outliers on classical and robust estimators, we illustrate a simple example. We have five sample values: 5.56, 5.25, 5.38, 5.86 and 5.43 of a variable x. To estimate the true value of x, and the variations among the values, we usually think about the sample mean:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.15}$$

and the standard deviation (StD):

$$\text{StD} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}. \tag{2.16}$$

We get $\bar{x} = 5.496$ and $\text{StD} = 0.232$ for the five sample points. If the first sample point is wrongly observed as 55.6 then we get $\bar{x} = 15.504$ and $\text{StD} = 22.416$, which are far from the true values. We could consider another location measure e.g. the median defined as the middle most observation of the sorted values. We calculate the median for both cases of the original dataset and contaminated dataset (after introducing the wrong value). Results are in Table 2.1. To find the median, the arrangements in ascending order are for: (i) original dataset: $5.25 \leq 5.38 \leq 5.43 \leq 5.56 \leq 5.86$, and for (ii) contaminated dataset: $5.25 \leq 5.38 \leq 5.43 \leq 5.86 \leq 55.6$. We get the median (middle observation; or third in the sorted values) as 5.43 for both datasets. We say the median is a robust estimator since in spite of changing one value to be an outlying (unusual) value, the median was unchanged, whereas the mean is sensitive to outliers, it changes from 5.496 to 15.504. In the case of StD, we get the values 0.232 and 22.416 for the real and contaminated datasets respectively. That means StD is non-robust and extremely sensitive to outliers. Therefore the BP of the mean and StD is merely $1/n$, which tends to 0 if n tends to a large value. A well-known robust alternative of StD is the Median Absolute Deviation (MAD) defined as:

$$\text{MAD} = a \times \underset{i}{median}|x_i - \underset{j}{median}(x_j)|, \tag{2.17}$$

where $a = 1.4826$ is a correction factor used to make the estimator consistent with the usual scale parameter of Gaussian distributions (Rousseeuw and Croux, 1993). We calculate the MAD values and get 0.193 and 0.267 for the real and contaminated samples respectively. The changed result is quite reasonable in the presence of the outlying case 55.6. Both the sample median and MAD have 50% BP. There are many scale estimators. The two alternatives of MAD are the Q_n and S_n estimators (Croux and Rousseeuw, 1992; Rousseeuw and Croux, 1993) defined as:

$$Q_n = c \cdot c_f \{|x_i - x_j|; i < j\}_{(k)}, \tag{2.18}$$

$$S_n = c \cdot c_f \cdot \underset{i}{median}\{\underset{j}{median}|x_i - x_j|\}, \tag{2.19}$$

where c_f is the consistency factor that depends on the data size, which makes the estimator unbiased, c is the constant factor (for $Q_n, c = 2.2219$ and for $S_n, c = 1.1926$). The variables $k = \binom{h}{2} \approx \binom{h}{2}/4$ and $h = \left[\frac{n}{2}\right] + 1$. The factor $\left[\frac{n}{2}\right]$ denotes the largest integer smaller than or equal to $\frac{n}{2}$. Both Q_n and S_n have the same BP of 50%. The Inter Quartile Range (IQR) is another robust scatter with BP 25%, which is the difference between the 3$^{\text{rd}}$ and 1$^{\text{st}}$ quartile of the data. It is commonly used in the boxplot to show variations in data (Section 2.8).

Table 2.1 Classical and robust location and scatter values.

	Dataset	Mean	Median	StD	MAD
Original	5.56, 5.25, 5.38, 5.86, 5.43	5.496	5.430	0.232	0.193
Contaminated	**55.6**, 5.25, 5.38, 5.86, 5.43	15.504	5.430	22.416	0.267
Without outlier	5.25, 5.38, 5.86, 5.43	5.480	5.405	0.264	0.190

2.2.3 Diagnostic Statistics

Diagnostic statistics has taken a different view from robust statistics. Rather than modifying the fitting method, diagnostics condition on the fit using standard methods to attempt to diagnose incorrect assumptions, allowing the analyst to modify them and refit under the new set of assumptions (Stahel and Weisberg, 1991). Rousseeuw and Leroy (2003) pointed out that the purpose of robustness is to safeguard against deviation from the assumptions, and the purpose of diagnostics is to find and identify deviation from the assumptions. It means that each views the same problem to achieve the same goal from opposite directions. In diagnostic statistics, first the outliers are identified, deleted or refitted (if necessary) and then the classical methods are performed on the cleaned (outlier-free) or modified data. Fung (1993) expressed the necessity of each other and pointed out that robust and diagnostic methods do not have to be competing but the complementary use of highly robust estimators and diagnostic measures provides a very good way to detect multiple outliers and leverage points. Hence, the goal of robust and diagnostic methods should be twofold: to identify outliers and to provide an analysis that has greatly reduced sensitivity to outliers (Ammann, 1993).

Figure 2.4 and Table 2.2 present an example to explore the necessity of outlier detection and help to understand the influence of outliers on model parameter estimation. We create 10 points in a 2D space. Nine of them (black points) form an almost linear pattern and one outlying (red point) case has a large difference mainly in the y direction. We fit the linear (straight line) model $y = \beta_0 + \beta_1 x$ using the LS method for the dataset with and without the outlier. Results in Table 2.2 show huge differences for the parameter values β_0 and β_1. For example, the value of $\beta_0 = 16.413$ for all the points and reduces to 0.497 when the outlying point is ignored in the fit. The β_1 values have opposite signs for the fits with and without outliers i.e. one has a negative gradient and the other has a positive gradient (excluding the outlying point) as shown in Figure 2.4.

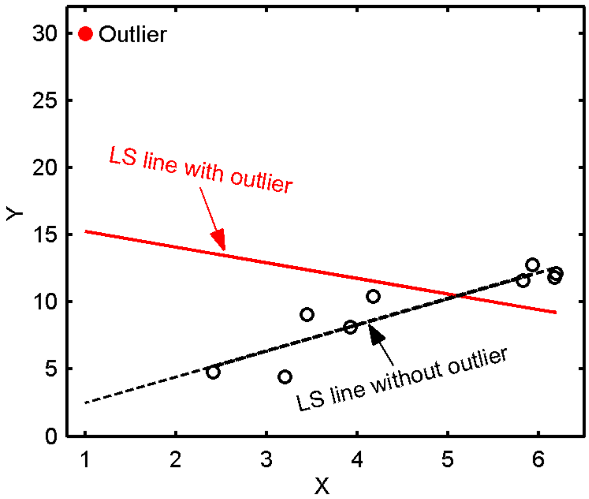


Figure 2.4 lines for LS with the outlier (red line) and without the outlier (black line).

Table 2.2 Classical and diagnostics model fitting.

Methods	Fitted line: $y = \beta_0 + \beta_1 x$	
	β_0	β_1
LS with outlier	16.413	-1.159
LS without outlier	0.497	1.954

2.3 Outlier Detection

This section details outlier detection approaches based on the well-known z-score (standard score) and robust estimators of multivariate location and scatter that can be used to get robust Mahalanobis (Mahalanobis, 1936) type distance.

2.3.1 Robust Outlier Detection

Outlier detection methods can be distinguished depending on the data dimension. One of the main ways is the distance-based approach that aims to find outliers by computing the distances of the points in a dataset from their majority (centre), and a point which is far from the bulk or centre of the data points is then treated as an outlier. A very simple method is the mean and standard deviation (StD) method based on Chebyshev's theorem (Barnett and Lewis, 1995). The method identifies an observation as an outlier if it falls outside the interval:

$$(\bar{x} - k\sigma, \bar{x} + k\sigma), \tag{2.20}$$

which is equivalent to:

$$\frac{|x_i - \bar{x}|}{\sigma_x} > k, \tag{2.21}$$

where $\bar{x}$ and σ are the mean and StD respectively. People often use $k = 1$, 2, or 3 in Eq. (2.20) to find outliers, because for the data which follows a Gaussian normal distribution, about 68%, 95%, and 99.7% among them lies within the intervals of 1, 2, and 3 StD respectively (Figure 2.5). The problem with this method is that it assumes that data is distributed as a Normal distribution, which is not always true. The z-score is a well-known measure of the distance deviated from the mean and normalized by the StD of the data. The method is also known as the so-called standard score, which is a distance-based measure that can be defined as a standardized residual:

$$z_i = \frac{|p_i - \bar{p}|}{\sigma_p} \ , \quad i = 1, \ldots, n \tag{2.22}$$

where $\bar{p}$ and σ_p are the centre (mean) and scatter (StD) of P.

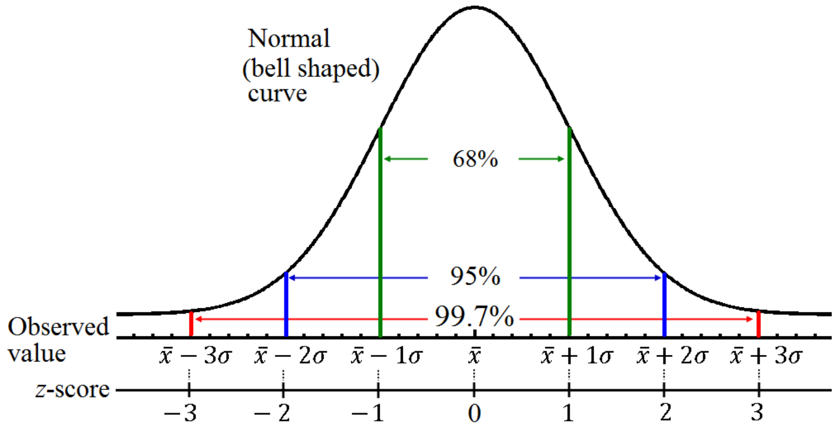


Figure 2.5 Normal (bell shaped) curve showing the proportion of observations within the $\bar{x} \pm k\sigma$ interval, and corresponding z-scores of the observations indicated on the z-score line.

Although the z-score is very simple and easy to compute, inclusion of classical mean and scatter makes its BP zero (Rousseeuw and Leroy, 2003). The most popular robust alternatives of the mean and StD are the median and the Median Absolute Deviation (MAD) respectively, both of which have the possible BP of 50% (Rousseeuw and Croux, 1993). To reduce the outlier sensitivity of the z-score, utilising the median and MAD (Eq. 2.17) in Eq. (2.22) can produces the robust z-score (Rz-score):

$$Rz_i = \frac{|p_i - \underset{j}{median}(p_j)|}{\text{MAD}}, \quad i = 1, \ldots, n \tag{2.23}$$

which is much more reliable and useful than the z-score for outlier contaminated samples. Observations with z_i or Rz_i values that exceed a certain cut-off (say 2.5 or 3.0) are usually considered as outliers (Rousseeuw and Croux, 1993). Other deletion diagnostics (Chatterjee and Hadi, 2012; Hadi and Simonoff, 1993), and robust approach based location and scatters, can be used as the alternatives of median and MAD in Eq. (2.23) to get variants of the robust z-score. Stahel (1981) and Donoho (1982) independently developed the so called outlyingness measure, obtained by low dimensional projection of the data. The outlyingness measure is defined as:

$$w_i = \underset{||v||=1}{max} \frac{|p_i v^T - \underset{j}{median}(p_j v^T)|}{\underset{i}{median}|p_i v^T - \underset{j}{median}(p_j v^T)|}, \tag{2.24}$$

where v is a direction and p_iv^T is the projection of the i^{th} point onto the v direction. The variable w_i is the maximum over all directions. Eq. (2.24) is similar to the robust z-score, where a projection p_iv^T is considered as an argument. The outlyingness measure is based on the idea that if a point is a multivariate outlier, then there must be some one dimensional projection of the data for which the point is a univariate outlier (Maronna and Yohai, 1995).

2.3.2 Robust Estimators of Location and Scatter

The estimation of multivariate location and scatter is one of the most difficult problems in robust statistics (Devlin et al., 1981; Huber, 1981; Stahel, 1981; Donoho, 1982; Rousseeuw, 1984; Hampel et al., 1986; Rousseeuw and van Zomeren, 1990; Rocke and Woodruff, 1996; Rousseeuw and Leroy, 2003; Maronna, 2005). In a multivariate setting, we can represent the dataset of n observations with m dimensions as a $P_{n\times m}$ matrix, $P = (p_1, \ldots, p_n)^T$, where the observation $p_i = (p_{i1}, \ldots, p_{im})$. The classical mean (or mean vector) and covariance matrix are the two well-known measures for the location and scatter of the data. The M-estimator (Maronna, 1976) is one of the most important robust estimators used to get robust location and scatter where M stands for Maximum Likelihood (ML). Huber (1964) was the first to devise the M-estimator and since then many versions of the M-estimator have been developed. The estimator robustifies the ML by down-weighting the extreme or outlying values using a weight function. It is computationally more efficient and robust than the ML Estimator (MLE). In the case of a univariate sample, a robust location M-estimator can be obtained as follows. Assume x_i is the i^{th} observation of a random variable x, T is a location estimator, the residual $r_i = x_i - T$ and a function of the residual is defined as $\rho(r)$. Then the M-estimate of T can be obtained by minimizing the sum of a function of the residuals, i.e.

$$\underset{T}{minimize} \sum_{i=1}^{n} \rho(r_i), \tag{2.25}$$

where $\rho(r_i)$ is continuous, positive definite i.e. $\rho(r) \geq 0$, symmetric i.e. $\rho(r) = \rho(-r)$, and generally with a unique minimum at 0, i.e. $\rho(r)$ increases as r increases from 0, but does not get too large. If $\rho = f$ (a probability density function) then the M-estimator becomes MLE. Differentiating w.r.t. the parameter T yields:

$$\sum_{i=1}^{n} \psi(r_i) = 0. \tag{2.26}$$

Different $\rho(r_i)$ and $\psi(r_i)$ combinations yield different M-estimators. Huber (1964) introduces the $\psi(r_i)$ as:

$$\psi(r) = \begin{cases} -k, & r < k \\ r, & -k \leq r \leq k \\ k, & r > k, \end{cases} \tag{2.27}$$

where k is the tuning constant. The M-estimator has many applications in different multivariate techniques e.g. robust regression (Fox, 2002). It has been used to develop the variants of well-known RANSAC algorithm in computer vision e.g. MSAC, MLESAC (Torr and Zisserman, 2000). The MSAC algorithm has been also used in photogrammetry and remote sensing (Vosselman and Maas, 2010). The M-estimator based segmentation has also been used in data visualization. The M-estimator has a BP of at most $1/m$. The multivariate M-estimator has a low breakdown value because of the possible implosion of the estimated covariance matrix (Debruyne and Hubert, 2009). Some of the recent robust estimators of mean and covariance matrix are: GM-estimators, MM-estimators, S-estimators, τ-estimators, multivariate rank estimators and depth-based estimators (Davies, 1987; Lopuhaä, 1991; Donoho and Gasko, 1992; Kent and Tyler, 1996; Maronna and Yohai, 1998; Rousseeuw and Struyf, 1998; Visuri et al., 2000; Rousseeuw and Leroy, 2003; Rousseeuw et al., 2006). Stahel (1981) and Donoho (1982) stated one example of high-breakdown and affine equivariant multivariate location and scatter, known as an outlyingness-mean and covariance matrix (Debruyne and Hubert, 2009). This uses the so-called outlyingness measure w_i in Eq. (2.24) by looking at all univariate projections of the data. For this estimator, robust distances are computed via the projection computation and the distances are used in the weight function. The resulting weighted-mean and covariance matrix are defined as:

$$\bar{p}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}, \tag{2.28}$$

$$\Sigma_w(p) = \frac{\sum_{i=1}^{n} w_i (p_i - \bar{p}_w)(p_i - \bar{p}_w)^T}{\sum_{i=1}^{n} w_i^{-1}}. \tag{2.29}$$

This procedure is affine equivariant and attains a 50% asymptotic BP when $n > 2m + 1$ (Donoho, 1982).

The Minimum Volume Ellipsoid (MVE) is a high breakdown robust estimator introduced by Rousseeuw (1984) that looks for an ellipsoid with the smallest volume that covers a subset of h non-contaminated data points, where $\frac{n}{2} \leq h < n$. The MVE has BP of $(n - h)/n$, and zero efficiency due to its low rate of convergence. Rousseeuw (1984) also introduced the Minimum Covariance Determinant (MCD) that has been recognized as a better estimator than MVE. The MCD finds h observations whose covariance matrix has the smallest determinant. It has several theoretical advantages: (i) better statistical efficiency because it is asymptotically normal, (ii) better accuracy, (iii) has a bounded influence function (Hubert et al., 2008), and (iv) attains a BP of 50%, when $h = \lfloor (n + m + 1)/2 \rfloor$ (Rousseeuw and Driessen, 1999). In addition, the MCD is affine equivariant that makes the analysis independent of the measurement scales of the variables (Hubert et al., 2012). Butler et al. (1993) showed the consistency and asymptotic normality of the MCD algorithm. Consistency is evident when the sampling distribution of the estimator becomes increasingly concentrated at the true parameter value as the sample size increases. Rousseeuw and Driessen (1999) pointed out that robust distances based on the MCD are more precise than those based on the MVE and hence better suited to expose multivariate outliers. In spite of its many advantages, it has been rarely used because it is computationally intensive. MCD works as the foundation of the Fast-MCD (Rousseeuw and Driessen, 1999) and Deterministic MCD (Hubert et al., 2012).

The Fast-MCD (FMCD) is a fast resampling algorithm developed by Rousseeuw and Driessen (1999) to efficiently estimate the MCD estimator. It can handle tens of thousands of points. The key component is the *C-step*. For each *C-step*, the Mahalanobis distances (Mahalanobis, 1936) are calculated and sorted in increasing order and the h points having the least Mahalanobis distances are selected. Then the mean and covariance are computed from the selected optimal set of h points. Finally, the Mahalanobis distances are calculated for all the points using the mean and covariance matrix. The algorithm starts by drawing random initial $(m + 1)$-subsets and performs the *C-step* on them, yielding consecutive h-subsets with decreasing determinant of the covariance matrix. To get an outlier-free initial subset of size $m + 1$, many (by default 500) initial random subsets need to be drawn, which is computationally intensive. For minimizing computational time only two *C-steps* are applied to each initial subset. FMCD uses selective iteration and nested extensions (when n is large, say $n > 600$) as two further steps to minimize its time. It then keeps the 10 results with the lowest determinant. From these 10 subsets, *C-steps* are performed until convergence to get the final h-subset. This h-subset is later used for determining the FMCD based robust mean vector and covariance matrix. An important advantage of the FMCD algorithm is that it allows for exact fit situations, that is, when h or more observation lie on a hyperplane (Rousseeuw and Driessen, 1999). The overall computation time for the FMCD algorithm is roughly proportional to the number of initial subsets. The reader is referred to Rousseeuw and Driessen (1999) for more details about the FMCD.

Hubert et al. (2012) introduced a Deterministic algorithm for the MCD (called DetMCD) to get robust location and scatter, which significantly reduces the number of *C-steps* run in the MCD and hence the computation time. FMCD needs to draw many random $(m + 1)$-subsets to obtain at least one outlier-free subset, whereas DetMCD starts from a few easily computed h-subsets and then runs *C-step* until convergence. It uses the same iteration step but does not draw a random subset. Rather it starts from only a few well-chosen initial estimators followed by the *C-steps*. DetMCD couples aspects of both the FMCD and the Orthogonalized Gnanadesikan and Kettenring (OGK) estimators (Maronna and Zamar, 2002). The DetMCD algorithm standardizes each variable by subtracting its coordinate wise median and dividing by its scale Q_n (Rousseeuw and Croux, 1993), this standardization makes the algorithm location and scale invariant. A new matrix S is generated based on the standardized observations with rows s_i^T $(i = 1, \ldots, n)$ and columns S_j $(j = 1, \ldots, m)$. Next, six initial estimates of the mean and covariance matrix $\hat{c}_k(S)$ and $\hat{\Sigma}_k(S)$ $(k = 1, \ldots, 6)$ are constructed. The six initial estimates are based on six different preliminary estimates that are found in a deterministic way (Hubert et al., 2012), i.e. without random sampling. Since the preliminary estimates Σ_k may have imprecise eigenvalues, the initial estimates for the mean and covariance matrix are computed by the following steps inspired by the following portion of the OGK algorithm (Maronna and Zamar, 2002):

- Compute the matrix E of eigenvectors of Σ_k and put $V = SE$.
- Estimate the covariance of S by $\hat{\Sigma}_k(S) = ELE^T$, where $L = diag(Q_n^2(V_1), \ldots, Q_n^2(V_m))$.
- The centre of S is estimated by first sphering the data, applying the coordinate wise median to the sphered data and transforming back, so $\hat{c}_k(S) = \hat{\Sigma}_k^{1/2}(med(S\hat{\Sigma}_k^{-1/2}))$.

For all six estimates of mean and covariance matrix, the statistical distances have been computed:

$$d_{ik} = D(s_i, \hat{c}_k(S), \hat{\Sigma}_k(S)), \tag{2.30}$$

which is a Mahalanobis type distance with the parameters in the brackets. For each initial estimate $(\hat{c}_k(S), \hat{\Sigma}_k(S))$, $k = 1, \ldots, 6$; the $h = \lceil n/2 \rceil$ observations with the smallest distance d_{ik} has been selected and then the *C-step* is applied until convergence. Results are six fully refined estimates of mean and covariance matrix, where the one with the smallest determinant is the raw DetMCD. Then a reweighting step is applied to get the final DetMCD. This algorithm is permutation invariant (the result does not depend on the order of the data) and is almost affine equivariant, whereas FMCD is not permutation invariant. Hubert et al. (2012) claimed and showed through simulation that DetMCD is much faster than FMCD and at least as robust as FMCD. The reader is referred to Hubert et al. (2012) and Hubert et al. (2014) for more information about the DetMCD algorithm.

2.3.3 Robust Distance

One of the most general techniques for the identification of an outlier in univariate data is: is its distance far away from the bulk (centre) of the data? However, for multivariate data, the distance of an observation from the centre of the data is not sufficient for outlier detection; the shape of the data has to be considered together with the location of the centre (Rousseeuw and van Zomeren, 1990; Rousseeuw and Leroy, 2003). The covariance matrix can be used to quantify the shape and size of the multivariate data. Mahalanobis Distance (MD) (Mahalanobis, 1936) is a well-known distance measure that considers covariance as well as the location of the centre. For a m-dimensional multivariate (m variate) sample P of size n, MD is defined as:

$$\mathrm{MD}_i = \sqrt{(p_i - c)^T \Sigma^{-1} (p_i - c)}, \quad i = 1, \ldots, n \tag{2.31}$$

where c is the estimated centre and Σ is the covariance matrix of the sample. Although it is still quite easy to detect a single outlier by means of MD, this approach no longer suffices for multiple outliers because of the masking effect (Rousseeuw and Driessen, 1999). Masking occurs when an outlying subset goes undetected because of the presence of another, usually adjacent, subset (Hadi and Simonoff, 1993). Hampel et al. (1986) pointed out that the MD is not robust because of the sensitivity of the mean and covariance matrix to outliers. It is necessary to use a distance that is based on robust estimators of multivariate location and scatter (Rousseeuw and Leroy, 2003).

Many authors use robust estimators (described in the previous section) to get a robust mean and covariance matrix and use them in Eq. (2.31) to obtain a robust version of MD, simply called Robust Distance (RD). Campbell (1980) proposed a robust distance by inserting M-estimators for c in Eq. (2.31). Unfortunately, the M-estimator has a low breakdown point and it goes down when there are more coordinates in which outliers can occur (Hampel et al., 1986). Rousseeuw and van Zomeren (1990) introduced RD based on MVE (Rousseeuw, 1984), defined as:

$$\mathrm{RD}_i = \sqrt{(p_i - c_{\mathrm{MVE}})^T \Sigma_{\mathrm{MVE}}^{-1} (p_i - c_{\mathrm{MVE}})}, \quad i = 1, \ldots, n \tag{2.32}$$

where c_{MVE} is the MVE estimate of location (average of the final h points from MVE) and Σ_{MVE} is the covariance matrix of the h points. Rousseeuw and van Zomeren (1990) mentioned that there is an opportunity to use the MCD based mean and covariance matrix in place of the MVE based mean and covariance matrix. The MCD based robust distance can then be defined as:

$$\mathrm{RD}_i = \sqrt{(p_i - c_{\mathrm{MCD}})^T \Sigma_{\mathrm{MCD}}^{-1} (p_i - c_{\mathrm{MCD}})}, \quad i = 1, \ldots, n. \tag{2.33}$$

Since MCD estimators need more computation time, MCD based RD was not popular until the Fast-MCD (Rousseeuw and Driessen, 1999) was introduced. Rousseeuw and Driessen (1999) showed that RD follows a Chi-square (χ^2) distribution with m (the number of variables) degrees of freedom, and the observations that have MD or RD values of more than $\sqrt{\chi^2_{m,0.975}}$ are identified as outliers.

2.4 Robust Principal Component Analysis

The robust version of PCA (RPCA) is for determining the PCs (eigenvectors) that are expected not to be influenced by outliers. Much research has been carried out on robustifying PCA over the years (Croux and Haesbroeck, 2000; Hubert et al., 2005; Candés et al., 2011; Feng et al., 2012). Existing methods can be categorized according to the dimensionality of the data. Some are appropriate for high dimensional data (Xu et al., 2010; Candés et al., 2011; Feng et al., 2012) and some are better for low-dimensional data (Croux and Ruiz-Gazen, 2005; Hubert et al., 2005). We are concerned with 3D point cloud data, where the number of dimensions is considerably smaller than the number of observations or points. Hence we are interested in an efficient method for low-dimensional data. Roughly they can be categorized into two potential methods: (i) those that try to find a robust estimation of the covariance matrix, and (ii) those based on Projection Pursuit (PP), (Friedman and Tukey, 1974) such as by Li and Chen (1985) and Hubert et al. (2002) that try to maximize certain robust estimates of univariate variance to obtain consecutive directions on which the data are projected. Covariance matrix based methods are limited in the case of insufficient data to robustly estimate a high-dimensional covariance matrix and the PP based methods are qualitatively robust and inherit the robustness characteristics of the adopted estimators (Feng et al., 2012). A first group of robust methods use a robust estimator of covariance matrix like M-estimators instead of the classic covariance matrix (Maronna, 1976; Campbell, 1980). Croux and Haesbroeck (2000) suggested using high-breakdown robust estimators such as the MCD to derive the covariance matrix. Croux and Ruiz-Gazen (1996) proposed robust PCA in which PCs are defined as projections of the data onto directions maximizing the robust scale Q_n. The spherical PCA and elliptical PCA are also proposed as the robust PCA in Locantore et al. (1999). Another way of getting robust PCA is to replace the LS cost function by a robust cost function such as the Least Trimmed Square (LTS) estimator (Rousseeuw, 1984; Rousseeuw and Leroy, 2003) or an M-estimator (Maronna, 2005).

Hubert et al. (2005) proposed a version of robust PCA, that combined the ideas of using the robust estimator of the covariance matrix and the PP to take advantages from both the approaches. In this thesis, we choose this method because it yields accurate estimates of outlier-free datasets and more robust estimates for contaminated data, is able to detect exact-fit situations and has the further advantage of outlier diagnostics and classification (Hubert et al., 2005), all of which are beneficial to our purpose. The RPCA (Hubert et al., 2005) involves the following steps. First, the data are pre-processed to make sure that the transformed data are lying in a subspace whose dimension m is less than the number of observations n without loss of information. Reducing the data space to the affine subspace spanned by the n observations is especially useful when $m \geq n$, but even when $m < n$, the observations may span less than the whole m-dimensional space (Hubert et al., 2005). A useful way for reducing the data space is by using the SVD of the mean-centred data matrix. Second, the h points, where $n/2 < h < n$, i.e. the 'least outlying' data points are identified, and a measure of outlyingness is computed by projecting all the data points onto many univariate directions, each of which passes through two individual data points. In order to keep the computation time down, the data set is compressed to PCs defining potential directions. Then, every direction for a point p_i is scored by its corresponding value of outlyingness (Stahel, 1981; Donoho, 1982):

$$w_i = \underset{v}{argmax} \frac{|p_i v^T - c_{\text{MCD}}(p_i v^T)|}{\Sigma_{\text{MCD}}(p_i v^T)}, \quad i = 1, \dots, n \tag{2.34}$$

where $p_i v^T$ denotes a projection of the i^{th} observation onto the v direction, and c_{MCD} and Σ_{MCD} are the MCD based mean and scatter (covariance matrix) on an univariate direction v respectively. The FMCD estimators are used as the robust estimators of the mean and scatter in Eq. (2.34). In the next step, an assumed h ($h > n/2$) portion of observations with the smallest outlyingness values are used to construct a robust covariance matrix Σ_h. The larger h can give a more accurate RPCA but a smaller h is better for more robust results. Users can fix it according to their own objectives and from knowledge of their particular data. We use $h = \lceil 0.5 \times n \rceil$ in our algorithms. Then, the method projects the observations onto the d dimensional subspace spanned by the d largest eigenvectors of Σ_h, and computes mean and the covariance matrix by means of the reweighted MCD estimator, with weights based on the robust distance of every point. The eigenvectors of this covariance matrix from the reweighted observations are the final robust PCs and the MCD mean serves as a robust mean. The resulting robust PCA is location and orthogonal invariant.

An extra advantage of the RPCA algorithm (Hubert et al., 2005) is that it can identify outliers. There are two types of outliers. One type is the orthogonal outlier that lies away from the subspace spanned by the first d (in our case two) PCs (for a plane) and is identified by using Orthogonal Distance (OD), which is the distance between the observation p_i and its projection $\hat{p}_i$ in the d-dimensional PCA subspace. For p_i it is defined as:

$$\mathrm{OD}_i = ||p_i - \hat{p}_i|| = ||p_i - \hat{\mu}_p - Lt_i^T||, \quad i = 1, \ldots, n \tag{2.35}$$

where $\hat{\mu}_p$ is the robust centre of the data, L is the robust loading (PC) matrix, which contains robust PCs as the columns in the matrix, and $t_i = (p_i - \hat{\mu}_p)L$ is the i^{th} robust score. The other type of outlier is identified by the Score Distance (SD) that is measured within the PCA subspace and is defined as:

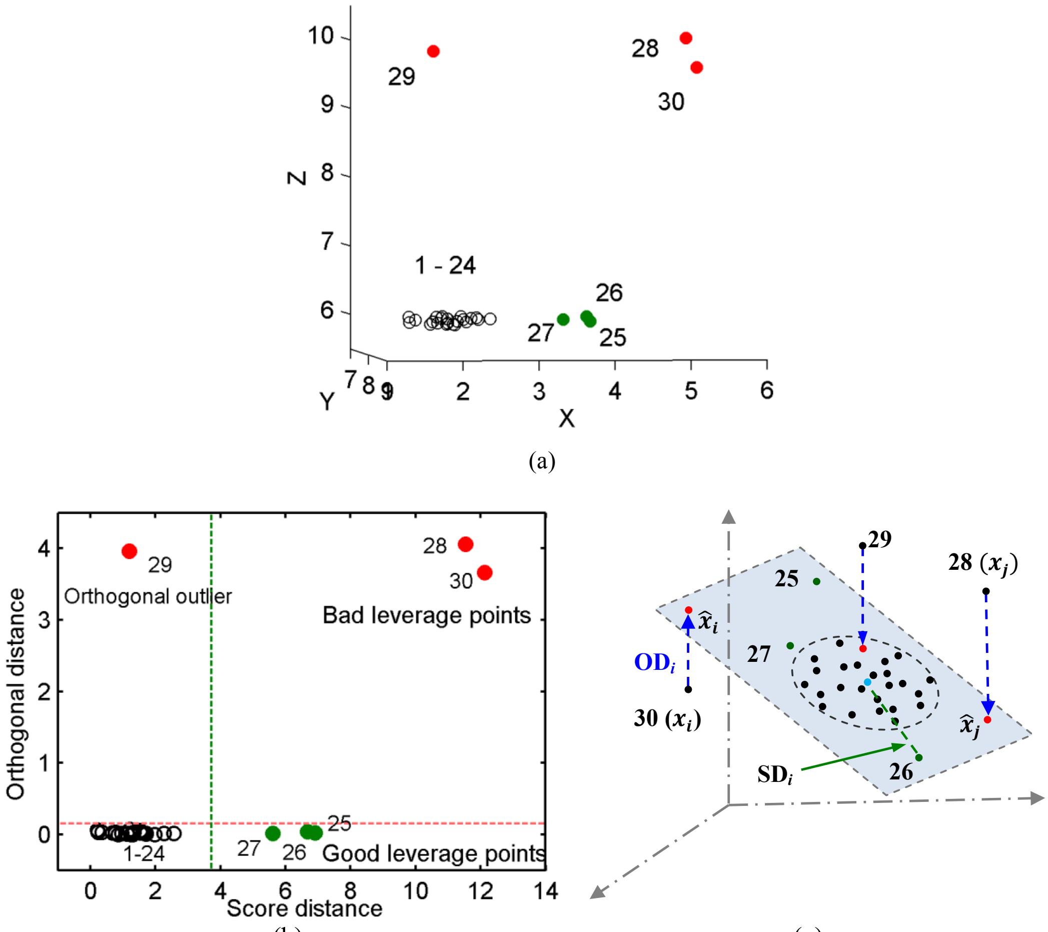
$$\mathrm{SD}_i = \sqrt{\sum_{j=1}^{d} (t_{ij}^2 / l_j)}, \quad i = 1, \ldots, n \tag{2.36}$$

where l_j is the j^{th} eigenvalue of the robust covariance matrix Σ_{MCD} and t_{ij} is the ij^{th} element of the score matrix:

$$\mathrm{T}_{n,d} = (P_{n,m} - 1_n c_{\mathrm{MCD}}) L_{m,d}, \tag{2.37}$$

where $P_{n,m}$ is the data matrix, 1_n is the column vector with all n components equal to 1, c_{MCD} is the robust centre, and $L_{m,d}$ is the matrix constructed by the robust PCs. OD and SD are sketched in Figure 2.6b. The cut-off value for the score distance is $\sqrt{\chi^2_{d,0.975}}$, and for the orthogonal distance is a scaled version of χ^2. A scaled version of χ^2 is a version of χ^2 $(g_1 \chi^2_{g_2})$, which gives a good approximation of the unknown distribution of the squared ODs (Box, 1954), where g_1 and g_2 are two parameters estimated by the method of moments (Nomikos and MacGregor, 1995). The reader is referred to Hubert et al. (2005) for more information about the RPCA algorithm.

In Figures 2.6(b and c), we illustrate the orthogonal and score outliers based on 30 3D artificial points (Figure 2.6a) including 6 outliers projected onto the fitted plane in Figure 2.6c. The points 25, 26 and 27 marked as green points in Figure 2.6c are essentially in the plane as their orthogonal distances are low although they are distant from the mean in the plane (score distance). In Figure 2.6b, they are identified as good leverage points. Points 28, 29 and 30 (red points) exceed the cut-off value of orthogonal distance so are treated as orthogonal outliers. Projecting these points into the plane show their score distances. Note that point 29 has a low score distance so would not be identified as an outlier without the orthogonal distance. In Figure 2.6c, the points 28 and 30 have large orthogonal and score distances and are treated as bad leverage points as shown in Figure 2.6b.



(a)

(b)

(c)

Figure 2.6 (a) Scatter plot of the data, outlier detection: (b) diagnostic plot of orthogonal distance versus score distance, and (c) fitted plane. Green points are distant in terms of score and red points are orthogonal outliers.

2.5 RANdom SAmple Consensus (RANSAC)

Fischler and Bolles (1981) introduced RANdom SAmple Consensus (RANSAC), which is a model-based robust approach used in many applications for extracting shapes and estimating the parameters of a model from data that may contain outliers. The RANSAC algorithm is possibly the most widely used robust estimator in the field of computer vision that is also often used in laser scanning data analysis for planar surface detecting, fitting, extraction and normal estimation. Deschaud and Goulette (2010) showed that RANSAC is very efficient at detecting large planes in noisy point clouds. Due to its ability to tolerate large fraction of outliers, the algorithm is a popular choice for a variety of robust estimation problems (Raguram et al., 2008). Depending on the complexity of the model, RANSAC can handle contamination levels of more than 50%, which is a common limit in robust statistics (Matas and Chum, 2004). RANSAC classifies data into inliers and outliers by using the LS cost function with maximum support (the number of data points that match with the model). It consists of two steps: hypothesize and test. First, a subset is randomly sampled and the required model parameters are estimated based on the subset. The size of the subset should be minimal i.e. the size (e.g. three points for a plane) of the random subset is the smallest needed to estimate the model parameters. In the second step, the model is compared with the data and its support is determined. This two-step iterative process continues until the likelihood of getting a model with better support than the current best model is lower than a given threshold (typically 1%–5%). Although usually the LS cost function is used in RANSAC, various cost functions have been used in this algorithm (Torr and Zisserman, 2000). RANSAC is popular for planar surface fitting because it is conceptually simple yet powerful and very general (Schnabel et al., 2007). Since its inception, many versions of RANSAC have been proposed to increase its efficiency (Torr and Zisserman, 2000; Matas and Chum, 2004; Raguram et al., 2008; Choi et al., 2009). Some of the methods try to optimize the process of model verification, while others try to modify the sampling process in order to preferentially generate a more useful hypothesis (Raguram et al., 2008). Literature shows there is no consensus as to which one is the best for every real-time situation of model fitting. Matas and Chum (2004) pointed out that the speed of the RANSAC algorithm depends on two factors: (i) the level of outlier contamination that determines the number of random samples that have to be taken to guarantee a certain confidence in the optimality of the solution, and (ii) the time that is spent evaluating the quality of each of the hypothesized model parameters which is proportional to the size of the data.

RANSAC can be sensitive to the choice of the correct noise threshold T. It finds the minimum of the cost function:

$$C_f = \sum_i \rho(e_i^2), \tag{2.38}$$

where e_i is the error of the i^{th} observation, and

$$\rho(e^2) = \begin{cases} 0 & \text{for} \quad e^2 < T, \\ constant & \text{for} \quad e^2 \geq T. \end{cases} \tag{2.39}$$

Torr and Zisserman (2000) showed that if T is set too high then the robust estimate can be very poor. To address this, they proposed MSAC (M-estimator SAmple Consensus), which minimizes the cost function in Eq. (2.38) with a robust error function ρ_2 defined as:

$$\rho_2(e^2) = \begin{cases} e^2 & \text{for} \quad e^2 < T \\ T^2 & \text{for} \quad e^2 \geq T \end{cases}, \tag{2.40}$$

which is the redescending M-estimator. The authors set $T = 1.96\sigma$ so that Gaussian inliers are only incorrectly rejected 5% of the time. In the same paper, Torr and Zisserman (2000) presented MLESAC (Maximum Likelihood Estimation SAmple Consensus), which adopts the same sampling strategy as in RANSAC to generate putative solutions, but chooses the solution that maximizes the likelihood rather than just the number of inliers. Vosselman and Klein (2010) investigated the importance of the MSAC algorithm for point cloud data analysis. Choi et al. (2009) evaluated the RANSAC family and showed that MSAC is one of the most accurate ones.

2.6 Outlier Detection in Data Mining, Pattern Recognition and Machine Learning

Besides statistics and computer vision, many efficient outlier detection approaches have been developed in several areas such as machine learning, pattern recognition and data mining, depending on application areas e.g. information systems, health care, network systems, news documentation, industrial machines, and video surveillance (Breuning et al., 2000; Hodges and Austin, 2004; Chandola, 2008; Chandola et al., 2009; Yang and Wang, 2006; Hido et al., 2011; Zimek et al., 2012; Aggarwal, 2013; Liu et al., 2013; Sugiyama and Borgwardt, 2013). Hodges and Austin (2004) stated that there is no single universally applicable or generic outlier detection approach. People are trying to develop more effective methods for their particular application area taking into account the characteristics of their data. We choose the following three algorithms that are based on different approaches, which have been recently proposed or are popular in the data mining, machine learning and pattern recognition literature.

2.6.1 Local Outlier Factor

Breuning et al. (2000) introduced the Local Outlier Factor (LOF) assuming that for many real-world datasets, there exist more complex data structures that can contain outliers relative to their local neighbourhoods. Since the LOF was introduced, there have been many variants of this algorithm developed, and it is considered to be accurate and efficient and has been frequently used for comparison with newly proposed methods (Kriegel et al., 2009; Schubert et al., 2014). Breuning et al. (2000) assigned a measure of being an outlier for each observation in a dataset called the LOF. The measure depends on how isolated an observation is w.r.t. the surrounding neighbourhood, particularly w.r.t. to the densities of the neighbourhood. To find the LOF for a point p_i, the algorithm uses three consecutive steps. First, the reachability distance of an observation p_i w.r.t. observation p_j is calculated as:

$$reach - dist_k(p_i, p_j) = max\{k - distance(p_j), d(p_i, p_j)\}, \tag{2.41}$$

where $k - distance(p_j)$ is the distance between p_j and its k^{th} neighbour, and $d(p_i, p_j)$ is the distance between p_i and p_j. The reachability distance of an observation to one of its neighbour is shown in Figure 2.7a. Second, the local reachability density for p_i is computed as:

$$lrd_{MinPts}(p_i) = \frac{1}{\frac{\sum_{p_j \in N_{MinPts}(p_i)} reach - dist_{MinPts}(p_i, p_j)}{|N_{MinPts}(p_i)|}}, \tag{2.42}$$

where the local reachability density is the inverse of the average reachability distance based on the $N_{MinPts}(p_i)$ nearest neighbours of p_i, which can be considered as the local neighbourhood of p_i. $|N_{MinPts}(p_i)|$ is the number of observations in the local neighbourhood. Figure 2.7b depicts the elements of local reachability density of p_i. Finally, the LOF of an observation is defined as the average of the ratio of the local reachability density of p_i and those of the $MinPts$-nearest neighbours to p_i, which is defined as:

$$\mathrm{LOF}_{MinPts}(p_i) = \frac{\sum_{p_j \in N_{MinPts}(p_i)} \frac{lrd_{MinPts}(p_j)}{lrd_{MinPts}(p_i)}}{|N_{MinPts}(p_i)|}. \tag{2.43}$$

A large LOF indicates that the observation p_i is a potential outlier. That means the density of all the neighbours of p_i is higher than the density of the p_i itself. Usually, outliers have larger LOF scores than a threshold in the range between 1.2 and 2.0, depending on the data (Goldstein, 2012). The reader is referred to Breuning et al. (2000) for more details about the LOF algorithm.

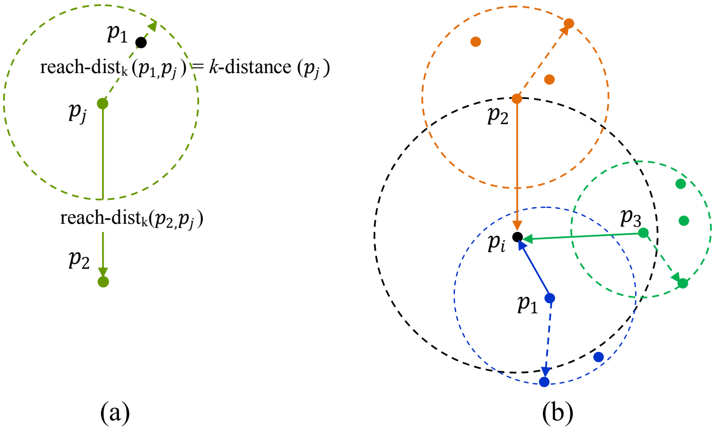

p_1
reach-dist$_k$ (p_1,p_j) = k-distance (p_j)
p_j
reach-dist$_k$$(p_2,p_j)$
p_2
(a)
p_2
p_3
p_i
p_1
(b)

Figure 2.7 Local outlier factor: (a) reachability distances of p_1 and p_2 to p_j, and (b) graphical representation of local reachability density for p_i with its nearest neighbours p_1, p_2 and p_3. Euclidean distances shown in solid lines and k-distances are shown in dotted lines, neighbourhood size $k = 3$, neighbourhood of the points are indicted by the coloured circles.

2.6.2 Direct Density Ratio Based Method

The density ratio based approach is one of the most well-known approaches in the statistical, machine learning and pattern recognition literature. It performs outlier detection using the ratio of the two probability density functions of the test and training datasets. The approach for identifying outliers in a test or validation dataset is based on a training or model dataset that only contains inlier data (Schölkopf et al., 2001; Kanamori et al., 2009). Density estimation is not trivial and getting an appropriate parametric model may not be possible. Therefore, Direct Density Ratio (DDR) estimation methods have been developed that do not require density estimation. Recently Hido et al. (2011) introduced an inlier based outlier detection method based on DDR estimation that calculates the importance or an inlier score defined as:

$$w(p) = \frac{p_{tr}(p)}{p_{te}(p)}, \tag{2.44}$$

where $p_{tr}(p)$ and $p_{te}(p)$ are the densities of identically and independently distributed (i.i.d.) training $\{p_j^{tr}\}_{j=1}^{n_{tr}}$ and test $\{p_i^{te}\}_{i=1}^{n_{te}}$ samples, respectively. It is plausible to consider observations with small inlier scores as outliers. Hido et al. (2011) used unconstrained Least Squares Importance Fitting (uLSIF), which originated from the idea of LSIF (Kanamori et al., 2009). In uLSIF, the closed-form solution is computed by solving a system of linear equations. The importance $w(p)$ in Eq. (2.44) is modelled as:

$$\hat{w}(p) = \sum_{l=1}^{b} \alpha_l \ \varphi_l(p), \tag{2.45}$$

where $\{\alpha_l\}_{l=1}^{b}$ are parameters and $\{\varphi_l(p)\}_{l=1}^{b}$ are basis functions such that $\varphi_l(p) \geq 0$. The parameters are determined by minimizing the following objective function:

$$\frac{1}{2} \int \left(\hat{w}(p) - \frac{p_{tr}(p)}{p_{te}(p)} \right)^2 p_{te}(p) \ dx. \tag{2.46}$$

The solution of uLSIF is computed through matrix inversion, and the leave-one-out-cross-validation score (Kanamori et al., 2009) for uLSIF is computed analytically. Hido et al. (2011) showed that the uLSIF is competitively accurate and computationally more efficient than the existing best methods e.g. OSVM (Schölkopf et al., 2001). The reader is referred to Hido et al. (2011) for further information about uLSIF.

2.6.3 Distance based Outlier Detection

Knorr and Ng (1998) first introduced the new paradigm of Distance Based (DB) outlier detection that generalises the statistical distribution based approaches. In contrast to statistical distribution based approaches, it does not need prior knowledge about the data distribution. In DB outlier detection, a point p is considered as an outlier w.r.t. parameters α, δ if at least a fraction α of the data has a distance from p larger than δ, that is:

$$|\{q \in P | d(p,q) > \delta\}| \geq \alpha n, \tag{2.47}$$

where $q \in P$, and $(\alpha, \delta) \in \mathbb{R}$; and $0 \leq \alpha \leq 1$ are the user defined parameters. But the problem is how to fix the distance threshold δ. Ramaswamy et al. (2000) proposed the k^{th} Nearest Neighbour (k^{th}NN) distance as a measure of outlyingness to overcome the limitation. The score of a point is defined as:

$$q_{k^{th}\mathrm{NN}}(p) := d^k(p; P), \tag{2.48}$$

where $d^k(p;P)$ is the distance between p and its k^{th}NN. This method is computationally intensive and Wu and Jermaine (2006) proposed a sampling algorithm to efficiently estimate the score in Eq. (2.48), defined as:

$$q_{k^{th}S_p}(p) := d^k(p, S_p(P)), \tag{2.49}$$

where $S_p(P)$ is a subset of P, which is randomly and iteratively sampled for each point in P. To save computation time without losing accuracy, recently Sugiyama and Borgwardt (2013) suggested sampling only once. They define the score as:

$$q_{S_p}(p) := \mathrm{min}_{q \in S_p} d(p,q), \tag{2.50}$$

where $\mathrm{min}_{q \in S_p} d(p,q)$ is the minimum distance between p and q, where q is a point in the subset S_p. Sugiyama and Borgwardt (2013) named the algorithm q_{S_p}, and stated that it outperforms state-of-the-art DB algorithms including Angle Based Outlier Factor (ABOF; Kriegel et al., 2008) and OSVM (Schölkopf et al., 2001) in terms of efficiency and effectiveness.

2.7 Regression Analysis

Regression analysis is one of the most important branches of multivariate statistical techniques that is routinely applied in most subjects including computer vision, data mining, machine learning, pattern recognition, photogrammetry and remote sensing (Wang and Suter, 2004; Subbarao and Meer, 2006; Bishop, 2006; Murphy, 2012; Nurunnabi and West, 2012; Nurunnabi et al., 2013b). It is appealing because it provides a conceptually simple method for investigating functional relationship among observed variables. For fitting a linear regression model, the LS method is traditionally used mainly because of its computational simplicity and for having some optimal properties under certain underlying assumptions (Chatterjee and Hadi, 1988, 2012; Nurunnabi et al., 2014b). Violation of these assumptions, and particularly the so-called implicit assumption that all observations are equally reliable and should have an equal role in determining the LS results and influencing the conclusions (Chatterjee and Hadi, 1988). It is known that usually outliers have unequal and may have extreme influence on the estimates (Rousseeuw and Leroy, 2003). A regression model defines a relation, such as: Y (dependent/response variable) is a function of X (explanatory/independent/regressor variables) and β (a vector of parameters of the model), i.e.

$$E(Y|X) = f(X, \beta). \tag{2.51}$$

The standard regression model is:

$$Y = X\beta + \epsilon, \tag{2.52}$$

where Y is a $n \times 1$ vector of response, X is a $n \times (m+1)$ full rank matrix of m explanatory variable(s) including one constant column of 1, β is a $(m+1) \times 1$ vector of unknown parameters and ϵ is a $n \times 1$ vector of i.i.d. random disturbances each of which follows a Gaussian normal distribution with mean zero and a constant variance σ^2. The LS method minimizes the error ϵ sum of squares to estimate the parameters, that means it minimizes:

$$S(\beta) = \sum_{i=1}^{n} \epsilon^2 = \epsilon^T \epsilon = (Y - X\beta)^T (Y - X\beta). \tag{2.53}$$

Therefore, LS estimators satisfy:

$$\frac{\delta S}{\delta \beta}|_{\beta} = 0, \tag{2.54}$$

$$\Rightarrow -2X^T Y + 2X^T X\hat{\beta} = 0, \tag{2.55}$$

$$\Rightarrow \hat{\beta} = (X^T X)^{-1} X^T Y. \tag{2.56}$$

The LS fit of the response is:

$$\hat{Y} = X\hat{\beta}. \tag{2.57}$$

The corresponding residual is defined as:

$$r = Y - \hat{Y} = Y - X\hat{\beta}. \tag{2.58}$$

It is known that the presence of outliers can substantially change the LS estimates and provides erroneous results and the wrong conclusions. Robust regression and regression diagnostics are two types of remedies that deal with outliers (Atkinson and Riani, 2000; Rousseeuw and Leroy, 2003; Nurunnabi et al., 2011; Chatterjee and Hadi, 2012; Nurunnabi et al., 2014b).

2.8 Boxplot

We use a number of visualisation techniques in this thesis to illustrate the results of the various algorithms. In particular, we use the boxplot (Tukey, 1977) as a graphical or qualitative performance measure because it has been recognized as a robust visualization and exploratory data analysis tool in many subjects including statistics, computer vision, machine learning, pattern recognition and remote sensing (Storer et al., 2010; Önskog et al., 2011; Nurunnabi et al., 2012a; Rexhepaj et al., 2013). It is especially useful when two or more sets of data are being compared. It can be used to explore the underlying data structure for a large amount of observations. In its simplest form, it shows five descriptive statistics in the same graph: (i) the minimum, (ii) the maximum, (iii) the lower/1st quartile Q_1 or 25th percentile, (iv) the upper/3rd quartile Q_3 or 75th percentile, and (v) the median (second quartile Q_2 or 50th percentile) value of a dataset. It appears as a rectangular box (Figure 2.8); a line is drawn across the box indicating the median value of the data, two 'whiskers' sprout from the two ends of the box and end at the positions where the dataset has minimum (lower adjacent) and maximum (upper adjacent) extreme values. The box length shows the variability of the data, and the position of the box w.r.t. its whiskers and the position of the line (median) in the box show whether the data is symmetric or skewed, either to the left (downward) or right (upward). The boxplot indicates observations which are far away from the bulk of the data, such as outliers and extreme observations. The points that are out of the reach of whiskers are treated as outliers, and within the length of whiskers are identified as extreme cases. The default value for the maximum length of whiskers for MATLAB® is $w = 1.5$, and points are classed as outliers if they are larger than $Q_3 + w(Q_3 - Q_1)$ or smaller than $Q_1 - w(Q_3 - Q_1)$. The default of 1.5 corresponds to approximately $\pm 2.7\sigma$ and 99.3% coverage if the data are normally distributed.

Figure 2.8 shows a boxplot together with the respective dotplot of a dataset of 25 points that contains regular points (grey), extreme points (blue) and outliers (red). Usually it is drawn vertically as shown. The boxplot can be used for exploring many samples at one time and many boxplots of several samples can be lined up alongside one another on a common scale and the various attributes of the samples compared at one time. Figure 2.9 shows four boxplots for four different datasets of 25 observations. Samples A and B appear to have similar median (centre) values, which exceed those of samples C and D. Samples A and B are reasonably symmetric, sample C contains two outliers (red asterisks) and sample D is skewed upward. To know more about boxplots, the reader is referred to Tukey (1977), McGill et al. (1978), and Velleman and Hoaglin (1981).

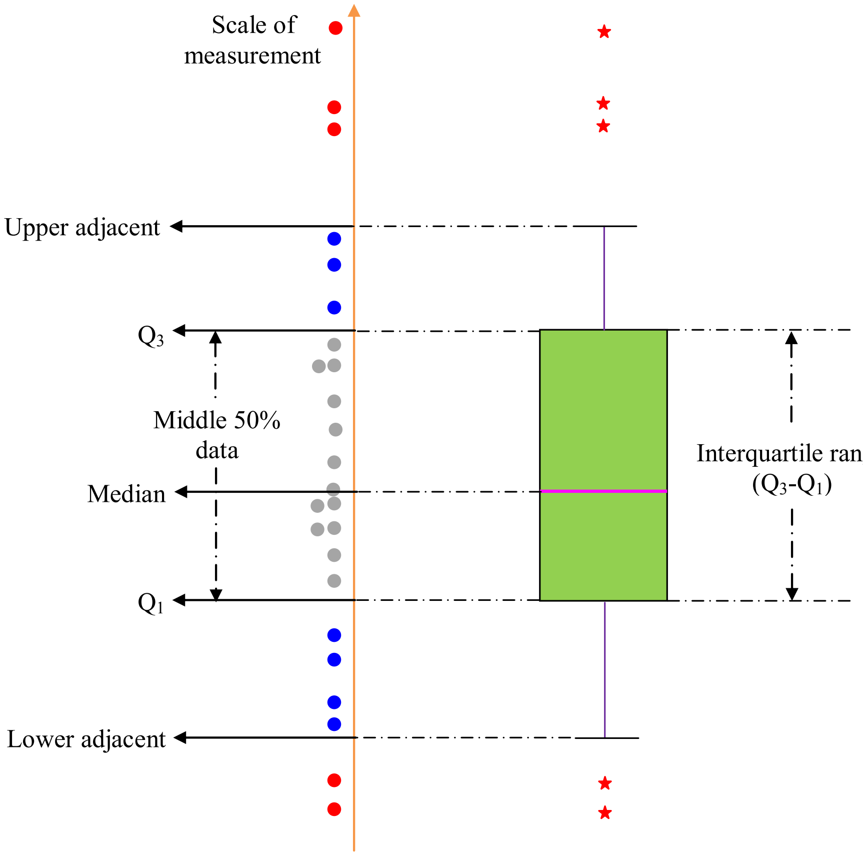


Figure 2.8 A typical boxplot; grey, blue and red blocked circles are regular, extreme and outlying observations respectively.

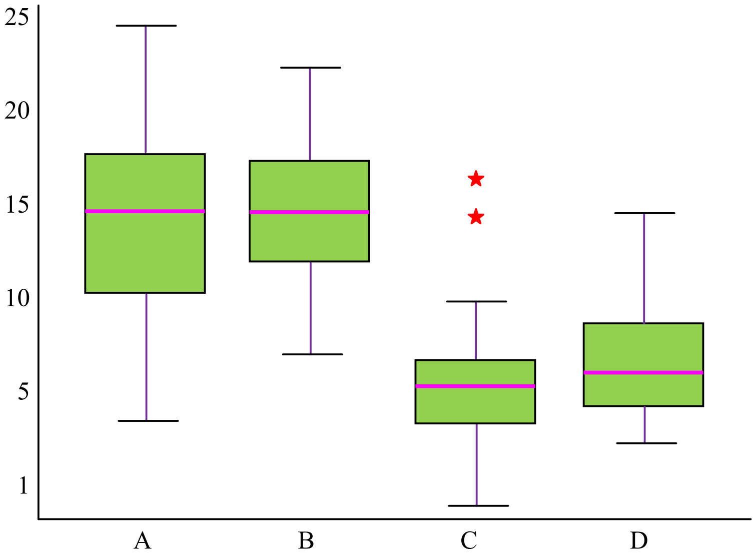


Figure 2.9 Comparison for four different datasets: A, B, C, D of size 25.

2.9 Existing Literature Related to Proposed Algorithms

In this thesis, we have added a review of the relevant literature in every chapter in which the new algorithms are introduced. This section presents a very brief literature survey related to the overall contribution of the research for robust feature extraction related to (i) planar surface fitting, (ii) outlier detection and saliency features estimation, (iii) segmentation, and (iv) ground surface extraction.

2.9.1 Planar Surface Fitting

Plane fitting, and the subsequent estimation of the plane parameters, is essential in point-based laser data processing. The accuracy of plane extraction and fitting is important for later steps involved in surface reconstruction and object modelling. The Least Squares (LS) method is the most well-known classical method for parameter estimation (Klasing et al., 2009). Hoppe et al. (1992) introduced one of the earliest methods for plane fitting using Principal Component Analysis (PCA). Later many authors used the PCA based approach in many ways (Pauly et al., 2002; Rabbani, 2006; Belton, 2008; Sanchez and Zakhor, 2012; Lari and Habib, 2014) for point cloud processing. The PCA based plane fitting method is also known as PlanePCA (Klasing et al., 2009). Fleishman et al. (2005) proposed a forward-search approach based robust moving least squares (Levin, 2003) technique for reconstructing a piecewise smooth surface. The method can deal with multiple outliers, but it requires very dense sampling and a robust initial estimator to start the algorithm. The RANSAC algorithm (Fischler and Bolles, 1981) has been used frequently for planar surface fitting and extraction (Schnabel et al., 2007; Gallo et al., 2011; Masuda et al., 2013). Deschaud and Goulette (2010) pointed out that RANSAC is very efficient in detecting large planes in noisy point clouds. The Hough Transform (Duda and Hart, 1972) has been used to detect geometric shapes and for plane detection in point clouds (Vosselman et al., 2004; Borrmann et al., 2011).

2.9.2 Outlier Detection and Saliency Features Estimation

Point cloud processing tasks frequently use information about local saliencies such as normals and curvature. One of the main problems for accurate normal and curvature estimation is the presence of outliers and/or noise. Outlier detection in laser scanning point cloud data is a challenging task (Hodges and Austin, 2004; Sotoodeh, 2006; Aggarwal, 2013). Outlier detection methods developed depend on the application areas. These include network systems, news documentation, information systems, and industrial machines (Breuning et al., 2000; Schölkopf et al., 2001; Hodges and Austin, 2004; Aggarwal, 2013). Outlier detection in statistics roughly can be categorised into distribution, distance and depth based approaches (Barnett and Lewis, 1995; Rousseeuw and Leroy, 2003). The notion of the distance based approach (Knorr and Ng, 1998) is mainly formulated for large data. Breuning et al. (2000) introduced the density based approach given that object points may be outliers relative to their local neighbourhood. The clustering based approach applies unsupervised clustering techniques mainly to group the data based on their local behaviour (Jiang and An, 2008). Another approach is used to learn a classifier from a set of known data, and then classifies test observations as either inliers or outliers using the learnt model (Schölkopf et al., 2001; Liu et al., 2013). Model based approaches can detect outliers in high-dimensional data but require much more time to construct a classifier (Liu et al., 2013). RANSAC is another approach, most widely used for robust model parameter estimation in the presence of noise and/or outliers. Several survey papers (Hodges and Austin, 2004; Rousseeuw and Hubert, 2011; Schubert et al., 2014) show that people are still trying to get more effective methods specifically for their domain of interest.

Many methods have been developed over the years to improve the quality and speed of normal and curvature estimation in point cloud data. Combinatorial and numerical approaches (Dey et al., 2005; Castillo et al., 2013) are two major categories for normal estimation. Dey et al. (2005) developed combinatorial methods for estimating normals in the presence of noise, but in general, this approach becomes infeasible for large datasets. Numerical approaches find a subset of points in the local neighbourhood that may represent the local surface of an interest point and is better in the presence of outliers and noise. Hoppe et al. (1992) estimated the normal at each point to the fitted plane of the nearest neighbours by applying regression or the 'total least squares' method, which can be computed efficiently by PCA. PCA based plane fitting can be shown to be equivalent to the Maximum Likelihood Estimation (MLE) method (Wang et al., 2001). Distance weighting (Alexa et al., 2001), changing neighbourhood size (Mitra et al., 2004) and higher-order fitting (Rabbani et al., 2006) algorithms have been developed to adjust PCA for better accuracy near sharp features and to avoid the influence of outliers on the estimates. Öztireli et al. (2009) used local kernel regression to reconstruct sharp features. Weber et al. (2012) claimed there is a problem with the reconstruction from Öztireli et al. (2009) as it does not have a tangent plane at a discontinuous sharp feature.

2.9.3 Segmentation

Segmentation is a process of classifying the data points into a number of locally homogenous groups. Existing algorithms can be classified roughly into three categories: (i) edge/border based, (ii) region growing based, and (iii) hybrid (Besl and Jain, 1988; Koster and Spann, 2000; Huang and Menq, 2001). In edge/border based methods, usually points positioned on the edges/borders are detected, and then points are grouped within the identified boundaries and connected edges. In region growing algorithms, generally a seed point is chosen first to grow a region, and then local neighbours of the seed point are combined with the seed point if they have similar surface point properties. The region growing algorithms can also be grid-based (Xiao et al., 2011) and line-based (Harati et al., 2007). The common idea is that region growing based methods are more robust to noise than edge-based methods (Liu and Xiang, 2008), but region growing based methods may suffer from the possibility of over and under segmentation (Chen and Stamos, 2007; Liu and Xiang, 2008). Hybrid methods involve both the boundary/edge and region growing based approaches (Woo et al., 2002). Scan-line based methods (Jiang et al., 2000; Khalifa et al., 2003) adopt a split-and-merge strategy based on grouping the scan lines along a given direction. The approach is not good for unordered point clouds having uneven point density because it is based on the grouping of the scan lines. Marshall et al. (2001) used LS fitting and identified surfaces of known geometric features within a segmentation framework. Klasing et al. (2009) identified the limitations of high computational cost for a large number of features. Poppinga et al. (2008) developed an efficient method of plane fitting by mean squared error computation. Crosilla et al. (2009) did statistical analysis of Gaussian and mean surface curvature for each sampled point for segmentation of laser point clouds. Castillo et al. (2013) introduced a point cloud segmentation method using surface normals computed by the constrained nonlinear least squares approach.

2.9.4 Ground Surface Extraction

Classification of the point cloud into ground and non-ground points namely ground surface extraction or filtering is useful in many point cloud processing tasks. For example, removing the ground points will make the analysis for above ground objects easier and can minimize the time and cost of the remaining analysis for many algorithms. Filtering methods can be categorized into four: (i) morphological filtering, (ii) progressive densification, (iii) surface based filtering, and (iv) segment based filtering. Lindenberger (1993) introduced one of the first morphological filtering methods, in which initially, a rough ground surface is extracted by using a seed point that is the lowest assuming that the lowest point belongs to the ground. Then the rough terrain is refined with an auto-regression process. Kilian et al. (1996) used different morphologic operators. Axelsson (2000) introduced a progressive Triangular Irregular Network (TIN); the algorithm uses the lowest point in large grid cells as the seeds of his approach. Subsequently, the first subset is triangulated in order to form a reference bare earth surface. Then, for each of the triangles within the TIN an additional terrain point is included if certain criteria are fulfilled. Kraus and Pfeifer (1998) introduced a surface based filtering technique that commences by considering all the points belonging to the ground surface and gradually removes those points that do not fit with a general surface model. Pfeifer et al. (2001) and Briese et al. (2002) embedded robust interpolation in a hierarchical approach that can handle different levels of resolution and reduces computation time. Akel et al. (2007) proposed an algorithm based on orthogonal polynomials for extracting terrain points from LiDAR data. Tóvári and Pfeifer (2005) proposed a two-step segmentation algorithm based on a region growing approach. Bartels et al. (2006) introduced a segmentation algorithm based on the central limit theorem where the statistical measure skewness is chosen to describe the characteristics of the point cloud distribution in a segmentation algorithm. Overviews about the filtering algorithms can be found in Pfeifer (2003), Sithole and Vosselman (2004), El-Sheimy et al. (2005), Kobler et al. (2007), and Briese (2010).

2.10 Conclusions

In this chapter we have briefly reviewed several traditional and state-of-the-art principles and related classical, diagnostic and robust methods from different domains with their advantages and disadvantages. Although these have been reviewed for general datasets, their relevance to 3D point clouds and large datasets is mentioned when needed, specifically because we are dealing with large datasets with low dimensionality. In the following chapters, we will adopt the relevant principles and methods in our proposed methods or will be used for comparison.

In the next chapter, we will investigate and propose algorithms for robust planar surface fitting in 3D point cloud data.

Chapter 3

"Don't judge each day by the harvest you reap but by the seeds that you plant."

Robert Stevenson

"...the statistician knows...that in nature there never was a normal distribution, there never was a straight line, yet with normal and linear assumptions, known to be false, he can often derive results which match, to a useful approximation, those found in the real world."

George Box

Robust Planar Surface Fitting

3.1 Introduction

Surface reconstruction and recognition is an important task for extracting information from point cloud processing in subjects including computer vision, computer graphics, computational geometry, photogrammetry, reverse engineering, remote sensing and robotics (Hoppe et al., 1992; Huang and Menq, 2001; Vosselman and Maas, 2010; Weber et al., 2012; Heo et al., 2013). Fitting planes and estimating subsequent plane parameters are essential in point-based representations. Much research has been carried out on accurate local surface fitting and local point set property (e.g. normal) estimation. In surface reconstruction, commonly the quality of the approximation of the output surface depends on how well the estimated surface normals approximate the true normals of the sampled surface (Dey et al., 2005). Surface segmentation, reconstruction, object modelling and rendering are related to each other. They are closely related to local normal and curvature estimation and mostly depend on accurate plane fitting (Hoffman and Jain, 1987; Hoppe et al., 1992; Huang and Menq, 2001; Li et al., 2010). The accuracy of plane extraction and fitting is a cornerstone for later steps of the object modelling pipeline. The Least Squares

(LS) method is the most well-known classical method for parameter estimation, and Principal Component Analysis (PCA) is one of the most popular techniques mainly used for dimension reduction. Both of these methods and their variants are popular for plane fitting. It is known that the methods are influenced by outliers and can lead to inconsistent and misleading results (Mitra and Nguyen, 2003). Point cloud data is acquired mostly by various measurement processes using a number of instruments (sensors) and can easily be distorted by noise and/or outliers. Sotoodeh (2006) pointed out that the physical limitations of the sensors, boundaries between 3D features, occlusions, multiple reflectance and noise can produce off-surface points that appear to be outliers. Apart from noise and outlier contamination, fitting an accurate plane to point cloud data can be complicated by non-uniform point density, incomplete regions of scanned objects and the presence of multiple structures. Many people use RANdom SAmple Consensus (RANSAC) to reduce outlier/noise effects and for robust model parameter estimation (Schnabel et al., 2007; Gallo et al., 2011; Sanchez and Zakhor, 2012). LS, its equivalent PCA and RANSAC are the three most popular and classical techniques for fitting planes in 3D data (Hoppe et al., 1992; Pauly et al., 2002; Schnabel et al., 2007; Klasing et al., 2009; Deschaud and Goulette, 2010; Heo et al., 2013).

It is logical, if the uncertainties of the sampled points in the point cloud data are known, then the outliers can be tested against prior knowledge. However, this is not always possible, and if it is possible, it is non-trivial. It has been demonstrated that the uncertainty of a point is highly depended on the attributes of the scanner and the scanner geometry, such as distance and surface orientation (Bae et al., 2005; Soudarissanane et al., 2011). This information is not always available, such as when a scene comprises multiple co-registered laser scans acquired from different positions. The properties only relate to a single point, not to the local sampled surface model. The surface properties will be based on pooled variance models based on each scan. It should also be pointed out that to find the scanner geometry, the local surface must be adequately defined, a process that will also be affected by the presence of outliers and could cause errors in the calculated uncertainties. In recognition of these factors, this chapter focuses on examining the points robustly, based on the local neighbourhood distribution.

In order to be resilient to outliers, robust and diagnostic statistics, two branches of statistics have been proposed (Huber, 1981; Hampel et al., 1986; Stewart, 1999). Robust statistics produce statistical procedures, which are stable w.r.t. small changes or deviations in the data, and even large changes in the underlying data pattern cannot cause a complete failure of the procedures. Alternatively, diagnostic statistics condition on the fit using standard methods to attempt to diagnose incorrect assumptions, allowing the analyst to modify them and refit under the new set of assumptions (Huber, 1991; Stahel and Weisberg, 1991). It is known that the two branches are complementary and their combined use is argued to be more effective for producing highly robust estimators and to detect multiple outliers and/or high leverage points (Rousseeuw and Leroy, 2003).

We want fast fitting as well as accuracy of planar surfaces to be able to efficiently process point clouds consisting of large amounts of unorganized points. We propose six variants of Fast-MCD (Rousseeuw and Driessen, 1999) and Deterministic MCD (DetMCD; Hubert et al., 2012) based robust algorithms that use diagnostic, robust and the combination of diagnostic and robust (diagnostic-robust) statistical approaches for planar surface fitting in 3D point cloud data, which are able to find outliers and robust estimates of the parameters at the same time. The proposed robust plane fitting methods also produce robust normal and curvature values. The accuracy and robustness of the methods are compared w.r.t. the size of the data, outlier percentage, point density variation, speed of computation and for different applications in point cloud analysis. We compare the new methods with LS, PCA, RANSAC and MSAC (M-estimator SAmple Consensus), and show that the results from the proposed methods are significantly better than the existing methods.

The remaining sections are as follows: Section 3.2 presents a short literature review. Section 3.3 formulates the necessary calculations for the proposed algorithms. Section 3.4 implements the proposed algorithms for fitting planes and to get robust local normals and curvatures from the best-fit-plane. In Section 3.5, we experiment, evaluate and compare the results and the performance of the proposed techniques with the other methods using simulated and real mobile laser scanning (MLS) datasets. Section 3.6 explores the computational speed and effort of the proposed algorithms followed by conclusions in Section 3.7.

3.2 Literature Review

Many studies have been carried out on reliable plane fitting/detection/extraction, and robust normal estimation in 3D point clouds and range data. Different methods have been developed in different disciplines (e.g. computer vision, computer graphics, computational geometry, robotics, photogrammetry, remote sensing, machine learning and statistics) according to their suitability and to meet their purposes of plane fitting (Wang et al., 2001; Deschaud and Goulette, 2010), surface reconstruction (Yoon et al., 2007; Sheung and Wang, 2009), sharp feature preserving (Fleishman et al., 2005; Weber et al., 2012; Wang et al., 2013) and normal estimation (Mitra and Nguyen, 2003; Klasing et al., 2009; Li et al., 2010; Boulch and Marlet, 2012). The methods have been formulated for different applications but they are interrelated. Although there are many methods that exist in the literature, three types of approaches have been thoroughly investigated and are popular as the foundations of many of the others; they are: LS, PCA and RANSAC.

Hoppe et al. (1992) introduced one of the earliest methods for plane fitting and normal estimation. This method has drawn much attention and uses PCA where tangent planes are estimated from the local neighbours of each sample point. Many authors use the PCA based approach in many ways (Pauly et al., 2002; Rabbani, 2006; Belton, 2008; Sanchez and Zakhor, 2012; Lari and Habib, 2014; Lin et al., 2014) for point cloud processing. The PCA based method can be defined as an optimization approach that minimizes the LS cost function. PCA based plane fitting is also known as *PlanePCA* (Klasing et al., 2009), which is a geometric optimization and can be shown to be the equivalent of the Maximum Likelihood Estimation (MLE) method if the points p_i $(i = 1, \dots, n)$ are regarded as measurements perturbed by independent Gaussian noise $N(0, \sigma^2)$ around their true position (Kanatani, 1996; Wang et al., 2001). In a study, Klasing et al. (2009) compared a number of optimization and averaging methods and showed that when using a k nearest neighbourhood the *PlanePCA* and the *PlaneSVD* (LS, Klasing et al., 2009) are the two most efficient methods for plane fitting and normal estimation in terms of both quality of results and speed. It is evident that the results from PCA are affected by outlying observations, because the mean vector and covariance matrix used here have an unbounded Influence Function (IF) and zero Breakdown Point (BP; Hampel et al., 1986). To avoid the outlier/noise influence on the estimates from PCA, robust versions of PCA have been introduced in the statistical literature (Hubert et al., 2005; Xu et al., 2010; Feng et al., 2012). Fleishman et al. (2005) proposed a forward-search approach based robust moving least squares (Alexa et al., 2001; Levin, 2003) technique for reconstructing a piecewise smooth surface and reliable normal estimation. Although, the method can deal with multiple outliers, it requires very dense sampling and a robust initial estimator to start the forward search algorithm. Sheung and Wang (2009) showed that forward-search misclassifies the noisy regions at corners since it fails to obtain a good initial fit.

The RANSAC algorithm is a model-based robust estimator used widely for planar surface detection, extraction, fitting, and normal estimation (Boulaassal et al., 2007; Schnabel et al., 2007; Gallo et al., 2011; Masuda et al., 2013). Boulaassal et al. (2007) used RANSAC to extract planar parts from building façades. Schnabel et al. (2007) developed two optimizations to RANSAC. Deschaud and Goulette (2010) claimed the algorithms of Schnabel et al. (2007) are slow for large datasets and also pointed out that RANSAC is very efficient in detecting large planes in noisy point clouds but very slow to detect small planes in large point clouds.

The Hough Transform (HT; Duda and Hart, 1972) is another model-based method used for detecting parameterized objects where each data point casts its vote in the parameter space. That is represented by a multi-dimensional histogram consisting of discrete cells. For a plane, this typically has four dimensions. The cell for the HT with the largest number of votes defines the most appropriate parameters for the model. Vosselman et al. (2004) used the HT to detect geometric shapes. Borrmann et al. (2011) used the HT for plane detection in point clouds. Deschaud and Goulette (2010) argued that it is too time consuming for fitting a model to a large dataset. The HT is also sensitive to accumulator design. Tarsha-Kurdi et al. (2007) showed that the HT is sensitive to the segmentation parameters and that RANSAC is more efficient than the HT in terms of processing time. We choose RANSAC, and not the HT for comparison because of its advantages over the HT and its popularity.

3.3 Necessary Calculations for the Proposed Algorithms

In this section, we propose six techniques for robust local planar surface fitting in laser scanning 3D point cloud data. The techniques can be classified into three algorithms based on the statistical approaches used: (i) diagnostics approach (ii) robust version of PCA, and (iii) the combination of diagnostics and robust PCA. Robust estimators of the mean vector (simply the mean) and covariance matrix from Fast-MCD and DetMCD are used in the three algorithms. The proposed algorithms, namely diagnostic PCA, robust PCA, and diagnostic robust PCA use a robust mean vector and a covariance matrix to get robust distance for finding outliers and to determine the 'outlyingness' measure w_i in robust PCA. Outlier detection methods involve robust distance, which uses the robust mean vector and covariance matrix. The workflow for the proposed algorithms is shown in Figure 3.1. Each of the stages in the workflow will be described in the following sections.

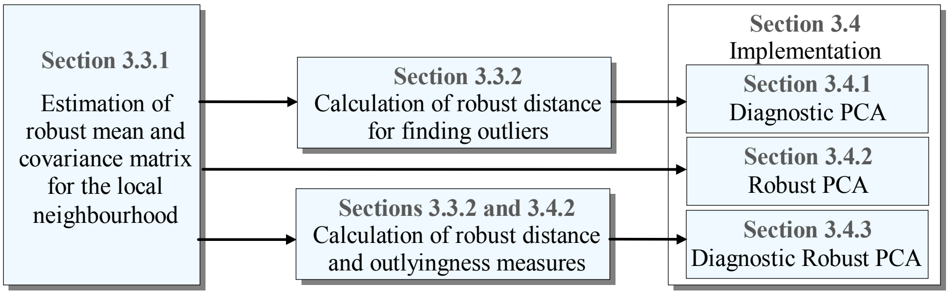

Section 3.3.1
Estimation of robust mean and covariance matrix for the local neighbourhood
Section 3.3.2
Calculation of robust distance for finding outliers
Sections 3.3.2 and 3.4.2
Calculation of robust distance and outlyingness measures
Section 3.4
Implementation
Section 3.4.1
Diagnostic PCA
Section 3.4.2
Robust PCA
Section 3.4.3
Diagnostic Robust PCA

Figure 3.1 Work flow for the proposed algorithms.

3.3.1 Derivation of Robust Mean Vector and Covariance Matrix

In this chapter, we are interested in fitting local planar surfaces to 3D point cloud data. We represent the point cloud of n points in three dimensions as a $P_{n\times3}$ matrix, $P = (p_1, \ldots, p_n)^T$, with the i^{th} observation $p_i = (p_{i1}, p_{i2}, p_{i3})$. As discussed in Section 2.3.2 Chapter 2, the Minimum Covariance Determinant (MCD) estimator (Rousseeuw, 1984) is a high breakdown estimator of the mean and covariance matrix. The MCD estimator searches for the h ($h > n/2$) observations whose covariance matrix has the lowest determinant. The computation of the MCD method is not easy and requires an exhaustive search in n points for all the subsets of h (written as h-subsets) points. Since the MCD estimator has many good theoretical properties including better statistical efficiency, being affine equivariant, having a bounded influence function, and having a breakdown point of 50%, we use the MCD approach for deriving the robust mean and covariance matrix. Although, the MCD in Rousseeuw (1984) was computationally very intensive, later Rousseeuw and Driessen (1999) and more recently Hubert et al. (2012) developed two versions of MCD, which are more efficient and significantly faster than the classical MCD without losing good statistical properties. We illustrate the workflow for the different stages of the MCD algorithm in Figure 3.2.

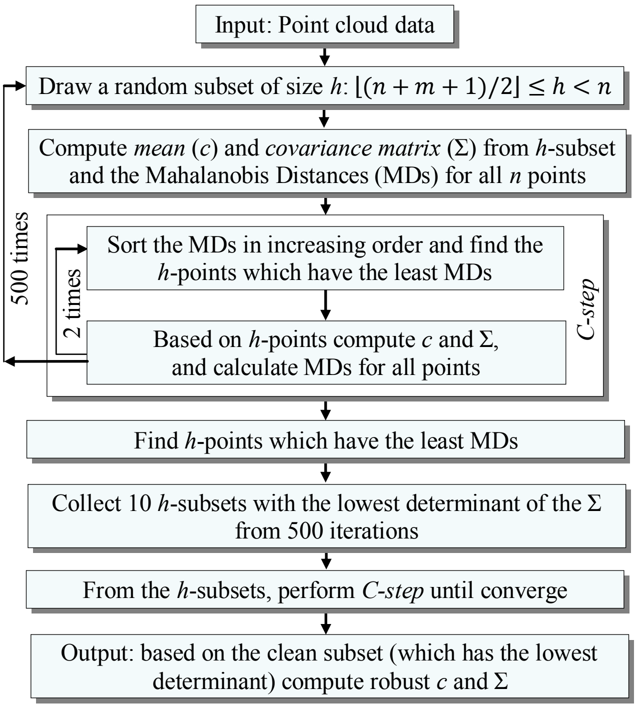

Input: Point cloud data
Draw a random subset of size h: $\lfloor (n+m+1)/2 \rfloor \leq h < n$
Compute *mean* (c) and *covariance matrix* (Σ) from h-subset and the Mahalanobis Distances (MDs) for all n points
Sort the MDs in increasing order and find the h-points which have the least MDs
Based on h-points compute c and Σ, and calculate MDs for all points
500 times
2 times
C-step
Find h-points which have the least MDs
Collect 10 h-subsets with the lowest determinant of the Σ from 500 iterations
From the h-subsets, perform *C-step* until converge
Output: based on the clean subset (which has the lowest determinant) compute robust c and Σ

Figure 3.2 Minimum Covariance Determinant (MCD) algorithm workflow.

In the proposed algorithms, both Fast-MCD (Rousseeuw and Driessen, 1999) and Deterministic MCD (Hubert et al., 2012) are used to get robust mean vector and covariance matrix. The Fast-MCD (FMCD) is a resampling algorithm which can avoid a complete enumeration to efficiently estimate the MCD for large amounts of data. To get an outlier-free initial subset of size $m + 1$, many initial random subsets need to be drawn, which is computationally intensive. Rousseeuw and Driessen (1999) fixed the number of iterations at 500 to get a good sample and to keep the computation time to an acceptable level. To minimize the computational time FMCD also uses only two *C-steps* for each of the 500 initial subsets, and uses *selective iteration* and *nested extensions* (when n is large, say more than 600) as two further steps. It then keeps the 10 results with the lowest determinant. From these 10 subsets, *C-steps* are performed until convergence to get the final h-subset. We use this h-subset to get the final FMCD based robust mean vector and covariance matrix. In addition to the advantages of the MCD, the FMCD algorithm allows exact-fit situations, i.e. when more than h observations lie on a hyper plane (Rousseeuw and Driessen, 1999).

Recently, Hubert et al. (2012) introduced a deterministic algorithm for the MCD (DetMCD) to get the robust mean vector (location) and covariance matrix (scatter). FMCD draws many random $(m + 1)$-subsets to obtain at least one outlier-free subset, whereas DetMCD starts from a few easily computed h-subsets and then performs the *C-steps* until convergence. It uses the same iteration step but does not draw a random subset. Rather it starts from only a few well-chosen initial estimators followed by the *C-steps*. DetMCD couples aspects of both the FMCD and the orthogonalized Gnanadesikan and Kettenring estimators (Maronna and Zamar, 2002). This algorithm is almost affine equivariant, and permutation invariant (the result does not depend on the order of the data) but FMCD is not permutation invariant. The authors claimed that DetMCD is much faster than FMCD and at least as robust as FMCD. The reader is referred to Hubert et al. (2012) for more details about DetMCD.

3.3.2 Computation of Robust Distance

We use the well-known distance based multivariate outlier detection technique for 3D point cloud data, where the distance considers the shape (covariance) as well as the centre of the data. Robust distance is employed to find outliers in the sampled data to fit a plane. Mahalanobis Distance (MD) in Eq. (2.31) Chapter 2 is the most popular multivariate measure that computes the distance of an observation from the mean of the data.

Although, it is possible to detect a single outlier by means of MD, this approach is no longer sufficient for multiple outliers because of the well-known masking effect (Rousseeuw and Driessen, 1999). Masking occurs when an outlying subset goes undetected because of the presence of another, usually adjacent, subset (Hadi and Simonoff, 1993). Replacing the classical mean vector and covariance matrix by robust counterparts, a robust method yields a tolerance ellipse that captures the covariance structure of the majority of the dataset. Rousseeuw and van Zomeren (1990) used the Minimum Volume Ellipsoide (MVE) based mean vector and covariance matrix, but we know that MVE has zero efficiency because of its low rate of convergence. We use two versions of robust distances using FMCD and the DetMCD based mean vector and covariance matrix in Eq. (2.33) Chapter 2 namely FRD (Fast-MCD based Robust Distance) and DetRD (DetMCD based Robust Distance). FRD and DetRD for the i^{th} point can be defined respectively as:

$$\mathrm{FRD}_i = \sqrt{(p_i - c_{\mathrm{FMCD}})^T \ \Sigma^{-1}_{\mathrm{FMCD}} \ (p_i - c_{\mathrm{FMCD}})}, \quad i = 1, \ldots, n \tag{3.1}$$

$$\mathrm{DetRD}_i = \sqrt{(p_i - c_{\mathrm{DetMCD}})^T \ \Sigma^{-1}_{\mathrm{DetMCD}} \ (p_i - c_{\mathrm{DetMCD}})}, \quad i = 1, \ldots, n. \tag{3.2}$$

The cut-off value for identifying outliers is to some extent arbitrary and mainly depends on knowledge about the data. Rousseeuw and van Zomeren (1990) and Rousseeuw and Driessen (1999) showed that the robust distance follows a Chi-square (χ^2) distribution with m (number of variables) degrees of freedom. The authors argued that the observations that have Mahalanobis distance or robust distance (FRD and DetRD) values $\geq \sqrt{\chi^2_{m,0.975}}$ can be identified as outliers.

To show the performance of MD, FRD and DetRD for multivariate outlier detection, we generate 30 points in two dimensions that have a linear pattern as shown in Figure 3.3. We deliberately deviate, from the majority pattern, one point in Figure 3.3a and five points in Figure 3.3b to generate single and multiple outliers in the datasets respectively. Based on the MD, FRD and DetRD values, corresponding ellipses are drawn. First, outliers are identified by using Chi-square criteria, then without the outliers the respective covariance matrices have been derived, which are later used to generate the ellipses for exploring the outliers effect. We see all the methods are successful in identifying a single outlier (Figure 3.3a) as the outlier falls outside the ellipses. In

Figure 3.3b, MD fails in the presence of multiple outliers as it includes them in the ellipse. The computed ellipses for MD for one or more outliers are significantly changed or distracted by the outliers. This is the well-known masking effect. The ellipses for FRD and DetRD are not significantly changed by the presence of the outliers and successfully identify all five outlying points without the ellipse directions being affected (Figure 3.3b).

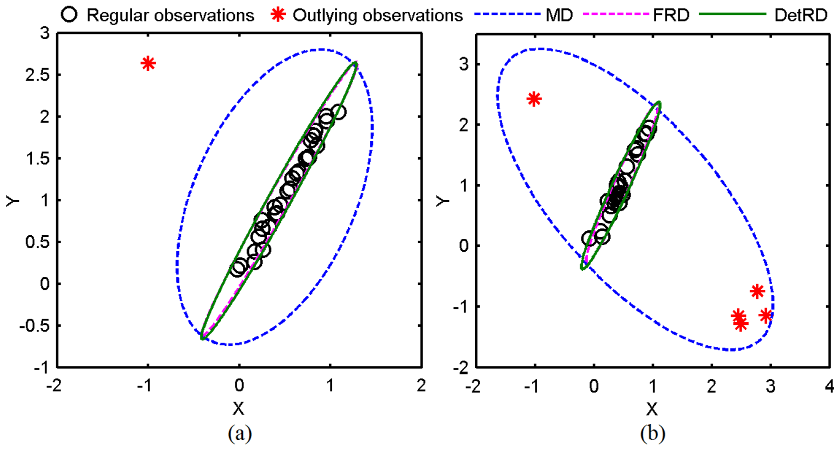


Figure 3.3 Outlier (red point) detection by MD, FRD and DetRD, in the presence of: (a) a single outlier, and (b) multiple and clustered outliers.

3.4 Implementation

The statistical algorithms implemented in this section for local planar surface fitting in 3D point cloud data use two complementary statistical paradigms: diagnostic and robust statistics. Based on the FMCD and DetMCD estimators, three algorithms are proposed: (i) diagnostic PCA, (ii) robust PCA, and (iii) diagnostic robust PCA. Diagnostic and robust statistics have the same objective of fitting a model that is resilient to outliers. However the analysis stages for diagnostic statistics occur in reverse order for robust statistics. In diagnostic statistics, first the outliers are detected and deleted and then the remainder of the data is fitted in the classical way, whereas in robust statistics, first a model is fitted that does justice to the majority of observations and then the outliers that have large deviations (e.g. residuals) from the robust fit are detected.

For local neighbourhood based point cloud processing, data points from a local planar surface are sampled from within a local fixed radius r or within a local neighbourhood of size k. We use the well-known k Nearest Neighbourhood (kNN) searching technique (Figure 3.4a) rather than the Fixed Distance Neighbourhood (FDN) method (Figure 3.4b) because kNN is able to avoid the problem of point density variation. We know point density variation is a common phenomenon particularly when we are dealing with mobile laser scanning data because of the movement of the data acquisition sensors (or vehicles) relative to the geometry of the sensors. Density varies as a function of orientation of a surface relative to the sensor, and as a function of the path taken by the sensor or vehicle and its velocity. A further advantage is that the same size of local neighbourhood can produce local statistics (e.g. normal and curvature) of equal support.

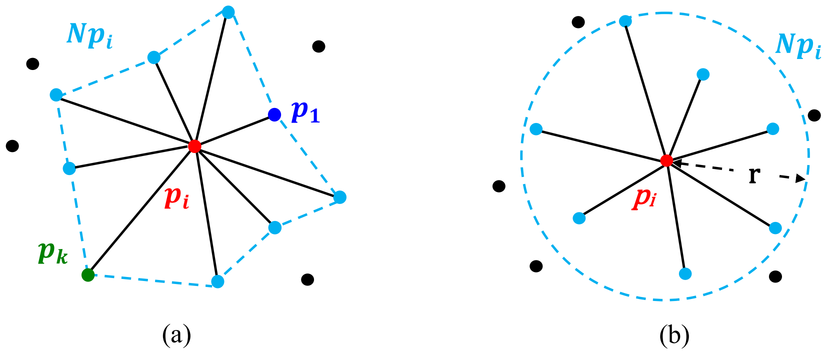


Figure 3.4 Local neighbourhood (region) for p_i: (a) k nearest neighbourhood, and (b) fixed distance neighbourhood.

3.4.1 Diagnostic PCA

The algorithm proposed here is inspired by diagnostic statistics, and couples the ideas of outlier diagnostics and classical PCA. First, we detect and remove outliers from the dataset, and then fit a planar surface using PCA to the cleaned data. For local planar surface fitting, we need to find the local region of an interest point p_i as shown in Figure 3.4.

After fixing a local neighbourhood Np_i, we find outliers in the neighbourhood using robust distance (FRD or DetRD) in Eq. (3.1) or Eq. (3.2). We then fit a plane using classical PCA to the cleaned data. The best-fit-plane is obtained by projecting all the inlier points onto the two Principal Components (PCs) with the highest eigenvalues. The third PC is the normal to the fitted plane, and the elements of the corresponding third eigenvector are the estimated plane parameters.

The algorithm for diagnostic PCA (called RD-PCA) based on robust distance is described in Algorithm 3.1: RD-PCA (Robust Distance based PCA) as follows.

Algorithm 3.1: RD-PCA (Robust Distance based PCA)

```
1. Input: point cloud $P$, neighbourhood size $k$, $\chi^2$ (Chi-square) cut-off=3.075.
2. Determine the local neighbourhood $Np_i$ for a point $p_i$ consisting of its $k$ nearest neighbours.
3. Calculate robust distance (FRD or DetRD) for each point in $Np_i$.
4. Classify the points in $Np_i$ into inliers and outliers according to the respective FRD or DetRD values and the $\chi^2$ cut-off value assigned.
5. Perform PCA on the inlier matrix.
6. Arrange the three PCs associated with their respective eigenvalues.
7. Find the two PCs that have the largest eigenvalues, and fit the plane by projecting the points onto the directions of the two PCs.
8. Output: normals, eigenvalues and the necessary statistics such as curvature.
```

The RD-PCA algorithm can be performed in two different ways: using FRD and DetRD in place of RD for finding outliers in the local neighbourhood. We name the FRD based diagnostic PCA and DetRD based diagnostic PCA as FRD-PCA and DetRD-PCA respectively.

3.4.2 Robust Principal Component Analysis

Robust statistics fit a model considering the consensus of the majority of observations and then as an extra benefit can find the outliers that have large deviations from the robust fit. We know that robust covariance matrix based methods and Projection Pursuit (PP; Friedman and Tukey, 1974) methods have some limitations. The robust covariance matrix based approach may face the problem of lacking sufficient data to estimate a high-dimensional robust covariance matrix. In contrast, the robustness of the PP based methods depends on the robustness of the adopted estimators. The solely PP based methods are faster but robust covariance matrix methods with PP give more robust PCs than the PP methods (Friedman and Tukey, 1974; Li and Chen, 1985). We choose robust PCA (RPCA) introduced by Hubert et al. (2005) because it yields accurate estimates of outlier-free datasets, produces more robust estimates for contaminated data, is able to detect exact-fit situations, is location and orthogonal invariant, and has the further advantage of outlier diagnostics and classification

This approach couples the idea of PP to make sure that the transformed data are lying in a subspace whose dimension is less than the number of observations, and then uses the robust covariance matrix based method to get the final robust PCs. In the case of 3D point cloud data we have the advantage that usually the data dimension $(m = 3) <$ the number of points in the dataset for fitting a plane. The RPCA algorithm can then be performed using the stages in Algorithm 3.2.

We perform the DetMCD based robust PCA algorithms by plugging the DetMCD based mean vector and covariance matrix for finding outlying cases into Eq. (3.3) and in the relevant places of the RPCA algorithm. The FMCD and DetMCD versions of RPCA are called FRPCA and DetRPCA respectively.

3.4.3 Diagnostic Robust PCA

Fung (1993) pointed out that robust and diagnostic methods do not have to be competing, and the complementary use of highly robust estimators and diagnostic measures provides a very good way to detect multiple outliers and leverage points. To see the effectiveness of using diagnostic and robust approaches at the same time, we propose the Diagnostic Robust PCA (DRPCA) algorithm, which is the combination of diagnostic and robust PCA. First the RDs are used to find outliers in a local neighbourhood to which we want to fit a plane. Then we use RPCA to fit the plane to the cleaned data. One of the DRPCA based algorithms uses FMCD based FRD and FRPCA and is called FDRPCA, and the other uses DetMCD based DetRD and DetRPCA and is called Deterministic Diagnostic Robust PCA (DetDRPCA). In DetDRPCA, we find candidate outliers using robust distance DetRD from the local surface (neighbourhood, Np_i). Finding outliers and removing them from the Np_i makes the data more homogeneous. Second, we use DetMCD based robust PCA (DetRPCA) to get the required PCs and the eigenvalues. The DetDRPCA method can be summarized in Algorithm 3.3.

Algorithm 3.2: RPCA (Robust PCA)

1. Input: point cloud P, neighbourhood size k.
2. Determine the local neighbourhood Np_i for a point p_i consisting of its k nearest neighbours.
3. Process the data to make sure that the data is lying in a subspace whose dimension is at most $k-1$.
4. Compute the measure of outlyingness for each point in the neighbourhood by projecting all the data points onto univariate directions passing through two individual data points. The dataset is compressed to PCs defining potential directions. The value of outlyingness for a point p_i is:
$$w_i = \underset{v}{argmax} \frac{|p_i v^T - c_{\text{FMCD}}(p_i v^T)|}{\Sigma_{\text{FMCD}}(p_i v^T)}, \quad i = 1, \ldots, k \tag{3.3}$$
where $p_i v^T$ denotes a projection of the i^{th} observation onto the v direction, c_{FMCD} and Σ_{FMCD} are the FMCD based mean vector and covariance matrix on a univariate direction v.
5. Construct a robust covariance matrix Σ_h using an assumed portion $(h > k/2)$ of observations with the smallest outlyingness values. We use $h = \lceil 0.5 \times k \rceil$ in our algorithm.
6. Project the observations onto the d dimensional subspace spanned by the d largest eigenvectors of Σ_h, and compute the mean vector and covariance matrix by means of reweighted FMCD estimator with weights based on the robust distance of every point.
7. The eigenvectors of this covariance matrix from the reweighted observations are the final robust PCs, and the FMCD mean vector serves as a robust mean vector.
8. Arrange the three PCs associated with their respective eigenvalues.
9. Find the two PCs that have the largest eigenvalues, and fit the plane by projecting the points onto the directions of the two PCs.
10. Output: normals, eigenvalues and the necessary local statistics such as curvature.
11. Outlier detection: calculate Orthogonal Distance (OD) and Score Distance (SD) using:
$$\text{OD}_i = ||p_i - \hat{p}_i|| = ||p_i - \hat{\mu}_p - L t_i^T||, \quad i = 1, \ldots, k \tag{3.4}$$
where $\hat{\mu}_p$ is the robust centre of the neighbourhood, L is the robust loading (PC) matrix, which contains robust PCs as the columns in the matrix, and $t_i = (p_i - \hat{\mu}_p)L$ is the i^{th} robust score; and
$$\text{SD}_i = \sqrt{\sum_{j=1}^{d} (t_{ij}^2 / l_j)}, \quad i = 1, \ldots, k \tag{3.5}$$
where l_j is the j^{th} eigenvalue of the robust covariance matrix Σ_{FMCD}, and t_{ij} is the ij^{th} element of the score matrix:
$$\text{T}_{k,d} = (P_{k,m} - 1_k c_{\text{FMCD}}) L_{m,d}, \tag{3.6}$$
where $P_{k,m}$ is the data matrix, 1_k is the column vector with all k components equal to 1, c_{FMCD} is the robust centre, and $L_{m,d}$ is the matrix constructed by the robust PCs. The cut-off value for the score distance is $\sqrt{\chi^2_{d,0.975}}$, and for the orthogonal distance is a scaled version of χ^2 (see Hubert et al., 2005).

Algorithm 3.3: DetDRPCA (DetMCD based Diagnostic Robust PCA)

```
1. Input: point cloud $P$, neighbourhood size $k$.
2. Determine the local neighbourhood $Np_i$ for a point $p_i$ consisting of its $k$ nearest neighbours.
3. Calculate robust distance using DetRD for all the points in $Np_i$.
4. Classify inliers (regular observations) and outliers based on the DetRD values.
5. Perform robust PCA using DetRPCA based on the inliers from Step 4.
6. Arrange the three PCs associated with their respective eigenvalues.
7. Find the two PCs that have the largest eigenvalues, and fit the plane by projecting the points onto the directions of the two PCs.
8. Output: normals, eigenvalues and the necessary statistics such as curvature.
9. Outlier detection: similar to Step 11 in Algorithm 3.2.
```

In Algorithm 3.3, if we calculate the robust distance in Step 3 using FRD, and robust PCA in Step 5 using FRPCA then we have the algorithm FDRPCA (Fast-MCD based Diagnostic Robust PCA).

3.5 Experimental Results

In this section, we experiment, evaluate and compare the results of the proposed techniques with other methods using simulated and real datasets. The simulated datasets in Section 3.5.2 will demonstrate and quantify the abilities of the proposed techniques to deal with the presence and effects of outliers, and will be compared to existing techniques commonly used including LS, PCA, RANSAC, and MSAC. At the same time we will show the comparative performance of the proposed algorithms. In Section 3.5.3, the techniques will be tested on real datasets captured from MLS. It will demonstrate the ability to more accurately perform common existing point cloud processing techniques in the presence of outliers. Such existing techniques include plane extraction, sharp feature preservation and segmentation.

3.5.1 Performance Measures Used for Evaluation

To show the performance, we fit planar surfaces using different methods, and estimate normal and eigenvalue characteristics. To determine the performance, we calculate three measures:

- The first is the bias (dihedral) angle θ (Wang et al., 2001) between the planes fitted to the data with and without outliers, defined as:

$$\theta = arcos|\hat{n}_1^T.\hat{n}_2|, \tag{3.7}$$

  where $\hat{n}_1$ and $\hat{n}_2$ are the two unit normals from the fitted planes with and without outliers respectively. To avoid the 180° ambiguity of the normal vectors, the absolute value in Eq. (3.7) is used.

- The second is the variation along the plane normal, which is defined by the least eigenvalue λ_0, in Figure 3.5.

- The third is the surface variation (Pauly et al., 2002), a measure of curvature, determined along the directions of the corresponding eigenvectors at the point p_i in a neighbourhood Np_i of size k that is defined as:

$$\sigma(p_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \quad \lambda_0 \leq \lambda_1 \leq \lambda_2 \tag{3.8}$$

  where λ_i, $i = 0,\ 1,\ 2$ is the i^{th} eigenvalue, and λ_2 and λ_1 are the two largest eigenvalues corresponding to the first two PCs.

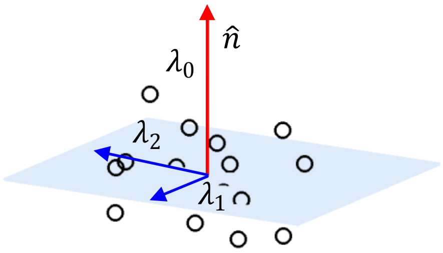


Figure 3.5 Point variation along the plane normal and the first two PCs.

3.5.2 Simulated Data

Simulated data are used to demonstrate and evaluate some typical behaviours including: (i) influence of outliers on bias angle θ, which can be considered as the effect on the estimated plane parameters, (ii) effect on bias angle of point density variation in surface directions (x, y) and surface thickness, and (iii) classification of points into inliers and outliers. Bias angles are estimated in terms of sample size and the percentage of outlier contamination. Statistical significance tests are used to check for any significant difference between the methods, to rank them, and to identify a reduced set of methods considered for further effective comparison.

The artificial datasets used in this section are generated from points randomly drawn from two sets of 3D (x, y, z) Gaussian normal distributions. One set is used to generate points on a plane and the other set is for outlying points. Figure 3.6 shows where regular points are generally in the plane with some variation due to noise, and the outlying points are far from the planar surface. The regular points in 3D have means (3.0, 3.0, 3.0) and variances (7.0, 7.0, 0.01), and the outlying points have means (8.0, 10.0, 12.0) and variances (7.0, 7.0, 1.0). We simulate the datasets for different sample sizes n and Outlier Percentages (OP). By performing experiments on several real MLS data, we observed neighbourhood sizes 20 to 200 are good for point cloud processing such as plane fitting, normals and curvatures estimation that are used later in point cloud processing task e.g. segmentation. Therefore, in the following sections we create different sizes of data, usually between 20 to 200, for simulation and necessary demonstrations, and use different values of that size for real point cloud data analysis. Figure 3.6 shows an example of the simulated data of 100 points with 20% outliers and the fitted planes for PCA and robust methods.

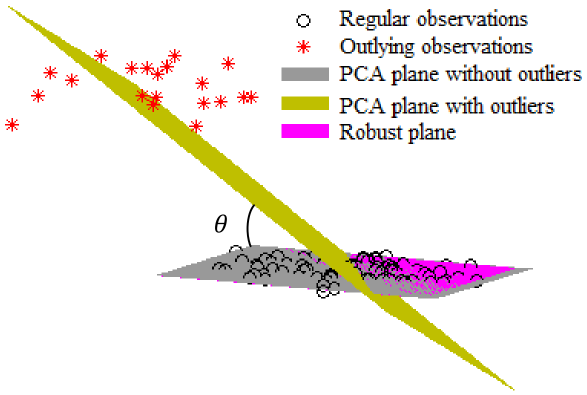


Figure 3.6 Simulated dataset of 100 points including 20% outliers; outliers influence on the fitted planes using PCA with and without outliers and a robust method.

3.5.2.1 Plane Fitting and Bias Angle Evaluation

We create 1000 datasets of 100 points including 20% outliers (e.g. one set is shown in Figure 3.6) to get statistically representative results. We fit the planes with and without outliers for all the datasets using ten methods: LS, PCA, RANSAC, MSAC, FRD-PCA, FRPCA, FDRPCA, DetRD-PCA, DetRPCA, and DetDRPCA. Parameters for RANSAC and MSAC have been set according to Zuliani (2011) and the noise level has been set up as the data generated. We calculate different descriptive measures as shown in Table 3.1 for bias angles (in degrees) from the 1000 fits. Results show that LS has the largest and deterministic MCD based diagnostic PCA (DetRD-PCA) has the smallest mean, median and standard deviation (StD) for bias angles. These results also demonstrate that bias angles from outlier resistant methods are lower than those for the non-robust methods.

Table 3.1 Descriptive measures (in degrees) for bias angles from different methods.

Methods	Mean	95% Confidence interval of mean		Minimum	Maximum	Median	StD
		Lower Bound	Upper Bound				
LS	53.006	50.460	53.470	10.221	90.647	53.082	16.432
PCA	39.746	39.418	39.980	32.129	52.871	39.521	2.347
RANSAC	0.801	0.756	0.945	0.007	4.128	0.563	0.868
MSAC	0.799	0.748	0.852	0.002	4.200	0.549	0.812
FRD-PCA	0.208	0.197	0.214	0.004	0.971	0.175	0.127
DetRD-PCA	0.205	0.194	0.212	0.004	1.098	0.172	0.121
FRPCA	0.246	0.245	0.254	0.015	1.032	0.219	0.143
DetRCPA	0.242	0.236	0.261	0.002	1.163	0.216	0.141
FDRPCA	0.218	0.200	0.226	0.012	1.022	0.192	0.136
DetDRPCA	0.213	0.211	0.229	0.009	1.637	0.190	0.132

To see the effect of different percentages of outlier contamination on a fixed number of data points, we again simulate 1000 datasets of 100 points with outlier presence ranging from 1% to 40%. We fit the planes for every dataset of 100 points with and without outliers, and calculate average bias angles. In Figure 3.7, we plot the average bias angles versus outlier percentages for different methods. In Figure 3.7a, it is clear that LS and PCA have large average bias angles while robust methods have very low average bias angles. Removing the LS and PCA results, Figure 3.7b, we see that RANSAC and MSAC have worse bias angles compared to the more robust statistical methods (FRD-PCA, FRPCA, FDRPCA, DetRD-PCA, DetRPCA and DetDRPCA). Figure 3.7c compares the robust statistical methods. Table 3.1 and Figure 3.7 show that the deterministic MCD (DetMCD) based methods DetRD-PCA, DetRPCA and DetDRPCA generally have less average bias angles than their FMCD based counterparts FRD-PCA, FRPCA and FDRPCA, respectively.

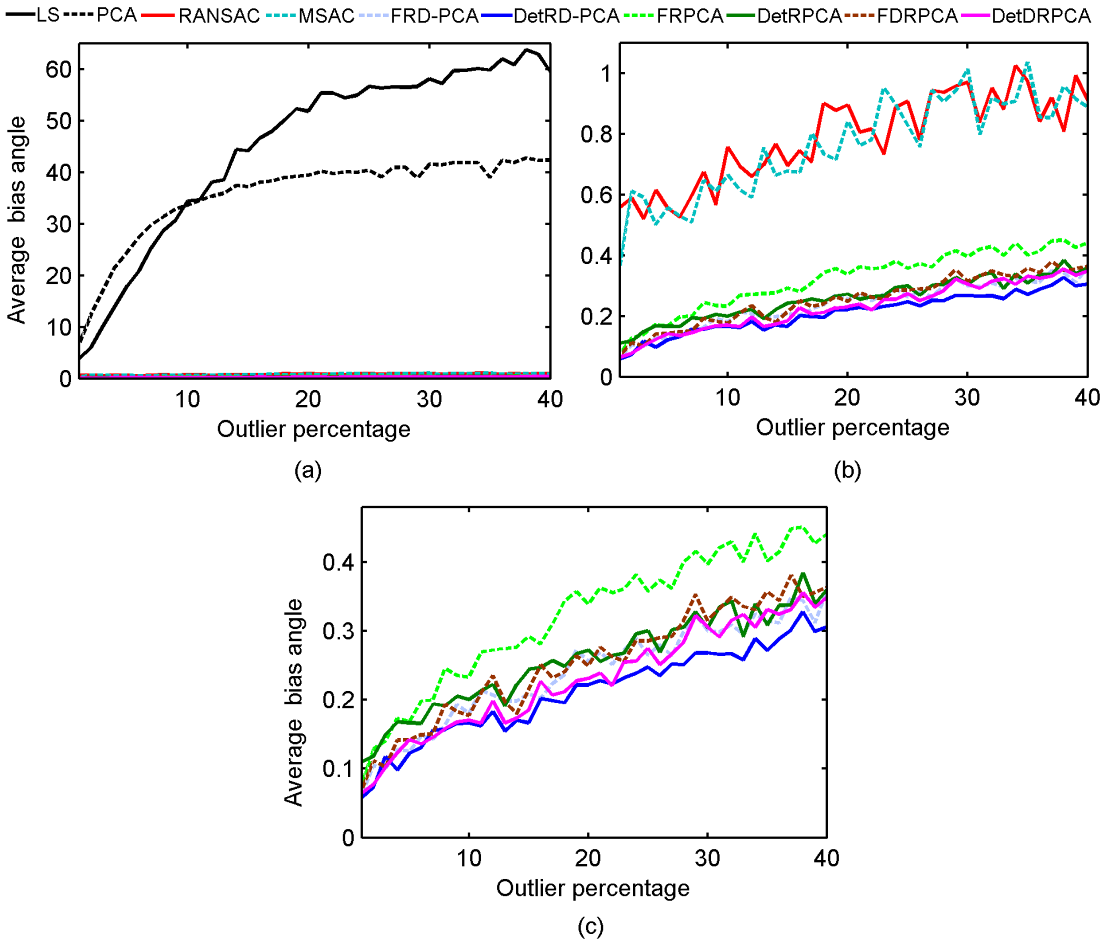


Figure 3.7 Average bias angles versus outlier percentage; $n = 100$ and outlier percentage = 1%–40%: (a) all methods, (b) all methods (except LS and PCA), and (c) robust statistical methods (except LS, PCA, RANSAC and MSAC).

To explore underlying robustness pattern of the results, we use boxplots (Figure 3.8) in which the boxes enclose the middle half of the results, i.e. the length of a box is the interquartile range with the ends of the box at the 1^{st} and 3^{rd} quartiles. The line across the box is the position of the median, and the ends of the whiskers show the minimum and the maximum extreme values of the non-outlying results. The '+' signs represent the outlying results. In Figure 3.8, boxplots are created for different methods based on the bias angles (used in Table 3.1) from 1000 runs for the data of 100 simulated points including 20% outliers. It clearly shows significantly better robustness of the statistical robust methods than LS, PCA, RANSAC and MSAC, and supports the findings in favour of robust statistical methods from Figure 3.7.

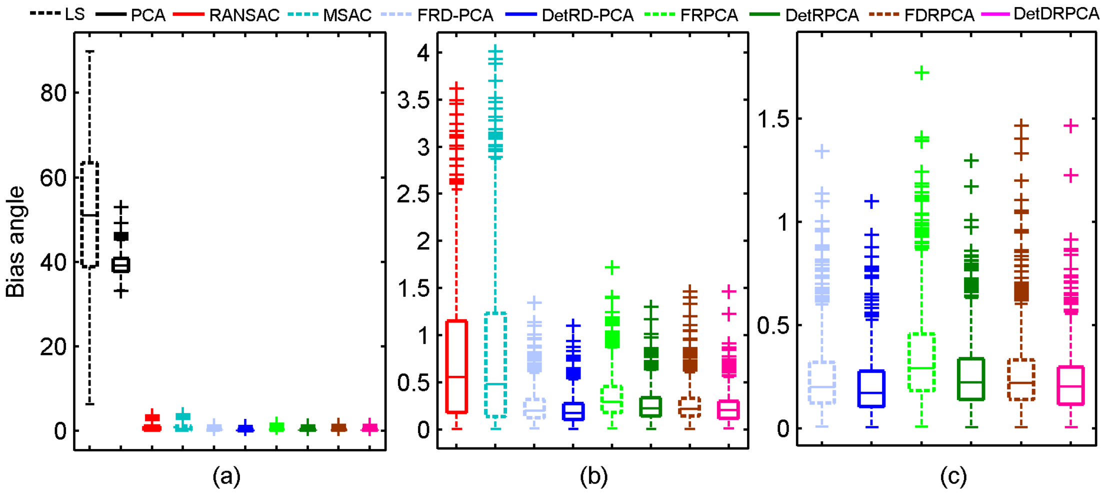


Figure 3.8 Boxplots for bias angles; $n = 100$ and outlier percentage = 20%, (a) all methods, (b) all methods except LS and PCA, (c) only robust statistical methods.

3.5.2.2 Outlier Influence and Sample Size on Bias Angle

We investigate the effect of different percentages of outlier contamination and sample size on bias angle, so we generate datasets for various sample sizes ($n = 20$, 50 and 200) and outlier percentages (1% to 40%). We perform 1000 runs for each and every outlier percentage and sample size. Since LS and PCA (Table 3.1, Figures 3.7 and 3.8) perform poorly they are ignored in this analysis and we concentrate only on the robust methods.

Figure 3.9 shows the results for average bias angles (in degrees) from 1000 datasets. In Figure 3.9a, for a small sample of size 20, we see RANSAC, MSAC and DetRPCA give inconsistent results for outlier percentages around 25% and more. This figure also shows that even for low point density and in the presence of a high percentage of outliers, DetRD-PCA performs better than the others. It is seen that with the increasing sample size, the robust statistical methods give better results (i.e. less bias angles) than RANSAC and MSAC. For every outlier percentage and sample size, DetMCD based methods perform better than the respective FMCD based methods, meaning DetRD-PCA, DetRPCA and DetDRPCA produce more accurate results than FRD-PCA, FRPCA and FDRPCA, respectively. DetRD-PCA has the least bias angle for every outlier percentage and sample size.

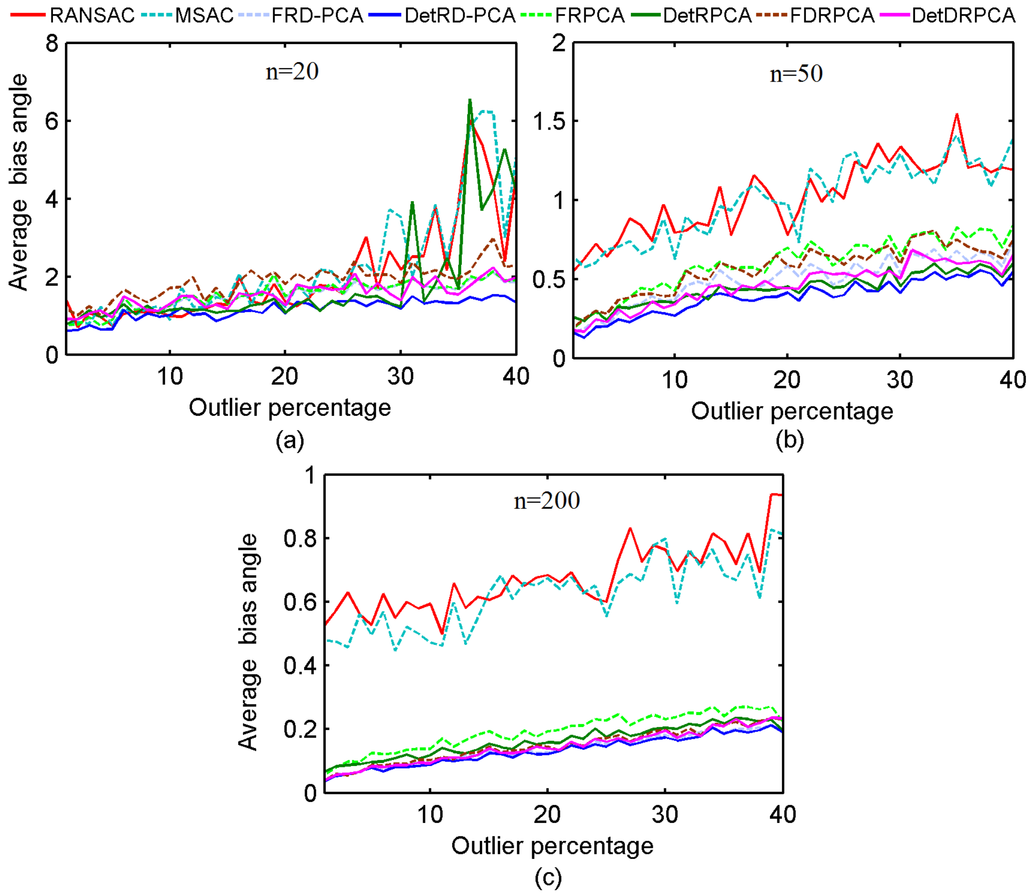


Figure 3.9 Average bias angles versus outlier percentages (1% to 40%): (a) $n = 20$, (b) $n = 50$, and (c) $n = 200$.

3.5.2.3 Statistical Significance Test

In Table 3.1, we see that sometimes there is much variability and sometimes very low difference between the average bias angle values from different methods. We explore the differences to determine if there is any statistically significant difference between the relevant pairs of methods. Since, we cannot guarantee that the bias angle values follow the normality assumption, we use the non-parametric 'Wilcoxon Signed Rank' statistical significance test (Sheskin, 2004; Hollander et al., 2014) based on the information from Table 3.1 and the relevant bias angles from 1000 runs. This test procedure, which is equivalent to the parametric 'dependent t-test' (Sheskin, 2004; Hollander et al., 2014) verifies the difference between two medians (in Column 7, Table 3.1) from two different methods (i.e. populations), in Columns 1 and 2 in Table 3.2. We test the null hypothesis H_0 w.r.t. the alternative hypothesis H_a:

- H_0–there is no significant difference between two medians from two methods.
- H_a–there is some difference between two medians i.e. the two methods perform significantly different.

Table 3.2 shows the results from the 'Wilcoxon Signed Rank' test obtained by using SPSS® software. We perform the test at the 5% level of significance α. Therefore, we may reject H_0 if the calculated α-value (Column 3 in Table 3.2) is less than 0.05, otherwise we may retain H_0. We see only four pairs: (i) RANSAC, MSAC (ii) FRD-PCA, DetRD-PCA (iii) FRPCA, DetRPCA and (iv) FDRPCA, DetDRPCA that retain H_0. That is, within the pairs they do not have statistically significant differences, because respective significant values exceed the significance level 0.05. Therefore, based on the test results, we may reach the decision: the four pairs perform similarly to each other, and the rest of the pairs have significant differences in plane fitting.

Table 3.2 Statistical significance test.

Methods		Significance (α-value)	Decision
LS	PCA	0.000	Reject H_o
RANSAC	MSAC	**0.447**	Retain H_o
FRD-PCA	DetRD-PCA	**0.894**	Retain H_o
FRPCA	DetRPCA	**0.679**	Retain H_o
FDRPCA	DetDRPCA	**0.077**	Retain H_o
PCA	RANSAC	0.000	Reject H_o
PCA	MSAC	0.000	Reject H_o
MSAC	DetRD-PCA	0.000	Reject H_o
MSAC	DetRPCA	0.000	Reject H_o
MSAC	DetDRPCA	0.000	Reject H_o
DetRD-PCA	DetRPCA	0.000	Reject H_o
DetRD-PCA	DetDRPCA	0.000	Reject H_o
DetRPCA	DetDRPCA	0.000	Reject H_o
RANSAC	DetRD-PCA	0.000	Reject H_o
RANSAC	DetRPCA	0.000	Reject H_o
RANSAC	DetDRPCA	0.000	Reject H_o

For the pairs in which H_0 are rejected, one method significantly performs better than the other. For example, PCA is better than LS, and RANSAC is better than PCA. For these cases the decisions are: reject H_0 (between the pairs they have significant median difference), and at the same time from Table 3.1 we get median (LS) > median (PCA), and median (PCA) > median (RANSAC).

Similarly, results from Table 3.1 and Table 3.2 illustrate that robust statistical methods perform significantly better than RANSAC and MSAC.

In the remainder of this chapter, for brevity and for better evaluation, we just consider PCA, RANSAC, MSAC and the DetMCD based methods (DetRD-PCA, DetRPCA and DetDRPCA) for comparison and performance evaluation.

3.5.2.4 Point Density Variation

To see the effect of point density variations on bias angle, we create datasets with different variations in surface directions (x-y axes) and in elevation (z axis). Point density is defined as the number of points that occurs in a specific unit volume. To generate datasets of different density, we keep the data size the same but change the variances of the Gaussian distribution from where the data have been drawn randomly. That is, a large variance in the point distribution gives low point density and vice versa. The size of the volume is considered in the surface directions (x and y). The rows of Table 3.3 contain six sets of variance combinations for regular R and outlier O data in the x and y directions. We simulate 1000 sets of 100 points with 20% outliers for every variance (I to VI, Table 3.3). Other parameters are the same as for the previous experiments. Figure 3.10a shows that PCA produces larger bias angles than the robust methods, and Figure 3.10b shows that all three deterministic MCD based methods, i.e. diagnostic PCA (DetRD-PCA), robust PCA (DetRPCA) and diagnostic robust PCA (DetDRPCA) have smaller bias angles than RANSAC and MSAC. In spite of the changes of point density, robust statistical methods produce more consistent results. The performance of DetRD-PCA and DetDRPCA are better than the others and show similar efficiency to each other.

Surface roughness may influence the surface fitting methods and can change the estimates. We calculate the bias angles for different methods for similar data generated as described in the previous experiments with different z-variances (0.001, 0.01, 0.02, 0.05 and 0.1) for regular observations. With increasing z-variances, results in Figures 3.10(c and d) show that DetRD-PCA, DetRPCA and DetDRPCA perform significantly better than the others, and produce more consistent results than PCA, RANSAC and MSAC.

Table 3.3 Variances for regular R and outlier O data.

Datasets	I	II	III	IV	V	VI
x(R,O) variances	(3,3)	(5,5)	(7,7)	(9,9)	(11,11)	(15,15)
y(R,O) variances	(3,3)	(5,5)	(7,7)	(9,9)	(11,11)	(15,15)

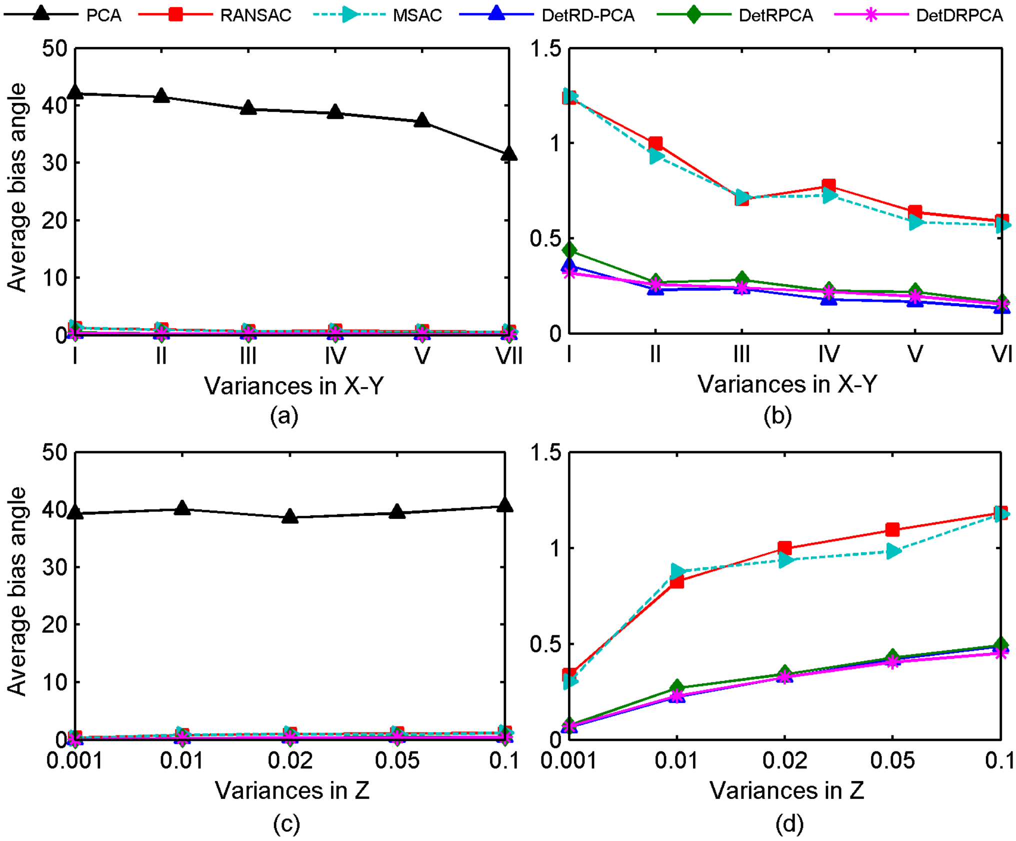


Figure 3.10 Average bias angle w.r.t. point density variation in x-y: (a) all methods, and (b) robust methods. Average bias angle w.r.t. z-variation: (c) all methods, and (d) robust methods.

3.5.2.5 Classification into Outliers and Inliers

We consider the robust methods: RANSAC, MSAC, DetRD-PCA, DetRPCA and DetDRPCA as classifiers that can group data into outliers and inliers. To show their performance as classifiers, we generate 100 datasets of 100 points including 20% outliers. We run the experiment for each dataset and calculate the number of correctly identified outliers and inliers. Figure 3.11 shows histograms of the number of inliers identified over all the runs. It shows that most of the time the three DetMCD based methods identify allmost all inliers correctly as the histograms are centered around 80 inliers. Figures 3.11(c–e) show that DetRD-PCA and DetRPCA identify inliers more accurately than DetDRPCA. In Figures 3.11(a and b), RANSAC and MSAC identify low percentages (around 20 to 40 out of the 80 inliers) inliers, i.e. they falsely show the majority of inliers as outliers. This is the well-known swamping effect. Swamping occurs when good observations are incorrectly identified as outliers because of the presence of another, usually remote, subset of observations (Hadi and Simonoff, 1993). The swamping effect can be considered as the False Positive Rate (FPR). We calculate True Positive Rate (TPR), also known as Sensitivity, True Negative Rate (TNR), FPR and 'Accuracy' defined by Fawcett (2006) and Sokolova et al. (2006):

- TPR=$\frac{\text{number of outliers correctly identified}}{\text{total number of outliers}} \times 100$,
- TNR=$\frac{\text{number of inliers correctly identified}}{\text{total number of inliers}} \times 100$,
- FPR=$\frac{\text{number of inliers identified as outliers}}{\text{total number of inliers}} \times 100$,
- Accuracy=
  $\frac{\text{number of correctly identified outliers} + \text{number of correctly identified inliers}}{\text{total number of points}} \times 100.$

Results in Table 3.4 show that RANSAC and MSAC correctly identify outliers but they wrongly identify many inliers as outliers, i.e. RANSAC and MSAC are highly affected by the swamping (FPR) phenomenon. RANSAC and MSAC perform similarly in terms of swamping and accuracy. On the other hand, DetRD-PCA and DetRPCA have more than 97% accuracy with 3.31% and 3.69% swamping rate, respectively. DetDRPCA is significantly better than RANSAC and MSAC with 92.87% accuracy and 100% sensitivity.

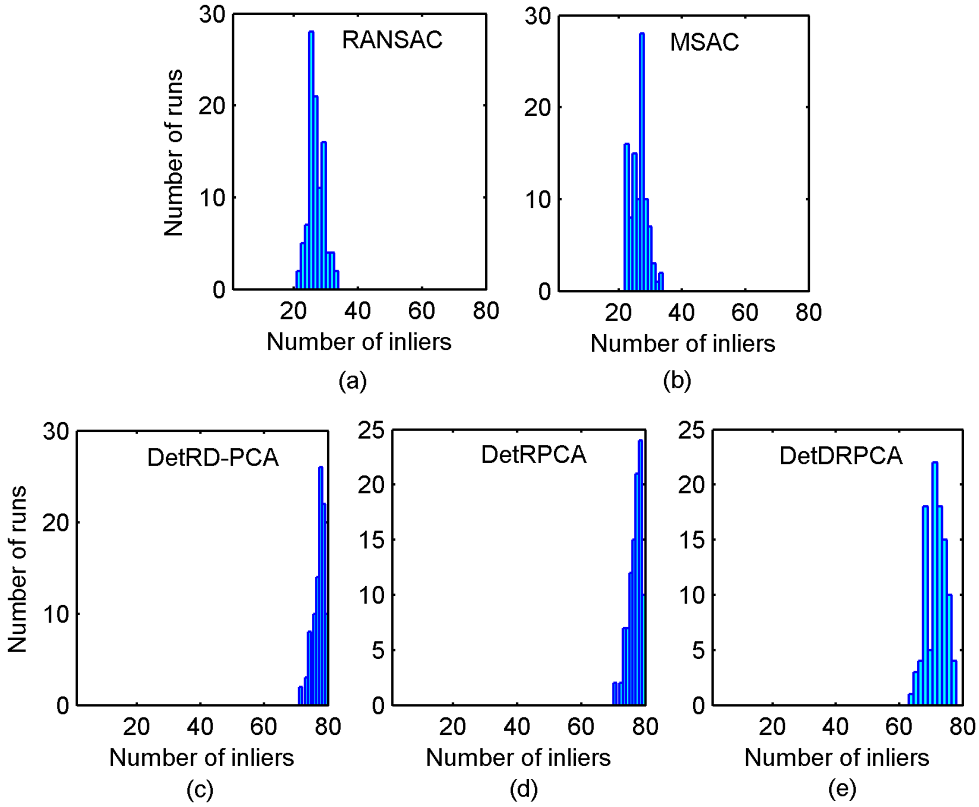


Figure 3.11 Histograms for number of correctly identified inliers.

Table 3.4 Classification performance.

Measures	RANSAC	MSAC	DetRD-PCA	DetRPCA	DetDRPCA
TPR (Sensitivity)	100.00	100.00	100.00	100.00	100.00
FPR (Swamping)	67.38	67.50	3.31	3.69	8.91
Accuracy	46.10	46.00	97.35	97.05	92.87

3.5.3 Laser Scanner Data

In this section, results are presented for the plane fitting methods on Mobile Laser Scanning (MLS) data. The data was captured using a system developed by a local survey company. The data has been collected by a vehicle moving at traffic speed. The system's rotating laser collects points along road corridors measuring everything visible to the scanner within a 30m range of the scanner. The data has been post-processed into x, y and z coordinates and has a positional accuracy of approximate 0.015m and a point precision of 0.006m.

This section illustrates that the saliency features (normal, least eigenvalue and curvature) determined by the proposed methods make substantial improvements to existing methods and algorithms used for point cloud processing (e.g. edge detection and segmentation). The performances of the estimated plane parameters and saliency features from the methods are evaluated in the context of applications of (i) plane fitting, (ii) sharp feature preservation and surface edge detection, and (iii) segmentation. We use two algorithms for: (i) sharp feature extraction/recovery, which is introduced as a classification algorithm (separation of points into edge/corner points and surface points) in Chapter 5 (earlier version published in Nurunnabi et al., 2012c), and (ii) region growing based segmentation, which is proposed in Chapter 5 (earlier version published in Nurunnabi et al., 2012d). The algorithms are briefly described as follows.

***Classification*:** The classification algorithm adopts the idea of outlier detection and considers sharp features as outliers having larger values of λ_0 (variation along the surface/plane normal). The algorithm estimates λ_0 values for all the points in the data based on their local neighbourhood Np_i of size k. The i^{th} point is identified as an outlier i.e. as an edge/corner point if:

$$\lambda_0 > \bar{\lambda}_0 + a \times \mathrm{StD}(\lambda_0), \tag{3.9}$$

where $\bar{\lambda}_0$ and $\mathrm{StD}(\lambda_0)$ are the mean and standard deviation (StD) of λ_0, and $a = 1$ or 2 or 3 (i.e. 1, 2, or 3 StD from the mean) based on knowledge of the data.

***Segmentation*:** Segmentation is the process of labelling a point cloud into homogeneous regions. This is useful for surface reconstruction, object detection and modelling. Generally, region growing based segmentation algorithms begin by searching for a seed point, assuming that appropriate seed point selection gives better segmentation results. Therefore, according to the segmentation algorithm, we fix a neighbourhood size k, and fit planes on the local neighbourhood using the proposed algorithms and estimates necessary local saliency features such as normals and curvatures for each point in the data. We choose the first seed point as the one with the lowest curvature value in the data, and then grow a region using local surface point proximity (distance between two points) and the coherence criteria (e.g. normal) based on the k nearest neighbourhood Np_i of the i^{th} seed point p_i. The algorithm considers

Orthogonal Distance (OD) for the i^{th} seed point to its best-fit-plane, Euclidean Distance (ED) between the seed point p_i and one of its neighbours p_j, and the angle difference θ between the seed point p_i and the neighbours p_j defined in Eq. (3.7). The angle is calculated from the unit normals at p_i and p_j. The region grows from the seed point p_i by adding one of its neighbours p_j, if they have OD, ED, and θ less than their respective pre-assigned thresholds. It then iterates by considering the neighbours of each of the new points added to the region until no more points can be added. The regions that have a size more than or equal to a minimum number of points R_{min} will be considered as the final segments for the data. The process of segmentation continues with further seed points until all the points in the point cloud have been processed. Further details are given about the segmentation Algorithm 5.4 in Chapter 5.

3.5.3.1 Plane Fitting

Dataset 3.1: Road scene dataset

We consider our first real MLS dataset shown in Figure 3.12a, a road scene including a lamp post along with a sign (zoomed-in boxes), which looks unclear because of the presence of vegetation around the sign. We name this the 'road scene' dataset. We extract the sign (Figure 3.12b, front view and Figure 3.12c, side view). This data may be regarded as a planar surface. In Figure 3.12c, we see some points created by vegetation that are not on the plane and can be considered as outliers. We use PCA to fit the planar surface. Figure 3.13a shows the points used to fit a plane using PCA (in blue), which deviate from the original points (in green). Figure 3.13b shows the fitted and extracted plane contains outliers projected onto the 2D approximation. The planar surface was not correctly estimated by PCA. The outliers are to the right of the diagram and the correct points to the left of the diagram. That means, the outliers appear as inliers in the PCA determined plane, which clearly shows the masking effect caused by the presence of multiple outliers. Figure 3.13c shows the fitted plane (in blue) using Deterministic MCD based diagnostic PCA (DetRD-PCA). The outlying points are identified accordingly and the fitted plane (in blue) is in the right direction. The points classified as part of the extracted plane using DetRD-PCA are shown in Figure 3.13d.

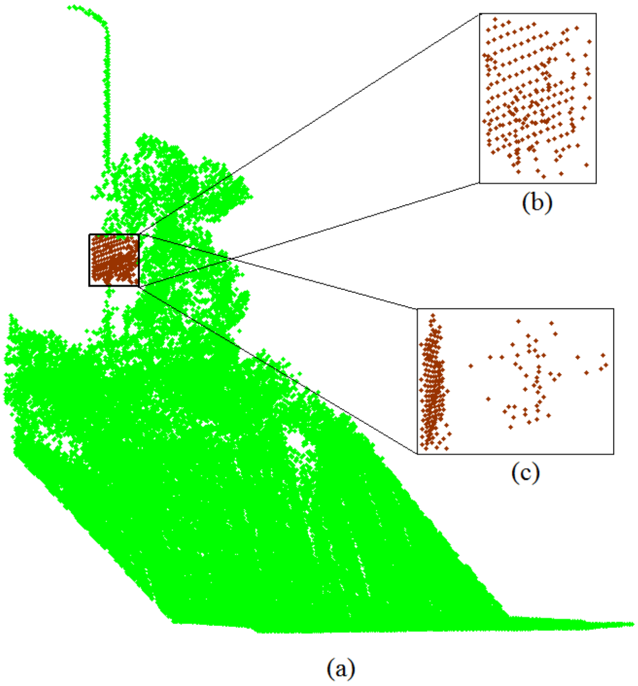


Figure 3.12 (a) Road scene dataset with a road sign in the zoomed-in boxes: (b) road sign front view, and (c) road sign side view.

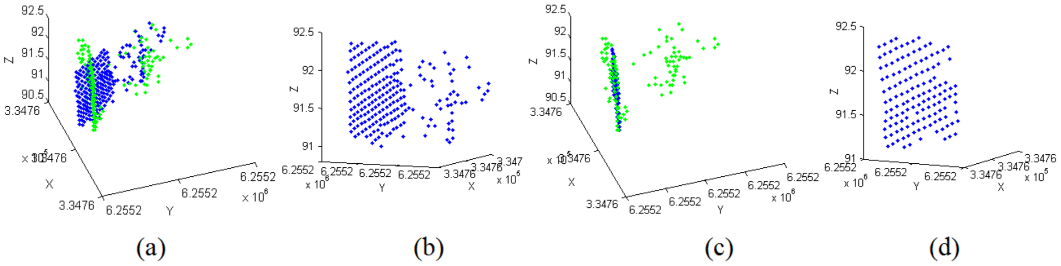


Figure 3.13 (a) Plane (blue) orientation by PCA, (b) fitted/extracted plane by PCA, (c) plane (blue) orientation by robust method (DetRD-PCA), and (d) fitted/extracted plane by robust method (DetRD-PCA).

3.5.3.2 Sharp Feature Preservation

It is known that a more accurate plane fit produces more accurate surface normals. Reliable and accurate normals are required to detect and recover sharp features, e.g. lines, edges and corners (Li et al., 2010). Sharp features such as edges and corners can delineate surface patches and are useful for accurate surface reconstruction. Many algorithms have been developed for sharp feature preservation (Fleishman et al., 2005; Li et al., 2010; Weber et al., 2012; Wang et al., 2013). This task is not trivial because of the possible presence of outliers/noise in the data. In this section we will show that our plane fitting algorithms produce reliable and robust normals, which is beneficial for sharp feature preservation.

It is observed that the normals on or near sharp features become overly smooth mainly because of two reasons: (i) neighbourhood points may be present locally from two or more surfaces (Figures 3.14a and b), and (ii) presence of outliers/noise (Figures 3.14c and d) in the local neighbourhood. In Figures 3.14(a and c), the PCA method counts all the points for plane fitting in a local neighbourhood (dotted circle), misrepresenting the normal at the vertex and smoothing out the sharp feature. The robust statistical methods used in the new algorithms group the majority of points that are homogeneous w.r.t. defined characteristics e.g. points that lie on or near a plane. Non-homogeneous points are regarded as outliers. Hence, the fitted plane would be the best-fit-plane for the region of the majority of points without outliers and the estimated normal represents the surface consisting of the majority of points. In Figures 3.14(b and d) robust/diagnostic methods (e.g. DetRD-PCA) consider the majority of points (magenta) excluding outliers and fit a plane and correctly estimate the normal without the influence of outliers. We see that robust normals (magenta) are correctly estimated on the corner (Figure 3.15a) and on an edge (Figure 3.15b) points but non-robust results using PCA (blue) fail to do so. A small amount of MLS point cloud data sampling a sharp edge, is shown in Figure 3.16a. DetRD-PCA (Figure 3.16c) results in normals that preserve that sharp transition while PCA results (Figure 3.16b) in smoothly changing normals (in the black circle).

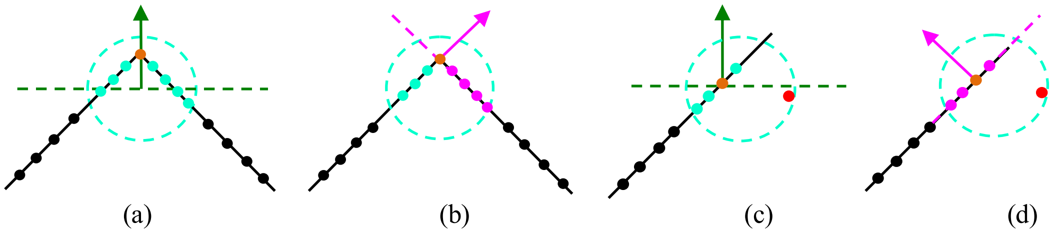


Figure 3.14 Neighbouring points in the dashed cyan circle are from two planar regions: (a) PCA plane (green dotted line) and normal (green arrow), and (b) robust plane (magenta dotted line) and normal (magenta arrow). Neighbouring points in a circle (cyan) include a noise point (red dot): (c) PCA plane (green dotted line) and normal (green arrow), and (d) robust plane (magenta dotted line) and normal (magenta arrow).

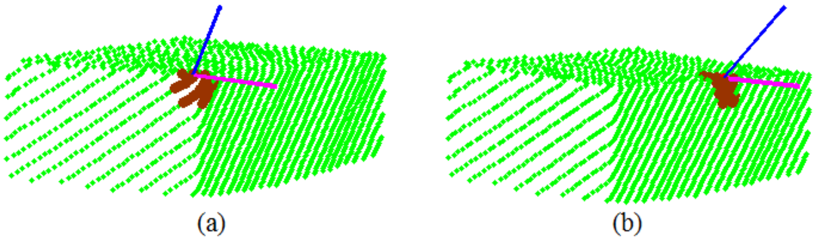


Figure 3.15 PCA normals are blue and robust normals are magenta, maroon points are the local neighbouring points: (a) normals on a corner point, and (b) normals on an edge point.

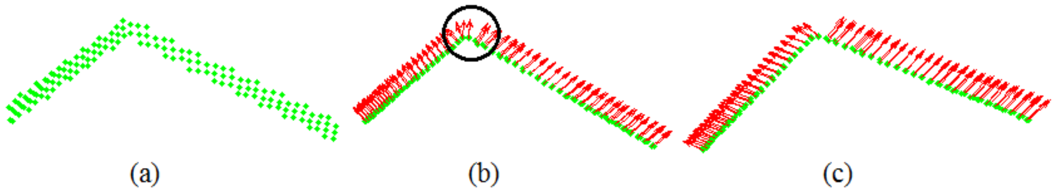


Figure 3.16 (a) Real point cloud data, normals on sharp region: (b) results using PCA, and (c) results using a robust method.

Dataset 3.2: Road-kerb-footpath and crown datasets

To show the performance for sharp feature detection, we take two small sets of vehicle based MLS data. One contains part of a road, kerb and footpath (Figure 3.17a) consisting of 11,774 points, called the 'road-kerb-footpath' dataset, that has sharp edges. The other dataset is a part of a roof crown extracted from a roadside building (Figure 3.17b), that we called the 'crown' dataset. This consists of 14 different regions and contains 3118 points. The crown dataset is a polyhedron consisting of edges, corners and bilinear surfaces with common edges. We know the angle of the tangent planes for bilinear surfaces varies along the edge. The case of varying angles in sharp features is important in real life datasets and could cause problems for feature detecting and reconstructing systems using global sets of parameters (Weber et al., 2012). To extract the sharp features, we use the classification algorithm in Chapter 5 (earlier version is in Nurunnabi et al., 2012c) to fit a plane to a local neighbourhood of size $k = 40$ at each point in the cloud. We choose the value of k based on similar real data experimentation. We calculate the λ_0 values and classify the points into inliers (surface points) and outliers (edge or corner points) according to Eq. (3.9), where $a = 1$. Classification results are in Figures 3.18 and 3.19 for the road-kerb-footpath dataset and the crown dataset respectively.

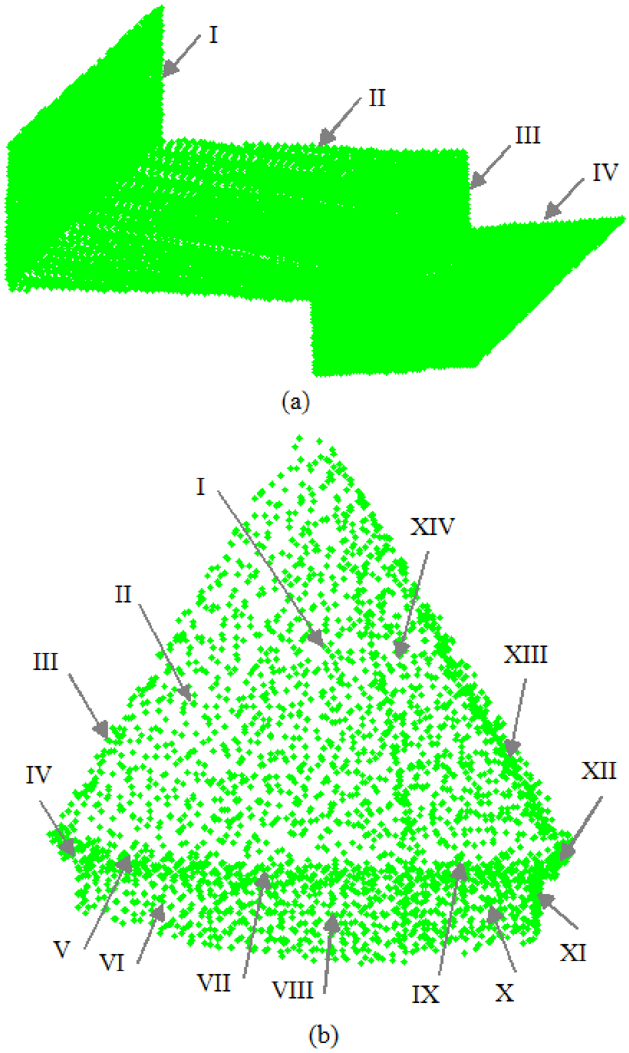


Figure 3.17 MLS point clouds: (a) road-kerb-footpath dataset, and (b) crown dataset.

The results for the two datasets show that PCA fails to recover the sharp features (edge/corner points). Although RANSAC and MSAC are robust methods, they do not successfully classify surface, edges and corners. Many surface points (e.g. in regions I, II and III of the road-kerb-footpath dataset) appear as edge points. Figures 3.18 and 3.19 show that the proposed DetRD-PCA, DetRPCA and DetDRPCA methods are more accurate than PCA, RANSAC and MSAC. Figures 3.19(d–f) show the proposed methods efficiently recover sharp features even for the crown dataset in the presence of bilinear surfaces. In Figure 3.18f, DetDRPCA underestimates the number of edge points. DetRD-PCA and DetRPCA are competitive in terms of accuracy, but DetRD-PCA takes less time than DetRPCA.

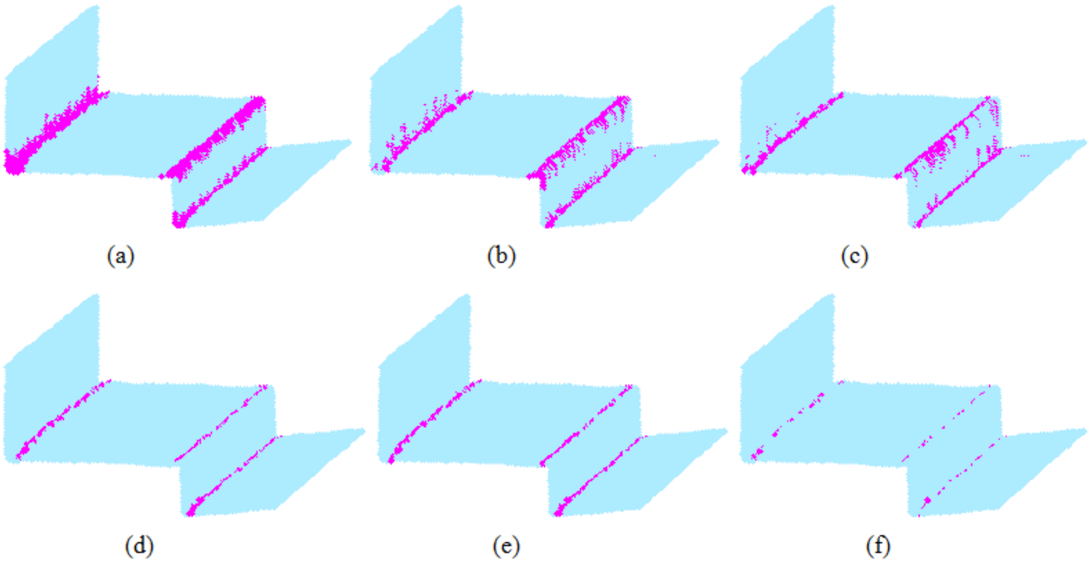


Figure 3.18 Edge points (in magenta) recovery for road-kerb-footpath dataset: (a) PCA, (b) RANSAC, (c) MSAC, (d) DetRD-PCA, (e) DetRPCA, and (f) DetDRPCA.

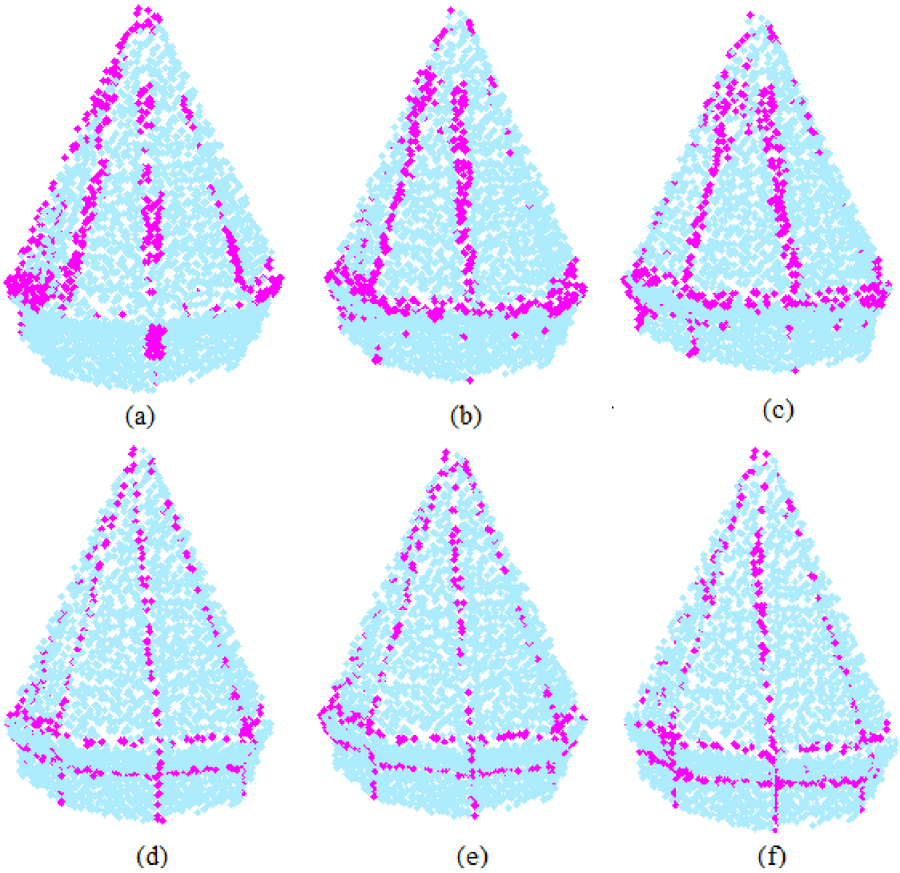


Figure 3.19 Edge and corner points (in magenta) recovery for crown dataset: (a) PCA, (b) RANSAC, (c) MSAC, (d) DetRD-PCA, (e) DetRPCA, and (f) DetDRPCA.

3.5.3.3 Segmentation

We compute the saliency features: normals and curvatures by different existing (PCA, RANSAC and MSAC) and the DetMCD based proposed methods (DetRD-PCA, DetRPCA and DetDRPCA) to evaluate and compare them for segmentation using the region growing algorithm described earlier (more in Chapter 5). To see the robustness for the estimated curvatures of the seed points for region growing for the different methods, boxplots were generated by the curvatures values for the crown dataset with neighbourhood size 40. We fix the neighbourhood size based on similar data experimentation. Results from the boxplots in Figure 3.20 show that DetRD-PCA, DetRPCA and DetDRPCA produce more robust curvature values than PCA, RANSAC and MSAC, because the lengths of the boxes of the proposed methods are the least, and it is also revealed that most of the values of curvature are comparatively less than the existing methods. Figure 3.20 shows the proposed methods arranged in order of their superiority in robustness from right to left: DetRD-PCA, DetRPCA and DetDRPCA.

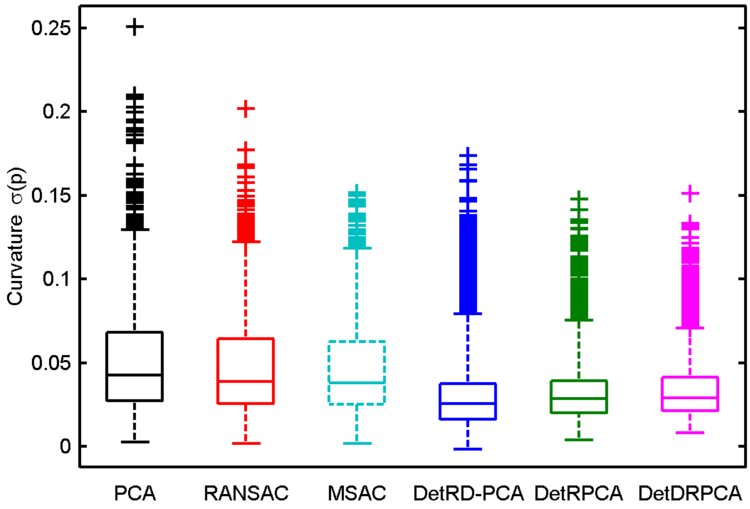


Figure 3.20 Boxplots of curvature values for crown dataset.

We use the segmentation algorithm proposed in Chapter 5 (an earlier version published in Nurunnabi et al., 2012d), briefly described previously that uses curvatures and normals. The segmentation results from the different methods are evaluated using two MLS datasets consisting of planar and non-planar complex object surfaces.

Dataset 3.2: Crown dataset

To perform the segmentation, we consider the crown dataset (Figure 3.17b) first. We set the required parameters: $k = 40$, angle threshold $\theta_{th} = 6°$, and minimum region size $R_{min} = 10$. Segmentation results are in Figure 3.21 and summarized in Table 3.5. In Figure 3.21(a), results show that PCA produces the worst results and failed to segment most of the different surfaces properly. We see in Table 3.5, PCA has only two correct or Proper Segments (PS) with errors consisting of two and seven Over Segments (OS) and Under Segments (US) respectively. A PS is identified as a true segment from manually determined ground truth i.e. one segment describes a single feature such as the wall of a house that is one planar surface. An OS occurs where one true segment is broken into two or more separate segments, and an US is where more than one true segment are wrongly grouped together as one segment. RANSAC and MSAC (Figures 3.21b and c) give similar results as both have three PS, three OS and six US. Using the normals and curvatures from the proposed robust statistical methods, the same segmentation algorithm performs well. DetRD-PCA (Figure 3.21d) and DetRPCA (Figure 3.21e) both properly segment all 14 regions without any OS and US, and DetDRPCA (Figure 3.21f) has 13 PS and two OS.

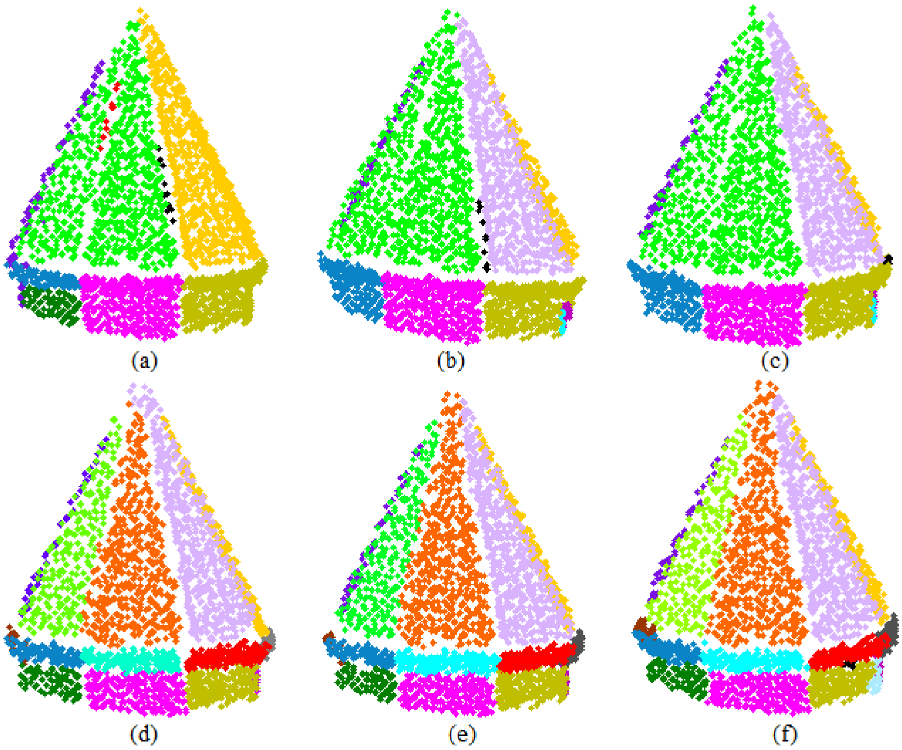


Figure 3.21 Segmentation results for the crown dataset: (a) PCA, (b) RANSAC, (c) MSAC, (d) DetRD-PCA, (e) DetRPCA, and (f) DetDRPCA.

Dataset 3.3: Traffic furniture dataset

The second dataset is also MLS data and acquired in a similar fashion as the first one. The data (Figure 3.22) is from the side of a road, and contains a lamp post, sign posts and road ground surfaces. We call this the 'traffic furniture' dataset. The data consists of 31,388 points and includes 16 different planar and non-planar complex object surfaces. An example of a complex surface is the incomplete cylindrical surface (see the selected box and inset Figure 3.22). We set parameters for the segmentation algorithm: $k = 50$, $\theta_{th} = 15°$, and $R_{min} = 10$. The parameters are fixed based on an empirical study on similar real data for segmentation. Figure 3.23 shows the quality of the segmentation results from different methods. In Table 3.5, the results for the traffic furniture dataset show the segmentation based on PCA (Figure 3.23a), RANSAC (Figure 3.23b) and MSAC (Figure 3.23c) are not accurate, as they are influenced by over and under segmentation. PCA and RANSAC have nine PS with seven OS and two US. Figures 3.23(d–f) show that the three sets of Deterministic MCD based segmentation results from DetRD-PCA, DetRPCA and DetDRPCA respectively are accurate without any OS and US. That means the normals and curvatures estimated from the proposed diagnostic, robust and combined diagnostic robust methods, are more reliable, robust and accurate than the other methods for segmentation.

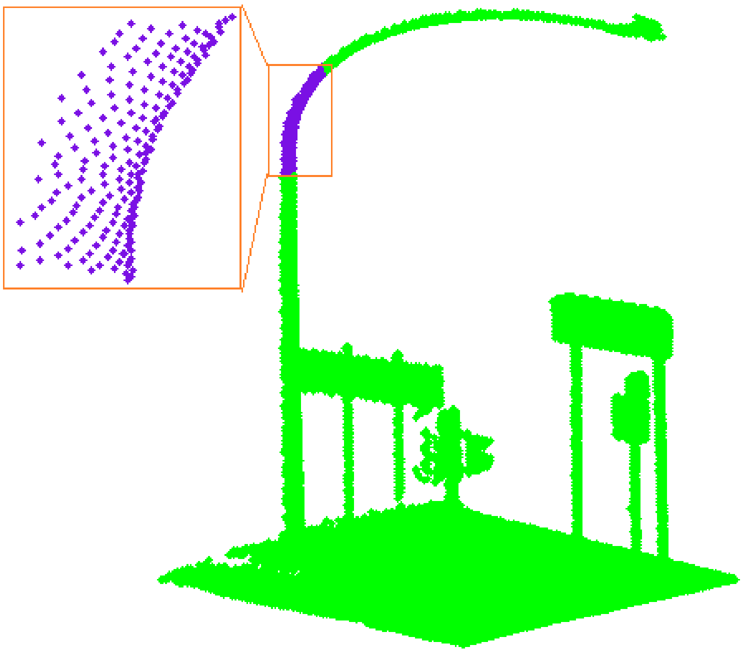


Figure 3.22 Traffic furniture dataset.

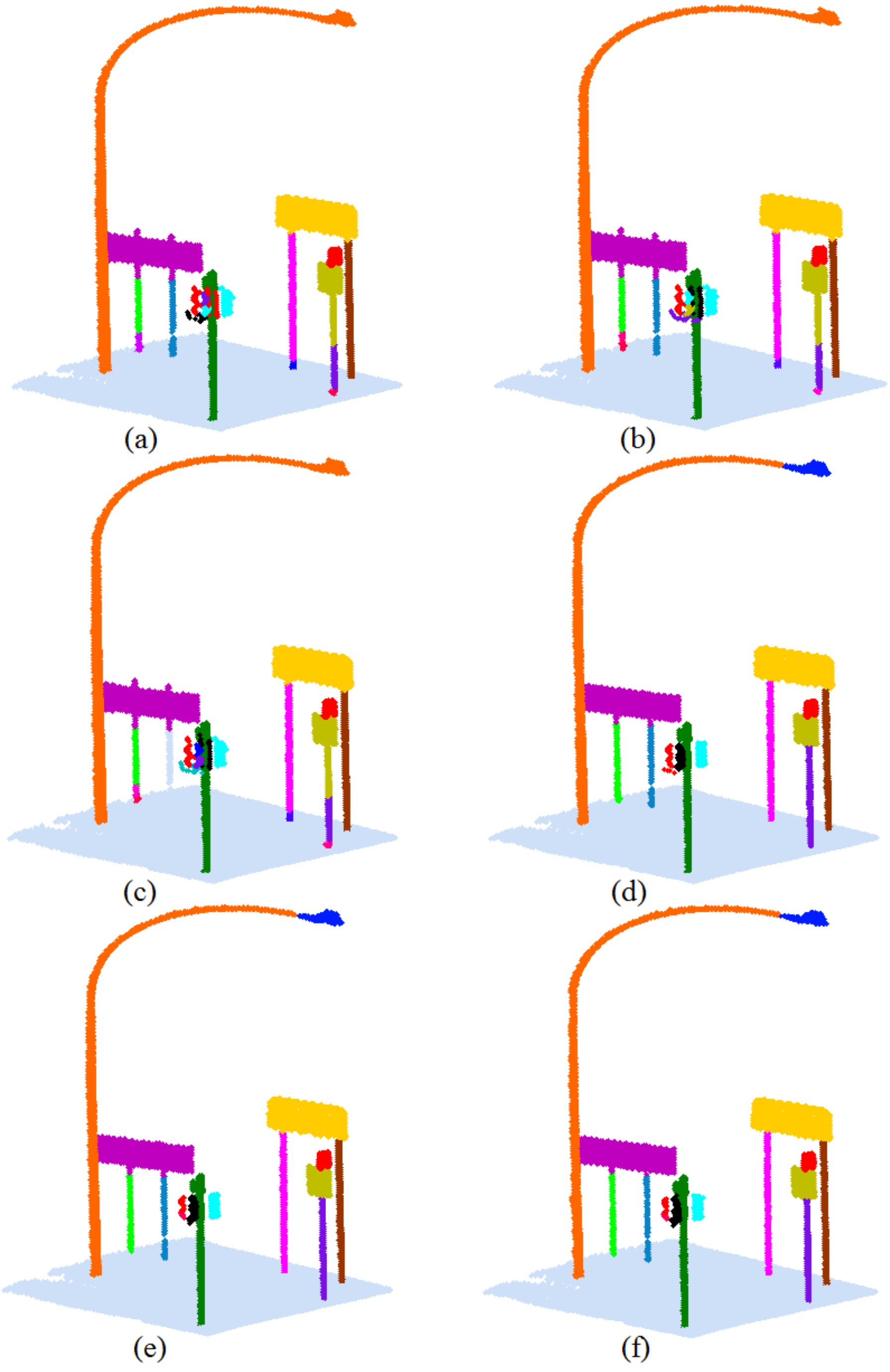

(a)
(b)
(c)
(d)
(e)
(f)

Figure 3.23 Segmentation results for traffic furniture dataset: (a) PCA, (b) RANSAC, (c) MSAC, (d) DetRD-PCA, (e) DetRPCA, and (f) DetDRPCA.

Table 3.5 Segmentation results for crown and traffic furniture datasets.

Methods	Crown dataset				Traffic furniture dataset			
	TS	PS	OS	US	TS	PS	OS	US
PCA	9	2	2	7	21	9	7	2
RANSAC	10	3	3	6	21	9	7	2
MSAC	10	3	3	6	20	7	7	3
DetRD-PCA	14	14	0	0	16	16	0	0
DetRPCA	14	14	0	0	16	16	0	0
DetDRPCA	16	13	2	0	16	16	0	0

3.6 Computational Speed and Effort

For many algorithms, there is a trade-off between the time taken to perform an algorithm and the accuracy of the results. In this chapter, so far we have been solely interested in the accuracy and robustness of the results. It has been demonstrated in the previous sections that the robust methods produce significantly better results than the classical methods in terms of accuracy and are able to reduce the influence of outliers. Therefore in this section, we compare the computational speed only for the robust methods. The main issue of the MCD algorithm is it is computationally intensive. The FMCD and DetMCD algorithms were developed to increase the computational efficiency of the MCD algorithm without loss of accuracy and robustness of the estimators. Hubert et al. (2012) demonstrated the computational efficiency of DetMCD over FMCD. We use MATLAB® as a common platform of the respective algorithms. We investigate the computational efficiency empirically for the proposed algorithms that use DetMCD estimators and compare them with FMCD based methods along with RANSAC and MSAC using existing MATLAB® functions. RANSAC and MSAC algorithms used Zuliani's RANSAC toolbox (Zuliani, 2011) and the necessary functions for FMCD and DetMCD based algorithms are implemented using the MATLAB® library for a robust analysis (Hubert et al., 2005, 2012).

To evaluate the computational speed of the proposed algorithms for plane fitting, we simulate datasets as for the previous experiments in Section 3.5.2 with different sample sizes 20, 50, 100, 500, 1000 and 10000, with 20% outliers. We fix the different sample sizes because we know computation speed is a function of data size, and the rate of outlier contamination is assumed 20% just as a representative value because usually the outlier contamination in point cloud data is not more than 50%. We simulate each of the dataset 1000 times. Results in Table 3.6 are the average times in seconds for plane fitting calculated using the MATLAB® profile function. Results show that every variant of the DetMCD based method is significantly faster than the respective FMCD based method. For example, for a sample size of 50, FRD-PCA takes 0.8151s and DetRD-PCA takes 0.0271s, which is 30 times faster, whereas RANSAC takes 0.1278s, which is 4.72 times slower than DetRD-PCA. In the case of a sample size of 10000, DetRD-PCA fits a plane in 0.4469s, which is 2.56 and 5.35 times faster than FRD-PCA (1.1456s) and RANSAC (2.3894s), respectively. MSAC takes a little more time than RANSAC, and DetDRPCA takes more time than DetRD-PCA and DetRPCA. Therefore, it shows that the DetMCD based methods are faster than the FMCD based methods for all data sizes. The DetMCD algorithms are faster for small samples and have the advantage that they will reduce the computation time for any local neighbourhood based point cloud processing tasks that use the local saliency features: normals and curvatures.

Table 3.6 Plane fitting time (in seconds).

Methods	Sample size					
	20	50	100	500	1000	10000
RANSAC	0.0694	0.1278	0.1717	0.3177	0.4355	2.3894
MSAC	0.0772	0.1473	0.1867	0.3481	0.4667	2.5380
FRD-PCA	0.8206	0.8151	0.8399	0.9680	1.0461	1.1456
DetRD-PCA	0.0250	0.0271	0.0326	0.0585	0.1355	0.4469
FRPCA	0.8201	0.8163	0.8523	0.9680	1.0430	1.2408
DetRPCA	0.0277	0.0314	0.0378	0.0661	0.1415	0.5748
FDRPCA	1.6284	1.6287	1.6909	1.9078	2.0620	2.3601
DetDRPCA	0.0631	0.0632	0.0634	0.1060	0.1996	0.8437

Theoretically, the evaluation of computational effort is not trivial for the proposed robust statistical algorithms. The reader is referred to Zuliani (2011), Rousseeuw and Driessen (1999) and Hubert et al. (2012) for more information about computational effort required for RANSAC, FMCD and DetMCD algorithms respectively. We implement all the algorithms using existing

MATLAB$^{®}$ functions assuming that they have been efficiently implemented by the respective developers. If we implement our algorithms in C or C++ then the computation time will be significantly reduced.

3.7 Conclusions

In this chapter, six variants of Fast-MCD and Deterministic MCD based diagnostic PCA, robust PCA and diagnostic robust PCA algorithms have been introduced for fitting planar surfaces in laser scanning 3D point cloud data. Experiments based on simulated and real mobile laser scanning datasets show that the DetMCD techniques outperform classical methods (LS and PCA) and are more robust than RANSAC, MSAC and Fast-MCD based methods (FRD-PCA, FRPCA and FDRPCA). Results from a statistical significance test ('Wilcoxon Signed Rank' test) show that the newly proposed algorithms are significantly more accurate than LS, PCA, RANSAC and MSAC. The proposed methods give better results in terms of (i) different percentage of outlier contamination, (ii) size of the data, (iii) point density variation, and (iv) classification of data into inliers and outliers. The proposed methods classify outliers and inliers accordingly and can reduce masking and swamping effects. RANSAC and MSAC misclassify inliers and outliers and are highly affected by the swamping phenomenon. Hence the resultant planes from RANSAC and MSAC are ill-fitted. The proposed DetMCD based algorithms are significantly faster than the RANSAC, MSAC, and Fast-MCD based robust counterparts FRD-PCA, FRPCA and FDRPCA. The normals and curvatures estimated from the proposed methods are more accurate and robust than the others. Results, using the normals and curvatures from the proposed algorithms in the experiments based on MLS data (for planar and non-planar incomplete complex object surfaces) for plane fitting, sharp feature preservation/recovery and segmentation tasks are more accurate and robust. Using the robust and accurate normals and curvatures it is possible to reduce over and/or under segmentation in a region growing segmentation process. Overall results show that the proposed DetRD-PCA and DetRPCA are very competitive to each other. We observe that DetRPCA gives inconsistent results for small sample sizes combined with a high percentage of outlier contamination. It is also demonstrated that DetRD-PCA performs better than DetRPCA in the presence of low point density and high percentage of outliers.

Similar to many other robust techniques, the proposed algorithms are not efficient for more than 50% of outliers and/or noise. In the next chapter, we will investigate faster methods and methods that can deal with the data even in the presence of 50% or more outliers.

Chapter 4

“Somewhere, something incredible is waiting to be known.”
Carl Sagan

”In nature, we never see anything isolated, but everything in connection with something else which is before it, beside it, under it and over it.”
Goethe

Outlier Detection, Point Cloud Denoising, and Robust Normal and Curvature Estimation

4.1 Introduction

Point cloud processing algorithms frequently use information about local saliencies such as normal and curvature at each point of the data. Research on reliable normal and curvature estimation have been carried out in the disciplines of computer graphics, computer vision, pattern recognition, photogrammetry, reverse engineering, remote sensing and robotics (Klasing et al., 2009; Li et al., 2010; Weber et al., 2012; Masuda et al., 2013; Wang et al., 2013). Accurate and robust feature extraction, surface reconstruction, object modelling and rendering applications heavily depend on how well the estimated local normals and curvatures approximate the true normals and curvatures of the scanned surface (Amenta and Kil, 2004; Dey et al., 2005). One of the main problems for robust normal and curvature estimation is the presence of outliers and/or noise in the data, which is an inevitable experience in point clouds mainly because of the physical limitations of sensors, discontinuities at boundaries between 3D features, occlusions, multiple reflectance and noise that produce off-surface points that appear to be outliers or gross errors (Sotoodeh, 2006). Outlier detection in point cloud data becomes complex because the points are usually unorganized, noisy, sparse, have inconsistent point density, have geometrical discontinuities, arbitrary surface shape with sharp features, incomplete regions of scanned objects, and no knowledge is available about the statistical distribution of the points. Moreover, it is common to get multiple model structures in the data that can create clustered outliers to one structure of interest but inliers to another structure of interest, e.g. a pole in front of a flat wall, may appear as pseudo-outliers. Fleishman et al. (2005) stated that when the underlying surface contains sharp features, the requirement of being resilient to noise is especially challenging since noise and sharp features are ambiguous, and most techniques tend to smooth important features or even amplify noisy samples. Despite recent progress in robust statistics, statistical learning theory, computer vision, data mining, machine learning and pattern recognition techniques for processing scattered point data, the problem of automatic outlier identification and their removal from scattered point data is a challenging task and is still the subject of much research (Barnett and Lewis, 1995; Hodges and Austin, 2004; Sotoodeh, 2006; Rousseeuw and Hubert, 2011; Hido et al., 2011; Aggarwal, 2013; Schubert et al., 2014).

Pioneering work in point cloud processing for surface reconstruction was conducted by Hoppe et al. (1992) who assumed that the underlying surface is smooth locally everywhere in the data. This assumption gives the advantage of approximating the local neighbourhood of an interest point by a planar surface. Since the work of Hoppe et al. (1992), PCA (and its derivations) based local saliency features have been used for point cloud processing including plane fitting, feature extraction, surface segmentation and reconstruction (Pauly et al., 2002; Rabbani, 2006; Belton, 2008; Lari and Habib, 2014; Lin et al., 2014). Although PCA is very popular, it has been shown that it is influenced by outliers and fails to reliably fit planar surfaces. Therefore, saliency features based on PCA are not robust and the resultant analyses can be erroneous and misleading (Mitra et al., 2004). PCA fails to preserve sharp features in the vicinity of geometric singularities such as corners or edges where the normals are discontinuous, because neighbouring points are used indiscriminately to compute the planar fit. The effect is smoothed normal estimates along the edges (Castillo et al., 2013). RANSAC is another approach, frequently used for robust model parameter estimation and to reduce the influence of noise and/or outliers on the estimates. Our results (Nurunnabi et al., 2012a,c) showed that RANSAC is not completely free from the problems of outliers. In spite of the limitations, PCA and RANSAC have been used as the foundations of many other methods, and are widely considered as the state-of-the-art.

This chapter introduces two robust versions of outlier detection algorithms that can identify a large percentage of clustered outliers as well as uniform outliers. We will see that the new algorithms significantly reduce the computation time comparing with the methods developed in Chapter 3. The algorithms coupled with PCA estimate robust local saliency features: normal and curvature. The new methods use local neighbourhood information of the data, assuming that in a certain sufficiently small local neighbourhood, the points are on a planar surface. The new algorithms consist of the following two sequential stages:

- Outliers and/or noise are detected in a local neighbourhood for every point in the data. Robust statistical approaches are used for outlier identification based on measures of the distance of a point to the plane of its local neighbours and the local surface point variation along the normal.
- The best-fit-plane and the relevant local parameters (normal and curvature) are estimated using PCA after eliminating the outlying cases found by the first stage.

The results of the proposed methods are compared with the results based on the existing methods from different disciplines including statistics (PCA, diagnostic PCA and robust PCA in Chapter 3), computer vision (RANSAC and MSAC), data mining (LOF: Breuning et al. (2000); q_{S_p}: Sugiyama and Borgwardt (2013)) and machine learning (uLSIF: Hido et al. (2011)). The accuracy, robustness and speed of computation of the methods are compared w.r.t. the size of the data, outlier percentage, and point density variation. The new methods are evaluated for different applications in point cloud processing including point cloud denoising, sharp feature preservation and segmentation.

The remainder sections of this chapter are arranged as follows. Section 4.2 reviews related work in the state-of-the-art. In Section 4.3, we describe the new outlier detection and saliency features estimation algorithms. In Section 4.4, the computational effort of the proposed algorithms is discussed. The algorithms are demonstrated and evaluated through comparison with the established techniques mentioned above using simulated and real laser scanning point cloud data in Section 4.5, followed by the concluding remarks in Section 4.6.

4.2 Literature Review

This chapter develops methods for outlier detection and estimating robust saliency features: normal and curvature. The related literature is reviewed in two sections: (i) outlier detection, and (ii) robust normal and curvature estimation.

4.2.1 Outlier Detection

Statistics has been the main discipline for outlier detection methods (Cook and Weisberg, 1982; Chatterjee and Hadi, 1988; Barnett and Lewis, 1995; Rousseeuw and Leroy, 2003). A number of outlier detection approaches have been developed in computer vision, data mining, pattern recognition and machine learning, and are referred to by different names e.g. anomaly detection, fault detection, fraud detection, novelty detection, exception mining or one-class classification. Different names depend on application areas, which include information systems, network systems, news documentation, structural health monitoring, industrial machines, and video surveillance (Worden, 1997; Knorr and Ng, 1998; Breuning et al., 2000; Schölkopf et al., 2001; Hodges and Austin, 2004; Kanamori et al., 2009; Aggarwal, 2013; Liu et al., 2013; Dervilis et al., 2014; Schubert et al., 2014). Two solutions have been developed for handling outliers: (i) outlier detection, and (ii) robust estimation.

Existing methods for outlier detection can be roughly categorised into four groups as follows. First, in statistics, these are broadly classified into distribution and depth based methods, where outliers are identified based on standard probability distributions that fit the data best, and in a k-dimensional space assigning a depth, respectively (Barnett and Lewis, 1995; Rousseeuw and Leroy, 2003). One of the main problems of distribution based approaches is information about the underlying data distribution may not always be available. The second type of outlier detection method is distance and/or density based methods. Knorr and Ng (1998) generalized the distribution based approach and formulated the notion of Distance Based (DB) outlier detection for large data. In contrast to DB methods that take a global view of the data, Breuning et al. (2000) introduced a density based approach assuming that object points may be outliers relative to their local neighbourhood. Distance and density based approaches triggered interest in the development of many variants of the algorithms, which are more spatially oriented (Kriegel et al., 2009; Sugiyama and Borgwardt, 2013; Schubert et al., 2014). Third, is the model based approach that is used to learn a model (classifier) from a set of known data, i.e. training data, and then classifies test observations as either inliers or outliers using the learnt model (Schölkopf et al., 2001; Tax and Duin, 2004; Liu et al., 2013). In this category, Tax and Duin (2004) introduced Support Vector Data Description (SVDD). Usually, model based approaches can detect outliers in high-dimensional data but require much more time to construct a classifier (Liu et al., 2013). Hido et al. (2011) pointed out that the solutions of the One-class Support Vector Machine (OSVM) and SVDD depend heavily on the choice of the tunning parameters and there seems to be no reasonable method to appropriately fix the values of the tuning parameters. The last approach, clustering based methods, apply unsupervised clustering techniques mainly to group the data based on their local data behaviour (Jiang and An, 2008). Small clusters that contain significantly less data points are identified as outliers. The efficiency of the clustering based methods highly depends on the clustering techniques that are involved in capturing the cluster structure of the regular (inlier) data (Liu et al., 2013). Several survey papers (Hodges and Austin, 2004; Hadi et al., 2009; Rousseeuw and Hubert, 2011; Nurunnabi et al., 2014b; Schubert et al., 2014) have been published in recent years that found a variety of algorithms covering the full range of statistics, computer vision, data mining, pattern recognition, and machine learning techniques. Hodges and Austin (2004) pointed out that there is no single universally applicable or generic outlier detection approach. People are trying to get more efficient and reliable methods based on their interest.

Robust approaches have been developed to avoid (or reduce) the outlier and/or noise influence on the estimates. Many versions of robust PCA have been introduced in the statistical literature (Hubert et al., 2005; Feng et al., 2012). Nurunnabi et al. (2012a,b,d, 2013a) used Fast-MCD (Rousseeuw and Driessen, 1999) and DetMCD (Hubert et al., 2012) based diagnostic and robust PCA (Hubert et al., 2005) for planar surface fitting. The RANSAC paradigm devised in computer vision is a model based algorithm known as a robust estimator. It has frequently appeared in computer graphics, machine learning, image processing, photogrammetry and remote sensing for planar surface detecting, fitting, extraction and normal estimation (Schnabel et al., 2007; Masuda et al., 2013). Schnabel et al. (2007) developed two optimizations to RANSAC. Deschaud and Goulette (2010) claimed that the algorithms of Schnabel et al. (2007) are slow for large datasets and also showed that although RANSAC is very efficient at detecting large planes in noisy point clouds, it is inefficient at detecting small planes in large point clouds. Duda and Hart (1972) proposed the Hough Transform (HT) which is used to detect and fit geometric shapes (Borrmann et al., 2011). However, Tarsha-Kurdi et al. (2007) showed that the HT is very sensitive to the segmentation parameters values, and RANSAC is more efficient in terms of processing time.

4.2.2 Robust Normal and Curvature Estimation

Many methods have been introduced to improve the quality and speed of normal and curvature estimation in point cloud data. Usually, methods are developed and tailored according to their suitability for the particular application e.g. plane fitting (Wang et al., 2001; Deschaud and Goulette, 2010), surface reconstruction (Hoppe et al., 1992; Sheung and Wang, 2009), sharp feature preserving (Fleishman et al., 2005; Weber et al., 2012; Wang et al., 2013) and normal estimation (Amenta and Bern, 1999; Mitra et al., 2004; Boulch and Marlet, 2012).

Although the methods are developed in different domains to serve different purposes, algorithms for normal estimation can be classified into two major approaches: (i) combinatorial approach (Dey et al., 2005) and (ii) numerical approach (Hoppe et al., 1992; Castillo et al., 2013). The combinatorial approach is based on the information extracted from Delaunay and Voronoi properties

(Amenta and Bern, 1999; Dey et al., 2005). Dey et al. (2005) developed combinatorial methods for estimating normals in the presence of noise. The authors showed that in general, this approach becomes infeasible for large datasets. Numerical approaches find a subset of points in the local neighbourhood that may represent the local surface of an interest point and is known to perform better in the presence of outliers and noise. Then the best-fit-plane to the selected subset is computed and the normal of the plane is treated as the estimated normal for the point of interest. Hoppe et al. (1992) estimated the normal at each point to the fitted plane of the nearest neighbours by applying regression (simply the 'total least squares'), which is regarded as a numerical approach that can be computed efficiently by PCA. PCA based plane fitting which is also known as *PlanePCA* (Klasing et al., 2009), is a geometric optimization and can be shown to be equivalent to the Maximum Likelihood Estimation (MLE) method (Wang et al., 2001). Klasing et al. (2009) compared a number of optimization and averaging methods and concluded their paper by stating that in the case in which a k Nearest Neighbour (kNN) graph is maintained and updated, the *PlanePCA* is the universal method of choice because of its superior performance in terms of both quality of results and speed. The PCA based method minimizes the LS cost function. Hence, the results from PCA are affected by outliers because the covariance matrix used here has an unbounded 'influence function' and a zero 'breakdown point' (Rousseeuw and Leroy, 2003; Hubert et al., 2005). Changing neighbourhood size (Mitra et al., 2004), distance weighting (Alexa et al., 2001), and higher-order fitting (Rabbani, 2006) algorithms have been developed to adjust PCA for better accuracy near sharp features such as corners and edges. It is claimed that such improvements in PCA fail to address the fundamental problem of determining which points in a given neighbourhood should contribute to the normal estimation (Castillo et al., 2013). Öztireli et al. (2009) used local kernel regression to reconstruct sharp features. Weber et al. (2012) claimed the reconstruction from Öztireli et al. (2009) does not have a tangent plane at a discontinuous sharp feature, but only gives the visual effect of a sharp feature during rendering. Fleishman et al. (2005) proposed a forward search approach based robust moving least squares technique for reconstructing a piecewise smooth surface and reliable normal estimation. The method can deal with multiple outliers but requires very dense sampling and a robust initial estimator to start the forward search algorithm. Sheung and Wang (2009) showed that the forward search misclassifies the region when it fails to obtain a good initial fit. Hence the resultant estimates may be unreliable.

4.3 Proposed Methods for Outlier Detection and Robust Saliency Feature Estimation

This section introduces two algorithms for outlier detection and robust saliency features: normal and curvature estimation in laser scanning point cloud data. The algorithms have four basic sequential tasks as shown in Figure 4.1. The process flow in Figure 4.1 is applied to all the points in the point cloud. The consecutive tasks in this process will be described in the next sections.

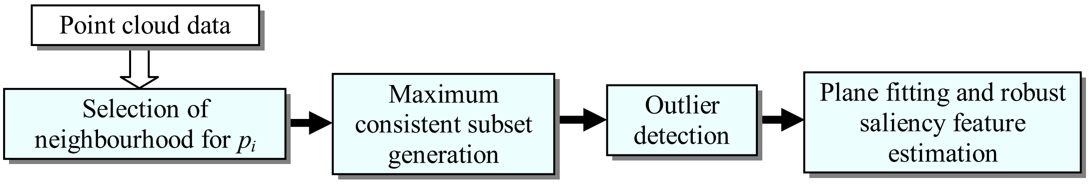

Point cloud data
Selection of neighbourhood for p_i
Maximum consistent subset generation
Outlier detection
Plane fitting and robust saliency feature estimation

Figure 4.1 The process of outlier detection and robust saliency feature estimation.

4.3.1 Neighbourhood Selection

Finding outliers globally in a point cloud is not appropriate because of the presence of multiple object structures, clustered and/or pseudo-outliers. Hence the objective of the new algorithms is to find outliers locally. We find outliers for each and every point within their local neighbourhood to get the benefit that an outlier-free local neighbourhood will create a covariance matrix which will produce more accurate and robust local saliency features. In the case of local neighbourhood based point cloud processing, it can be assumed that within a local neighbourhood of appropriate size, data points should be sampled from a local planar surface (Hoppe et al., 1992). Therefore, for local planar surface fitting and to get the parameters, we need to find an appropriate local region (surface) for an interest point p_i by searching its local neighbourhood.

The two well-known neighbourhood selection methods that have been widely used in point cloud data analysis are the Fixed Distance Neighbourhood (FDN) and the k Nearest Neighbourhood (kNN) (Samet, 2006). The FDN method selects all points within a fixed radius r around p_i, whereas kNN finds the k points having the least distance from p_i. We prefer kNN, because it can manage the problem of point density variation. We know point density variation is a common occurrence when dealing with MLS data because of the variation in movement of the data acquisition sensors (or vehicles) and the variation in orientation of a surface w.r.t. the scanner. In addition, the local statistics (e.g. normals) will be computed for the same number of points.

4.3.2 Finding the Maximum Consistent Set

In our algorithms, we follow the basic strategy of diagnostic statistics, outlier detection and then fitting after removal of the outliers using the classical methods (Rousseeuw and Leroy, 2003). After finding outliers in a local neighbourhood, we remove the outliers and fit the plane to the remaining points with PCA, and estimate the local saliencies: normals and curvatures using the best-fit-plane parameters and the estimated eigenvalues. The algorithms serve two purposes: (i) outlier detection, and (ii) robust saliency feature estimation. Outlier detection by using the new methods takes less computation time and are more accurate than the diagnostic methods in Chapter 3. Outlier detection can also be used in point cloud denoising, see Section 4.5.3.1. It is known that off-surface points can appear as noise which may be treated as outliers or gross errors (Sotoodeh, 2006). Robust saliency features can also be used for local neighbourhood based point cloud processing e.g. sharp feature preserving (Sections 4.5.3.2) and segmentation (Sections 4.5.3.3).

Our algorithms (an earlier version published in Nurunnabi et al., 2013c) can be classified based on the outlier detection procedures involved. The two proposed robust outlier detection methods use the robust z-score (Rz-score) and robust Mahalanobis Distance (RMD). The algorithms couple the idea of using point to plane Orthogonal Distance (OD) and the λ_0 (variation along the surface normal) for identifying outliers. Only the h-subset (a set of points that contains h points) of the majority of good points in a local neighbourhood that are most reliable, homogenous and have minimum sorted ODs are used to fit the plane and to calculate respective λ_0 values. The decision based on majority of consistent points is a fundamental idea of robust statistics (Rousseeuw and Leroy, 2003). Moreover, fixing h can remove the problem of choosing an explicit value of the error threshold, which is a major problem in the RANSAC paradigm (Subbarao and Meer, 2006). In general, we set $h = \lceil 0.5k \rceil$ to get the majority of consistent points. In order to get the best h-subset, the algorithm starts with a random h_0-subset, where the h_0-subset has minimal points (in case of plane fitting, $h_0 = 3$). If the rank of this subset is less than h_0, randomly add more points gradually to the subset until the rank is equal to h_0. The technique of finding the h-subset by using the h_0-subset also reduces the iteration time, because h_0 is considerably less than h. Based on the outlier-free minimal subset (MS) the h-subset can produce a better set of plane parameters. Consequently it gives a better and more accurate normal and the relevant error scale (point to plane orthogonal distance) for the most consistent h-subset, which is used to get the best-fit-plane and robust saliency features. To get an outlier-free h_0-minimal subset, one could iterate (randomly sampling) ${}^{k}C_{h_0}$ times (C means combination), but the number of iterations increases rapidly with the increase in k (the number of points in a local neighbourhood). We employ a Monte Carlo type probabilistic approach (Rousseeuw and Leroy, 2003) to calculate the number of iterations I_t. If we set P_r for the desired probability that at least one outlier-free h_0-subset can be found from the ϵ percentage (outlier rate: probability that a point is an outlier) of outlier contaminated data, then $P_r = 1 - (1 - (1 - \epsilon)^{h_0})^{I_t}$, and I_t can be expressed as:

$$I_t = \frac{log(1 - P_r)}{log(1 - (1 - \epsilon)^{h_0})}. \tag{4.1}$$

Therefore, $I_t = f(p_r, \epsilon, h_0)$, where $h_0 = 3$. We use $P_r = 0.9999$. Users have the freedom to choose P_r based on their knowledge about the data. Fixing a larger probability increases the number of iterations giving a more accurate, more consistent subset with a high probability of the subset being outlier-free. It is known that the number of iterations is a trade-off between accuracy and efficiency. The outlier rate ϵ is generally unknown a *priori*. A smaller ϵ than the real outlier percentage in the data can be influenced by the masking effect. However, an excessively large value of ϵ can create swamping. Experience of MLS data reveals that generally, the majority (more than 50%) of points are inliers within a local neighbourhood. To keep the computation safe, we assume $\epsilon = 0.5$ for real data. The user can change ϵ based on knowledge about the presence of outliers in their data. We find a h-subset for every iteration, based on the minimum Orthogonal Distance (OD) respective to the corresponding fitted plane of the h_0-subset, and calculate λ_0 values for all the h-subsets from the I_t iterations. It is reasonable to assume that the plane that is related to the least λ_0 value also has maximum surface consistency (i.e. minimum variation along the normal) among all the h-subsets. Since, the maximum consistency is attained at $\lambda_0 \approx 0$, we get maximum surface point consistency from the points that have minimum ODs to the fitted plane. We name the method of getting most consistent h-subset as Maximum Consistency with Minimum Distance (MCMD). The algorithms for outlier detection and robust saliency feature estimation are described in the next two Sections 4.3.3 and 4.3.4, respectively.

Our algorithms are motivated by the concept of robust outlier detection in statistics: detecting the outliers by searching for the model fitted by the majority of the data that have some similarities (Rousseeuw and Leroy, 2003; Rousseeuw and Hubert, 2011). The subset of majority points used in our algorithms includes the most homogenous and consistent points to each other. The Maximum Consistent Set (MCS) of homogeneous points in a local neighbourhood can be derived by the steps outlined in Algorithm 4.1.

Algorithm 4.1: MCS (Maximum Consistent Set)

1. Input: neighbourhood Np_i of a point p_i of size k.
2. To get the h-subset for the MCS in a local neighbourhood of a point of interest p_i, we randomly choose h_0 points (in our case $h_0 = 3$, the minimum required number of points for fitting a plane).
3. For the above h_0-subset, we fit a plane by PCA and calculate ODs for all the points in the local neighbourhood to the fitted plane and sort them according to their ODs (Figure 4.2a) as:
$$|\mathrm{OD}(p_1)| \leq, \dots, \leq |\mathrm{OD}(p_h)|, \dots, \leq |\mathrm{OD}(p_k)|,$$
where $\mathrm{OD}(p_i) = (p_i - \bar{p})^T \cdot \hat{n}$ is the OD for the point p_i to the fitted plane, and $\bar{p}$ and $\hat{n}$ are the mean and the unit normal of the fitted plane, respectively.
4. Fit the plane to the above sorted h-subset in Step 3, and calculate the λ_0 value for that plane and store it to the list of previous λ_0 values, defined as $S(\lambda_0)$.
5. Iterate Steps 2 to 4 for I_t times given by Eq. (4.1). We get the λ_0 values for I_t times in $S(\lambda_0)$.
6. Find the h-subset of points for which λ_0 is minimum in $S(\lambda_0)$. This is the required MCS (magenta ellipse in Figure 4.2c) in the local neighbourhood.
7. Output: The MCS for p_i.

4.3.3 Finding Outliers in a Local Neighbourhood

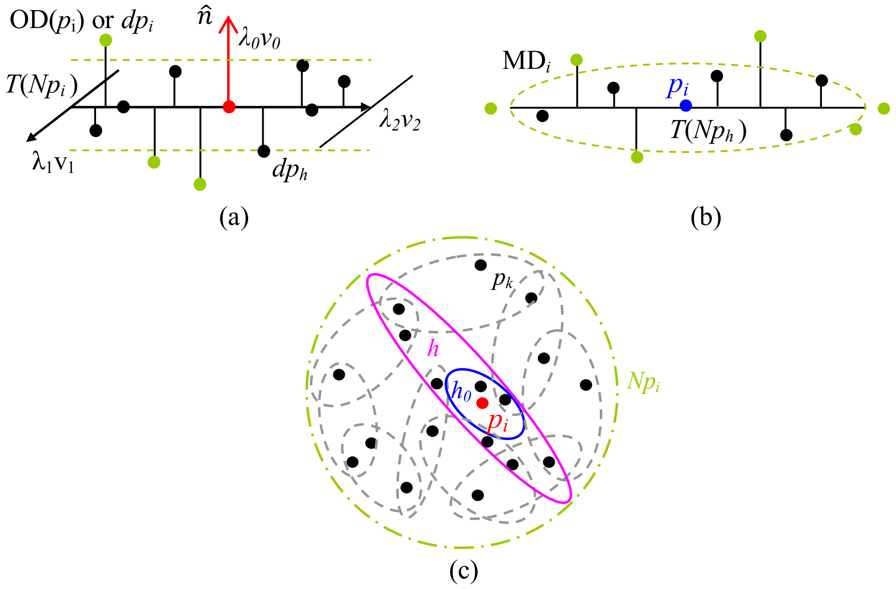
The proposed algorithms identify outliers in a local neighbourhood Np_i of a point p_i in two ways: (i) using the Rz-score in Algorithm 4.2, and (ii) using the robust Mahalanobis Distance (RMD) in Algorithm 4.3. The methods are called as MCMD_Z and MCMD_MD, and summarized in Algorithms 4.2 and 4.3 respectively.

Algorithm 4.2: MCMD_Z (Robust z-score based Outlier Detection)

1. Input: neighbourhood Np_i of a point p_i of size k.
2. Fit the plane using the h-MCS from Step 7 in Algorithm 4.1, Section 4.3.2, and calculate the mean
$$\bar{p}_h = \frac{1}{h}\sum_{i=1}^{h}(p_{x_i}, p_{y_i}, p_{z_i}), \tag{4.2}$$
and the normal $\hat{n}_h$ of the fitted plane.
3. Calculate the robust ODs (Figure 4.2a) for all the points in the local neighbourhood using $\bar{p}_h$ and $\hat{n}_h$ from Step 2, where
$$\text{OD}(p_i) = (p_i - \bar{p}_h)^T.\hat{n}_h, \quad i = 1, \ldots, k \tag{4.3}$$
4. Calculate the Rz-score for all points using the ODs from Step 3 defined as:
$$Rz_i = \frac{|\text{OD}_i - \underset{j}{median}(\text{OD}_j)|}{\text{MAD(OD)}}, \quad i = 1, \ldots, k \tag{4.4}$$
where MAD $= a.\underset{i}{median}|p_i - \underset{j}{median}(p_j)|$, and $a = 1.4826$ is a correction factor used to make the estimator consistent.
5. Observations with Rz_i-score values greater than 2.5 are identified as outliers for p_i.
6. Output: inliers and outliers in a neighbourhood Np_i for the point p_i.

Algorithm 4.3: MCMD_MD (Robust MD based Outlier Detection)

1. Input: neighbourhood Np_i of a point p_i of size k.
2. Calculate mean $\bar{p}_h$ and covariance matrix Σ_h using the h-MCS from Step 7 in Algorithm 4.1.
3. Calculate robust MDs (Figure 4.2b) for all points in the neighbourhood as:
$$\text{RMD}_i = \sqrt{(p_i - \bar{p}_h)^T \Sigma_h^{-1} (p_i - \bar{p}_h)}, \tag{4.5}$$
where $\bar{p}_h$ and Σ_h^{-1} are the mean and the inverse of the scatter matrix from Step 2.
4. Point p_i will be identified as an outlier if
$$\text{RMD}_i > \sqrt{\chi^2_{3,0.975}} = 3.075. \tag{4.6}$$
5. Output: inliers and outliers in a neighbourhood Np_i for the point p_i.



(a)

(b)

(c)

Figure 4.2 (a) Point to plane orthogonal distance OD(p_i) or dp_i, (b) robust MD ellipse MD$_i$, and (c) olive dotted big circle is a local neighbourhood Np_i, ash dotted circles/ellipses are h_0-subsets, magenta ellipse is the MCS, the blue ellipse is the h_0-subset w.r.t. the MCS which produces the least λ_0 value.

4.3.4 Plane Fitting and Robust Saliency Features Estimation

By removing the outliers, using any one of Algorithms 4.2 and 4.3 from Section 4.3.3, we get an outlier-free neighbourhood for the i^{th} point p_i. We now perform PCA to fit the plane for the cleaned neighbourhood. Estimated eigenvalues and eigenvectors are used to get the required robust saliency features for applications in point cloud processing. For example, the least eigenvector (i.e. third PC) is used as the robust normal $\hat{n}$ (defines the plane parameters) and the surface variation along the directions of the corresponding eigenvectors at the point are known as the robust curvature $\sigma(p)$. The method for robust plane fitting and robust saliency features: normal and curvature estimation is in Algorithm 4.4 as follows.

Algorithm 4.4: Robust Saliency Features Estimation

```
1. Input: point cloud $P$.
2. Determine the local $k$ nearest neighbourhood $Np_i$ of a point $p_i$.
3. Find the MCS (Algorithm 4.1) in the $Np_i$.
4. Calculate the $Rz$-score in Eq. (4.4) or RMD in Eq. (4.5) for all $i \in Np_i$.
5. Classify the points in $Np_i$ into inliers and outliers according to the MCMD_Z
   (Algorithm 4.2) or MCMD_MD (Algorithm 4.3) using $Rz$-score or RMD values
   respectively with their corresponding cut-off values assigned.
6. Perform PCA on the cleaned $Np_i$.
7. Arrange the three PCs associated with their respective eigenvalues.
8. Plane fitting: find the two PCs that have the largest eigenvalues, and fit the plane
   by projecting the inlier points onto the directions of the two PCs.
9. Output: robust saliency features: normals, eigenvalues and curvatures.
```

4.4 Computational Effort

Computational effort for the proposed algorithms for outlier detection can be estimated by grouping the steps in the algorithms into two general steps: (i) finding the Maximum Consistent Set (MCS), and (ii) classifying the points in a neighbourhood into inliers and outliers. Three steps are performed in each iteration to find the MCS:

Step A: The cost of estimating the plane parameters with a MS (Minimal Subset) of h_0 points can be defined as:

$C_{estimate}(h_0)$.

Step B: The cost of calculating OD_i $(i = 1, \ldots, k)$ for all the points in the local neighbourhood and sorting them in increasing order is:

$k \ \text{OD}_{i(calculation)} + \text{OD}_{i(sorting)}$.

Step C: The cost of estimating the plane parameters with MCS of sorted h points is:

$C_{estimate}(h)$.

After completing the iteration process, we need to find the minimum λ_0 from the iterations for which the cost is:

$minimum \ \lambda_{0(finding)}$.

To classify points into inliers and outliers, the algorithm uses either MCMD_Z or MCMD_MD. In MCMD_Z, ODs and Rz-scores are calculated for all k points based on the MCS, which are then used to classify points into inliers and outliers according to the predefined cut-off value of 2.5. In MCMD_MD, RMDs are calculated and points are classified into inliers and outliers using the cut-off value of 3.075 assigned in Eq. (4.6). Hence the costs for MCMD_Z and MCMD_MD are:

$$k(\mathrm{OD}_{i(calculation)} + Rz_{i(calculation)} + p_{i(classification)})$$

and

$$k(\mathrm{MD}_{i(calculation)} + p_{i(classification)}),$$

respectively. Therefore, the overall complexity of the proposed algorithms for the worst case situation can be summarized as:

$$O\left(\begin{array}{l} I_t(C_{estimate(h_o)} + k\mathrm{OD}_{i(calculation)} + \mathrm{OD}_{i(sorting)} + C_{estimate}(h)) \\ \quad + \ minimum \ \lambda_{0(finding)} \\ + \begin{cases} k(\mathrm{OD}_{i(calculation)} + Rz_{i(calculation)} + p_{i(classification)}) & ;\mathrm{MCMD\_Z} \\ k(\mathrm{MD}_{i(calculation)} + p_{i(classification)}) & ;\mathrm{MCMD\_MD} \end{cases} \end{array}\right), \tag{4.7}$$

where $I_t = \frac{log(1-P_r)}{log(1-(1-\epsilon)^{h_0})}$.

4.5 Experiments and Evaluation

The algorithms presented in this chapter are demonstrated and evaluated in terms of accuracy, robustness, breakdown point, classification into outliers and inliers, and speed of computation using simulated and real (vehicle borne MLS) datasets. Estimated local saliency features: λ_0, normal and curvature are evaluated for sharp feature preserving and segmentation of 3D point cloud data. In this chapter, we ignore Least Squares (LS) for comparison since it is really incomparable to the robust methods in terms of accuracy. We just consider PCA as a non-robust technique and compare the proposed methods (MCMD_Z and MCMD_MD) with the robust statistical methods: Fast-MCD based Diagnostic PCA (FRD-PCA), Fast-MCD based Robust PCA (FRPCA), Deterministic MCD based Diagnostic PCA (DetRD-PCA), and Deterministic MCD based Robust PCA (DetRPCA), and computer vision techniques: RANSAC and MSAC that are used in Chapter 3. We also compare them with three recently proposed and/or well-known outlier detection methods from the data mining and machine learning literature. These are: LOF, q_{S_p}, and uLSIF briefly described in Chapter 2. We use these outlier detection methods to find outliers in the local neighbourhood of an interest point and then use PCA on the cleaned (after removing the outlying cases) data to get the required saliency features.

4.5.1 Measures of Performance Evaluation

To evaluate the performance of the two proposed algorithms, we fit the plane for a local neighbourhood Np_i of a point of interest p_i using the different methods, and estimate local normal and eigenvalue characteristics e.g. λ_0 and $\sigma(p)$. To measure performance, we calculate three measures:

(i) The bias (dihedral) angle θ between the planes fitted to the local neighbourhood with and without outliers, which is defined as:

$$\theta = arccos|\hat{n}_1^T \cdot \hat{n}_2|, \tag{4.8}$$

where $\hat{n}_1$ and $\hat{n}_2$ are the two unit normals from the fitted planes with and without outliers, respectively. We take the absolute value in Eq. (4.8) to avoid the 180° ambiguity of the normal vectors.

(ii) The variation along the plane normal i.e. the least eigenvalue λ_0.

(iii) The curvature $\sigma(p_i)$ of a point p_i, which is the measure of the surface variation along the directions of the corresponding eigenvalues (Pauly et al., 2002) defined as:

$$\sigma(p_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \qquad \lambda_2 \geq \lambda_1 \geq \lambda_0. \tag{4.9}$$

We use these three measures for a number of evaluation purposes for simulated and real point cloud data analyses as described in the following sections. These measures were used in Chapter 3 and repeated here for clarity.

4.5.2 Simulated Datasets

The simulated datasets used in the following sections are generated by randomly drawing samples from two sets of 3D (x, y, z) multivariate Gaussian normal distributions, one set for Regular R observations and the other set for

Outlying O observations. We create the regular observations assuming that they are from a planar surface, hence the variations among the points in the z (elevation) direction are significantly lower than the variations in the x and y (plane) directions. The regular observations in 3D have means of (2.0, 2.0, 2.0) and variances of (6.0, 6.0, 0.01). Usually, the outlying points are far from the planar surface, so we create the outlying points with means (7.0, 6.0, 8.0) and variances (2.0, 2.0, 1.5). Figure 4.3a depicts the pattern/distribution of a dataset of 50 points including 10 (20%) outlying points. The outlying points appear as clustered outliers marked as red and the regular points are marked as black. We simulate the datasets for different sample sizes n and Outlier Percentages (OP) as needed. As in Chapter 3, by necessity, we generate datasets of size 20 to 200 because several empirical studies show the local neighbourhood used for the real MLS point cloud data analysis (e.g. segmentation) perform well with those sizes.

4.5.2.1 Accuracy and Robustness

We calculate the bias angle θ in Eq. (4.8) to evaluate the accuracy of the plane parameters. For getting statistically reliable results, we simulate 1000 sets of 50 3D points including 10 (20%) clustered outliers which follow a Gaussian normal distribution with the same parameters (mean and variance) as described in the previous section. An example of a dataset is shown in Figure 4.3a. Planes fitted by the different methods are shown in Figure 4.3b, in which the PCA and LOF planes of all the points are tilted away from the real plane of 40 regular points with a large bias angle, and the planes of all the points from the robust methods (RANSAC, MSAC, FRD-PCA, FRPCA, DetRD-PCA, DetRPCA, MCMD_Z and MCMD_MD) are approximately aligned with the plane without outliers. Although uLSIF and q_{S_p} produce less bias angles than PCA and LOF, still they have significantly larger bias angles than the robust methods. From the results of the 1000 runs, we calculate various descriptive measures including mean, median, standard deviation (StD) and Quartile Range (QR=3$^{\text{rd}}$ quartile – 1$^{\text{st}}$ quartile) of bias angles (in degrees) shown in Table 4.1. Results show that in every case of location and scatter statistics: mean, median, StD, and QR, the proposed methods have lower values than the others. LOF has the largest values for all the measures. Based on the values of mean and

StD for the bias angles in Table 4.1 we can arrange the methods according to their rank of superiority as: MCMD_Z, MCMD_MD, DetRD-PCA, DetRPCA, FRD-PCA, FRPCA, MSAC, RANSAC, uLSIF, q_{S_p}, PCA and LOF. There is one exception which is q_{S_p} that has a larger StD of θ than PCA and LOF.

The influence of uniform or scattered outliers on different methods are investigated for plane fitting. We simulate 1000 sets of 50 points including 10 (20%) outliers which follow a Uniform distribution within -9 to $+9$ for all three axes (x, y, z) and the regular observations are as in the previous experiment. Figure 4.3c portrays a dataset with uniform outliers. Results in Table 4.2 show that uLSIF, q_{S_p} and LOF have been improved significantly having values for θ of mean $2.55°$, $4.61°$ and $8.49°$, respectively. That means uLSIF, q_{S_p} and LOF perform significantly better in the presence of uniform outliers and produce less bias angles than their respective results in the presence of clustered outliers. The fitted planes in Figure 4.3d support the results in Table 4.2. PCA and the other robust methods perform almost similarly to how they perform in the presence of clustered outliers.

We use the boxplot as a visualisation tool, which gives more insight into the robustness of the descriptive measures for the bias angle values from the 1000 runs. Figure 4.4a shows that LOF and PCA have by far the worse results. Results from column 9 in Table 4.1 and the length of the boxes in the boxplots in Figure 4.4 support the fact that MCMD_Z and MCMD_MD have 50% of θ values within the minimum quartile ranges 0.316 and 0.344, respectively. That means the two proposed methods (MCMD_Z and MCMD_MD) produce more robust results than the others. In Figure 4.4b, we exclude the boxes for PCA, LOF and q_{S_p}, and the results clearly show better robustness for diagnostic and robust statistical methods FRD-PCA, FRPCA, DetRD-PCA, DetRPCA, MCMD_Z and MCMD_MD than RANSAC, MSAC and uLSIF. In Table 4.1 and Figure 4.4c, we see that Deterministic MCD based DetRD-PCA and DetRPCA are slightly better than the Fast-MCD based FRD-PCA and FRPCA, respectively. In addition to less bias angles, the proposed methods have less outlying results as indicated by the '+' symbols. Figure 4.4d shows the boxplots for the results from the 1000 runs for the datasets in Figure 4.3c. Almost similar conclusions can be drawn from Table 4.2 and the boxplots in Figure 4.4d in the presence of uniform/scattered outliers in a dataset.

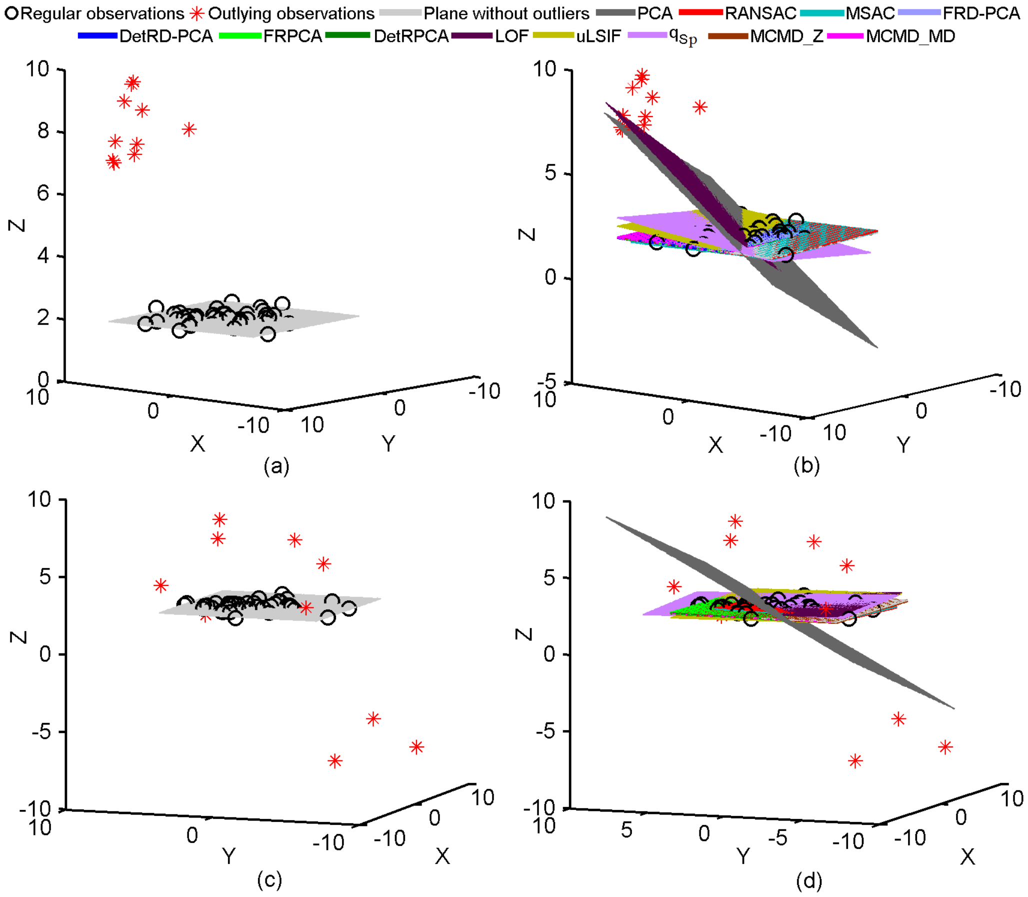


Figure 4.3 (a) Dataset of 50 points with 20% clustered outliers, (b) fitted planes for $n = 50$, OP=20, (c) dataset of 50 points with 20% scattered outliers, and (d) fitted planes for $n = 50$, OP=20.

Table 4.1 Descriptive measures for bias angles (in degrees) from different methods in the presence of clustered outliers.

Methods	Mean	95% Confidence interval of mean		Min.	Max.	Median	StD	QR
		Lower Bound	Upper Bound					
PCA	34.388	34.137	34.639	24.013	87.377	34.324	4.038	4.644
RANSAC	1.168	1.094	1.241	0.000	7.019	0.813	1.183	1.565
MSAC	1.080	1.004	1.156	0.000	6.724	0.629	1.019	1.500
FRD-PCA	0.535	0.510	0.560	0.007	4.074	0.442	0.406	0.411
DetRD-PCA	0.428	0.407	0.448	0.027	2.922	0.353	0.331	0.347
FRPCA	0.661	0.632	0.690	0.012	3.893	0.577	0.464	0.513
DetRPCA	0.479	0.458	0.501	0.014	3.122	0.412	0.345	0.360
LOF	41.457	40.961	41.949	24.224	89.423	40.201	7.976	6.867
uLSIF	6.097	5.954	6.235	0.562	17.938	5.731	2.304	2.769
q_{S_p}	24.144	23.209	25.105	0.017	43.968	30.262	15.302	30.168
MCMD_Z	0.391	0.375	0.407	0.006	2.459	0.346	0.253	0.316
MCMD_MD	0.424	0.404	0.444	0.006	2.186	0.349	0.319	0.344

Table 4.2 Descriptive measures for bias angles (in degrees) from different methods in the presence of scattered/uniform outliers.

Methods	Mean	95% Confidence interval of mean		Min.	Max.	Median	StD	QR
		Lower Bound	Upper Bound					
PCA	27.593	26.588	28.598	0.442	89.819	25.339	16.231	19.237
RANSAC	1.184	1.109	1.258	0.005	9.140	0.824	1.204	1.574
MSAC	1.157	1.079	1.234	0.000	7.379	0.730	1.202	1.569
FRD-PCA	0.547	0.522	0.578	0.006	4.259	0.438	0.439	0.456
DetRD-PCA	0.419	0.402	0.440	0.008	2.479	0.352	0.305	0.438
FRPCA	0.675	0.643	0.706	0.025	3.326	0.550	0.504	0.547
DetRPCA	0.473	0.453	0.495	0.022	2.583	0.393	0.335	0.488
LOF	8.490	7.839	9.238	0.061	62.821	3.507	11.193	9.612
uLSIF	2.550	2.432	2.664	0.056	12.277	2.116	1.874	2.146
q_{S_p}	4.611	4.378	4.845	0.034	18.722	3.603	3.775	4.666
MCMD_Z	0.427	0.409	0.445	0.016	1.905	0.366	0.286	0.353
MCMD_MD	0.522	0.501	0.542	0.012	2.244	0.452	0.335	0.420

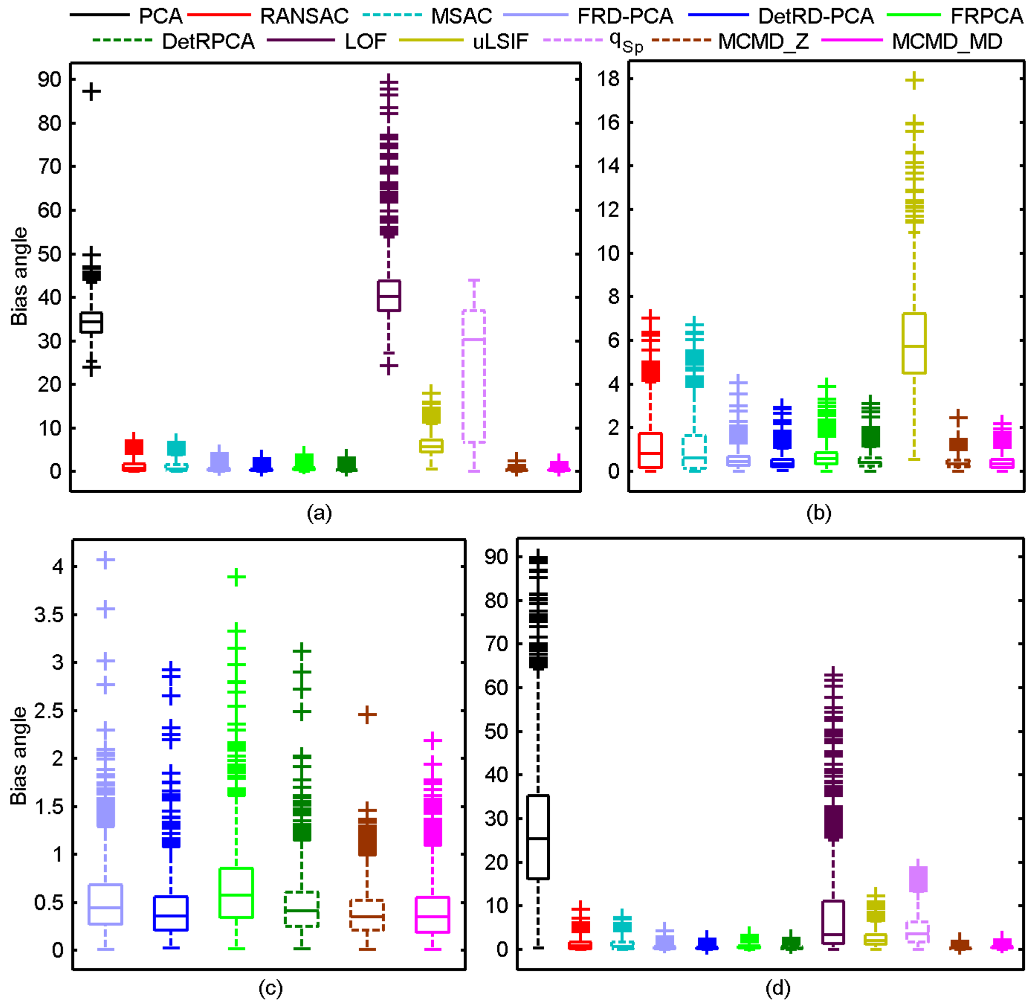


Figure 4.4 Boxplots of bias angles for $n = 50$, OP=20, clustered outliers: (a) all methods (b) all methods excluding PCA, LOF and q_{S_p}, (c) only robust statistical methods, and (d) boxplots for $n = 50$, OP=20, scattered outliers; all methods.

4.5.2.2 Breakdown Point Evaluation

The Breakdown Point (BP), which is considered as a robustness measure is evaluated by using the bias angle θ defined in Eq. (4.8). A bias angle between the best-fit-planes from the data with and without outliers for a robust method should be theoretically zero. We generate 1000 datasets of 100 points using the same parameters as for the previous experiment for clustered outliers with outlier percentages of 1% to 80%. Since in the previous section we see DetRD-PCA and DetRPCA perform better than their counterparts FRD-PCA and FRPCA, respectively, we omit FRD-PCA and FRPCA for the rest of the chapter. We calculate the values of θ (in degrees) from the fitted planes with and without outliers using the different methods for every dataset of outlier percentage. The average θ is calculated from the 1000 samples. Results are portrayed in Figure 4.5a. Figure 4.5a clearly shows that PCA breaks down in the presence of just one outlier. That means PCA has a BP $\approx 0\%$. The values of average θ from PCA and LOF increase with increasing outlier percentage, which indicates that the influence of outliers is unbounded on θ for PCA and LOF. Even for 1% of outliers uLSIF produces an average $\theta = 3.626°$, and continues with an approximately linear pattern with between 1% to 80% outliers present. Figure 4.5a shows that DetRPCA, DetRD-PCA, MCMD_Z, MSAC, RANSAC, and MCMD_MD breakdown approximately at 41%, 47%, 49%, 64%, 64% and 74% of outlier presence, respectively. The results show that the proposed MCMD_MD attains the highest BP. DetRPCA, DetRD-PCA and MCMD_Z produce more accurate results (less bias angles) than RANSAC, MSAC and uLSIF until they break down at 41%, 47% and 49% respectively. To explore the deviations of the robust methods, we consider outlier percentages 1% to 40%, and exclude PCA, LOF and q_{S_p} from Figure 4.5b, and also uLSIF from Figure 4.5c. Figures 4.5(a–c) clearly reveal the better performance of the proposed methods in the presence of outliers.

To visualize the performance for a high percentage of outlier contamination, we generate two datasets of 50 points contaminated with 70% clustered outliers and 80% scattered outliers. The fitted planes in the presence of clustered and scattered outliers are shown in Figures 4.6(a and b), respectively. We see that only MCMD_MD successfully fits the planes in the presence of 70% clustered and 80% uniform outliers. Although uLSIF fitted planes tolerate a high percentage of outliers its bias angles are significantly larger than for MCMD_MD. In Figure 4.6b, in the presence of 80% scattered outliers, although RANSAC fits the plane almost at the right orientation, it is influenced by outliers and the size of the plane is enlarged. That means some outlying points are considered as inliers, which is the well-known masking effect, but MCMD_MD is not affected by such a limitation. The boxplots in Figure 4.6c show the results of bias angles from 1000 runs for the simulated datasets of 50 points with 70% clustered outliers. The boxplots reveal that only MCMD_MD gives robust estimates and the others break down. The length of the box for MCMD_MD is the least and almost along the zero line. Similar findings can also be seen in Figure 4.5a.

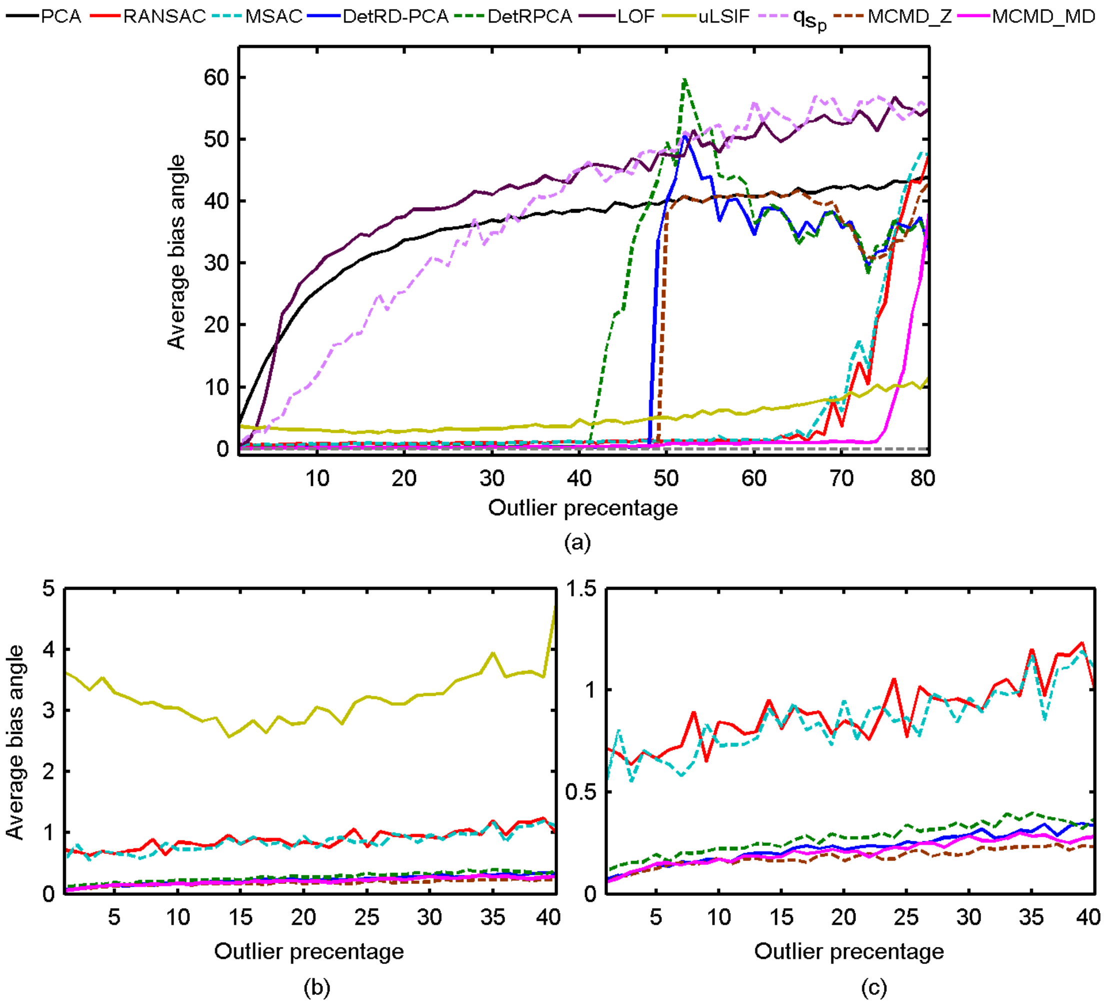


Figure 4.5 Average bias angle θ versus outlier percentage, (a) all methods, (b) all methods excluding PCA, LOF and q_{S_p}, (c) all methods excluding PCA, LOF, q_{S_p} and uLSIF.

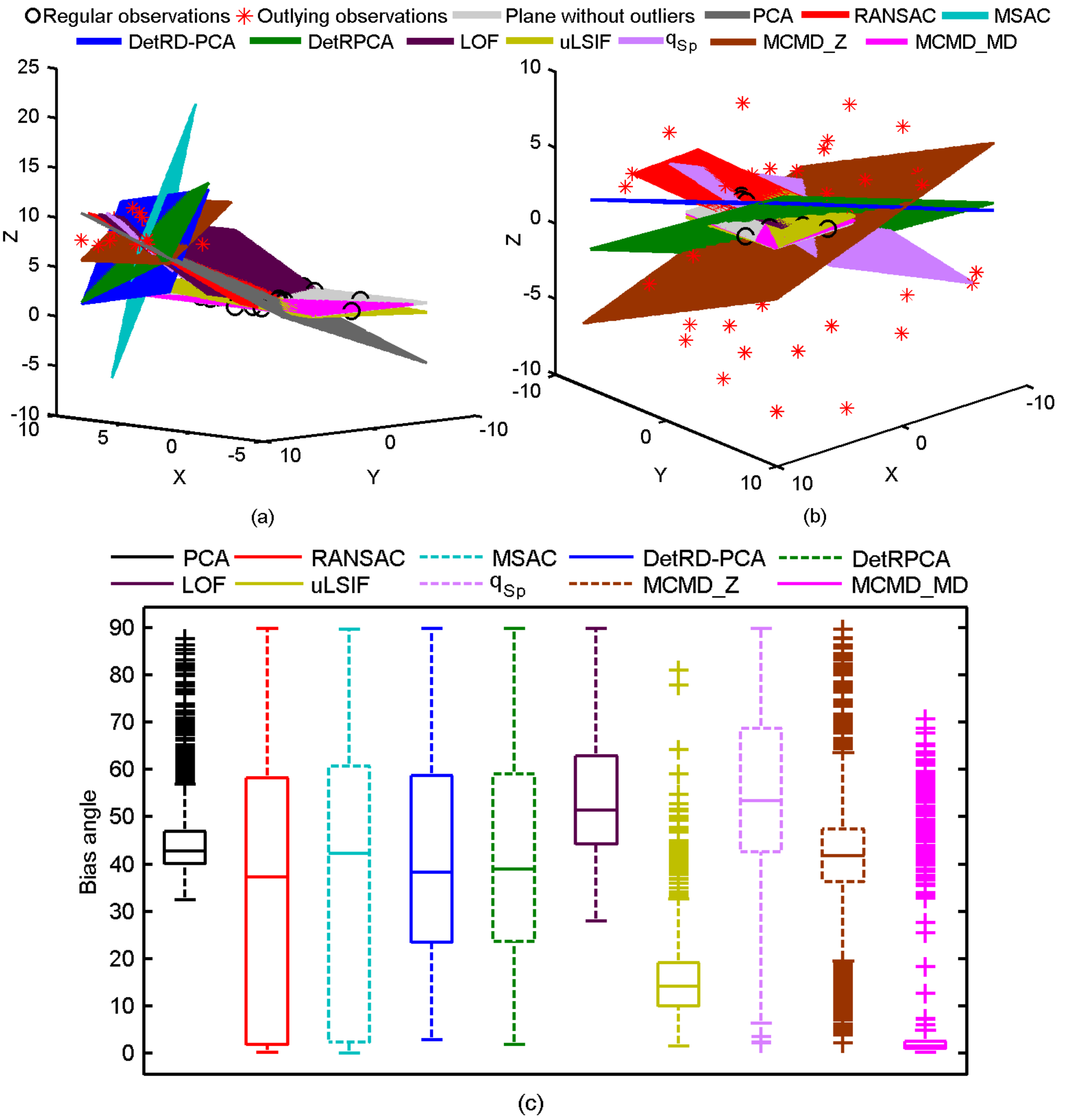


Figure 4.6 Fitted planes: (a) $n = 50$, OP=70, clustered outliers, (b) $n = 50$, OP=80, scattered outliers, and (c) boxplots of bias angles for $n = 50$, OP=70, clustered outliers.

4.5.2.3 Influence of Sample Size and Outlier Percentage on Bias Angles

For investigating the effect of sample size and different percentages of outlier presence in the data, we generate datasets for various sample sizes n of 20, 50 and 200, and outlier percentages 1% to 40%. We carried out 1000 runs for each and every sample size and outlier percentage. In the previous experiment, Table 4.1 and Figure 4.5 show that there are big gaps for bias angles between two groups: (i) PCA, LOF, q_{S_p} and uLSIF, and (ii) the robust methods (RANSAC, MSAC, DetRD-PCA, DetRPCA, MCMD_Z and MCMD_MD). We concentrate now only on the robust methods. Results for average bias angles are shown in Figure 4.7. In Figure 4.7a, for a small sample of size 20, we see RANSAC and MSAC give inconsistent results for almost all percentages of outliers, and DetRPCA breaks down at around 25% outliers. For n of 50 and 200, Figures 4.7(b and c) show that RANSAC and MSAC both have larger bias angles than the other robust methods. MCMD_MD always performs better than DetRD-PCA, DetRPCA, RANSAC and MSAC. MCMD_Z has the least bias angles for almost all the cases of sample sizes and outlier percentages.

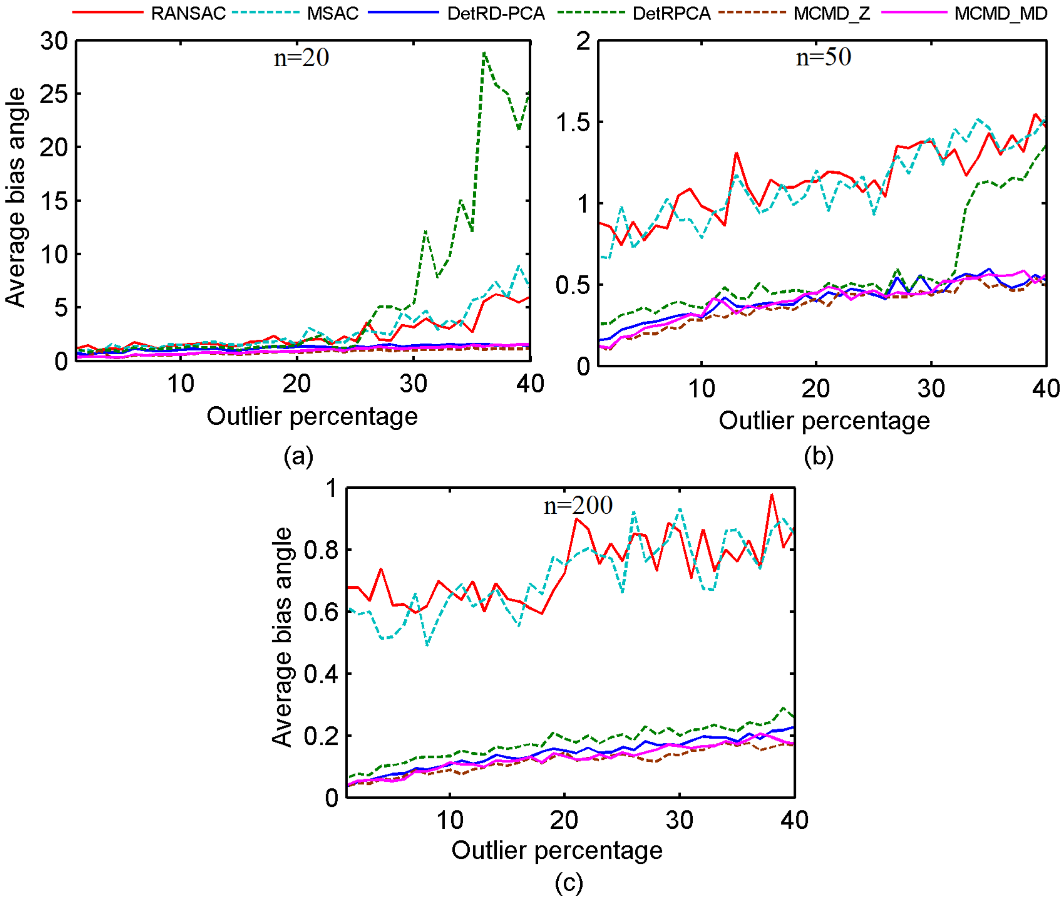


Figure 4.7 Average bias angle versus outlier percentage for: (a) n=20, (b) n=50, and (c) n=200.

4.5.2.4 Effect of Point Density Variation on Bias Angles

It is hypothesised that point density variation affects plane parameter estimation and consequently bias angle θ. To see the effect, we simulate datasets with different variations in the surface directions (i.e. x-y directions). We generate 1000 datasets of 50 points including 10 (20%) outliers for six (I to VI in Table 4.3) different combinations of variances in x and y. The rows of Table 4.3 show the combinations of variances for Regular R and Outlier O data. Other necessary parameters for the datasets are the same as used previously. The results in Figure 4.8a show that robust methods give low θ values (i.e. less influenced by outliers in the presence of point density variation) compared with PCA. In Figure 4.8b where PCA is removed, the proposed MCMD_Z and MCMD_MD results are clearly lower than those for RANSAC, MSAC, DetRD-PCA and DetRPCA.

As for point density variation, surface thickness/roughness influences surface fitting methods. To measure the effect of roughness on the estimates, we change the variance along the elevation or z axis. We simulate 1000 datasets of 50 points with 20% outliers. The z variances for regular observations are 0.001, 0.01, 0.02, 0.05, and 0.1. Figure 4.8c shows PCA is markedly worse than all the other methods. Figure 4.8d shows that RANSAC and MSAC have larger and steadily increasing θ values than the other robust methods. All the methods have increasing departures for the bias angles with the increase of z variance, but MCMD_Z and MCMD_MD have better accuracy than the other methods for all values of the z variance.

Table 4.3 Variances for Regular R and Outlier O data.

Datasets	I	II	III	IV	V	VI
x(R,O) variances	(2,1)	(6,2)	(8,4)	(10,6)	(12,8)	(15,10)
y(R,O) variances	(2,1)	(6,2)	(8,4)	(10,6)	(12,8)	(15,10)

4.5.2.5 Outlier Detection and Performance as a Classifier

We evaluate the performance of RANSAC, MSAC, DetRD-PCA, DetRPCA and the two proposed methods: MCMD_Z and MCMD_MD for outlier detection and as classifiers to categorize the points into inliers and outliers. We generate random datasets for a sample size of 100 with outlier percentages of 5%, 20% and 40% using the same input parameters used previously. We perform 1000 runs for each outlier percentage. We find outliers and classify the points into inliers and outliers for every method, and calculate the Correct Outlier Identification Rate

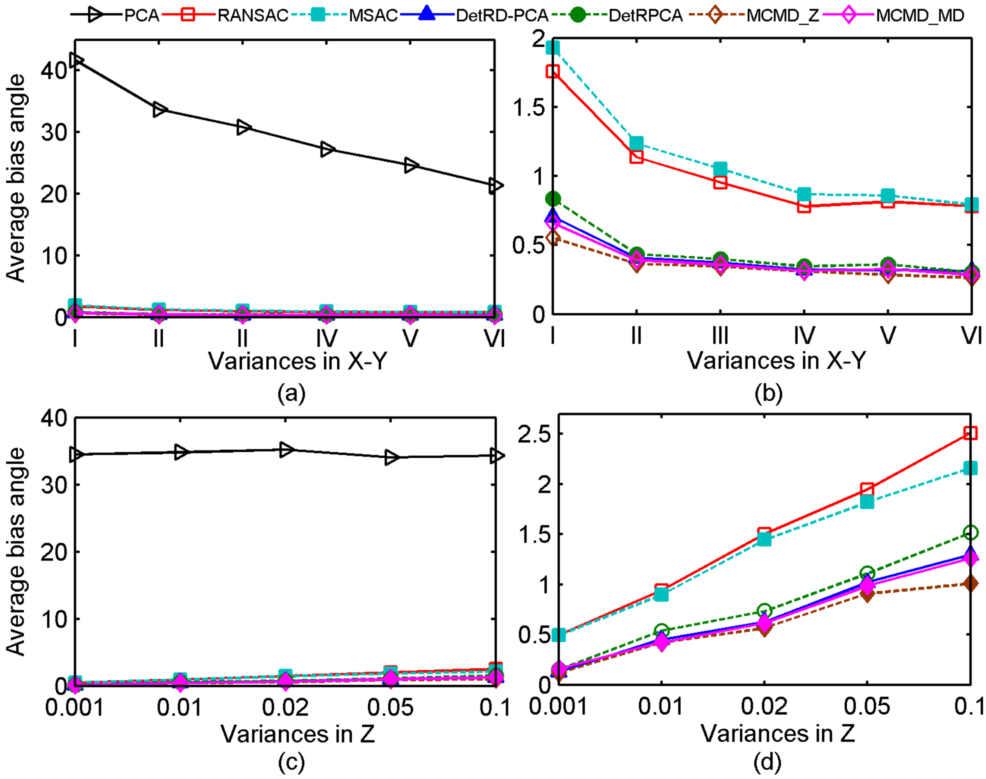


Figure 4.8 Average bias angle θ versus: (a) variances in x-y axes for all the methods, (b) variances in x-y axes for robust methods, (c) variances in z axis for all the methods, and (d) variances in z axis for robust methods.

(COIR), Correct Inlier Identification Rate (CIIR) and number of inliers identified as outliers, which are considered as True Positive Rate (TPR), True Negative Rate (TNR) and False Positive Rate (FPR) or false alarm or Swamping Rate (SR), respectively. We also compute accuracy (Acc) based on the classifications. The measures (in percentages) are defined in Fawcett (2006) and Sokolova et al. (2006) as:

- TPR (COIR)=$\frac{\text{number of outliers correctly identified}}{\text{total number of outliers}} \times 100$,
- TNR (CIIR)=$\frac{\text{number of inliers correctly identified}}{\text{total number of inliers}} \times 100$,
- FPR (SR)=$\frac{\text{number of inliers identified as outliers}}{\text{total number of inliers}} \times 100$,
- Accuracy=
$$\frac{\text{number of correctly identified outliers + number of correctly identified inliers}}{\text{total number of points}} \times 100.$$

These measures were used in Chapter 3 and repeated here for clarity. Table 4.4 shows the average TPR, TNR, FPR and Accuracy from 1000 runs. Results show that RANSAC and MSAC have lower rates of correctly identified inliers (TNR) than the robust statistical methods (DetRD-PCA, DetRPCA, MCMD_Z and MCMD_MD). RANSAC and MSAC are more affected by the swamping (FPR) phenomenon i.e. misclassify inliers as outliers at a very high rate. So, RANSAC and MSAC fit a plane with a very high rate of misclassification error and with lower support of inliers. We see that for 20% outliers, the RANSAC plane is fitted based on only 34 inlier points (TNR of 33.99). Our methods: MCMD_Z and MCMD_MD correctly identify outliers with a very low FPR or SR. For example, for the dataset with 20% of outliers, MCMD_Z and MCMD_MD have accuracies of 99.75% and 98.45%, and SR (FPR) of 0.31% and 1.94% respectively. DetRD-RPCA and DetRPCA perform significantly better than RANSAC and MSAC but they are less efficient than MCMD_Z and MCMD_MD. Table 4.4 reveals that the proposed methods have the higher rate of TPR, TNR and Accuracy with a very low rate of swamping for all the cases of outlier percentages.

Table 4.4 Classification into inliers and outliers.

Sample size	Methods	Outlier percentage											
		5				20				40			
		TPR	TNR	FPR	Acc	TPR	TNR	FPR	Acc	TPR	TNR	FPR	Acc
100	RANSAC	100	32.61	67.39	35.98	100	33.99	66.01	47.19	100	35.48	64.52	61.29
	MSAC	100	32.06	67.94	35.46	100	33.26	66.74	46.61	100	35.13	64.87	61.08
	DetRD-PCA	100	95.32	4.68	95.55	100	96.79	3.21	97.43	100	98.02	1.98	98.81
	DetRPCA	100	95.19	4.81	95.43	100	96.93	3.07	97.54	89	96.18	3.82	93.40
	MCMD_Z	100	97.65	2.35	97.77	100	99.69	0.31	99.75	100	100.00	0.00	100.00
	MCMD_MD	100	97.88	2.12	97.99	100	98.06	1.94	98.45	100	98.10	1.90	98.86

To investigate the variation in performance for classification (e.g. accurate inlier identification), we generate datasets of 100 points with 20% outliers and run the experiment 100 times. We calculate the number of inliers correctly identified, and generate the histogram of the number of runs versus the number of inliers correctly identified for every run. In Figure 4.9, the histograms show that most of the time the proposed methods perform significantly better than the other methods and successfully identify inliers, as the histograms for MCMD_Z and MCMD_MD are centered very close to 80 inliers, whereas histograms for RANSAC and MSAC show that they identify inliers around 20% to 40% out of the possible 80%.

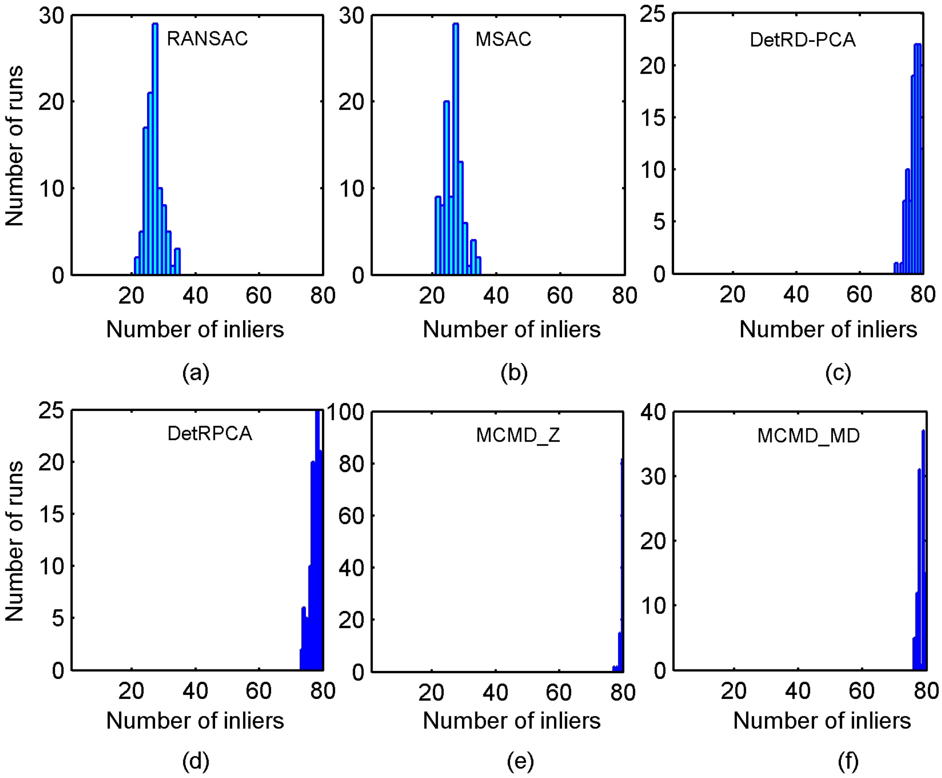


Figure 4.9 Histograms for the number of runs versus the number of correctly identified inliers.

4.5.2.6 Speed of Processing

A positive benefit of the proposed algorithms is speed of computation. We evaluate speed as a function of sample size and outlier percentage. We generate 1000 datasets of (i) different sample sizes 20, 50, 100, 1000, and 10000 with fixed 20% percentage of outliers (results in Table 4.5), and (ii) different percentages (1% to 40%) of outliers with 50 sample points, (results are in Figure 4.10a). For more accurate quantitative comparison, some of the results (OP=5%, 10%, 20%, 30% and 40%) are given in Table 4.6. All the results in this section are counted in seconds using the MATLAB® profile function. Results are the mean computation time from 1000 runs for each and every sample. Results show the new methods take significantly less time than existing robust methods (DetRD-PCA, DetRPCA, RANSAC and MSAC). For example, for a sample size of 50 (Table 4.5, Column 3), MCMD_Z and MCMD_MD take only 0.0085s and 0.0084s respectively, which are approximately 15 times less than RANSAC (0.1262s). This difference improves up to 58.6 and 64.2 times for the proposed methods when the number of sample points increases to 10,000. Although time increases with the increase of sample size, the times for the proposed MCMD_Z and MCMD_MD increase at lower rates than the times for RANSAC and MSAC. Hence the time difference will be more when large datasets are used for local neighbourhood based feature extraction. MSAC and DetRPCA take a little more time than RANSAC and DetRD-PCA, respectively. For fitting planes of 50 points with 5% outliers Table 4.6 shows RANSAC (0.0731s) takes 15.55 and 18.74 times more computation than MCMD_Z (0.0047s) and MCMD_MD (0.0039s) respectively, and around 11 times more computation when the dataset is contaminated with 40% outliers. Figure 4.10a shows the time for both DetRD-PCA and DetRPCA are almost equal for 1% to 40% outliers, i.e. their lines show a linear pattern, because both the methods perform the same number of iterations. Table 4.6 shows the proposed methods are significantly faster than the DetMCD based approaches, e.g. for the data with 10% outliers (Column 3), MCMD_Z (0.0056s) is 4.63 and 5.25 times faster than DetRD-PCA (0.0259s) and DetRPCA (0.0294s), respectively. MCMD_MD performs a little faster than MCMD_Z.

Table 4.5 Mean computation time (in seconds) for different sample sizes.

Methods	Sample size				
	20	50	100	1000	10000
PCA	0.0004	0.0004	0.0004	0.0005	0.0017
RANSAC	0.0672	0.1262	0.1657	0.4268	2.3692
MSAC	0.0777	0.1476	0.1863	0.4637	2.5190
DetRD-PCA	0.0252	0.0282	0.0330	0.1402	0.4602
DetRPCA	0.0280	0.0317	0.0373	0.1456	0.5834
MCMD_Z	0.0084	0.0085	0.0087	0.0108	0.0404
MCMD_MD	0.0080	0.0084	0.0079	0.0101	0.0369

Table 4.6 Mean computation time (in seconds) for different outlier percentages.

Methods	Outlier percentage				
	5	10	20	30	40
PCA	0.0004	0.0003	0.0004	0.0004	0.0005
RANSAC	0.0731	0.0919	0.1224	0.1655	0.2406
MSAC	0.0846	0.1054	0.1360	0.1947	0.2845
DetRD-PCA	0.0265	0.0259	0.0274	0.0273	0.0256
DetRPCA	0.0294	0.0294	0.0307	0.0301	0.0316
MCMD_Z	0.0047	0.0056	0.0085	0.0130	0.0211
MCMD_MD	0.0039	0.0051	0.0077	0.0124	0.0203

A major issue concerning the speed of processing for iterative methods is the number of iterations performed. The iteration procedure used in our algorithms is similar to the classical RANSAC. Zuliani (2011) developed MATLAB® code for the RANSAC and MSAC algorithms following the developments of Tordoff and Murray (2005) to get better results in the presence of noise. In Table 4.7, it is shown that the number of iterations increases with the increase of outlier percentage in the data. We see that the number of iterations that increases for the code of Zuliani (2011) is more than the number of iterations for the proposed algorithms, in Eq. (4.1). In Figure 4.10b, we show results for RANSAC(Z), and MSAC(Z) for the iteration process of Zuliani (2011) and RANSAC(C) and MSAC(C) for the classical iteration process. We know more iterations can produce more chances of getting an outlier-free subset and as a consequence we can get better estimates. To see the effect of more iterations and the generation of comparable results of Zuliani (2011) for parameter estimation, we generate 1000 sets of 50 points with outlier percentages 1% to 40% and calculate average bias angles for MCMD_Z, MCMD_MD, RANSAC and MSAC. We choose a sample size of 50 because we observe taking local neighbourhood sizes of 30 to 200 works well for local saliency features such as normal and curvature estimation used in real MLS data processing, and the presence of outliers in real data is usually not more than 30% to 40%. We calculate the number of iterations (Table 4.7) for outlier percentages 5%, 10%, 20%, 30%, and 40%, and average bias angles that are calculated twice for RANSAC and MSAC algorithms, one for the number of iterations I_t in Eq. (4.1), which is similar to our methods, and the other following Zuliani (2011). Results are portrayed in Figure 4.10b. This shows that estimates (bias angles) for MCMD_Z and MCMD_MD are better (lower) than RANSAC and MSAC for both the cases of iteration procedures. Figure 4.10b also shows that results for RANSAC and MSAC from Zuliani (2011) are more accurate and robust than the classical RANSAC and MSAC methods. However, the proposed methods produce better results with less iterations for both RANSAC and MSAC algorithms.

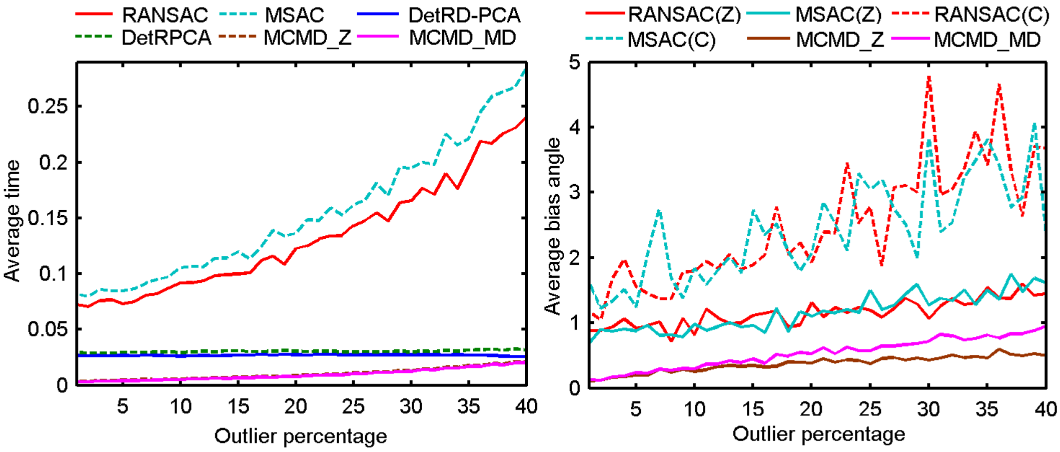


Figure 4.10 Graphs showing (a) average time (in second) versus outlier percentage, and (b) average bias angle versus outlier percentage.

Table 4.7 Number of iterations for different outlier percentages.

Methods	Outlier percentage				
	5	10	20	30	40
RANSAC(Z)	350	442	583	794	1186
RANSAC(C)	5	8	13	22	38
MSAC(Z)	395	495	630	907	1360
MSAC (C)	5	8	13	22	38
MCMD_Z	5	8	13	22	38
MCMD_MD	5	8	13	22	38

4.5.2.7 Effect of Noise Variance on Bias Angles and the Variances of the Estimated Normals

It is known that noise can change the orientation and estimates of the normals. To see the influence of different values of the variances of artificial noise on the parameter estimation for the normals for real data, we take a MLS small dataset, which contains 300 points from a planar surface. To evaluate the robustness of the normals for changes in variance, we add 25% Gaussian noise of mean 0.0 and different variances: 0.01, 0.02, 0.03, 0.05 and 0.1. We fit the planes for every point before and after adding the noise with the same size of neighbourhood $k = 20$. Figure 4.11a shows the data contaminated with 25% Gaussian noise of mean zero and 0.03 variance, and Figures 4.11(c and d) show the quality of the estimated normals from PCA and a robust method e.g. MCMD_Z. We plot the normals without noise (in blue) and after adding the noise (in red). Figure 4.11c shows that deviations between PCA normals are distracted in the places where noise is included in a local neighbourhood. The MCMD based proposed method MCMD_Z produces almost similar normals (Figure 4.11d) with and without noise and looks almost similar in orientation at every point. The normals are as smooth as in Figure 4.11b, where normals are portrayed without noise.

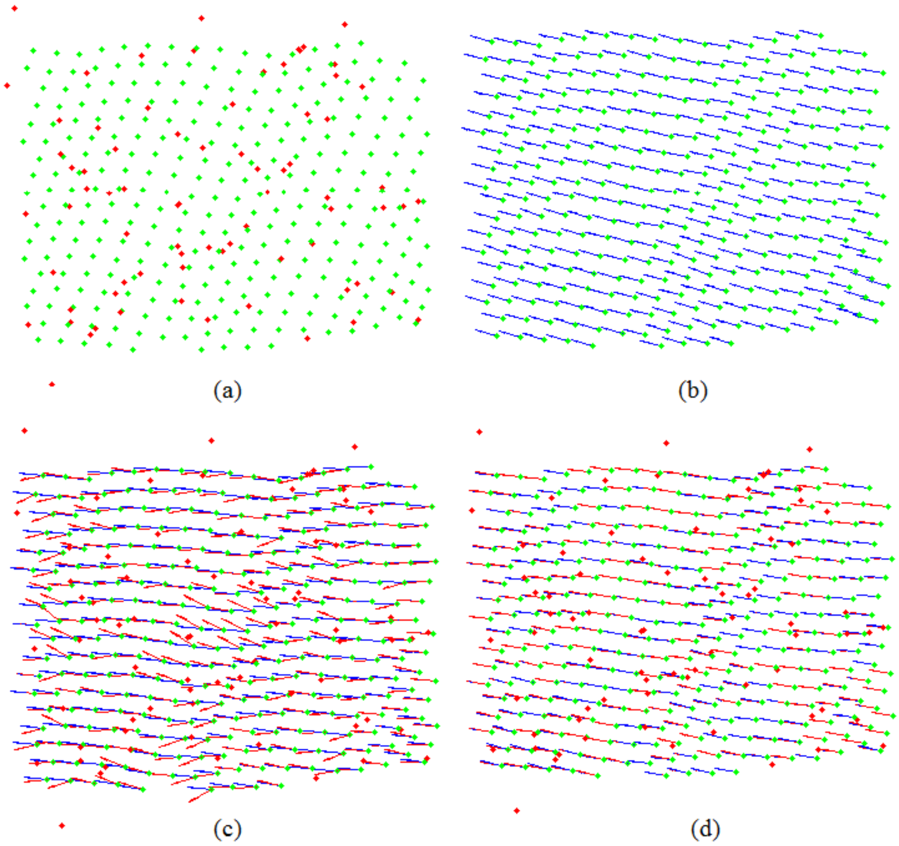


Figure 4.11 (a) Laser data (green points) with 25% noise (red points), (b) PCA normals without noise, (c) PCA normals; two normals for every point before (in blue) and after (in red) adding noise, and (d) MCMD_Z normals; before (in blue) and after (in red) adding noise.

We calculate average bias angles and the variances of the estimated normals as defined by Wang et al. (2001):

$$V_{\hat{n}} = arctan\left(\sqrt{\frac{\lambda_2+\lambda_1}{2}}\right), \tag{4.10}$$

where λ_2 and λ_1 are the two largest eigenvalues of the covariance matrix Σ of the local normals for noisy data. Table 4.8 contains the average bias angles (left columns) and the variances (right columns) of the estimated normals respectively for the different noise variances. Results show that PCA and RANSAC have increasing bias angles with the increase in noise variance, and RANSAC and MSAC produce lower bias angles than PCA. DetRD-PCA, DetRPCA, MCMD_Z and MCMD_MD give significantly lower bias angles than RANSAC and MSAC, and they have almost similar results to each other over the values of noise variance. Overall, the proposed methods produce better and more robust normals (i.e. lower bias angles) than the other methods. Similar type of conclusions can be drawn for the results of variances of the normals in Table 4.8. That means the proposed methods have lower variances, more consistent normals, and lower bias angles with the change of noise variances.

Table 4.8 Average bias angle (in degrees) and variances (in degrees) of the estimated normals.

Methods	Noise variances									
	0.01		0.02		0.03		0.05		0.1	
PCA	1.842	1.378	2.984	2.144	4.186	2.924	5.298	4.336	5.489	8.552
RANSAC	1.842	1.378	2.435	1.928	2.616	2.031	2.681	2.148	2.895	2.414
MSAC	1.840	1.378	2.343	1.862	2.473	1.903	2.422	1.895	1.266	1.478
DetRD-PCA	0.613	0.679	0.551	0.652	0.522	0.584	0.486	0.595	0.480	0.586
DetRPCA	0.625	0.680	0.556	0.659	0.549	0.591	0.493	0.592	0.484	0.558
MCMD_Z	0.518	0.502	0.474	0.515	0.458	0.503	0.418	0.511	0.464	0.508
MCMD_MD	0.577	0.529	0.510	0.574	0.495	0.528	0.423	0.547	0.471	0.528

4.5.3 Mobile Laser Scanning Data

This section evaluates the performance of the estimated λ_0 (variation along the normal), curvature $\sigma(p)$ and the normal $\hat{n}$ in the context of (i) denoising, (ii) sharp feature preserving, and (iii) segmentation, for real Mobile Laser Scanning (MLS) data. The datasets are collected from a local mobile mapping survey company that captured data using a vehicle based laser scanning system along a road corridor at typical traffic speed.

4.5.3.1 Denoising in Point Cloud Data

Dataset 4.1: Signpost dataset

The proposed algorithms are able to remove noise and recover the detail from real point cloud data. To demonstrate it we consider a vehicle borne mobile laser scanning dataset, shown in Figure 4.12a. We call the dataset the 'signpost' dataset. The dataset contains 21,820 points of planar (signs), almost planar (road surfaces) and non-planar (signposts) objects. Although there is some real noise in the data, we deliberately add some more artificial noise i.e. 10% (2182 points) Gaussian noise with mean 0.0m and StD 0.1m to the real data. The noisy data can be seen in Figure 4.12b. To identify noise, for the proposed algorithms, we calculate the Rz-score given in Eq. (4.4) or the RMD values given in Eq. (4.5) for all the points based on their respective local neighbourhoods. The point p_i is defined as noise if Rz_i or RMD$_i$ exceeds their respective cut-off values. We perform all the methods with neighbourhood size $k = 50$, and calculate the values of correct noise (outlier) identification rate, correct inliers (regular points) identification rate, false positive rate and accuracy, which are COIR (TPR), CIIR (TNR), FPR and Accuracy, respectively as defined in Section 4.5.2.5. We also count the number of Correctly Identified Noise (CIN) and Correctly Identified Regular (inlier) points (CIR). FPR is the rate of inliers identified as noise known as swamping, and FNR which is the rate of noise identified as regular points known as masking. Results (in percentages) are in Table 4.9. Results show that MCMD_Z has the largest accuracy (96.48%) with minimum FPR (3.43%) and FNR (4.40%). Results from DetRD-PCA and DetRPCA are better than RANSAC and MSAC, but DetRD-PCA and DetRPCA have higher rates of masking (FNR) and swamping (FPR) comparing with the proposed methods. Figures 4.12(c and d) are the result after removing the noise using MCMD_Z and MCMD_MD respectively, which are similar to Figure 4.12a before adding noise. Results also demonstrate that a few real noise points in the red rectangular in Figure 4.12a are completely removed in Figure 4.12c (MCMD_Z) and in Figure 4.12d (MCMD_MD).

Table 4.9 Performance evaluation for denoising.

Methods	CIN	CIR	TPR	TNR	FPR	FNR	Acc
RANSAC	1404	18559	64.34	85.05	14.95	35.66	83.17
MSAC	1515	19057	69.43	87.34	12.66	30.57	85.71
DetRD-PCA	1920	20209	87.99	92.62	7.38	12.01	92.20
DetRPCA	1909	20088	87.49	92.06	7.94	12.51	91.65
MCMD_Z	2086	21071	95.60	96.57	3.43	4.40	96.48
MCMD_MD	2075	20921	95.10	95.88	4.12	4.90	95.81

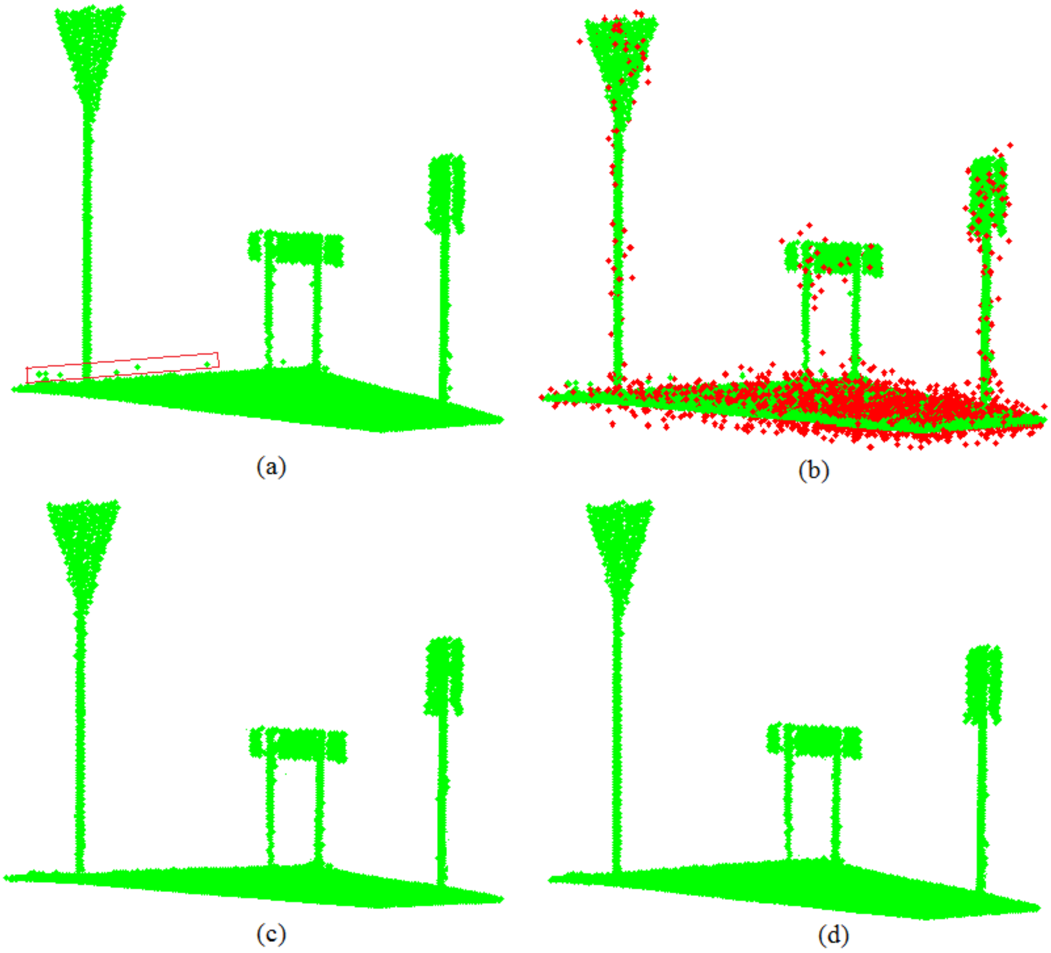


Figure 4.12 Signpost dataset; point cloud denoising: (a) original point cloud data, (b) data with 10% noise (red points) added, (c) result using MCMD_Z, and (d) result using MCMD_MD.

4.5.3.2 Sharp Feature Preservation and Recovery

Accurate and robust normals can be used for detecting and extracting sharp features (lines/edges/corners). Many methods have been introduced for sharp feature recovery (Fleishman et al., 2005; Li et al., 2010; Weber et al., 2012; Wang et al., 2013). This task is not easy because normals on or near sharp features become overly smooth mainly for two reasons: (i) neighbourhood points may come from multiple regions, and (ii) the presence of outliers and/or noise inherent due to the scanning process. The strength of the proposed algorithms is because they remove outliers from the local neighbourhood and depend only on the majority of consistent observations in the local neighbourhood. Hence they can automatically avoid the influence of outliers/noise and the points from the other regions that are small in number in the local neighbourhood. Then the normal represents the best-fit-plane and the surface from which the MCS (maximum consistent set) comes from.

Dataset 4.2: House dataset

Amenta and Bern (1999) mentioned that regression-based techniques tend to smooth sharp features, and thus fail to correctly estimate normals near edges. To investigate this, we choose a region (Figure 4.13b) near an edge from a real MLS dataset (Figure 4.13a) that captured a house, and estimate normals with $k = 20$. For better visualisation the view of the chosen region (blue points in Figure 4.13a) is rotated to align the two planes with the normals in the plane of the page. Figure 4.13c shows that PCA fails to get perfect normals near the edge and smooths the transition of the normals from one plane to the other across the edge because it computes PCA for varying proportions of the points belonging to the two planes. DetRD-PCA (Figure 4.13f) and DetRPCA (Figure 4.13g) normals are more accurately classified than those for RANSAC (Figure 4.13d). In Figure 4.13e, in the black circle, we see MSAC fails to preserve correct orientation for a point. The point from the right most surface shows the wrong orientation and is similar to the orientations of the points in the left most surface. This is a problem known as sampling anisotropy, which occurs especially when scanning objects with access constraints or abrupt variations (Boulch and Marlet, 2012). MCMD_Z (Figure 4.13h) and MCMD_MD (Figure 4.13i) efficiently construct the normals with correct orientation near edges, are able to retrieve sharp features better than the others, and can avoid the problem of sampling anisotropy.

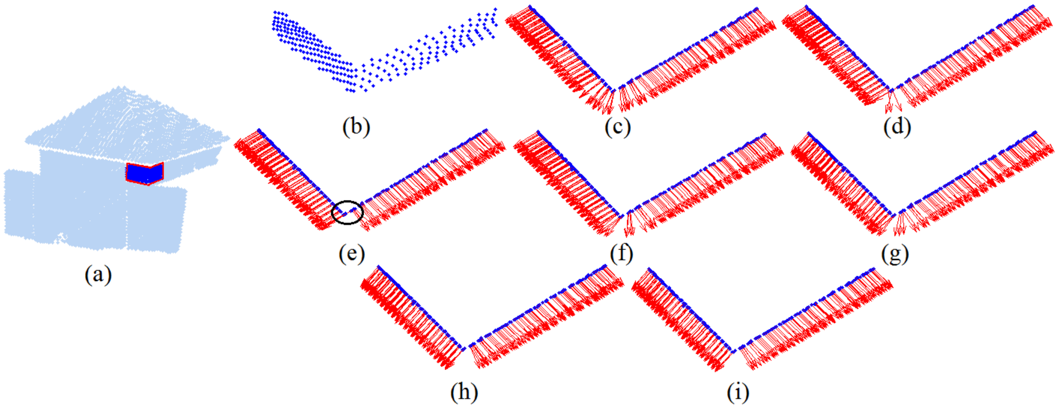


Figure 4.13 (a) Real point cloud data, (b) sample data for normal estimation, normals plots: (c) PCA, (d) RANSAC, (e) MSAC, (f) DetRD-PCA, (g) DetRPCA, (h) MCMD_Z, and (i) MCMD_MD.

Dataset 4.3: Box like and Crown dataset

For a more comprehensive example of sharp feature recovery, we pick two MLS datasets. One contains 3339 points from a 'box like' object (Figure 4.14a) consisting of three edges and a corner, and another dataset that is a part of a crown shaped roof extracted from a roadside building (Figure 4.16a), that we name the 'crown' dataset. The 'crown' dataset is of 3,118 points and represents a polyhedron having bilinear surfaces with common edges and corners. We know that the angle of the tangent planes for bilinear surfaces varies along the edges and could cause problems for feature detecting and reconstructing systems using global sets of parameters (Weber et al., 2012). The methods we use aim to solve this problem by using local instead of global parameters.

We use our proposed classification algorithm in Chapter 5 to extract the sharp features for the box like and crown datasets. The classification algorithm defines sharp features as outlying cases comparing with the surface points and uses a typical outlier identification rule based on the least eigenvalues estimated for the local region of the points in the data. The algorithm considers the i^{th} point as a sharp point (on an edge/corner) if its corresponding least eigenvalue is:

$$\lambda_0 > \bar{\lambda}_0 + a \times \sigma(\lambda_0), \tag{4.11}$$

where $\bar{\lambda}_0$ and $\sigma(\lambda_0)$ are the mean and StD of λ_0, and a is chosen to be 1 or 2 or 3 StD based on prior knowledge of the data or by investigating the scatter plot or histograms of the λ_0 values. We use $a = 1$ in the following applications.

We perform Algorithm 4.4 to fit planes for every point in each dataset with neighbourhood size $k = 30$, and calculate the least eigenvalues λ_0. Using Eq. 4.11 the results in Figures 4.14b and 4.16b show that PCA is not good for recovering the edge/corner points correctly for both the datasets. Even RANSAC (Figure 4.14c) and MSAC (Figure 4.14d) do not successfully classify surface and edge/corner points. Many surface points appear as edge/corner points because of the smoothing effect around edges and corners. Figure 4.14g (MCMD_Z) and Figure 4.14h (MCMD_MD) show that new methods perform more accurately than the others. Figures 4.16(g and h) show that the proposed methods efficiently recover sharp features even in the presence of bilinear surfaces. To see the performance of the methods in the presence of noise, we deliberately add 20% artificial Gaussian noise with mean 0.0 and StD 0.1 to the dataset in Figure 4.14a. In Figures 4.15(b–d), results for PCA, RANSAC and MSAC for the noisy data (Figure 4.15a) are now worse than in Figures 4.14(b–d). More surface points are misclassified as edges/corners. The proposed methods still produce better identification than the other existing methods for edge and corner points as shown in Figures 4.15(g and h). We see MCMD_Z and MCMD_MD are competitive with DetRD-PCA and DetRPCA. We have showed MCMD_Z and MCMD_MD take less time than DetRD-PCA and DetRPCA, and MCMD_MD tolerates more (75%) outliers than MCMD_Z.

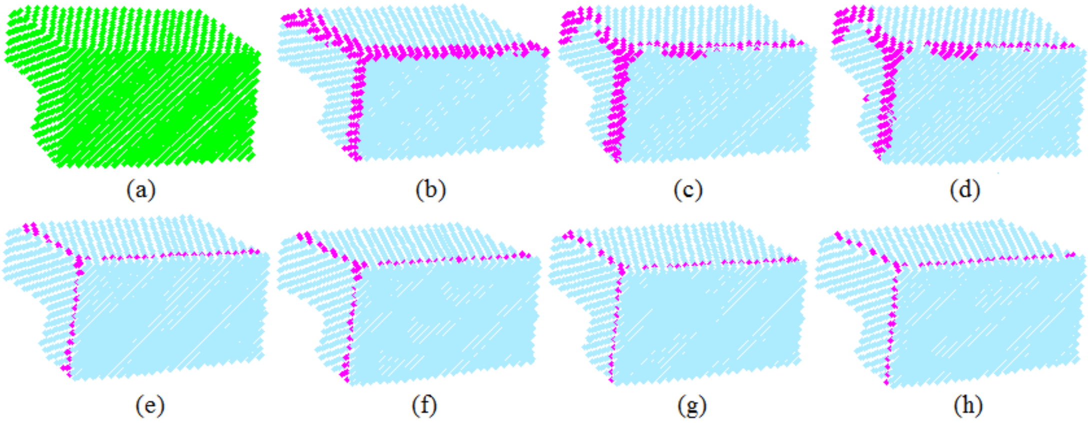


Figure 4.14 Classification of object points or sharp feature (edge and corner points) recovery for the box like dataset: (a) real data. Results for: (b) PCA, (c) RANSAC, (d) MSAC, (e) DetRD-PCA, (f) DetRPCA, (g) MCMD_Z, and (h) MCMD_MD.

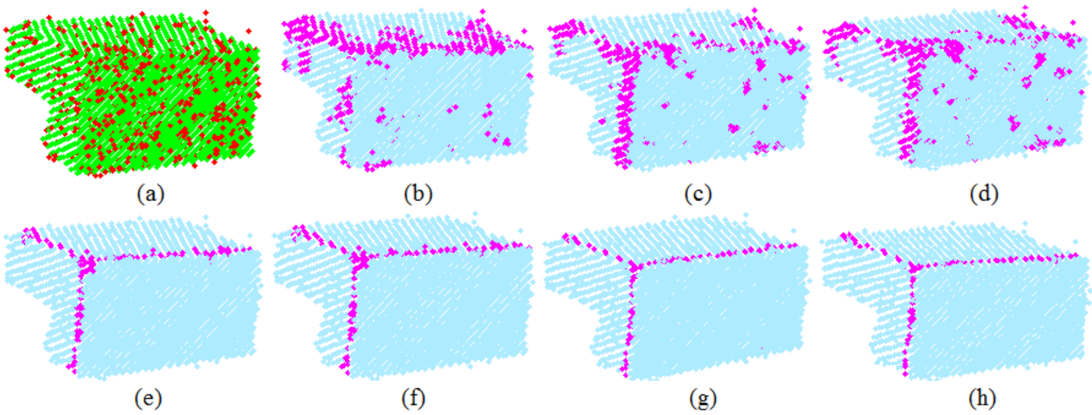


Figure 4.15 Classification of object points or sharp feature (edge and corner points) recovery for the noisy box like dataset: (a) real data with noise (red points). Results for: (b) PCA, (c) RANSAC, (d) MSAC, (e) DetRD-PCA, (f) DetRPCA, (g) MCMD_Z, and (h) MCMD_MD.

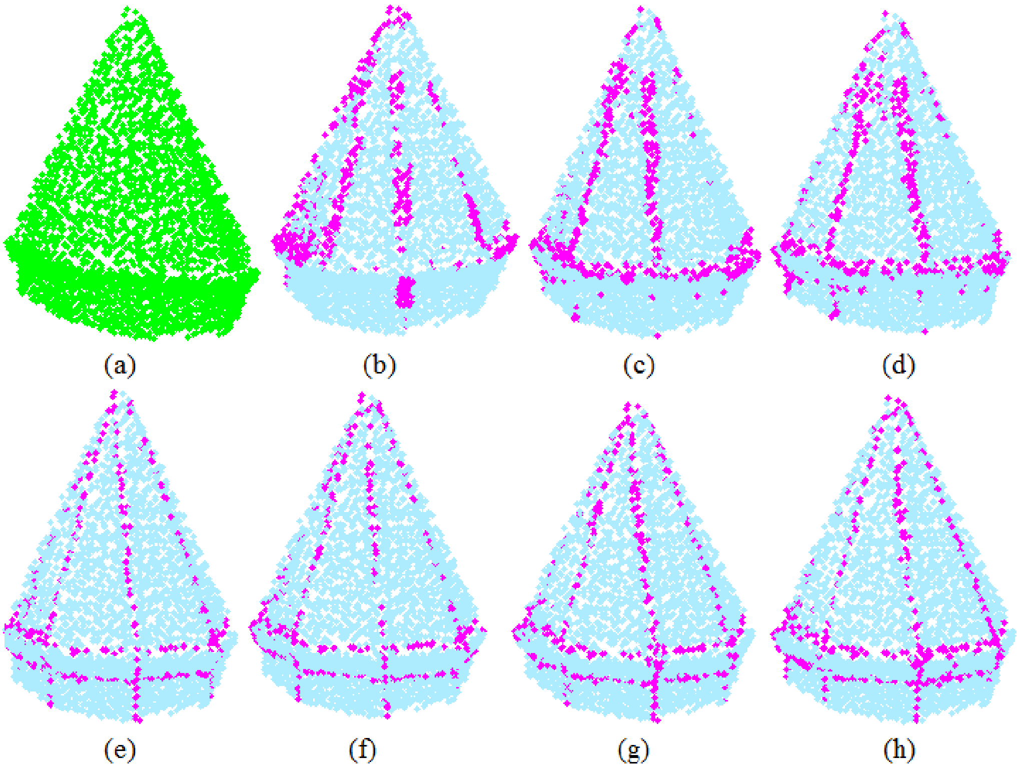


Figure 4.16 Classification of object points or sharp feature (edge and corner points) recovery for the crown dataset: (a) real data. Results for: (b) PCA, (c) RANSAC, (d) MSAC, (e) DetRD-PCA, (f) DetRPCA, (g) MCMD_Z, and (h) MCMD_MD.

4.5.3.3 Segmentation

Segmentation extracts and groups homogeneous points and labels them as the same regions. We evaluate the estimated normals and curvatures from the proposed methods based on our proposed segmentation Algorithm 5.4 in Chapter 5. The segmentation algorithm is based on a region growing approach that starts by searching for a seed point which has the minimum curvature $\sigma(p)$ value in the dataset. The algorithm grows regions using local surface point criteria (normal and curvature). The local surface point criteria are calculated based on the k nearest neighbourhood Np_i for all the points in the data. The algorithm considers Euclidean Distance ED$_{ij}$ between the seed point p_i and one of its neighbours p_j, Orthogonal Distance OD$_j$ for the j^{th} point to the best-fit-plane of the i^{th} seed point and its neighbours, and the angle difference θ_{ij} between the seed point p_i and p_j. A neighbour p_j of the seed point p_i will be added to the current region R_c and the current seed point list S_c if:

$$\left.\begin{array}{ll} (i) & \mathrm{OD}_j < \mathrm{OD}_{th}, \\ (ii) & \mathrm{ED}_{ij} < \mathrm{ED}_{th}, \text{and} \\ (iii) & \theta_{ij} < \theta_{th}. \end{array}\right\} \tag{4.12}$$

where OD_{th}, ED_{th} and θ_{th} are the thresholds of the respective characteristics. OD_{th} and ED_{th} are fixed automatically within the segmentation algorithm, and θ_{th} is a user defined threshold (see Chapter 5). R_c will grow until no more candidate points are available, and a segment is considered as significant if its size is larger than a predefined threshold for minimum region size R_{min}. We evaluate the estimates of normals and curvatures from the proposed MCMD_Z and MCMD_MD and the existing methods used previously for point cloud segmentation using the following two sets of real point cloud data.

Dataset 4.4: Traffic furniture dataset

The first dataset (Figure 4.17a), acquired by a MLS system consists of 25,731 points that describes part of a footpath, and contains road side furniture including road signs, long and approximately cylindrical surfaces (signs and light poles). We label this as the 'traffic furniture' dataset. It contains twenty one (nine planar and twelve non-planar) surfaces. For the segmentation algorithm, we set parameters $k = 50$, $\theta_{th} = 12°$, and minimum region size $R_{min} = 10$. The parameters are fixed based on similar preliminary data experiments. Segmentation results are in Figure 4.17 and Table 4.10. The points that are classified as belonging to each segmented surface are shown in the same colour. We evaluate the segmentation results based on Proper Segment (PS), Over Segment (OS) and Under Segment (US) values. A proper segment is recognized as a true segment from manually determined ground truth. An over segment is found when a true segment is wrongly broken into more segments, and under segment occurs when more than one true segment is wrongly joined to one segment. Our experiment shows PCA segmentation results are the worst: only 14 surfaces are properly segmented with seven OS and one US. RANSAC segments 17 surfaces properly but produces five OS. MSAC (Figure 4.17d) has 17 PS with six OS. DetRD-PCA (Figure 4.17e), DetRPCA (Figure 4.17f), MCMD_Z (Figure 4.17g), and MCMD_MD (Figure 4.17h) accurately segment all 21 (nine planar and 12 non-planar) surfaces without any OS and US.

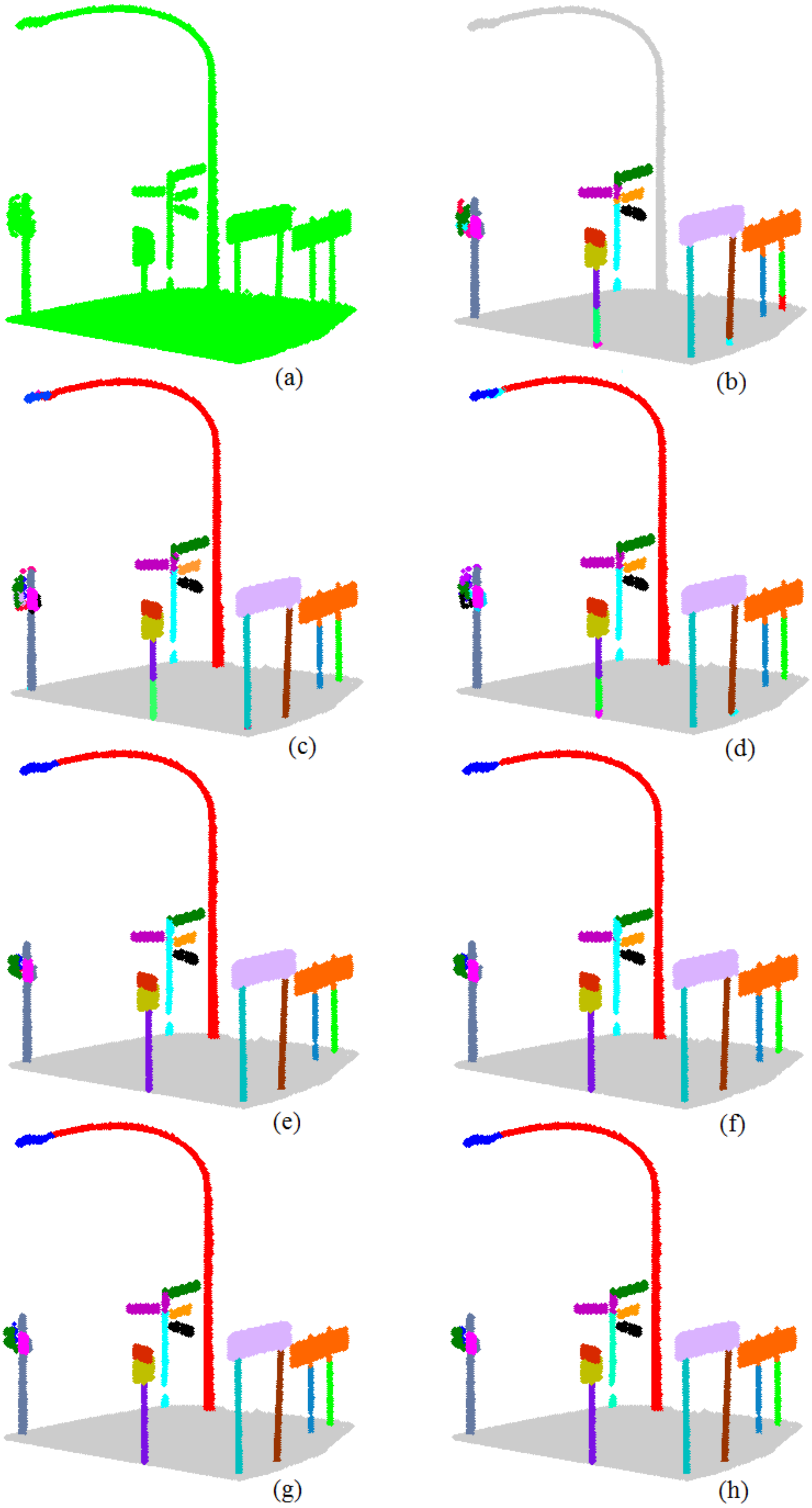


Figure 4.17 (a) Traffic furniture dataset. Segmentation results: (b) PCA, (c) RANSAC, (d) MSAC, (e) DetRD-PCA, (f) DetRPCA, (g) MCMD_Z, and (h) MCMD_MD.

Dataset 4.5: Bus stop dataset

Our last dataset consists of 53,024 points and is also acquired by MLS. It includes a building roof, bus shelter, bench, umbrella, sign post, sign board, road, kerb and footpath. We label this as the 'bus stop' dataset, which is shown in Figure 4.18a. We use the same segmentation algorithm used for the traffic furniture dataset with the parameters $k = 50$, $\theta_{th} = 5°$ and $R_{min} = 10$. It is clear in Figure 4.18a that several surfaces are incomplete and the point density is not homogenous. The dataset contains 51 different planar and non-planar surfaces. Results in Figures 4.18(b and c) show that PCA and RANSAC fail to segment or separate the road, kerb and footpath properly. In addition they have many OS and US. The other robust methods segment the three surfaces (road, kerb and footpath) properly. They also successfully preserve sharp features and properly segment the parts of the canopies of the three umbrellas. Results for the bus stop dataset in Table 4.10 show that MCMD_Z (Figure 4.18g) and MCMD_MD (Figure 4.18h) properly segment 48 and 50 surfaces respectively. MCMD_Z and MCMD_MD both have only one US and OS respectively, whereas PCA, RANSAC, and MSAC have 30, 21 and 19 occurrences of OS respectively.

Table 4.10 Performance evaluation in segmentation.

Methods	Traffic furniture dataset				Bus stop dataset			
	TS	PS	OS	US	TS	PS	OS	US
PCA	25	14	7	1	69	19	30	11
RANSAC	25	17	5	0	72	32	21	2
MSAC	27	17	6	0	70	35	19	3
DetRD-PCA	21	21	0	0	61	43	9	0
DetRPCA	21	21	0	0	61	43	8	0
MCMD_Z	21	21	0	0	50	48	0	1
MCMD_MD	21	21	0	0	52	50	1	0

4.6 Conclusions

Based on robust and diagnostic statistical approaches, this chapter proposes two outlier detection and robust saliency features such as λ_0, normal and curvature estimation methods, for laser scanning 3D point cloud data. The developed methods combined basic ideas of robust and diagnostic statistics. First, the

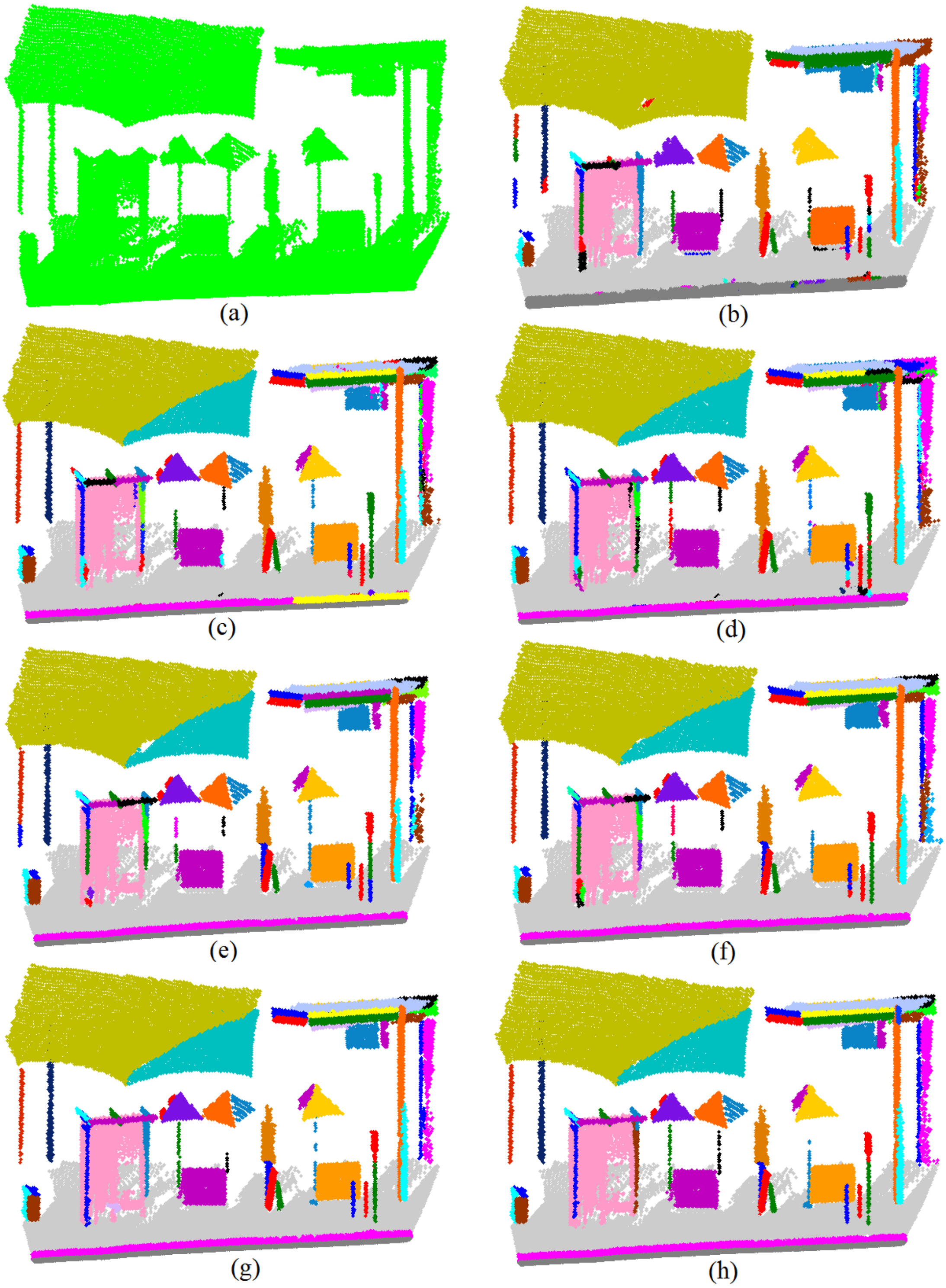


Figure 4.18 (a) Bus stop dataset. Segmentation results: (b) PCA, (c) RANSAC, (d) MSAC, (e) DetRD-PCA, (f) DetRPCA, (g) MCMD_Z, and (h) MCMD_MD.

algorithms estimate the best-fit-plane based on the majority of consistent data within the local neighbourhood of each point of interest. Then find the outliers locally for every point in their respective neighbourhood based on the results from the majority of good points. Second, the algorithms employ PCA to get the required saliency features: variation along the surface normal, normals and curvatures for every point based on the inlier points (after removing the outliers) found in its local neighbourhood. Results for simulated and real point cloud data show that the methods have various advantages over other existing techniques including: (i) being computationally simpler, (ii) being able to efficiently identify high percentages of clustered and uniform outliers, (iii) being more accurate and robust than PCA and RANSAC, (iv) being significantly faster than robust versions of PCA, (v) being able to denoise point cloud data, and (vi) being more efficient for sharp feature recovery. Moreover, the robust saliency features based on the proposed techniques can reduce over and under segmentation, and give significantly better segmentation results than existing methods for planar and non-planar complex surfaces. In summary, results reveal that the newly proposed MCMD_Z and MCMD_MD methods are more accurate than several well-known existing robust methods: robust PCA, RANSAC and MSAC and outlier detection methods such as LOF, q_{S_p} and uLSIF from the computer vision, data mining, pattern recognition, machine learning and statistics literature.

The next chapter will investigate the problems of outliers and/or noise for segmentation processes and proposes robust segmentation algorithms in laser scanning point cloud data.

Chapter 5

"If I can't picture it, I can't understand it."
Albert Einstein

"The greatest value of a picture is when it forces us to notice what we never expected to see."
John Tukey

Robust Segmentation

5.1 Introduction

Segmentation is a process of classifying and labelling the locally homogenous data points into a number of separate groups or regions, each corresponding to the specific shape of a surface of an object. It is a fundamental, and actively researched, task for many applications including object shape recognition, geometry analysis, modelling, surface reconstruction, and feature extraction those are widely used in subjects such as computer vision, computer graphics, image processing, pattern recognition, photogrammetry, remote sensing, reverse engineering and robotics (Huang and Menq, 2001; Liu and Xiang, 2008; Klasing et al., 2009; Liang et al., 2011; Heo et al., 2013; Barnea and Filin, 2013; Michel et al., 2014). It is recognized as an important task in processing point cloud data obtained from LiDAR or laser scanning, and the quality of the results largely determines the success of information retrieval (Besl and Jain, 1988; Wang and Shan, 2009). Hoover et al. (1996) gave a formal definition of segmentation in range image analysis. Many authors (Rabbani, 2006; Dorninger and Nothegger, 2007; Wang and Shan, 2009) adopt this definition in point cloud segmentation. Let R represents a spatial region of a whole point cloud P. The objective of the segmentation is to partition R into a number of sub-regions R_i, such that:

(i) $\bigcup_i^n R_i = R$
(ii) $R_i(p_i) \neq \phi$ for any i, $\bigcup_i^n R_i(p_i) = P$
(iii) R_i is a connected region, $i = 1, \ldots, n$
(iv) $R_i \cap R_j = \phi$ for all i and j, $i \neq j$
(v) $L(R_i) = \mathit{True}$ for $i = 1, \ldots, n$, and
(vi) $L(R_i \cap R_j) = \mathit{False}$ for $i \neq j$,

where $L(R_i)$ is a logical predictor over the points p_i in the region R_i, which is a measure of similarity that groups the points in one region and separates them from the points in other regions (Wang and Shan, 2009).

For point cloud data obtained from laser scanning and defined as a collection of unorganized points of geo-referenced x, y, z coordinates, segmentation is not trivial because the 3D points in the point cloud data are usually incomplete, sparse, have uneven point density, are not connected to each other, and there is no knowledge about the statistical distribution or any boundaries that separate the data into groups. Complex topology and the presence of singularities and/or sharp features (e.g. edges and corners) in object surfaces further exacerbate the intrinsic complexity. Outliers and noise are common in laser scanning data. The gross outliers, clusters and pseudo-outliers frequently occur in point cloud data because of the presence of multiple model structures in the same dataset. Therefore, the points appear as outliers to one structure of interest but inliers to another structure. The presence of different types of outliers and noise make the segmentation process challenging and more complicated. The sources and causes of outliers were described in Chapters 3 and 4.

Segmentation based on region growing is one of the most commonly used segmentation approaches. Principal Component Analysis (PCA) has been widely used to estimate local saliency features (normals and curvature) that are used as the cornerstones for many region growing based segmentation procedures (Hoppe et al., 1992; Pauly et al., 2002; Rabbani, 2006; Belton, 2008). The influence of outliers/noise on the estimated normals and curvatures used in segmentation can produce several problems including the tendency to smooth sharp features. Mitra et al. (2004) showed that the sensitivity of PCA to outliers means it fails to fit a plane accurately. We have shown in Chapters 3 and 4 that the resultant plane parameters from PCA are unreliable, non-robust and misleading. Hence, the segmentation results based on plane parameters and related saliency features can be inaccurate, unreliable and non-robust. Li et al. (2010) pointed out that if correct normals are robustly estimated for each point, the geometry of even strongly corrupted point-clouds can be perceived. Chapters 3 and 4 show that robust and diagnostic statistical approaches with PCA can provide principal components that are not much influenced by outliers/noise and can produce robust normals and curvatures that can be used for robust and reliable segmentation.

This chapter proposes two region growing based robust segmentation algorithms and one merging algorithm for laser scanning produced 3D point cloud data. The algorithms use robust diagnostic PCA to reduce outlier influence on the estimates and to get reliable local saliency features and use these for region growing. Using robust saliency features, the new algorithms produce accurate and robust segmentation results for complex objects surfaces. We show the accuracy, efficiency and robustness of the proposed algorithms for segmenting point cloud data in the presence of outliers and/or noise, singularities (e.g. edges and corners), and open surface boundaries from planar and non-planar complex objects. The proposed methods are favorable because of their high resistance to outliers and noise. Using accurate and robust local surface point properties, these can reduce over and/or under segmentation. The algorithms are efficient for non-planar smooth surfaces as well as for planar surfaces. A merging algorithm is developed to join the segmentation results of sliced point cloud data. Slicing into smaller point clouds is needed to deal with large volumes of point cloud data that can be generated e.g. by vehicle based Mobile Mapping Systems (MMS).

Arrangement for the following sections are: Section 5.2 gives a brief literature review. Section 5.3 has two main subsections that illustrate a number of issues that need to be addressed before considering the robust segmentation algorithms. Section 5.4 proposes an algorithm for robust segmentation to extract multiple planar surfaces. Section 5.5 proposes a second robust segmentation algorithm which is able to segment both planar and non-planar surfaces for complex objects. A merging algorithm is introduced in Section 5.6.

Within the respective sections, the proposed algorithms are demonstrated, evaluated and compared through experiments using synthetic and real laser scanning point cloud datasets. In Section 5.7, we give a brief indication of how the segmentation results can be used for object class recognition and can help in 3D modelling. Section 5.8 concludes the chapter.

5.2 Literature Review

Many algorithms have been developed to improve the performance and quality of the segmentation process. Existing segmentation algorithms can be classified into three main categories: (i) edge/border based (Huang and Menq, 2001; Lee et al., 2004), (ii) region growing based (Besl and Jain, 1988; Xiao et al., 2013), and (iii) hybrid (Koster and Spann, 2000; Woo et al., 2002).

For the edge/border based segmentation methods, usually points positioned on the edges/borders are identified, a border linkage process constructs the continuous edge/border, and then points that are within the identified boundaries and connected edges are grouped. Huang and Menq (2001) developed their border based approach by using three successive steps: (i) manifold domain construction, (ii) border detection, and (iii) mesh patch grouping. Castillo et al. (2013) pointed out that such methods often detect disconnected edges that make it difficult for a filling procedure to identify closed segments because of noise or spatially uneven point distributions.

Segmentation algorithms based on the region growing approach can be classified roughly into two types: (i) grid-based, and (ii) point-based. Xiao et al. (2011) presented grid-based and sub-window based region growing algorithms. The grid-based algorithm is argued to be better for structured data such as images, whereas the point-based region growing algorithm works for both structured and unstructured data (Xiao et al., 2011). 3D point clouds are unstructured because the location of a point relative to, say, its nearest neighbours cannot be determined from the location or index of the point. Since we are dealing with unstructured point cloud data, we concentrate on point-based algorithms. In the point-based approach, usually a seed point is chosen first to grow a region, then local neighbours of the seed point are grouped with the seed point if they have similar surface point properties such as curvature and orientation. In point-based region growing, only one point is added at a time, but some algorithms consider a sub window or line segment as the growth unit. This attempts to obtain homogeneity within regions and/or correspondingly dissimilarities between the different regions. Harati et al. (2007) proposed line-based region growing using the so-called bearing angle as a flatness metric from the pixel neighbourhood information. A problem with this method is that the bearing angle cannot be calculated correctly in a cluttered environment (Xiao et al., 2013). In general, region growing methods are more robust to noise than edge-based methods when using global information (Huang and Menq, 2001; Liu and Xiang, 2008). But, this type of method can suffer from the possibility of over and under segmentation, the problem of determining region borders accurately, and sensitivity to the location of initial seed regions (Chen and Stamos, 2007; Liu and Xiang, 2008). Several authors use a smoothness constraint for finding smoothly connected areas, usually those that match curvature and surface normals (Rabbani et al., 2006; Klasing et al., 2009). Smoothness constraints based on residuals and curvature can segment planar and non-planar or curved objects, but the inaccurate estimates of the normals and curvatures of points near region boundaries can cause inaccurate segmentation results, and the presence of outliers and/or noise can create problems of over and/or under segmentation.

Both the boundary/edge and region growing based segmentation approaches (e.g. Woo et al., 2002) are used simultaneously in hybrid methods to overcome the limitations in the respective approaches and give better results. Hence, the success of hybrid methods depends on the success of either or both of the methods.

Excluding the above three main approaches, many other methods have been introduced. Lari and Habib (2014) proposed an adaptive approach for segmentation of planar and cylindrical features using the internal characteristics (e.g. local point density variation) of the utilized point cloud. Koster and Spann (2000) developed a clustering approach based on similarity measures from a statistical test. Scan-line based methods (Jiang et al., 2000; Khalifa et al., 2003) adopt a split-and-merge strategy based on grouping the scan lines along a given direction. The extension of scan-line based methods into point clouds requires deciding directions and constructing scan lines by slicing point clouds which makes segmentation depend on orientation (Wang and Shan, 2009). The approach is not good for unordered/unstructured point clouds having uneven point density because it is based on the grouping of the scan lines. These types of situations commonly appear in 3D point cloud data. The region growing algorithm of Hähnel et al. (2003) learnt a smooth model in indoor and outdoor environments. However it is slow due to it employing the octree for the nearest neighbour searching algorithm. Poppinga et al. (2008) extended the work of Hähnel et al. (2003) to make the algorithm faster by optimizing pixel information, and presented an efficient method of plane fitting by Mean Squared Error (MSE) computation. Later, Xiao et al. (2013) used MSE and developed a region growing approach, which is good for planar surface segmentation. Marshall et al. (2001) used Least Squares (LS) fitting and identified surfaces of known geometric features within a segmentation framework, and the authors of the paper concluded that generalizing this method to more complex surfaces would be hard. Klasing et al. (2009) identified the limitations of fitting higher order surfaces and geometric primitive based methods including the problem of predicting the segmentation results as well as the high computational cost for a large number of features. Schnabel et al. (2007) partly reduced the complexity by employing RANSAC in segmentation. Benkö and Várady (2004) proposed a segmentation method that is good for smoothly connected regions using various tests including error measures, similarity, and geometric and statistical indicators. Castillo et al. (2013) introduced a point cloud segmentation method using surface normals computed by the constrained nonlinear least squares approach. Crosilla et al. (2009) carried out statistical analysis of Gaussian and mean surface curvature for each sampled point for segmentation of laser point clouds.

5.3 Steps to Proposed Algorithms

In this section, we investigate the problems of object surface segmentation in point cloud data and set up appropriate criteria to implement the proposed segmentation algorithms described later.

5.3.1 Problem Formulation

It is a fundamental idea to segmentation that points in a segment have similar characteristics. First, we investigate the problems of identifying the underlying pattern of surface points, and then formulate the expected characteristics for the points to be in the same object surface.

It is logical to assume that every point in a sufficiently-small (reasonably sized) local area (neighbourhood) is on a planar surface. This assumption of points on a plane is useful, and leads to using the information about local saliency features to check the behaviour of a point on a smooth surface. Figure 5.1 (shown in one dimension for clarity) illustrates that under certain conditions points on different local planar surfaces (Figure 5.1a) may be on the same smooth surface (Figure 5.1b), where three different planes of different orientations appeared as a single smooth surface. We see three planar surfaces (Figure 5.1a) that appear to have discontinuities (gaps) between them. The first two planes from the left appear to have a crease edge and the last two planes appear to have a step edge at their boundaries. If the gaps between the two boundary points of different planar surfaces are not enough to consider them separate then they may be co-surface points under certain coherence criteria, otherwise discontinuity appears in the gaps.

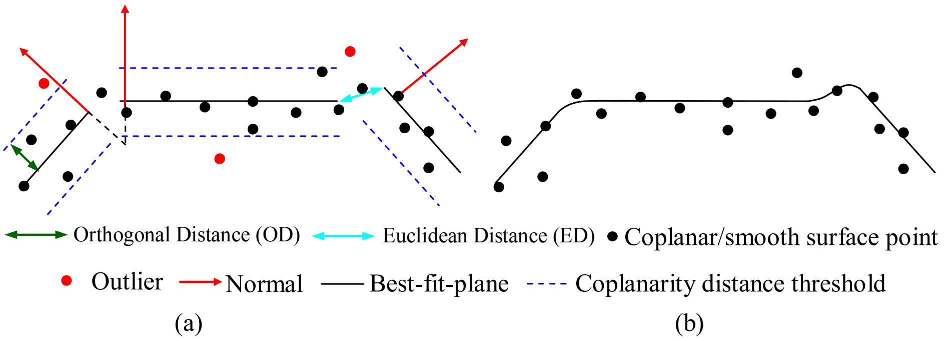


Figure 5.1 (a) Three different planar surfaces, and (b) co-planar smooth surface.

5.3.1.1 Edges, Gaps and Outliers

In this Section, we explain why edges, gaps and outliers should be considered carefully at the time of segmentation. Figure 5.1 (in the previous section) shows that points near gaps between two boundary points and edges/corners need more attention when determining which points belong to which specific surface. Edges and gaps are the two situations where properties or attributes of the surface point may be falsely estimated. The presence of outliers may intensify the problems of edges and gaps. Therefore, exploring the properties of edges, gaps and outliers in the data needs to be investigated for a proper understanding about an object's surfaces, and will help when estimating reliable surface point attributes and for formulating appropriate test criteria for robust and accurate segmentation.

Edges: Many authors use edge/corner information to separate different surfaces. Near edges and corners, known as geometric singularities, normals are usually differently oriented and discontinuous. Hoffman and Jain (1987) stated that edge points may delineate surface patches and therefore be useful for modelling. A common effect is rounded or smoothed normal estimates along edges (Castillo et al., 2013). Three most common types of edges in point cloud data shown in Figures 5.2(a, b and c) are: (i) step/jump edges that occur where a surface undergoes a discontinuity and the boundary points on the two parallel planes close to the discontinuity have the same orientation, (ii) crease/corner edges e.g. where two sides of a roof meet, and (iii) smooth or virtual edges that can be characterized by continuity of the orientation of normals i.e. smoothly changing across the surface, but discontinuities of curvature e.g. where curvature goes from +ve to −ve suddenly. Usually, a step edge appears when an object obstructs another object, and for the crease edges, the normals of the surface points are influenced by different planes.

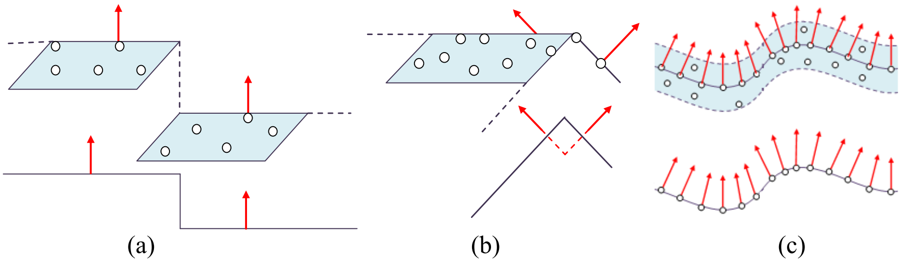


Figure 5.2 Three types of edges; normals (red arrows): (a) step edge, (b) crease edge, and (c) smooth edge.

Gaps: Improper sensor alignment, error in data acquisition because of faulty sensors, unexpected interruption in data collection, surface point density variation and/or obstacles that may obstruct the laser pulse may cause gaps such as discontinuities and holes in the data. Figure 5.3 shows some types of gaps that are common in point cloud data. There is a possibility to wrongly join the two different surfaces together into one segment if the gaps between two individual surface points cannot be identified properly. In addition, real gaps can be filled by faulty boundary extension in the presence of outliers/noise. So, a thorough analysis of the neighbouring surface points based on their proximity criteria is useful for a proper understanding about the gaps between the relevant neighbouring points, and helps to avoid the problems of misleading gaps.

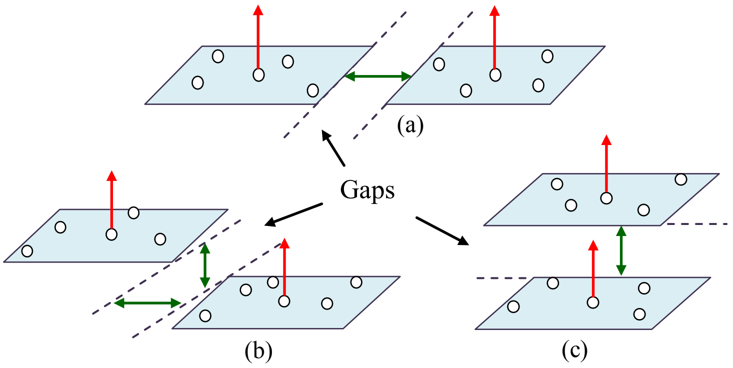


Figure 5.3 Gaps in different surface positions, red and green arrows show normal orientation and directions of gaps, respectively: (a) gap between two horizontally distant planes, (b) gap between two horizontally as well as vertically distant planes, and (c) gap between two vertically distant planes.

Outliers: Usually outliers are classified as the points that are far from the majority of points in the data, and/or do not follow the same pattern as the majority of points (Barnett and Lewis, 1995; Rousseeuw and Leroy, 2003). Moreover, the existence of multiple structures in a dataset may create psuedo-outliers. In a general sense, noise can appear as off-surface points and behave like outliers in many cases (Sotoodeh, 2006). Covariance statistics based on an outlier contaminated local neighbourhood may produce inaccurate normals and curvatures. The presence of outliers in different positions (especially around or on the edges and boundaries) on a surface causes errors in the estimates of local saliency features such as normals. For example, the effect may cause continuous/smoothed normals along the edges and corners. Outliers between two points in a neighbourhood can produce erroneous discontinuities in a homogeneous surface. Points in a local neighbourhood in the presence of outliers results in the tangent plane being biased to the direction of the outliers. The inclusion of outliers between the gaps of two neighbouring surface points can erroneously join two surfaces. Figure 5.4 illustrates the influence of outliers in different positions in a local neighbourhood that causes the change of real orientation of the local plane. Figure 5.4a shows how the presence of an outlier between two vertically distant parallel planes changes the orientations of the two, and wrongly joins them together, Figure 5.4b shows how an off-surface point may appear as an outlier and changes the orientation of a plane to its own direction, and Figure 5.4c shows how the presence of an outlier between a pair of horizontally distant surfaces joins them erroneously.

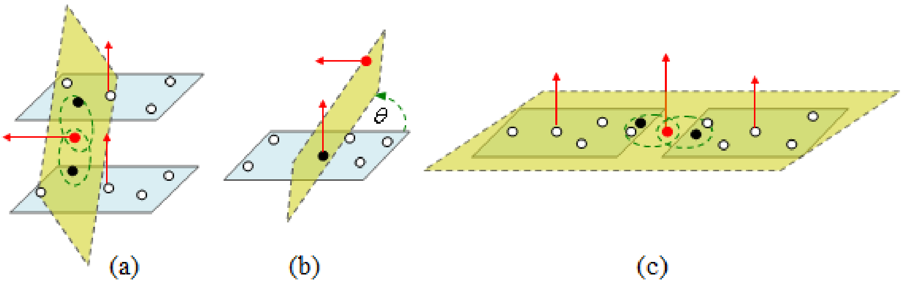


Figure 5.4 Influence of the presence of an outlier (red point) in different positions: (a) outlier between two vertically distant parallel planes, (b) outlier as an off-surface point, and (c) outlier between two horizontally distant co-planar surfaces.

5.3.2 Robust Saliency Features Estimation

The plane normal, the variation along the plane/surface normal, and the curvature defined as the surface variation, are the three most important geometric properties that have been used in many applications to find the coherence and proximity for points in a point cloud (Pauly et al., 2002; Rabbani et al., 2006; Belton, 2008; Wang et al., 2012b; Lin et al., 2014).

Normals: Visual surface orientation has been widely used for object surface reconstruction, recognition and 3D modelling, which can be represented by the unit normal of the fitted plane at a point p_i and for its local neighbourhood. Segmentation can be considered as a pre-stage of surface reconstruction, and in surface reconstruction the quality of the approximation of the output surface depends on how well the estimated normals approximate the true normals of the sampled surface (Dey et al., 2005). Robust and accurate normals are the preconditions to detect and reconstruct sharp features. In PCA, the third PC is orthogonal to the first two PCs and is considered as the estimate of the normal $\hat{n}$ to the fitted plane of the neighbourhood of p_i. A recent survey (Klasing et al., 2009) showed that the PCA based approach (Hoppe et al., 1992) is one of the most efficient and popular methods of normal estimation. Yoon et al. (2007) specified that the presence of outliers in normal estimation is the most likely source of problems when using state-of-the-art surface reconstruction and segmentation techniques. Since PCA is influenced by outliers, normals estimated by PCA are not free from outlier effects. In addition, PCA normals may make a sharp edge smooth, which reduces the angle difference between two consecutive normals that may cause under segmentation.

Variation along the surface normal: The least eigenvalue λ_0 that is related with the third PC (used as the normal), expresses the least amount of variation among the surface points through the third PC. The third PC is orthogonal to the first two PCs, and measures the variation of the points along the elevation or z axis, i.e. the variation along the normal. In other words, λ_0 estimates how much the points deviate from the tangent plane (Pauly et al., 2002). Therefore, it has been used as a measure of the noise level in the data. It can evaluate the quality of a plane fit: the smaller the value of λ_0 the better the quality of the plane fit. A planar surface should have a λ_0 value of theoretically zero as illustrated in Figure 5.5c. It is also revealed in Figure 5.5c that if a surface is curved or non-planar, then λ_0 will be non-zero and it should not be considered as an appropriate characteristic with which to measure the shape of a smooth curve.

Curvature: Stewart (1995) defined curvature as a measure of the rate of change of surface normals as well as in the tangential directions of a point on a surface. It is fairly popular in point cloud data analysis. Usually, truncation error and noise error have been used for imperfect curvature estimations. Truncation error is caused by sampling density, and the noise error is caused by the presence of outliers and/or noise. Huang and Menq (2001) pointed out that the truncation error can be effectively reduced by increasing the sampling density. However, increasing the sampling density also increases the frequency of noise components, thus enlarging the noise error for surface property estimation. There are many methods for curvature estimation that can be used in many ways in segmentation algorithms (Besl and Jain, 1988; Pauly et al., 2002; Rabbani et al., 2006; Sullivan, 2008). Gaussian and mean curvature is proposed (Besl and Jain, 1988), but it is criticized for over segmentation and inefficiency even on a very simple scene with low noise (Powell et al., 1998; Rabbani et al., 2006). Rabbani et al. (2006) proposed the residual with a percentile based cut-off value as the curvature measure. Klasing et al. (2009) pointed out the limitations of using the residual as the curvature measure in segmentation. He claimed that the residuals are not normalized. Pauly et al. (2002) defined surface variation at a point p as:

$$\sigma(p) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \quad \lambda_0 \leq \lambda_1 \leq \lambda_2 \tag{5.1}$$

where λ_i is the i^{th} eigenvalue. The authors mentioned that surface variation is closely related to curvature, and is more suitable for simplification of point-sampled surfaces than curvature estimation based on function fitting. We consider the surface variation in Eq. (5.1) to serve our purpose of understanding the local shape of a smooth surface and as an alternative to the curvature, because it has the benefits: (i) it is normalized, (ii) it is more consistent for different point densities and neighbourhood sizes, and (iii) it considers surface variations in all the three directions. We use surface variation as the curvature. The curvature $\sigma(p) = 0$ if all points lie on a plane as illustrated in Figure 5.5d, and the maximum value of $\sigma(p) = 1/3$ is assumed for completely isotropically distributed points (Pauly et al., 2002).

All three local saliency features: normal $\hat{n}$, λ_0 (variation along the normal), and curvature $\sigma(p)$ can be estimated by PCA. To avoid the vulnerability to the outliers and/or noise on the estimates, we can use a robust and/or diagnostic version of PCA. In Chapters 3 and 4, it is seen that performing existing robust and/or diagnostic statistical methods e.g. FRPCA, and FRD-PCA, usually takes more time than PCA, so it needs to use faster robust statistical methods to process large volumes of point cloud data. While searching for faster methods, at the same time we must ensure that results should be accurate and robust. Although, the recently proposed Deterministic MCD (DetMCD; Hubert et al., 2012) based methods can produce robust results that are faster than FRPCA and FRD-PCA (Chapter 3), DetMCD methods are still slower than Maximum Consistency with Minimum Distance (MCMD) based methods (Chapter 4). Adopting the MCMD algorithm described in Chapter 4 (an earlier version published in Nurunnabi et al., 2013c), we get faster computation with more accurate and robust results. We use the MCMD_Z method (Chapter 4) to find outliers and remove them from the local neighbourhood, and then fit the plane by PCA to the cleaned neighbourhood for getting robust local saliency features: $\hat{n}$, λ_0, and $\sigma(p)$. We define the method as MCMD_Z based robust diagnostic PCA. We name this Robust Diagnostic PCA (RDPCA) because it uses the MCMD_Z based robust diagnostic technique to find outliers in a local neighbourhood and then uses PCA to get the PCs for the saliency features. The process for robust local saliency feature estimation is summarized in Algorithm 5.1.

Algorithm 5.1: Robust Saliency Feature Estimation using RDPCA

1. Input: point cloud P, neighbourhood size k, and the Maximum Consistent Set (MCS) size h.
2. Determine the k nearest neighbourhood Np_i of a point p_i.
3. Determine the MCS using Algorithm 4.1 (Chaprer 4) for Np_i.
4. Calculate the Rz-score using OD for all $i \in Np_i$, where
$$Rz_i = \frac{|\mathrm{OD}_i - \underset{j}{median}(\mathrm{OD}_j)|}{\mathrm{MAD(OD)}}, \tag{5.2}$$
$$\mathrm{OD}(p_i) = (p_i - \bar{p}_h) \cdot \hat{n}_h, \tag{5.3}$$
where $\bar{p}_h$ and $\hat{n}_h$ are the mean vector and the normal of the MCS, respectively.
5. Find the outliers in Np_i using the Rz-score. Points with an Rz-score more than 2.5 are identified as outliers.
6. Perform classical PCA to the cleaned (excluding outliers) Np_i.
7. Arrange the three PCs associated with their respective eigenvalues from highest to lowest.
8. Output: robust normal $\hat{n}$, the least eigenvalue λ_0, and the curvature $\sigma(p)$.

5.3.2.1 Effects of Neighbourhood Size on λ_0 and $\sigma(p)$

We explore the effect of neighbourhood size k of a point p_i on λ_0 and $\sigma(p)$ based on real MLS data, and calculate λ_0 and $\sigma(p)$ values for a point p_i with size $k = 20$ to 300. Using some similar MLS data experiments, we observe that depending on the data, usually $k = 20$ to 200 or 300 is acceptable for MLS point cloud data processing. We pick a box like object shown in Figure 5.5a that contains three planar surfaces with three edges and a corner. The results in Figures 5.5(c and d) show that if the i^{th} point comes from a planar surface then there is little effect of changing the value of k on λ_0 and $\sigma(p)$ respectively, but the results also reveal that if the selected point p_i comes from a non-planar surface, shown in Figure 5.5b, then the value of λ_0 and $\sigma(p)$ are increasing with increasing size of k. However, it is seen that the values of the estimates for RDPCA are more linearly increasing and much smaller than the corresponding values for PCA for every size of k.

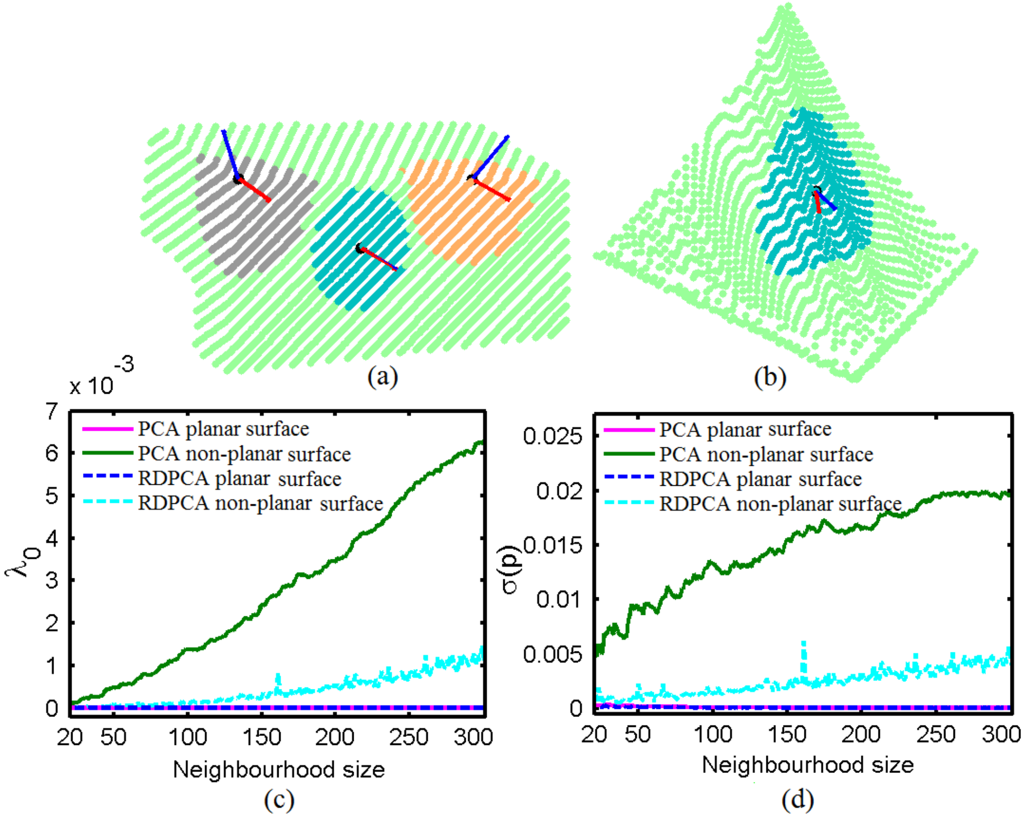


Figure 5.5 Effects of neighbourhood size on λ_0 and $\sigma(p)$: (a) planar surfaces, (b) non-planar surface, (c) λ_0 versus neighbourhood size, and (d) $\sigma(p)$ versus neighbourhood size.

Now we investigate and compare the effect of neighbourhood size on λ_0 and $\sigma(p)$ when the interest point p_i comes from a planar surface (cyan) or an edge (sandy brown) or a corner (ash) in Figure 5.5a. For PCA, Figure 5.6a shows that most of the λ_0 values are zero or near zero when the point p_i is from a planar surface. However, for points that come from edges or corners, λ_0 gradually increases with the size of k. The three results are clearly different with λ_0 increasing most for corners than for edges. That means the values of λ_0 for the points in a neighbourhood that come from most surfaces e.g. three for corner point, are larger than for those that come from less surfaces e.g. two for an edge point. For RDPCA (Figure 5.6c), we get λ_0 values almost equal to zero for surface, edge or corner points. We take the logarithm for the λ_0 values to make the difference more visible, but as shown in Figure 5.6c the results for surface, edge and corner points are still very close to zero, meaning there is no significant difference between them. The reason for this is that RDPCA considers the most consistent set of points that are from an individual surface, which can be understood by the orientation in Figure 5.5a of the respective normals (red) from RDPCA. In Figures 5.6(b and d), for $\sigma(p)$, almost similar conclusions as for the λ_0 can be drawn for the results using PCA and RDPCA respectively.

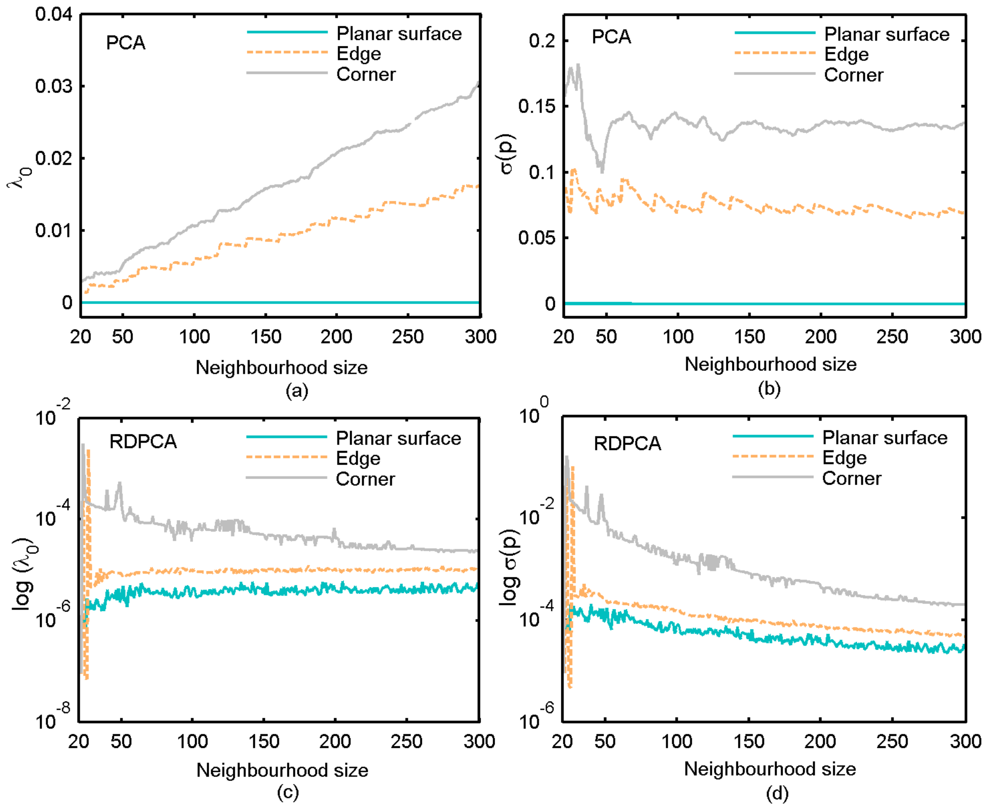


Figure 5.6 Effects of neighbourhood size on λ_0 and $\sigma(p)$ for the planar surface, edge and corner points: (a) PCA results, λ_0 versus neighbourhood size; (b) PCA results, $\sigma(p)$ versus neighbourhood size; (c) RDPCA results, $\log(\lambda_0)$ versus neighbourhood size; and (d) RDPCA results, $\log \sigma(p)$ versus neighbourhood size.

5.3.2.2 Robustness of λ_0 and $\sigma(p)$

This section demonstrates the influence of outliers and/or noise on the estimates of λ_0 and $\sigma(p)$. We add a range of 5% to 40% Gaussian noise with 0.1 variance to the dataset in Figure 5.5a. Figure 5.7a shows the dataset for Figure 5.5a to which 25% noise has been added. We consider the generated artificial noise as off-surface points and simply call these outliers. We pick a point from a planar surface in Figure 5.7a, find its neighbourhood of size $k = 30$, and calculate the values for λ_0 and $\sigma(p)$ for every percentage of noise contamination using PCA and RDPCA. We choose $k = 30$ based on an empirical study on similar real data. Figures 5.7(b and c) graph λ_0 versus outlier percentage and $\sigma(p)$ versus outlier percentage respectively. Figures 5.7(b and c) show λ_0 and $\sigma(p)$ for PCA (green) have much variation and increasing tendency with increasing percentage of outlier contamination. However, λ_0 and $\sigma(p)$ for RDPCA (magenta) are near zero for all values of percentage outlier contamination, which shows the values of the estimates for RDPCA are not influenced by noise/outliers i.e. are robust. Boxplots in Figures 5.7(d and e) are generated from the λ_0 and $\sigma(p)$ values from Figures 5.7(b and c) respectively, that explore significantly improved robustness of the estimated saliency features from RDPCA.

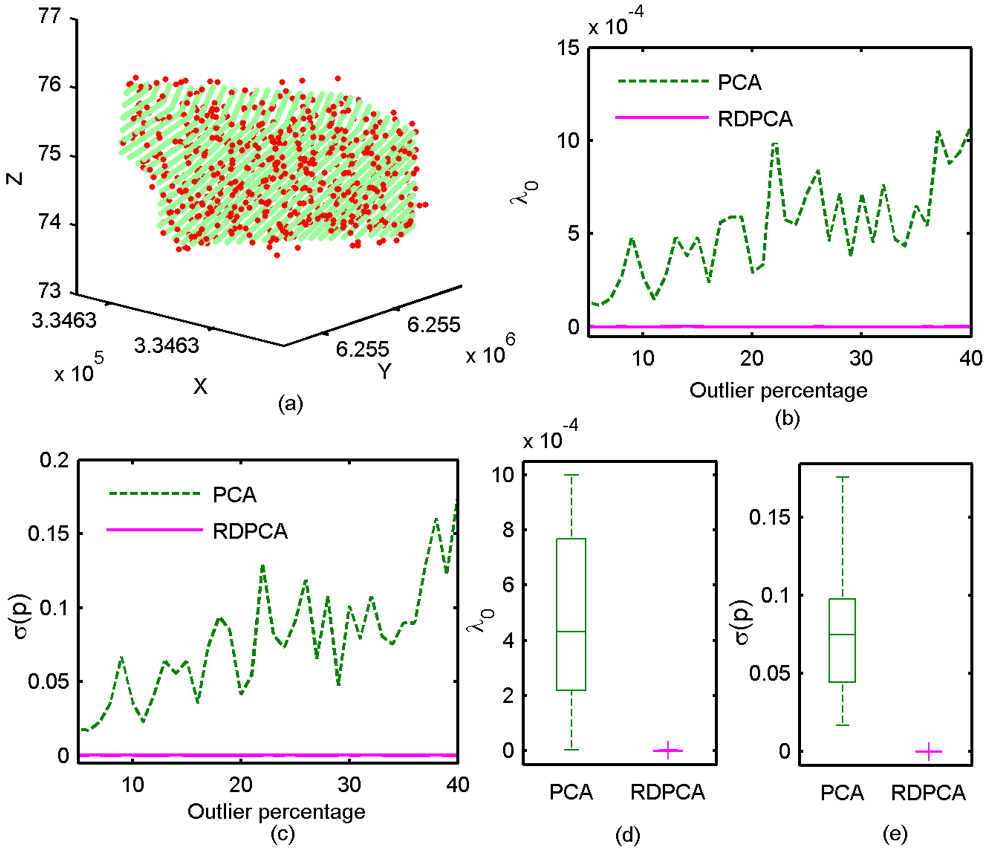


Figure 5.7 (a) Real MLS data with 25% outlier (noise); outlier influence on λ_0 and $\sigma(p)$ of a surface points, PCA (green) and RDPCA (magenta); (b) λ_0 versus outlier percentage, and (c) $\sigma(p)$ versus outlier percentage; (d) PCA and RDPCA boxplots of λ_0 for different percentages (5% to 40%) of outlier contamination, and (e) PCA and RDPCA boxplots of $\sigma(p)$ for different percentages (5% to 40%) of outlier contamination.

5.4 Robust Segmentation for Multiple Planar Surfaces Extraction

This section proposes a segmentation algorithm for multiple planar surface extraction from laser scanning 3D point cloud data.

5.4.1 Algorithm Implementation

The algorithm described in this section can be categorized as a hybrid approach that consists of three steps: classification, region growing and merging. In the first step, the points in the data are classified into edge, corner and surface points. Then in the second step, region growing is performed on the identified surface points. Our method using robust saliency features minimizes both over and/or under segmentation. Over segmentation occurs when more surface regions are detected than actually exist. Under segmentation occurs when more than one true region are combined. We prefer over segmentation in the second step as the problem of over segmentation can be overcome by merging incomplete regions in the third step. The three steps of the segmentation algorithm are now detailed in the following three subsections.

5.4.1.1 Classification

Classification means the separation of the point cloud data into sharp features i.e. edge and corner points, and surface points. If the neighbourhood of a point come from a noise or outlier-free planar surface, then the least eigenvalue λ_0 estimated by the neighbourhood should have a value of zero or near zero. Usually, the points on or near the edge have a neighbourhood from different adjacent feature surfaces, and will have λ_0 values considerably larger than λ_0 values from the surface points. We can consider edges and corner points as the outlying cases compared with the surface points. We follow the general rule for finding outliers i.e. a point is considered as an outlier if:

$$\text{mean}(.) + a \times \text{StD}(.), \tag{5.4}$$

where $a = 1$ or 2 or 3; depending on the data and needed for getting larger λ_0 values. We consider the i^{th} point that has

$$\lambda_0 > \text{mean}(\lambda_0) + 1 \times \text{StD}(\lambda_0) \tag{5.5}$$

to be an edge/corner point. Although, the mean and StD are not robust measures, we wish to see the influence on them of outliers. We could use the median and MAD as robust alternatives to mean and StD respectively, and since they are robust measures, the influence of the outliers on them will be reduced because of their robustness quality. In Eq. (5.5) we want to be sensitive to the influence of outliers and/or sharp features. It is reasonable that the measure for edge and/or corner point detection depends on the data and neighbourhood size, so we cannot recommend using a specific threshold. We can determine the threshold values by using the percentile values of λ_0 for the respective histograms and/or index plots such as for the example shown in Figure 5.9. The rest of the points with smaller λ_0 values are considered as the surface points.

5.4.1.2 Region Growing

To extract planar surfaces, it is reasonable that points in the same region should have similar normal orientation and low bias angle θ between the neighbouring points (Hoppe et al., 1992; Powell et al., 1998; Woo et al., 2002; Mitra and Nguyen, 2003). If we remove the edge and corner points for the multiple planar surfaces, then we can easily find the points in the same planar surface using the region growing approach. It is sufficient to fix a small threshold for the bias angles between the normals of two neighbouring points on a surface. The bias angle between two neighbouring points is defined as:

$$\theta = arccos|\hat{n}_1 \cdot \hat{n}_2|, \tag{5.6}$$

where $\hat{n}_1$ and $\hat{n}_2$ are the two unit normals for the i^{th} point and one of its neighbours. We exclude the edge points from region growing. The region growing starts by searching for a seed point, which has the lowest λ_0 value in the data after eliminating edge points. It is reasonable to consider the i^{th} point as the seed point, which corresponds to the lowest λ_0 value because the i^{th} point and its neighbours should make the most planar surface for the whole data. We consider the least λ_0 because we are using normals for region growing, and each normal is related to the least eigenvalue for a point, and the least eigenvalue measures the variation along the normal to the tangent plane. Removing the edge points can give us an estimate of where the junctions between separating planes are in the data prior to region growing. We assume S is the surface points set and region growing commences from the seed point Sp_i with the lowest λ_0 for the current region R_c. The points whose bias angles are less than an angle threshold θ_{th} are added to R_c and used as the next seed points for R_c and will be removed from S. We fix an appropriate θ_{th} so that we can avoid under segmentation and bias the region growing to over segmentation. If necessary, the problem of over segmentation can be overcome by merging. After getting a complete region, we select seed point Sp_i for the next region from the remaining points in S that has the minimum λ_0. If a region contains greater than or equal to a minimum number R_{min} of points it will be considered as a useful region, otherwise it will be considered as an insignificant region. The same process of region growing will be continued until the surface point set S is empty. The region growing process is summarized in Algorithm 5.2.

Algorithm 5.2: Region Growing for Multiple Planar Surface Extraction

1. Input: surface points set S (excluding edge points), normals, λ_0, θ_{th} and R_{min}.
2. Find initial seed point Sp_i from the S, which has minimum λ_0, and put it into current region R_c, current seed point list S_c, and remove from S.
3. Find k nearest neighbourhood Np_i for each seed point in S_c.
    (a) Calculate θ between Sp_i and its neighbours.
    (b) Find the points in S that have $\theta < \theta_{th}$.
    (c) Put them into R_c and S_c, and remove from S.
4. If size R_c is larger than or equal to R_{min}, insert R_c into the region list R.
5. Repeat Steps 2 to 4 until S is empty.
6. Sort the regions in R.
7. Output: sorted regions list R.

5.4.1.3 Region Merging

This step merges the neighbouring co-planar regions that are the consequence of over segmentation and should belong to the same feature or object surface. It is assumed that merging a specific region with a larger and most appropriate or closest neighbouring region should change the Mean Squared Error (MSE) less than the MSE for merging with any other distant or inappropriate region. The MSE for a region can be defined as in Poppinga et al. (2008):

$$\text{MSE} = \frac{1}{l} \sum_{i=1}^{l} (\hat{n} \cdot p_i + d)^2, \tag{5.7}$$

where l is the size of the sample, $\hat{n}$ is the unit normal, and d is the bias or distance from the origin to the plane for the region. To avoid faulty merging, a threshold is fixed based on knowledge of the data or determined from experiments for similar types of data so that the least Difference in MSE (DMSE) does not exceed a DMSE threshold DMSE_{th}. The merging procedure is summarized in Algorithm 5.3.

Algorithm 5.3: Region Merging for Over Segmented Regions

```
1. Input: sorted regions in $R$ from the Algorithm 5.2, and $\text{DMSE}_{th}$.
2. Find neighbouring regions $R_{ij}$ that have the same edge or border points for each region $R_i \in R$.
3. Calculate the MSE for $R_i$ and $R_i \cup R_{ij}$.
4. Calculate   $\text{DMSE} = |\text{MSE}(R_i) - \text{MSE}(R_i \cup R_{ij})|$, for all $R_{ij}$.   (5.8)
5. Merge $R_{ij}$ with $R_i$ for which $\text{DMSE} \leq \text{DMSE}_{th}$, and remove $R_{ij}$ from $R$.
6. Sort the regions in $R$.
7. Output: sorted regions list $R$ after merging.
```

5.4.2 Experiments

The proposed segmentation (classification, segmentation and merging algorithms) method are evaluated through simulated and real MLS point cloud datasets. We perform the algorithms for edge detection or classification of points, region growing and merging based on the saliency features: normal and λ_0, that are estimated by using PCA, RANSAC and RDPCA. We label the segmentation results as PCA, RANSAC and RDPCA depending on the saliency feature estimation methods used in the proposed segmentation process.

5.4.2.1 Simulated Data

Dataset 5.1: Unit cube dataset

We simulate a unit cube as shown in Figure 5.8a that has six surfaces consisting of 6,000 points, for which every surface is generated as an individual planar surface in 3D. We perform PCA, RANSAC and RDPCA to get λ_0 and the normal for every point with a neighbourhood size $k = 30$. Using Eq. (5.5), we determine the edge points for the cube as shown in magenta in Figures 5.8(b, c and d). Results show that PCA (Figure 5.8b) and RANSAC (Figure 5.8c) identify many surface points wrongly as edge points. However, Figure 5.8d shows RDPCA identifies edge points more accurately. We perform Algorithm 5.2 with $\theta_{th} = 2°$ and $R_{min} = 10$ to extract the six surfaces. The results in Figures 5.8(e, f and g) show that the algorithm properly segments all the planar surfaces. The segmentation results shown in Figure 5.8g that are based on RDPCA normals and λ_0 values are better than PCA (Figure 5.8e) and RANSAC (Figure 5.8f). Even RANSAC produces several over segments as shown in the ellipses in Figure 5.8f.

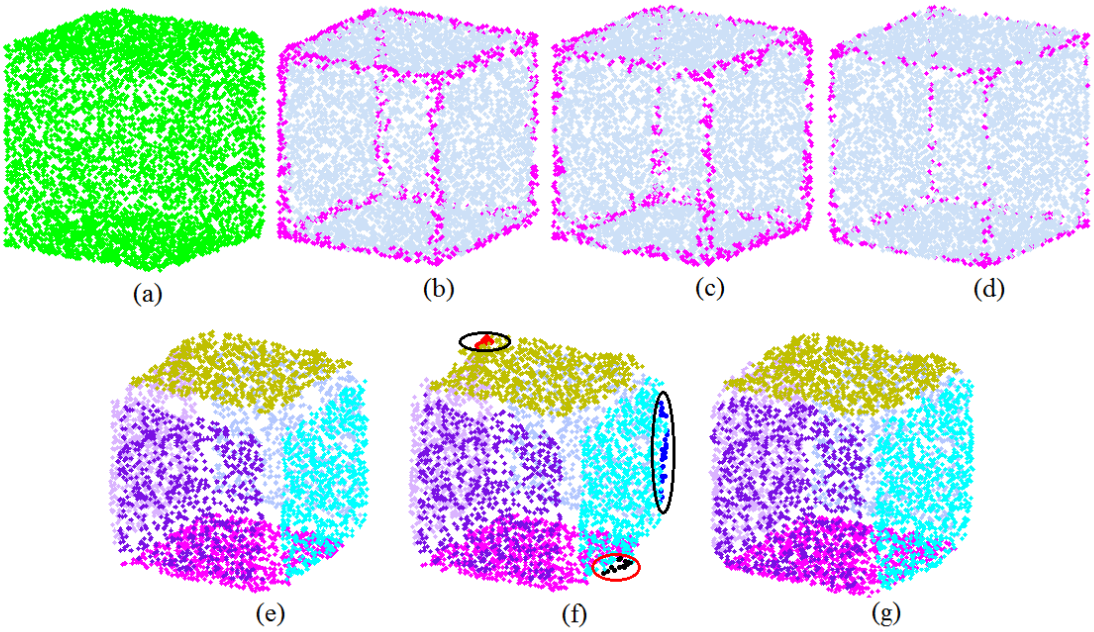


Figure 5.8 (a) Simulated unit cube. Edge (in magenta) detection: (b) PCA, (c) RANSAC, (d) RDPCA. Segmentation results: (e) PCA, (f) RANSAC, over segmented regions are shown by ellipses, and (g) RDPCA.

5.4.2.2 Real MLS Data

Dataset 5.2: Road-kerb-footpath-fence dataset

We consider a small point cloud dataset shown in Figure 5.9a that consists of 2021 points acquired by using a moving vehicle-based mobile mapping system. It has four planar surfaces that are parts of a road pavement, kerb, footpath and fence. We name this the 'road-kerb-foothpath-fence' dataset. We find the edge points of the data. Based on the empirical study, we set the neighbourhood size $k = 30$ and perform RDPCA Algorithm 5.1 to get robust saliency features. We also perform PCA and RANSAC for comparison. In Figure 5.9b, boxplots show that the λ_0 values from RDPCA are much more robust than the λ_0 values from PCA and RANSAC. Figures 5.9(c, d and e) and Figures 5.9(f, g and h) are the histograms and index plots of λ_0 values for PCA, RANSAC and RDPCA respectively. We use Eq. (5.5) to draw the cut-off lines in Figures 5.9(f, g and h) for identifying the edge points (magenta) in Figures 5.10(a, b and c). Results show that PCA (Figure 5.10a) and RANSAC (Figure 5.10b) cannot identify edge points properly with many surface points wrongly identified as edge points. However Figure 5.10c shows that RDPCA identifies edge points more accurately.

Figures 5.10(d,e and f) show the visual orientation of the normals for the detected edge points. We see PCA normals for the edge points in the blue marked box in Figure 5.10d, which are smooth and too similar to differentiate them according to their position and to know from which planar surface the points come from. The normals for PCA look smoothed out because the neighbourhood for those points have many common points from two different surfaces. Since RANSAC is a robust method and saliency features are based on a consensus set within a local neighbourhood, it gives better results than PCA but Figure 5.10e shows differentiation is not satisfactory. In Figure 5.10f, normals from RDPCA are clearly separated into two directions that indicate from which points and their neighbours they come from, and represent the respective surface. Results for RDPCA also prove that the Algorithm 4.1 (Chapter 4) finds the maximum consistent set (MCS) accurately and produces robust saliency features for identifying edge points and proper segmentation.

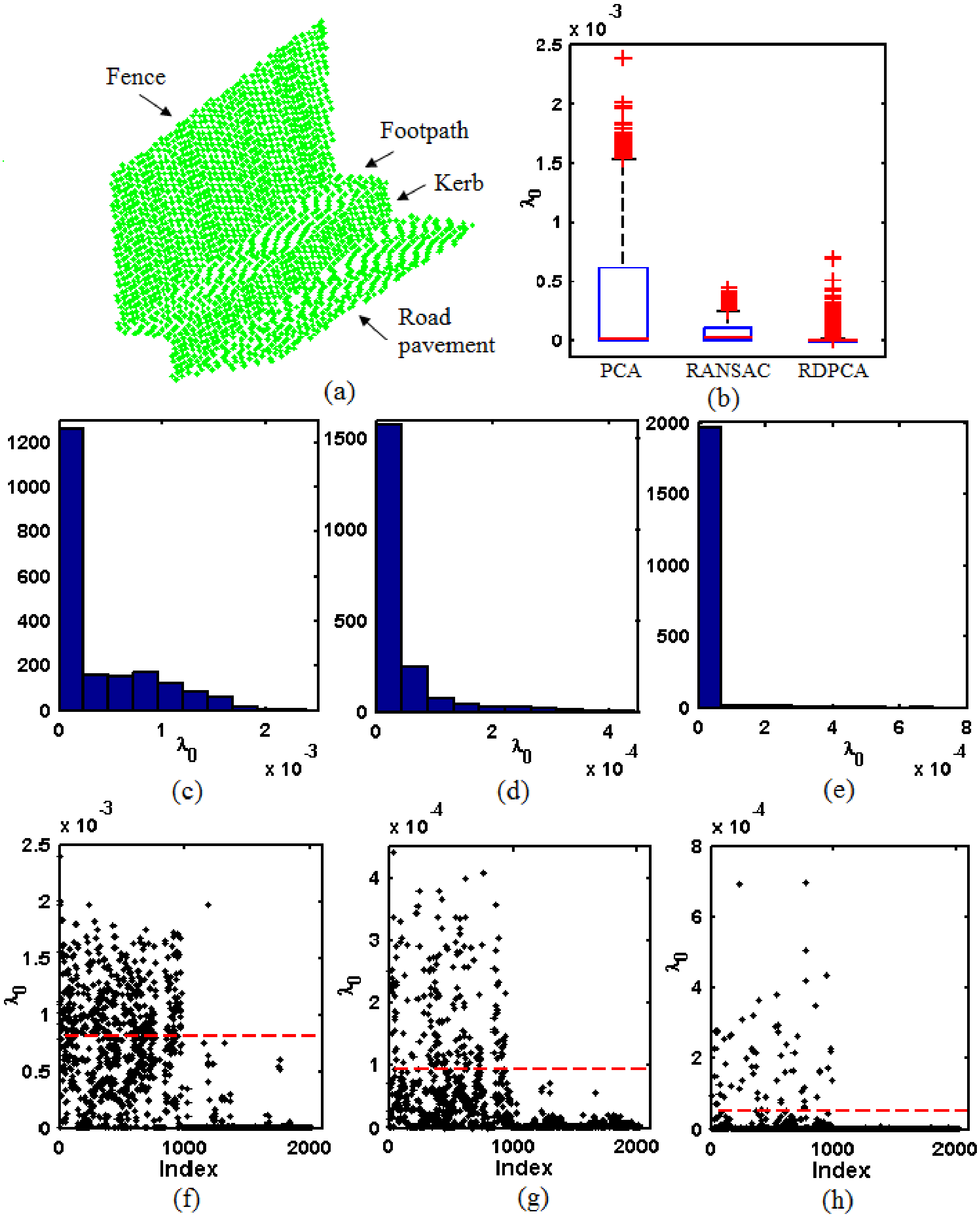


Figure 5.9 (a) Real MLS point cloud data, (b) boxplots of λ_0 values for PCA, RANSAC and RDPCA. Histograms of λ_0 values: (c) PCA, (d) RANSAC, and (e) RDPCA. Index plots of λ_0 values (red lines indicate cut-off lines): (f) PCA, (g) RANSAC, and (h) RDPCA.

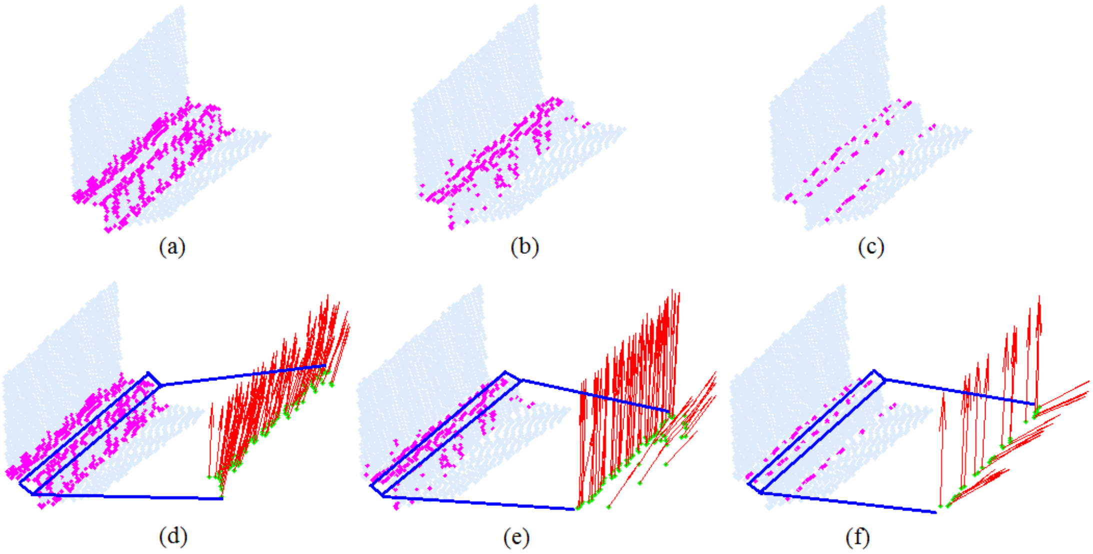


Figure 5.10 Identification of edge points (magenta in colour): (a) PCA, (b) RANSAC, and (c) RDPCA; normals (red quiver) for the selected edge points: (d) PCA, (e) RANSAC, and (f) RDPCA.

We now use the proposed Algorithm 5.2 to segment the road-kerb-foothpath-fence dataset shown in Figure 5.9a. Based on empirical investigation on similar real MLS data, we set the parameters $\theta_{th} = 5°$ and $R_{min} = 10$ to perform segmentation. Figure 5.11d shows that using RDPCA based robust saliency features the segmentation algorithm properly segments all the planar surfaces, whereas PCA based results shown in Figure 5.11a are over segmented with three segments for the kerb and fence surfaces. Over segmentation i.e. a single surface erroneously split into multiple surfaces, occurs because the calculated PCA normals have equal weight for all the regular and outlying points in the respective neighbourhood and hence may be influenced by outliers. On the other hand, RDPCA normals are based on the MCS, which can avoid the influence of the outlying cases in the neighbourhood. In the case of RANSAC, Figure 5.11c shows many points in the footpath surface that disappear from the segmentation results because they are wrongly identified as edge points and were removed before region growing.

Using Algorithm 5.3 and setting $\text{DMSE}_{th} = 1.0e^{-04}$, we perform the merging task for the over segmented regions for PCA in Figure 5.11a. After merging the regions, the PCA based segmentation results are shown in Figure 5.11b. We see all the regions in the fence and kerb surfaces shown in Figure 5.11a are accurately merged in Figure 5.11b. However the RDPCA based segmentation results in Figure 5.11d do not need any merging and are significantly better than the results in Figures 5.11b for PCA and 5.11c for RANSAC.

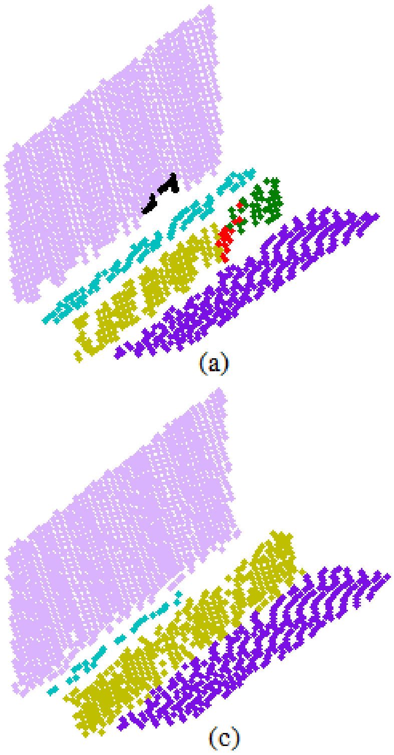


Figure 5.11 Region growing and segmentation: (a) PCA region growing, (b) PCA merging and segmentation, and (c) RANSAC segmentation, and (d) RDPCA segmentation.

5.5 Robust Segmentation for Laser Scanning Point Cloud Data

Although planar surfaces are frequently seen in the real world e.g. along road corridors and on industrial sites, the segmentation strategy that uses planes as the only available model will result in extreme over segmentation for curved objects e.g. pole, sign post and different traffic furniture, that are common in LiDAR and MLS data. In essence a curved surface is likely to be segmented as a number of planar patches. In this section, we propose a robust segmentation algorithm for both planar and non-planar or curved complex object surfaces.

5.5.1 Algorithm Formulation and Implementation

The basic ideas of the region growing approach are used in our segmentation algorithm. The region growing approach continues to grow a region around seed points depend on some pre-assigned criteria. The segmentation algorithm introduced in this section uses four basic and consecutive tasks shown in Figure 5.12.

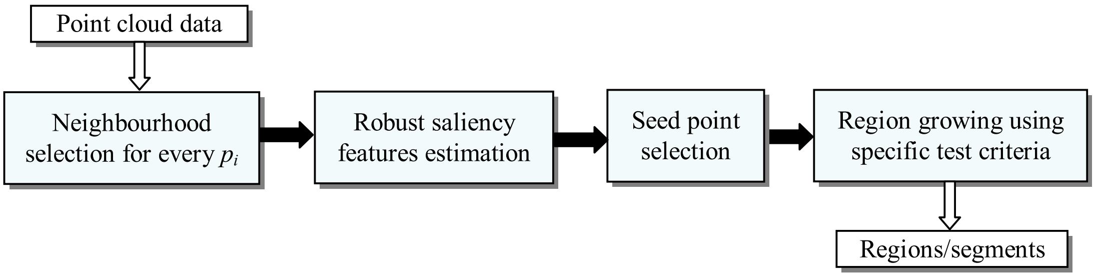

Point cloud data
Neighbourhood selection for every p_i
Robust saliency features estimation
Seed point selection
Region growing using specific test criteria
Regions/segments

Figure 5.12 Robust segmentation process.

Task 1. ***Neighbourhood selection*:** It is known that proper neighbourhood selection for an interest point is an important task for accurate normal and curvature estimation. Three methods are common for neighbourhood search: (i) fixed distance neighbourhood, (ii) neighbourhood within a voxel, and (iii) k Nearest Neighbourhood (kNN). For the first two, the numbers of points in a neighbourhood are different due to uneven sampling. We choose the kNN searching technique because it can deal well with the data that has an uneven point density and can adapt the area of interest w.r.t. the data density. We use the K-D tree based kNN search algorithm to get k points in the local neighbourhood Np_i of p_i, mainly because kNN search can produce normals and curvatures with an equal number of points support. It is also better for avoiding the uneven point density that is a common event in MLS data, because of the movement of the data acquisition vehicle and sensors. Neighbourhood size is a major concern for reliable local saliency feature estimation. Hoffman and Jain (1987) pointed out that a smaller neighbourhood gives normals more susceptible to noise. Many authors suggest using a larger k for better normals (Hoffman and Jain, 1987; Besl and Jain, 1988; Rabbani et al., 2006). Yang and Feng (2005) pointed out that using a large number of points can adversely affect the local characteristics of the normal vector but the local geometry is better represented by a smaller number of points. Since the quality of the surface normals depends heavily on the structure of the surface geometry it is better to investigate the problem of fixing the size of k empirically rather than analytically. A neighbourhood size should be carefully choosen so that neighbours in the local neighbourhood become co-planar. It can be done by related real data experimentation and/or simulation.

Task 2. ***Robust saliency feature estimation*****:** We can estimate normals and curvatures for all the points based on their local neighbourhood in the data using the PCA approach. But normals and curvatures from PCA are not robust, so we use RDPCA (Algorithm 5.1) as an alternative to the PCA algorithm for getting robust normals and curvatures, or any necessary local saliency features.

Task 3. ***Seed point selection*****:** Region growing is started with a seed point p_i that has the least curvature value, because it is reasonable that region growing will be more successful for the area where the surface is smoother and as a consequence the surface variation is lower. We use surface variation $\sigma(p)$ as the curvature because it measures the local properties (variations) of a smooth planar or curved surface in every direction (Pauly et al., 2002).

Task 4. ***Region growing*****:** From the selected seed point a region grows gradually based on the spatial connectivity among the points. We define two points that are spatially connected or close or in the same region if they follow some proximity and/or coherence criteria. We fix three distance measurements as the test criteria to get two points that are sufficiently close to consider them in the same homogeneous region. The measures are: (i) point to point Euclidean Distance (ED), (ii) point to plane Orthogonal Distance (OD), and (iii) angular distance between the two points. The three measures are sketched in Figure 5.13 and can be calculated as follows.

We find a k nearest neighbourhood Np_i for the i^{th} seed point p_i. We calculate the ED between a pair of points as:

$$\mathrm{ED}_{ij} = ||p_i - p_j||, \tag{5.9}$$

where p_i is the seed point and p_j is one of its neighbours in Np_i. Since the data density may be uneven, we consider two points that are close to be in the same region if they are as close as the majority of the points in the neighbourhood. We decide p_j is sufficiently close to p_i if:

$$\mathrm{ED}_{ij} < \mathrm{ED}_{th} = \mathrm{median}\{\mathrm{ED}_{ij}\}, \tag{5.10}$$

where ED_{th} is the ED threshold, and $\{\mathrm{ED}_{ij}\}$ is the set of all ED_{ij}s between the seed point and its local neighbours.

We compute ODs for all the neighbours of p_i. The OD for the i^{th} point p_i to its best-fit-plane generated by its neighbours can be defined as:

$$\mathrm{OD}_i = (p_i - \bar{p})^T \cdot \hat{n}, \tag{5.11}$$

where $\bar{p}$ and $\hat{n}$ are the mean vector and the unit normal of the best-fit-plane, respectively. To reduce outlier effects and to make the surface smooth, we define the general rule of unusual (outlying) point identification as the OD threshold OD_{th} defined as:

$$\mathrm{OD}_{th} = \mathrm{mean}\{\mathrm{OD}(Np_i)\} + a \times \mathrm{StD}\{\mathrm{OD}(Np_i)\}, \tag{5.12}$$

where StD is the standard deviation, and $a = 1$, or 2 or 3. To make OD_{th} robust, we use the median and Median Absolute Deviation (MAD) instead of mean and StD in Eq. (5.12). Hence, we consider the i^{th} point as a co-planar surface point if:

$$\mathrm{OD}_i < \mathrm{OD}_{th} = \mathrm{median}\{\mathrm{OD}(Np_i)\} + 2 \times \mathrm{MAD}\{\mathrm{OD}(Np_i)\}, \tag{5.13}$$

where $\{\mathrm{OD}(Np_i)\}$ is the set of OD_is for all the points in the neighbourhood of p_i, $a = 2$ and

$$\mathrm{MAD} = b \cdot \underset{i}{\mathrm{median}} |p_i - \underset{j}{\mathrm{median}}(p_j)|, \tag{5.14}$$

where $b = 1.4826$ to make the estimator consistent (Rousseeuw and Croux, 1993).

We also consider the angular distance between two points if they are spatially close. The dihedral angle θ sometimes called the bias angle between two points is used to measure the angular distance that is defined in Eq. (5.6). Two spatially close points will be co-surface points and on the same smooth surface if θ is less than a user defined threshold θ_{th}.

Therefore, region growing starts with an initial seed point p_i that has the least curvature value in Eq. (5.1) and finds its local neighbours for a current region R_c. A neighbour p_j will be added to R_c and the current seed point list S_c and removed from P, if the following three conditions are satisfied:

$$\left.\begin{array}{ll} (i) & \mathrm{OD}_j < \mathrm{OD}_{th}, \\ (ii) & \mathrm{ED}_{ij} < \mathrm{ED}_{th}, \text{and} \\ (iii) & \theta_{ij} < \theta_{th}. \end{array}\right\} \tag{5.15}$$

R_c will continue to grow until no more seed points are available in S_c. If the size of R_c is less than a minimum number R_{min} of points then the region will be considered as an insignificant region and be ignored. After growing a complete region, we select the next seed point for the next region from the remaining points in P that has the least $\sigma(p)$. This region growing process will continue until P is empty. The robust segmentation process is summarized in Algorithm 5.4.

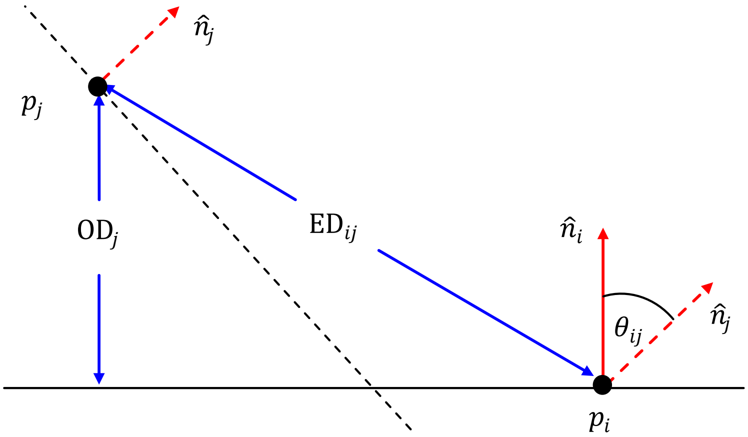


Figure 5.13 Distances between p_i and p_j used in the segmentation algorithm: Euclidean distance ED_{ij}, orthogonal distance OD_j, and angular distance θ_{ij}.

Algorithm 5.4: Robust Segmentation

```
Point Cloud: $P$
Built kd-tree
    $T \leftarrow kd - tree(P)$
Find $k$ Nearest Neighbours for each point in $P$
    [$Np$, ED]=$k$NNsearch $(T, P, k)$
Robust saliency features estimation (Algorithm 5.1)
    $[N, K] = normal - curvature(P, Np, h, \epsilon, P_r)$
Region Growing
Input:
    $P$: point cloud
    $Np$: cached nearest neighbours for each point in $P$
    ED: cached Euclidean Distance for each point to its neighbours
    $N$: list of normals ($\hat{n}$) for each point in $P$
    $K$: list of curvatures for each point in $P$
    $\theta_{th}$: angle threshold
    $R_{min}$: minimum region size
Initialize: list of regions $R \leftarrow \phi$, points not in any region $R' \leftarrow \phi$
while $P$ is not empty do
    $R_c \leftarrow \phi$, $S_c \leftarrow \phi$
    Select initial seed point $p_i$ from $P$ with the least curvature in $K$
    $R_c \xleftarrow{insert} p_i$, $S_c \xleftarrow{insert} p_i$, and $P \xrightarrow{remove} p_i$
    for each point in $S_c$ do
        Select nearest neighbours of the $i^{th}$ seed point from $Np$
        Find ED$_{ij}$s for the $i^{th}$ seed point and its neighbours from ED
        Calculate OD$_j$s for all the neighbours of the $i^{th}$ seed point using $\hat{n}_i$
        ED$_{th}$ = median{ED$_{ij}$}
        OD$_{th}$ = median{OD$_j$} + 2 × MAD{OD$_j$}
        Find the points from $Np_i$, whose ED$_{ij}$ < ED$_{th}$ and OD$_j$ < OD$_{th}$ and put them in a list $L$
        for $j$=1 to size($L$) do
            if $p_j$ is in $P$ then
                if $\theta_{ij} < \theta_{th}$ then
                    $R_c \xleftarrow{insert} p_j$, $S_c \xleftarrow{insert} p_j$, and $P \xrightarrow{remove} p_j$
                end if
            end if
        end for
        if$R_c \geq R_{min}$ then
            $R \xleftarrow{insert} R_c$
        else
            $R' \xleftarrow{insert} R_c$
        end if
    end for
end while
Output: list of regions $R$
```

5.5.2 Advantages of the Proposed Robust Segmentation Method over the Existing Methods

The region growing based segmentation algorithms available in the literature that use similar attributes, conditions and/or saliency features, as used in our algorithm have following limitations.

(i) Many methods that use different curvature (Besl and Jain, 1988) and or high level derivatives can handle non-planar surfaces but often this leads to over segmentation (Rabbani et al., 2006).

(ii) Segmentation methods that use residuals such as OD as the smoothness constraint and curvatures (Rabbani et al., 2006) can suffer the problems of equal residuals and normals i.e. insignificant bias angles, from non-robust (LS or PCA) methods that may cause under segmentation.

(iii) Some methods use a large number of parameters (Jiang et al., 1996). Although the method of Jiang et al. (1996) works better than the curvature based approach e.g. Besl and Jain (1988) and works for industrial scenes, it has limited application to unstructured point cloud data (Rabbani et al., 2006).

(iv) Some methods that originated for segmentation into planar surfaces use OD, ED and MSE (Xiao et al., 2013), some use ED and MSE (Poppinga et al., 2008), or some other combination of OD, ED, MSE and θ. However, these methods have some limitations when segmenting curved or non-planar surfaces.

(v) Non-robust methods that use OD, normals and curvatures can handle curved objects but the unreliable estimates from outlier and/or noise contaminated point clouds leads to high rates of over and/or under segmentation.

The segmentation algorithm developed in this section uses three conditions at a time in Eq. (5.15) and a robust approach (RDPCA) to get robust saliency features: normal and curvature that reduces over and/or under segmentation and produces significantly better, more accurate and robust results.

In the literature, some authors use only θ, some use θ and curvature, some use OD and ED, and some suggest θ and ED for region growing in their segmentation algorithms (Rabbani et al., 2006; Klasing et al., 2009; Xiao et al., 2013). We argue that all the three criteria (attributes): OD, ED and θ are necessary for better results in region growing and segmentation process. We illustrate this requirement by taking two pieces of planar surface from a real MLS dataset shown in Figure 5.14a that have same orientation. We add 20% Gaussian noise with mean (0.0, 0.0, 0.0) and StD (0.1, 0.1, 0.1) to make the surface noisy as shown in Figure 5.14b. The noise points shown in red are created in a way to make them off-surface points that can be treated as outliers (Sotoodeh, 2006). We perform RDPCA with $k = 25$, and $\theta = 5°$. For the results in Figure 5.14c we use only θ as the region growing criterion but it fails to separate the planes because the surfaces have similar orientations/normals. The result of Figure 5.14e shows that the use of θ and ED is necessary to separate the surfaces, and in Figure 5.14d, it can be seen that θ and OD are necessary to remove the outliers but is unable to separate the two surfaces. Finally, we use all three conditions θ, OD and ED. The results in Figure 5.14f demonstrate the necessity of using all three conditions in Eq. (5.15) for proper segmentation and to remove the noise effects. The final segmentation results are noise free and the two surfaces are properly extracted or separated.

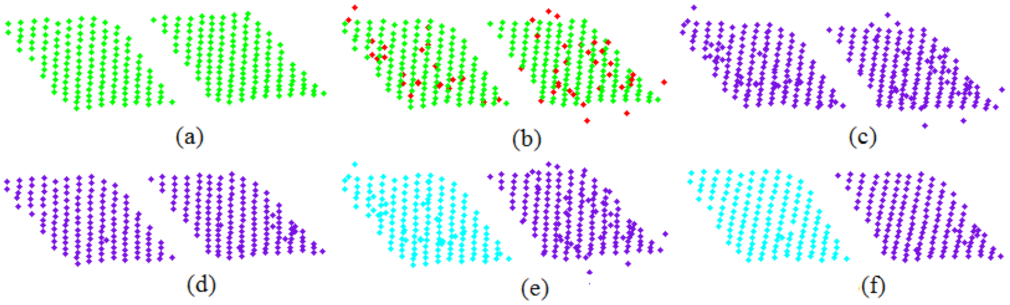


Figure 5.14 Use of three distance measures in region growing: (a) two parallel planar surface data, (b) outlier (red points) contaminated data. Robust segmentations based on: (c) θ, (d) θ and OD, (e) θ and ED, and (f) θ, OD and ED.

5.5.3 Experiments and Evaluation

The new segmentation method of Algorithm 5.4 is demonstrated and evaluated in this section through experiments on two simulated and two real laser scanning point cloud datasets. Our proposed method is based on robust saliency features i.e. normal and curvature $\sigma(p)$ from RDPCA. To see the necessity of using robust normal and curvature in the proposed algorithm, we also perform the new segmentation algorithm based on the saliency features that are estimated by using PCA and RANSAC. Based on the saliency feature estimation method used in our algorithm, we label the segmentation results as: PCA, RANSAC and RDPCA.

Accuracy measurement: We calculate the well-known performance measures: (i) recall (r, surface segmentation rate), (ii) precision (p, correctness of the segmented surface), and (iii) F-score (F, overall accuracy) to measure the accuracy of the segmentation results on real data. The measures are defined in Fawcett (2006), and Li et al. (2012) as:

$$\mathrm{r} = \frac{\text{number of PS}}{\text{number of PS} + \text{number of US}} \times 100, \tag{5.16}$$

$$\mathrm{p} = \frac{\text{number of PS}}{\text{number of PS} + \text{number of OS}} \times 100, \tag{5.17}$$

$$\mathrm{F} = 2 \times \frac{\mathrm{r} \times \mathrm{p}}{\mathrm{r} + \mathrm{p}}, \tag{5.18}$$

where PS = Proper Segment, US = Under Segment, and OS = Over Segment. A Proper Segment is identified as a true segment from manually determined ground truth i.e. one segment describes a single feature such as the wall of a house that is one planar surface. An Over Segment is where one true segment is broken into two or more separate segments, and an Under Segment is where more than one true segments are wrongly grouped together as one segment.

5.5.3.1 Simulated Data

Dataset 5.3: Stair case dataset

The simulated data shown in Figure 5.15a used here has been created as a stair case of eight planar surfaces consisting of 19,500 points. We call this the 'stair case' dataset. To perform the segmentation, we set the required parameters: $k = 30$, angle threshold $\theta_{th} = 2°$, and minimum region size $R_{min} = 10$. The segmentation results are shown in Figure 5.15(b, c and d). The

PCA results of Figure 5.15b show that many points in edges are missing because the orientations of the normals of the points near or on the edges do not represent a specific plane properly as we observed in Figure 5.10d. As a result, at the time of region growing the points make many small individual regions of size less than R_{min} and are ignored. A similar type of problem also occurs for RANSAC segmentation shown in Figure 5.15c with many under grown segments seen in the areas near edges. That means both PCA and RANSAC are affected by the over segmentation problem. Using the normals and curvatures from the proposed RDPCA, the same segmentation algorithm performs significantly better. Figure 5.15d shows that the RDPCA segmentation results are accurate, and extract eight planar surfaces without any over or under segments.

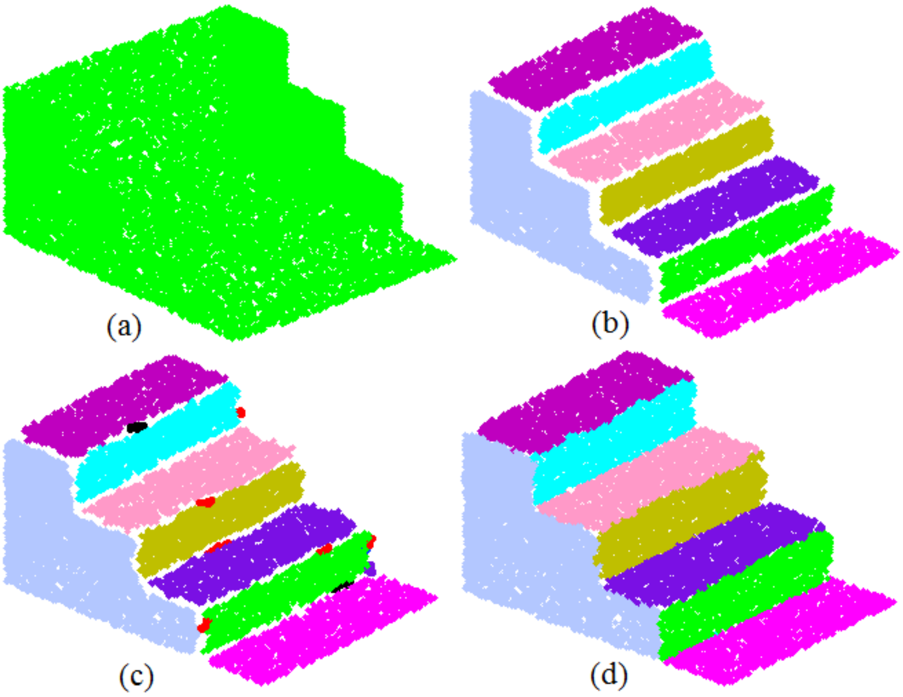


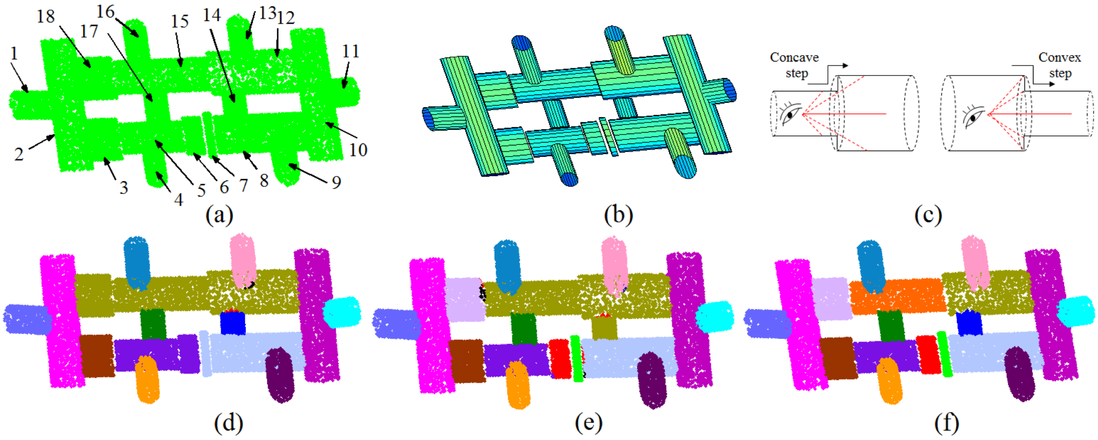
Figure 5.15 (a) Stair case dataset; segmentation results: (b) PCA, (c) RANSAC, and (d) RDPCA.

Dataset 5.4: Cylinder dataset

To demonstrate the segmentation algorithm for non-planar objects, we simulate a 3D dataset of 27,100 points that contains a set of 18 cylindrical surfaces attached in different positions and orientations. The dataset shown in Figure 5.16a for the model in Figure 5.16b is labeled as the 'cylinder' dataset. The cylinders have various radii between 0.18m and 0.3m, and various lengths between 0.12m and

1.8m. For example, radius and length of the 2nd cylinder are 0.3m and 1.8m, respectively. The cylinders are joined in such a way that many concave and convex steps as shown in Figure 5.16c are generated between pairs of consecutive cylinders; they are joined horizontally as well as vertically, which are common in industrial structures.

Using PCA, RANSAC and RDPCA with a neighbourhood size $k = 30$, we estimate necessary normals and curvatures. The segmentation algorithm is performed with angle threshold $\theta_{th} = 13°$ and minimum region size $R_{min} = 10$. Segmentation results are shown in Figures 5.16(d, e and f). Results show that over and under segmentation appear in Figures 5.16(d and e) for PCA and RANSAC, respectively. PCA wrongly groups seven cylinders in three places for Cylinders: 5 and 6; 7 and 8; 12, 15 and 18; and RANSAC joins 3 cylinders i.e. Cylinders 12, 14, and 15 into one segment. PCA has two over segments in Cylinders 13 and 14, and RANSAC has six over segments. However, the proposed RDPCA algorithm extracts all 18 cylinders without any over and under segmentation as shown in Figure 5.16f.



(a) (b) (c)

(d) (e) (f)

Figure 5.16 (a) Simulated cylinder dataset, (b) object model, (c) concave and convex steps. Segmentation results: (d) PCA, (e) RANSAC, and (f) RDPCA.

5.5.3.2 Real MLS Data

In this section, we demonstrate and evaluate segmentation Algorithm 5.4 on real laser scanning datasets. We also compare our algorithm with two recently proposed segmentation algorithms: Rabbani et al. (2006) and Xiao et al. (2013).

Dataset 5.5: House dataset

The first dataset shown in Figure 5.17a, consists of 7,517 points forming part of a road side building of 11 planar surfaces, acquired using a moving vehicle based laser scanner. We call this the 'house' dataset. We use Algorithm 5.4 to segment the data and fix the parameters: $k = 30$, $\theta_{th} = 5°$, and $R_{min} = 10$. We demonstrate the same segmentation algorithm but use the saliency features: normals and curvatures calculated by PCA, RANSAC and RDPCA. In Figure 5.17b, the PCA based segmentation results are very poor, produce a total of 11 segments in which eight segments are properly segmented, but there are one under segment and one over segment in Surface 1. Many points are missing in Surface 9 near Surface 8, because missing points are in under-grown regions i.e. those having less than 10 points. In Figure 5.17c, the RANSAC based algorithm produces 10 planes properly but also produces one over segmented region in Surface 10. However, the RDPCA based algorithm accurately segments all 11 planar surfaces without any over or under segment as shown in Figure 5.17d.

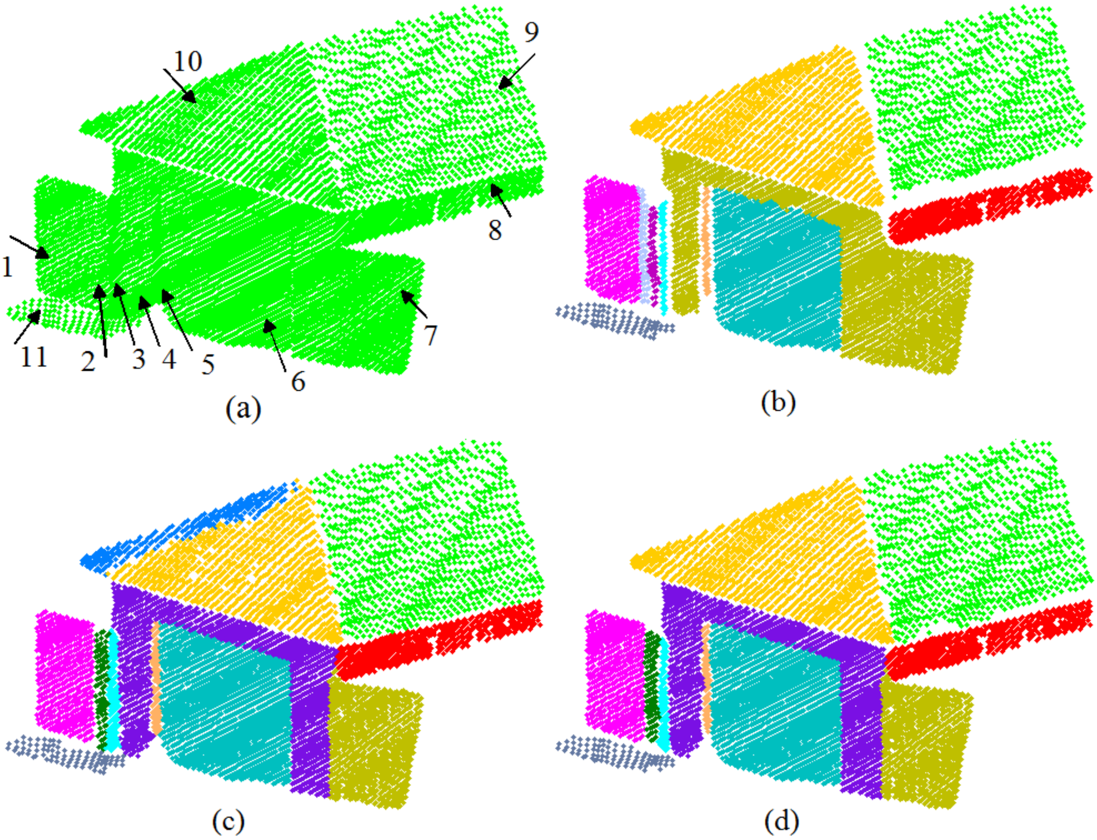
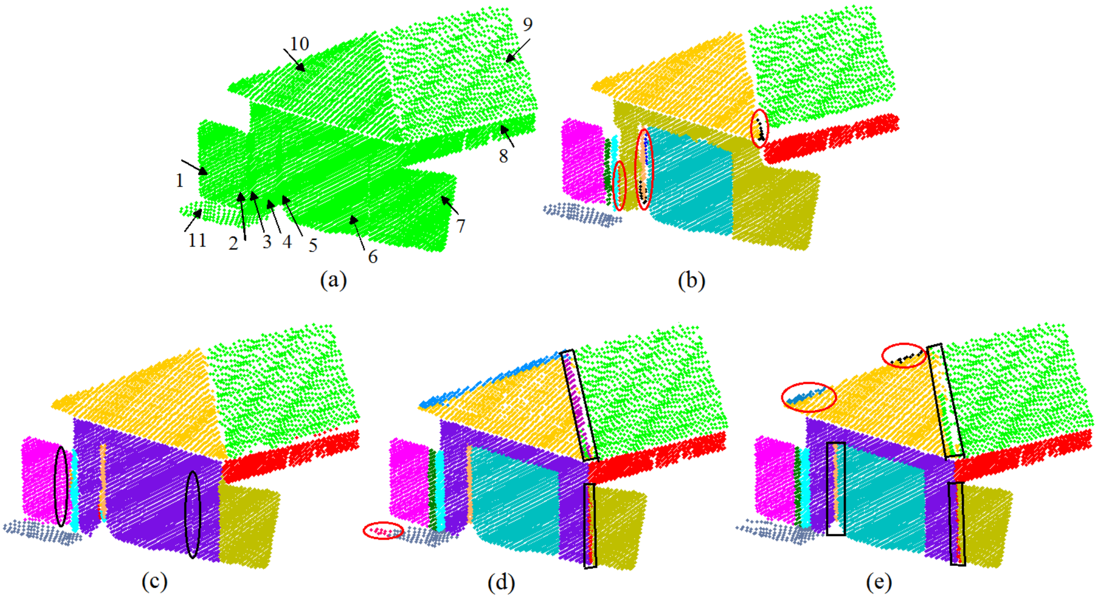


Figure 5.17 (a) House dataset. Segmentation results: (b) PCA, (c) RANSAC, and (d) RDPCA.

Comparison with existing methods: The performance of the proposed algorithm is compared with two existing methods. We implement the methods of Rabbani et al. (2006) and Xiao et al. (2013) introduced for segmentation of point cloud data. We consider these algorithms because: (i) they have been recently proposed, (ii) both are region growing based, and (iii) both use local saliency features. Rabbani et al. (2006) defined a residual that approximates curvature and uses this residual as a smoothness constraint. This algorithm starts region growing from the point with the minimum residual, considers it as the seed point, and adds the nearest neighbours of the seed point into the current region if the angle between the seed point and each neighbour is less than a pre-defined threshold. The region growing continues until all the points in the data are considered. The reader is referred to Rabbani et al. (2006) for more details about the algorithm. We label the Region Growing method of Rabbani et al. (2006) as RGR. We perform the algorithm using PCA and RDPCA based saliency features with $k = 30$, $\theta_{th} = 5^\circ$. The results for the PCA based algorithm (i.e. RGR) in Figure 5.18b show that Surfaces 4 and 7 are wrongly grouped. This is because the non-robust normals from PCA smoothed the edge points and compromised the region growing. The several over segmented results are seen in the red ellipses. RDPCA based RGR (i.e. saliency features are calculated by RDPCA), which is labelled as RGR-RDPCA produces results in Figure 5.18c. Surfaces 1 and 2, and Surfaces 4 and 6 are combined together as indicated by the black ellipses. Although the orientations of the respective surfaces are the same, they are significantly distant w.r.t. each other that they could not be separated without using ED. But ED is not used in RGR.

In Xiao et al. (2013)'s algorithm, the authors fitted a plane to each point of the data and its k nearest neighbours w.r.t. Mean Squared Error defined as: MSE $= \frac{1}{k}\lambda_0$, where λ_0 is the least eigenvalue. Although Xiao et al. (2013) used an Octree based search algorithm, to be consistent with the results in our algorithm, we use a K-D tree based kNN search algorithm in this chapter. The algorithm starts by determining a seed point that has minimum plane fitting error (MSE). Region growing starts with the seed point and its neighbours. A new point in the data is added to the current region if: (i) the distance from the new point to the optimal plane fitted to the current region along with a new point is smaller than a pre-assigned distance threshold, and (ii) the plane fitting error of the current region along with the new point is less than a MSE threshold MSE$_{th}$. The region growing process continues until all the points have been processed. The reader is referred to Xiao et al. (2013) for more details about the algorithm. We perform the algorithm using PCA based saliency features. We label the Region Growing method of Xiao et al. (2013) as RGX. We set the required parameters: Euclidean distance threshold ED$_{th}$ = 0.2, MSE$_{th}$ = 0.01 and point to plane distance threshold OD$_{th}$ = 0.03 as advised in their paper. In Figure 5.18d, we see two examples of over segmentation in Surface 10 and one over segment in Surface 11 as signified by the red ellipse. In the black rectangles, some points from Surfaces 4 and 10 are wrongly included with Surfaces 8 and 9 respectively, because of the limitation of the MSE threshold. Setting an inappropriate MSE may include points from different surfaces wrongly into the surfaces i.e. having the same orientation that are currently being considered for region growing. To see the effect of OD$_{th}$, now we change OD$_{th}$ = 0.03 to 0.05, keeping the other two thresholds the same and run the algorithm again, with the results shown in Figure 5.18e. Several points from Surfaces 5, 4, 9 and 10 are wrongly grouped with Surfaces 4, 8 10 and 9, respectively. These are shown in black rectangles. In the red ellipses, we see some over segmented regions in Surface 10. Table 5.1 contains a summary of the results from the RDPCA algorithm and the algorithms of RGR and RGX.



(a)

(b)

(c)

(d)

(e)

Figure 5.18 (a) House dataset. Segmentation results: (b) RGR, (c) RGR-RDPCA, (d) RGX, OD$_{th}$ = 0.03, and (e) RGX, OD$_{th}$ = 0.05.

To evaluate the performance of the methods of RGR and RGX for noisy data, we add 25% Gaussian noise with mean $= 0.0$ and StD $= 0.3$ for all three directions (x, y, z) to the points in Figure 5.18a. Figure 5.19a shows the original data as green points with the noisy data shown as red points. The algorithms RGR, RGX and RDPCA are run again with the same parameters as used for Figures 5.18(b and d) and 5.17d respectively. The results for RGR shown in Figure 5.19b produces seven PS with five OS and one US. The results for RGX shown in Figure 5.19c produces five PS with three OS. For Figure 5.19c, along with several over segments, a major error is many points from Surfaces 3, 4, 4, 9 and 10 are wrongly included in Surfaces 1, 8, 10, 10 and 9 respectively. However the proposed RDPCA algorithm properly extracts all 11 surfaces without any OS and US. The results in Table 5.1 show the accuracy rate (F) for RGR, RGX and the RDPCA are 70.00%, 76.92% and 100%, respectively. Results in Table 5.1 reveal that in the presence of noise, both RGR and RGX produce worse results and less accuracy compared with the dataset without noise.

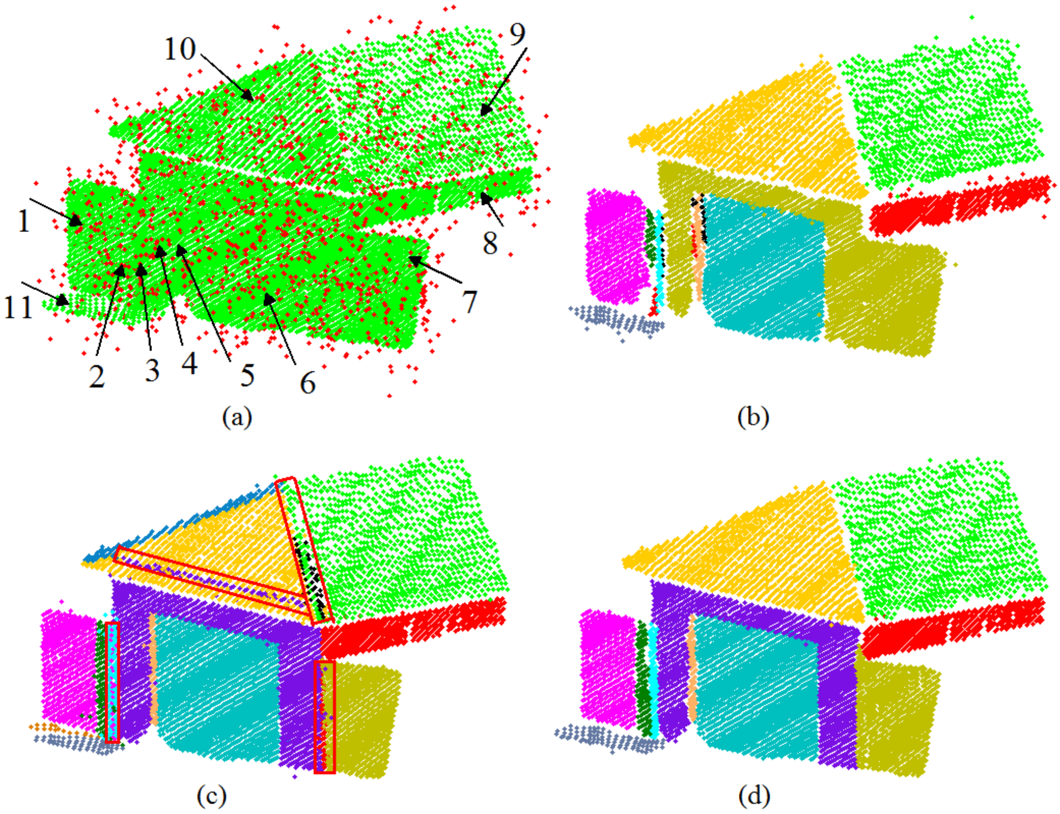


Figure 5.19 (a) House dataset with 25% noise (red points) added. Segmentation results: (b) RGR, (c) RGX, and (d) RDPCA.

Table 5.1 Comparison of segmentation performance with existing methods.

Data Name	Methods	TS	PS	OS	US	r	p	F
House dataset	RGR (Fig 5.18b)	14	7	4	1	87.50	63.64	73.68
	RGX (Fig 5.18d)	14	7	3	0	100.00	70.00	82.35
	RDPCA (Fig 5.17d)	11	11	0	0	100.00	100.00	100.00
House dataset with 25% noise	RGR (Fig 5.19b)	14	7	5	1	87.50	58.33	70.00
	RGX (Fig 5.19c)	14	5	3	0	100.00	62.50	76.92
	RDPCA (Fig 5.19d)	11	11	0	0	100.00	100.00	100.00

Dataset 5.6: Traffic furniture dataset

The dataset in Figure 5.20a is a MLS dataset consists of 170,815 points that describes part of a road, kerb and footpath. It contains road side furniture including road signs, long and approximately cylindrical surfaces i.e. signs and light poles. We call this the 'traffic furniture' dataset. It contains 25 surfaces, 10 of which are planar and 15 are non-planar. We set parameters: k =50, $\theta_{th} = 15°$, and $R_{min} = 10$ to perform the segmentation algorithm. The segmentation results are shown in Table 5.2 and Figure 5.20(b, c and d). Figure 5.20b shows that based on PCA normals and curvatures, the proposed algorithm properly segments 10 surfaces with eight and six OS and US, respectively. RANSAC gives a total of 25 segments in which 17 surfaces are properly segmented, but has also four and three OS and US, respectively. Significantly, PCA (Figure 5.20b) and RANSAC (Figure 5.20c) fail to separate the road, kerb and footpath into individual surfaces which are under segmented into one surface. Figure 5.20d shows that the algorithm based on RDPCA saliency features accurately segments all 25 surfaces without any OS and US. Quantitative performance measures in Table 5.2 (columns 7, 8 and 9) show that in terms of recall (r), precision (p) and accuracy (F), PCA and RANSAC have overall accuracies of 58.82% and 82.93% respectively, whereas RDPCA has a 100% success rate for all the three measures r, p and F.

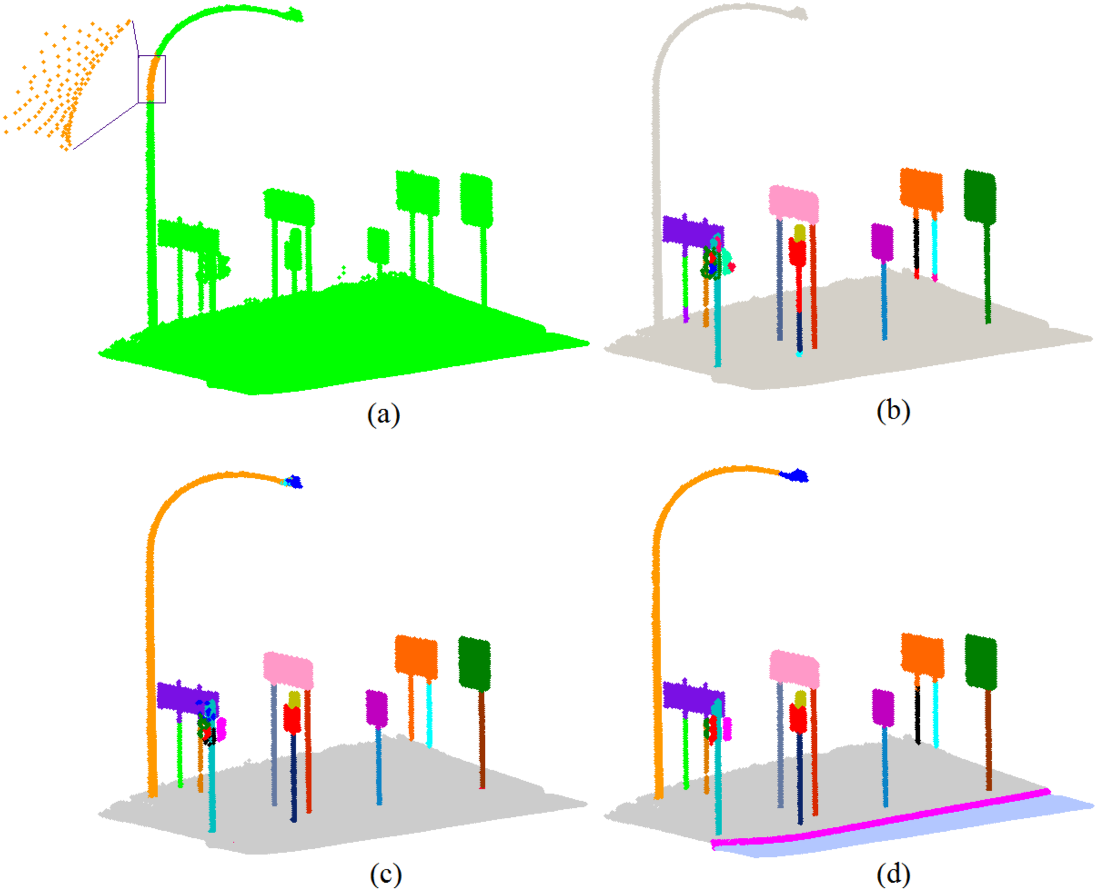


Figure 5.20 (a) Traffic furniture dataset. Segmentation results: (b) PCA, (c) RANSAC, and (d) RDPCA.

To evaluate the performance of the proposed method for noisy data, we add 15% Gaussian noise with mean=0.0 and StD=0.2 to the previous dataset (Figure 5.20a). The resulting noisy dataset is shown in Figure 5.21a. After adding noise, PCA (Figure 5.21b) produces only eight PS with 32 and four OS and US respectively, and RANSAC (Figure 5.21c) gives 16 PS with 10 and two OS and US, respectively. Based on robust saliency features, RDPCA (Figure 5.21d) gives significantly better results of 24 PS and one OS. In Table 5.2, the results for the noisy data show the accuracy rates for PCA, RANSAC and RDPCA are 30.77%, 72.73% and 97.96%, respectively.

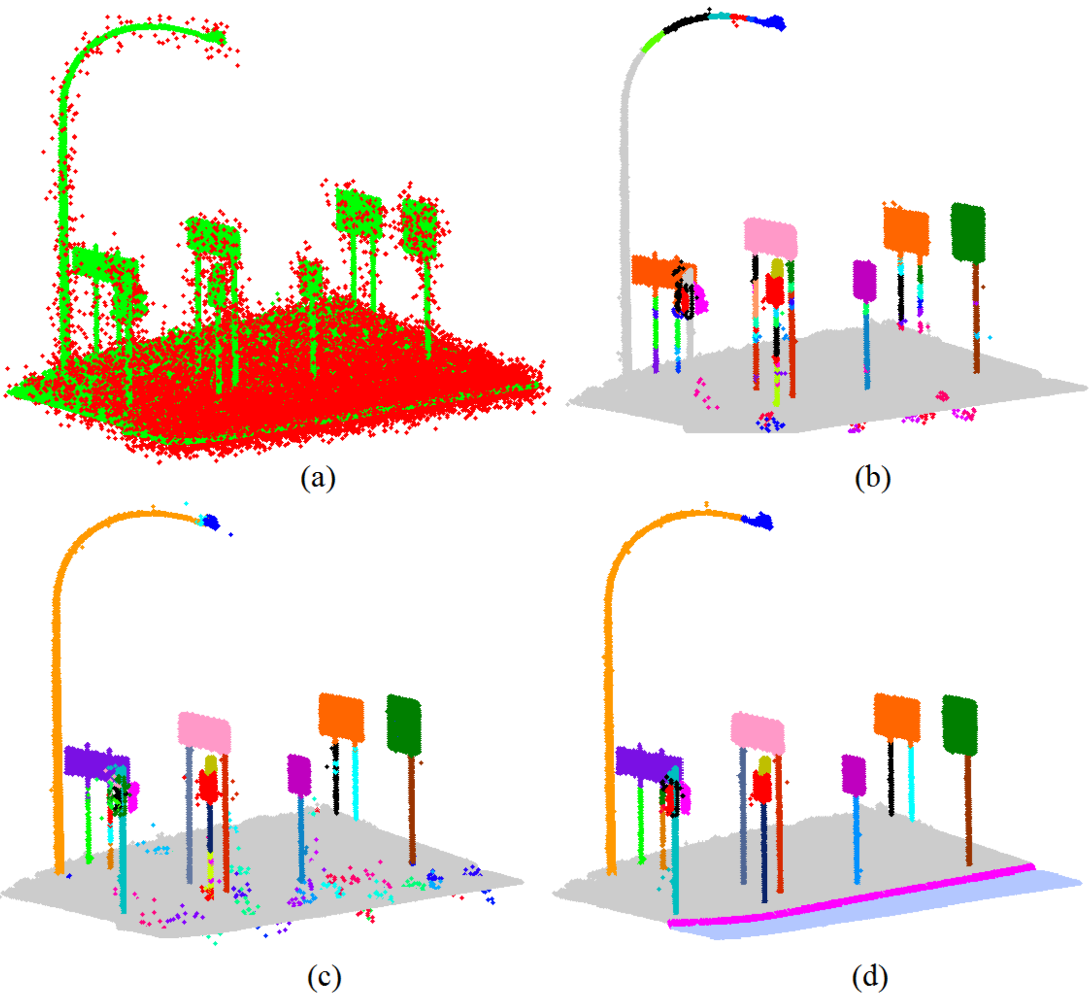


Figure 5.21 (a) Noisy traffic furniture dataset. Segmentation results: (b) PCA, (c) RANSAC, and (d) RDPCA.

Table 5.2 Performance evaluation for the proposed segmentation algorithm.

Data Name	Methods	TS	PS	OS	US	r	p	F
Traffic furniture data	PCA	26	10	8	6	62.50	55.56	58.82
	RANSAC	25	17	4	3	85.00	80.95	82.93
	RDPCA	25	25	0	0	100.00	100.00	100.00
Traffic furniture data with 15% noise	PCA	53	8	32	4	66.67	20.00	30.77
	RANSAC	31	16	10	2	88.89	61.54	72.73
	RDPCA	26	24	1	0	100.00	96.00	97.96

5.6 Segmentation Results Merging for Large Data

It is not feasible to segment a large volume of point cloud data at one time i.e. hold the complete dataset in memory. Common practice is to manually divide the data into slices and process slice by slice. This is satisfactory given that many objects are local in extent and each slice can be made to cover them completely. However this is not the case for linear features such as crash barriers and the pavement. A method is needed to process strips one at a time and then merge the results seamlessly. This section extends our segmentation algorithm for merging several pieces of segmented slices.

5.6.1 Algorithm Implementation

Assume, we need to segment a large point cloud dataset. To make the task easy, we slice the large data into a reasonable number of pieces along the appropriate surface direction, usually in the x or y direction. For each object, we need to segment all the pieces individually and then merge the resultant segments. With the intention of merging the segmentation results of sliced point cloud data, we first slice the dataset into a number of pieces having a significant size of common overlapping region for the successive pieces. For example, we can have four slices (S_1, S_2, S_3 and S_4) of a point cloud dataset in which each adjacent pair of slices has an overlapping region. Any segments that do not occur in an overlap region are regarded as part of the final result. Segments that fall in the overlapping regions have the potential of being modified and merged. We perform our segmentation Algorithm 5.4 on every individual slice. To merge the segmentation results, we start from the segmentation results for S_1 and S_2. We search for the segments in S_1 and S_2 that contain points from the common region of S_1 and S_2 called S_{12}. The segments of S_1 that do not occur in S_{12}, we put in a list of final segments S_P for P. Now, we create a new dataset S_1S_2 of the points from the segments that have common points in S_1 and S_2 and re-segment the new dataset. The segments from S_1S_2 and the segments of S_2 that are not involved with S_{12} together will be considered as the updated segmentation results of S_2 for merging with the segments of S_3. This way of segmentation and merging for consecutive pairs of slices such as: S_2, S_3 or S_3, S_4 continues until the last slice. The process of merging the segments for a large point cloud data is summarized in Algorithm 5.5.

Algorithm 5.5: Segments Merging from Different Slices

Input:

n_s: number of slice

S: list of all slices $(S_1, S_2, S_3, \cdots, S_{n_s})$

Ls: list of segments of each slice S_i in S

k: neighbourhood size

θ_{th}: angle threshold

R_{min}: minimum number of points to build a significant region

Output:

S_P: list of segments for the whole dataset P

```
for $i$= 1 to $(n_s - 1)$ do
    if common points between $S_i$ and $S_{i+1}$ then
        Find the segments that have common points in $Ls_i$ and $Ls_{i+1}$ and put them into $CS_i$ and $CS_{i+1}$, respectively
        Find the segments that have no common points in $Ls_i$ and $Ls_{i+1}$ and put them into $S_P$ and $NS_{i+1}$, respectively
        Merge the segments $CS_i$ and $CS_{i+1}$ and put them into $M$
        $T \leftarrow kd\_tree(M)$
        $[Np, ED] \leftarrow k\mathrm{NN}Search(T, M, k)$
        $[N, K] \leftarrow normal - curvature(M, Np, h, \epsilon, P_r)$ [Algorithm 5.4]
        $MS \leftarrow RegionGrowing(M, Np, ED, N, K, \theta_{th}, R_{min})$ [Algorithm 5.4]
        Update $S_{i+1}$ and $Ls_{i+1}$ and $S_P$
        if $(i+1) < n_s$ then
            $S_{i+1} \leftarrow CS_i \cup S_{i+1}$
            $Ls_{i+1} \leftarrow MS \cup NS_{i+1}$
        else
            $S_P \leftarrow S_P \cup MS \cup NS_{i+1}$
        end if
    else
        if $(i+1) < n_s$ then
            $S_P \leftarrow S_P \cup Ls_i$
        else
            $S_P \leftarrow S_P \cup Ls_i \cup Ls_{i+1}$
        end if
    end if
end for
```

5.6.2 Experiments

Dataset 5.7: Road corridor dataset

In this section, we demonstrate our proposed robust segmentation and merging algorithms on a large MLS dataset. This dataset of a road corridor consists of many objects including the road surface comprising pavement, kerb and footpath, road side furniture such as road signs, a fence, long and approximately cylindrical surfaces including signs and light poles, and more complex surfaces such as lamps. We label this the 'road corridor' dataset shown in Figure 5.22a. It consists of 978,029 points and covers about 50m of travel. We slice the dataset along the y axis into eight equal parts keeping 0.5m common or overlapping regions for pairs of consecutive slices. Common regions are shown in red in Figure 5.22a. We set the parameters: $k = 50$, $\theta_{th} = 10°$, and $R_{min} = 10$, the same as for the previous experiments. We know the values of the parameters depend on the data and are based on the size, complexity and structure of the objects. We fix the parameters using experience from knowledge of similar data and/or from prior experiments. We keep the parameters the same for all eight segmentations with the results in Figures 5.22(b to i) showing that all eight slices are properly segmented and almost all the surfaces of the objects have been successfully extracted.

To demonstrate the efficiency of the proposed merging algorithm, we perform our method on the eight different segmentation results of Figures 5.22(b to i). The final results of merging are shown in Figures 5.23(a and b) for two different viewpoints: side view and along the road view respectively. Results in the figures show that segments are properly determined across the slices. For example, consider the road pavement, kerb and footpath that are continuous surfaces for the most part of the dataset of Figure 5.22a, and have common regions between pairs of successive slices. Figures 5.23(a and b) show they are properly merged through the slices. Another example is the big building along the left side of the road that consists of blue, purple, red, cyan, olive and green parts. It occurs across a number of slices but these are accurately merged in Figure 5.23.

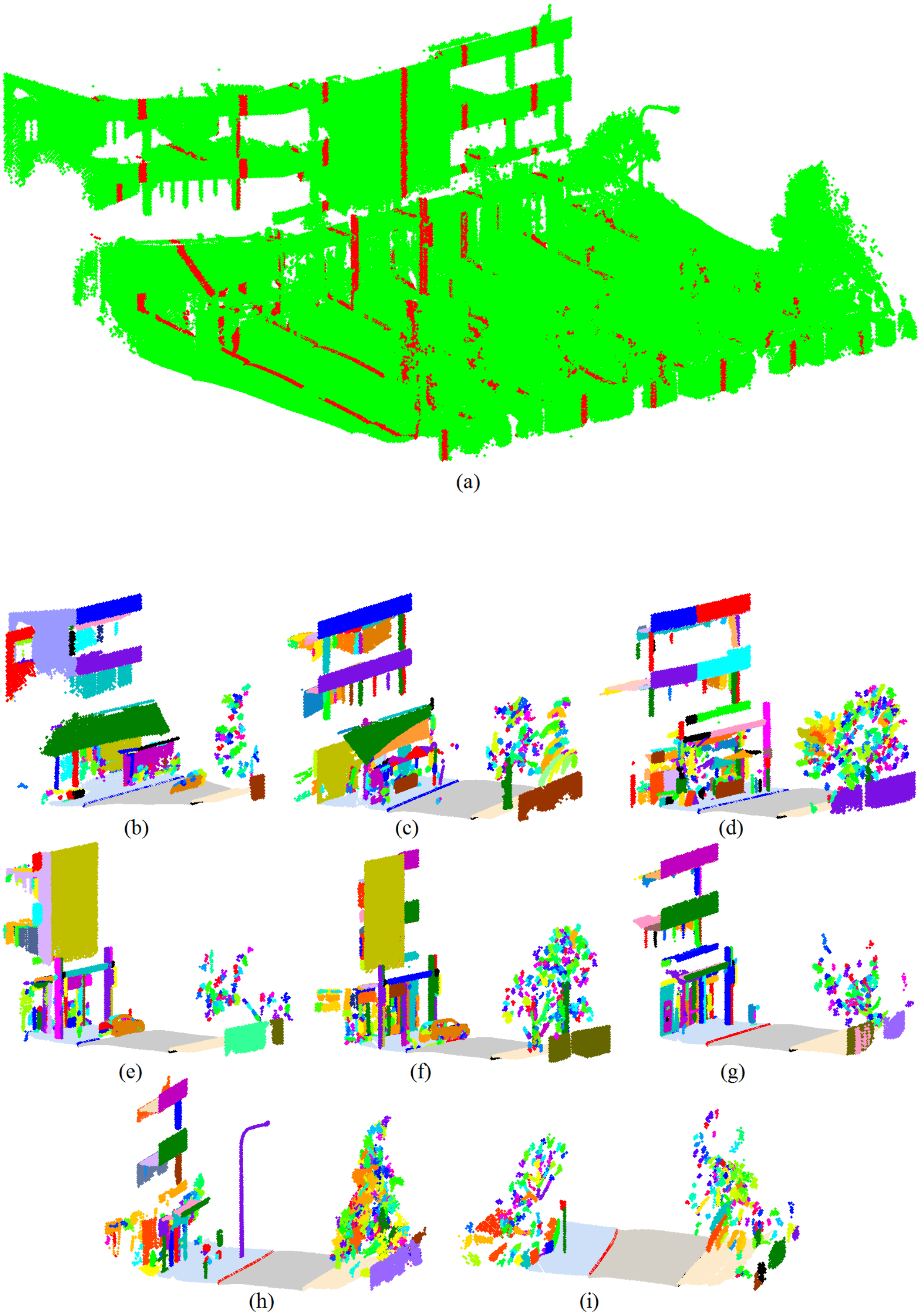


Figure 5.22 (a) Road corridor dataset with common boundaries of slices in red. Segmentation results for eight slices (b), (c), . . . ,(i).

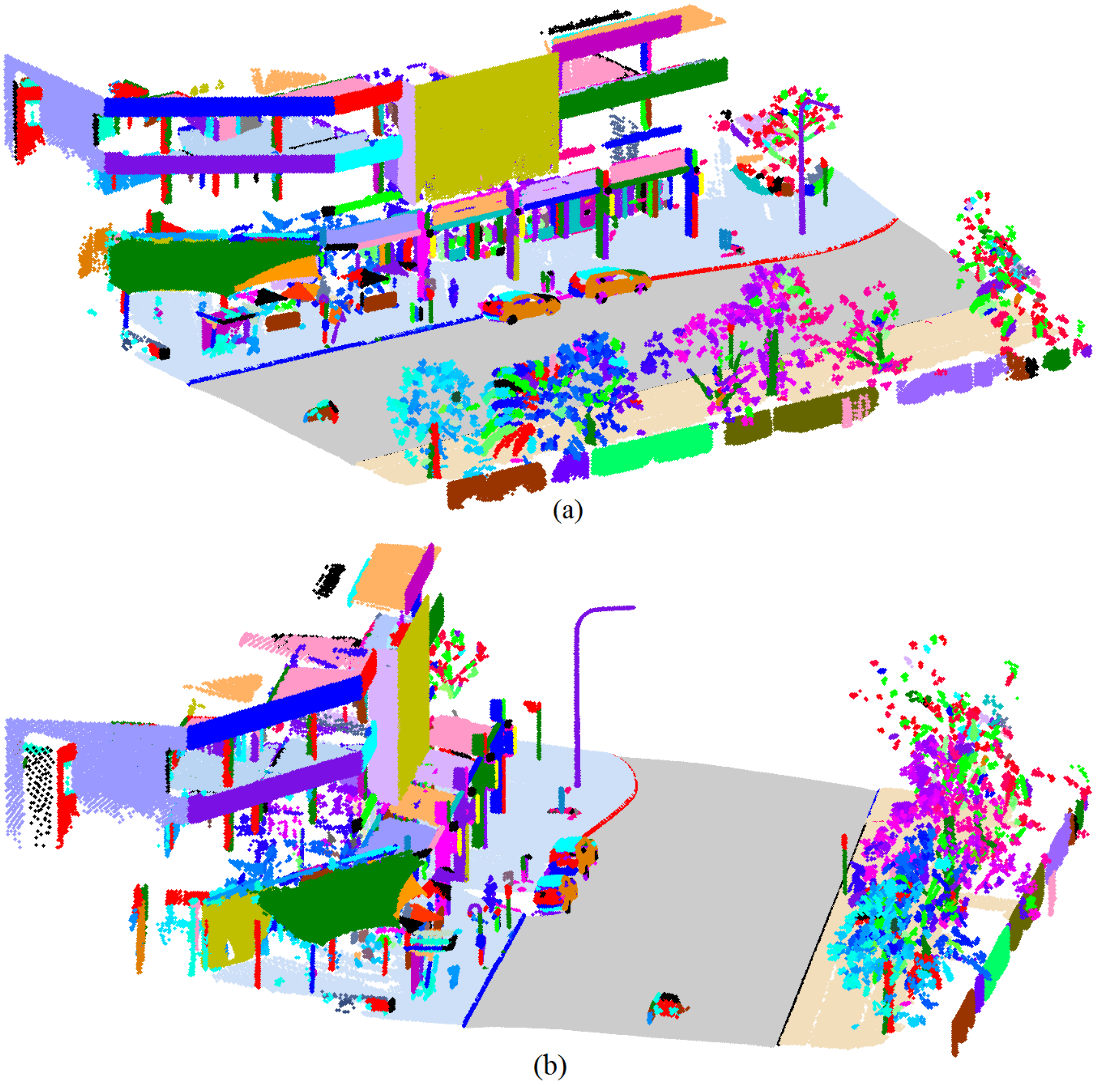


Figure 5.23 Final segmentation results for road corridor dataset after merging: (a) right side view, and (b) along the road view.

5.7 Objects Class Recognition

The next stage is to take the output of the point cloud labelling or segmentation algorithms and cluster the labelled points to determine a high level description of the surfaces and objects e.g. a vertical cylinder, or a horizontal planar surface. This is beyond the scope of this thesis but an idea of an approach worth pursuing is now proposed. We take a part of the small dataset from Figure 5.20a. We perform our segmentation algorithm with results shown in Figure 5.24. The local normals for each labelled point in Figure 5.24 are projected onto the Gaussian sphere shown in Figure 5.25. The dataset contains signs that are planar surfaces, poles that are cylindrical surfaces etc. The projection of the labelled normals onto the Gaussian sphere indicates clusters that correspond to the different features and their types. The projected results clearly show the surface types of the resultant segments. For example, planar surfaces like the signs with labels 3, 4, 9 in Figure 5.24 are projected as clustered points on the Gaussian sphere as each planar surface is made up of normals that are approximately the same. Cylinders occur as long thin clusters because they have a range of normal orientations in one direction but are constant in the other direction. As all the cylinders in Figure 5.24 are vertically aligned, the clusters are all horizontal lines and overlap each other on the sphere (Figure 5.25). Objects with spherical surfaces form large spread out clusters. This feature type recognition can be used later for object modelling and reconstruction purposes through clustering methods. There would still be a need to determine the parameters of each surface component.

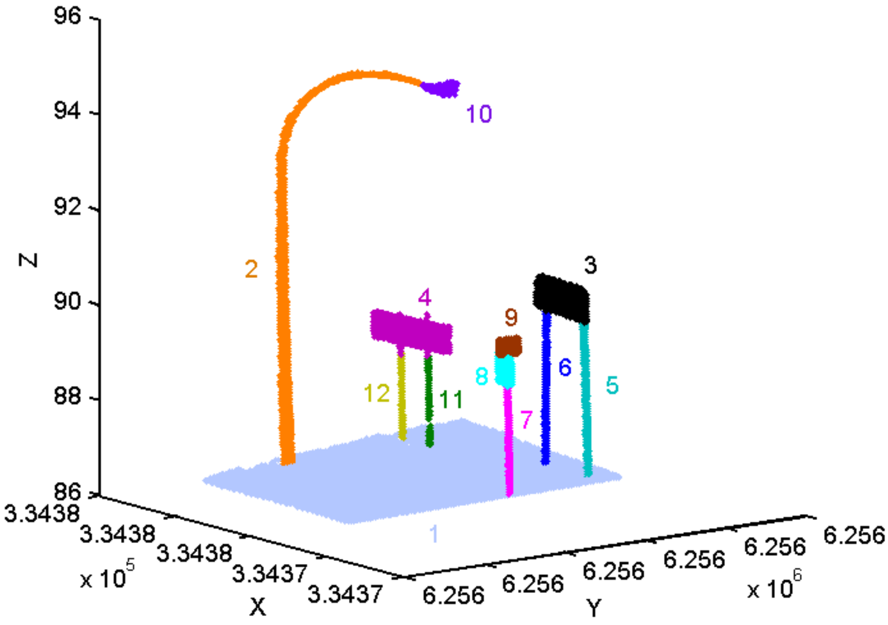


Figure 5.24 Segmentation results for a part of the traffic furniture dataset (Figure 5.20a). Results labels 12 different segments for 12 different surfaces.

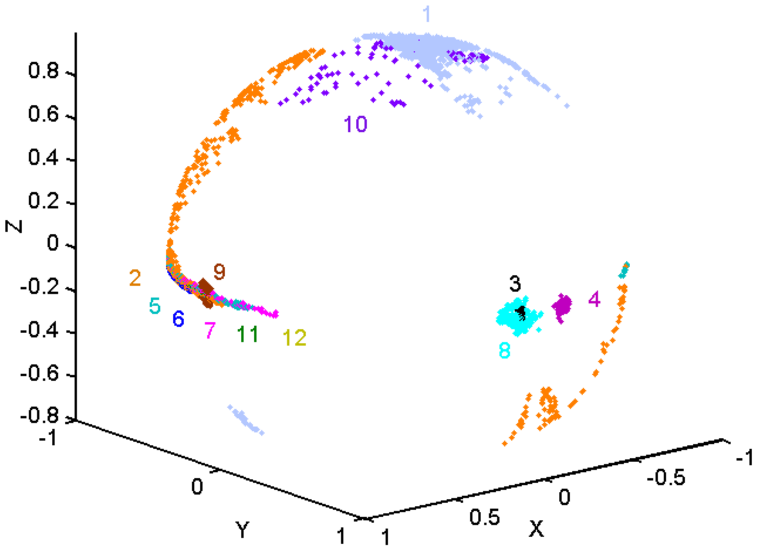


Figure 5.25 Gaussian Sphere; different labels (1 to 12) are given for the respective segmented surfaces in Figure 5.24. Colours are matched with the segmentation results.

5.8 Conclusions

This chapter proposes algorithms for: (i) classification of points into edge/corner points and surface points, (ii) robust segmentation for multiple planar surface extractions, (iii) robust segmentation for planar and/or non-planar complex objects surfaces, and (iv) merging segments from different pieces of point cloud data. In the proposed methods, the basic ideas of robust and diagnostic statistics are coupled in MCMD_Z with PCA and used to get robust saliency features. First, the algorithms estimate the best-fit-plane based on the majority of consistent data, within the local neighbourhood of each point of interest, and then finds the outliers locally for every neighbourhood based on the results from the majority of good points. In the second stage, the required saliency features: normals and curvatures, are estimated for every point by PCA based on the cleaned data consisting of only inlier points, found in its local neighbourhood. The method for robust saliency features is named RDPCA. The robust saliency features found from the cleaned local neighbourhood are used for region growing to segment the point cloud data. Results for experiments on artificial and real MLS point cloud data show that the proposed RDPCA based algorithms have the advantages that it: (i) is computationally simpler, (ii) is significantly faster than robust versions of PCA (see Chapter 4), (iii) can efficiently handle high percentages of clustered and uniform outliers, (iv) outperforms PCA and is significantly better than RANSAC for classification and segmentation, (v) is semi-automatic, and depends only on two user defined parameters (neighbourhood size and angle threshold), (vi) reduces over and under segmentation, and (vii) produces more accurate and robust results.

The RDPCA algorithm based on MCMD_Z (described in Chapter 4) breaks down at more than 50% outliers, but using MCMD_MD in the segmentation Algorithm 5.4 can increase outlier tolerance levels up to 75% and gives efficient segmentation.

The next chapter will investigate filtering algorithms in point cloud data and will propose variants of robust filtering (ground surface extraction) methods based on robust locally weighted regression.

Chapter 6

”Science knows no country, because knowledge belongs to humanity, and is the torch which illuminates the world. Science is the highest personification of the nation because that nation will remain the first which carries the furthest the works of thought and intelligence.”

Louis Pasteur

”To improve is to change; to be perfect is to change often.”

Winston Churchill

Robust Ground Surface Extraction

6.1 Introduction

In many application areas of Mobile Laser Scanning (MLS), it is required to classify points into ground and non-ground points. Ground points are those that have the lowest heights and need to be separated from those making up vegetation, walls, poles etc. Examples of situations where separation is needed include corridor mapping, road assets management, infrastructure planning, environmental risk management and protection, vegetation analysis and for maintenance of urban street scenes (Wagner et al., 2004; Kraus et al., 2006; Pfeifer and Mandlburger, 2009; Briese, 2010; Pu et al., 2011; Serna and Marcotegui, 2014). It is also needed to determine terrain and off-terrain information for further analysis. Extracting the ground surface is also useful for many other point cloud post processing tasks. Removing the ground from the data can simplify, and can minimize time and cost for segmentation, feature extraction, surface reconstruction and modelling of above ground features. The same argument is valid when only considering ground points. If the objective is to get information only related to the ground surface (e.g. objects like road pavement, kerb, footpath, road markings) then again it is better to minimize the data size by removing the non-ground points. Separation of ground points from non-ground points is closely related with Digital Terrain/Elevation Modelling (DTM/DEM) (Hebert and Vandapel, 2003; El-Sheimy et al., 2005; Pfeifer, 2005; Kraus et al., 2006; Pfeifer and Mandlburger, 2009; Pu et al., 2011; Garouani and Alobeid, 2013; Yang et al., 2013). A DTM/DEM is a mathematical representation, i.e. a model of the bare earth (ground surface) in digital form (Briese, 2010). It is different to a Digital Surface Model (DSM) which contains the vegetation and built environment.

The problem of filtering in point cloud data is formally defined (Pfeifer and Mandlburger, 2009; Briese, 2010) as follows. For a given set of points $P = \{p_1(x_1, c_1), \ldots, p_n(x_n, c_n)\}$, a point embedded in 3D space: $x_i = (x_i, y_i, z_i) \in R^3$ has an individual classification label c_i. The task is to find a classifier function $f : x \to c$, which maps each point p_i to its classification label $c_i \in C = \{\text{terrain, off-terrain}\}$. In this way the classification label c_i of each point p_i can be filtered and assigned to the attribute value 'terrain' or 'off-terrain'.

Many methods have been developed for DTM/DEM generation or filtering, which means classification of point cloud data into ground (terrain) and non-ground (off-terrain) points. This has been mainly covered in areas such as statistics, computer vision, pattern recognition, photogrammetry and remote sensing (Vosselman, 2000; Bartels et al., 2006; Crosilla et al., 2011; Zhou et al., 2012). To meet the challenges associated with classification/ground filtering, many methods have been introduced over the last two decades (Lindenberger, 1993; Kraus and Pfeifer, 1998; Brovelli et al., 2004; Wagner et al., 2004; El-Sheimy et al., 2005; Belton and Bae, 2010; Crosilla et al., 2011). A comparative analysis of the different methods was conducted by the ISPRS Working Group (WGIII/3; Sithole and Vosselman, 2004) which showed that no method is sufficiently good for every dataset, and the problems addressed by the authors are not entirely solved (Pfeifer and Mandlburger, 2009). Many of them do not perform well in the presence of multiple structures like ramps, sharp edges, steep slopes and isolated ground points. Hence, there is much interest in developing new efficient methods in order to solve the problems and to get good quality results.

It is known that parametric polynomials estimate parameters that best fit the data for a pre-specified family of functions. In many cases, this method yields easily interpretable models that do a good job of explaining the variation in the data, but it is not always true. The chosen family of functions can be overly-restrictive for some types of data (Avery, 2012). Fan and Gijbels (1996) showed that even a 4^{th} order polynomial fails to give visually satisfying fits. As an alternative, higher order fits may be attempted, but this may leads to numerical instability. As a remedy, the Locally Weighted Regression (LWR) approach can be used. We choose LWR because it satisfies many desirable statistical properties. Most importantly, it adapts well to bias problems at boundaries and in regions of high curvature (Cleveland and Loader, 1996). Fitting within a local neighbourhood considers local point density accurately, which is not always possible for global model polynomial fitting for the whole dataset. We know that significant point density variation is typical in laser scanner point cloud data, and it may create problems (Sithole and Vosselman, 2004, 2005). In particular, for steep slopes, this type of global parametric model fitting may lead to misclassification results and local fitting typically gives better results.

In this chapter, we propose a new algorithm based on Robust LWR (RLWR). Local fitting uses a locally weighted interpolation function based on local neighbourhood for each and every point. It finds the fine detail in the point cloud by smoothing. The algorithm proposed in this chapter is an iterative process. A predefined robust weight function is imposed for each iteration according to the residual values, which are the deviations between the z_i values and their current fits $\hat{z}_i$. Inclusion of a robust weight function in the proposed algorithm makes estimates robust and down-weights the height error of the points w.r.t. the fit for the intermediate steps in a robust fashion. Moreover, it reduces the influence of outliers on the fits. The remaining of the chapter consists of the following sections.

Section 6.2 gives a short review of the relevant literature. Section 6.3 contains brief discussions about related principles and methods used in the proposed algorithms. In Section 6.4 an algorithm is proposed for classification of ground and non-ground points. The performance of the algorithm is demonstrated and evaluated using several real MLS datasets in Section 6.5. Section 6.6 concludes the chapter.

6.2 Literature Review

Many filtering algorithms have been developed over the years. An overview can be seen in Pfeifer (2003), Sithole and Vosselman (2004), El-Sheimy et al. (2005), Kobler et al. (2007), and Briese (2010). In this section we review some well-known methods that are relevant here and show research advancement in this area.

Existing filtering methods based on different concepts with different complexities and performance characteristics can be categorized into four general groups as follows: (i) morphological filtering, (ii) progressive densification, (iii) surface based filtering, and (iv) segment based filtering. In mathematical morphologic filtering, the concept of mathematical morphology (Haralick and Shapiro, 1992) has been used. Lindenberger (1993) published one of the first morphological filtering methods, in which initially, a rough ground surface is extracted by using a seed point that is the lowest based on the assumption that the lowest point belongs to the ground. Then the rough terrain is refined with an auto-regression process. This algorithm is vulnerable to the size of the structure element (Shan and Sampath, 2005). Later Kilian et al. (1996) used different morphologic operators, and Vosselman (2000) developed a slope based filter incorporating the idea of maximum admissible height difference between two points as a function of the distance between the points. Zakšek and Pfeifer (2004) noticed that although the morphologic filtering algorithm is effective in areas with small differences it is not so good in areas with steep slopes.

Progressive densification algorithms start with a small subset of the data and iteratively increase the amount of information used to classify the whole dataset step-by-step. Axelsson (2000) introduced a progressive Triangular Irregular Network (TIN). The algorithm uses the lowest point in large grid cells as the seeds for his approach. Subsequently, the first subset is triangulated in order to form a reference bare earth surface. Then, for each of the triangles within the TIN an additional terrain point is included if certain criteria are fulfilled. This iterative process continues until no further points can be added to the TIN. Sohn and Dowman (2002) proposed a similar type of algorithm where the initial TIN seed point is the lowest point in the four corners of the entire area. Then, in the following 'downward step' the lowest point within each triangle is added to the TIN, and the step continues until no point below the TIN can be added.

Surface based filtering algorithms start by considering all the points belonging to the ground surface and gradually removes those points that do not fit with a general surface model. Kraus and Pfeifer (1998) introduced a surface based filtering technique, using robust interpolation and linear prediction, which is an iterative process based on linear least squares interpolation. This algorithm integrates the filtering and DTM interpolation in the same process. It determines an individual weight between 0 and 1 for each irregularly distributed point in the dataset in such a way that the modelled surface represents the terrain. Finally, all the data points are classified into ground and non-ground points based on a predefined height difference threshold value w.r.t. the final DTM. Pfeifer et al. (2001) and Briese et al. (2002) embedded robust interpolation in a hierarchical approach that can handle different levels of resolution and reduce time. Zakšek and Pfeifer (2004) claimed that a robust interpolation method is more efficient than morphologic filtering in steep slopes covered by forest. Akel et al. (2007) proposed an algorithm based on orthogonal polynomials for extracting terrain points from LiDAR data. The authors pointed out that in contrast to other interpolation methods, orthogonal polynomials are not affected by truncation errors, round-off errors, ill-conditioned cases and unstable systems. The use of a high-degree interpolation function makes it possible to fit a global function that can describe the terrain at a given level of detail. Fan and Gijbels (1996) claimed higher order fits may lead to numerical instability.

Segmentation/clustering approaches classify whole segments (homogeneous regions) rather than one single point. This approach classifies segments (a group of points) based on local geometrical relations like height, slope or curvature in a certain neighbourhood (Sithole and Vosselman, 2005). Tóvári and Pfeifer (2005) proposed a two-step segmentation algorithm that starts from a seed point for region growing, examines k neighbourhood points to see whether they fulfil certain criteria, and then uses robust interpolation for point groups. The authors require more information about segmentation parameters and explicit break-line information to get more accurate filter results. Edge based clustering is introduced by Brovelli et al. (2004). This detects edges by using a threshold of the gradient. Points inside the closed edges are considered as the object points and the rest are considered as the terrain points. Pfeifer and Mandlburger (2009) pointed out that the segmentation based algorithms have advantages in areas strongly influenced by human building activities (houses, streets, dams, embankments, etc.), and is not affected by edge effects. Methods that use statistical tools, e.g. skewness balancing introduced by Bartels et al. (2006) is mainly a segmentation algorithm based on the central limit theorem where the statistical measure skewness is chosen to describe the characteristics of the point cloud distribution and has been used as a termination criterion in a segmentation algorithm. This algorithm has been further developed by combining the kurtosis measure proposed by Crosilla et al. (2011). Kobler et al. (2007) introduced the 'Repetitive Interpolation' filter that cannot be assigned to one of the above specific approach, as it works on a pre-filtered dataset.

Some of the filtering algorithms work on raster data structures. These algorithms introduced in the digital image processing area, use fast neighbourhood operations. However, the main disadvantage of this type of algorithm is that it results in loss in precision (Axelsson, 1999), and may lead to undesired effects. For example gaps can occur caused by occlusion (Briese, 2010).

6.3 Related Principles and Methods for the Proposed Algorithms

The algorithm proposed in this chapter mainly uses the concepts of regression analysis. This section presents the basic ideas of regression analysis, robust regression and re-weighted regression.

6.3.1 Regression and Robust Regression

Regression analysis is an important statistical tool for fitting a model equation to observed variables frequently employed in many areas e.g. computer vision, data mining, machine learning, pattern recognition, photogrammetry and remote sensing (Meer et al., 1991; Bishop, 2006; Nurunnabi and Dai, 2012; Nurunnabi and West, 2012; Stal et al., 2014). The ease of computation and the presence of optimal properties, when the underlying error distribution is Gaussian, have made the Least Squares (LS) method the most popular form of regression. However the method becomes unreliable and produces misleading results if the noise has a non-zero mean component and/or if outliers are present in the data (Chatterjee and Hadi, 1988; Meer et al., 1991; Rousseeuw and Leroy, 2003; Nurunnabi et al., 2014b). The classical linear model is:

$$y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{im}\beta_m + \epsilon_i, \quad i = 1, \ldots, n \tag{6.1}$$

where n is the sample size, $x_{i1}, \ldots, x_{im}$ are the explanatory variables, y_i is the response variable, $\beta_0, \beta_1, \ldots, \beta_m$ are the regression coefficients or parameters, and ϵ_i is the error term, assuming the error term follows a normal distribution with mean 0 and variance σ^2. Applying a regression estimator gives the regression coefficient $\hat{\beta}$, where:

$$\hat{y}_i = \hat{\beta}_0 + x_{i1}\hat{\beta}_1 + \cdots + x_{im}\hat{\beta}_m, \quad i = 1, \ldots, n \tag{6.2}$$

and

$$r_i = y_i - \hat{y}_i, \quad i = 1, \ldots, n \tag{6.3}$$

where r_i is the i^{th} residual. The classical LS method for estimating the regression parameters is:

$$\underset{\hat{\beta}}{minimize} \sum_{i=1}^{n} r_i^2. \tag{6.4}$$

It is known that the presence of outliers can substantially change the LS estimates and produces erroneous results and the wrong conclusions (Chatterjee and Hadi, 1988; Rousseeuw and Leroy, 2003). To avoid the influence of outliers, robust regression and regression diagnostics are two approaches that have been developed in statistics (Atkinson and Riani, 2000; Rousseeuw and Leroy, 2003; Chatterjee and Hadi, 2012; Nurunnabi et al., 2014b). Robust regression first fits a regression to the majority of the data and then finds outliers defined as those points that possess large residuals from the robust output. There are many robust regression methods such as M, GM or S-estimator based methods, Least Median of Squares (LMS) regression, Least Trimmed Squares (LTS) regression, and Reweighted Least Squares (RLS) regression (Rousseeuw and Leroy, 2003). The most popular robust regression techniques are: LMS, LTS and RLS (Rousseeuw and Leroy, 2003; Chatterjee and Hadi, 2012). LMS was proposed by Hampel (1975) and later developed by Rousseeuw (1984). It minimizes the median of the squared residuals instead of minimizing the sum of the squared residuals i.e.

$$\underset{\hat{\beta}}{minimize}\ \underset{i}{median}\ r_i^2. \tag{6.5}$$

This estimator effectively ignores almost half of the observations having the largest residuals. Unfortunately, LMS possesses poor asymptotic efficiency (i.e. slow convergence rate). Rousseeuw (1984) also introduced LTS regression defined as:

$$\underset{\hat{\beta}}{minimize} \sum_{i=1}^{h} r_i^2, \tag{6.6}$$

where $r_1^2 \leq \cdots \leq r_{n/2}^2 \leq \cdots \leq r_h^2 \leq \cdots \leq r_n^2$ are the ordered squared residuals (the residuals are first squared and then ordered). The LTS is similar to LS, the only difference is that the largest $(n-h)$ squared residuals are not used in the summation, thereby allowing the fit to stay away from the outliers (Rousseeuw and Leroy, 2003). Like LMS, this estimator is also equivariant for linear transformations and is related to projection pursuit (Friedman and Tukey, 1974). Both the methods have the highest possible BP of 50%. The highest BP is achieved when h is approximately $n/2$. Another type of regression is weighted least squares (Rousseeuw and Leroy, 2003), which finds outliers first then assigns a weight to each point according to the outlyingness of the point as:

$$w_i(x) = \begin{cases} 0, & \text{if the } i^{th} \text{ point is identified as an outlier} \\ 1, & \text{if the } i^{th} \text{ point is not an outlier.} \end{cases} \tag{6.7}$$

Then the model is refitted by using LS. Hence, the Weighted Least Squares (WLS) can be defined as:

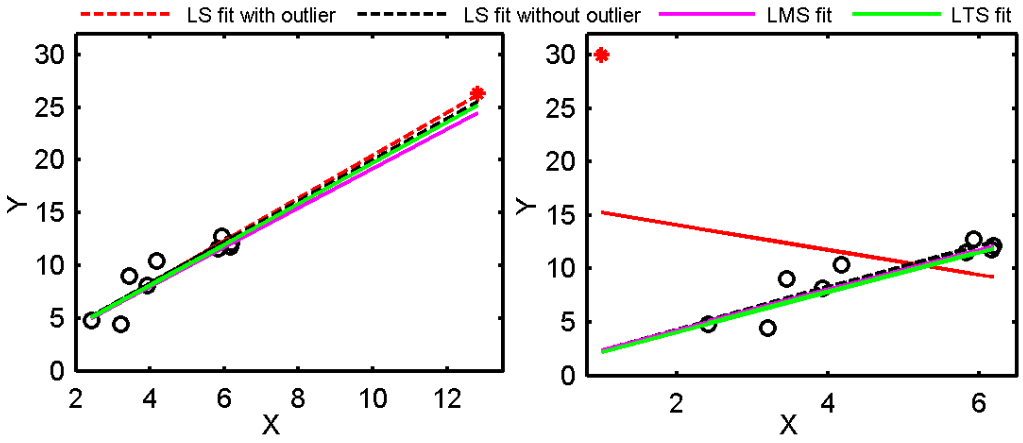
$$\underset{\hat{\beta}}{minimize} \sum_{i=1}^{n} w_i r_i^2. \tag{6.8}$$

The outliers can be identified by using robust regression or using regression diagnostics approaches.

Regression diagnostics is designed to detect and delete, or refit if necessary, the outliers first and then to fit the good data by the LS method (Rousseeuw and Leroy, 2003). Many regression diagnostic methods have been developed (Cook and Weisberg, 1982; Nurunnabi et al., 2011; Chatterjee and Hadi, 2012;

Nurunnabi et al., 2014b). However, for reasons of popularity and robustness, in this chapter we employ only the LMS and LTS regression in our proposed algorithm.

To see the outlier effects on LS and to explore the necessities of robust regression, we demonstrate the non-robust (LS) and robust (LMS and LTS) methods on a simulated dataset. In Figures 6.1(a and b), we create two small datasets of 10 points in 2D space. Regular points are in black and the red points are the outliers. Regular points are the same for both the datasets and outliers are created in different directions. In Figures 6.1a, the outlier follows the linear pattern defined by the majority of the points, and in Figures 6.1b the outlier is in the y (response variable) direction and does not follow the pattern defined by the rest of the nine points. However, it is clear that both the points are sufficiently distant from the bulk of the points. Using LS with and without the outliers, LMS and LTS techniques, we fit a simple linear regression model $y = \beta_0 + \beta_1 x + \epsilon$. Results are shown in Table 6.1 and in the respective figures. Figures 6.1a shows that all the fitted lines for the different methods are almost in the same direction although they have slightly different parameter values as shown in Table 6.1. In the case of Dataset 2 shown in Figures 6.1b, we see LS totally failed to find the pattern of the majority points as lines are in the reverse direction for the data with and without the outlier. However, the lines produced using LMS and LTS are similar to the LS line without the outlier, which means the robust lines are not affected by the outlier. The robust methods (LMS and LTS) and regression diagnostics (LS fitting without outlier) give almost similar results. We also calculate the coefficient of determination R^2 for the LS method. The coefficient of determination measures the ability of the fitted model to represent the observed data (Montgomery et al., 2012). In Table 6.1 for Dataset 2 with the outlier, we get $R^2 = 8.38\%$ for the LS model that increases to 82.98% without the outlier. This is the same as for Dataset 1 without the outlier. It is interesting that for Dataset 1, LS has a larger value of 95.99% for the fit with the outlier than for the fit without the outlier which is 82.98%. The type of outlier in Dataset 1 is called a good leverage point (Rousseeuw and Leroy, 2003; Nurunnabi et al., 2014b), which is a point far from the majority of the data but matches with their linear pattern. The results show that in the presence of an outlier the accuracy measurement R^2 can give rise to inaccurate decisions as it may produce a lower value of R^2. In spite of the presence of outlier, results for the two datasets are the same for the robust methods based on LMS and LTS.



(a)

(b)

Figure 6.1 Regression model fitting: (a) outlier in both x and y directions, and (b) outlier in y direction.

Table 6.1 LS, LMS and LTS regression based model parameter estimation.

Dataset	Methods	Coefficients		R^2
		β_0	β_1	
Dataset 1	LS with outlier	0.162	2.030	0.9599
	LS without outlier	0.496	1.954	0.8298
	LMS	0.555	1.867	0.8576
	LTS	0.435	1.888	NA
Dataset 2	LS with outlier	16.143	-1.159	0.0838
	LS without outlier	0.496	1.954	0.8298
	LMS	0.556	1.867	0.8576
	LTS	0.435	1.887	NA

6.4 Proposed Algorithm

The ground surface extraction or filtering algorithm proposed in this section for classifying ground and non-ground surface points can be considered as a robust interpolation method within the group of surface based filtering methods (Kraus and Pfeifer, 1998; Briese, 2010). It couples the idea of locally weighted regression and the robustification of the weighted regression. It works as a classification method to distinguish in-ground (terrain) and non-ground points (off-terrain objects: buildings, trees, walls, poles, etc.).

6.4.1 Locally Weighted Regression

Locally Weighted Regression (LWR) is a nonparametric statistical approach introduced by Cleveland (1979) and later developed by many others (e.g. Jacoby, 2000; Loader, 2004). It is used to model regression functions or surfaces between explanatory (independent) variable(s) and the response (dependent) variable without any prior specified functional relation between the variables. The LWR is usually termed 'lowess' (LOcally WEighted Scatterplot Smoother) or 'loess'. It is a procedure in which a regression surface is determined by fitting parametric functions locally in the space of the independent variables using weighted least squares in a moving fashion. This is similar to the way that a time series is smoothed by moving averages (Cleveland and Grosse, 1991). Let y_i and $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$; $i = 1, 2, \ldots, n$ be the measurements of dependent and independent variables respectively. Assume that the dataset is modelled as:

$$y_i = g(x_i) + \epsilon_i, \tag{6.9}$$

where ϵ_i are independent and normally distributed with mean 0 and variance σ^2, and $g(x_i)$ is a smooth function of x_i. LWR gives an estimate $\hat{g}(x_i)$ at any value of x_i in the space of independent variables. LWR is nonparametric in the sense that it does not specify the functional form of the whole dataset and no specific assumption is made globally for $g(x)$ but locally around a point x_i. We can assume that $g(x)$ can be well approximated by a member of a simple class of parametric functions (according to Taylor's theorem, any differentiable function can be approximated locally by a straight line). To estimate $g(x)$ at a point x_i, LWR uses a local neighbourhood $N(x_i)$ of k $(1 \leq k \leq n)$ observations in the x space which are closest to x_i. A smoothing parameter $\propto$ $(0 < \propto < 1)$ determines the size of k, which gives the proportion of points that is to be used in each neighbourhood for local regression. A larger local neighbourhood i.e. larger $\propto$, makes the fit smoother. But a smaller local neighbourhood can give a more robust fit. Every point in the local neighbourhood is weighted according to its distance to the interest point x_i. Alternatively, a local neighbourhood can also be considered as a bandwidth or fixed distance $h(x)$, and a smoothing window $x_i \pm h(x)$ may be used for fitting a point x_i. If the same number of observations is on either side of the interest point, the weight function is symmetric, otherwise it is asymmetric. A linear or non-linear polynomial e.g.

quadratic, function of the independent variables can be used to fit the model using the Weighted Least Squares (WLS) method. If locally quadratic fitting is used, the fitting variables are the independent variable(s), their squares, and their cross-products. Locally quadratic fitting tends to perform better than linear fitting where the regression surface has substantial curvature (Cleveland and Devlin, 1988). The local parametric function should be chosen to produce an estimate that is sufficiently smooth without distorting the underlying pattern of the data. LWR uses a weight function $w(x)$ for the least squares fit. A common function is the 'tricube' weight function, defined as:

$$w_i(x) = \begin{cases} \left[1 - \left(\frac{d(x_i,x_j)}{max_{j \in N(x)} d(x_i,x_j)}\right)^3\right]^3 & ; \quad j \in N(x) \\ 0 & ; \quad j \notin N(x), \end{cases} \tag{6.10}$$

where $d(x_i, x_j)$ is the distance between x_i and x_j in x-space. The value of $w_i(x)$ is a maximum for the point closest to x_i and reduces to 0 for the k^{th} nearest x_j to x_i. Points that are too far away with 0 weights will be classified as outliers and deemed influential on the analysis. Figure 6.2 depicts the shape of the tricube weight function.

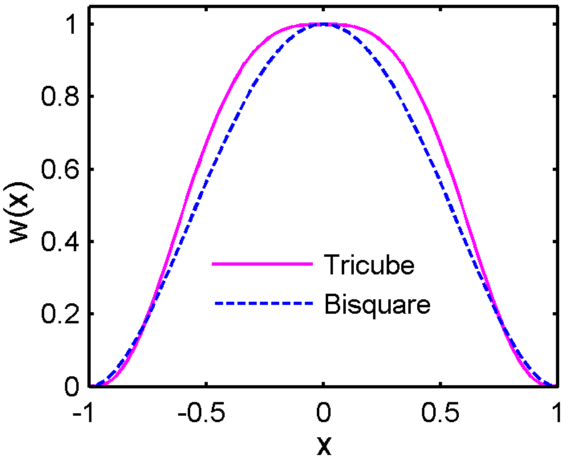


Figure 6.2 Tricube and bisquare weight functions.

Finally, the estimates of the parameters of Eq. (6.9) are the values of the parameters that minimize:

$$\sum_{i=1}^{n} w_i(x)(y_i - g(x_i))^2. \tag{6.11}$$

The coefficients from each local neighbourhood are used to estimate the fitted values at x_i, $\hat{g}(x_i)$. Then the ordered pairs of x_i, $\hat{g}(x_i)$ give the fitted regression line for the whole dataset.

6.4.2 Robustification of Locally Weighted Regression

As for classical regression, LWR may be strongly influenced by outliers because of its least squares nature and hence can give inaccurate non-robust results. The problems of outliers are compounded by the fact that the local regressions typically involve a subset of the complete dataset. Therefore, any erroneous data point will compromise a significant proportion of the points used in the local estimation and their degree of influence may cause false estimates (Jacoby, 2000). To reduce the effects of outliers and to get a robust fit of the model we use two alternative approaches: (i) assigning a robust weight to each data point in the neighbourhood, which is similar to diagnostic concepts, and (ii) fitting by using robust regression e.g. LMS and LTS, for each point with its local neighbourhood.

Cleveland (1979) used the well-known 'bisquare' weight function to get robust locally weighted regression. The bisquare weight function is defined w.r.t. the residuals of the locally weighted fit as:

$$B(r_i^*) = \begin{cases} \left(1 - r_i^{*^2}\right)^2, & for\ |r_i^*| < 1 \\ 0 \quad\quad\quad\ , & for\ |r_i^*| \geq 1, \end{cases} \tag{6.12}$$

where

$$r_i^* = \frac{r_i}{6\text{MAD}}, \tag{6.13}$$

MAD is the median of the $|r_i|$, and

$$r_i = y_i - \hat{g}(x_i). \tag{6.14}$$

The shape of the bisquare weight in Eq. (6.12) is shown also in Figure 6.2, which is steeper than the tricube weight function in Eq. (6.10). To estimate the new set of Robust LWR (RLWR) coefficients, the bisquare weight function is used, and the following function is minimized:

$$\sum_{i=1}^{n} B(r_i^*) w_i(x)(y_i - g(x_i))^2. \tag{6.15}$$

The newly estimated coefficients are used to obtain a new set of fitted values for $\hat{g}(x_i)$. This robustness steps are repeated until the values of the estimated coefficients converge. In this chapter, we repeat the robustness step two times, and the results for similar data experiments show that two iterations is satisfactory for getting the final fit. We name the method Robust Locally Weighted Least Squares (RLWLS). The reader is referred to Cleveland (1979); Cleveland and Devlin (1988); Cleveland and Grosse (1991) for more details about the use and advantages of weight functions, choosing criteria for weight functions, iteration time and overall RLWR.

To explore the fitting process for LS based LWR, simply Locally Weighted Least Squares (LWLS) and RLWLS, we generate a 2D dataset of 361 points including one outlier (green) as shown in Figure 6.3. The regular points are generated in a similar way to Moran (1984) having the following relationship:

$$y = \begin{cases} 0.4x + \epsilon & 0 \leq x \leq 10 \\ 3 + 0.1x + \epsilon & 11 \leq x \leq 30 \\ 12.6 - 0.267x + \epsilon & 26 \leq x \leq 45 \\ 0.5 + \epsilon & 44 \leq x \leq 70 \\ 1.5 + 0.05x + \epsilon & 65 \leq x \leq 100, \end{cases} \tag{6.16}$$

where ϵ follows Gaussian normal distribution with mean 0.0 and StD 1.0. We generate 60 points for each of the first four functions and 120 points for the fifth function in Eq.(6.16). The outlier point is in position $x = 15$ and $y = 12$. Fitted lines shown in Figure 6.3 for LWLS and RLWLS are drawn in blue and red respectively. In Figure 6.3 the interest point (green, which is an outlier) is fitted locally with its neighbours (points within the vertical dot lines), the blue and the red points are the LWLS and RLWLS fits, respectively. The RLWLS fit is closer than the LWLS fit to the majority points of the local neighbourhood and follows the direction of the majority points.

In the second type of approach, robust regression is employed to get the robust fit for all the points in the data. Therefore, we use LMS or LTS regression as the

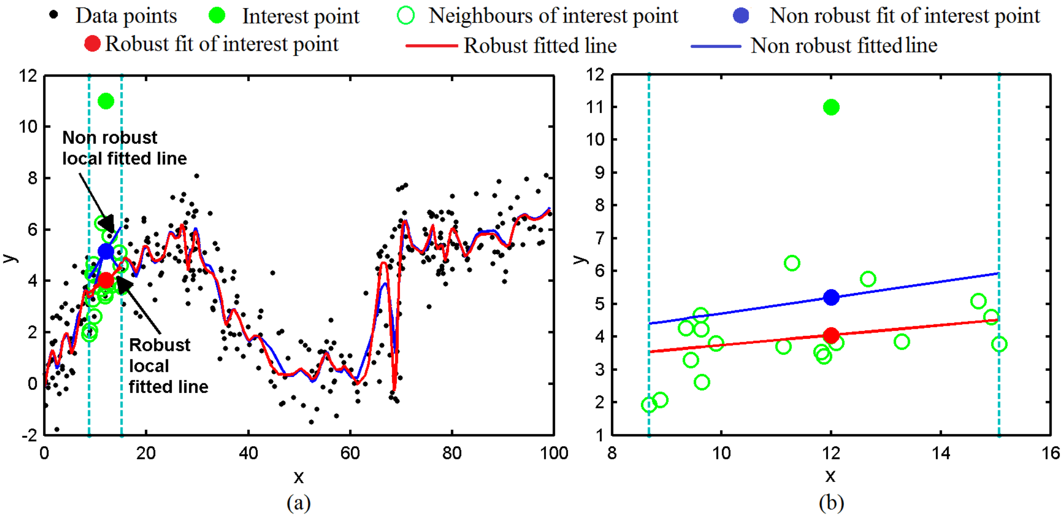


Figure 6.3 Locally weighted regression: (a) fitting for the whole dataset, and (b) fitting only for the interest point (green dot) with its local neighbourhood.

alternative to LS for the neighbourhood of each point in the data. That means, Locally Weighted LMS (LWLMS) and Locally Weighted LTS (LWLTS) robust regression can be performed as follows:

$$\underset{\hat{\beta}}{minimize}\ \underset{i}{median}\ w_i(x)r_i^2, \tag{6.17}$$

and

$$\underset{\hat{\beta}}{minimize} \sum_{i=1}^{h} w_i(x)(r_i^2)_{i:n}, \tag{6.18}$$

respectively, where in Eq. (6.18), $r_1^2 \leq \cdots \leq r_{n/2}^2 \leq \cdots \leq r_h^2 \leq \cdots \leq r_n^2$ are the ordered squared residuals.

To see the fitting performance of RLWLS, LWLMS and LWLTS for a dataset of unspecified and unknown pattern, we create an artificial dataset shown in Figure 6.4a. The dataset consists of 120 regular points, which follow 12 different local linear models, with each of the models consisting of a local group of 10 points. The 12 subsets or local groups of data points in 2D have the following mathematical relationship:

$$y = \beta_0 + \beta_1 x, \tag{6.19}$$

where β_0 and β_1 are fixed for each individual dataset but different to each other, and the x variable follows a Uniform distribution within a certain interval. To make the dataset noisy, we deliberately put another set of 120 points that follow a Gaussian normal distribution with mean 0, and StD 0.1 w.r.t. each and every individual regular point. The noisy dataset of 240 points is shown in Figure 6.4b. We fit the LS line for the regular dataset of 120 points. Figure 6.4c shows the linear model (blue) is not representative of the whole dataset and does not represent the real line (black) through the points. So we need to fit the data locally to extract the underlying pattern of the real data. In the case of noisy data in Figure 6.4d, we see the linear patterns within the small regions are now difficult to observe.

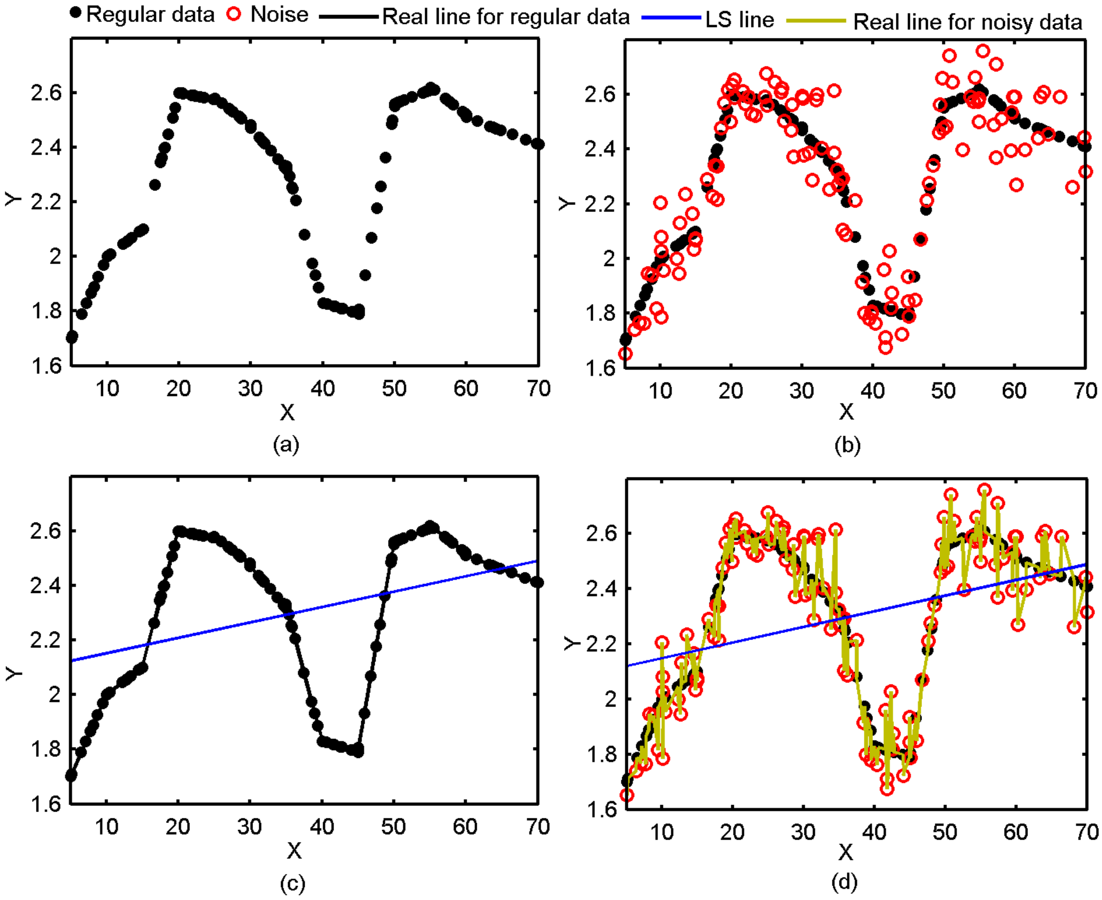
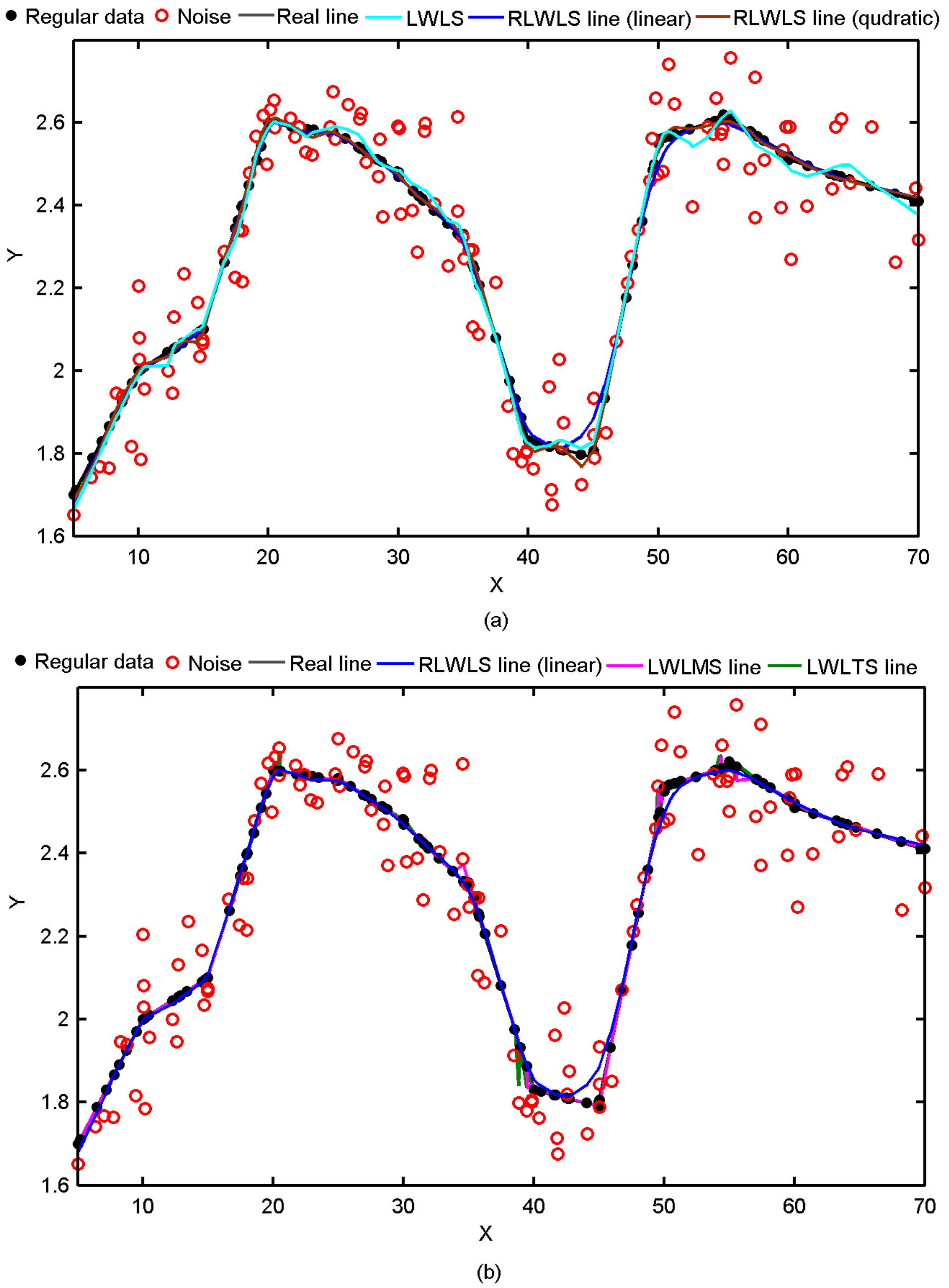


Figure 6.4 (a) Simulated data of 120 points, (b) simulated data with 120 noise points (red) added, (c) real line (black) and LS fitted line (blue) for simulated data, and (d) real line (olive) and fitted line (blue) for simulated data with 120 noise points added.

Figure 6.5a shows the result of locally weighted regression, with LWLS (cyan line) fitting the pattern locally. The fit is not correct in many places as indicated by the differences w.r.t. the real (black) line. We use both linear and quadratic functions for RLWLS regression. RLWLS using the linear functions gives a smoother (blue) line than the RLWLS using quadratic functions (maroon). In Figure 6.5b, we combine the results for LWLMS (magenta) and LWLTS (green) with RLWLS (blue) and real (black). The results show that robust regression based methods: LWLMS and LWLTS perform almost similarly to RLWLS, and closely represent the real line without the added noise.



(a)

(b)

Figure 6.5 Simulated data of 120 points with 120 noise points added: (a) real line without noise, and fitted lines for LWLS, RLWLS (linear) and RLWLS (quadratic), and (b) real line without noise, and fitted lines: RLWLS (linear), LWLMS and LWLTS for the data with noise.

6.4.3 Implementation

The laser scanned point clouds considered in this chapter are acquired along transport corridors using vehicle mounted laser scanners. In such a case, the long dataset is typically sliced into manageable 'stripes' for processing and then the results are merged. Ground surfaces such as the road pavements and footpaths are usually considered as the lowest features locally. Ground points can be defined as the points on the lowest, smooth, nominally horizontal surface (Belton and Bae, 2010). Based on this important property our algorithm proceeds to find the lowest level of the respective local region for every point in a stripe. A local region or neighbourhood is defined for every point in a stripe. Searching the local neighbourhood for a given point in an unstructured point cloud is not trivial. We discussed in earlier chapters, two well-known local neighbourhood determination methods called Fixed Distance Neighbourhood (FDN) and k Nearest Neighbourhood (kNN). In Section 6.4.1 we discussed how a fixed size kNN or fixed bandwidth (distance) smoothing window related to kNN and FDN respectively, can be used for selecting the local neighbourhood. It is shown that point density variation may misrepresent the real shape of a surface. Hence we use kNN to avoid the problem of point density variation that occurs with FDN. The point density variation is an usual event because of the movement of the data acquisition vehicle and the scanner geometry. For each stripe of the data, the algorithm used here processes the two dimensional orthogonal profiles x-z and y-z. That is along the scanning path and perpendicular to the scanning path. Since x-z and y-z profiles have different slopes, using both the profiles can balance the ground label from the directions. The method is performed iteratively over two main steps as follows.

First, RLWR for each point with its local neighbours is used to get a robust nonlinear fit for the whole stripe. We can use linear or quadratic fitting for every local neighbourhood of size k. We use linear fitting assuming that, for a sufficiently small size of neighbourhood, linear fitting will be a good approximation to a non-linear or polynomial fit. We can use RLWLS or LWLMS or LWLTS for robust polynomial fitting to the stripe x-z and y-z. The second step combines the sequence of four tasks as follows.

- **Task 1.** Calculation of residuals $r_i = z_i - \hat{z}_i$, where $\hat{z}_i$ is the fit of z_i.

- **Task 2.** Classification of points into two categories: points above the fitted RLWR (RLWLS/LWLMS/LWLTS) line and the points on or below the fitted RLWR line.

- **Task 3.** Use of bisquare robust weight function in Eq. (6.12) to down-weight the z-values of the points which are above the fitted line, while the rest of the points are given weight 1 (i.e. points on or beneath the fitted line will be unchanged). The reweighted z values will be considered as the new z values for the next fit. Figure 6.6 indicates the down-weighting process for the z_i point, which is necessary for the next fit. If, after using the bisquare weight function, the value of a fitted z becomes less than the lowest z-value of the corresponding neighbourhood, the new fitted z-value can be replaced by the lowest one to make it meaningful. This is possible as, in a local neighbourhood, the lowest features are generally regarded as the ground surface. However, if the local neighbourhood contains any 'low outlier' (Sithole and Vosselman, 2004) then we replace the outlier by the point that has the least z value among the inlier set. LWLMS and LWLTS can classify the points into inliers and outliers (see Rousseeuw and Leroy, 2003), and RLWLS has the ability to ignore the influence of low outliers when it uses robustification (i.e. use of bisquare weight function and iteration process).

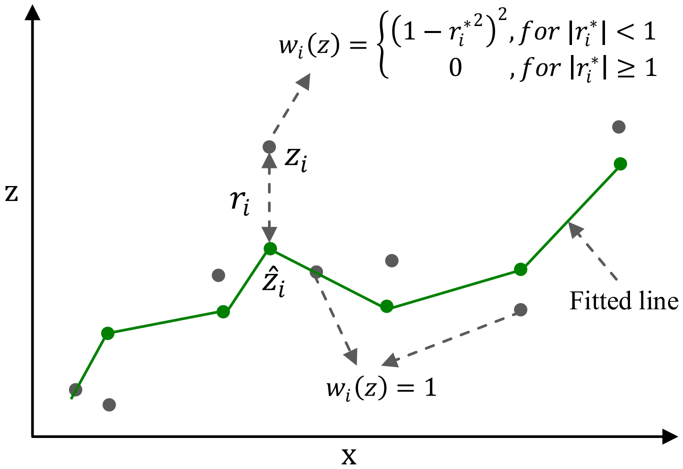

$w_i(z) = \begin{cases} (1 - r_i^{*2})^2, for\ |r_i^*| < 1 \\ 0 \quad , for\ |r_i^*| \geq 1 \end{cases}$
z_i
r_i
z
$\hat{z}_i$
Fitted line
$w_i(z) = 1$
x

Figure 6.6 Down weighting for the x-z stripe to get the z values for the next fit. Black points are real points, green dots are the respective fitted points and the green line represents the fitted line.

- **Task 4.** The new set of z-values is used to get the next RLWR fit. Tasks 1 to 3 will be continued until the difference Δ between the two Root Mean Squared Errors dRMSE from the two latest consecutive fitted polynomials is insignificant. We consider $\Delta = 0.005$ for our algorithms. The final RLWR fit is considered as the lowest or ground level fit for the current stripe and the points between a band created by the lowest level and lowest level ± a predefined threshold (based on similar data experiments) are considered as ground surface points from the profile. Finally, common points that are identified as ground points from the results of x-z and y-z profiles are classified as the ground points for the stripe. The threshold values for x-z and y-z may vary because the x and y axes measure different directions. For example, in the case of mobile mapping through road corridors, we may assume the y axis is the horizontal direction along the road and the x axis is in the horizontal direction across the road. Therefore, the thresholds for an x-z stripe depend on the difference between the points from the two opposite sides of the road and the threshold for y-z depends on the difference between the points of the two most distant positions on the road. Hence a smaller stripe has the advantage of enabling the fixing of the threshold values easily and accurately.

The ground surface extraction process is shown in Figure 6.7, and the robust classification method for point cloud data into ground and non-ground points is summarized for $x - z$ profiles in Algorithm 6.1, and the algorithm will be performed in the same fashion for $y - z$ profiles.

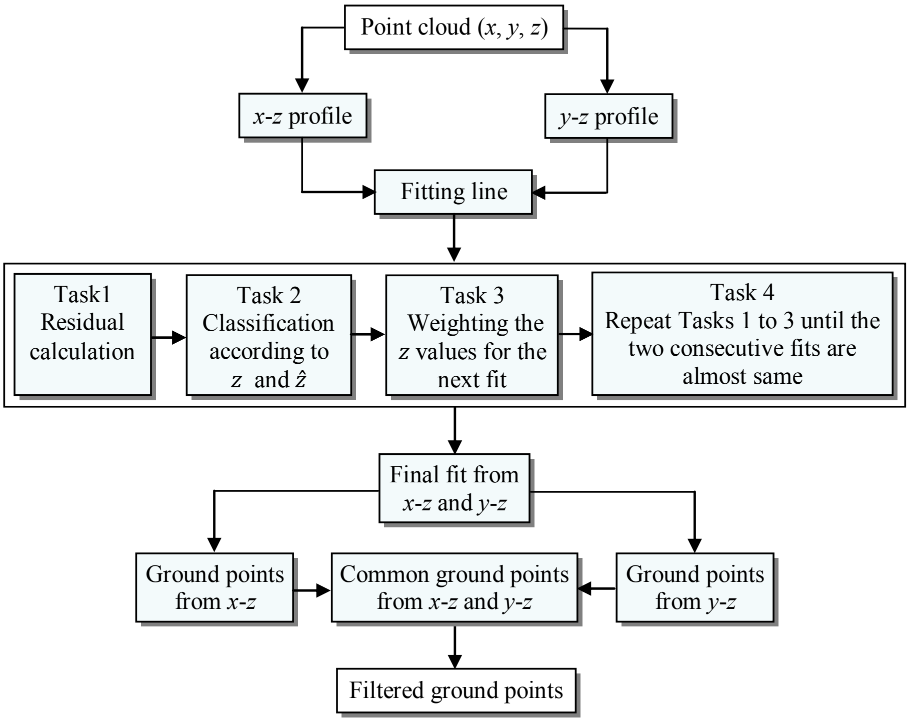

Point cloud (x, y, z)
x-z profile
y-z profile
Fitting line
Task1 Residual calculation
Task 2 Classification according to z and ẑ
Task 3 Weighting the z values for the next fit
Task 4 Repeat Tasks 1 to 3 until the two consecutive fits are almost same
Final fit from x-z and y-z
Ground points from x-z
Common ground points from x-z and y-z
Ground points from y-z
Filtered ground points

Figure 6.7 Robust ground surface extraction process.

Algorithm 6.1: Ground Surface Extraction

```
Input:
        $P$: Point cloud $P(x, y, z)$
        $k$: Neighbourhood size
        $\Delta$: Threshold for difference between consecutive RMSE (dRMSE)
        $\delta$: Threshold added to the lowest level or final fit to get ground surface points
Output:
        $g$: Ground surface points
        $ng$: Non-ground surface points
1.  for $i$=1 to size($P$) do
2.      Find $k$ nearest neighbourhood $Np_i$ of $p_i$ in $x$-direction
3.      Fit locally weighted regression in the $Np_i$
4.  end for
5.  Find residuals $r_i \leftarrow z_i - \hat{z}_i$
6.  Calculate bisquare weight, $w_i$ using Eq. (6.12)
7.  if $z_i$ larger than $\hat{z}$ then
8.      $z_i \leftarrow z_i * w_i$
9.  end if
10. if weighted $z_i < min(z_i)$ of $Np_i$ then
11.     $z_i \leftarrow min(z_i)$ of $Np_i$
12. end if
13. Repeat Step 3 to Step 12 until $|\text{RMSE}_i - \text{RMSE}_{i-1}| < \Delta$
14. if $z_i \leq \hat{z}_i + \delta$ then
15.     $g \leftarrow p_i$
16. else
17.     $ng \leftarrow p_i$
18. end if
```

6.5 Experiments and Evaluation

In this section, the proposed algorithm is demonstrated and evaluated through experiments on five real MLS datasets. The datasets were captured in the same way as the data used in the previous chapters. We consider datasets consisting of different types of complex objects on and close to the road in urban areas. We assess the results visually and compare them with those for the robust segmentation Algorithm 5.4 proposed in Chapter 5.

Quantitative measurement: To measure the quantitative performance of the proposed filtering Algorithm 6.1, we calculate the measures: Type I error, Type II error, total error and accuracy rate. Following the rules in Sithole and Vosselman (2004) the measures are defined as:

$$\text{Type I error} = \frac{b}{a+b}, \tag{6.20}$$

$$\text{Type II error} = \frac{c}{c+d}, \tag{6.21}$$

$$\text{Total error} = \frac{b+c}{e}, \tag{6.22}$$

$$\text{Accuracy} = \frac{a+d}{e}, \tag{6.23}$$

where a: number of ground points correctly identified as ground points, b: number of ground points incorrectly identified as non-ground points, c: number of non-ground points incorrectly identified as ground points, d: number of non-ground points correctly identified as non-ground points, e = total number of data points. We calculate the measures and compare with the segmentation results using Algorithm 5.4 proposed in Chapter 5. We also calculate the number of ground g and non-ground ng points extracted from every method.

Dataset 6.1: Tree-pole-wall dataset

Our first dataset consists of 17,696 points, involves a tree, a light pole, part of a road side wall, part of a roof that overlaps the tree, and road surfaces. We name this dataset the 'tree-pole-wall' dataset. The dataset is shown in Figure 6.8a. We perform three proposed robust (RLWLS, LWLMS and LWLTS) ground surface extraction methods for the two bi-dimensional x-z and y-z profiles of the dataset. We use LWR with the above mentioned tricube weight function for every point in the dataset with their respective local neighbourhood of size 200. We fix the neighbourhood size based on knowledge about the data density and from earlier experiments on similar data. We fit and calculate residuals $r = z - \hat{z}$, and perform the down-weighting based on the bisquare robust weight function to reduce the influences of extremely high off-terrain points. The iteration process of fitting and down-weighting continues until the difference between two Root Mean Squared Errors (dRMSE) from two consecutive fits is insignificant or almost zero. In this chapter we stop iterating when we get the dRMSE to be less than 0.005. The results in Figures 6.8(b and c) show that using RLWLS after 6 iterations we get the ground level (magenta line) for both x-z and y-z profiles. We also perform LWLMS and LWLTS for the dataset accordingly. Figures 6.8(d and e) and 6.8(f and g) show the results of fitting using LWLMS and LWLTS respectively. The fitted lines from consecutive iterations are shown in different colours. For example in Figures 6.8a, the Iterations 1, 2, 3 and 6 are shown in red, yellow, green and magenta respectively. Iteration numbers for the respective methods are given in Column 12 in Table 6.2.

After getting the ground level from the respective methods, we add a threshold value δ to the z of the ground level. Points within 0.25m and 0.35m vertical distance or z from the ground level for x-z and y-z profiles respectively are treated as ground surface points. Figures 6.9(a and b), 6.9(c and d) and 6.9(e and f) show the results of the classified ground (grey colour) and non-ground points from x-z and y-z profiles for RLWLS, LWLMS and LWLTS respectively. The common ground points (grey colour) from x-z and y-z profiles from the three robust methods: RLWLS, LWLMS and LWLTS are shown in Figures 6.10(a, b and c), which are the final results for ground surface extraction from the respective methods.

We compare the results with the segmentation results from our proposed Algorithm 5.4 in Chapter 5. We set the required parameters for the segmentation algorithms: neighbourhood size $k = 30$, angle threshold $\theta_{th} = 15°$, and minimum region size $R_{min} = 2$. Segmentation results are portrayed in Figure 6.10d. We calculate ground g and non-ground ng points for the different methods and count the points that match with the results from the segmentation algorithm. We consider road kerb and footpath with the ground surface for the segmentation algorithm. The performance measures: accuracy (Acc) and errors rates, are calculated by using Eqs. (6.20), (6.21), (6.22) and (6.23) based on the segmentation and concern ground surface extraction results. The results for segmentation (Seg.) and ground surface extraction are shown in Table 6.2. We see the results are similar for all three robust methods and they have very low error rates of less than 1% and accuracy rates of approximately 97%.

We calculate the time to perform the three methods using the MATLAB® profile

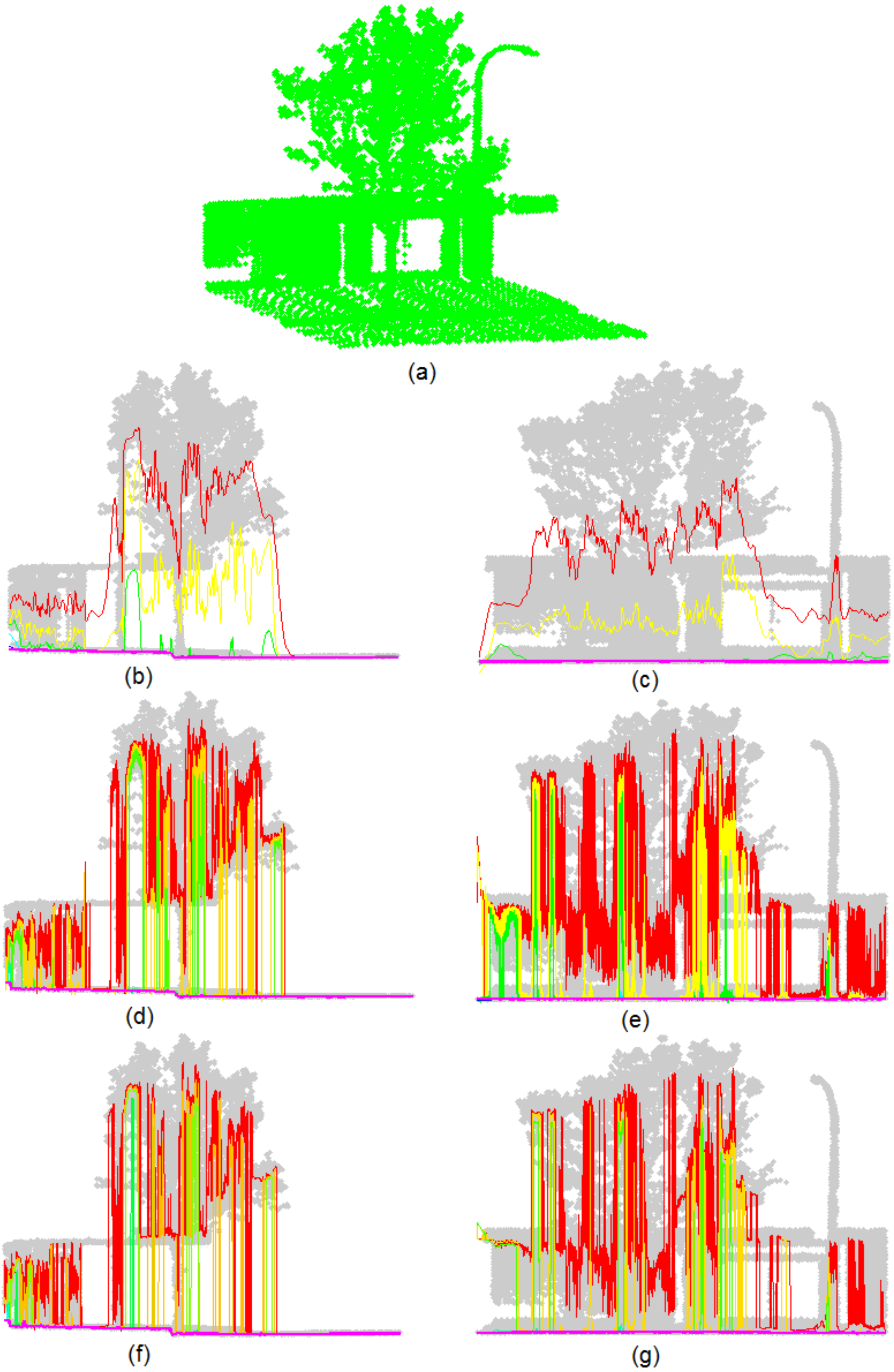


Figure 6.8 (a) Tree-pole-wall dataset, iterative fittings for x-z and y-z profiles: (b) RLWLS; x-z, (c) RLWLS; y-z, (d) LWLMS; x-z, (e) LWLMS; y-z, (f) LWLTS; x-z, and (g) LWLTS; y-z.

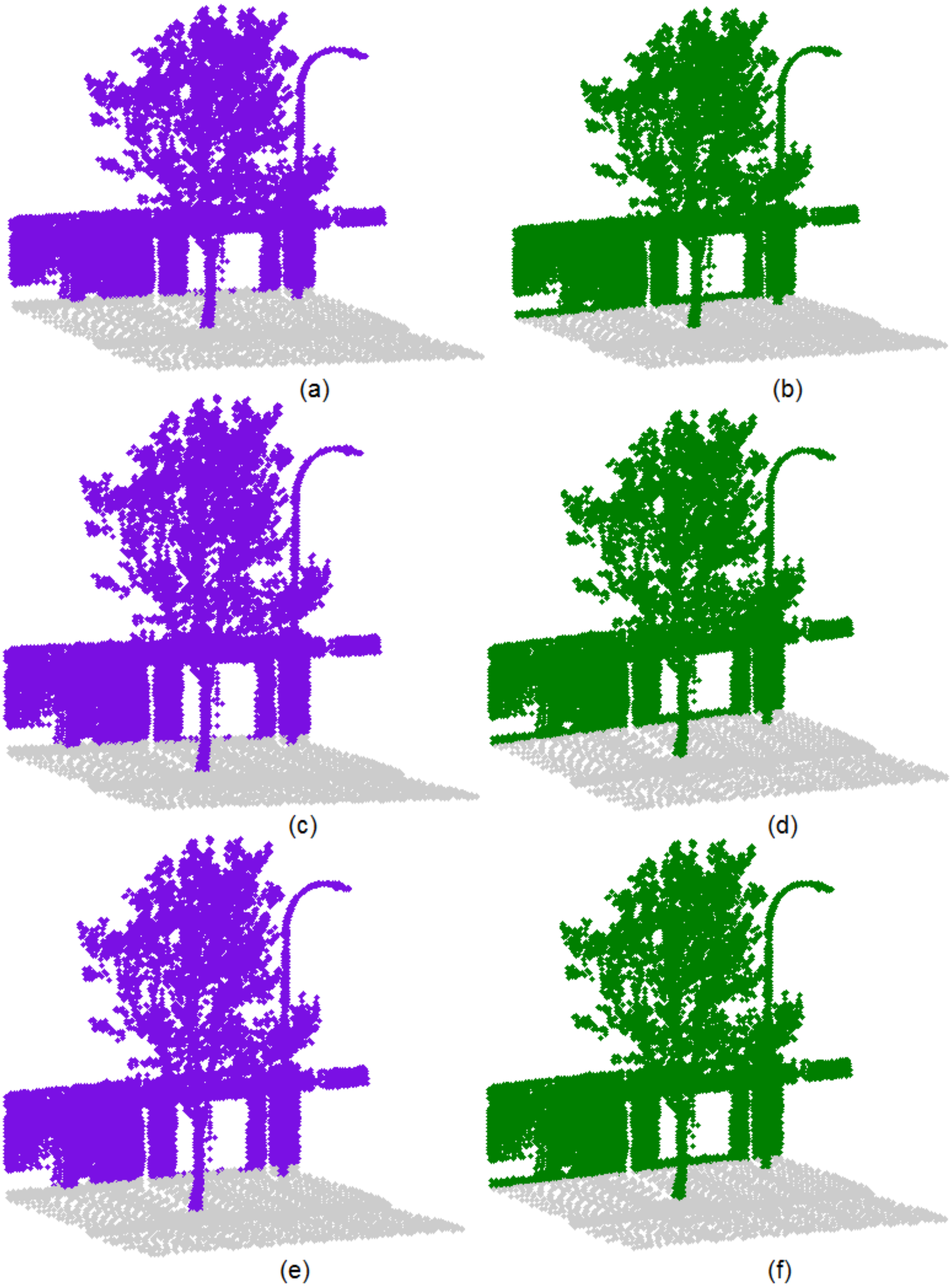


Figure 6.9 Ground surface extraction for the tree-pole-wall dataset from x-z and y-z profiles: (a) RLWLS; x-z, (b) RLWLS; y-z, (c) LWLMS; x-z, (d) LWLMS; y-z, (e) LWLTS; x-z, and (f) LWLTS; y-z.

function. Time in seconds (s) for the respective ground surface extraction methods are given in the last column of Table 6.2. Results show the RLWLS, LWLMS and LWLTS take 116.64s, 439.53s and 1740.10s respectively, which reveals RLWLS takes significantly less time than LWLMS and LWLTS without a reduction in the quality of the results.

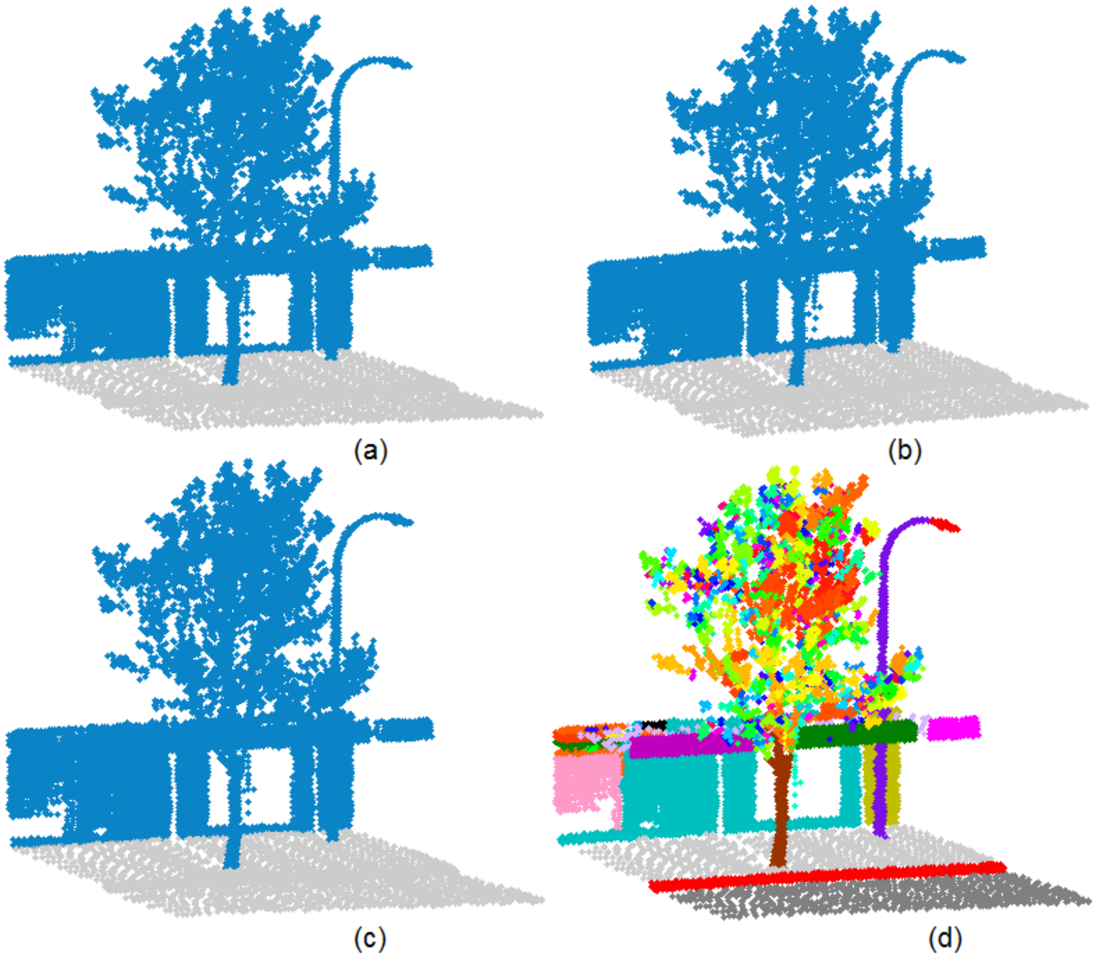


Figure 6.10 Ground surface extraction for the tree-pole-wall dataset: (a) RLWLS, (b) LWLMS, (c) LWLTS, and (d) segmentation result.

Table 6.2 Filtering accuracy measures for the tree-pole-wall dataset.

Methods	g	ng	a	b	c	d	Type I error (%)	Type II error (%)	Total error (%)	Acc (%)	No. of iteration (x-z, y-z)	Time (s)
Seg.	2307	14949										
RLWLS	2454	15242	2306	1	122	14827	0.043	0.82	0.69	96.82	6, 6	116.64
LWLMS	2457	15239	2306	1	125	14824	0.043	0.84	0.71	96.80	7, 6	439.53
LWLTS	2458	15238	2306	1	126	14823	0.043	0.84	0.72	96.79	8, 7	1740.10

Dataset 6.2: Traffic signal dataset

Meng et al. (2010) pointed out that errors are mainly found in difficult to recognize low height features such as bushes, short walls, and on the boundaries of the ground and non-ground objects. It is also more difficult to identify ground points in an area covered by dense urban features, such as power poles, flags and cars. Our 2^{nd} dataset shown in Figure 6.11a, consists of 59,523 points that have been taken in such an urban area where a short wall, a car, a power pole, two small road barriers and traffic signals are present. We name this dataset as the 'traffic signal' dataset.

We run RLWLS, LWLMS and LWLTS for the two x-z and y-z profiles. We use locally weighted regression using the tricube weight function for every point w.r.t. their local neighbourhood of size 200. We calculate residuals and perform the down-weighting based on the bisquare robust weight function as for the previous experiment to reduce the influences of extremely high non-ground points. The iteration process terminates when dRMSE < 0.005. We find ground surface points below 0.35m and 1.35m from the estimated ground level for x-z and y-z profiles respectively. The final ground points (grey colour), which are the common points for the x-z and y-z profiles, are plotted in Figures 6.11(b, c and d) for RLWR, LWLMS and LWLTS, respectively.

Now, we compare the ground points filtering results with the segmentation results in Figures 6.11e obtained by using the segmentation algorithm: Algorithm 5.4. For the segmentation algorithm, we set the parameters: $k = 30$, $\theta_{th} = 10°$, and $R_{min} = 2$. The error and accuracy measures in Eqs. (6.20), (6.21), (6.22) and (6.23) are calculated by comparing with the segmentation results by counting the ground and non-ground points. RLWLS, LWLMS and LWLTS perform almost the same with their overall accuracy rates of 98.22%, 98.24% and 98.21% respectively. The results in Table 6.3 show that RLWLS needs less iterations and time than LWLMS and LWLTS to reach the ground level.

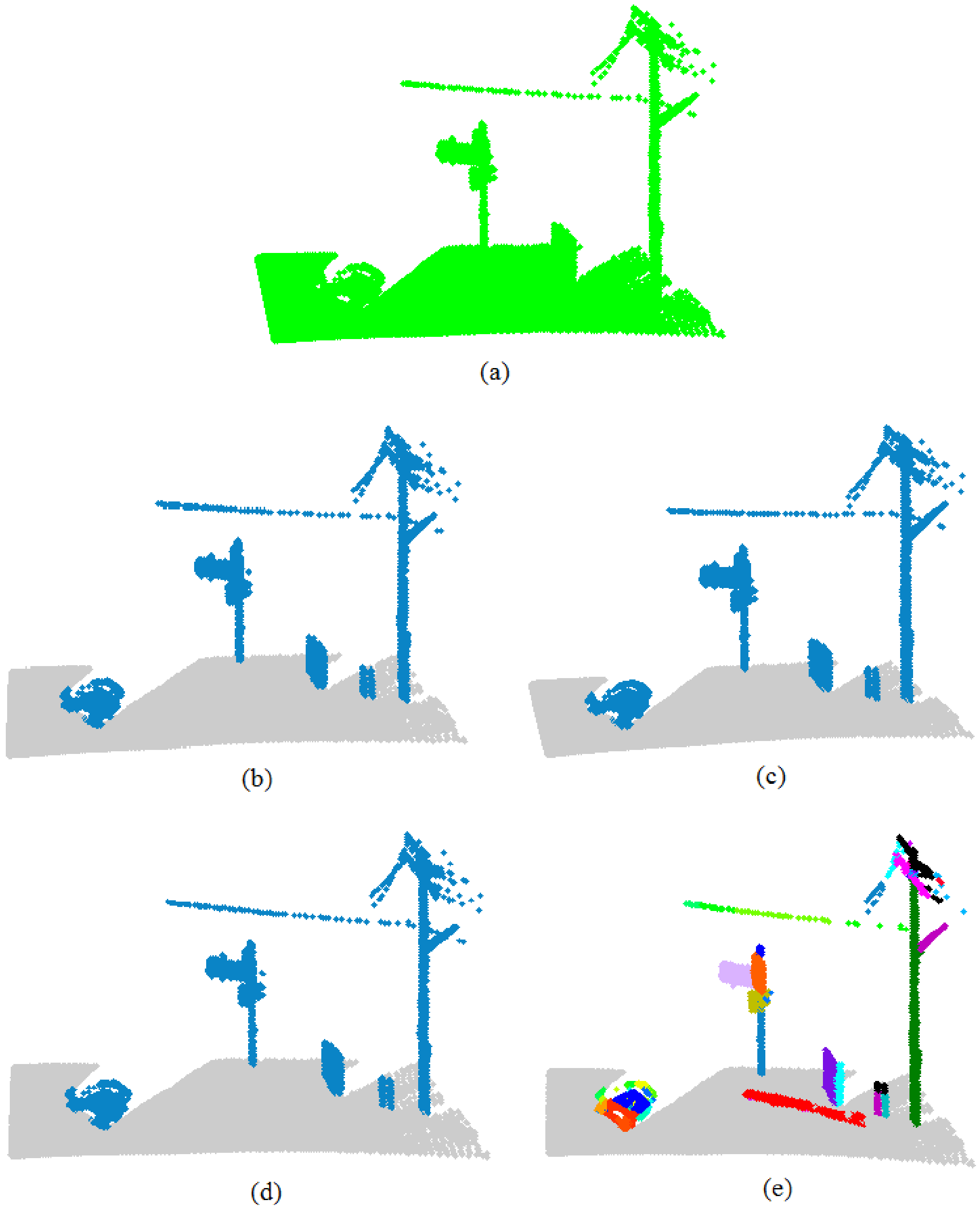


Figure 6.11 Ground surface extraction for the traffic signal dataset: (a) dataset. Ground surface extraction results: (b) RLWLS, (c) LWLMS, (d) LWLTS; and (e) segmentation result.

Table 6.3 Filtering accuracy measures for the traffic signal dataset.

Methods	g	ng	a	b	c	d	Type I error (%)	Type II error (%)	Total error (%)	Acc (%)	No. of iteration (x-z, y-z)	Time (s)
Seg.	53953	5425										
RLWLS	54188	5335	53459	494	423	5002	0.009	7.8	1.54	98.22	4, 4	505.80
LWLMS	54184	5339	53468	485	420	5005	0.009	7.7	1.52	98.24	5, 6	1188.83
LWLTS	54203	5320	53468	485	437	4988	0.009	8.1	1.55	98.21	5, 3	1953.73

Dataset 6.3: Road furniture dataset

We consider our 3^{rd} dataset shown in Figure 6.12a consisting of 39,234 points that contains mainly road side furniture including a big billboard, bus shelter, cylindrical and planar surfaces (sign on a pole). It also contains part of a road, kerb and footpath. We label the dataset as the 'road furniture' dataset.

We run the algorithms: RLWLS, LWLMS and LWLTS for the two x-z and y-z profiles. We fit locally weighted regression, and perform the algorithms by using tricube weight as the local weight and bisquare weight functions for down weighting the extremely high z values for every point with neighbourhood size 200. The iteration process terminates at dRMSE < 0.005. Ground surface points were found below 0.30m and 0.66m from the estimated ground level for the x-z and y-z profiles respectively. The common points for the x-z and y-z profiles, i.e. final ground points (grey colour) for RLWR, LWLMS and LWLTS are plotted in Figures 6.12(b, c and d) respectively.

We run segmentation Algorithm 5.4 to evaluate the ground surface extraction methods. The segmentation results obtained with the parameters: $k = 30$, $\theta_{th} = 10°$, and $R_{min} = 2$ are plotted in Figure 6.12e. The accuracy and the errors measures are calculated by comparing with the segmentation results. The results are in Table 6.4. Table 6.4 shows the proposed algorithms have no Type-I error for the dataset, and total errors are only 0.73%, 0.74% and 0.75% for RLWLS, LWLMS and LWLTS respectively, with more than 95% accuracy for ground surface extraction for all methods. The times required for performing RLWLS, LWLMS and LWLTS are 442.83s, 639.97s and 2231.43s respectively.

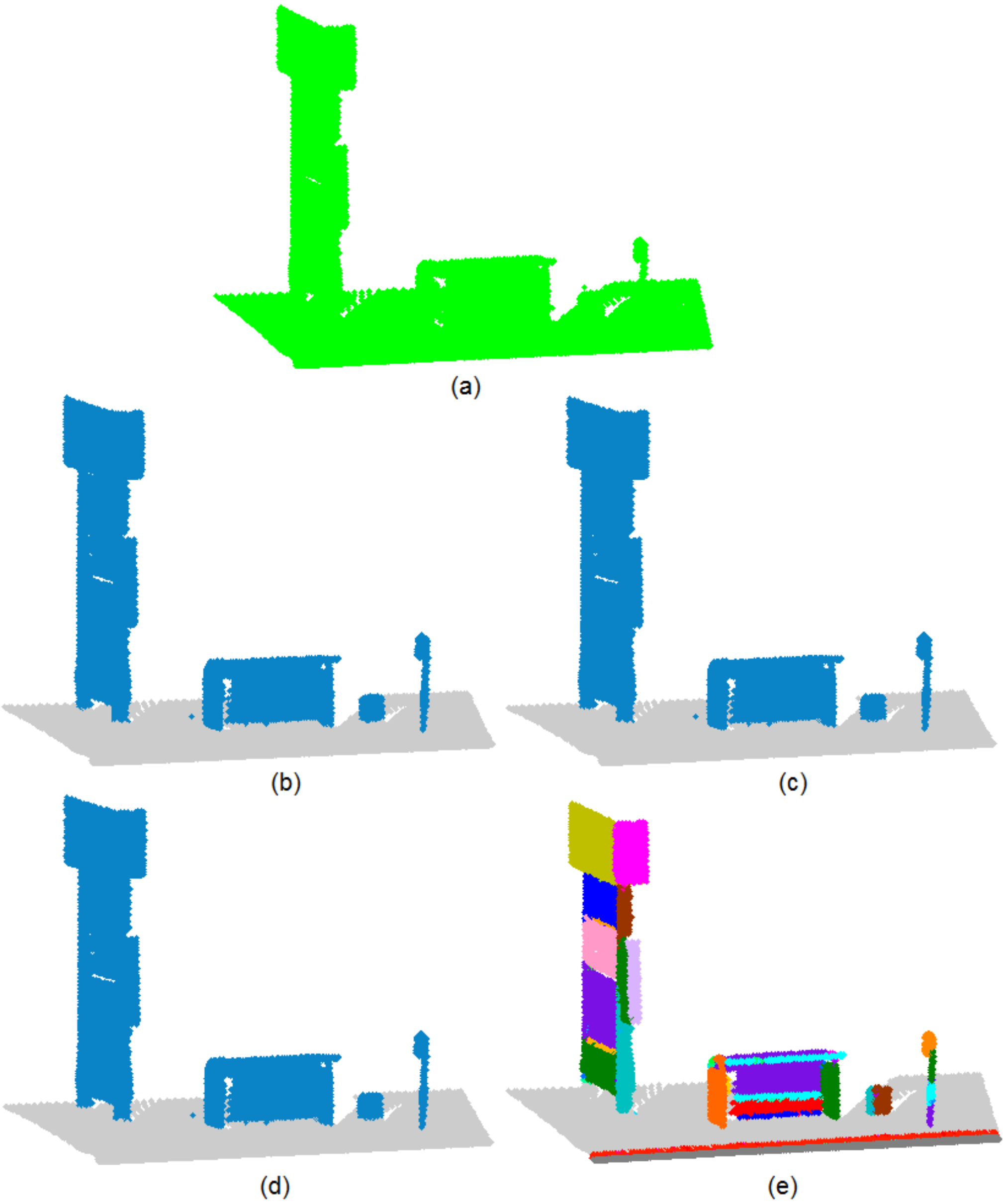


Figure 6.12 Ground surface extraction for the road-furniture dataset: (a) dataset. Ground surface extraction results: (b) RLWLS, (c) LWLMS, (d) LWLTS; and (e) segmentation result.

Table 6.4 Filtering accuracy measures for the road-furniture dataset.

Methods	g	ng	a	b	c	d	Type I error (%)	Type II error (%)	Total error (%)	Acc (%)	No. of iteration (x-z, y-z)	Time (s)
Seg.	24989	12686										
RLWLS	26130	13104	24989	0	287	12399	0	2.26	0.73	95.29	5, 7	442.83
LWLMS	26133	13101	24989	0	290	12396	0	2.29	0.74	95.29	6, 5	639.97
LWLTS	26136	13098	24989	0	293	12393	0	2.31	0.75	95.28	6, 6	2231.43

Dataset 6.4: Tree-wall dataset

Presence of low outliers in the data: We now evaluate the performance of the proposed algorithm for ground surface extraction in the presence of low outliers. The points that normally do not belong to the landscape and have originated from multi-path errors and errors in the laser range finder are treated as low outliers (Sithole and Vosselman, 2004). The problem with the low outliers is that most of the filtering algorithms assume that the lowest points belong to the terrain (Sithole and Vosselman, 2004; Belton and Bae, 2010). Although the LS method cannot find outliers, robust methods (RLWLS, LWLMS and LWLTS) have the opportunity to identify outliers.

In this section, for our analysis we take a part of the tree-pole-wall dataset consisting of 9,373 points that includes a tree, a wall and the ground surface. We name this dataset the 'tree-wall' dataset. We create six artificial low outliers shown as red asterisks in Figure 6.13a. We run non-robust LWLS and robust RLWLS methods. We observe RLWLS is significantly faster than LWLMS and LWLTS, so we just consider RLWLS as the representative of the robust methods.

We run LWLS and RLWLS algorithms for the two bi-dimensional x-z and y-z profiles of the dataset. The algorithms use the previously used weight functions and a local neighbourhood size of 100. The iteratively fitted lines for the x-z and y-z profiles for LWLS and RLWLS are shown in Figures 6.13(b and c) and 6.13(d and e) respectively. Fitted lines in Figures 6.13(b and c) show that they are not free from low outlier effects. Besides, Figure 6.13f shows that the ground levels extracted by LWLS are influenced by outliers but RLWLS (Figure 6.13g) was able to ignore the outliers and properly determine the ground levels. Hence, the final ground surface extracted with RLWLS is free from non-ground points, whereas, in Figure 6.13f many ground points are identified as off-ground points (blue). We also perform our proposed segmentation algorithm with $k = 30$, $\theta_{th} = 10°$, and $R_{min} = 2$. Results for RLWLS and segmentation in Figures 6.13(g and h) respectively are almost the same. Table 6.5 shows the accuracy rate for RLWLS is 99.03% and for LWLS is 93.65%, which certainly proves that RLWLS extracts the ground surface properly even in the presence of low outliers in the data.

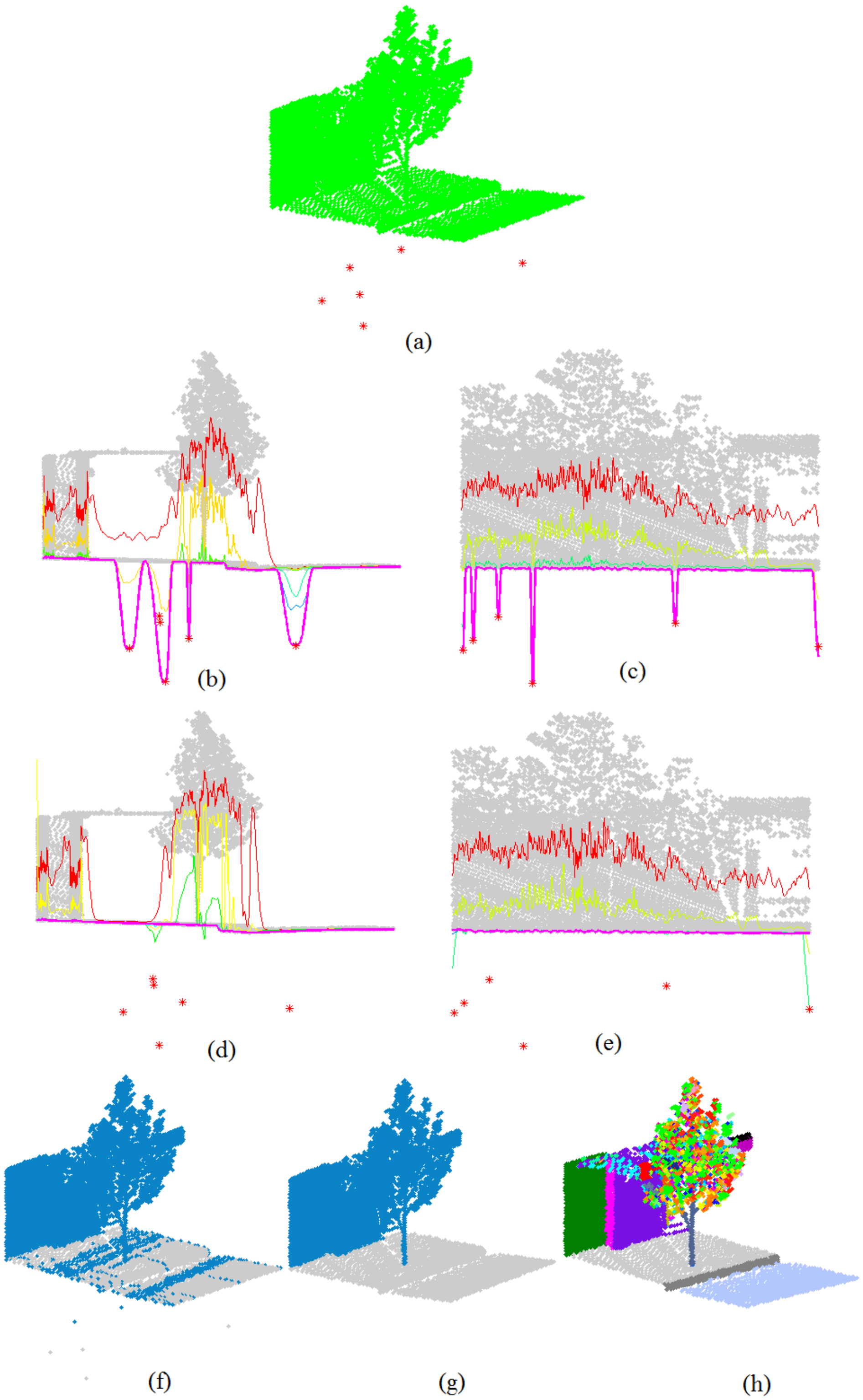


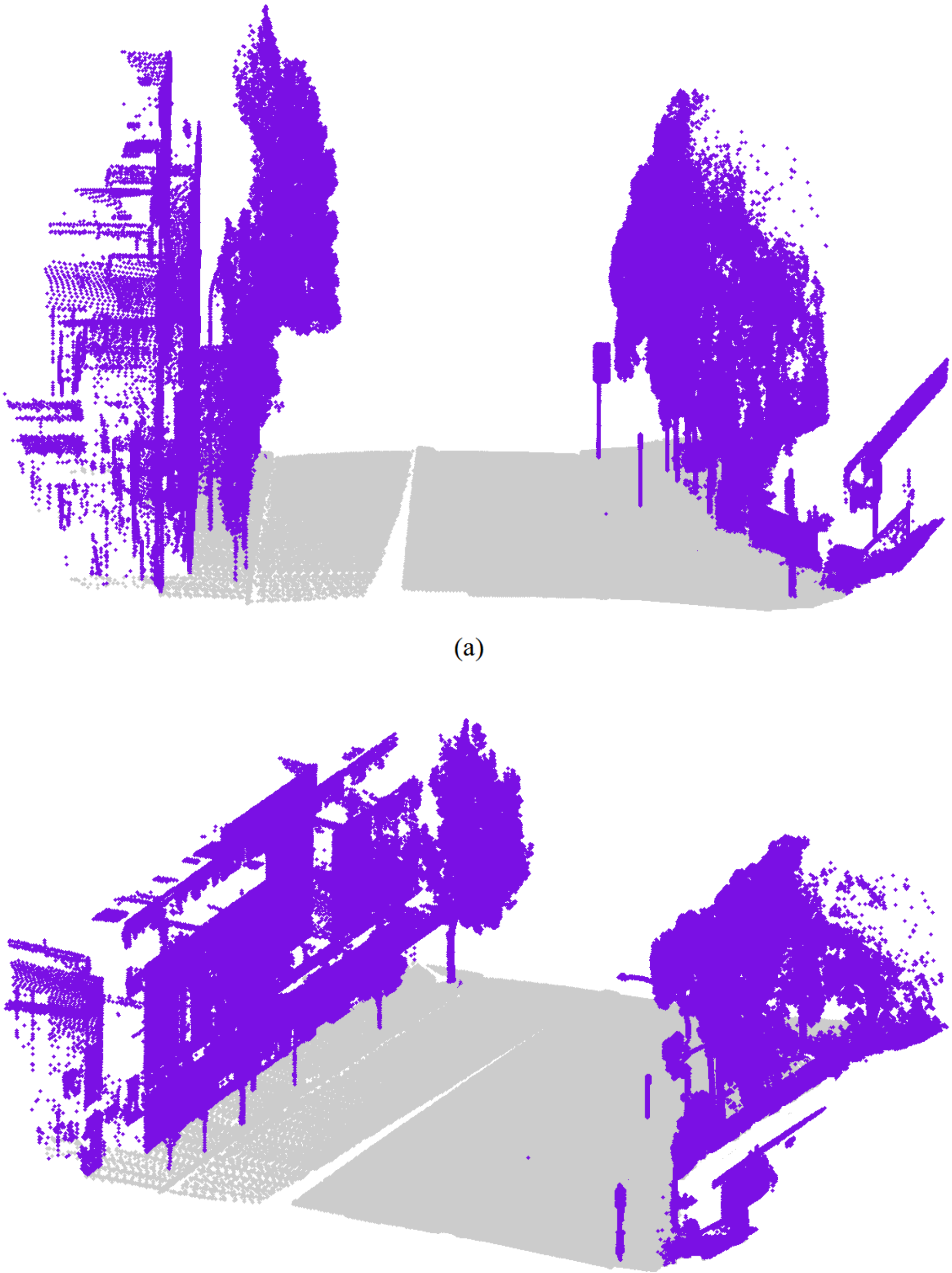
Figure 6.13 Ground surface extraction in the presence of low outliers: (a) real data with low outliers (red points), (b) iterative fittings using LWLS for x-z profile, (c) iterative fittings using LWLS for y-z profile, (d) iterative fittings using RLWLS for x-z profile, (e) iterative fittings using RLWLS for y-z profile, (f) results for LWLS, (g) results for RLWLS, and (h) segmentation results.

Table 6.5 Filtering accuracy measures for the tree-wall dataset with outliers.

Methods	g	ng	a	b	c	d	Type I error (%)	Type II error (%)	Total error (%)	Acc (%)	No. of iteration (x-z, y-z)	Time (s)
Segmentation	1875	7498										
LWLS	1459	7920	1370	505	85	7413	26.93	1.13	6.29	93.65	7, 5	25.34
RLWLS	1960	7413	1875	0	85	7413	0	1.13	0.91	99.03	6, 5	59.38

Dataset 6.5: Road corridor dataset

Now we apply the proposed robust ground surface extraction algorithm to a large MLS dataset of 1,060,300 points and covers about 40m road area. The dataset is shown in Figure 6.14 and consists of large trees, buildings, small walls, fence, signposts, light poles and different types of complex objects (e.g. combination of toroidal, long cylindrical and approximately cylindrical objects). We break the data into 10 slices along the y-axis. We run the ground surface extraction algorithm RLWLS for both the x-z and y-z profiles. We use the same parameters as for the previous experiments. The final ground surface points are the common ground points of the two different profiles x-z and y-z. Figures 6.14(a and b) are the front and side views of the results respectively. The results show that the proposed RLWLS based robust method efficiently classifies ground (grey colour) and non-ground (blue colour) surface points in areas covered by dense urban features.



(a)

(b)

Figure 6.14 Ground surface extraction for the road-corridor dataset: (a) front view, and (b) side view.

6.6 Conclusions

In this chapter three robust locally weighted regression (RLWR) based variants of ground surface extraction methods: RLWLS, LWLMS and LWLTS are proposed. The locally weighted regression based statistically robust approach can extract ground surfaces in urban areas. Most urban features such as complex large buildings, large trees, short walls, sign posts, power poles, traffic signals, vehicles are efficiently separated from the ground surfaces. Although the method is an iterative process using local weights, it runs with a very low number of iterations that minimizes the computation time. The method is fast and applying the proposed ground surface extraction technique means post-processing tasks that only operate on non-ground or ground data e.g. tree finding, only need to operate on part of the data. Moreover, our algorithm depends on only a few parameters. In addition, a major advantage of the new methods is that they can efficiently handle the presence of low outliers. We have observed all three robust methods: RLWLS, LWLMS and LWLTS are similar in terms of accuracy, but RLWLS takes significantly less time than LWLMS and LWLTS. Quality assessment based on comparing with the proposed robust segmentation algorithm gives more than 95% correct classification rate of the ground and non-ground surface points. When considering the determination of the ground points, the majority of non-ground points can be excluded without filtering out points belonging to small vertical surfaces like road kerbs.

The next chapter will cover concluding remarks of all the major Chapters 3, 4, 5 and 6 in this thesis, conclude the whole thesis and will present some suggestions for future research.

Chapter 7

"True knowledge exists in knowing that you know nothing."
Socrates

"A conclusion can help us to stop to take a rest
but does not allow us to decide this is the end."
Abdul Nurunnabi

Conclusions and Future Works

In this thesis we have addressed several aspects of feature extraction in laser scanning point cloud data. Focusing on mobile laser scanning data, we were able to develop and contribute automatic and semi-automatic robust feature extraction methods for robust planar surface fitting, outlier detection, point cloud denoising, robust saliency features estimation, robust segmentation and robust ground surface extraction. The following are the concluding remarks based on the achievements mentioned in this thesis.

7.1 Achievements

7.1.1 Robust Planar Surface Fitting

In the real world, the plane is a major component of the most commonly found man made objects. Plane fitting and the resultant plane parameters are essential for point-based representations in many disciplines such as computer aided design, computer graphics, computer vision, reverse engineering, robotics, photogrammetry and remote sensing.

Using Fast-MCD and Deterministic MCD (DetMCD) based diagnostic PCA, robust PCA and diagnostic robust PCA algorithms, we were able to develop six variants of planar surface fitting algorithms in laser scanning 3D point cloud data. Several experiments using simulated and real mobile laser scanning datasets showed that the DetMCD techniques outperform classical methods (LS and PCA) and are more robust than RANSAC, MSAC and Fast-MCD based methods. The proposed methods give better results in terms of (i) higher percentage of outlier contamination tolerated, (ii) larger datasets, (iii) greater point density variation, and (iv) better classification of data into inliers and outliers. The proposed methods classify outliers and inliers accordingly and can reduce masking and swamping effects. Performing the Wilcoxon Signed Rank test showed that the newly proposed algorithms are significantly more accurate than non-robust methods: LS and PCA; and robust methods: RANSAC and MSAC. Using simulated data it was determined that the DetMCD based proposed algorithms: DetRD-PCA, DetRPCA and DetDRPCA are significantly faster than the Fast-MCD based methods: FRD-PCA, FRPCA and FDRPCA; and that RANSAC and MSAC are slower (especially for large datasets) than the proposed DetMCD methods. Results for plane fitting, sharp feature preservation/recovery and segmentation were more accurate and robust when the normals and curvatures from the proposed algorithms were used on MLS data (planar and non-planar incomplete complex object surfaces). Applying the methods to real data proved that using the robust and accurate normals and curvature values reduces over and/or under segmentation. Overall the proposed DetRD-PCA and DetRPCA produced results that are comparable. However, DetRD-PCA performs better than DetRPCA in the presence of low point density and a high percentage of outliers. As is the case for Fast-MCD and DetMCD, the proposed algorithms are not suitable when the dataset contains more than 50% outliers and/or noise.

7.1.2 Outlier Detection and Robust Saliency Features Estimation

The presence of outliers and noise is common in laser scanning data which means outlier detection methods and methods for robust saliency features estimation are needed for many point cloud processing tasks.

This thesis proposed two outlier detection methods applicable to laser scanning point cloud data. Robust and diagnostic statistical approaches coupled with PCA resulted in two methods for robust saliency feature (normals and curvature) estimation termed as MCMD_Z and MCMD_MD. In the proposed methods, first the best plane is fitted to the majority of consistent data within the local neighbourhood of each point of interest, and then the outliers are detected locally for every neighbourhood based on identifying the majority of consistent or homogeneous points. Later, the required saliency features (normals and curvatures) are estimated for every point by using PCA based on the inlier points found in their respective local neighbourhood. The proposed algorithms were demonstrated through real and simulated datasets. Results showed that the outlier detection methods: (i) are computationally simpler, (ii) are able to efficiently identify high percentages of clustered and uniform outliers, (iii) are able to denoise point clouds, and (iv) are significantly faster than Fast-MCD and Deterministic MCD based robust and diagnostic statistical methods, and existing computer vision, data mining, machine learning techniques such as RANSAC, MSAC, LOF, q_{S_p} and uLSIF. Estimated robust normals and curvatures were used for point cloud processing, and the results proved that based on the estimated saliency features (normals and curvatures) sharp features such as edges and corners could be recovered efficiently. The robust saliency features, based on the proposed techniques, were used for point cloud segmentation of planar and non-planar complex surfaces. It was shown that they were efficient and able to reduce over and under segmentation, and can produce more accurate and robust segmentation results than existing methods. In summary, the results showed that the newly proposed MCMD_Z and MCMD_MD methods are more accurate, faster and produce robust results for point cloud processing tasks where normals and curvature are used. The proposed methods based on MCMD_MD are able to deal with up to 75% of outliers in the data.

7.1.3 Robust Segmentation

Segmentation for grouping and labelling the homogeneous and spatially close points into different regions is an important task for point cloud processing such as surface reconstruction, object shape and geometry analysis, and feature extraction.

This thesis has devised two robust segmentation algorithms, one is for multiple planar surface extraction and the other is for planar and non-planar object surfaces. The first one is a hybrid technique that combines classification, region growing and merging (if necessary). Initially points are classified into edge/corner points and surface points and then region growing is used to group points while excluding the edge/corner points. Finally similar and spatially close regions are merged. The second segmentation algorithm solely depends on the region growing approach. These methods employ robust and diagnostic statistics and are coupled with PCA to get robust saliency features. The proposed RDPCA algorithm uses robust saliency features (normals and curvatures) that are estimated by the proposed methods described earlier in Chapter 4. The robust saliency features found from the cleaned local neighbourhood are used for region growing based on three distance measures (OD, ED and θ) in the segmentation process. The developed algorithms have several advantages: (i) they are computationally simpler, (ii) they are significantly faster than robust versions of PCA e.g. Fast-MCD and Deterministic MCD based PCA, (iii) they are able to efficiently handle high percentages (up to 75%; using MCMD_MD algorithm) of clustered and uniform outliers, (iv) they outperform PCA and are significantly better than RANSAC for classification and segmentation, (v) they produce more accurate and robust results, (vi) they reduce over and under segmentation, and (vii) they are semi-automatic, depending only on two user defined parameters: neighbourhood size and angle threshold.

In many cases, it is necessary to segment point clouds slice by slice because a large dataset cannot be held in memory. This thesis also introduced a merging algorithm that takes the results for each slice and seamlessly merges them. Results for real MLS point cloud data show that the proposed algorithm can correctly merge many consecutive pieces of segmented slices. A limitation of the algorithm is: that may be sensitive to the segmentation parameters that may need to be to adjusted for different slices.

7.1.4 Robust Ground Surface Extraction

In many applications of point cloud processing e.g. transport corridor asset management, object surface reconstruction and modelling, and feature extraction, it is helpful to classify points into ground and non-ground surface points.

This thesis introduced a ground surface extraction algorithm that is able to separate ground and non-ground object surfaces. The method uses locally weighted regression where two types of robust regression (LMS and LTS) and a robust weight function have been used to robustify the locally weighted regression. Three variants of ground surface extraction methods are proposed. The statistically robust approaches: RLWLS, LWLMS and LWLTS can extract ground surfaces in urban areas. Results from the experiments using real MLS datasets showed that most of the urban features such as complex large buildings, large trees, short walls, sign posts, power poles, traffic signals, and vehicles were efficiently separated from the ground surfaces. A number of real data experiments showed that RLWLS is significantly faster than robust regression based LWLMS and LWLTS for data point classification and is comparable with LWLMS and LWLTS without compromising its efficiency. RLWR can reduce the cost of the required point cloud post-processing tasks because it allows segmentation and other algorithms to concentrate on the above ground features that typically contain approximately half of all the data points. The new method can also efficiently handle the presence of low outliers. Comparison with the robust segmentation Algorithm 5.4 showed that the proposed robust filtering algorithm can correctly classify more than 95% of ground and non-ground surface points. The method can also exclude the majority of non-ground points without filtering out points belonging to small vertical surfaces e.g. road kerbs, when considering the determination of ground points.

7.2 Future Research

Much research has been carried out on feature extraction in many disciplines including computer vision, computer graphics, image processing and robotics, where the applications of laser scanning data are frequently seen. As we determined from the literature, the use of robust statistics has rarely been applied in the areas of photogrammetry and remote sensing. Our observations are that the limited use of robust statistics is mainly because: (i) many robust methods are not computationally efficient, are not easy to apply and making inference is not typical, and (ii) most of the methods do not tolerate more than 50% outliers. In this thesis it was successfully proved that the use of robust statistics significantly improve the accuracy and the robustness of the results in the presence of outliers and noise.

We have developed methods for planar surface fitting which were used later for robust saliency features including normal and curvature estimation and finally for segmentation. These need to be explored for other geometric primitives such as sphere, cylinder and other more complex shapes allowing more accurate, robust and representative object modelling. Using the fitted robust planar surfaces there is the opportunity for generating robust registration techniques for point cloud data collected from different overlapping scans. Although using MCMD_MD we were able to deal with up to 75% outliers in the data, future research is suggested to develop methods that can find outliers even in the presence of more than 75% outliers, given that multiple complex structures in point cloud data may produce more than 75% of pseudo outliers and noise. The segmentation methods developed in this thesis are more appropriate for smooth surface segmentation. Future research is needed to develop more efficient robust segmentation and surface reconstruction methods for non-smooth surfaces. The proposed segmentation algorithms have the potential for future research in object detection, recognition and modelling. The proposed ground surface extraction can be improved to filter the points belonging to small vertical surfaces such as road kerbs. Further work can be proceed for the development of more automated process for ground surface extraction, and for more specific classification and recognition of ground and non-ground objects.

Although robust methods take much time, the proposed algorithms significantly reduce computation time and are more accurate. The computation time will decrease if we implement the algorithms in C or C++.

Bibliography

Aggarwal, C. (2013). *Outlier Analysis*. Springer, New York, USA.

Agostinelli, C., Filzmoser, P., and Salibian-Barrera, M. (2007). Final report. In *Proceedings of the International Workshop on Robust Statistics and R*, pages 1–11, Banff, Alberta, Canada.

Akel, N. A., Filin, S., and Doytsher, Y. (2007). Orthogonal polynomials supported by regional growing segmentation for the extraction of terrain from LiDAR data. *Photogrammetric Engineering and Remote Sensing*, 73(11):1253–1266.

Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. T. (2001). Point set surfaces. In *Proceedings of the 12th IEEE International Conference on Visualization*, pages 21–28, San Diego, California, USA.

Amenta, N. and Bern, M. (1999). Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry*, 22(4):481–504.

Amenta, N. and Kil, Y. J. (2004). Defining point-set surfaces. *ACM Transactions on Graphics*, 23(3):264–270.

Ammann, L. P. (1993). Robust singular value decompositions: a new approach to projection pursuit. *Journal of the American Statistical Association*, 88(422):505–514.

Arditi, R., Garozzo, M., Laddomada, F., Paris, R., Rossi, S., Rotondi, A., and Zampa, F. (2010). New mobile LiDAR and satellite technologies for a better knowledge of roads - application to modern motorways and the case of the ancient appian way. In *ASECAP Annual Study and Information Days*, pages 120–128, Oslo, Norway.

Atkinson, A. C. and Riani, M. (2000). *Robust Diagnostic Regression Analysis*. Springer, New York, USA.

Avery, M. (2012). Literature review for local polynomial regression. http://www4.ncsu.edu/ mravery/AveryReview2.pdf, Accessed: 20/05/2013.

Axelsson, P. (1999). Processing of laser scanner data–algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2–3):138–147.

Axelsson, P. (2000). DEM generation from laser scanner data using adaptive TIN models. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 33(B4/1):110–117.

Bae, K., Belton, D., and Lichti, D. D. (2005). A framework for position uncertainty of unorganised three-dimensional point clouds from near-monostatic laser scanners using covariance analysis. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/W19):7–12.

Barnea, S. and Filin, S. (2013). Segmentation of terrestrial laser scanning data using geometry and image information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76:33–48.

Barnett, V. and Lewis, T. (1995). *Outliers in Statistical Data*. John Wiley and Sons, New York, USA.

Bartels, M., Wei, H., and Mason, D. C. (2006). DTM generation from LiDAR data using skewness balancing. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 566–569, Hong Kong, China.

Becker, C., Fried, R., and Kuhnt, S. (2013). *Robustness and Complex Data Structures*. Springer, Heidelberg, Berlin, Germany.

Belton, D. (2008). *Classification and Segmentation of 3D Terrestrial Laser Scanner Point Clouds*. PhD Thesis, Department of Spatial Sciences, Curtin University of Technology, Australia.

Belton, D. and Bae, K.-H. (2010). Automatic post-processing of terrestrial laser scanning point clouds for road feature surveys. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(5):74–79.

Benkö, P. and Várady, T. (2004). Segmentation methods for smooth point regions of conventional engineering objects. *Computer-Aided Design*, 36(6):511–523.

Beraldin, J.-A., Blais, F., and Lohr, U. (2010). Laser scanning technology. In Vosselman, G. and Maas, H.-G., editors, *Airborne and Terrestrial Laser Scanning*, pages 1–42. Whittles Publishing/CRC Press, Scotland, UK.

Berkmann, J. and Caelli, T. (1994). Computation of surface geometry and segmentation using covariance techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1114–1116.

Besl, P. J. and Jain, R. C. (1988). Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):167–192.

Bishop, C. (2006). *Pattern Recognition and Machine Learning.* Springer, New York, USA.

Borrmann, D., Elseberg, J., Lingemann, K., and Nüchter, A. (2011). The 3D Hough Transform for plane detection in point clouds: a review and a new accumulator design. *3D Research, Springer*, 2(2):1–13.

Boulaassal, H., Landes, T., Grussenmeyer, P., and Tarsha-Kurdi, F. (2007). Automatic segmentation of building facades using terrestrial laser data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 3/W52):65–70.

Boulch, A. and Marlet, R. (2012). Fast and robust normal estimation for point clouds with sharp features. *Computer Graphics Forum*, 31(5):1765–1774.

Box, G. E. P. (1953). Non-normality and tests on variances. *Biometrika*, 40(3/4):318–335.

Box, G. E. P. (1954). Some theorems on quadratic forms applied in the study of analysis of variance problems: effect of inequality of variance in the one-way classification. *The Annals of Mathematical Statistics*, 25(2):290–302.

Breuning, M., Kriegel, H. P., Ng, R., and Sander, J. (2000). LOF: Identifying density-based local outliers. In *Proceeding of the ACM SIGMOD International Conference on Management of Data*, pages 93–104, Dallas, Texas, USA.

Briese, C. (2010). Extraction of digital terrain models. In Vosselman, G. and Mass, H.-G., editors, *Airborne and Terrestrial Laser Scanning*, pages 135–167. Whittles Publishing/CRC Press, Scotland, UK.

Briese, C., Pfeifer, N., and Dorninger, P. (2002). Applications of the robust interpolation for DTM determination. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(3A):55–61.

Brovelli, M., Cannata, M., and Longoni, U. (2004). LiDAR data filtering and DTM interpolation within GRASS. *Transactions in GIS*, 8(2):155–174.

Butler, R. W., Davies, P. L., and Jhun, M. (1993). Asymptotics for the minimum covariance determinant estimator. *Annals of Statistics*, 21(3):1385–1401.

Campbell, N. A. (1980). Robust procedures in multivariate analysis I: robust covariance estimation. *Journal of the Royal Statistical Society, Serics C*, 29(3):231–237.

Candés, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *Journal of the ACM*, 58(3):11.

Castillo, E., Liang, J., and Zhao, H. (2013). Point cloud segmentation and denoising via constrained nonlinear least squares surface normal estimates. In Breuß, M., Bruckstein, A., and Maragos, P., editors, *Innovations for Shape Analysis: Models and Algorithms*, pages 283–298. Springer, New York, USA.

Chandola, V. (2008). Real-time credit card fraud detection. *Expert Systems with Applications*, 35(4):1721–1732.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: a survey. *ACM Computing Surveys*, 41(3):Article No. 15.

Chatterjee, S. and Hadi, A. S. (1988). *Sensitivity Analysis in Linear Regression.* John Wiley and Sons, New York, USA.

Chatterjee, S. and Hadi, A. S. (2012). *Regression Analysis by Examples.* John Wiley and Sons, New York, USA, 5th edition.

Chen, C. C. and Stamos, I. (2007). Range image segmentation for modeling and object detection in urban scenes. In *Proceeding of the 6th International*

Conference on 3-D Digital Imaging and Modeling, pages 185–192, Quebec, Canada.

Choi, S., Kim, T., and Yu, W. (2009). Performance evaluation of RANSAC family. In *Proceedings of the British Machine Vision Conference*, pages 1–12, London, UK.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836.

Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610.

Cleveland, W. S. and Grosse, E. (1991). Computational methods for local regression. *Statistics and computing*, 1(1):47–62.

Cleveland, W. S. and Loader, C. L. (1996). Smoothing by local regression: principles and methods. In Haerdle, W. and Schimek, M. G., editors, *Statistical Theory and Computational Aspects of Smoothing*, pages 10–49. Springer, New York, USA.

Cook, R. D. and Weisberg, S. (1982). *Residuals and Influence in Regression*. Chapman and Hall, London, UK.

Crosilla, F., Macorig, D., Sebastianutti, I., and Visintini, D. (2011). Points classification by a sequential higher–order moments statistical analysis of LiDAR data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 5/W12):1–6.

Crosilla, F., Visintini, D., and Sepic, F. (2009). Automatic modeling of laser point clouds by statistical analysis of surface curvature values. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 5/W1):1–6.

Croux, C. and Dehon, C. (2013). Robust estimation of location and scale. In El-Shaarawi, A. H. and Piegorsch, W. W., editors, *Encyclopedia of Environmetrics*. John Wiley and Sons, New York, USA.

Croux, C. and Haesbroeck, G. (2000). Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika*, 87(3):603–618.

Croux, C. and Rousseeuw, P. J. (1992). Time-efficient algorithms for two highly robust estimators of scale. In *Proceedings of the 10th Symposium on Computational Statistics*, volume 1, pages 411–428, Heidelberg.

Croux, C. and Ruiz-Gazen, A. (1996). A fast algorithm for robust principal components based on projection pursuit. In *Proceedings of the 12th Symposium in Computational Statistics*, pages 211–216, Barcelona, Spain.

Croux, C. and Ruiz-Gazen, A. (2005). High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95:206–226.

Davies, L. (1987). Asymptotic behavior of S-estimators of multivariate location parameters and dispersion matrices. *The Annals of Statistics*, 15(3):1269–1292.

Davies, P. L. and Gather, U. (2004). Robust statistics. Technical Report 20, Center for Applied Statistics and Economics (CASE), University Berlin, Germany.

Debruyne, M. and Hubert, M. (2009). The influence function of the Stahel–Donoho covariance estimator of smallest outlyingness. *Statistics and Probability Letters*, 79(3):275–282.

Dervilis, N., Cross, E. J., Barthorpe, R. J., and Worden, K. (2014). Robust methods of inclusive outlier analysis for structural health monitoring. *Journal of Sound and Vibration*, 333(20):5181–5195.

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

Devlin, S. J., Gnandesikan, R., and Kettenring, J. R. (1981). Robust estimation of dispersion matrices and principal components. *Journal of the American Statistical Association*, 76(374):354–362.

Dey, T. K., Gang, L., and Sun, J. (2005). Normal estimation for point cloud: a comparison study for a Voronoi based method. In *Proceedings of the Eurographics Symposium on Point-Based Graphics*, pages 39–46, New York, USA.

Diamataras, K. I. and Kung, S. Y. (1996). *Principal Component Neural Networks: Theory and Applications.* John Wiley and Sons, New York, USA.

Donoho, D. and Gasko, M. (1992). Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *The Annals of Statistics*, 20(4):1803–1827.

Donoho, D. L. and Huber, P. J. (1993). The notion of breakdown point. In Bickel, P. J., Doksum, K., and J. L. Hodges, J., editors, *A Festschrift for Erich L. Lehmann*, pages 157–184. Wadsworth, Belmont, California, USA.

Donoho, L. (1982). *Breakdown properties of multivariate location estimators.* PhD Qualifying paper, Harvard University, Boston, USA.

Dorninger, P. and Nothegger, C. (2007). 3D segmentation of unstructured point clouds for building modelling. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/W49A):191–196.

Duda, R. O. and Hart, P. E. (1972). Use of Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15.

El-Sheimy, N. (2005). An overview of mobile mapping system. In *Proceedings of the From Pharaohs to Geoinformatics and GSDI-8*, Cairo, Egypt.

El-Sheimy, N., Valeo, C., and Habib, A. (2005). *Digital Terrain Modelling: Acquisition, Manipulation, And Its Applications.* Artech House, USA.

Ševljakov, G. L. and Vilčevskij, N. O. (2002). *Robustness in Data Analysis: Criteria and Methods.* VSP BV, The Netherlands.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications.* Chapman and Hall, London, UK.

Fawcett, T. (2006). Introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Feng, J., Xu, H., and Yan, S. (2012). Robust PCA in high-dimension: a deterministic approach. In *Proceedings of the 29th International Conference on Machine Learning*, pages 249–256, Edinburgh, Scotland, UK.

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

Fleishman, S., Cohen-Or, D., and Silva, C. (2005). Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics*, 24(3):544–552.

Fox, J. (2002). Robust regression; appendix to an R and S-PLUS companion to applied regression. http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-robust-regression.pdf, Accessed: 05/02/2014.

Friedman, J. and Tukey, J. (1974). A projection-pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23(9):881–889.

Fung, W. K. (1993). Unmasking outliers and leverage points: a confirmation. *Journal of the American Statistical Association*, 88(422):515–519.

Gallo, O., Manduchi, R., and Rafii, A. (2011). CC-RANSAC: fitting planes in the presence of multiple surfaces in range data. *Pattern Recognition Letters*, 32(3):403–410.

Garouani, E. A. and Alobeid, I. A. (2013). Digital surface model generation for 3D city modeling. In *8th National GIS Symposium in Saudi Arabia*, Dammam, Saudi Arabia.

Goldstein, M. (2012). FastLOF: an expectation-maximization based local outlier detection algorithm. In *Proceedings of the 21st International Conference on Pattern Recognition*, pages 2282–2285, Tsukuba, Japan.

Golub, G. H. and Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420.

Graham, L. (2010). Mobile mapping system overview. *Photogrammetric Engineering and Remote Sensing*, pages 222–228.

Hadi, A. S., Imon, A. H. M., and Werner, M. (2009). Detection of outliers. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1):57–70.

Hadi, A. S. and Simonoff, J. S. (1993). Procedures for the identification of outliers. *Journal of the American Statistical Association*, 88(424):1264–1272.

Hähnel, D., Burgard, W., and Thrun, S. (2003). Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 44(1):15–27.

Hampel, F., Ronchetti, E., Rousseeuw, P. J., and Stahel, W. (1986). *Robust Statistics: The Approach Based on Influence Functions*. John Wiley and Sons, New York, USA.

Hampel, F. R. (1968). *Contributions to the Theory of Robust Estimation*. PhD Thesis, University of California, Berkeley, USA.

Hampel, F. R. (1975). Beyond location parameters: robust concepts and methods. *Bulletin of the International Statistical Institute*, 46:375–391.

Haralick, R. M. and Shapiro, L. G. (1992). *Computer and Robot Vision*. Addison-Wesley, Reading, Mass.

Harati, A., Gächter, S., and Siegwart, R. (2007). Fast range image segmentation for indoor 3D-SLAM. In *proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles*, pages 475–480, Baudis, France.

Hawkins, D. M. (1980). *Identification of Outliers*. Chapman and Hall, London, UK.

Hebert, M. and Vandapel, N. (2003). Terrain classification techniques from ladar data for autonomous navigation. In *Proceedings of the Collaborative Technology Alliances Conference*, Robotics Institute, Carnegie Mellon University, USA.

Heo, J., Jeong, S., Park, H.-K., Jung, J. H., Han, S., Hong, S., and Sohn, H.-G. (2013). Productive high-complexity 3D city modeling with point clouds collected from terrestrial LiDAR. *Computers, Environment and Urban Systems*, 41:26–38.

Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., and Kanamori, T. (2011). Statistical outlier detection using direct density ratio estimation. *Knowledge Information System*, 26(2):309–336.

Hodges, V. J. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126.

Hoffman, R. and Jain, A. K. (1987). Segmentation and classification of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):608–620.

Hollander, M., Wolfe, D. A., and Chicken, E. (2014). *Nonparametric Statistical Methods*. John Wiley and Sons, New York, USA, 3rd edition.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D., Bowyer, K., Eggert, D., Fitzgibbon, A., and Fisher, R. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689.

Hoppe, H., Rose, T. D., and Duchamp, T. (1992). Surface reconstruction from unorganized points. In *Proceedings of the ACM SIGGRAPH*, volume 26, pages 71–78, Chicago, USA.

Huang, J. and 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.

Huber, P. J. (1964). Robust estimation of location parameter. *Annals of Mathematical Statistics*, 35:73–101.

Huber, P. J. (1981). *Robust Statistics*. John Wiley and Sons, New York, USA.

Huber, P. J. (1991). Between robustness and diagnostics. In Stahel, W. and S. Weisberg, S., editors, *Direction in Robust Statistics and Diagnostics*, pages 121–130. John Wiley and Sons, New York, USA.

Hubert, M., Rousseeuw, P., Vanpaemel, D., and Verdonck, T. (2014). The DetS and DetMM estimators for multivariate location and scatter. *Computational Statistics and Data Analysis*, http://dx.doi.org/10.1016/j.csda.2014.07.013.

Hubert, M., Rousseeuw, P. J., and Branden, K. V. (2005). ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79.

Hubert, M., Rousseeuw, P. J., and Stefan, V. A. (2008). High-breakdown robust multivariate methods. *Statistical Science*, 23(1):92–119.

Hubert, M., Rousseeuw, P. J., and Verboven, S. (2002). A fast robust method for principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 60(1–2):101–111.

Hubert, M., Rousseeuw, P. J., and Verdonck, T. (2012). A deterministic algorithm for robust scatter and location. *Journal of Computational and Graphical Statistics*, 21(3):618–637.

Huffel, S. V. and Vandewalle, J. (1991). *The Total Least Squares Problem: Computational Aspects and Analysis*. SIAM, Philadelphia, PA, USA.

Ip, A. W. L., El-Sheimy, N., and Mostafa, M. M. R. (2007). Performance analysis of integrated IMU/DGPS systems for mobile mapping system. In Tao, C. V. and Li, J., editors, *Advances in Mobile Mapping Technology*, pages 63–78. Taylor and Francis Group, London, UK.

Jacoby, W. (2000). Loess: a nonparametric, graphical tool for depicting relationships between variables. *Electoral Studies*, 19(4):577–613.

Jiang, S. Y. and An, Q. B. (2008). Clustering-based outlier detection method. In *Proceedings of the 5th IEEE International Conference on Fuzzy Systems and Knowledge Discovery*, pages 429–433, Jinan, Shandong, China.

Jiang, X. Y., Bunke, H., and Meier, U. (1996). Fast range image segmentation using high-level segmentation primitives. In *Proceeding of the 3rd IEEE Workshop on Applications of Computer Vision*, pages 83–88, Florida, USA.

Jiang, X. Y., Bunke, H., and Meier, U. (2000). High-level feature based range image segmentation. *Image and Vision Computing*, 18(10):817–822.

Johnson, R. A. and Wichern, D. W. (2002). *Applied Multivariate Statistical Analysis*. Prentice Hall, New Jersey, USA, 5th edition.

Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer, New York, USA.

Kamberov, G. and Kamberova, G. (2004). Topology and geometry of unorganized point clouds. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission*, pages 743–750, Thessaloniki, Greece.

Kanamori, T., Hido, S., and Sugiyama, M. (2009). A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445.

Kanatani, K. (1996). *Statistical Optimization for Geometric Computation: Theory and Practice.* Elsevier Science, Amsterdam.

Kent, J. and Tyler, D. (1996). Constrained M-estimation for multivariate location and scatter. *The Annals of Statistics*, 24(3):1346–1370.

Khalifa, I., Moussa, M., and Kamel, M. (2003). Range image segmentation using local approximation of scan lines with application to CAD model acquisition. *Machine Vision and Applications*, 13(5–6):263–274.

Kilian, J., Haala, N., and Englich, M. (1996). Capture and evaluation of airborne laser scanner data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 31(B3):383–388.

Klasing, L., Althoff, D., Wollherr, D., and Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3206–3211, Kobe, Japan.

Knorr, E. M. and Ng, R. T. (1998). Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th International Conference on Very Large Databases (VLDB)*, pages 392–403, New York, USA.

Kobler, A., Pfeifer, N., Ogrinc, P., Todorovski, L., Ostir, K., and Dzeroski, S. (2007). Repetitive interpolation: a robust algorithm for DTM generation from aerial laser scanner data in forested terrain. *Remote Sensing of Environment*, 108(1):9–23.

Koster, K. and Spann, M. (2000). MIR: an approach to robust clustering-application to range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):430–444.

Kraus, K., Karel, W., Briese, C., and Mandlburger, G. (2006). Local accuracy measures for digital terrain models. *The Photogrammetric Record*, 21(116):342–354.

Kraus, K. and Pfeifer, N. (1998). Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(4):193–203.

Kriegel, H.-P., Kroger, P., Schubert, E., and Zimek, A. (2009). LoOP: local outlier probabilties. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 1649–1652, Hongkong, China.

Kriegel, H.-P., Scubert, M., and Zimek, A. (2008). Angel-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 442–452, Lasvegas, USA.

Kutterer, H. (2010). Mobile mapping. In Vosselman, G. and Maas, H.-G., editors, *Airborne and Terrestrial Laser Scanning*, pages 293–311. Whittles Publishing/CRC Press, Scotland, UK.

Kwon, S.-W., Boshe, F., Kim, C., Haas, C. T., and Liapi, K. A. (2004). Fitting range data to primitives for rapid local 3D modeling using sparse range point clouds. *Automation in Construction*, 13(1):67–81.

Lari, Z. and Habib, A. (2014). An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:192–212.

Lay, D. C. (2012). *Linear Algebra and its Applications.* Pearson, Boston, USA.

Lee, Y., Park, S., Jun, Y., and Choi, W. (2004). A robust approach to edge detection of scanned point data. *The International Journal of Advanced Manufacturing Technology*, 23(3-4):263–271.

Leslar, M., Wang, J. G., and Hu, B. (2010). A comparison of two new methods of outlier detection for mobile terrestrial LiDAR data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 1):78–84.

Levin, D. (2003). Mesh-independent surface interpolation. In Brunnett, Hamann, and Mueller, editors, *Geometric Modeling for Scientific Visualization*, pages 37–49. Springer, New York, USA.

Li, B., Schnabel, R., Klein, R., Cheng, Z., Dang, G., and Jin, S. (2010). Robust normal estimation for point clouds with sharp features. *Computers and Graphics*, 34(2):94–106.

Li, G. and Chen, Z. (1985). Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and Monte Carlo. *Journal of the American Statistical Association*, 80(391):759–766.

Li, W. E., Guo, Q., Jakubowski, M.-K., and Kelly, M. (2012). A new method for segmenting individual trees from the Lidar point cloud. *Photogrammetric Engineering and Remote Sensing*, 78(1):75–84.

Liang, J., Park, F., and Zhao, H. (2011). Robust and efficient implicit surface reconstruction for point clouds based on convexified image segmentation. Technical report, Department of Mathematics, University of California, Irvine, 340 Rowland Hall, Canada.

Lichti, D. and Skaloud, J. (2010). Registration and calibration. In Vosselman, G. and Maas, H.-G., editors, *Airborne and Terrestrial Laser Scanning*, pages 43–133. Whittles Publishing/CRC Press, Scotland, UK.

Lin, C.-H., Chen, J.-Y., Su, P.-L., and Chen, C.-H. (2014). Eigen-feature analysis of weighted matrices for LiDAR point cloud classification. *ISPRS Jouurnal of Photogrammetry and Remote Sensing*, 94:70–79.

Lindenberger, J. (1993). *Laser-Profilmessungen zur topographischen gelaedeaufnahme*. PhD Thesis, Institut für Photogrammetrie, Fakultät Luft- und Raumfahrttechnik und Geodäsie, Deutsche Geodaetische Kommission, Series C, No. 400, Munich, Germany.

Liu, B., Xiao, Y., Cao, L., Hao, Z., and Deng, F. (2013). SVDD-based outlier detection on uncertain data. *Knowledge Information Systems*, 34(3):597–618.

Liu, Y. and Xiang, Y. (2008). Automatic segmentation of unorganized noisy point clouds based on the Gaussian map. *Computer-Aided Design*, 40(5):576–594.

Loader, C. (2004). Smoothing: local regression techniques. Technical Report 2004, 12, Center for Applied Statistics and Economics (CASE), University Berlin, Germany.

Locantore, N., Marron, J. S., Simpson, D. G., Tripoli, N., Zhang, J. T., and Cohen, K. L. (1999). Robust principal component analysis for functional data. *Test*, 8(1):1–73.

Lopuhaä, H. P. (1991). Multivariate τ-estimators for location and scatter. *The Canadian Journal of Statistics*, 19(3):307–321.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. In *Proceedings of the National Institute of Science*, volume 2, pages 49—55, India.

Maronna, R. A. (1976). Robust M-estimators of multivariate location and scatter. *The Annals of Statistics*, 4(1):51–67.

Maronna, R. A. (2005). Principal components and orthogonal regression based on robust scales. *Technometrics*, 47(3):264–273.

Maronna, R. A., Martin, R. D., and Yohai, V. J. (2006). *Robust Statistics: Theory and Methods*. John Wiley and Sons, New York, USA.

Maronna, R. A. and Yohai, V. (1998). Robust estimation of multivariate location and scatter. In Kotz, S. and C. Read, D. B., editors, *Encyclopedia of Statistical Sciences*, pages 589–596. John Wiley and Sons, New York, USA.

Maronna, R. A. and Yohai, V. J. (1995). The behavior of the Stahel-Donoho robust multivariate estimator. *Journal of the American Statistical Association*, 90(429):330–341.

Maronna, R. A. and Zamar, R. H. (2002). Robust estimates of location and dispersion for high-dimensional datasets. *Technometrics*, 44(4):307–317.

Marshall, D., Lukacs, G., and Martin, R. (2001). Robust segmentation of primitives from range data in presence of geometric degeneracy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):304–314.

Masuda, H., Tanaka, I., and Enomoto, M. (2013). Reliable surface extraction from point-clouds using scanner-dependent parameters. *Computer Aided Design and Applications*, 10(2):265–277.

Matas, J. and Chum, O. (2004). Randomized RANSAC with $T_{d,d}$ test. *Image and Vision Computing*, 22:837–842.

McGill, R., Tukey, J. W., and Larsen, W. A. (1978). Variations of boxplots. *The American Statistician*, 32(1):12–16.

Meer, P. (2004). Robust techniques for computer vision. In Medioni, G. and Kang, S. B., editors, *Emerging Topics in Computer Vision*, pages 107–190. Prentice Hall, New Jersey, USA.

Meer, P., Mintz, D., Rosenfeld, A., and Kim, D. Y. (1991). Robust regression methods for computer vision: a review. *International Journal of Computer Vision*, 6(1):59–70.

Meng, X., Currit, N., and Zhao, K. (2010). Ground filtering algorithms for airborne LiDAR data: a review of critical issues. *Remote Sensing*, 2(3):833–860.

Michel, J., Youssefi, D., and Grizonnet, M. (2014). Stable mean-shift algorithm and its application to the segmentation of arbitrarily large remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(2):952–964.

Mitra, N. J. and Nguyen, A. (2003). Estimating surface normals in noisy point cloud data. In *Proceedings of the 19th ACM Symposium on Computational Geometry*, pages 322–328, San Diego, California, USA.

Mitra, N. J., Nguyen, A., and Guibas, L. (2004). Estimating surface normals in noisy point cloud data. *Special Issue of the International Journal of Computational Geometry and Applications*, 14(4–5):261–276.

Montgomery, D. C., Peck, E., and Vining, G. G. (2012). *Introduction to linear regression analysis*. John Wiley and Sons, New York, USA, 5th edition.

Moran, G. W. (1984). *Locally-Weighted-Regression Scatter-Plot Smoothing (LOWESS): A Graphical Exploratory Data Analysis Technique*. Master's Thesis, Department of Spatial Sciences, Naval Postgraduate School, Monterey, California.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press, London, England.

Nomikos, P. and MacGregor, J. F. (1995). Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37(1):41–59.

Novak, K. (1993). Data collection for multi-media GIS using mobile mapping systems. *GIM*, 7(3):30–32.

Önskog, J., Freyhult, E., Landfors, M., Rydèn, P., and Hvidsten, T. R. (2011). Classification of microarrays; synergistic effects between normalization, gene selection and machine learning. *BMC Bioinformatics*, 12(1):390, doi:10.1186/1471–2105–12–39.

Nurunnabi, A., Belton, D., and West, G. (2012a). Diagnostic-robust statistical analysis for local surface fitting in 3D point cloud data. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 1–3, pages 269–274.

Nurunnabi, A., Belton, D., and West, G. (2012b). Robust and diagnostic statistics: a few basic concepts in mobile mapping point cloud data analysis. In *Proceedings of the International Conference on Statistical Data Mining for Bioinformatics, Health, Agriculture and Environment*, pages 591–602, Rajshahi, Bangladesh.

Nurunnabi, A., Belton, D., and West, G. (2012c). Robust segmentation for multiple planar surface extraction in laser scanning 3D point cloud data. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*, pages 1367–1370, Tsukuba Science City, Japan.

Nurunnabi, A., Belton, D., and West, G. (2012d). Robust segmentation in laser scanning 3D point cloud data. In *Proceedings of the Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, Fremantle, Australia.

Nurunnabi, A., Belton, D., and West, G. (2013a). Diagnostics based principal component analysis for robust plane fitting in laser data. In *Proceedings of the 16th International Conference on Computer and Information Technology (ICCIT)*, pages 484–489, Khulna, Bangladesh.

Nurunnabi, A., Belton, D., and West, G. (2014a). Robust statistical approaches for local planar surface fitting in 3D laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 96:106–122.

Nurunnabi, A., Hadi, A. S., and Imon, A. H. M. R. (2014b). Procedures for the identification of multiple influential observations in linear regression. *Journal of Applied Statistics*, 41(6):1315–1331.

Nurunnabi, A. and West, G. (2012). Outlier detection in logistic regression: a quest for reliable knowledge from predictive modeling and classification. In *Proceedings of the IEEE 12th International Conference on Data Mining (ICDM), Workshops on Reliability Issues in Knowledge Discovery (RIKD)*, pages 643–652, Brussels, Belgium.

Nurunnabi, A., West, G., and Belton, D. (2013b). Robust locally weighted regression for ground surface extraction in mobile laser scanning 3D data. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume II–5/W2, pages 217–222.

Nurunnabi, A., West, G., and Belton, D. (2013c). Robust outlier detection and saliency features estimation in point cloud data. In *Proceedings of the 10th Canadian Conference on Computer and Robot Vision (CRV 2013)*, pages 98–105, Regina, Canada.

Nurunnabi, A. A. M. and Dai, H. (2012). Robust-diagnostic regression: a prelude for inducing reliable knowledge from regression. In Dai, H., Liu, J. N. K., and Smirnov, E., editors, *Reliable Knowledge Discovery*, pages 69–90. Springer, New York, USA.

Nurunnabi, A. A. M., Imon, A. H. M. R., and Nasser, M. (2011). A diagnostic measure for influential observations in linear regression. *Communications in Statistics-Theory and Methods*, 40(7):1169–1183.

Öztireli, A. C., Guennebaud, G., and Gross, M. (2009). Feature preserving point set surfaces based on nonlinear kernel regression. *Computer Graphics Forum*, 28(2):493–501.

Pauly, M., Gross, M., and Kobbelt, L. P. (2002). Efficient simplification of point sample surface. In *Proceedings of the International Conference on Visualization*, pages 163–170, Washington, D.C., USA.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572.

Petrie, G. (2010). An introduction to the technology: mobile mapping systems. *GEOinformatics Magazine*, 13:32–43.

Pfeifer, N. (2003). Oberflächenmodelle aus laserdaten. *VGI–Österreichische Zeitschrift für Vermessung & Geoinformation*, 91(4/03):243–252.

Pfeifer, N. (2005). A subdivision algorithm for smooth 3D terrain models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):115–127.

Pfeifer, N. and Briese, C. (2007). Geometrical aspects of airborne laser scanning and terrestrial laser scanning. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 3/W52):311–319.

Pfeifer, N. and Mandlburger, G. (2009). LiDAR data filtering and DTM generation. In Shan, J. and Toth, C. K., editors, *Topographic Laser Ranging and Scanning*, pages 307–334. CRC Press, Taylor & Francis Group, London, New York.

Pfeifer, N., Stadler, P., and Briese, C. (2001). Derivation of digital terrain models in the SCOP++ environment. In *Proceedings of the OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Terrain Models*, Stockholm, Sweden.

Poppinga, J., Vaskevicius, N., Birk, A., and Pathak, K. (2008). Fast plane detection and polygonalization in noisy 3D range image. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3378–3383, Nice, France.

Powell, M. W., Bowyer, K. W., Jiang, X., and Bunke, H. (1998). Comparing curved-surface range image segmenters. In *Proceedings of the 6th IEEE International Conference on Computer Vision*, pages 286–291, Bombay, India.

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

Rabbani, T. (2006). *Automatic Reconstruction of Industrial Installations Using Point Clouds and Images.* PhD Thesis, Photogrammetry and Remote Sensing, TU Delft, The Netherlands.

Rabbani, T., van den Heuvel, F. A., and Vosselman, G. (2006). Segmentation of point clouds using smoothness constraint. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):248–253.

Raguram, R., Frahm, J. M., and Pollefeys, M. (2008). A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *Proceeding of the 10th European Conference on Computer Vision (ECCV)*, pages 500–513, Marseille, France.

Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD, International Conference on Management of Data*, pages 427–438, Dallas, Texas, USA.

Rexhepaj, E., Agnarsdòttir, M., Bergman, J., Edqvist, P., Bergqvist, M., Uhlèn, M., Gallagher, W. M., Lundberg, E., and Ponten, F. (2013). Distinguish melanoma from non-melanoma cells in histopathological tissue microarray sections. *PLoS ONE*, 8(5):1–15.

Rocke, D. M. and Woodruff, D. L. (1996). Identification of outliers in multivariate data. *Journal of the American Statistical Association*, 91(435):1047–1060.

Rousseeuw, P., Debruyne, M., Engelen, S., and Hubert, M. (2006). Robustness and outlier detection in chemometrics. *Critical Reviews in Analytical Chemistry*, 36(3–4):221–242.

Rousseeuw, P. and Struyf, A. (1998). Computing location depth and regression depth in higher dimensions. *Statistics and Computing*, 8(3):193–203.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880.

Rousseeuw, P. J. (1991). Tutorial to robust statistics. *Journal of Chemometrics*, 5(1):1–20.

Rousseeuw, P. J. and Croux, C. (1993). Alternative to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283.

Rousseeuw, P. J. and Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223.

Rousseeuw, P. J. and Hubert, M. (2011). Robust statistics for outlier detection. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):73–79.

Rousseeuw, P. J. and Leroy, A. (2003). *Robust Regression and Outlier Detection.* John Wiley and Sons, New York, USA.

Rousseeuw, P. J. and van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85(411):633–639.

Samet, H. (2006). *Foundations of Multidimensional and Metric Data Structures.* Morgan Kaufmann, San Francisco, USA.

Sanchez, V. and Zakhor, A. (2012). Planar 3D modelling of building interiors from point cloud data. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 1777–1780, Florida, USA.

Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226.

Schölkopf, B., Patt, J. C., Shawe-Taylor, J. C., Smola, A. J., and Williamson, R. C. (2001). Estmating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471.

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *Proceedings of the 7th International Conference on Artificial Neural Networks*, pages 583–588, Lausanne, Switzerland.

Schubert, E., Zimek, A., and Kriegel, H.-P. (2014). Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining Knowledge Discovery*, 28(1):190–237.

Schwarz, K. P. and EI-Sheimy, N. (2007). Digital mobile mapping systems-state of the art and future trends. In Tao, C. V. and Li, J., editors, *Advances in Mobile Mapping Technology*, pages 3–18. Taylor and Francis, London, USA.

Searle, S. R. (2006). *Matrix Algebra Useful for Statistics.* John Wiley and Sons, New York, USA.

Serna, A. and Marcotegui, B. (2014). Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:243–255.

Shakarji, C. M. (1998). Least-squares fitting algorithms of the NIST algorithm testing system. *Journal of Research of the National Institute of Standards and Technology*, 103(6):633–641.

Shan, J. and Sampath, A. (2005). Urban DEM generation from raw LiDAR data: a labelling algorithm and its performance. *Photogrammetric Engineering and Remote Sensing*, 71(2):217–226.

Shan, J. and Toth, C. K., editors (2009). *Topographic Laser Ranging and Scanning.* CRC Press, Taylor & Francis Group, London, New York.

Sheskin, D. J. (2004). *Handbook of Parametric and Nonparametric Statistical Procedures.* Chapman and Hall/CRC, USA, 3rd edition.

Sheung, H. and Wang, C. C. (2009). Robust mesh reconstruction from unoriented noisy points. In *Proceedings of the SIAM/ACM Joint Conference on Geometric and Physical Modeling*, pages 13–24, San Francisco, USA.

Sithole, G. and Vosselman, G. (2004). Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(1):85–101.

Sithole, G. and Vosselman, G. (2005). Filtering of airborne laser scanner data based on segmented point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 3/W19):66–71.

Sohn, G. and Dowman, I. (2002). Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(3A):336–344.

Sokolova, M., Japkowicz, N., and Szpakowicz, S. (2006). Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In *Proceedings of the 19th Australian Joint Conference on Artificial Intelligence*, pages 1015–1021, Hobart, Australia.

Sotoodeh, S. (2006). Outlier detection in laser scanner point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):297–302.

Soudarissanane, S., Lindenbergh, R., Menenti, M., and Teunissen, P. (2011). Scanning geometry: influencing factor on the quality of terrestrial laser scanning points. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(4):389–399.

Stahel, W. and Weisberg, S., editors (1991). *Direction in robust statistics and diagnostics, Part II.* Springer, New York, USA.

Stahel, W. A. (1981). *Robust Estimation: Infinitesimal optimality and covariance matrix estimators.* PhD Thesis, Department of Mathematics, Eidgenössische Technische Hochschule (ETH), Zurich, Switzerland.

Stal, C., Briese, C., Maeyer, P. D., Dorninger, P., Nuttens, T., Pfeifer, N., and Wulf, A. D. (2014). Classification of airborne laser scanning point clouds based on binomial logistic regression analysis. *International Journal of Remote Sensing*, 35(9):3219–3236.

Stewart, C. V. (1999). Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537.

Stewart, C., V. (1995). MINPRAN: a new robust estimator for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intellgence*, 17(10):925–938.

Storer, M., Roth, P. M., M., U., Bischof, H., and Birchbauer, J. A. (2010). Efficient robust active appearance model fitting. In Ranchordas, A., Pereira, J. m., Araùjo, H. J., and Tavares, J. M. R. S., editors, *Computer Vision, Imaging and Computer Graphics: Theory and Applications*, pages 229–241. Springer-Verlag, Berlin, Germany.

Subbarao, R. and Meer, P. (2006). Beyond RANSAC: user independent robust regression. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition Workshop*, page 101, New York, USA.

Sugiyama, M. and Borgwardt, K. M. (2013). Rapid distance-based outlier detection via sampling. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 467–475, Navada, USA.

Sullivan, J. M. (2008). Curvature measures for discrete surfaces. In *Proceedings of the SIGGRAPH Asia 2008 Course Notes*, pages 10–13, Singapore.

Tao, C. V. and Li, J., editors (2007). *Advances in Mobile Mapping Technology.* Taylor and Francis Group, London, UK.

Tarsha-Kurdi, F., Landes, T., and Grussenmeyer, P. (2007). Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from LiDAR data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 3/W52):407–412.

Tax, D. and Duin, R. (2004). Support vector data description. *Machine Learning*, 54(1):45–66.

Tordoff, B. J. and Murray, D. W. (2005). Guided-MLESAC: faster image transform estimation by using matching priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1523–1535.

Torr, P. H. S. and Zisserman, A. (2000). MLESAC: a new robust estimator with application to estimating image geometry. *Journal of Computer Vision and Image Understanding*, 74(1):138–156.

Toth, C. K. (2009). R & D of mobile LiDAR mapping and future trends. In *Proceedings of the ASPRS annual conference*, New York, USA.

Tóvári, D. and Pfeifer, N. (2005). Segmentation based robust interpolation a new approach to laser data filtering. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 3):79–84.

Tukey, J. W. (1977). *Exploratory Data Analysis.* Addison-Wesley, Reading, Mass.

Tukey, J, W. (1960). A survey of sampling from contaminated distributions. In Olkin, I., Ghurye, S., Hoeffding, W., Madow, W., and Mann, H., editors, *Contribution to Probability and Statistics*, pages 448–485. Stanford University Press, California, USA.

Velleman, P. F. and Hoaglin, D. C. (1981). *Applications, Basics, and Computing of Exploratory Data Analysis.* Duxbury Press, Pacific Grove, Canada.

Visuri, S., Koivunen, V., and Oja, H. (2000). Sign and rank covariance matrices. *Journal of Statistical Planning and Inference*, 91(2):557–575.

Vosselman, G. (2000). Slope based filtering of laser altimetry data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 33(B3):935–942.

Vosselman, G., Gorte, B. G. H., Sithole, G., and Rabbani, T. (2004). Recognizing structure in laser scanner point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 8/W2):33–38.

Vosselman, G. and Klein, R. (2010). Visualisation and structuring of point clouds. In Vosselman, G. and Maas, H.-G., editors, *Airborne and Terrestrial Laser Scanning*, pages 45–79. Whittles Publishing/CRC Press, Scotland, UK.

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

Wagner, W., Eberhöfer, C., Hollaus, M., and Summer, G. (2004). Robust filtering of airborne and laser scanner data for vegetation analysis. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 8/W2):56–61.

Walpole, R. E., Myers, R. H., and Myers, S. L. (1998). *Probability and Statistics for Engineers and Scientists*. Prentice Hall International Inc., New Jersey, USA.

Wang, C., Tanahashi, H., and Hirayu, H. (2001). Comparison of local plane fitting methods for range data. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 663–669, Kauai, HI, USA.

Wang, H., Chin, T.-J., and Suter, D. (2012a). Simultaneously fitting and segmenting multiple-structure data with outliers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1177–1192.

Wang, H. and Suter, D. (2004). Robust adaptive-scale parametric model estimation for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1459–1474.

Wang, J. and Shan, J. (2009). Segmentation of LiDAR point clouds for building extraction. In *Proceedings of the American Society for Photogrammetry Remote Sensing Annual Conference*, pages 870–882, Baltimore, Maryland, USA.

Wang, J., Yang, Z., and Chen, F. (2012b). A variational model for normal computation of point clouds. *The Visual Computer*, 28(2):163–174.

Wang, Y., Feng, H.-Y., Delorme, F.-E., and Engin, S. (2013). An adaptive normal estimation method for scanned point clouds with sharp features. *Computer-Aided Design*, 45(11):1333–1348.

Weber, C., Hahmann, S., Hagen, H., and Bonneau, G.-P. (2012). Sharp feature preserving MLS surface reconstruction based on local feature line approximations. *Graphical Models*, 74(6):335–345.

Woo, H., Kang, E., Wang, S. Y., and Lee, K. H. (2002). A new segmentation method for point cloud data. *International Journal of Machine Tools and Manufacture*, 42(2):167–178.

Worden, K. (1997). Structural fault detection using a novelty measure. *Journal of Sound and vibration*, 201(1):85–101.

Wu, M. and Jermaine, C. (2006). Outlier detection by sampling with accuracy guarantees. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 767–772, Philadelphia, USA.

Xiao, J., Zhang, J., Adler, B., Zhang, H., and Zhang, J. (2013). Three-dimensional point cloud plane segmentation in both structured and unstructured environments. *Robotics and Autonomous Systems*, 61(12):1641–1652.

Xiao, J., Zhang, J., Zhang, J., Zhang, H., and Hildre, H. P. (2011). Fast plane detection for SLAM from noisy range images in both structured and unstructured environments. In *Proceedings of the International Conference on Mechatronics and Automation (ICMA)*, pages 1768–1773, Beijing, China.

Xu, H., Caramanis, C., and Mannor, S. (2010). Principal component analysis with contaminated data: the high dimensional case. In *Proceedings of the 23rd International Conference on Learning Theory (COLT)*, Haifa, Israel.

Yang, B., Fang, L., and Li, J. (2013). Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79:80–93.

Yang, D. O. and Feng, H.-Y. (2005). On normal vector estimation for point cloud data from smooth surfaces. *Computer-Aided Design*, 37(10):1071–1079.

Yang, W. S. and Wang, S. Y. (2006). A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications*, 31(1):56–68.

Yoon, M., Lee, Y., Lee, S., Ivrissimtzis, I., and Seidel, H.-P. (2007). Surface and normal ensembles for surface reconstruction. *Computer-Aided Design*, 39(5):408–420.

Zakšek, K. and Pfeifer, N. (2004). An improved morphological filter for selecting relief points from a LIDAR point cloud in steep areas with dense vegetation. Technical report, Delft Institute of Earth Observation and Space systems, TU Delft, The Netherlands.

Zhou, Y., Yu, Y., Lu, G., and Du, S. (2012). Super-Segments based classification of 3D urban street scenes. *International Journal of Advance Robotic Systems*, 9(1):1–8.

Zimek, A., Schubert, E., and Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387.

Zuliani, M. (2011). RANSAC for Dummies. http://vision.ece.ucsb.edu/ zuliani/research/ransac/docs/ransac4dummies.pdf, accessed 25-11-2011.

Note: *Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.*