Segmentation and Classification of Airborne Laser Scanner Data

George Sithole

Publications on Geodesy 59

NCG Nederlandse Commissie voor Geodesie Netherlands Geodetic Commission

Delft, May 2005

Segmentation and Classification of Airborne Laser Scanner Data
George Sithole
Publications on Geodesy 59
ISBN 90 6132 292 8
ISSN 0165 1706

Published by: NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission,
Delft, The Netherlands
Printed by: Optima Grafische Communicatie, Optima Graphic Communication, Rotterdam,
The Netherlands
Cover illustration: George Sithole

NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission
P.O. Box 5058, 2600 GB Delft, The Netherlands
T: +31 (0)15 278 28 19
F: +31 (0)15 278 17 75
E: ncg@lr.tudelft.nl
W: www.ncg.knaw.nl

The NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission is part of
the Royal Netherlands Academy of Arts and Sciences (KNAW)

Abstract

Segmentation and Classification of Airborne Laser Scanner data

George Sithole

Various methods have been developed to measure the physical presence of objects in a landscape with high positional accuracy. A new method that has been gaining popularity is Airborne Laser Scanning (ALS). ALS works by scanning a landscape (the collection of ground, buildings, vegetation, etc.,) in multiple passes. In each scan pulses of laser light are emitted from an airborne platform and their return time is measured, thus enabling the range from the point of emission to the landscape to be determined. The product of airborne laser scanning is a cloud of points in 3D space. ALS is capable of delivering very dense and accurate point clouds of a landscape in a relatively short time. However, in spite of the ability to measure objects with high positional accuracy, the automatic detection and interpretation of individual objects in landscapes remains a challenge. An example of just such a challenge is the classification of point clouds produced by ALS. The classification of ALS point clouds consists firstly in the labeling of points as either object or bare earth. The labeled object points are then further labeled as either building or vegetation. As a measurement technique ALS holds great promise and motivated by the desire to promote it, research has been conducted here to automate the detection of bare earth, buildings and vegetation in ALS point clouds.

Several algorithms have been developed to automatically detect the bare earth (the topsoil or any thin layering covering it) in ALS point clouds. They are generally referred to in the ALS community as filtering algorithms. An experimental study (conducted as part of this research) of filtering algorithms determined that in flat and uncomplicated landscapes (i.e., small to medium sized buildings standing well off a fairly flat ground) algorithms tend to do well. Significant differences in accuracies of filtering appear in landscapes containing steep slopes and discontinuities.

These differences are a result of the ability of algorithms to preserve discontinuities while detecting large objects. A solution for this problem was determined to lie in the segmentation of ALS point clouds. If segmentation can be achieved in such a manner that all bare earth points are gathered into the same surface segment and the points from each object are gathered into their own surface segments, then filtering can be done on the basis of surfaces rather then points. This should offer a more reliable classification since topological information can be used in addition to geometric information to classify surface segments. On the strength of the study a new segmentation based filtering algorithm was developed.

For the developed filtering algorithm a new segmentation algorithm was devised. The algorithm works by first slicing a point cloud into contiguous and parallel vertical profiles. This slicing is done in several directions. After the slicing the points in the profiles are segmented based on proximity. The segmentation of profiles yields line segments. Next, the line segments are linked together through their common points to obtain surface segments. The adjacencies of line segments in profiles are aggregated to determine the adjacency of surface segments (including overlapping surfaces). The adjacency of surface segments provides contextual information. This contextual information is used to associated particular spatial arrangement of surfaces with objects or the bare earth.

The new segmentation algorithm also has the advantage that it facilitates the detection of bridges. The experimental study also identified the ability to detect bridges as one means to improve the accuracy of filtering. Using the developed segmentation algorithm a novel bridge detection algorithm was developed. Line segments are essentially cross sections of surface segments. In the bare earth line segments that are above adjacent line segments (in the same profile) are potentially from bridges. Therefore, points that lie at the intersection of such line segments are also potentially from bridges. In this manner seed bridge points are identified. These seed points are then used to detect bridges. The algorithm has the advantage that it is able to readily detect curved bridges, bridges that do not have parallel sides and bridges that branch into parts.

The classification of buildings and vegetation uses geometric and radiometric characteristics determined for surface segments. A point cloud consisting of object points is segmented by the proximity of points. The n geometric and radiometric characteristics of surface segments are mapped into an n-dimensional feature space. Using a supervised classification the surface segments are classified as either building or vegetation.

The new algorithms were tested on real data and showed improvement over current algorithms, particularly in complex urban landscapes. It is envisaged that with the inclusion of external data, e.g., thematic maps, existing digital elevation models and infra red imagery, the classification accuracy of ALS point clouds can be further improved .

Samenvatting

Segmentatie en classificatie van Airborne Laser Scanner gegevens

George Sithole

Voor het vastleggen van de geometrie van een landschap - een deel van de aarde en alles wat zich daarop bevindt - is de populariteit van Airborne Laser Scanning (ALS) de laatste jaren sterk toegenomen. Met ALS wordt een landschap in stroken afgetast, waarbij korte pulsen laserlicht worden uitgezonden vanaf een vliegend platform. Het gereflecteerde signaal wordt opgevangen en de richting en looptijd van de puls worden gemeten. Hieruit wordt de afstand naar en de positie van een punt in het landschap afgeleid. Daarmee resulteert ALS in een 3D puntenwolk van hoge dichtheid en precisie die in een relatief korte tijd verkregen wordt. Automatische detectie en interpretatie van individuele objecten in het landschap - nodig vanwege de arbeidsintensieve handmatige verwerking van de grote hoeveelheden gegevens - vormt een grote uitdaging. Een voorbeeld van een dergelijke uitdaging is de classificatie van met ALS geproduceerde puntenwolken. Deze classificatie bestaat uit twee stappen: eerst worden de objecten gedetecteerd die zich op de aarde bevinden, dan worden deze objecten onderverdeeld in gebouwen en vegetatie. Het automatisch onderscheiden van gebouwen, vegetatie en de aarde waar zich deze objecten op bevinden is het onderwerp van het onderzoek dat in dit proefschrift beschreven wordt.

Vele algoritmen zijn ontwikkeld om de kale aarde - de aarde zonder gebouwen of vegetatie - uit een ALS-puntenwolk te extraheren, een taak die in de ALS-gemeenschap beter bekend staat als filteren. Een studie naar acht filteralgoritmen is uitgevoerd als onderdeel van het onderzoek met als doel het kwalificeren van de prestaties van deze algoritmen en het inventariseren van de aanwezige problemen bij het onderscheiden van objecten en kale aarde in een landschap. De test toonde aan dat in vlakke en eenvoudige landschappen (zoals kleine tot middelgrote

gebouwen in een nagenoeg vlak terrein) alle algoritmen voldeden. Significante verschillen deden zich voor bij landschappen met steile hellingen en discontinuteiten in het terrein. De oorzaak lag in het uiteenlopende vermogen van de algoritmen om de discontinuteiten in stand te houden bij het detecteren van grote objecten. Segmentatie van de ALS-puntenwolken is de oplossing voor dit probleem. Filtering kan gedaan worden op basis van vlakken in plaats van punten wanneer zodanig gesegmenteerd kan worden dat alle kale-aarde-punten verzameld worden in n vlaksegment en de punten van een bepaald object in een ander vlaksegment. Deze aanpak maakt een betrouwbaardere classificatie mogelijk omdat naast geometrische informatie ook topologische informatie gebruikt wordt. Daarom werd een nieuw filteralgoritme ontworpen gebaseerd op segmentatie om de problemen die zich voordeden in de genoemde studie te overwinnen.

Een nieuw segmentatiealgoritme vormt de basis voor het ontwikkelde filteralgoritme. Daarbij wordt de puntenwolk eerst in aaneensluitende parallelle profielen verdeeld en de punten gesegmenteerd op basis van nabijheid zodat lijnsegmenten ontstaan. Dit wordt herhaald voor verschillende windrichtingen waarna vlaksegmenten gecreerd worden door verschillend gerichte lijnsegmenten te combineren op basis van gemeenschappelijke punten. De nabijheid van lijnsegmenten in de profielen wordt vervolgens gebruikt voor het classificeren van de vorm van de vlaksegmenten. Deze vorm van de vlaksegmenten wordt dan weer gebruikt voor het classificeren van de vlaksegmenten als kale aarde of object.

Een voordeel van het nieuwe segmentatiealgoritme is dat het de detectie van bruggen mogelijk maakt. De eerder genoemde studie toonde aan dat filteralgoritmen significant verbeteren met de mogelijkheid bruggen te detecteren. Het ontwikkelde algoritme voor de detectie van bruggen kan als volgt geschetst worden. Punten in lijnsegmenten die hoger liggen dan naburige lijnsegmenten (in hetzelfde profiel) behoren mogelijk tot een brug. Behoort een dergelijk punt ook tot een kruisend lijnsegment met dezelfde eigenschap, dan wordt dit punt gebruikt als startpunt voor de brugdetectie. Het algoritme is in staat om gekromde bruggen te detecteren, bruggen zonder parallelle begrenzingen of bruggen die zich splitsen.

Het algoritme voor het onderscheiden van gebouwen en vegetatie maakt gebruik van een feature-based approach. Daarbij wordt een puntenwolk eerst gesegmenteerd op basis van de nabijheid van de punten. Vervolgens vormen n geometrische en radiometrische eigenschappen de n-dimensionale kenmerkenruimte (de zogenaamde feature space). Deze wordt gebruikt om met behulp van een supervised classification gebouwen en vegetatie te onderscheiden.

De nieuwe algoritmen zijn getest op echte data en lieten verbeteringen zien ten opzichte van bestaande algoritmen, met name in complexe stedelijke gebieden. Het is te verwachten dat met de toevoeging van externe gegevens (zoals thematische kaarten, bestaande digitale hoogtemodellen of infraroodbeelden) en het gebruik van ALS-gegevens waarbij de golfvorm van de gereflecteerde straling wordt geregistreerd, de nauwkeurigheid van de classificatie van ALS-puntenwolken verder zal verbeteren.

Acknowledgment

Arthur Ashe said "Success is a journey, not a destination". In this regard my journey of the last four years would not have been possible if I had not been accompanied by a number of people. They offered, guidance when the path was not clear, encouragement when the path was steep, and companionship whenever the journey became wearisome. For this I express to them all my gratitude.

I owe many thanks to my promoter and supervisor Professor George Vosselman for first offering me the opportunity to conduct this research and later guiding and focusing my ideas. Without his constructive criticism and technical advice this thesis and the research documented in it would not have been possible.

My colleagues in the section Photogrammetry and Remote Sensing, Sagi Filin, Ben Gorte, Norbert Pfeifer, Tahir Rabbani and Ildiko Suveg also proved to be a great source of technical advice. My discussions with them provided a lot of insight into many subjects related to my own work.

The development of the new classification algorithms benefited much from tests on different data sets. The following organizations and companies are thanked for providing data sets. EuroSDR[1] provided data from for the cities of Stuttgart and Vaihingen in Germany (data acquired by Blom Norkart Mapping AS[2]), TerraImaging provided data for the city of Nijmegen in the Netherlands and the Rijkwaterstraat provided data for the town of Wijhe in the Netherlands (data acquired by Fugro-Inpark[3]).

The ISPRS filter test would not have been possible without the help and co-operation of participants who took time from their schedules to filter the data sets. I wish to extend my gratitude to Peter Axelsson, Christian Briese, Maria Brovelli, Magnus Elmqvist, Norbert Pfeifer, Marco Roggero, Gunho Sohn, Roland Wack and Andreas Wimmer.

---

[1] European Spatial Data Research (formerly OEEPE) - `www.eurosdr.org/`

[2] Blom Norkart Mapping AS (formerly Fotonor AS) `http://www.fotonor.no/`

[3] Fugro-Inpark - `http://www.fugro-inpark.nl/`

I would like to thank Frank van den Heuvel, Norbert Pfeifer and Siyka Zlatanova for proof reading this thesis and offering their constructive comments. Also appreciated is Jasper van Loon's translation of the abstract (samenvatting) into Dutch.

There are other members of the section Photogrammetry and Remote Sensing that deserve mention for their help in solving many of my administrative problems. They are Inge Flim, Peter Plugers, Magda Boekee, Lidwien de Jong-Bakker, and Vera van Bragt.

In implementing software for the classification algorithms various software libraries were used. Sander Dijkman provided the initial code for the OpenGL graphical user interface. The graphical user interface toolkit (called XForms) was obtained from T. C. Zhao and Mark Overmars, and the 2-dimensional Delaunay triangulation library (called Triangle) was obtained from Jonathan Richard Shewchuk[4]

Being away from home (Zimbabwe) proved much more difficult than first anticipated. But, it has been my fortune to have had friends who made me feel at home away from home. They are my former colleagues Siyka Zlatanova, Sagi Filin and Sander Oude Elberink, my former house mates Frans Tilma, Reinier Ringers, and Joris Gernaey, and finally but not least Rami Barends.

---

[4] Jonathan Richard Shewchuk - `http://www-2.cs.cmu.edu/~quake/triangle.html`

Contents

Abstract/Samenvatting **i**

List of Figures **iii**

List of Tables **vii**

1 Introduction **1**
1.1 Digital terrain acquisition 1
1.2 Scope of Research 2
1.3 Research method and objectives 4
1.4 Contribution to knowledge 6
1.5 Outline of the thesis 7

2 ALS and Filtering Algorithms **9**
2.1 Introduction 9
2.2 Airborne Laser Scanning 9
2.3 Filtering algorithms 13
2.4 Discussion 28

3 ISPRS Filter Test **33**
3.1 Introduction 33
3.2 Test Data 34
3.3 Evaluated filter algorithms 36
3.4 Result of Comparisons 36
3.5 Discussion 47

4 Filter Design **51**
4.1 Introduction 51
4.2 The weakness of current assumptions 52
4.3 Formulating new assumptions 57
4.4 Algorithm framework 62

5 Segmentation **65**

5.1 Introduction 65
5.2 Previous work 67
5.3 New segmentation strategy for object and bridge detection 71
5.4 Examples 85
5.5 Discussion 89

6 Classification 93
6.1 Introduction 93
6.2 Detecting macro objects 93
6.3 Detecting bridges 103
6.4 Detecting micro objects 112
6.5 Detection of man made and natural objects 113
6.6 Discussion 118

7 Results and Quality Analysis 119
7.1 Introduction 119
7.2 Assessment against ISPRS filter test data 120
7.3 Wijhe - High density data with reflectance 130
7.4 Nijmegen - Medium density data with RGB attributes 134
7.5 Nijmegen data - overpass 138
7.6 Using first pulse returns to detect natural objects in last pulse returns 139
7.7 Discussion 140

8 Conclusion and future work 143
8.1 Conclusion 143
8.2 Future work 144

Bibliography 147

A Analysis of point and segment attributes 155
A.1 Wijhe Point Attribute Characteristics 155
A.2 Nijmegen Point Attribute Characteristics 158

B Algorithm Parameters 163
B.1 Macro object detection parameters 163
B.2 Bridge detection parameters 167
B.3 Micro object detection parameters 168
B.4 Man made and natural object detection parameters 170

C ISPRS Filter Test Results 173

D Example of macro object and bridge detection 181

List of Figures

2.1 Airborne laser scanning. 12
2.2 Shaded relief made from a point cloud 13
2.3 Morphological algorithm. 14
2.4 Surface fitting 16
2.5 Weight function 17
2.6 Axelsson's progressive TIN densification algorithm. 18
2.7 Slope based algorithms. 19
2.8 Concept of an edge based algorithm. 22
2.9 Reasoning in horizontal slices. 26
2.10 Scan labeling algorithm 28

3.1 Two sites used in the ISPRS test 35
3.2 Examples of difficult scenarios for filtering 37
3.3 Sample data for quantitative comparison and assessment 41
3.4 Quantitative comparison of site 22 42
3.5 The effect of landscape on filter performance 45
3.6 Type I and Type II errors of Site 1 vs. resolution. 47

4.1 Neighborhoods that contain objects and bare earth, and neighborhoods that contain only bare earth are homeomorphic 53
4.2 Neighborhoods that contain objects and bare earth, and neighborhoods that contain only objects are homeomorphic 54
4.3 Vertical and lateral separation of points 55
4.4 Reduction of point spacing in strip overlaps 56
4.5 Zero ground returns. 57
4.6 A conceptual and logical representation of the real world 58
4.7 Extended logical representation. 60
4.8 A simple bridge. 62
4.9 Overall filtering flow 63

5.1 Region growing 70
5.2 Surface representation using planar curves 73

5.3 Segmentation by profile intersection. 74
5.4 Segmenting a point cloud. 75
5.5 Overlapping surfaces segmented 75
5.6 Labeling by curve fitting 79
5.7 Consecutive labeling 80
5.8 Medial axis between curves 81
5.9 Labeling by crust 82
5.10 Labeling by proximity 83
5.11 Minimum spanning tree 84
5.12 Segmentation example 1: Building on slope. 86
5.13 Segmentation example 2: Bridge connected to the bare earth. . . . 87
5.14 The slope of a bridge edge to adjacent bare earth. 88
5.15 Segmentation example 3: Flyovers. 88
5.16 Line shapes 90
5.17 Surface shapes 91

6.1 Topological ordering of a landscape 94
6.2 Tiered object. 95
6.3 Macro object detection. 97
6.4 Macro object extraction. Iteration 1 98
6.5 Macro object extraction. Iteration 2 99
6.6 A surface as a combination of different shapes. 100
6.7 Shape from orientation 101
6.8 Conventional bridges, interchanges and flyovers 103
6.9 Bridges and tunnels 104
6.10 Bridge and small object. 105
6.11 Bridge detection. 108
6.12 Bridge detected. 109

7.1 Total errors(%) over the 15 samples 120
7.2 Type I errors over the 15 samples 121
7.3 Type II errors over the 15 samples 122
7.4 Sample 11 123
7.5 Sample 12 123
7.6 Sample 21 123
7.7 Sample 22 124
7.8 Sample 23 124
7.9 Sample 24 125
7.10 Sample 31 125
7.11 Sample 41 126
7.12 Sample 42 126
7.13 Sample 51 127
7.14 Sample 52 128
7.15 Sample 53 128

7.16 Sample 54 . . . 129
7.17 Sample 61 . . . 129
7.18 Sample 71 . . . 129
7.19 Wijhe test . . . 131
7.20 Nijmegen test . . . 136
7.21 Bridge detection . . . 138
7.22 Using first pulse returns to classify vegetation in last pulse returns. 140
7.23 A possible use of first pulse data . . . 141

A.1 Wijhe point characteristics. . . . 156
A.2 Scatterplots - Wijhe. . . . 157
A.3 Nijmegen data characteristics. . . . 158
A.4 Scatterplots - Nijmegen. . . . 160
A.5 1D Plots for segment characteristics - Nijmegen. . . . 161

B.1 The choice of slope threshold . . . 167

List of Tables

3.1 Meaning of Good, Fair and Poor (used in Table 3.2) 39
3.2 Qualitative comparisons of filters 40

6.1 Example of a set of associations. 102

7.1 Summary of Wijhe results 130
7.2 Summary of Nijmegen results 135

B.1 Algorithm Parameters: macro object detection 166
B.2 Algorithm Parameters: bridge detection 169
B.3 Algorithm Parameters: micro object detection 169
B.4 Algorithm Parameters: man made and natural object detection . . 172

Chapter 1

Introduction

1.1 Digital terrain acquisition

Knowledge of the geometric character of the bare earth is essential to many civil design and planning applications. This knowledge is acquired by sampling a landscape and building digital terrain models from the samples. Samples from a landscape are commonly acquired by the techniques of field survey, photogrammetry, satellite remote sensing, InSAR and in recent years Airborne Laser Scanning (ALS).

Before the sampled data can be used for modelling it has to be preprocessed. The above sampling techniques (for the exception of field survey) sample not only the bare earth, but also the objects (buildings, trees, etc.,) residing on it. The preprocessing, or filtering as it is commonly known, is meant to distinguish between samples that have come from objects and those that have come from the bare earth. Filtering is a non-trivial and important procedure, because the quality of filtered data has a direct impact on the quality of modelling. Put differently, errors in the filtered data are carried over into the digital terrain modelling.

The data acquired by ALS is a cloud of points in a three dimensional reference frame, where each point is a sample from a scanned landscape. When this data is first acquired, it has to be preprocessed. This preprocessing includes tasks such as *modelling of systematic errors*, *filtering*, *feature detection and extraction*, *quality control* and *packaging* [1].

The acquired data normally contain millions of points. Because of this the processing

---

[1] Packaging involves data reduction and the formatting of the data so that it can be used in further processes such as volume computations, DEM generation, and object modelling.

ing and packaging of ALS point clouds has to be highly automated. Therefore, in the context of this thesis any processing of ALS data will from here on be understood as being either semi-automatic or fully automatic.

Of the above mentioned tasks, manual classification (including filtering) and quality control pose the greatest challenges, consuming an estimated 60 to 80% of processing time (Flood, 1999, 2001b,a), thus underlining the importance and motivation of research into filtering algorithms.

1.2 Scope of Research

Motivated by the importance of filtering and the importance of bare earth models for many engineering applications, the work here is focused on devising semi-automatic and automatic algorithms for filtering ALS point clouds with a high degree of accuracy and reliability.

Manual filtering

Although impractical, filtering of ALS data could be done manually. Manual filtering employs the human cognitive process to readily distinguish between objects and the bare earth in the ALS data.

ALS data, usually, do not have associated with them semantic information (strength of reflectance data excluded). Therefore, the height of points and their spatial relationship becomes the only means of classifying terrain and non-terrain points. However, because of our knowledge of the real world, as humans we are able to associate this spatial relationship between points in laser data point clouds with features in the real world, even though semantic information may be lacking. The human cognitive process also permits us to filter different types of landscapes and identify large and small objects with ease. Furthermore, given auxiliary information such as airborne imagery we are able to assimilate that information to aid our cognitive process and enhance the reliability of filtering. Besides cognition, humans also rely on intuition, allowing them to make guesses as to the nature of an object even when evidence to support the identity of an object is minimal.

Once objects have been identified, they can be removed, thus leaving behind the bare earth. The problem with manual filtering is that because of the large volume of data being handled, it is very time consuming. To accelerate manual filtering special tools to aid classification in operator selected regions can be used. However, such semi-automatic solutions are useful only where the size of the ALS data is relatively small, and besides they still require operator supervision. To overcome the problem of filtering large volume data automatic filters are required.

Automated filtering

A number of automatic filtering algorithms have been developed. Most of them share a common weakness in that they don't work for all types of landscape and because of this, their behaviour is unpredictable. Three reasons can be offered for this lack of universality and reliability.

The first cause of failure lies in the assumptions employed by filter algorithms. A filter is designed with the belief that there are certain characteristics that distinguish the bare earth from objects. There are many assumptions that can be used but unfortunately, assumptions are not foolproof and are landscape specific. For example, one assumption used is that the bare earth does not contain gradients greater than some predefined minimum value. Naturally, filters designed with this assumption cannot be used in steeply sloped landscapes.

Secondly, algorithms do not consider the context of point(s) in relation to their neighborhoods. Current automatic filters classify a set of points based on their functional relationship to neighboring points, but not on the meaning of the points in relation to the form of the neighborhood. For example, a point on a building is compared to neighboring points to determine its classification. However, those neighboring points may themselves be points on a building, in which case any classification of the point will be unreliable.

The third reason is that most filters only use the positional information of points in a point cloud. External information (particularly radiometric and topological) needs to be used to reinforce the filtering process where the classification of features is in doubt. Sources of external information include aerial images and existing maps.

The work described here is aimed at overcoming the weaknesses described above.

General Assumptions about the data

ALS systems do not use the same scanning mechanism. Therefore, the spatial distribution of points in the data can differ. To ensure that the developed algorithm is as general as possible the following assumptions have been made about the data:

1. *Assumption* - All points in the data are free of systematic errors. Some laser scanner data contain very low and very high points (blunders). Therefore, blunders will be treated because they can cause filtering and thinning strategies to critically breakdown.

2. *Assumption* - The spatial distribution of points is anisotropic (does not have the same properties in all directions) and the spacing of points is not uniform.

3. *Assumption* - GPS, INS and other navigation information for all the scans are not available. Because of this, the developed algorithm should be blind to the manner in which the point cloud was formed, and potentially making it applicable to point clouds acquired by techniques other than ALS.

1.3 Research method and objectives

The objectives of the research can be stated as follows: Develop an automated or semi-automated classification algorithm that when applied to ALS data, will discriminate bare earth, bridge, building and vegetation points. Moreover, the algorithm should work for all landscape types, and should operate on raw ALS data.

To achieve the objectives, the research was subdivided into three parts, (i) literature study of current filtering algorithms, (ii) experimental test of current filtering algorithms (iii) design, development and test of a new algorithm.

Study of current filtering algorithms

This first part of the research involved a literature review of current filtering algorithms. The main aim was to understand the basic concepts of algorithms and the assumptions upon which they are based. Filters differ because designers choose to use different mathematical concepts to describe the landscape, and driven by scientific endeavour, designers are continually challenged to develop more novel techniques. They also differ on the assumptions made about the characteristics that distinguish the bare earth from other objects. This part of the research is covered in chapter 2.

Test of current filtering algorithms

While literature studies provide insights into the concepts of algorithms, true knowledge of the operations of filters can only be gained by applying them to real data. For this purpose an experimental study of filter algorithms was carried out under the auspices of ISPRS Working Group III/3 (Sithole and Vosselman, 2002b) with a desire to, (i) further the knowledge of filtering algorithms, (ii) determine the comparative performance of existing filter algorithms, (iii) determine the influence of the point density of ALS data on filter performance, and (iv) determine filtering problems that remain unsolved.

Sample ALS data sets were chosen and offered to the ALS community for filtering. These sample data sets were also manually filtered to serve as control data. Eight developers participated in the test. The participant's filter results were compared against the control data and the performance of the algorithms was assessed. The results of the assessments were later published, (Sithole and Vosselman, 2003b, 2004). The results of the tests are described in chapter 3.

Design and development of a new algorithm

The filter test identified problems in filtering that remained unsolved. It was deemed that even with modifications current filter algorithms would not be able to reliably overcome the problems. Therefore, a new filter algorithm was developed. This algorithm was developed in three steps, (i) design, (ii) segmentation, and (iii) classification.

Filter design

The question of what constitutes the bare earth was revisited, with the aim of identifying basic assumptions that should be used, and how they should be used in combination. This question of assumptions is treated in chapter 4.

Segmentation of ALS data

One problem with current filters is that they are based on point neighborhood tests, and do not study the context of the neighborhoods. Because of this it was resolved that the new filter should work on global geometric structures, or segments, and filtering should be done based on the relationship between these structures. Therefore, the new filter algorithm first segments the ALS data and then classifies the segments. Some segmentation strategies have been tried by Brovelli et al. (2002); Filin (2002); Lohmann (2000) and Roggero (2002) but they are ill suited to point clouds that contain overlapping surfaces. Because of this a new segmentation approach based on intersecting profiles has been proposed, and is discussed in chapter 5.

Classification of ALS segments

In line with the objectives of the research, the primary aim of classification is to identify bare earth and object segments. However, segmentation of the data offers the opportunity to identify other features in the data. In this regard, the research aimed to identify three further types of features in a landscape, i.e., bridges, man made objects and natural objects.

Bridges and overpasses are of interest because their classification can be bare earth or object depending on the application of the data. Current filter algorithms do not explicitly filter bridges. Therefore, the ability to filter bridges would enhance the reliability of filtering.

Each point in an ALS data may carry a reflectance, or RGB signature. The point signatures can be used in the segmentation or they can be aggregated to yield signatures for the segments. Furthermore, some ALS systems are capable of recording multiple pulse returns that can be used to assist in differentiating between vegetation and the bare earth. In the work here, the signatures are aggregated for the segments and used as additional information to distinguish between man made and natural objects.

The classification of segments is treated in chapter 6.

Assessment

To gauge the effectiveness of the new filter, the filter is applied to the test data used in the experimental study of filter algorithms. The results are presented and discussed in chapter 7. Furthermore, issues related to the quality of filtered data are discussed.

1.4 Contribution to knowledge

The research done in the development of a new filtering algorithm contributes to the field of ALS in the following ways:

1. *Development of the filtering problem* - A more elaborate definition of the landscape has made it possible to better define the filtering problem.
2. *ISPRS Filter test/assessment of current filtering strategies* - An experimental comparison of filter algorithms was not available to the ALS community. The ISPRS filter test has partly filled this gap. Furthermore, the comparisons and reference data generated are available to filter developers, thus providing them with a benchmark to test and improve their algorithms.
3. Segmentation algorithm - The standard approach to filtering is to use triangulated irregular networks (TINs). The approach taken here was to use a scan line based algorithm. The scan line algorithm was extended in two ways, (i) most scan line algorithms work on range images, the approach here works on a point cloud (ii) scan line algorithms only work on parallel scans running in a single direction and compare adjacent scan lines, the approach

here has scans running in more than one direction and intersecting scans are compared. This segmentation algorithm can easily be adapted for use in modeling data other than ALS data.

4. Classification of bridges - Most of the current filtering algorithms distinguish between objects and the bare earth. The new filtering algorithm goes beyond this and also identifies bridges.

1.5 Outline of the thesis

The thesis is organized into eight chapters. The contents of chapters two to seven have been described above. In chapter 8 the objectives and results of the research are summarized and aspects of the research that require further work are outlined. Some of the work presented in this thesis is partly drawn from several papers published since 2001, Sithole (2001); Sithole and Vosselman (2002a, 2003a,b, 2004).

Chapter 2

ALS and Filtering Algorithms

2.1 Introduction

The purpose of this chapter is to lay a foundation for discussing filter design and development in succeeding chapters. Firstly, ALS and its core principles are discussed and from there the main filtering algorithms that have been developed to date are described.

2.2 Airborne Laser Scanning

Photogrammetry and satellite remote sensing, are amongst the disciplines that have contributed the most to developments in mapping. Together they have made it possible to map the relief of landscapes at large scales, at sub-meter accuracies and in relatively short times.

Despite all the advances that have already been made, faster, cheaper, and more accurate digital terrain techniques are still being sought. In the last ten years, ALS has emerged as a strong complimentary tool for large scale 3D abstraction of landscapes. Compared to photogrammetry[1] ALS is an active system, capable of delivering very dense (1 point/m^2 or higher) and relatively accurate point clouds

---

[1] Some comparisons between photogrammetry and ALS can be found in Ackermann (1999); Baltsavias (1999c); Schenk (1999); Kasser and Egels (2002). Some comparisons between remote sensing and ALS for applications in forestry can be found in Hyyppa and Hyyppa (1999).

(0.15-0.25m in elevation and 0.3-1m in planimetry for flying heights below 2km) of the terrain in a relatively short time. It is made more attractive because:

- it can be used at any time of the day,
- it penetrates through forest canopies,
- it readily abstracts digital landscapes in digital format.

Although the first optical laser was developed by Hughes aircraft in the 1960s, the application of lasers to earth measurement only started in the 1970s when they were used in Airborne Profile Recorders (APR). These systems obtained profiles of the landscape by measuring the vertical range from an airplane to the landscape. The position of the airplane was determined by means of photographs taken during the flight and in turn positions of points in the landscape were determined. More detail on APRs can be found in Lindenberger (1991, 1993). Because the position of an airplane was determined using photographs, the planimetric precision of the measured points suffered. This lack of precision made APRs obsolete. However, with the advent of commercial GPS in the 1980s interest in the use of lasers for digital landscape acquisition was renewed. This interest has led to the advancement and promotion of ALS to the extent that current systems provide point clouds with accuracies measured in centimeters and point spacings measured in decimeters. Commercial use of ALS has expanded considerably and there are now currently (year 2004) 88 companies involved in the manufacture and application of laser systems, worldwide. Furthermore, there are 16 different scanner systems available on the market[2]

Currently ALS is being used for topographic mapping, vegetation mapping (forests, flatlands, etc.,), corridor mapping (roads, railways, power lines, etc.,), urban modelling (cityscapes, wireless communications, etc.,), engineering works (volume computations, etc.,), and coastal engineering and management.

As a discipline, ALS is still evolving and as it is combined with imaging technologies, the range of applications and products will increase. This is a major motivating factor for research in this area.

Principle of ALS

The concept behind ALS is to obtain numerous scans of a landscape from single flight line/s and then aggregate these scans to obtain a discrete model of the

---

[2] More information on developments in ALS, and current state of the art can be found in Wever and Lindenberger (1999) and further, details on the projected growth of ALS can be found in Flood (1999, 2001b,a). More information on system developers can be found in Baltsavias (1999b); Airborne1 (2004).

landscape, figure 2.1. The scans are usually linear and orthogonal to the line of flight. The relevant aspects of ALS shall now be briefly outlined [3].

Components of an ALS system

The major components of an ALS system are (Baltsavias, 1999b):

1. LRF (Laser Range Finder) - for laser emission and detection, and range determination
2. GPS (Global Positioning System) - scan point position determination
3. IMU (Inertial Measurement Unit) - for attitude determination
4. Computer - control of the on-line data acquisition
5. Data Storage Unit - data storage
6. CCD cameras and other sensors for imaging the scanned landscape (these are optional)

These components are all mounted on an aircraft or helicopter platform. The main technical parameters of ALS can be found in Baltsavias (1999b).

Point measurement: In an ALS system, the return time of an emitted laser pulse from surfaces in a landscape are used to measure the range from the point of emission (on an airborne platform) to the landscape. The round trip time of the pulse is measured by electronically analyzing the waveform of a returned pulse [4]. The range from the point of emission (pulse) to the landscape can now be obtained by multiplying the speed of light with half the return time. An array of range measurements, typically linear, is called a scan.

Because the range and the position of pulse emissions and the attitude of the line of sight are known by lidar, GPS and IMU respectively, the position of points in the landscape can be determined in a 3D frame. Ranges within a scan are measured at rates upwards of 5kHz. Current state of the art systems are able measure at rates of about 100kHz.

Scan characteristics: The spacing between points depends on the measurement rate, the scan angle, flying height and aircraft speed. The scan angle (the angle subtended by the two furthest ends of a scan) ranges from 1° to 65°. Flying heights

---

[3] More details on the operational specifics of ALS are described by Baltsavias (1999a); Lohmann and Koch (1999); Wehr and Lohr (1999).

[4] A more detailed explanation of this analysis can be found in Katzenbeisser (2003).

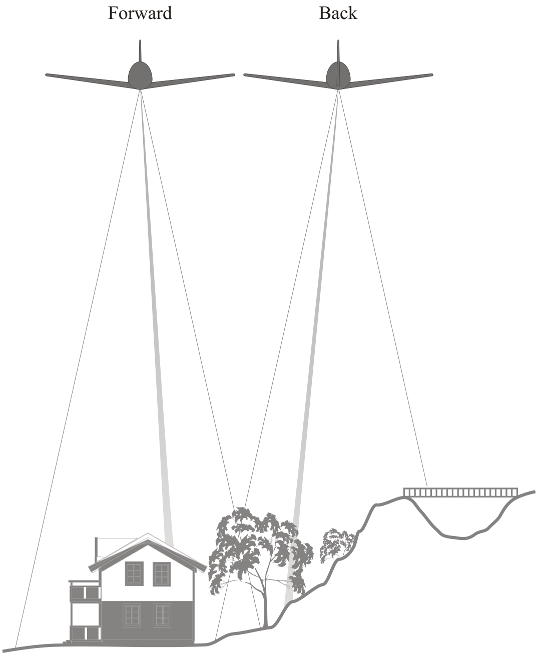


Figure 2.1: Airborne laser scanning. The landscape is scanned in strips, and the scans from the strips are combined to form a point cloud.

normally range from 100m to 1km, although new systems can be used at heights up to 3km. Therefore, point spacing can range anywhere from 0.1m to 5m.

Point characteristics: Because objects on the ground differ in material composition and height, the signal strength of the reflected pulse (i.e., the echo of the emitted pulse) is also recorded. Several reflections of a pulse maybe detected. The first reflected pulse is assumed to contain more hits off vegetation than the second pulse. Therefore, first pulse returns are used in orthophoto production and forestry and vegetation inventory applications, while second pulse returns are used for bare earth measurement applications.

Just as the return waveform is used to measure the return time of a pulse, most systems also use it to measure the strength of the returned pulse. The materials on the landscape have different spectral characteristics and because of this, a low resolution image of the landscape can be obtained from the strength of the returned pulse. Typically, the radiation used in lidar is in the IR part of the EM spectrum. Therefore, materials like vegetation will tend to appear bright, earth and asphalt will appear dark, and deep-water bodies will absorb radiation. Because of this

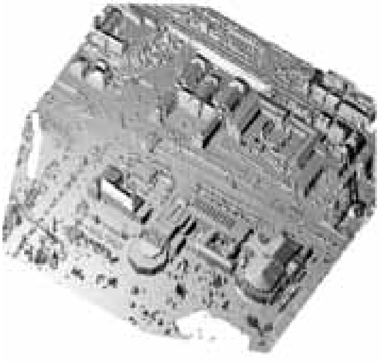


Figure 2.2: Shaded relief made from a point cloud

reflectance can be used to some extent for classification. Some ALS systems also capture imagery during scanning. Therefore, an RGB triplet can also be associated with each point.

Point cloud: When all the scans are aggregated a cloud of points in a 3D reference frame is obtained, e.g. figure 2.2. The most notable characteristic of ALS point clouds is that they are large and dense, often containing millions of points.

2.3 Filtering algorithms

The rest of this chapter deals entirely with the separation between objects and the bare earth. The separation between natural and man made objects shall be treated in chapter 6.

A number of algorithms have been developed for semi automatically/automatically extracting the bare earth from point clouds obtained by ALS and InSAR. While the mechanics of some of these algorithms have been published, those of others are not known because of proprietary restrictions. Some comparison of known filtering algorithms and difficulties have been cited in Huising and Pereira (1998); Haugerud and Harding (2001); Tao and Hu (2001).

For the purpose of discussion a point cloud, V, will be treated as a set of attributed points in three-dimensional space. Where, v, is an attributed point with coordinate triplet x and attribute a.

$$V = \{v | v(x \in \mathbb{R}^3, a \in \mathbb{R})\} \tag{2.1}$$

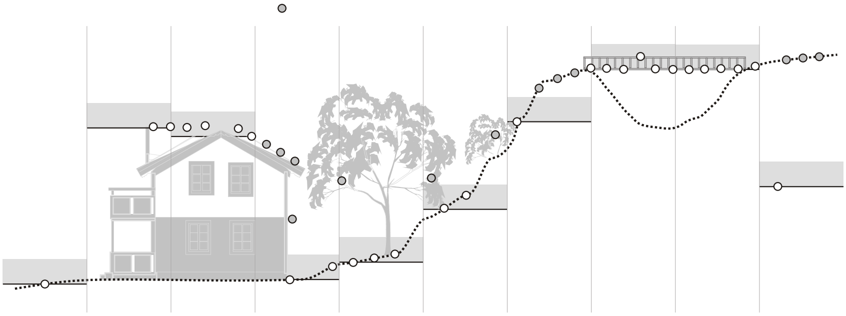


Figure 2.3: Morphological algorithm. White circles are classified as bare earth and gray circles as object.

The attribute of a point can take on two values, 0 or 1, denoting object and bare earth respectively. Filtering is the removal from V of points with label 0. The removal of zero labeled points yields the set of bare earth points B. Therefore, before filtering can be done the ALS points have to be attributed or labeled.

Morphological filter - J. Lindenberger: 1993

Assumption - The lowest points in a neighborhood belong to the bare earth.

This filter is based on the concepts of mathematical morphology, which is a set-theoretic method of image analysis providing a quantitative description of geometrical structures. The algorithm is applied to ALS scans.

In this algorithm (Lindenberger, 1993; Petzold et al., 1999), first a rough terrain model is calculated by the lowest points found in a moving window of rather large size. All the points with height difference exceeding a given threshold are filtered out, and a more precise DTM is determined. Figure 2.3 shows an example of how the algorithm works. This step is repeated several times, reducing the window size every time. The result is influenced by the final window size and the final threshold below which points are expected to be terrain points. A small window size leads to points on large buildings remaining in the file of the so-called ground points. A large window size smoothes the terrain and removes discontinuities. A high threshold value that is accepted in the final step leads to many vegetation points classified as ground points, and a small threshold again removes small terrain discontinuities. The parameters depend on the morphology of the terrain and have to be different for flat, hilly and mountainous regions.

Another variation on this algorithm is presented by Kilian et al. (1996). In this variation, the window size is increased with every iteration. Furthermore, in each iteration points that fall within the height threshold are assigned a weight directly proportional to the size of the window (points on flat ground are rewarded). Finally, points with the largest accumulated weight over all iterations are classed as bare earth and the remaining points are classed as object.

Hierarchical surface regularization - K. Kraus, N. Pfeifer, C. Briese: 1998

Assumption - The bare earth is a patchwork of piecewise continuous overlapping surfaces.

In this algorithm (Kraus and Pfeifer, 1998; Pfeifer et al., 1998, 1999; Pfeifer and Stadler, 2001; Kraus and Pfeifer, 2001; Briese and Pfeifer, 2001; Briese et al., 2002) the derivation of the terrain as well as the classification of the original points is performed in a hierarchic method. In each hierarchy level robust interpolation for the classification of the points and the surface derivation is done (figure 2.4). A rough approximation of the terrain, f, is first computed using the points of the respective hierarchy level. The vertical distance (residual) of the points to this approximate surface, f, is then used in a weight function (figure 2.5) to assign weights to all points. Points above the surface are given a small weight and those below the surface are given a large weight. The surface, f, is then recomputed using kriging considering the individual weights. In this way the recomputed surface, f, is attracted to the low points.

The process is iterated until a certain number of iterations have been reached or the computed surface does not change significantly between iterations, shown in figure 2.4. On completion of the iterations, a point is labeled based on its height above (or below) the surface, f. The labeling function is given by:

$$\phi_{N,\varepsilon}(v_i) = \left\{ \begin{array}{lll} 0 & v_i \in N & |f(v_i) - h(v_i)| > \varepsilon \\ 1 & else & \end{array} \right\} \tag{2.2}$$

Where N is a neighborhood over which f is continuous, $h(v_i)$ is the height of point v_i, and ε is a predefined threshold.

This robust interpolation has been extended to the hierarchic robust interpolation (Pfeifer and Stadler, 2001). It works in a coarse to fine approach using data pyramids (i.e. using coarser and coarser selections of the original points at higher pyramid levels). Starting with the coarsest/highest level of points the robust interpolation is applied. To move from one level to the next finer/lower one, the surface of the coarser level is compared to the points of the finer level. Those within a predefined threshold are selected and are the input for the robust interpolation

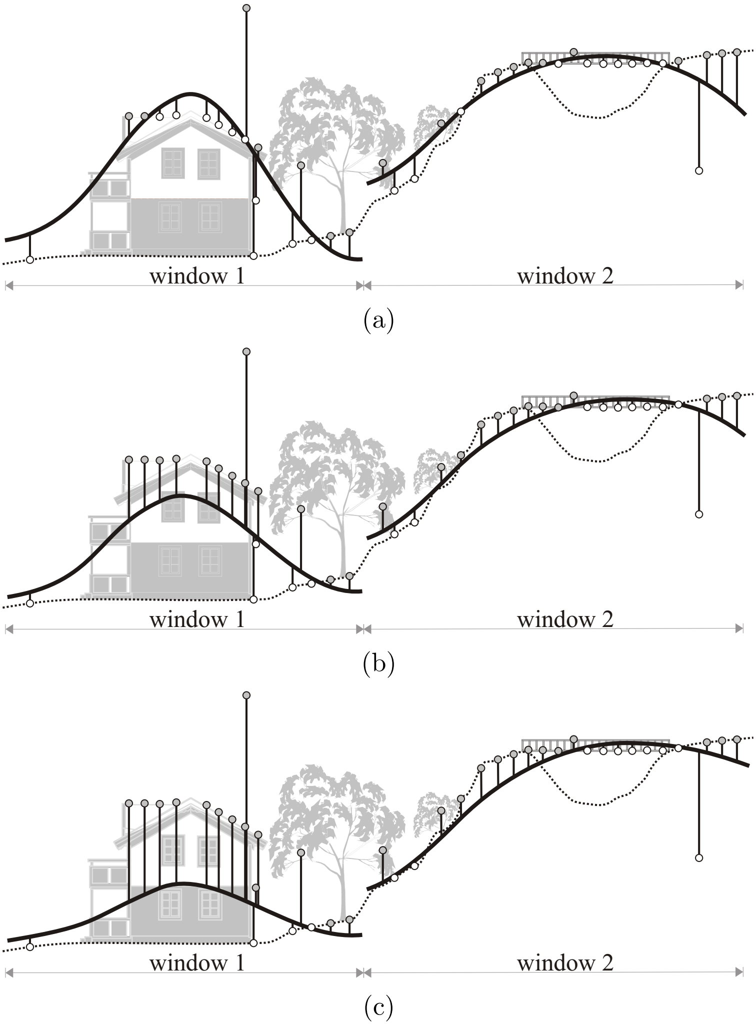


Figure 2.4: Surface fitting, (a) First fit, (b) First iteration, (c) Second iteration

on the next finer level.

A characteristic of the bare earth is that it contains discontinuities (break-lines). These cannot be modeled with smooth surfaces. Therefore, Briese and Pfeifer extended the algorithm to handle break-lines (Briese and Pfeifer, 2001).

A variant on this algorithm has been developed by Schickler and Thorpe (2001). Their algorithm uses break-lines, curvature constraints and slope constraints to

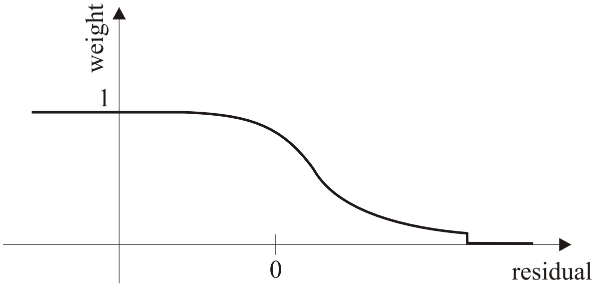


Figure 2.5: Weight function

control the estimated surface. Additional to this they input a classification map (vegetation types, water bodies, urban areas, etc.,) into their algorithm and associate with each class type a parameters set ideal for that class type.

Progressive TIN densification - P. Axelsson: 1999

Assumption - The bare earth is locally and globally flat.

A sparse TIN, G, is derived from neighborhood minima. This TIN is a first approximation of the bare earth, figure 2.6(top). In iterative steps this TIN is progressively densified to the laser point cloud, figure 2.6(bottom). In each iteration a point is added to the TIN if the point meets certain criteria in relation to the triangle that contains it. The criteria are that a point must be within a minimum distance to the nearest triangle node and the angle between the triangle normal and the line joining the point and node must be above a given threshold. At the end of each iteration the TIN and the data-derived thresholds are recomputed. New thresholds are computed based on the median values estimated from the histograms at each iteration. Histograms are derived from the angle points make to the TIN facets and the distance to the facet nodes. The iterative process ends when no more points are below the threshold.

The labeling function for this algorithm is:

$$\phi(v_i) = \left\{ \begin{array}{ll} 1 & v_i \in TIN \\ 0 & else \end{array} \right\} \tag{2.3}$$

Where TIN is the triangulation obtained after the final densification.

The main strength of this algorithm lies in its ability to explicitly model surfaces with discontinuities, which is a particularly useful characteristic in urban areas.

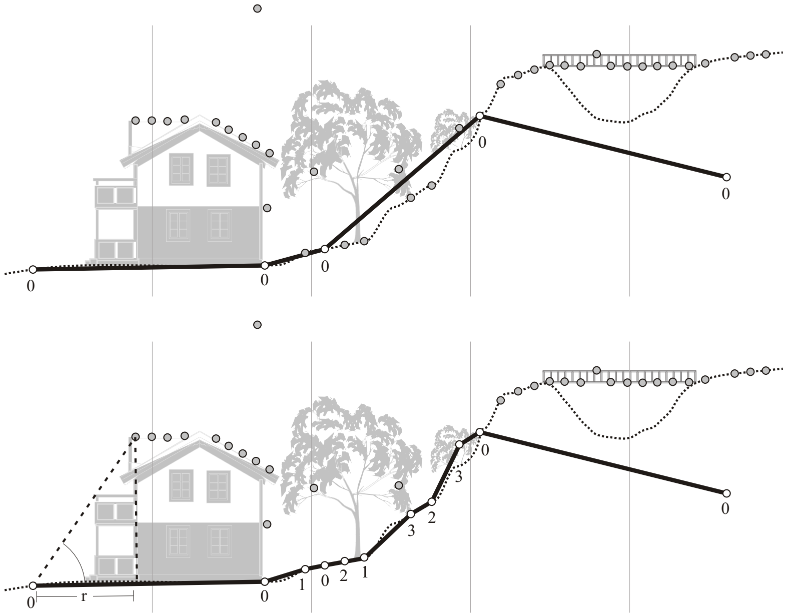


Figure 2.6: Axelsson's progressive TIN densification algorithm. First approximation of the bare earth based on neighborhood minima(top), and subsequent approximations of the bare earth based on angle, θ and distance to nearest node, r (bottom).

Further details can be found in (Axelsson, 1999, 2000, 2001). A variant on this algorithm has been developed by Voegtle and Steinle (2003).

Morphological filter/Slope based filter - G. Vosselman: 2000

Assumption - Gradients in the bare earth are bounded.

This filter is also based on the concepts of mathematical morphology. This filter, designed by Vosselman (Vosselman, 2000; Vosselman and Maas, 2001), approximates the local geometrical structure of the bare earth using a structuring element usually in the form of an inverted funnel of radius, r (but also an inverted cone sometimes). The structuring element is a hypothesis on the maximum height difference between any two points on the bare earth with respect to the distance between them.

A structuring element is centered (planimetric) on a point, and then raised un-

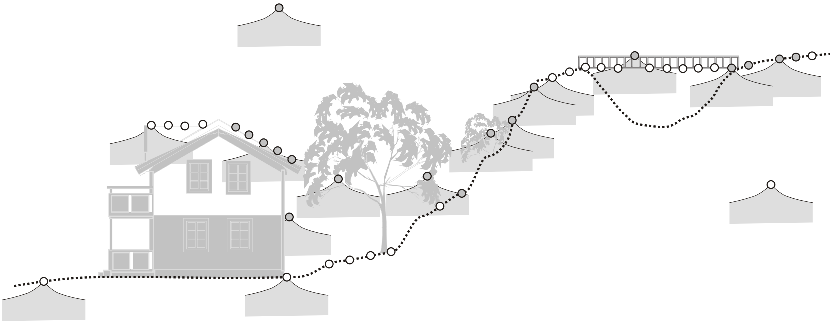


Figure 2.7: Slope based algorithms. The structuring element is an inverted cone.

til it touches the point. After that, if there are no neighboring points beneath the structuring element then the point is accepted as bare earth, otherwise it is accepted as object. The neighborhood function is given by:

$$\phi_{N,r}(v_i) = \left\{ \begin{array}{ll} 0 & \forall v_j \in N \quad \exists \quad \Delta h(v_i, v_j) > \Delta h_{max}(d(v_i, v_j)) \\ 1 & else \end{array} \right\} \tag{2.4}$$

Where $\Delta h(vi, vj)$ is the height difference between a point, v_i, and another point v_j in its neighborhood, and $\Delta h_{max}(d(v_i, v_j))$ is the maximum expected height difference between two points in the bare earth at planimetric distance d apart. The structuring element is essentially the neighborhood function for values of d ranging from 0 to r.

The algorithm works by applying the structuring element at every point in the point cloud. Furthermore, the structuring element applied at each point is the same. Figure 2.7 shows an example of how the algorithm works. The structuring element can be tuned using a training set containing only bare earth points. Histograms are generated for points at distance, d, apart. From these histograms, a maximum height distribution is obtained, and this becomes the structuring element.

Adaptive slope based filter - G. Sithole and G. Vosselman: 2001

Assumption - Gradients in the bare earth are locally bounded.

This filter (Sithole, 2001) is a variant on the slope based filter developed by Vosselman. In Vosselman's filter the structuring element applied at every point is the same. This limits the filters application in landscapes where the bare earth is steep. To improve the performance of the algorithm in steep slopes, in this filter, the shape of the structuring element is altered in tune with the slope of the bare earth.

The neighborhood function 2.4 now becomes:

$$\phi_{N,r}(v_i) = \left\{ \begin{array}{ll} 0 & \forall v_j \in N \quad \exists \quad \Delta h(v_i, v_j) > m d(v_i, v_j) |\nabla_u f| \\ 1 & else \end{array} \right\} \tag{2.5}$$

Where f is a functional representation of the bare earth, and $|\nabla_u f|$ is the largest gradient which is in the direction u in the x-y plane. The multiplier, m, is used to increase the maximum threshold, to avoid over filtering in flat terrain.

A coarse approximation of the bare earth, f, is obtained by griding the point cloud and deriving the grid values from the lowest point in each grid. From this grid (or image) a slope (gradient) map is obtained. The algorithm is run as in the original slope based filtering, except now the shape of the structuring element at a point is adjusted in tandem with the slope in the gradient map below it.

Adaptive slope based filter - M. Roggero: 2001

Assumption - Gradients in the bare earth are locally bounded.

Another variant on the slope based filter is that presented by Roggero (2001). In this filter, the shape of the structuring element is also adapted to the slope of the bare earth at a point.

Because the bare earth is not known, it is estimated using a local linear regression criterion. In the linear regression, each point is compared to the lowest point in the neighborhood. The distance and height difference from the lowest points are weighted and used as observations in the linear regression. The distances and height differences are weighted in such a way that points furthest from the lowest point contribute less to the parameters of the line. The assumption is that the further a point is from the lowest point the less effect it is likely to have on the local slope. The estimated parameters and their standard deviation are used to compute the maximum height differences from the regressed line at defined distances from the lowest point. A curve is obtained from these maximum heights above the regressed line. This curve represents the initial bare earth. Once an initial bare earth has been determined points are classified as bare earth, object, or unclassified, based on their distance from the initial bare earth.

TIN thinning, de-spiking - R.A. Haugerud and D.J. Harding: 2001

Assumption - Curvature in the bare earth is bounded.

The de-spiking algorithm developed by Haugerud and Harding (Haugerud and Harding, 2001) classifies points based on their contribution to local surface aberrations. The algorithm assumes that the terrain surface is intrinsically smooth, and it proceeds to remove points that display strong curvature. Curvature is here defined by a Laplacian of a parametric surface, f.

$$\phi_N(v_i) = \left\{ \begin{array}{lll} 0 & v_i \in N & |\nabla^2 f(v_i)| > \varepsilon \\ 1 & else & \end{array} \right\} \tag{2.6}$$

Where $\nabla^2 f(v_i)$ is the Laplacian of the continuous surface, f, defined over a neighborhood N and ε is a predefined threshold.

Firstly, a TIN is generated. Next, the algorithm searches for local strong curvatures; specifically, sharp upward convexities. When such a point is found it is dropped from the point cloud and not used in further computations. Only a few points are removed from a locale at a time and therefore the process of searching for local strong curvatures is iterative. The iterations are stopped when the fraction of newly identified non-ground returns drop below 0.1%.

Active contour models - M. Elmqvist: 2001

Assumption - The bare earth is a patchwork of piecewise continuous surfaces.

This algorithm estimates the ground surface by employing active shape models. A membrane, f, defined over a neighborhood, N, is floated upwards from beneath the neighborhood and allowed to cling to the low points (Elmqvist, 2001; Elmqvist et al., 2001; Elmqvist, 2002). The manner in which the membrane sticks to the data points is determined by an energy function. For the membrane to stick to the ground points, it has to be chosen in such a way that its energy function is minimized. Material characteristics of the membrane (e.g. rigidity, elasticity) control the form of the membrane and the low points to which it will cling. The choice of form for the membrane also defines the form of the bare earth. Any point within a buffer of the membrane is labeled as bare earth and the rest as object. The labeling function is given by:

$$\phi_{N,\varepsilon}(v_i) = \left\{ \begin{array}{lll} 0 & v_i \in N & |f(v_i) - h(v_i)| > \varepsilon \\ 1 & else & \end{array} \right\} \tag{2.7}$$

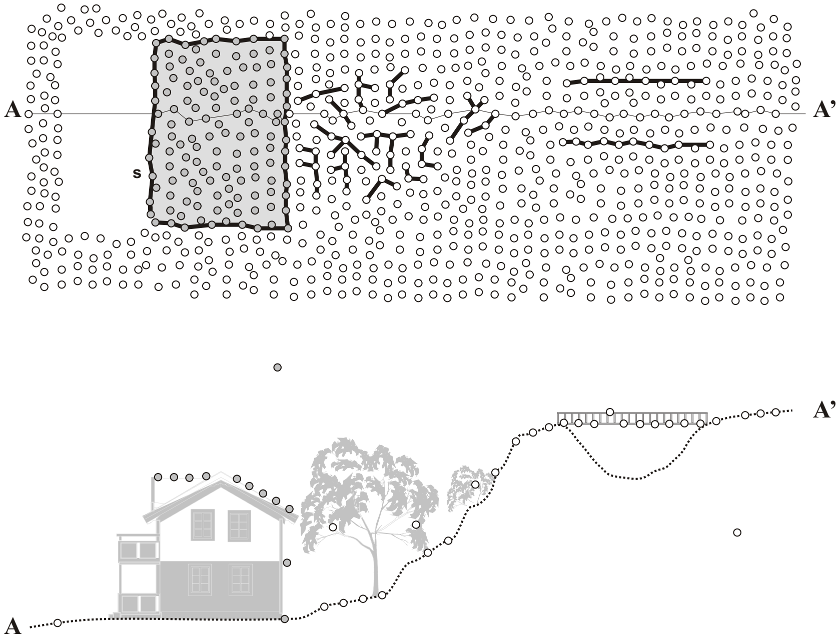


Figure 2.8: Concept of an edge based algorithm.

The membrane used in this instance is an active shape model, and the model is fitted to the low points by minimization of the model's energy function (the energy function defines the material characteristics of the active shape model).

Edge based Clustering - M. Brovelli: 2002

Assumption - Points inside closed edges belong to objects.

Viewed planimetrically (figure 2.8) objects in a landscape standout from the background (bare earth) by the fact that they have distinct edges that together form a closed boundary. Therefore, points within the closed boundaries, s, are accepted as being part of an object. This is the concept of an edge based filtering algorithm, and is used in the filter designed by Brovelli (Brovelli et al., 2002, 2004). The labeling function is given by:

$$\phi_{N,s}(v_i) = \left\{ \begin{array}{ll} 1 & v_i \in N \\ 0 & else \end{array} \right\} \tag{2.8}$$

Where N is a closed neighborhood defined by the closed boundary, s. In a preprocessing step the point cloud is gridded and then tiled. Each tile is set to have 200 x 200 splines. The splines are interpolated on the data. Put very simple (the actual process is much more detailed), points above a spline are potential objects and those below are potential bare earth points. Edges exist at the boundary between bare earth points and object points. Connecting edges (because this part of the algorithm works on a raster, this step also involves a region growing step) are linked and if they close, then the points within the closed edges are accepted as potential objects, provided their height is equal to or greater than the mean edge height. The method also uses the height difference between corresponding first pulse and last pulse points to enhance labeling.

Minimum block classification - R. Wack and A. Wimmer: 2002

Assumption - The lowest point in a neighborhood belongs to the bare earth.

This algorithm is a variant on the morphological filter by Kilian et. al. In this algorithm (Wack and Wimmer, 2002) object raster elements are detected in a hierarchical approach. A 9m raster DEM is generated from a raw point cloud (9m is used to overcome large buildings or dense vegetation). The height value of each raster element is computed from the lowest height from 99% (to overcome the problem of low outliers) of all points within the raster element. Because of the size of the raster elements, most buildings and dense vegetation should now not exist in the DEM. In the next step, all none terrain raster elements are detected and removed (this assumes that objects are characterized by sharp elevation change in the landscape). This is achieved by using a Laplacian of Gaussian (LoG) operation on the 9m DEM. The resulting 9m DEM is used as basis for computing a 3m DEM. From the point cloud a 3m raster is obtained. The representative height of each element is computed from those points inside the 3m elements that are within a given threshold of the corresponding height in the 9m DEM. Remaining raster elements that do not contain bare earth are again detected by an LoG operation on the 3m DEM. Where such elements are detected, their heights are replaced with those from the 9m DEM. At a resolution of 3m and below, a weight function that considers the standard deviation of the data points within each raster element and the shape of the terrain is applied to the output of the LoG operation. This is because at resolutions below 3m break-lines in the bare earth can appear as elements that don't contain bare earth points. In a repetition of the above procedure the 3m DEM is now used to obtain a 1m DEM, and so on. To achieve good results user intervention is required in setting optimal parameter in the determination of the initial 9m DEM. After that, no further user intervention is required.

Progressive TIN densification/ Regularization Method - G. Sohn: 2002

Assumption - The bare earth is locally and globally flat.

The algorithm (Sohn and Dowman, 2002) is based on a two-step progressive densification of a TIN, G. Points in the TIN at the end of the densification are accepted as a representation of the bare earth, and the rest as object. In the downward densification, four points closest to the corners of the rectangular bounds of the point cloud are chosen, and triangulated. The lowest point within each triangle is added to the triangulation. This process is repeated for the triangles in the new triangulation. This process of densification and re-triangulation is repeated until no triangle has a point beneath it. Points in the TIN are accepted as being part of the bare earth.

The downward densification does not catch all the bare earth points; some points above the triangles may yet belong to the bare earth. Therefore, an upward densification has to be done. This step is somewhat similar to Axelsson's TIN densification. A buffer is defined above every triangle (from the downward densification). Those points within a triangle's buffer are tested using MDL (Minimum Description Length) to find which gives the flattest tetrahedral. Those points yielding the flattest tetrahedral are added to the triangulation. This process is repeated until no triangle has a point in its buffer.

The labeling function is given by:

$$\phi(v_i) = \left\{ \begin{array}{ll} 1 & v_i \in TIN \\ 0 & else \end{array} \right\} \tag{2.9}$$

Where TIN is the triangulation obtained after the final densification.

Wavelets - T. Thuy Vu and M. Tokunaga: 2002

Assumption

- Points inside closed edges belong to objects.

This algorithm (ThuyVu and Tokunaga, 2002), like Brovelli's algorithm is edge based. The algorithm's labeling function is given by:

$$\phi_{N,s}(v_i) = \left\{ \begin{array}{ll} 1 & v_i \in N \\ 0 & else \end{array} \right\} \tag{2.10}$$

Where N is a closed neighborhood defined by the closed boundary, s. The abstraction of the closed boundary s, is based on on wavelet theory. In the algorithm a point cloud is first converted into a range image. The range image is then segmented using wavelets. Wavelets are functions that partition data into different frequency components, and then study each component with a resolution matched to its scale. It is particularly useful in analyzing physical situations where the signal contains discontinuities and sharp spikes. Thuy Vu and Tokunaga note that objects appear in the real world within a range of scales, and it is this scale information that they use to segment the data. In the horizontal plane they generate a multi-scale edge map and use it to distinguish objects of different sizes. They simultaneously perform a K-means clustering of the point cloud based on elevation. The multi-scale edge map and the result of the clustering are then used together to segment the image, i.e., points that fall within the edge boundary are separated from others.

Reasoning in horizontal slices - Q. Zhan, M. Molenaar and K. Tempfli: 2002

Assumption - Objects can be modeled, but the bare earth is too complex to be modeled,

This algorithm (Zhan et al., 2002) is designed for building detection in ALS, which is in a sense the converse of bare earth detection. In a preprocessing step the ALS data is gridded. Horizontal profiles of the landscape are then generated. In each profile connected component labeling is done. Components are then connected across profiles, resulting in a tree of components with the ALS point cloud at the root. Each branch of the tree is then analyzed. If adjacent components do not exceed a given size, do not differ greatly in size and location (defined by a component's center of mass) then the two components and all components above them in the branch are labeled as building.

Although not done in the algorithm, what is not labeled as building can now be labeled as bare earth. The labeling function can then be given by:

$$\phi(v_i) = \left\{ \begin{array}{lllll} 0 & \exists & v_i \in s_j & L(s_j) = 0 & s_j \in S \\ 1 & else & & & \end{array} \right\} \tag{2.11}$$

Where S is the set of all components, s_j is a component in S, and $L(s_j)$ is the label of component, s_j.

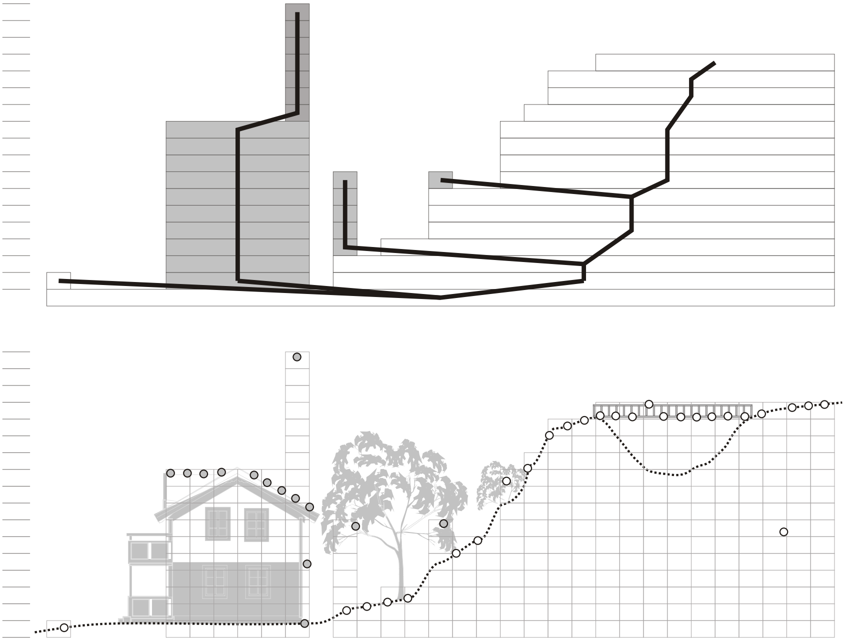


Figure 2.9: Reasoning in horizontal slices. The greyed grids shown in the top figure, the components in the horizontal slices, are the detected buildings.

Morphological filter - H. Masaharu, and K. Ohtsubo: 2002

Assumption - In a neighborhood points in a given height range are bare earth.

This algorithm (Masaharu and Ohtsubo, 2002) starts by selecting the lowest points in a neighborhood, as in the algorithm by Kilian et. al.. All non-selected points are labeled as object. Because the neighborhoods are relatively small, a neighborhood can lie entirely on an object. Therefore, some of these lowest points are object points. The next step of the algorithm aims to identify these object points. A neighborhood, N, of radius, r, around a low point, v_i, is defined. This neighborhood contains only low points from the first step. The mean, μ_h, and standard deviation, σ_h, of the height of all points in N is determined. The labeling function for the low points is given by:

$$\phi_{N,r,\mu_h,\sigma_h}(v_i) = \left\{ \begin{array}{ll} 0 & |h(v_i) - \mu_h)| > \sigma_h \\ 1 & else \end{array} \right\} \tag{2.12}$$

This labeling of the non-object points is repeated until $sigma_h$ becomes acceptably small. Three or four repetitions are indicated to be sufficient.

Segment Filtering - N. Abo Akel, O. Zilberstein and Y. Doytsher: 2003

Assumption - The bare earth is a collection of connected components, occupying a large area.

In this algorithm (Akel et al., 2003) the point cloud is first gridded. This gridded data is then Delaunay triangulated. A region growing is done in which triangles are connected under the constraint that the angle between their normals is below a given a threshold and the height difference between their center of mass is below a given threshold (a method similar to that by Gorte (2002).

Segments, whose area is below a given maximum threshold are labeled as object. This maximum threshold is chosen based on the size of the largest building in the point cloud.

$$\phi(v_i) = \left\{ \begin{array}{lllll} 0 & \exists & v_i \in s_j & L(s_j) = 0 & s_j \in S \\ 1 & else & & & \end{array} \right\} \tag{2.13}$$

Where S is the set of all segments, s_j is a segment in S, and $L(s_j)$ is the label of segment, s_j.

Scan Labeling - A. Sampath and J. Shan: 2003

Assumption - Traveling along a scan, point sets that are between discontinuities and lie above their neighborhoods are object.

Unlike the other algorithms mentioned so far, this algorithm (Sampath and Shan, 2003, 2004; Shan and Sampath, 2004) works on ALS scans [5] and not on whole point clouds. Every point, v_i in a scan has two temporary labels, $a_{i,LR}$ and $a_{i,RL}$. In figure 2.10 the $a_{i,LR}$ and $a_{i,RL}$ label are shown next to every point. The labels, $a_{i,LR}$ and $a_{i,RL}$, are the top and bottom digits respectively.

In a first step the algorithm works along a scan from left to right, figure 2.10. The first point in a scan is assumed to be bare earth and hence its temporary label, $a_{i,LR}$, is set to 1. The first and second points are compared. If they meet a given continuity criteria the first point's LR label is transfered to the second point. If

---

[5] a scan being a single sweep of pulses obtained by one oscillation of a mirror in an oscillating mirror scan system

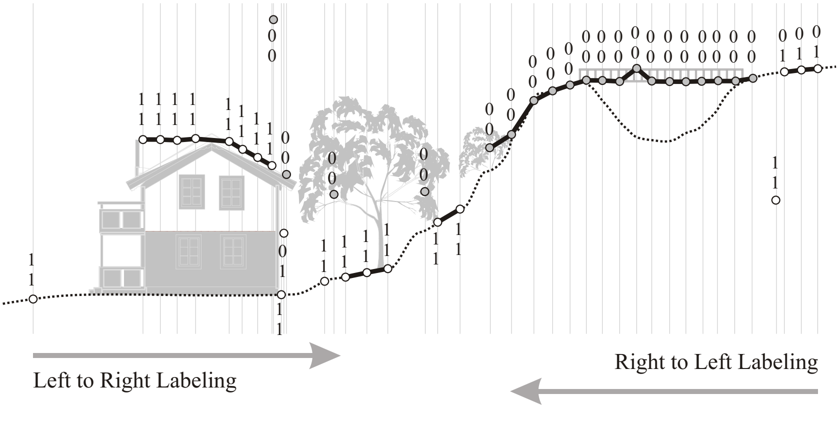


Figure 2.10: Scan labeling algorithm. The solid line indicates points that meet continuity criteria.

they do not meet the given continuity criteria, then the second point's LR is set to 0 if the second point is higher than the first point, and 1 if the second point is lower than the first point. This label transfer is repeated for all adjacent points in the scan, always working from left to right.

The first step is repeated, but this time starting from the last point and working from right to left. Furthermore, labels are now put in the RL label. The label function is now given by:

$$\phi(v_i) = \left\{ \begin{array}{ll} 0 & a_{i,LR} + a_{i,RL} = 0 \\ 1 & else \end{array} \right\} \tag{2.14}$$

2.4 Discussion

The above is not meant to be an exhaustive list of all algorithms. Rather it is intended to show novel ideas used to filter ALS data. What the algorithms also show is that our conception of objects and the bare earth can differ slightly. This difference may lead to different filter approaches. From a design perspective, it is worthwhile to look at the common features amongst the filter algorithms. This knowledge can be used in the design of an improved algorithm. These algorithms are built from combinations of different elements. The discussion here will be focused on identifying the essential characteristics of filters, because they will be referred to in succeeding chapters.

Data Structure

The output of an laser scanning survey is a cloud of irregularly spaced 3D points. Some filter algorithms (Axelsson, Kraus and Pfeifer, Sohn, Roggero, Sithole) work with the raw point cloud. However, to take advantage of image processing toolkits some filtering algorithms (Brovelli, Elmqvist, Wack) re-sample the point cloud into an image grid, before filtering.

Test neighborhood and the number of points filtered at a time

Filters always operate on a local neighborhood. The filtering of a neighborhood can be done in three possible ways.

1. Point-to-Point (1:1) - In these algorithms two points are compared at a time. The discriminant function is based on the positions of the two points. If the output of the discriminant function is above a certain threshold then one of the points is assumed to belong to an object. Only one point is classified at a time.
2. Point-to-Points (1:m) - In these algorithms neighboring points (of a point of interest) are used to solve a discriminant function. Based on the output of the discriminant function the point of interest can then be classified. One point is classified at a time.
3. Points-to-Points (n:m) - In these algorithms several points are used to solve a discriminant function. Based on the discriminant function the points can then be classified. More than one point is classified in such a formulation.

Measure of Discontinuity

Objects are assumed to generate discontinuities in the bare earth. For example, a building breaks the continuity in the terrain. Therefore, all algorithms classify based on some measure of discontinuity. Some of the measures of discontinuity used are, height difference, slope, shortest distance to TIN facets, and shortest distance to parameterized surfaces.

Filter concept

Every filter makes an assumption about the structure of bare earth points in a local neighborhood. For example, bare earth points in a locale must fit a given parametric surface. Four distinct concepts were observed.

1. Slope based - In these algorithms the slope or height difference between two points is measured. If the slope exceeds a certain threshold then the highest point is assumed to belong to an object.
2. Block-minimum - Here the discriminant function is a horizontal plane with a corresponding buffer zone above it. The plane locates the buffer zone, and the buffer zone defines a region in 3D space where bare earth points are expected to reside.
3. Surface base - In this case the discriminant function is a parametric surface with a corresponding buffer zone above and below it. The surface locates the buffer zone, and as before the buffer zone defines a region in 3D space where ground points are expected to reside.
4. Clustering / Segmentation - The rational behind such algorithms is that any points that cluster must belong to an object if their cluster is above its neighborhood. Additionally organizing points into higher-level structures allows the classification of groups of points based on the spatial organisation of surfaces in a point cloud.

Single step vs. iterative

Some filter algorithms classify points in a single pass while others iterate, and classify points in multiple passes. The advantage of a single step algorithm is computational speed. However, computational speed is traded for accuracy by iterating the solution, with the rationale that in each pass more information is gathered about the neighborhood of a point and thus a much more reliable classification can be obtained.

Replacement vs. Culling

In culling a filtered point is removed from a point cloud. Culling is typically found in algorithms that operate on irregularly spaced point clouds. In a replacement, a filtered point is returned to the point cloud with a different height (usually interpolated from its neighborhood). Replacement is typically found in algorithms that operate on regularly spaced (rasterized) point clouds.

Using first pulse and reflectance data

Nowadays scanners record multiple pulse returns. This feature is advantageous in forested areas, where the first pulse is usually on the vegetation and subsequent pulses are from surfaces below the vegetation canopy. Additional to multiple pulse measurements the intensity of the returned pulses is also measured. Different surfaces in the landscape will absorb/reflect pulses differently and therefore it may be possible to use this information in classifying points. Only the algorithm by Brovelli uses the first pulse. None of the described filters made use of the reflectance intensity.

Using external data

External information, such as orthopotos, line maps, land use maps, existing DEMs can be used to augment the filtering process; a view already shared by (Kraus and Pfeifer, 1998; Ackermann, 1999; Axelsson, 1999). While most filters rely solely on the information contained in the ALS data, other filters, particularly those that aim to detect buildings attempt to use other information to enhance the filtering process. Currently not many filters make use of external data.

The brief discussion on the characteristics of filters concludes this chapter. In the next chapter, the experimental results of some of the filters described above are presented.

Chapter 3

ISPRS Filter Test

3.1 Introduction

Some comparison of known filtering algorithms and difficulties have been published by Huising and Pereira (1998), Haugerud and Harding (2001), and Tao and Hu (2001). However, an experimental comparison was not available Therefore, in pursuance of the objectives of this research an experimental study was conducted by the author[1] to compare the performance of various automatic filters developed to date. This study was done with the aim of:

1. *Determining the comparative performance of existing filters.* It is accepted that filters will not be perfect and that most will not be universally applicable. They will work under most scenarios (combination and distribution of features in a landscape), but there are situations in which they will fail. Therefore, it is of interest to find what filter strategy will work under what circumstances.
2. *Determining the performance of filtering algorithms under varying point densities.* Cost efficiency is a significant factor in the choice of resolution at which the landscape is scanned. The lower the resolution the lower the flight cost. However, the choice of resolution also depends on the level fidelity required in the abstraction. The lower the resolution the lower the fidelity. Therefore, a balance has to be struck between lowering costs and ensuring fidelity. Therefore, it is of interest to find out the impact of resolution on the quality of filtering, in relation to the algorithm used.

---

[1] The study was conducted under the auspices of ISPRS Working Group III/3 “3D Reconstruction from Airborne Laser Scanner and InSAR Data”

3. *Determining problems in the filtering of point clouds that still require further attention.*

In line with the objectives of the research, a web site was set up (http://www.geo.tudelft.nl/frs/isprs/filtertest/) where data sets were provided for to the ALS community for testing. Individuals and groups were approached and invited to participate in the study by processing all data sets if possible. Results were received from eight participants. The algorithms used by the participants came from a cross-section of the most common strategies (or variants) for extracting the bare earth from airborne laser scanner point clouds and thus provide a good view on the state of the art.

This chapter is drawn in great part from the following publications, (Sithole and Vosselman, 2003b, 2004), and the final report of the ISPRS filter comparison (Sithole and Vosselman, 2003c).

The ALS data used in the test is described in section 3.2. Sections 3.3 and 3.4 describe the evaluated filter algorithms and the results of the experiments. In section 3.5 the results are discussed with respect to the objectives of the test.

3.2 Test Data

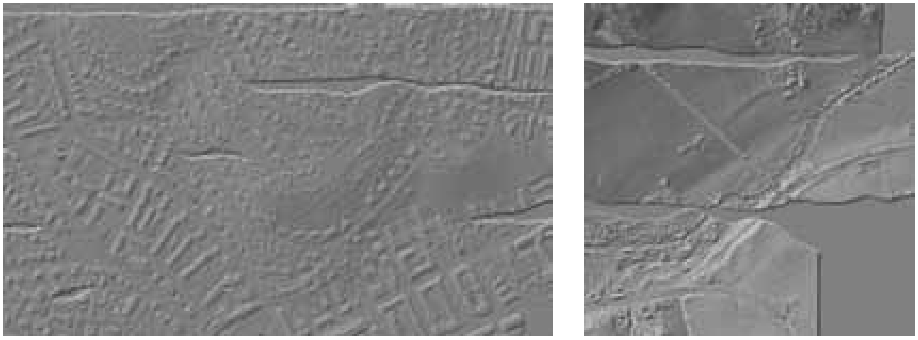
Within the framework of the OEEPE[2] project on laser scanning (Petzold and Axelsson, 2001), FOTONOR AS acquired data with an Optech ALTM scanner over the Vaihingen/Enz test field and the Stuttgart city centre. With kind permission of the OEEPE subsets of this dataset were selected for the comparison of filtering algorithms. Reference data was produced by interactively filtering the datasets.

Data provided to the participants

Eight test sites (four urban and four rural) were chosen because they contained a variety of characteristics that were expected to be difficult for automatic filtering. The datasets included landscapes with steep slopes, dense vegetation, densely packed buildings with vegetation in between, large buildings (a railway station), multi-level buildings with courtyards, ramps, underpasses, tunnel entrances, bridges, a quarry (with break-lines), and data gaps. The urban sites were recorded with a point spacing of 1-1.5m. The rural sites had a point spacing of 2-3.5 m. Both first and last pulse data were available.

---

[2] The European Organization for Experimental Photogrammetric Research (OEEPE) is an organisation founded in 1953 to further photogrammetric research. In 2002 it was renamed European Spatial Data Research (EuroSDR).



Site 1 Site 8

Figure 3.1: Two sites used in the ISPRS test

Sites 1 and 8 (figure 3.1) were also provided at two reduced resolutions to test the filter performance at three different point cloud resolutions. Thus, twelve data sets were provided to the participants. The reduced resolution data sets had 2-3.5 m and 4-6 m point spacing for the urban dataset and 4-5.5 m and 7-10 m point spacing for the rural dataset. To obtain the first reduced resolution the scan lines in the original point clouds were detected. Every second point was selected from every second scan line. Similarly, the second reduced point cloud was produced from the first reduced point cloud.

Reference data sets

The reference data was generated by manually filtering the data sets. Use was made of several interactive tools for removing points on vegetation and buildings. Only after careful inspection, the classifications were accepted. Aerial photographs were available to assist in the interpretation of the point clouds. All points in the datasets were labelled bare earth or object. The definition of what should be considered bare earth is, however, a subjective one. For the purpose of this test, bare earth was defined as the topsoil or any thin layering (asphalt, pavement, etc.) covering it. According to this definition bridges, gangways, etc., were treated as objects. Ramps leading toward bridges, however, were classified as bare earth. Furthermore, the bare earth was treated as being a piecewise continuous surface. Therefore, courtyards were also accepted as being part of the bare earth if they were near the surface interpolated between the points on surrounding streets. From the eight (non-reduced) data sets fifteen samples were abstracted. These fifteen samples were representative of different environments. They were more focused in respect to the expected difficulties as identified in the qualitative comparison. The fifteen samples were used in the quantitative analysis.

3.3 Evaluated filter algorithms

Eight individuals/groups submitted results for the test. The algorithms have already been described in chapter 2. The algorithms tested were those submitted by:

1. *Magnus Elmqvist*: FOI (Swedish Defence Research Institute)
2. *Gunho Sohn*: University College London
3. *Marco Roggero*: Politecnico di Torino
4. *Maria Brovelli*: Politecnico di Milano
5. *Roland Wack, Andreas Wimmer*: Joanneum Research, Institute of Digital Image Processing
6. *Peter Axelsson*: DIGPRO
7. *George Sithole, George Vosselman*: TU Delft
8. *Norbert Pfeifer, Christian Briese*: TU Vienna

3.4 Result of Comparisons

The filter results of the participants were analysed in various ways. The data of all eight test sites were used to visually assess the performance of the algorithms in several difficult landscape types. Qualitative analysis was followed by a quantitative analysis using the fifteen sub-samples that dealt with specific cases. Furthermore, the effect of the point density on the performance of the filter algorithms was assessed quantitatively.

Qualitative assessment

Based on previous experience and the results from the participants, a list was made of circumstances under which the filter algorithms are likely to fail. These situations relate to outliers in the data, object complexity, objects that are attached to the bare earth, vegetation, and discontinuities in the bare earth. Several examples of difficult to filter landscapes are shown in figure 3.2.

The performance assessment of all filter algorithms was based on a visual examination and comparison of the filtered data sets. The qualitative assessment of filters is summarized in Tables 3.1 and 3.2. The main problems faced by the filter

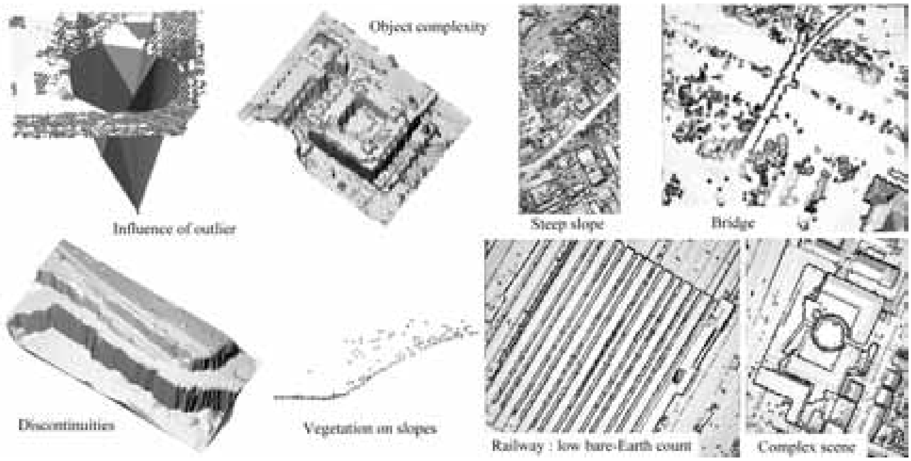


Figure 3.2: Examples of difficult scenarios for filtering

algorithms were found in the reliable filtering of complex scenes, filtering of buildings on slopes, filtering of disconnected bare earth (courtyards), and discontinuity preservation. The examined cases are described below in more detail.

Outliers

Many datasets contain points that are far above or below the bare earth surface.

Low outliers - These are points that normally do not belong to the landscape. They originate from multi-path errors and errors in the laser range finder. Most filters work on the assumption that the lowest points in a point cloud must belong to the bare earth. These points are naturally an exception to the rule. Many algorithms also work on the assumption that points neighbouring a lower point must belong to an object. In practice, this assumption usually holds. However, in cases where the lowest point is an outlier, the assumption fails completely, resulting in an erosion of points in the neighborhood of the low outlier, figure 3.2.

High outliers - These are points that also normally do not belong to the landscape. They originate from hits off objects like birds, low flying aircraft, or errors in the laser range finder. Most filters handle such features easily, because they are so far elevated above neighbouring points. It is included here for completeness only.

Object complexity

For various reasons objects may be difficult to detect:

Very large objects - Because many of the filtering algorithms are localized, large objects may not be filtered completely if the size of objects exceeds that of the test neighborhood.

Very small objects (elongated objects, low point count) - Examples of such objects are vehicles.

Very low objects (walls, cars, etc.) - The closer an object is to the bare earth, the more difficult it becomes for algorithms to differentiate between it and the bare earth. This problem is complicated even further by the need not to incorrectly filter off small but sharp variations in the bare earth.

Complex Shape/Configuration - A major difficulty posed by urban environments is the variety and complexity of objects found in them. This complexity manifests itself in the shape, configuration, and lay of objects.

Disconnected bare earth (courtyards, etc.) - In urban environments, it is common to find patches of bare earth enclosed by objects. The decision of whether an enclosed patch is bare earth is not always clear-cut.

Attached objects

Objects in this category have surfaces that on one side are seamlessly connected to the bare earth, but show clear height differences with the bare earth on other sides. Examples are:

Building on slopes - Such objects have roofs that are elevated above the bare earth on some sides and minimally or not at all on other sides.

Bridges - Artificial structures spanning the gap (road, river, etc.,) between bare earth surfaces.

Ramps - Natural/Artificial structures spanning the gaps between bare earth surfaces; where one is lower than the other.

Vegetation

Vegetation on slopes - Vegetation points can be filtered based on the premise that they are significantly higher than their neighborhoods. This assumption naturally falls away in steeply sloped bare earth where points may lie at the same height as vegetation points. The classification problem becomes harder with increasing

Rating	Item filter rating	Influence rating
Good	Item filtered most of the time (> 90%)	No influence
Fair	Item not filtered a few times	Small influence on filtering of neighboring points
Poor	Item not filtered most of the time (< 50%)	Large influence on filtering of neighbouring points

Table 3.1: Meaning of Good, Fair and Poor (used in Table 3.2)

surface roughness in the bare earth.

Low vegetation - Similar to the problem of low objects, except this time complicated by steep slopes.

Very dense vegetation - Dense vegetation canopies prevent hits from the bare earth.

Discontinuity

Preservation (Steep slopes) - Generally objects are filtered because they are discontinuous from the bare earth. Occasionally it also happens that the bare earth contains discontinuities. At discontinuities in the bare earth some filters will operate as they would on objects. Therefore, discontinuities in the bare earth are lost.

Quantitative assessment

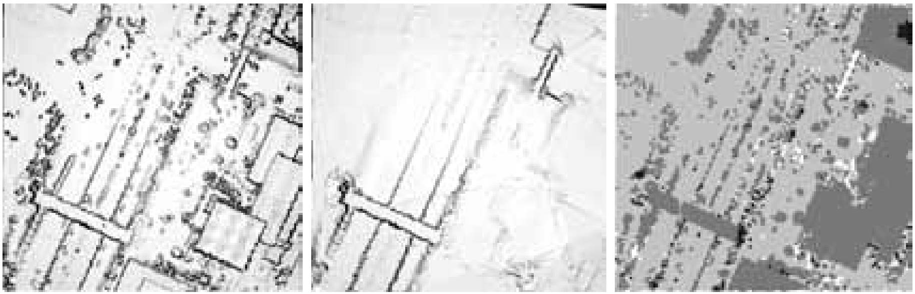
The quantitative assessment was done by generating cross-matrices (for the 15 subsets of the dataset) and generating visual representations of the cross-matrices in order to view the distribution of the errors and understand their causes. An example is shown in figure 3.3. The cross-matrices were then used to evaluate Type I (rejection of bare earth points) and Type II (acceptance of object points as bare earth) errors, and visual representations were then used to determine the relationship of Type I and Type II errors to features in the landscape. It must be stressed that what is presented here covers the difficulties in filtering as observed in the data and in general all the filters worked quite well for most landscapes.

Figure 3.4 shows an actual comparison, accompanied by a visual representation of the errors. The scene is taken from an urban environment (break-lines, large buildings, vegetation, bridges, etc.). It is fairly complex and difficult for filters.

Participant	Elmqvist	Sohn	Roggero	Brovelli	Wack	Axelsson	Sithole	Pfeifer
Outliers								
High points	●●●	●●●	●●●	●●●	●●●	●●●	●●●	●●●
High points influence	●●●	●●●	●●●	●●●	●●●	●●●	●●●	●●●
Low points	●●●	●●	●●	●●●	●●●	●●	●●	●●●
Low points influence	●●●	●●●	●●●	●●●	●●●	●	●	●●●
Object complexity								
Large objects	●●●	●●●	●●●	●●●	●●●	●●●	●●●	●●●
Small objects	●●	●●	●●●	●●	●●	●●●	●●	●●●
Complex objects	●●	●●	●●	●●	●●	●●	●●	●●
Low objects	●	●	●●●	●●●	●●●	●●	●●	●●
Disconnected bare earth	●●	●●	●●	●●	●●	●●	●●	●●
Detached objects								
Building on slopes	●●●	●●	●●	●●	●●	●●●	●●	●●●
Bridges	●●●/R	●●●/R	●●●/R	●●●/R	●●●/R	●●	●●●/R	●●●/R
Ramps	●	●	●	●	●	●●	●	●
Vegetation								
Vegetation	●●●	●●●	●●●	●●●	●●●	●●●	●●●	●●●
Vegetation on slopes	●●●	●●●	●●	●●	●●	●●	●●	●●●
Low vegetation	●●●	●●	●●	●●	●●●	●●	●●	●●●
Discontinuity								
Preservation	●	●	●	●	●●	●●	●	●●

●●● Good; ●● Fair; ● Poor; R Removed

Table 3.2: Qualitative comparisons of filters



Correct bare earth	Type I error
Type II error	Correct object

Unused 121

		Filtered			
		BE	Obj		
Refer.	BE	21880	602	22482	68.99%
	Obj	588	9515	10103	31.01%
		22468	10117	32585	
		68.95%	31.05%		

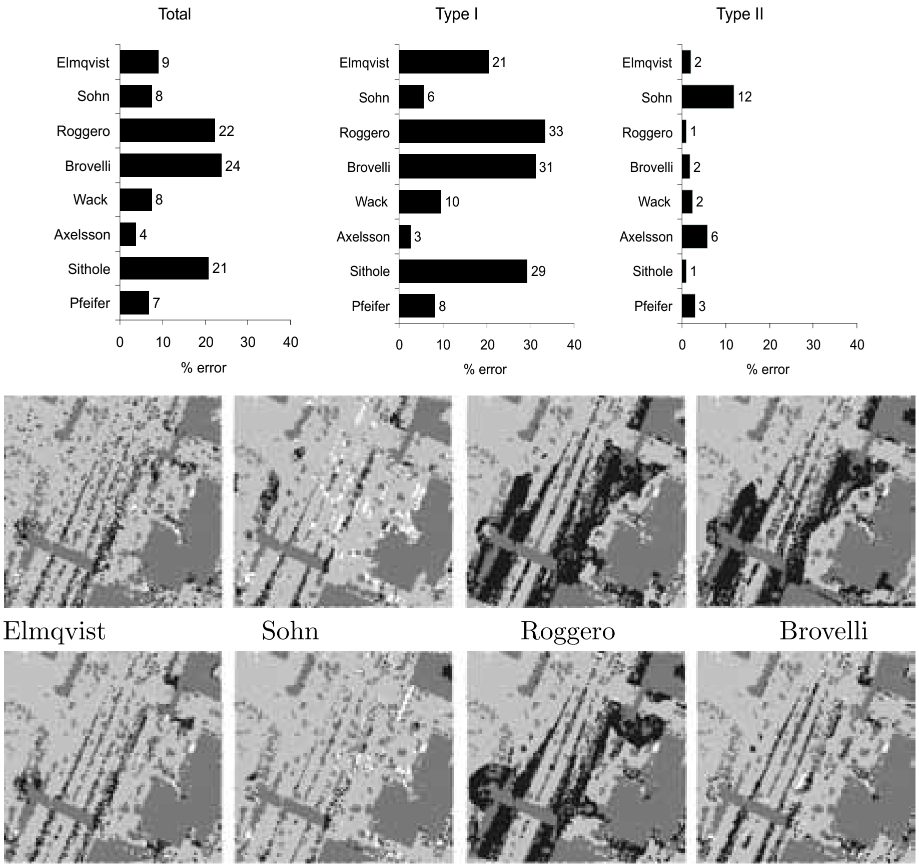
ratio BE-Obj/ Obj-BE 1.02

Figure 3.3: Sample data for quantitative comparison and assessment

From the numbers and the visuals it can be seen that each filter responds differently to the features in the landscape (some doing better than others). After some remarks on the computation of errors and a global analysis, filter problems related to several difficult object and landscape features are discussed in more detail.

Type I vs Type II errors

The output from some participant's filters is gridded or altered in position from the original. Because of this, DEMs were generated for the participant's filtered data and the height of the points in the reference data were compared against these DEMs. Using a predefined threshold (20 cm) to account for interpolation errors and based on height comparison, all points were labelled as correct bare earth, Type I error, Type II error or correct object. Therefore, the Type I and II errors have to be understood in the context of height comparison of the reference against the filtered DEMs.



Elmqvist Sohn Roggero Brovelli

Wack Axelsson Sithole Pfeifer

Figure 3.4: Quantitative comparison of site 22

All the filtering algorithms examined make a separation between object and bare earth based on the assumption that certain structures are associated with the former and others with the latter. This assumption while often valid, fails sometimes. This failure is caused by the fact that filters are blind to the context of structures in relation to their neighbourhoods. Because of this a trade off is involved between making Type I (reject bare earth points) and Type II errors (accept object points).

The computed errors (over all the data sets) ranged from 0-64%, 0-19%, 2-58% for Type I, Type II and the Total errors respectively. Most filters focus on minimizing Type II errors (as shown in figure 3.4), except the filters by Axelsson and Sohn. In other words filter parameters are chosen to remove as many object points as possible, even objects that are small and close to the ground. The downside of this is that many valid bare earth points could be removed. This tendency to minimizing Type II errors, partly suggests that participants consider the cost of Type II errors to be much higher than that of Type I errors.

While Type I and Type II errors show the tendency of the choice of filter parameters, the influence of the filter algorithm can be gauged from the total error. For example in figure 3.4 it can be seen that a filter may have a high Type I or Type II error and yet have a relatively small total error. This depends on the proportions of bare earth and object points in the scene.

Analysis of the results

Steep Slopes

Height differences (discontinuities) are a key to separating the bare earth and objects. Therefore, points significantly above their neighbours are assumed to be off objects. This assumption easily holds in flat bare earth, but becomes more difficult as the slope of the bare earth increases. Therefore, as expected most filters had difficulties on steep slope (e.g., figure 3.2), but some filters (those that minimized Type I errors) also generated small total errors on steep slopes. The explanation for this could lie in the fact that for most filters, if Type I errors are minimized then more object points are classified as bare earth (increasing Type II errors). However, in general, there are fewer object points then there are bare earth points, and hence a Type II error has a smaller influence on the total error than a Type I error. Nonetheless, filtering in steep bare earth still remains a problem especially at reduced resolutions.

Discontinuities (preservation)

As already mentioned the bare earth and objects are assumed to be separated by discontinuities. However, break-lines (figure 3.2) in the bare earth are an exception to this assumption, hence the need to retain discontinuities in these exceptions. The two slope based filters (Roggero and Sithole) have the most difficulty with discontinuities in the bare earth as can be seen in figure 3.4. This is borne by the large number of Type I errors. However, when the height difference at discontinuities increases the performance of the slope-based filters remains the same. This is not the case with some of the other filters, where a discontinuity can also influence filtering in the neighborhood of the discontinuity. Another interesting aspect of filtering at discontinuities is where the Type I errors occur. Some filters only cause Type I errors at the top edge of discontinuities, whereas others cause errors at both the top and bottom edges. The potential for the latter case happening is relatively higher for surface based filters.

Bridges

Filters do not make a reasoned distinction between objects that stand clear of the bare earth and those that are attached to the bare earth, e.g., bridges (figure 3.2). From the results in figure 3.4, it can be seen that the removal of bridges can be complete, partial or not at all. All the algorithms for the exception of Axelsson's seem to remove bridges consistently. A possible reason for this could be the method of point seeding used in the algorithm. Another problem with the filtering of bridges relates to the decision made about where a bridge begins and ends. This problem is detected by Type II errors at the beginning and end of bridges (bridges in the test were treated as objects). This error though is generally not large. Similar to bridges are ramps. Ramps bear similarity to bridges in that they span gaps in the bare earth. However, they differ in that they do not allow movement below them. As such, ramps were treated as bare earth in the reference data. All the tested algorithms filtered off the ramps.

Complex scenes

Shown in the scene (figure 3.2) is a plaza surrounded on three sides by a block of buildings. From the plaza, it is possible to walk onto the road to the east and also descend via stairs to the road below (west). Further, complicating matters there is a sunken arcade in the centre of the plaza. Defining what is and what is not bare earth in such a scenario is difficult. In this example, the plaza and arcade were assumed to be bare earth based on the rationale that it is possible to walk without obstruction from the plaza to the roads on the west and east. However, this assumption is very subjective. For this scene, the filters that make use of local surface assumptions (Pfeifer, Sohn, and in particular Axelsson) performed best. However, if the arcade, the stairs and the plaza had been accepted as being object, then the morphological filters would have done better.

Outliers

The number of outliers (both high and low) was relatively small and therefore their contribution to Type I and Type II errors was small. However, their influence on filtering in their neighbourhoods can be considerable. The filters by Axelsson and Sithole produce such Type I errors. While single outliers cause problems for a few filters, numerous closely spaced outliers will cause problems for many filters. Even more, the influence of numerous outliers on their neighbourhoods can be significant depending on the concept base of the filter.

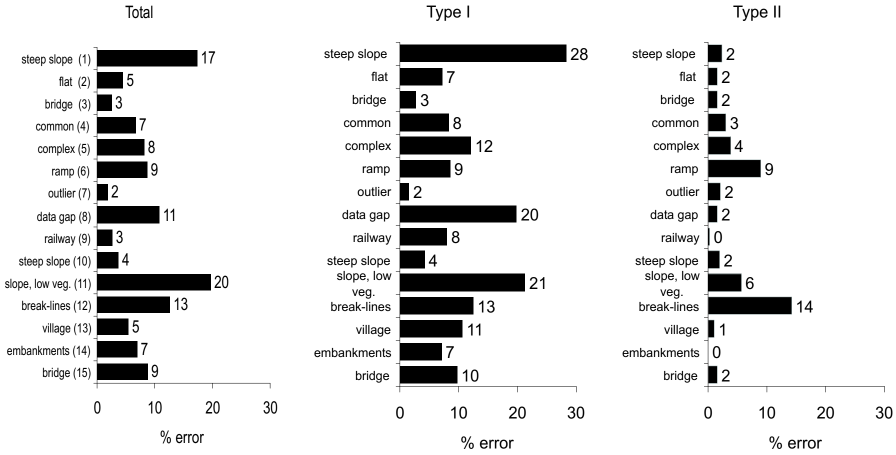


Figure 3.5: The effect of landscape on filter performance

Outlier detection prior to filtering would negate the influence problem on filters that are susceptible to outliers.

Vegetation on slopes

Most of the filters do well in identifying vegetation on slopes and even better on flat landscapes. However, on steep slopes filtering is sometimes done at the cost of increased Type I errors, and in the case of the Elmqvist and Brovelli filters quite significantly.

Low bare earth point count

Because filters depend on detecting structures, especially those that detect bare earth it is essential that there be enough sample bare earth points. Low bare earth counts can be found in areas where there are high densities of objects or vegetation (such as railway stations, e.g., figure 3.2). However, most of the filters did well in identifying bare earth points despite the low count of bare earth points. Part of the reason for this could be that they all assume that the lowest points are bare earth.

Overview

Although the fifteen sub sites used for the quantitative analysis, all contained one or more difficult features to filter, the performance of filters differed substantially from subset to subset. The result from one of the eight filter algorithms, across the fifteen sub sites is shown in figure 3.5 . It can be seen that two steep slopes generate the largest total error. In the first case (sub site 1, 17% total error), the slope is heavily covered by buildings and vegetation. The second case (sub site 11, 20% total error) is covered by low vegetation. Mixed buildings and vegetation and low vegetation on steep slopes are difficult cases, hence the large errors are expected. For both sites, there is a clear emphasis on reducing Type II errors (stacked chart to the right). A balancing of Type I and II errors will most probably reduce the total error. Sub site 10 also has a steep slope with vegetation. But the vegetation on these slopes is elevated (strong discontinuity), thus making them easier to filter. The next site to generate large errors is sub site 12, which contains break-lines (it is in fact a quarry). The balanced Type I and Type II errors show that effort was placed in keeping the total error small. The data gap also generates a large total error. This appears to be because a lot of effort is put in reducing Type II errors.

Because of space constraints all the landscapes cannot be described here. However, it is instructive to see that while some landscapes will generate more filtering errors than others, the choice of parameters and balancing of Type I and II errors also plays a significant part in the performance of a filter algorithm.

Effect of resolution

As the resolution of the data is lowered, it is harder to separate the bare earth from the objects. To determine how filters cope with reduced point densities, the data of two sites, site 1 and 8 (figure 3.1 were provided at three different resolutions. The results of the participants were compared to the reference data of the corresponding resolution. A cross-matrix was generated for each comparison. The results for the urban site are shown in the charts in figure 3.6 (again the same presentation as in figure 3.5). For some of the participants there was no data at some of the resolutions (Elmqvist, Brovelli and Sithole).

The results do not allow firm conclusions to be drawn on the effect of the resolution on the filter performance. In general, it can be seen that the Type I errors are quite large for site 1. The effects of the reduced resolution are most likely minor compared to the errors that are already made due to the complexity of the scene (buildings and vegetation on a steep slope). The different filters show quite different responses to the reduction in resolution. This may in part be explained by the fact that some participants tweaked their filter parameters to obtain optimal results at different resolutions, whereas others did not.

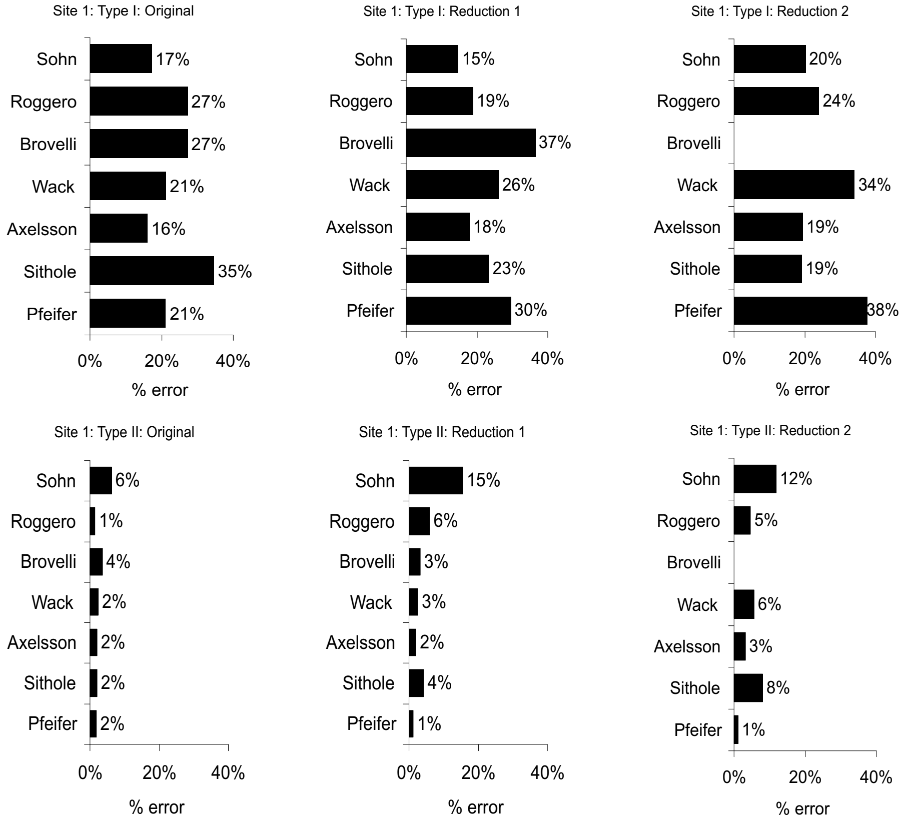


Figure 3.6: Type I and Type II errors of Site 1 vs. resolution.

The results for site 8 (not shown in figure 3.6) in general showed a slight increase in both Type I and Type II errors for reduced resolution levels. The strong increase in Type I errors was observed for the filter of Wack. This is most likely due to a large number of interpolation errors when comparing the gridded filtered data to the reference point cloud.

3.5 Discussion

What has been presented are some of the striking difficulties in filtering as observed in the data. In general all the filters worked quite well in landscape of low complexity (characterized by gently sloped bare earth, small buildings, sparse vegetation, and a high proportion of bare earth points). As seen in section 3.4. The performance of a filter can differ depending on the feature content of a landscape. The quantitative evaluations of the data provided by the participants resulted in numerous tables with error counts and standard deviations. The complete analysis can not be presented here (because of space restrictions) but it is available in the full report (Sithole and Vosselman, 2003c) of the study[3]. The problems that pose the greatest challenges appear to be complex cityscapes (multi-tier buildings, courtyards, stairways, plazas, etc.,) and discontinuities in the bare earth. It is expected that tailoring algorithms specifically for these areas may improve results, albeit by a small amount.

Concerning the filter strategy, it was noted that overall the surface based filters appear to yield better results. Generally, they use more context than other filter strategies and are therefore better able to separate points on a (ground) surface from other points.

A decision always has to be made between minimizing Type I and Type II errors. The question of which error to minimize depends on the cost of the error for the application that will use the filtered data. From a practical point of view, it will often depend very much on the time and cost of repairing the errors manually, which is done during quality control. Experience with manual filtering of the data showed that it is far easier to fix Type II errors than Type I errors. Generally, Type II errors are conspicuous. In contrast, Type I errors result in gaps in the landscape, and deciding whether a gap has been caused by a Type I error or from the removal of objects is usually not possible. While the costs of Type II errors are thus considered to be lower, it is striking to see that most algorithms produced much more Type I errors than Type II errors. This raises the question whether the filter algorithms can be more tuned toward the reduction to Type I errors, even if this is at the expense of an increased number of Type II errors.

Point density

More tests on decreasing resolution will need to be done, as the test sites chosen have proved inadequate to obtain a conclusive picture of the effects of resolution on filtering. The complexity of the sites has meant that even at the highest resolutions the filters have difficulties, which then masks the performance of the filters at lower resolutions. Nonetheless, in choosing the scan resolution the filter concept used becomes critical, especially in landscapes with steep slopes. Additionally a large Type I error does not necessarily mean the resulting digital elevation model will be poor. Importantly it depends on where Type I and Type II errors occur in the landscape.

---

[3] The report can also be obtained from the web site setup for the study, `http://www.geo.tudelft.nl/frs/isprs/filtertest/`

The next chapter will look at the conceptual development of a new filtering algorithm, with the objective to solve the problems identified in this chapter.

Chapter 4

Filter Design

4.1 Introduction

The algorithms presented in chapter 2 were shown to separate between objects and the bare earth using different assumptions. The use of all these assumptions arise from (i) the designers conception of the bare earth, (ii) the desire for a simple algorithm, and (iii) resource limitations.

The first two reasons are the most significant because they relate to the design of a filter algorithm. Chapter 3 tested the strengths and weakness of the assumptions on real data. Because the intention is to design a filter applicable to as many landscape types as possible the strengths of the assumptions are used to formulate new assumptions. From these new assumptions, a filter framework is designed. This chapter details the development of new assumptions and proposes a framework based on these assumptions. For the purpose of succeeding discussions the common assumptions in chapter 2 are re-listed here in the order given below. The assumptions are ordered according to their perceived complexity.

A1 The lowest points in a neighborhood belong to the bare earth.

A2 In a neighborhood points within a given height range are bare earth.

A3 Gradients in the bare earth are globally bounded, i.e., for the entire bare earth slopes are never greater than a certain threshold

A4 Gradients in the bare earth are locally bounded. i.e., for a bare earth neighborhood slopes are never greater than a certain threshold, and this threshold can differ between neighborhoods.

A5 Curvature in the bare earth is bounded.

A6 The bare earth is locally and globally flat.

A7 The bare earth is a patchwork of piecewise continuous surfaces.

A8 Objects can be modeled, but the bare earth is too complex to be modeled.

A9 Points inside closed edges belong to objects.

A10 The bare earth is a collection of connected components, occupying a large area.

These different assumptions give rise to three questions that need to be answered before new assumptions are formulated.

1. *Are the assumptions used correct?* - Every assumption is correct in respect to the type of landscape for which it was designed. Because of this, there is no such thing as a bad filter, but rather a filter that is incorrectly applied.
2. *Which is the best assumption?* - Stated differently which assumption accommodates the greatest number of bare earth and object types. This is a difficult question because it requires us to objectively define the bare earth and object. However, an objective definition of bare earth and object is impossible. Firstly, because we lack the capacity to mathematically describe the bare earth or objects exactly (at all scales and in all its forms). The best we can do is to describe it in approximate terms (slope, curvature, roughness, etc.,). Secondly, the definition of bare earth is subjective. No two people will share the same definition of the bare earth. What maybe bare earth to one person may not be bare earth to another. The filter test in chapter 3 showed that filters based on assumption A7 performed the best. But this is in itself not conclusive because not all filters and landscape types were tested. What it indicates is that given any landscape assumption A7 has a better chance of working.
3. *Are there better assumptions that could be formulated?* - The filter test in chapter 3 showed that the assumptions used are inadequate, especially for urban landscapes and landscapes that contain steep slopes. This hints at an insufficiency in the modelling and hence an inadequacy in the assumptions. This suggests that better assumptions can be formulated.

4.2 The weakness of current assumptions

A cursory examination of a point cloud gives the impression that filtering is a relatively simple task. This is because human cognitive skills and intuition are ap-

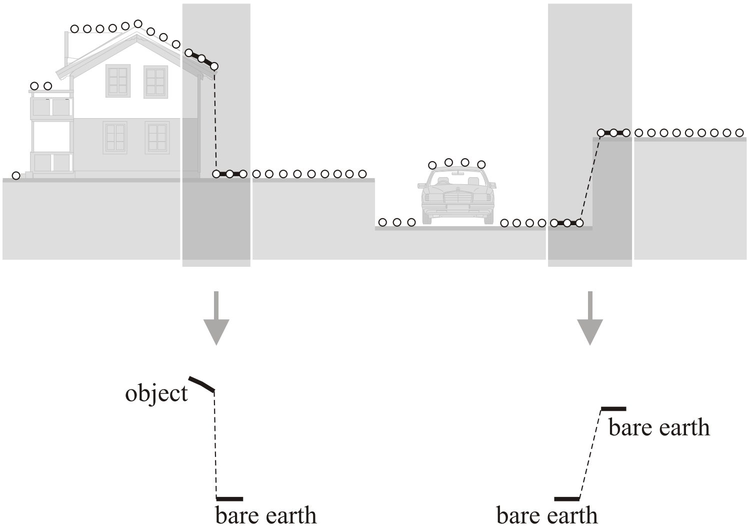


Figure 4.1: Neighborhoods that contain objects and bare earth, and neighborhoods that contain only bare earth are homeomorphic

plied to the task. However, duplicating human cognitive skills and intuition is not simple. The best that can be done is to define and apply simple rules on the radiometry, geometry, and topology of sample points abstracted from the landscape. Defining these rules is difficult because of the characteristics of a landscape, and the characteristics of ALS data.

Characteristics of Landscape

The difficulty here arises from the nature and arrangement of objects and the bare earth in the landscape. Furthermore, these difficulties will appear in every ALS point cloud without exception.

1. *Complexity of the Landscape* - Here complexity is in reference to the form of the terrain (not to be mistaken with roughness), in particular discontinuities (e.g., embankments, raised platforms, terraces, etc.,). Conventional filters, because they operate on small neighborhoods will suffer because locally bare earth-to-bare earth relations and bare earth-to-object relations are homeomorphic as shown in figure 4.1. This complexity invalidates assumptions A1 through A5.

2. *Complexity of objects* - While most buildings in urban areas are regular in

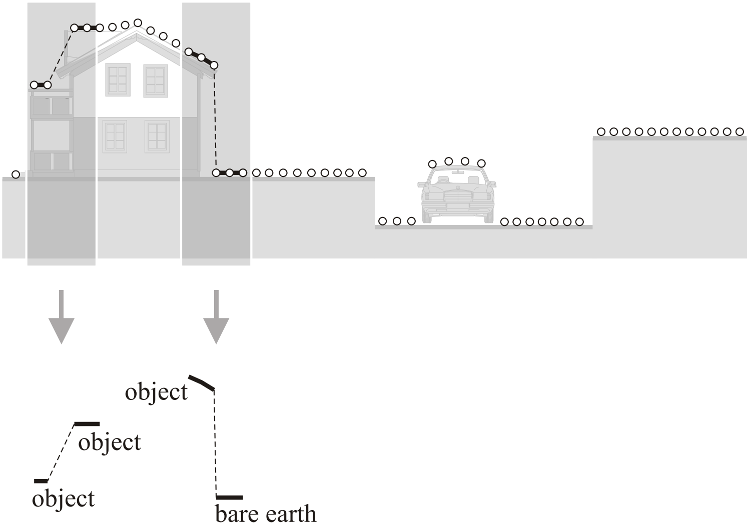


Figure 4.2: Neighborhoods that contain objects and bare earth, and neighborhoods that contain only objects are homeomorphic

shape (blocks, prisms, etc.,) there are others that are more complex (layered roofs, platforms, etc.,). Again conventional filters, because they operate on small neighborhoods will suffer because locally object-to-object relations and bare earth-to-object relations are homeomorphic as shown in figure 4.2. This complexity invalidates assumptions A1 through A5.

3. *Proximity of objects to the bare earth* - The relation between bare earth and objects is in direct proportion to the vertical and lateral separation between them. Therefore, the closer an object is to the bare earth the more difficult it becomes to distinguish it from the bare earth, this is demonstrated in figure 4.3. As objects and the bare earth near each other all the stated assumptions become invalid.

4. *Object size* - Objects are generally differentiable from the bare earth in that globally there is more bare earth than any single object. However, locally this is not the case. In a neighborhood it is possible that there maybe more object points than bare earth points. This presents problems for assumptions A1 through A5. More generally, any filter that operates on a local neighborhood will face problems if there are more object points than there are bare earth points.

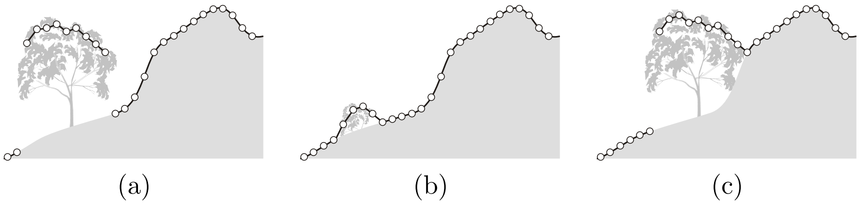


Figure 4.3: Vertical and lateral separation of points, (a) desirable vertical and lateral separation, (b) poor vertical separation, (c) poor lateral separation.

Characteristics of the data

Unlike the difficulties presented by landscape characteristics for formulating rules, the difficulties in filtering due to data characteristics can be controlled to a certain extent. Furthermore, the difficulties will not be present in every data set.

1. *Resolution of the point cloud* - The resolution of a point cloud has a direct influence on the spatial definition of objects. The less defined an object the more it starts to resemble the bare earth. In a low resolution point cloud assumptions A3 through A10 become invalid for the simple reason that discontinuities become difficult to discern. In a high resolution point cloud assumptions A3 through A7, A9 and A10 can also start to break down because of the noise present in such point clouds.

2. *Data gaps* - Objects that lie along or near the edge of a point cloud are usually only partially abstracted (truncated). This invalidates assumption A9 and possibly A8.

3. *Variation in point density and strip overlaps* - Variations in planimetric point density depend on the ALS scan system used, the scan angle, the scan swath and the flying speed of the aircraft. Scanning is done so that the variations are kept as small as possible. Typically, a point cloud is built by combining strips of scans. Because of this, point clouds have higher densities in areas where strips overlap. Figure 4.4 shows two strips that are combined to yield a point cloud. As can be seen in the overlap region the point spacing has been reduced. Because proximity measures in filters depend on point density, in a point cloud formed by combining strips, discriminating points based on proximity becomes problematic. A proximity threshold that works in overlapping regions may lead to type I errors in the non-overlapping regions. Similarly, proximity thresholds that work in the non-overlapping regions may lead to type II errors in the overlapping regions. This characteristic of the data can cause problems for assumptions A1, A2, A5, A8 and A10. Naturally,

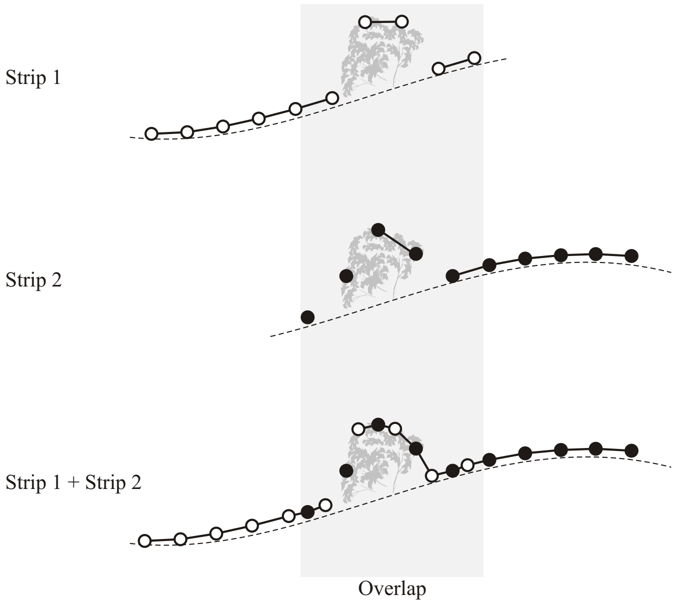


Figure 4.4: Reduction of point spacing in strip overlaps

filtering individual strips at a time would avoid this problem. However, strips usually have to be combined to avoid the problem of truncated objects.

4. *Ambiguity* - Objects like bridges and overpasses are man made extensions to the bare earth. The decision to keep or remove such objects requires that bridges be explicitly identified. Assumptions A1 through A10 cannot account for objects like bridges.

5. *Zero ground returns (lack of bare earth information)* - Typically the penetration rate of ALS in vegetated areas is around 25% (Lindenberger, 1991), especially when scanning is done in winter when foliage on trees is in decline. However, this penetration rate can sometimes be very low to the extent that there are very few bare earth samples, as shown in figure 4.5. In the absence of bare earth samples meaningful filtering is not possible. This invalidates assumptions A1 through A10.

4.3 Formulating new assumptions

As we move down the list of assumptions, A1 through A10, the neighborhoods examined in the point cloud become larger and the emphasis is placed increasingly on working on connected-point-structures (surfaces, clusters, and segments) rather than individual points. Larger neighborhoods provide a greater view on the data and permit a contextual analysis of of points. For example, we can now better deal with the homeomorphy shown in figures 4.1 and 4.2.

Furthermore, the characteristics of the landscape start to have a lesser effect on filtering, and the characteristic of the data become the dominant problem. In a sense, this is welcome because the characteristics of the landscape cannot be altered, whilst we can control the characteristics of the data.

For this reason from here on emphasis shall be placed on working with connected-point-structures, overcoming the problems that result from characteristics of the landscape, and detecting bridges. Furthermore, connected-point-structures will be referred to as segments. Before new assumptions can be formulated, a model of the landscape has to be devised.

Modelling the landscape

Any landscape is composed of three distinct classes of features, the bare earth, natural objects, and man made objects. These form a minimum description of the topography. Others are more elaborate in their classification of landscapes. For example Lohmann (2000) lists eight classes. However, many filter designers already view the topography in respect to these three classes (although they will use alternative words like terrain/vegetation/buildings or ground/vegetation/buildings). This description of the topography is deliberately very general with a view that each class can later be broken into sub classes. To formulate new assumptions a

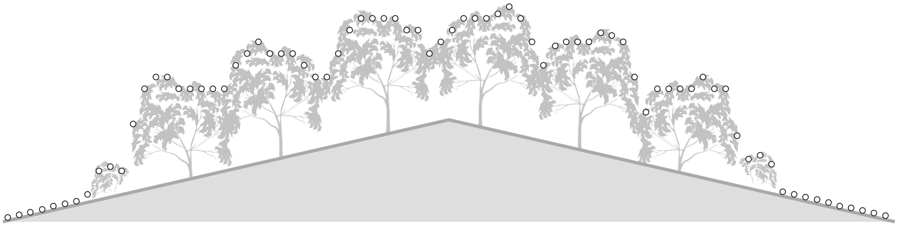


Figure 4.5: Zero ground returns. In the area covered by vegetation there are no hits off the bare earth

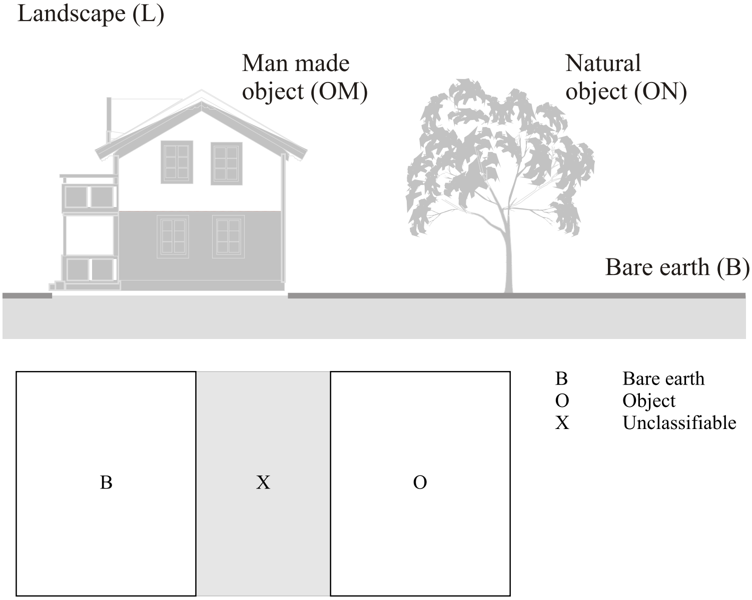


Figure 4.6: A conceptual and logical representation of the real world

more elaborate description of the classes is required. This elaboration has to be on two levels, (conceptual [1] and logical [2]).

Figure 4.6 shows one possible conceptualization of the real world, in which there is the bare earth, on top of which resides natural and man made objects. The elements of this models are defined below:

1. *Landscape, L* - A scene consisting of the bare earth and other objects.
2. *Objects, O* - A collection of natural, O_N, and man made, O_M, features in a landscape.
3. *Bare Earth, B* - The topsoil or any thin layering (asphalt, pavement, etc.) covering the topsoil. In any landscape the bare earth is broken where it is covered by a man made object.

---

[1] Conceptual - *a description suitable for language.*

[2] Logical - *a description suitable for programming.*

Bare earth and Objects

The conceptual model is setup to describe the real world in very general terms, with the idea of identifying the elements to be modeled and establish, the relationships between the elements. A landscape acquired by ALS is necessarily discrete. The logical model is setup to relate the points (or point clusters) in a point cloud to the elements in the conceptual model in a manner consistent with the real world. In the logical model the landscape, L, is replaced by the point cloud, V and elements of the bare earth, B, and object, O, are replaced by points or surfaces. In the real world, a person can readily distinguish between the bare earth and objects. For example, no person would confuse a car for bare earth. Because of this in the conceptual model the membership of the sets B and O are crisp. In a point cloud distinguishing between the bare earth and objects is much more difficult and situations can arise in which points (or point clusters) cannot be classified or are misclassified. Because of this uncertainty, a set X is introduced to accommodate points whose classification is in doubt. The membership of X is subjective because each person will interpret points better or worse than others. This inability to classify arises from either an insufficiency of evidence or a conflict of evidence.

Natural objects, man made objects, and bridges

The logical model is not yet complete, because it does not account for natural objects, O_N, it does not account for man made objects, (O_M), and it does not account for bridges O_{MA}. Figure 4.7 shows an extension of the logical model in figure 4.6 that accommodates natural objects, man made objects and bridges.

Natural objects mostly refer to vegetation. However, motor vehicles, lampposts and other small objects are treated as natural object[3]. This anomaly is entertained because it is desired that the set, O_M, only contain buildings. This is because for most urban applications the most sought after feature are buildings. The boundary of the sets O_M and O_N lie on that of X to indicate that the memberships of $O_M \cap B$ and $O_N \cap B$ are uncertain. A further set X_O is defined for uncertainty in the membership of $O_M \cap O_N$.

As shown in chapters 3 and 2, current filtering strategies only make the distinction between two types of features in a landscape: bare earth and object. From the results of the ISPRS study it is evident that this distinction is inadequate because objects like bridges and flyovers that are attached to the bare earth surface, are not dealt with appropriately. Some applications may require objects like bridges to be filtered, whereas other applications would prefer to have them preserved in the point cloud. Therefore, it is necessary to explicitly detect such objects in the landscape, thus allowing the user to specify whether such objects should be kept

---

[3] Outliers are also treated as natural object

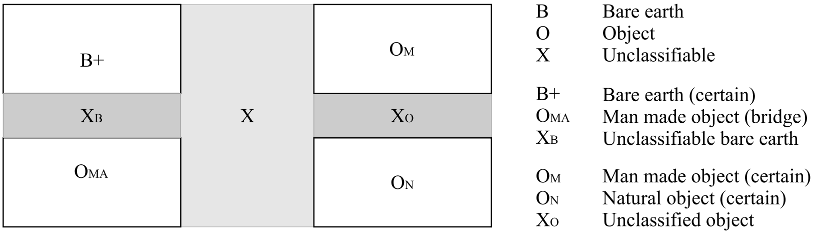

B+
OM
XB
X
XO
OMA
ON
B Bare earth
O Object
X Unclassifiable
B+ Bare earth (certain)
OMA Man made object (bridge)
XB Unclassifiable bare earth
OM Man made object (certain)
ON Natural object (certain)
XO Unclassified object

Figure 4.7: Extended logical representation with natural and man made objects.

or removed (Sithole and Vosselman, 2003a).

The logical model can now be summarized as:

1. $B \cup O = V$
2. $B \cap O = X$
3. $O_N \subset O$, points that are certainly natural objects
4. $O_M \subset O$, points that are certainly man made objects
5. $O_M \cup O_N \cup X_O \cup X = O$
6. $B_+ \subset B$, points that are certainly bare earth
7. $O_{MA} \subset B$, bridge is part of the bare earth
8. $B_+ \cup O_{MA} \cup X_B \cup X = B$

Excluding X the set B is comprised of two sets, B_+ and O_{MA}. These are the sets of points that are definitely bare earth and bridge respectively. A further two sets B_+ and O_{MA} are defined to represent points that are know to be bare earth (certain) and bridges respectively. The set B_+ shares a boundary with X to indicate that there is uncertainty in the membership of $B \cap O$. A set, X_B, is introduced to accommodate the uncertainty in the membership of $B_+ \cap O_{MA}$. This uncertainty arises from a difficulty in defining where a bridge begins and ends.

Based on this logical model:

1. Firstly, bare earth and objects are separated.
2. Secondly, objects are separated into natural objects and man made objects.
3. Thirdly, bridges are separated from the bare earth.

The next section looks at the new assumptions that are used to effect the above listed separations.

The new assumptions

The elements of all sets except V shall be treated as being point clusters/segments instead of points. This is in line with the reasoning at the beginning of section 4.3. For the new filtering strategy the following assumptions are adopted:

B1 A point cloud is free of systematic errors: Systematic errors in a point cloud would distort the geometry and topology of objects, and thus invalidate all other assumptions.

B2 In a point cloud objects are separated by discontinuities and the point spacing varies in direct proportion to the gradient of slopes in the landscape.

B3 All points in a segment are part of the same class.

B4 The surface of each object is encompassed in one segment: The topological relation between objects is established by the relationship between their boundaries, and segments provide a means to obtain object boundaries. Therefore, to establish correct relationships between objects, objects have to be contained entirely within one segment.

B5 The perimeter of each object segment is mostly raised above its neighborhood: In general, objects are raised above their neighborhoods. Therefore, any segment that has its perimeter higher than neighboring segments is in likelihood an object.

B6 The surface of the bare earth is a collection of segments - but there is only one bare earth: Because of data gaps (e.g., absorption of pulses by water bodies, gaps between strips, etc.,) or enclosures (e.g., courtyards, etc.,) the bare earth maybe broken up into several segments. However as an entity there is only one bare earth.

B7 In any landscape the bare earth in its entirety occupies more space than any single object (provided that the landscape sampled is large).

B8 Man made objects and natural objects are distinguishable by their roughness: Man made objects by design tend to have smooth surfaces, whereas natural objects tend to be rough.

B9 Man made objects and natural objects are distinguishable by their radiometry (if every point in a point cloud is associated with a reflectance or RGB value).

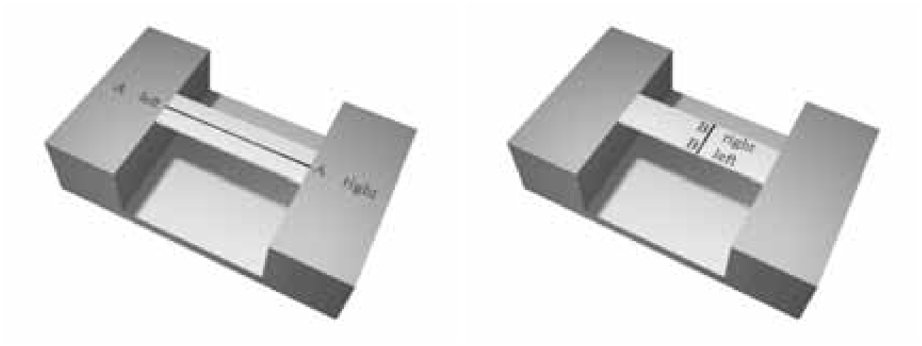


Figure 4.8: A simple bridge. It is connected to the bare earth at A left and A right. It is also disconnected from the bare earth across its length, B left and B right.

B10 If corresponding first and last returns are spaced far apart, then the first return is object.

B11 Bridges are connected to the bare earth along their width: In figure 6.8 the bare earth is connected to the bare earth at A(left) and A(right).

B12 Bridges are connected to the bare earth on at least two sides. See figure 6.8.

B13 Bridges are greater in length than in width. See figure 6.8.

B14 Along the length of a bridge, diametrically opposite points on the perimeter are raised above the bare earth: In figure 6.8 the points at B(left) and B(right) are above the bare earth.

Comparing the new assumptions against assumptions A1 through A10, it can be seen that the analysis of a point cloud has been shifted from the study of the geometric and radiometric characteristics of points and placed on the geometric and radiometric characteristics of segments. What should stand out is that the final filter will be based on a series of assumptions rather than a single assumption. The reasoning behind each assumption will be explained in chapter 6. In the next section, an algorithm framework that encapsulates the new assumptions shall be outlined.

4.4 Algorithm framework

Figure 4.9 shows the overall filtering process. This flow is constructed of a series of segmentation-classification steps. The rationale behind the segmentation-

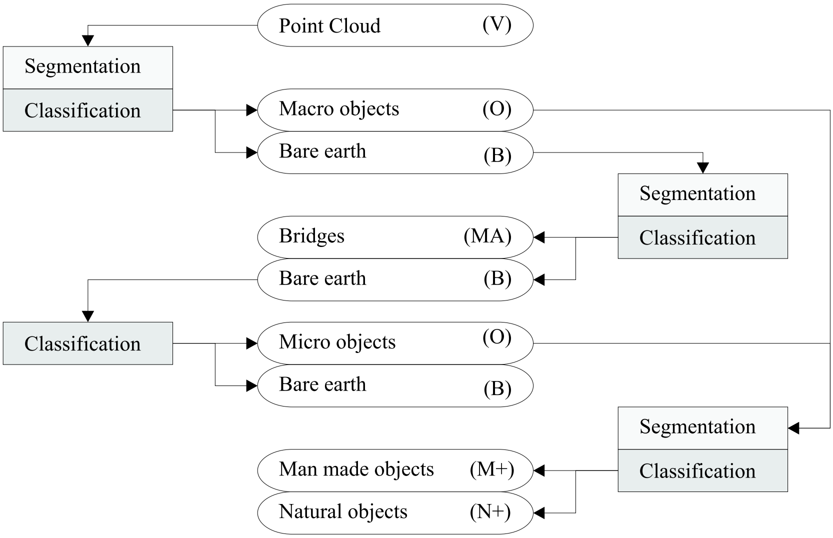


Figure 4.9: Overall filtering flow

classification steps shall now be explained.

1. Detecting macro objects

   In the logical model separation is in the first instance made between the bare earth and object. With respect to assumption B3 the emphasis is placed on detecting object points. The points that remain after the detection of objects are treated as being potentially bare earth. Potentially, because these remaining points may still contain bridges.

   The filter test in chapter 3 showed that Type I and II errors have to balanced. This balancing is necessitated by the fact that it is difficult to detect macro (large) and micro (small) objects simultaneously. To avoid the balancing of Type I and II errors the object detection process is split so that macro and micro objects are detected separately. This first step detects macro objects, primarily because they are the most prevalent objects in a landscape.

   The point cloud is segmented using assumption B1 and objects are detected based on assumption B4, B5, B6 and B7.

2. Detecting bridges

   The remaining points are segmented again and bridges detected. Bridges are detected based on assumptions B11, B12, B13 and B14. The remaining points are treated as being potentially bare earth. Potentially, because these remaining points may still contain micro objects and points whose classification may never be certain.

3. Detecting micro objects

   This is the second part of the object detection. In this step, there is no segmentation because objects are of a size that would not yield meaningful segments. Micro object detection may result in the removal of bridge edges that are essential for bridge detection. For this reason, micro object detection is done after the bridge detection.

4. Detecting man made and natural objects

   Points classified as object are segmented and then classified as man made object or natural object based on assumptions B8 and B9.

Different segmentation procedures are used at each step (except step 3). The reasons for their use and their mechanics shall be explained in chapter 5. The classification of the segments is then explained in chapter 6

Chapter 5

Segmentation

5.1 Introduction

The purpose of segmentation is to obtain higher level information from the points in a point cloud. This information is usually knowledge of the extent of homogeneous regions in a landscape. These homogeneous regions can later be classified with respect to the contents of a landscape, e.g., buildings, vegetation, and bridges.

In the algorithm framework set out in section 4.4, a point cloud is segmented and classified in five steps. The segmentation approach employed in each step is different and suited to the features being sought. For some of the steps the existing segmentation procedures were deemed inadequate. Therefore, in this respect a novel segmentation approach is proposed here specifically for the detection of objects and bridges.

The purpose of this chapter is to discuss the segmentation strategies employed. Firstly, possible segmentation algorithms are discussed. Next, the choice of segmentation algorithms is explained. Finally. the new algorithm is rationalized and detailed. The classification of these segments is later explained in chapter 6.

The limitations of segmentation

With respect to the conceptual model one of the assumptions made was that the surface of each object should be encompassed by one segment (assumption B4). For example, all points belonging to a building segment should be delimited by its roof outline. This presents a problem because if a building is covered by a multi-tiered roof, each roof outline will be considered a different object. Because of this the assumption cannot be satisfied. On the other hand trees are distinguishable by clustering points based on the proximity between points. Because of this, trees with interlocking branches will be merged into the same segment. Again the assumption cannot be satisfied. The objectives of this research do not require that buildings or whole trees in their entirety be identified. Therefore, the above limitations should pose little ill effects on the classification of points.

However, because of the above limitations buildings and trees standing close to each other may also be merged into the same segment. This mixing of different features is difficult to overcome and will affect the classification of such segments.

The homogeneity of segments

A segment is characterised as being homogeneous. This homogeneity of segments is mainly determined by the geometric constraints placed on point neighborhoods. The choices of constraints are themselves determined by the application and purpose of the segments [1]. Typically, these geometric constraints are based on variations in gradient, and curvature in the neighborhood of a point.

This neighborhood can be broadened so that it includes a mass of points (as opposed to a few points). The homogeneity of a segment can then be defined in terms of this mass of points conforming to a mathematical surface. This is an attractive means of expressing homogeneity because it emphasizes on the form of a surface rather than local variations as are defined by gradient and curvature. However, expressing discontinuity in this fashion is not without its problems. Surfaces in the real world contain discontinuities within themselves and therefore the mathematical surfaces have to accommodate discontinuities.

This difficulty can be overcome to some extent by expressing homogeneity in terms of proximity. The rationale being that on a discrete surface points are closest to their immediate neighbors. As such, points that are within a certain distance of each other must belong to the same surface. This lends to the idea of surface reconstruction based on proximity graphs (Klein and Zachmann, 2004).

To conclude this section the influence of ALS data on homogeneity (and as such segmentation) is briefly discussed. The characteristics of ALS data influence segmentation in two ways:

1. *the resolution of the data* - In low resolution ALS data objects lose definition and therefore homogeneity constraints have to be relaxed. High resolution ALS data suffer from low amplitude and high frequency noise, which offers problems for homogeneity constraints based on measures like curvature.

---

[1] This application dependency of segments means that no segment is good or bad, but rather fit for a particular use

2. *overlapping surfaces* - In the real world objects sometimes overlap each other or the bare earth. For example, bridges overlap the bare earth and the branches of trees hang over roofs. Because of this neighborhood systems that are based on planimetric proximity will show great variety (gradient, curvature, etc.,) in the overlap regions. Therefore, neighborhood systems should ideally be based on proximity in 3D and hence allow for segments that overlap in planimetry.

5.2 Previous work

Considerable work has been done in segmentation as shown by the following comparison studies and reviews (Fu and Mui, 1981; Haralick, 1983, 1985; Hoover et al., 1996; Sahoo et al., 1988; Zhang, 1997) . While these algorithms are mostly applied to range images, they are adaptable to irregular point clouds. A study of published segmentation algorithms shows that the algorithms fall into broadly four groups:

1. Pattern based techniques - Cluster analysis[2]
2. Edge detection techniques
3. Graph based techniques
4. Region growing techniques

Segmentation will be explained as:

$$\Theta V = \{\theta v \quad | \quad \forall v \in V\} \tag{5.1}$$

Where ΘV is the segmentation operation on a point cloud, V, and θv is the segment/cluster assignment for a single point, v, in V. The result of the segmentation are segments, s, with the following properties:

1. $S = \{s | s \subset V\}$
2. $\Theta V \Rightarrow S$
3. $\bigcup s_i = V \quad where \quad |s_i| > 0$
4. $s_i \cap s_j = \emptyset \quad where \quad i \neq j$

That is, each segment is a closed subset of the point cloud, V, the segmentation operation, Θ, will determine the character and number of segments, every point in the cloud belongs to a segment, and no two segments have points in common.

---

[2] Grouping of points based on their relationships in an n dimensional feature space.

Pattern based techniques

In these algorithms, n number of geometric or radiometric measures (features) are determined for each point, v. These n features are represented in an n-dimensional feature space. Thereafter, points that occupy compact and disjoint regions (called clusters) in the feature space are treated as distinct segments. Typical features used include the position of an ALS point (this is necessary), the surface normal at an ALS point, the coefficients of a best fitting surface, and the reflectance of an ALS point. The surface normal is estimated by the best-fitting plane (using least squares adjustment) in a small neighborhood of an ALS point.

Common techniques used for demarcating the feature space include K-Nearest Neighbour (KNN), and Maximum Likelihood methods. For more on feature space demarcation techniques the reader can refer to (Jain and Dubes, 1988; Jain et al., 1999, 2000). The performance of the algorithms is dictated by the choice of features and feature space partitioning technique.

Some examples of pattern based segmentation are presented below to demonstrate possible approaches.

1. Jain and Hoffman (Hoffman and Jain, 1987; Jain and Dubes, 1988) describe the segmentation of range images using the position, depth, and normals at pixels to obtain six dimensional pattern vectors. The planimetric position (x,y) of the pixels serve to provide connected segments in planimetry. The depth (z) provides connected segments in height and the normals serve as a means for detecting "crease" edges (where two surfaces meet sharply). To avoid bias each of the features is normalized so that they have unit variance.

   Many clustering techniques were tried and a form of K-means algorithm was found to perform best. The authors indicate that to obtain a good segmentation result, the clustering algorithm should be directed to over segment the point cloud, and then merge the segments in a post processing step. This is done to avoid under segmentation in the event that feature space is too complex.

2. Filin (2002) applies a clustering algorithm to ALS data using the position, the parameters of a best fitting plane at a point (plane fitted to the neighbors of a point), and the average height difference of a point to its neighbors to form seven-dimensional pattern vectors. The feature space is clustered using a mode seeking algorithm. In a post clustering step, clusters, in the feature space, with a cardinality below a given value are rejected because of insufficiency. The clusters that remain are then further tested to determine if they contain outliers or whether they need to be split. Finally clusters that share similarities are merged.

3. Roggero (2002) applies a hierarchical clustering algorithm to ALS data using

the second order symmetric 3D tensor for ALS points to form n dimensional pattern vectors. Using principal component analysis (PCA) on the n features a five-dimensional feature space is generated. Some of the features used include, the total static moment, curvature and the reflectance of the ALS points. The algorithm is rather involved and will not be further detailed here. Unlike the algorithms by Jain and Filin, this method includes a region growing scheme alongside the clustering scheme that avoids the need to merge and split clusters in a post processing step.

4. Other pattern based algorithms exist but they have only been applied to range images of close range objects. Therefore, their efficacy for segmenting ALS data has not been tested. Some other examples of pattern based segmentation algorithms are provided in an experimental comparison of segmentation algorithms by Hoover et al. (1996).

Edge based segmentation

Some examples of edge based segmentation have already been discussed in section 2.3. Essentially the algorithms search for planimetric edges in ALS data. If edges forming closed boundaries are found, then all points within a closed edge are deemed to belong to the same region. Therefore, the performance of the segmentation is determined by the edge detector.

Typically, in such segmentation algorithms the data is first converted to a raster. This makes possible the application of classical image edge detection schemes (Abdou and Pratt, 1979; Davis, 1975; Fram and Deutsch, 1975; Heath et al., 1996; Peli and Malah, 1982; Schachter and Rosenfeld, 1978).

Proximity graph based segmentation

Proximity graph based algorithms are based on the simple notion that points within segments are closer to each other than they are to points in other segments. The algorithms begin by imposing an attribute graph $G(V, E_A)$ on the point cloud, V. The attributes of the edges, E_A, are based on a predefined proximity measure. Edges in G that do not meet a defined proximity criterion are identified and removed to yield a reduced graph in which the connected components are treated as segments.

Zahn (1971) uses a minimum spanning tree (MST) graph and defines inconsistent edges (edges that fail to meet the proximity criterion) as those edges whose weight is significantly greater than the average of nearby edge weights. In this case, the edge weight is given by the Euclidean length of the edge. If weights are stored in the edge attributes A, then the algorithm can be represented as follows:

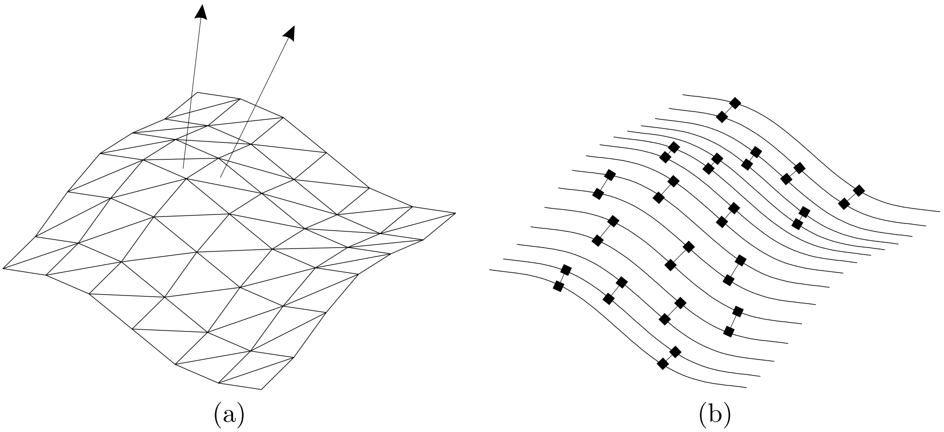


Figure 5.1: Region growing. Region growing in TINs (a), region growing by scan lines (b)

1. Impose minimum spanning tree on the point cloud, V, $G_{MST}(V, E_A)$.
2. Weed edges whose weight (A) is greater than a give threshold, α, $G_{MST}(V, E_{A\leq\alpha}) = G_{MST}(V, E_A) - G_{MST}(E_{A>\alpha})$.
3. Segments are given by $s_i = (V|V \in G_i(V, E_{A\leq\alpha}))$ where $G_i(V, E_{A\leq\alpha})$ is a connected graph in $G_{MST}(V, E_{A\leq\alpha})$.

Additional to MST, other useful graphs are Delaunay triangulations, relative neighborhood graphs (RNG) and Gabriel graphs (Bose et al., 1993).

The algorithms by Axelsson and Sohn (section 2.3) are a variant of the graph based algorithms in that they search for only one segment. This segment is formed by triplets of points that have the property that they are close to each other. However, they differ in that unlike the conventional method described above, they are formed by repeatedly adding rather than removing edges.

Segmentation by region growing

In these algorithms typically planar or non-planar surface patches, seeds, are first selected in the point cloud and then these patches are connected or grouped based on adjacency measures (difference in normals, slope difference, curvature difference, etc.,), figure 5.1(a). A typical example region growing algorithm is presented by Gorte (2002).

Another implementation of a region growing algorithm on ALS data is presented by Lee and Schenk (2001, 2002). They impose a Delaunay triangulation on a point cloud and connect the triangles in the TIN based on Gestalt principles.

Other forms of region growing take advantage of the fact that point clouds are usually built from sequences of scans. In these approaches, points are first joined along scans, using polynomials or splines. The scan lines are then joined by proximity functions. In effect there is a region growing along the scans and then another across the scans. This is shown in figure 5.1(b). Such an algorithm was proposed by Jiang and Bunke (1994) and although simple in concept it has been found to perform favourably when compared to other algorithms (Hoover et al., 1996). An example and extension of the algorithm has been presented by Patane and Spagnuolo (2002).

5.3 New segmentation strategy for object and bridge detection

The filter test demonstrated that many of the current filter algorithms do not do well at discontinuities in the landscape. Including a segmentation step in the filter algorithm is designed to overcome this problem. Points are gathered together in one segment and the segment is then classified. This is in opposition to classifying points one at a time.

Segments are classified based on their geometric, radiometric and topological characteristics. However, the segments obtained by the four different approaches are not always ideal for the application here. Before explaining the necessity of a new segmentation approach, the usefulness of the segments yielded by the current approaches is briefly discussed below.

> *Pattern based techniques* - A problem with this algorithm is choosing the right features and normalizing them to avoid bias. Feature based clustering in ALS avoids geometric modelling on the presumption that geometric models describing the data are not readily discernible. The problem with this is that it is difficult to establish the relationship between the segmentation criterion and the properties of the desired segments. For this reason pattern based segmentation algorithms were deemed unsuitable for the segmentation here.
>
> *Edge based segmentation* - While essentially a sound concept, edge based algorithms suffer when the data is noisy (e.g., low vegetation), which can lead to open boundaries, multiple edges, and disconnected edges. Additionally, to obtain appropriate closing boundaries, the data also has to be processed

at appropriate scales. Automating the determination of this scale can be problematic. However, there are other aspects of edge based algorithms that make them unsuitable. Edge based algorithms cannot cope with overlapping surfaces in ALS data. This means that the edges of bridges and buildings can be under segmented.

Proximity graph based segmentation - Proximity based algorithms assume that a point in an object is closer to its neighbours than it is to points in other objects. This is a valid assumption considering that ALS scans are done from high altitudes using small opening angles and therefore the density of points at discontinuities (e.g., walls) is very small. Unlike edge based segmentation, proximity segmentation can work in the presence of surface overlaps. Because of this, proximity segmentation was considered suitable. In the event that other variables like reflectance are used to determine edge weights, the consistency measure becomes especially critical and prior knowledge of the relation between geometric, radiometric and semantic characteristics has to be available.

Segmentation by region growing - If the components in the region growing are triangles in a 2D TIN, then the algorithm will have problems in noisy data and data in which points from different surfaces overlap. However, the component connectivity inherent in region growing algorithms introduces topology, which is important in the classification stage. Because, of this, region growing algorithms were also deemed suitable.

While segmentation by proximity and region growing have desirable strengths they share a weakness in that the segments they yield do not carry topological information that allows handles in genus 1 (or higher) surfaces to be detected. Bridges are examples of handles in genus 1 (or higher) surfaces. The challenge was to devise a segmentation approach that explicitly derived the above topological information and combined the strengths of region growing and proximity based segmentation. This was achieved by modifying the scan line based segmentation algorithm. The sections to follow explain how this was achieved.

Concept

A surface can be thought of as a field with an infinite number of points, defined by an implicit equation $f(x, y, z) = 0$. If the surface contains discontinuities, it can be approximated using multiple parametric surface patches. Alternatively, the surface can be approximated by a web of planar curves all passing through the points. The greater the number of curves the better the representation of the surface. In scan line algorithms the planar curves are all parallel, i.e. they all run in the same direction. Here however, the planar curves are allowed to

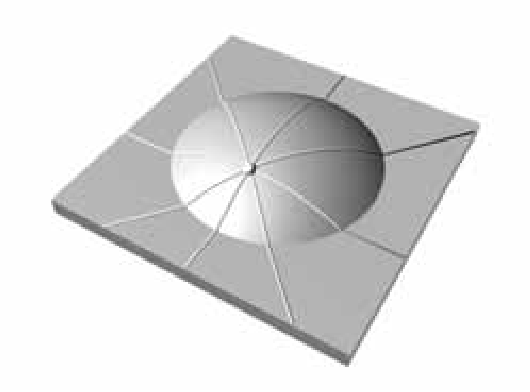


Figure 5.2: Surface representation using planar curves

run in all directions. Figure 5.2 shows an example of four planar curves passing through a single point on the surface. The curves (profiles) become approximations of the surface, and interpolation of points on the surface is most accurate at the intersection of the curves. To approximate the surface a set of planar curves would have to be defined at a suitable number of points on the surface.

Figure 5.3(a) demonstrates how a surface is reconstructed using profiles. Shown in the figure is a piecewise continuous surface. A number of profiles are defined. Because the surface is piecewise continuous the profiles will terminate at the discontinuities. If two profiles intersect then that is an indication that the points on those profiles belong to the same surface. In this way by testing the intersection of profiles, points belonging to the same surface are identified.

The preceding discussion assumed a continuous domain. However, the domain of an ALS point cloud is discrete. The effect of this is that:

1. The extent of profiles has to be restricted to a given radius about a point, on the reasoning that the surface representation becomes poor the farther we travel from the intersection point.
2. The number of profiles passing through a point has to be kept finite for practical reasons (e.g., resource limitations).
3. Profiles cannot be thin, but rather have to be thick in order to capture a sufficient number of points to sample surface characteristics. The point spacing will determine the adequate profile thickness.

This segmentation approach has some distinct advantages:

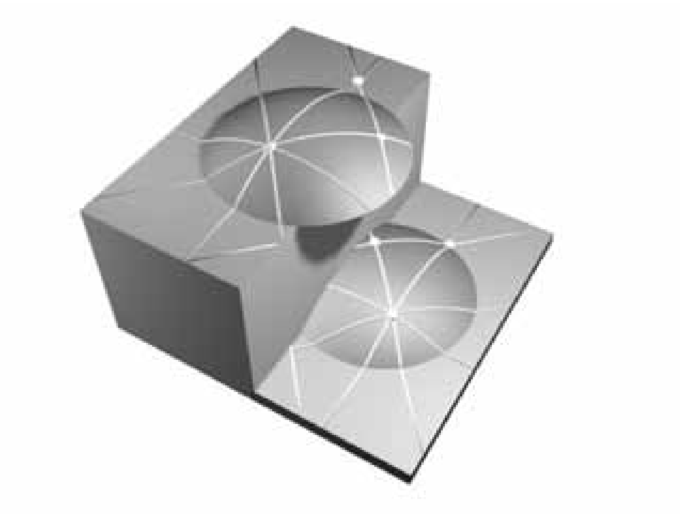


Figure 5.3: Segmentation by profile intersection.

1. Profiles simplify structure detection in a point cloud. This is an important aspect of the segmentation strategy and is the basis of the novel bare earth-object and bare earth-bridge classification.
2. It allows segmentation of overlapping surfaces.
3. It is fast.

The main disadvantages of the approach are that it is memory intensive and the concept can breakdown if insufficient profiles are used.

Segmentation by profile intersection - the algorithm

Figure 5.4 shows a simple example of the segmentation approach. The point cloud contains two surfaces, the top of the plate represented by white circles and the top of the cube represented by black circles. The point cloud is partitioned in three directions (top). In every profile, points are connected if they are on the same surface. When the three partitions are overlaid, points on the same surface interconnect (bottom). Figure 5.5 shows how the algorithm works for overlapping surfaces.

In a point cloud the surfaces are not known. Therefore, points are connected if they meet predefined criteria. Different criteria for connecting points shall be

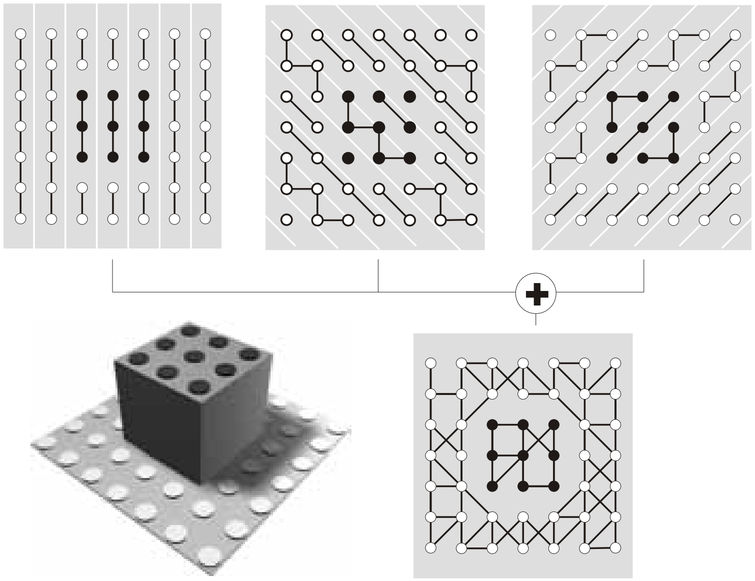


Figure 5.4: Segmenting a point cloud. Two surfaces (the top of the plate and the top of the cube) are segmented using three different profile directions.

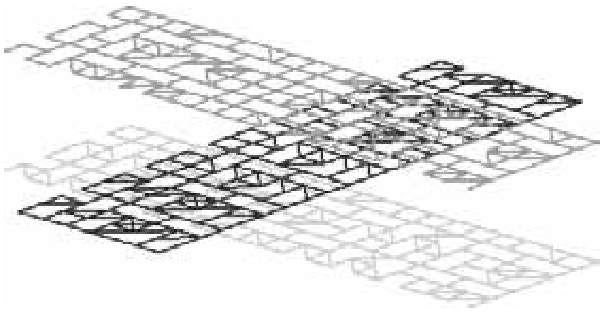


Figure 5.5: Overlapping surfaces segmented

discussed in section 5.3. The profile segmentation algorithm consists of three steps (also described in (Sithole and Vosselman, 2002a, 2003a)).

1. A point cloud is partitioned to yield a series of profiles lying at different orientations, φ.
2. Points in the profiles are connected to yield line segments. Points on the

same line segment are deemed to be off the same surface.

3. Profiles passing through the same point are assigned to the same segment.

Each step is explained in the sub sections below.

Partitioning the point cloud

A point cloud is partitioned by slicing it into contiguous planar profiles. Each profile has a thickness w and is oriented in the direction φ in the xy plane. Because all profiles in a given direction are contiguous, points within a profile are not shared with other profiles in the same direction. Because profiles are run in different directions, a family of orientations is defined.

$$D = \{\varphi | \varphi \in \mathbb{R}\} \tag{5.2}$$

The family of profiles running in the same direction are

$$P_{(w,\varphi)} = \{p(w,\varphi) | p \subset V\} \tag{5.3}$$

Where p is a single profile in the family P. The profiles, P, have the following properties:

1. $\bigcup p_i = V$
2. $p_i \cap p_j = \emptyset, \quad i \neq j$

That is, the point cloud is the union of all profiles with the same orientation, and that no two profiles with the same orientation share common points and that points in the profiles are sequentially ordered.

Segmenting the profiles

After partitioning the point cloud, each profile is in turn itself further segmented to yield line segments l.

$$\Theta p_{(w,\varphi)} = \{\theta v | \forall v \in p_{(w,\varphi)}\} \tag{5.4}$$

Each profile, p, has the following properties:

1. $\Theta p_{(w,\varphi)} \Rightarrow L$ where L is a set of line segments

2. $L_i = \{l | l \subset V\}$
3. $l_j \cap l_k = \emptyset, \quad where \quad j \neq k$

That is, the segmentation operation Θ on a profile, p, will determine the character and number of line segments, no two line segments share common points and points in the line segments are sequentially ordered. It should be noted that the line segments are only defined in a 2D plane.

Surface Segmentation

The overlaying of the line segments yields a disconnected graph[3], G, in which the connected sub-graphs, G_i, are the desired surface segments.

Because G is a disconnected graph no two segments share points in common. From the above it can be seen that the segmentation depends on:

1. The thickness, w, of the profiles.
2. The number of directions, φ, in which the profiles are run.
3. The profile segmentation operation $\Theta p_{(w,\varphi)}$.

Of the three, the profile segmentation operation has the greatest impact on the final segmentation. The next section will therefore look at possible profile segmentation operations.

Profile segmenting operation

To simplify the segmentation of the profiles, each profile is first transformed from a 3D frame to a 2D frame, sorted and indexed along the profile. Sorting and indexing of the points ensures that correspondence between points in both frames is preserved.

The profile segmenting operation is achieved in a two step process. The first step involves labeling points that belong together, and in the second step points that have the same label are connected based on their order. There are several methods by which labeling can be done, and these are detailed in the sections below:

---

[3] Except in the event that one surface segment is obtained, in which case the graph G is connected.

Labeling by curve fitting

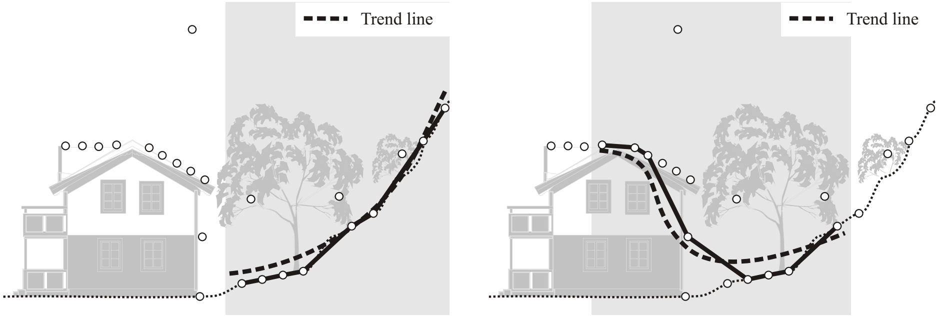
This approach works on the premise that a set of points that fit a parametric curve must be off the same surface. The algorithm works as follows:

1. Select a curve function.
2. Place a window of width δ at the beginning of the profile.
3. Select all points inside this window.
4. Fit a curve to the selected points, figure 5.6(a) (in the figure the window is not at the beginning of the profile, and the figure is simply used to demonstrate the curve fitting).
5. In the window select those points that are within a given distance ϵ_h to the curve.
6. If of the selected points in step 5:
    (a) none have a label, then obtain a unique label and assign it to them all.
    (b) one of the points has a label but the others not, then transfer this label to them all.
    (c) many of the points have a label then find the mode label and assign it to them all.
7. Move the window, but ensure that it still overlaps the previous window position.
8. Repeat steps 3 to 6.
9. When the window has reached the end of the profile and the points in it are labeled, then repeat steps 2 to 8 for all unlabeled points (recalling that only one curve is detected in a window at a time), until no unlabeled points remain.

The curve function chosen is typically a quadratic or cubic polynomial, and fitting is done by least squares. However, noise in the profile leads to a poor fit. One means of overcoming this problem is to use an iterative least squares fitting scheme as used by Pfeifer et al. (1999), 2.3.

The success of labeling depends on the curve fitting method, the noise in the points, and the width of the window.

> *Noise* - A curve will attract to a meaningful surface if there are an adequate number of representative points from that surface. However, noise (points that belong to other surfaces) will attract a curve away from the surface.



(a) (b)

Figure 5.6: Labeling by curve fitting

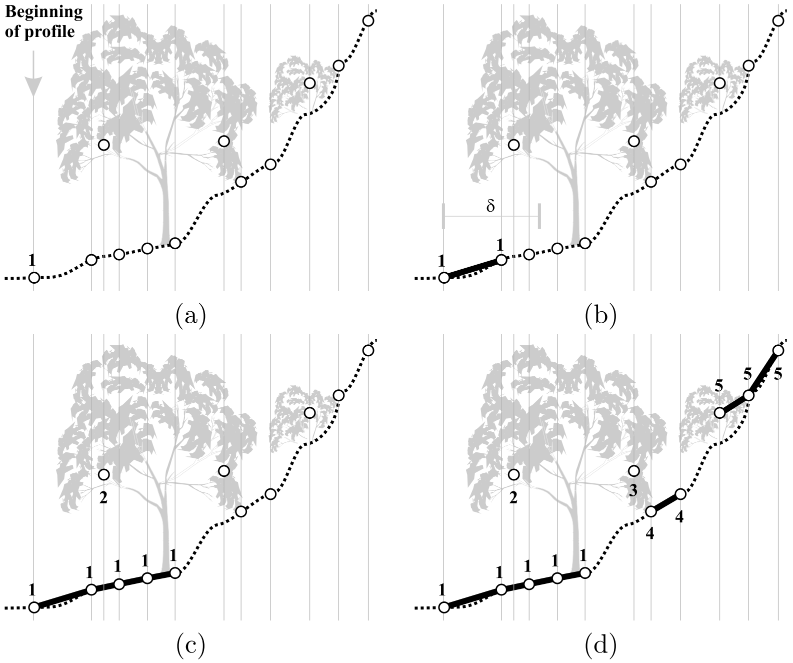
Width of the window - A successful fit depends on there being only one surface in the window. In figure 5.6(b) there are two surfaces (the roof of the building and the bare earth). The fit has traced a single curve between the surfaces, as a result the labeling algorithm fails.

A major drawback with labeling by curve fitting is that it assumes that there is a continuous curve within the window, figure 5.6(a). However, particularly in urban areas this is not the case, figure 5.6(b). This means that the curve fitting has to be coupled with a discontinuity detection. This is an inconvenience and even then labeling is not always predictable. The methods described below proved to be more predictable.

Consecutive Labeling

In this approach, two points are tested at a time. A continuity criterion enforced on a geometric relationship between two points hypothesizes whether the two points are off the same surface. Because points in the profiles are sequentially ordered according to the distance from the ends of the profile, a test of adjacent points simulates a simple random walk through the profile.

1. Assign a label to point v_1; figure 5.7(a).
2. Test points, v_i and v_{i+1}; figure 5.7(b). If v_i and v_{i+1},
   (a) satisfy the continuity criterion, then assign the label of v_i to v_{i+1}, advance i by 1 and repeat step 2; figure 5.7(c).
   (b) do not satisfy the continuity criterion, then search for the next nearest forward point in the neighborhood, δ, of v_i that satisfies the continuity



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

Figure 5.7: Consecutive point labeling. Move to beginning of label (a), label adjacent points (b), extend the label to other points (c), continue labeling unlabeled points (d)

criterion. Next, assign it the label of v_i, and continue the labeling from this new point; figure 5.7(c).

If a satisfactory point in δ cannot be found, then move to the first unlabeled point in the profile. Assign this point a new label, and continue the labeling procedure from this point; figure 5.7(d).

3. Stop the procedure when all points have been labeled.

Two different continuity criteria were tried:

1. *Slope*
   $|\Delta x(v_i, v_j)/\Delta h(v_i, v_j)| < \epsilon_{slope}$ in the neighborhood δ of v_i.

2. *Epsilon-Delta*
   $|\Delta h(v_i, v_j)| < \epsilon_h$ in the neighborhood δ of v_i.

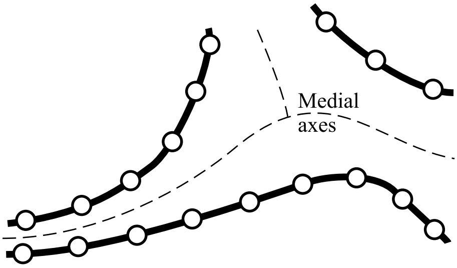


Figure 5.8: Medial axis between curves

Where $\Delta x(v_i, v_j)$ and $\Delta h(v_i, v_j)$ are the planimetric distance and height difference between points v_i and v_j respectively. The limits on the criterion are given by the threshold ϵ.

Labeling with the slope criterion is used in the bridge detection algorithm (section 6.3), and labeling with the epsilon-delta criterion is used in the object detection algorithm for relatively flat landscapes. (section 6.2).

The advantage of the approach is that it is both simple and fast, which is desirable considering that typically thousands of profiles have to be processed. The weakness of the approach is that it does not do well in steep slopes.

Labeling by Crust curve

Amenta, Bern and Eppstein (Amenta et al., 1998), propose a novel approach of searching for a curve in a planar set of points.

The concept of their approach is that there exists a medial axis between two or more curves, figure 5.8. If in a point set medial axes can be found, then all points on the same side of the medial axes must belong to the same curve.

The algorithm works as follows:

1. Generate a Voronoi diagram from the points in a profile, figure 5.9(a). The medial axis can be discerned by connecting the Voronoi nodes.
2. Delaunay triangulate the original points and Voronoi nodes combined, figure 5.9(b).
3. Obtain polygonized curves by connecting all the original points that are linked by an edge in the Delaunay triangulation, figure 5.9(c).

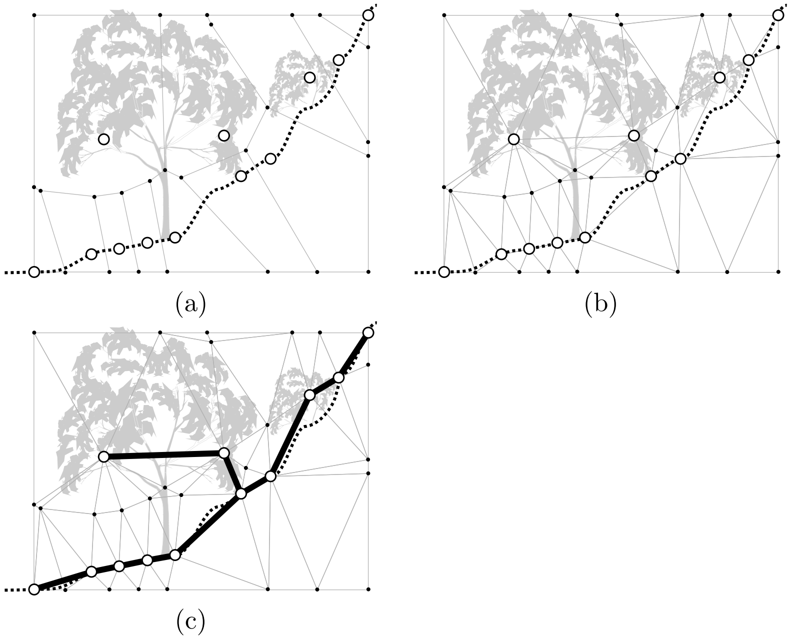

(a)
(b)
(c)

Figure 5.9: Labeling by crust. Voronoi diagram of original points (a), Delaunay triangulation of original and Voronoi points (b), connecting all original points joined by an edge (c).

4. Assign to each polygonised curve a unique label and later transfer these labels to the points that make up the curves.

A characteristic of the approach is that the required sampling density varies with the local feature size on the curve so that areas of low detail can be sampled less densely.

Tested on high density data sets the algorithm was found to perform poorly. High density ALS data usually contains low amplitude high frequency noise. Because the crust curve algorithm adapts to point spacing crust curves are found even within the noise. The result of this is that many thinly spaced curves are detected in the bare earth. For this reason the crust curve algorithm was found unsuitable for curve detection in profiles.

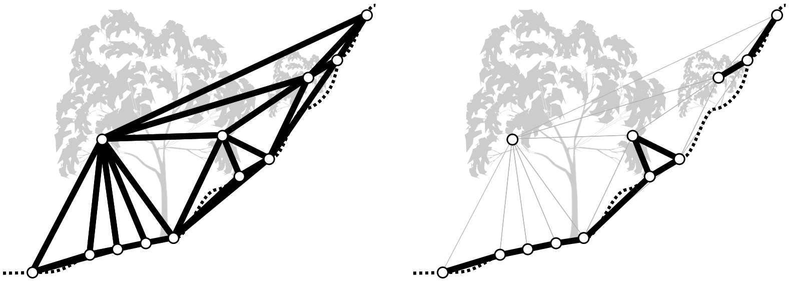

(a) (b)

Figure 5.10: Labeling by proximity. Delaunay triangulation (a), removal of edges that that are greater than a given weight (b).

Labeling by proximity

This approach works on the assumption that points on the same curve are closer to each other than they are to points in other curves. The algorithm works as follows:

1. Delaunay triangulate a profile p, $G_{Prox}(p, E_A)$, figure 5.10(a).
2. Compute a weight, A, for each edge.
3. Remove edges whose weight is greater than a given threshold, α, $G_{Prox}(p, E_{A\leq\alpha}) = G_{Prox}(p, E_A) - G_{Prox}(E_{A>\alpha})$, figure 5.10(b).
4. Assign to each line segment a unique label and later transfer these labels to the points that make up the line segments.

The weight A is given by:

$$A = (x_j - x_i)^2 + k(y_j - y_i)^2 \tag{5.5}$$

Where the weight A is a proximity measure between edge end points v_i and v_j, and (x_i, y_i) and (x_j, y_j) are the coordinate tuples of points v_i and v_j respectively. The closer two points are to each other the smaller the weight A. The proximity field is scaled differently along the x and y axes so that points along the horizontal are closer to each other than points along the vertical. This variable scaling is done on the assumption that in an ALS point cloud, “close points on the horizontal are much more likely to belong to the same surface than points close on the vertical”.

The variable scaling is achieved using the parameter k (> 1). The proximity field defined by equation 5.5 represents an ellipse. The shape of this ellipse is

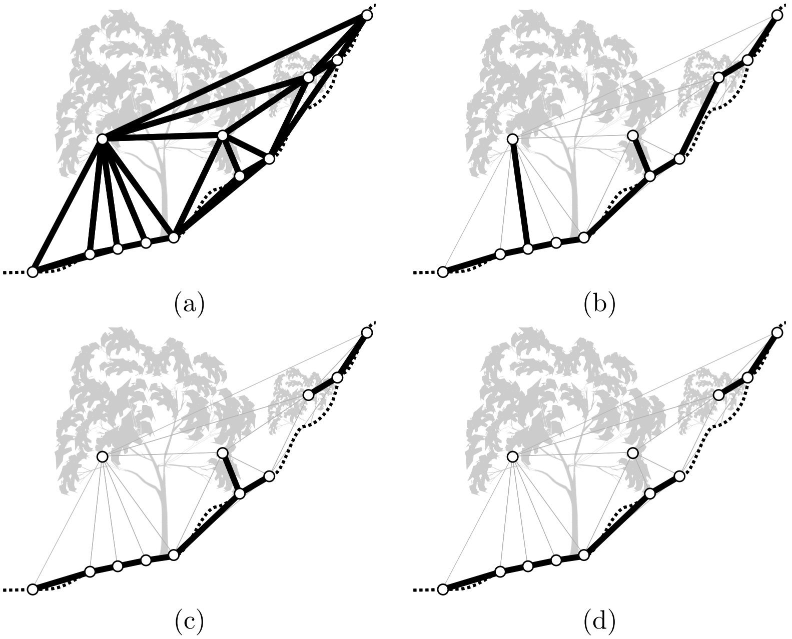


Figure 5.11: Minimum spanning tree. Delaunay triangulation (a), minimum spanning tree (b), removal of edges that that are greater than a given weight (c), removal of dangling edges (d).

determined by the parameter k. If the closest point allowable in planimetry is set equal to δ and the closest point allowable in the vertical is set equal to ϵ_h then $k = (\delta/\epsilon_h)^2$. Necessarily the threshold α (on A) is equal to δ^2.

This labeling approach is more robust to noisy data and works fairly well in steeper slopes. However, where there is low and dense vegetation (particularly shrubs and hedges) it has a tendency to merge them with the bare earth. This effect can be seen in figure 5.10(b). The next approach is designed to alleviate this problem.

Labeling using minimum spanning trees

This approach works on the principle that for a point in the neighborhood of n-points the two closest to it must be on the same curve that it is on. The curves that are obtained are determined by the distance measure chosen. The algorithm works as follows:

1. Delaunay triangulate a profile , figure 5.11(a).

2. Compute a weight, A, for each edge (the same as in the proximity labeling).
3. Determine a minimum spanning tree, $G_{MST}(p, E_A)$, based on the weights of the edges, figure 5.11(b).
4. Remove edges whose weight is greater than a given threshold, α, $G_{MST}(p, E_{A\leq\alpha}) = G_{MST}(p, E_A) - G_{MST}(E_{A>\alpha})$, figure 5.11(c).
5. Remove dangling edges, figure 5.11(d).
6. Assign to each line segment a unique label and later transfer these labels to the points that make up the line segments.

The minimum spanning tree[4] is a sub graph of the graph obtained in the proximity labeling. However, the minimum spanning tree has the advantage that it does not contain any cycles. Cycles are problematic because they strongly connect points to line segments. Hence, if a point in a cycle does not belong to a line segment it becomes difficult to identify and remove it (because deciding which point in a cycle should be removed is non trivial).

For the bare earth-object separation, this labeling approach was found to yield the most desirable segments, i.e., better separation between bare earth and object points in the segments. For this reason labeling by minimum spanning tree was used in all bare earth-object classifications (section 6.2).

The labeling by slope is used in the bridge detection and the labeling by proximity is used in the man made and natural object detection. The reason for these choices shall be explained in chapter 6.

5.4 Examples

To conclude this chapter the application of the algorithm is discussed using simulated data. Three examples are presented to show the mechanics of the algorithm and prepare the reader for the discussion on the classification procedures in the next chapter.

Examples of segmentation

The segmentation operations are performed on three simulated data sets. The composition of the data is designed to imitate difficult landscapes or show interesting scenarios.

---

[4] There are various algorithms for creating minimum weight spanning trees, but the one used in the implementation is Kruskal's algorithm as described by Hartsfield and Ringel (1994).

Data characteristics:
Slope of the bare earth 45°
Slope of the roof 60°

Segmentation characteristics:
Minimum span tree labeling
Profile thickness $= 2.0m$
Num. profile directions $= 3$
$\delta = 2.0m$
$\epsilon_h = 0.40m$

Figure 5.12: Segmentation example 1: Building on slope (point spacing 1-1.4m).

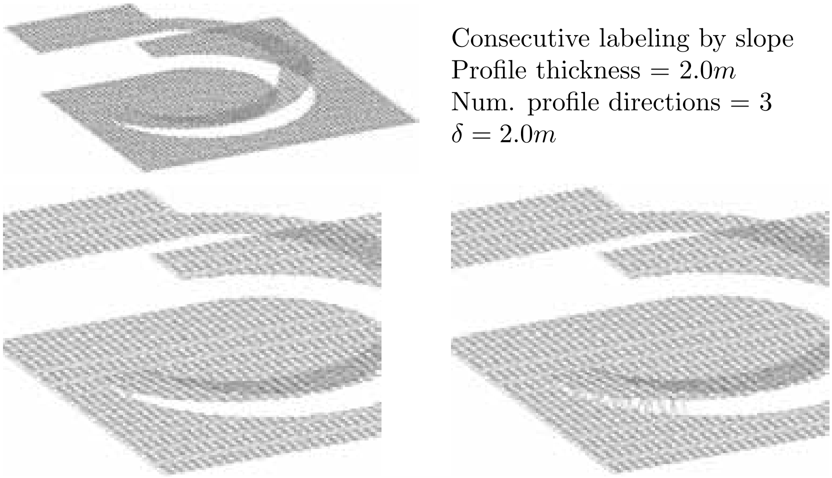
Example 1

This example, figure 5.12 presents a steep landscape on which there is a building with steeply slanted roofs. The segmentation yields two segments. The roof points are merged into one segment and the bare earth points into another. This is the desirable result.

The profile thickness is chosen a little greater than the point spacing. This ensures that the profile is thin and adequately samples the landscape along the profile. Too thick a profile results in a high point density across the profile, which causes the concept of the algorithm to break down, because the concept is based on the idea that profiles are thin. Too thin a profile results in a sparse profile that causes the concept to also breakdown, because the point cloud is discrete and too thin a profile results in a poor sampling of the landscape along the profile. Profiles are run in three directions. For object detection, a minimum of three orientation directions has been found to work well.

Typically the range δ is chosen to be greater than the point spacing. About twice the point spacing has proved to work well. However, in the presence of steep slopes the lateral separation between the bare earth and objects is reduced. In this case too large a δ causes objects to merge into the bare earth. For this reason in the example the neighborhood range δ is chosen a little greater than the point spacing.

The labeling of points is done using a minimum spanning tree, which does better in steeper slopes. For this reason in the filtering, the segmentations for (i) bare earth - object classification, (ii) natural object - man made object classification, all use a minimum spanning tree labeling procedure.



Consecutive labeling by slope
Profile thickness = $2.0m$
Num. profile directions = 3
$\delta = 2.0m$

$\epsilon_{slope} = 45°$

$\epsilon_{slope} = 75°$

Figure 5.13: Segmentation example 2: Bridge connected to the bare earth (point spacing 1-1.4m).

Example 2

This example, figure 5.13 presents two bare earth segments connected by a curved ramp. The ramp allows a shallow descent from the highest bare earth segment to the lowest. The segmentation yields only one segment because the bridge is connect to the bare earth. As in the previous example the profile thicknesses is chosen a little greater than the point spacing. Consecutive labeling by slopes is used to label the profiles. The choice of this labeling algorithm is explained in figure 5.14. The threshold on the slopes is set at 45°. This threshold may appear to be very large. In this case the purpose of the segmentation is to determine discontinuities along the ramp, which slope can exceed 45°. Furthermore, the benefit of the large threshold is that it allows points on steep bare earth slopes to be merged into the same segment.

Too large a threshold can result in the ends of a bridge to be under determined. This can be see in figure 5.13. An angle of 75° causes the lower ends of the bridge to blend into the bare earth.

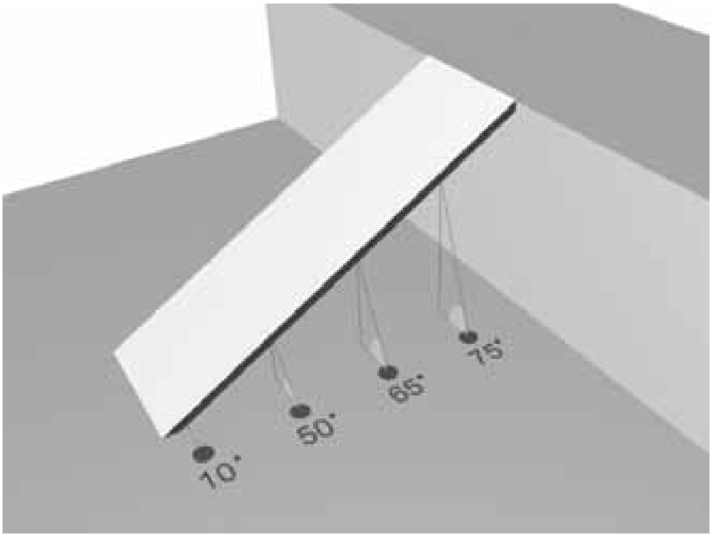
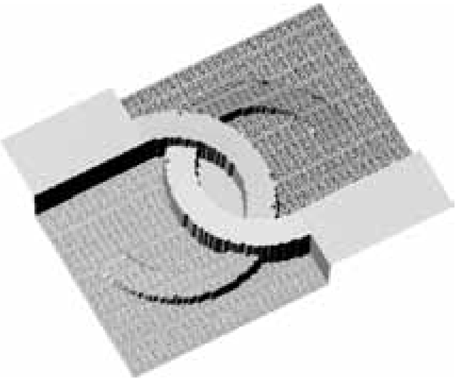


Figure 5.14: The slope of a bridge edge to adjacent bare earth. The slope is smallest at the lower end of the bridge. In a segmentation the chosen slope threshold determines where the bridge tears from the bare earth. The smaller the threshold the greater the tear and hence the better the determination of the bridge ends.



Consecutive labeling by slope
Profile thickness = $2.0m$
Num. profile directions = 3
$\delta = 2.0m$
$\epsilon_{slope} = 45°$

Figure 5.15: Segmentation example 3: Flyovers (point spacing 1-1.4m).

Example 3

This example, figure 5.13, is a variation on example 2. Three bare earth segments are connected by two ramps. Furthermore, the segments fly over each other.

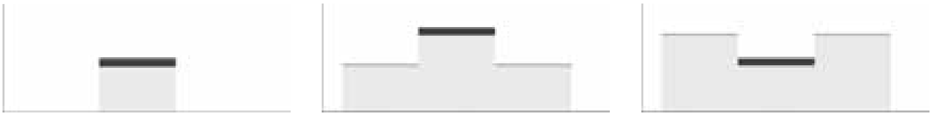
The segmentation yields five segments. This is the desirable result. Parts of the ramp are merged with the higher bare earth segments and the remaining parts with the lower bare earth segments. The flyovers break each others continuity. By extending the range, δ, it is possible to bridge these breaks, but the danger is that the flyovers could also be merged into the same segment, which is undesirable. Therefore, the range is kept small and the flyovers allowed to break into parts. As shall be shown later this does not have a negative effect on the classification of bridges.

5.5 Discussion

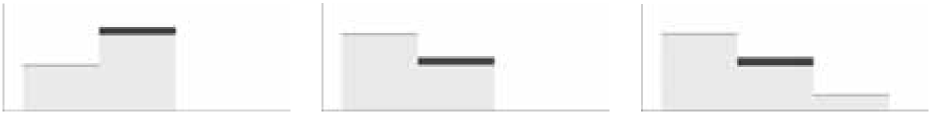
Line segment shapes

In section 5.3 it was stated that the new segmentation algorithm simplified structure detection. This simplification is made possible by the topological property of adjacent line segments in the profiles. From the topological properties of line segments additional topological properties can be obtained for the surface segments. Figure 5.16 shows nine possible relationships for a line segment (shown by a thick line) to neighboring line segments (shown by thin lines). These relationships are referred to here as the shape, $\psi(l)$, of a line segment. The nine shapes are:

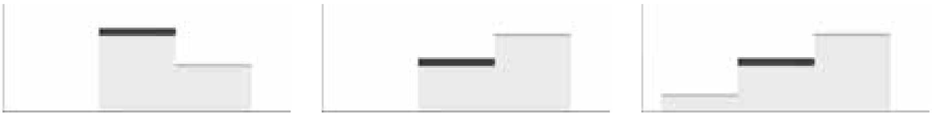
1. *No shape* - the line segment is not neighbored on either side.
2. *Raised* - the line segment is neighbored on both sides by line segments lower than itself.
3. *Lowered* - the line segment is neighbored on both sides by line segments higher than itself.
4. *Raised left* - the line segment is only neighbored on its left side, and by a line segment lower than itself.
5. *Raised right* - the line segment is only neighbored on its right side, and by a line segment lower than itself.
6. *Lowered left* - the line segment is only neighbored on its left side, and by a line segment higher than itself.
7. *Lowered right* - the line segment is only neighbored on its right side, and by a line segment higher than itself.
8. *Terraced left* - the line segment is neighbored on both sides, the line segment on the left being higher and that on the right being lower.



No shape Raised Lowered



Raised left Lowered left Terraced left



Raised right Lowered right Terraced right

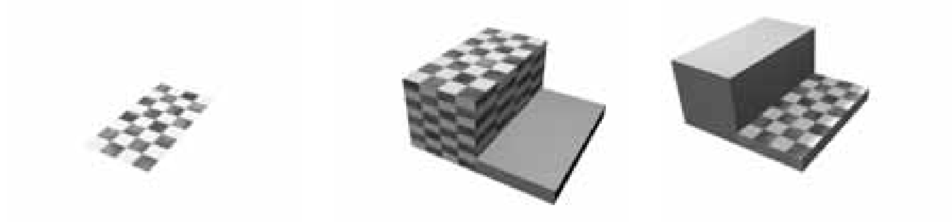
Figure 5.16: Line shapes. Relationships of a line segment (shown by the checkered surface) to neighboring line segments (shown by the gray surfaces)

9. *Terraced right* - the line segment is neighbored on both sides, the line segment on the left being lower and that on the right being higher.

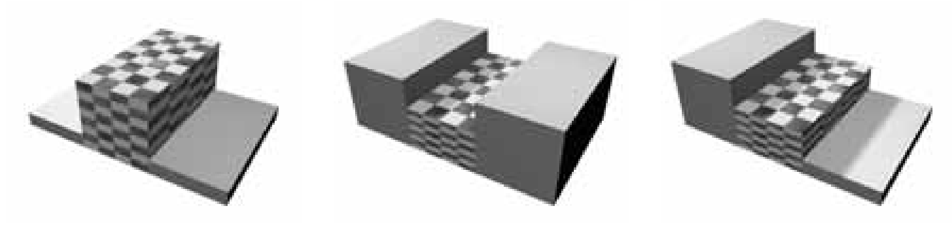
Just as the shape of a line segment describes its relationship to neighboring line segments, then similarly the line segments in a surface segment, together, describe the relationship of a surface segment to neighboring surface segments, figure 5.17. Therefore, similarly a surface segment is made from a combination of the following surface shapes, *No shape*, *Raised*, *Lowered*, *High*, *Low* and *Terraced*. The determination of the composition of the shape of a surface segment is explained in section 6.2. Based on the shape composition of a surface contextual classification can be done (as shall be explained in the next chapter).

Suitability of reflectance for controlling labeling

Typically, every point in an ALS point cloud comes with a value that represents the strength of the returned laser pulse. This fourth variable offers the possibility of extending the control of the labeling algorithms. The strength of the returned pulse depends on the spectral characteristic of the laser used and the properties of the reflecting surface. Therefore, in addition to labeling points based on proximity, points can also be labeled based on material composition. In other words if two adjacent points in a profile have similar reflectance values then they most likely



No shape High Low



Raised Lowered Terraced

Figure 5.17: Surface shapes. Relationships of a surface segment (shown by a thick line) to neighboring surface segments (shown by thin lines)

are off the same surface and therefore belong to the same line segment.

The use of reflectance data was considered and found to be unusable for three reasons. Firstly, reflectance data tends to be very noisy, as evidenced by the fact that two adjacent points on the same surface can have considerably differing reflectance values.

Secondly, surfaces are covered by many different types of materials and again reflectance values within a surface can differ considerably. For example labeling by reflectance would lead to an over segmentation of the bare earth.

Thirdly, the material covering buildings and vegetation is also found in the bare earth. Because of this in any segmentation based on reflectance, buildings will merge with pavements and trees with the ground (if covered by vegetation). This is undesirable.

Because of the above reasons, reflectance (as well as RGB triplets obtained from imagery) was not used to control labeling.

Suitability of first pulse data for controlling labeling

Some ALS point clouds are delivered as first pulse and last pulse returns. This also offers another means to control the labeling process. If the first and last pulses are far apart in height then the first pulse is assumed to be off an object. This assumption was thought to have potential for improving the labeling process. However, the use of first pulse data was found to fail in logic.

Firstly, the difference between the first and last pulse only provides evidence for the class (i.e., object) of the first pulse. The class of the last pulse remains unknown. Yet, segmentation is done on the last pulse (because in vegetated areas it contains more bare earth points than the first pulse).

Secondly, a discrimination based on the distance between the first and last pulse is essentially a classification of points as either *object* or *unclassified.* The purpose of segmentation is to organise points in such a way that higher level reasoning becomes possible. Therefore, the classification by testing the proximity between the first and last pulse negates the need for segmentation.

For the above reasons information based on first pulse data is not used to control the segmentation. However, as shall be explained in the next chapter it can be used in the man made - natural object classification.

Computational issues

It has already been mentioned that the algorithm is fast but memory intensive. On an AMD 800 MHz machine with 256MB of memory, using consecutive labeling 500000 points were processed in about 2 minutes. Using more elaborate labeling algorithms (minimum spanning tree or proximity) increases the processing time to about 3 minutes. This is a direct result of the Delaunay triangulations.

However, the rest of the algorithm, i.e., the profile generation and the intersection of the profiles are near linear operations. Therefore, provided memory is available the consecutive labeling of 1 million points can be expected to take about 4 minutes. The same though does not apply to the more elaborate labeling schemes. Therefore, the labeling of 1 million points using minimum spanning tree could last more than 6 minutes.

In terms of memory, most of the storage requirements are for keeping track of the line segments. Furthermore, each line segment and its parts are accompanied by internal variables that further increase storage requirements. These internal variables are kept in order to reduce the running time. Therefore, storage space is sacrificed for performance. This choice has been made on the expectation that advances in electronics will improve computer performance, and thus the algorithm's high memory usage will with time become less significant.

Chapter 6

Classification

6.1 Introduction

In chapter 4 a filter framework was proposed. This framework was developed based on new assumptions derived from a conceptual and logical modelling. In the framework the classification of a point cloud was divided into four steps:

1. Detecting macro objects,
2. Detecting bridges,
3. Detecting micro objects, and
4. Detecting man made and natural objects

This chapter details the algorithm designed for the detection of the features in each step.

6.2 Detecting macro objects

In the filtering framework, it was stated that segmentation should precede the classification of points. Using the characteristics of the segments points can then be classified. Each segment is described by a set of radiometric, geometric and topological characteristics. Conventional algorithms have depended on geometric and radiometric characteristics of point clusters for classification. Topological characteristics have been less used but as shall be explained are potentially more practical than the other two characteristics.

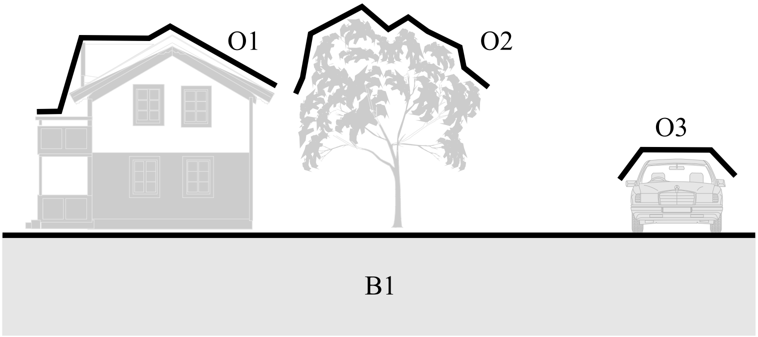


Figure 6.1: Topological ordering of a landscape

Using geometric characteristics

Typical geometric characteristics of a segment include *position/location*, *orientation*, *roughness/texture*, *perimeter length*, *area*, *elongation/eccentricity* and *compactness.* For detecting objects, of the above only the *roughness/texture*, *perimeter length*, *area*, *elongation* and *compactness* are of practical worth because it can be argued that for many objects the value of these characteristics is bounded. For example if the maximum expected building area is know then all segments with an area greater than this maximum cannot be a building. However, choosing an expected maximum value for a characteristic is subjective and can lead to problems in the event of under or over segmentation.

Using radiometric characteristics

Every point in a point cloud is associated with at least one radiometric characteristic, i.e., the reflectance strength of the returned pulse. Some ALS systems also capture an image simultaneous with the pulse measurement, thus allowing an RGB triplet to be associated with each point. Radiometric characteristics for a segment can then be obtained by aggregating the radiometric characteristics of the points in the segments. Typically these are based on statistical measures such as the median, mode, or n^{th} percentiles. Therefore, many radiometric characteristics could be derived for a segment. However, for detecting objects, radiometric characteristics are problematic in that the material coverage of the bare earth is not homogeneous. Objects are limited in extent and are therefore usually covered by one type of material (mineral or vegetable). The bare earth on the other hand occupies a large space and is usually covered by different types of materials. This combined with the fact that the radiometry of objects can differ from landscape to landscape makes this characteristic problematic for object detection. This is essentially a problem of image understanding and requires knowledge modelling that is still a problem far from being solved.

Using topological characteristics

The topology, i.e., the spatial relationship between segments, can be described by the adjacency and connectivity of segments. Topology introduces the notion that there is an order in the construction of a landscape, in both planimetry and height. Objects are objects because they exist above other objects or the bare earth, and the bare earth has nothing beneath it. In figure 6.1 the segments O1, O2, and O3 belong to objects because they are *raised.* Segment B1 belongs to the bare earth because it is *lowered.* Therefore, the shape of the surface segments (as obtained by the profile intersection technique) implicitly contains the topological relationships between segments.

Algorithm for macro object detection

The algorithm for detecting macro objects is based on the shape of surface segments, as explained in section 5.5. To explain how the algorithm works the tiered object in Figure 6.2 shall be used. Before proceeding the reader is encouraged to have a look at the example in appendix D. It will help in understanding the algorithm.

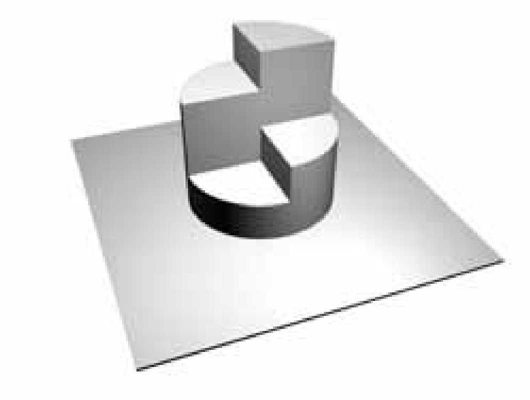


Figure 6.2: Tiered object

Algorithm 6.1: Detecting macro objects

```
Data: Point cloud, $V$
Result: Point cloud, $V$, classified as object or bare earth
begin
    while raised object found do
6.1.1       Segment $V$. $\Theta_{PI,MST} V \Rightarrow S$
        foreach line segment $l$ in the profiles $P$ do
6.1.2           Get the shape of $l$. $\psi(l) \Rightarrow \mu_l$
        end
        foreach $s$ in $S$ do
6.1.3           Get the shape of $s$. $\psi(s) \Rightarrow \mu_s$
            if $\mu_s = Raised$ then
                foreach point $v$ in $s$ do
6.1.4                   Label $v$ as object
                end
            end
        end
6.1.5       Remove from $V$ all $v$ labeled as object.
        $V = V - \{v | v \in V, \quad \phi(v) = object\}$
    end
    foreach point $v$ in $V$ do
        if $\mu_v \neq Raised$ then
6.1.6           Label $v$ as bare earth
        end
    end
end
```

Step 6.1.1: The preferred segmentation procedure is the profile intersection with minimum spanning tree (section 5.3), because it is better able to handle steep slopes in the bare earth; see example 1 in section 5.4. In landscapes where the bare earth is relatively flat the consecutive labeling algorithm will work equally well, section 5.3.

Step 6.1.2: Once segmentation has been achieved, the shape of each line segment is determined as explained in section 5.5. Figure 6.3(a) shows the example point cloud segmented in four directions. The shape of every line segment in the top right surface segment is *raised.* This can be appreciated by looking at the line segments passing through point 2.

Step 6.1.3: Because every line segment is *raised*, the shape of the surface segment can also be said to be *raised.* Therefore, based on the assumption that the perimeter of an object is mainly raised above its neighborhood it is argued that this surface belongs to an object, figure 6.3(a). The determination of the shape of a segment is explained in section 6.2.

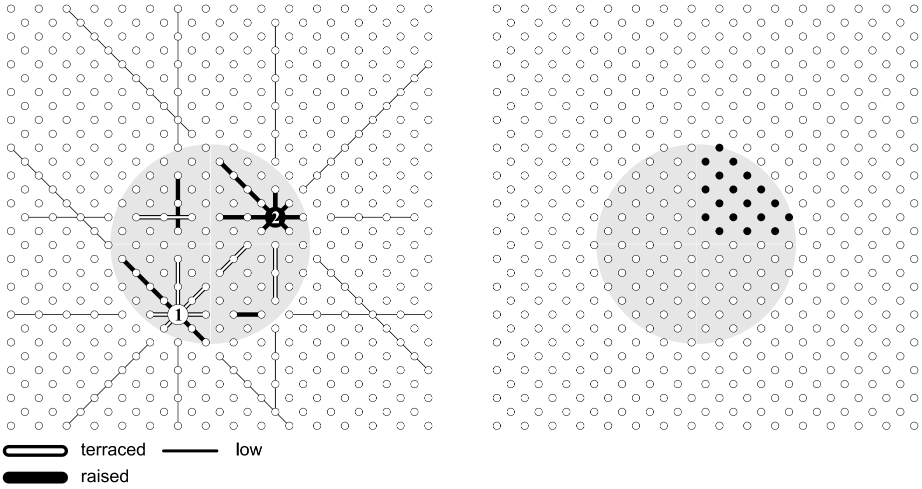
Step 6.1.4: Once a segment is identified as belonging to an object then every point in the segment is classified as object, figure 6.3(b).

Step 6.1.5: Next, detected object points are removed from the point cloud. This can be thought of as a stripping away of objects from the landscape.

The top left, bottom left and bottom right surface segments contain mostly *terraced* line segments. This can be appreciated by looking at the line segments passing through point 1 in figure 6.3(a). Because most line segments are *terraced*, the shape of the surface can also be said to be *terraced*. These surface segments cannot belong to an object because their perimeters are not raised above their neighborhoods. Therefore, the surface segments are kept indeterminate. This is a safe assumption because terraces are found in both objects and the bare earth.

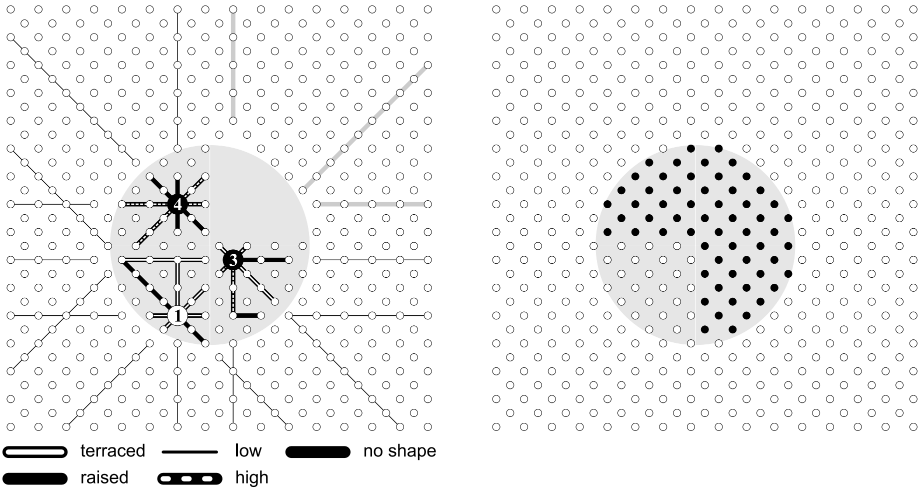
To classify the indeterminate segments, steps 6.1.1 through to 6.1.5 have to be repeated. Again this can be thought of as a gradual stripping away of *objects* in order to get at the *bare earth*.

Therefore, in the first repetition the top left and bottom right surface segments are



(a) (b)

Figure 6.3: Macro object detection.



(a) (b)

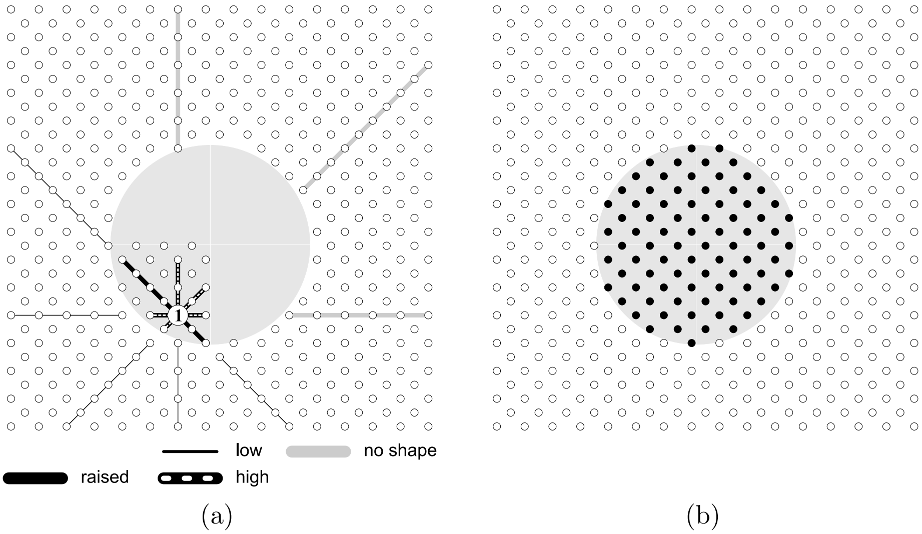
Figure 6.4: Macro object extraction. Iteration 1

predominantly composed of *raised* and *high* line segments. This can be appreciated by looking at the line segments passing through points 3 and 4, figure 6.4(a). The top left and bottom right surface segments are shaped as *raised* and *high* respectively. *Raised* and *high* segments belong to objects on the assumption that the perimeter of objects are raised above their neighborhoods. Therefore, the surface segments and the points within surface segments are classified as *object*, figure 6.4(b).

The shape of the bottom left segment remains unchanged as can be seen by comparing the line segments passing through point 1 in figure 6.3(a) with those in figure 6.4(a).

In the second repetition the bottom left surface segment is classified as object because the line segments passing through it are predominantly *high*; figure 6.5(a) and (b). Usually three or four repetitions are sufficient.

Step 6.1.6: In a final step all points that have not been labeled as *object* are labeled as *bare earth*. This labeling is done to indicate that the points are potentially bare earth.



(a) (b)

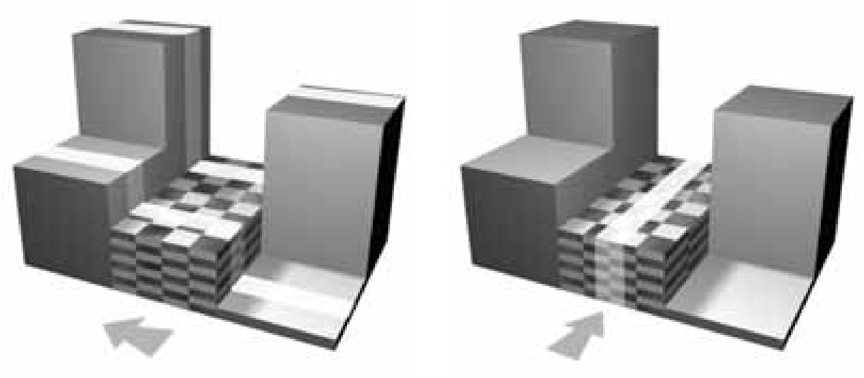
Figure 6.5: Macro object extraction. Iteration 2

Determining the shape of a surface segment

As stated in section 5.5 a surface can assume six different shapes, *No shape*, *Raised*, *Lowered*, *High*, *Low* and *Terraced*. In the discussions above the shape of a surface segment has been decided by finding the line segment shape that is dominant within the segment. In other words the shape of a surface segment is determined by the line segment shape, μ, with the highest frequency. The shape grade, $f(\psi(s) = \mu)$ (value between 0 and 1), or the indicator that a surface segment is of shape μ, is given by:

$$f(\psi(s) = \mu) = \frac{|M_{s,\mu}|}{|M_s|} \tag{6.1}$$

Where $\psi(s)$ is the shape of surface segment s, $|M_s|$ is the cardinality of the set of line segment shapes in surface segment s, and $|M_{s,\mu}|$ is the cardinality of the set of line segments with shape μ in surface segment s. A necessary property of the shape grade is:



In this direction the surface is partly terraced and partly lowered

In this direction the surface is wholly no shape

Figure 6.6: A surface as a combination of different shapes. The checkered surface is lowered, terraced and no shape to varying degrees.

$$\sum_{\mu=\substack{raised,\\ lowered,\\ high,\\ low,\\ terraced,\\ noshape}} f(\psi(s)=\mu)=1 \tag{6.2}$$

Equation 6.2 can be understood to mean that each shape grade serves as evidence of a surface segments' affinity for a particular shape. Because of this, a surface segment is treated as having not one, but six shapes to varying extents. Figure 6.6 shows a surface that is a combination of different shapes.

Equation 6.1 assumes that all line segments share the same orientation. Figure 6.7(a) and (b) show a *terraced* surface segment that has been profiled in two and three directions respectively. The shape grade for the surface segment being raised is 1/6 and 1/11 for (a) and (b) respectively. This shape grade rapidly diminishes as more profiles are added. Looking at figure 6.7(b), in spite of the dominance of the terraced line segments the surface still appears to be moderately raised. This is because the single raised line segment carries all the shape information along its orientation. It is therefore argued that $f(\psi(s)=\mu)$ is biased by the number of profile orientations. To compensate for this bias the shape grade in equation 6.1 is redefined so that the frequency along each orientation is weighted by the reciprocal of the number of profile orientations, $|D|$ (equation 5.2). In other words, the shape of a surface segment is determined by aggregating the surface shape along each orientation. The shape grade is now given by:

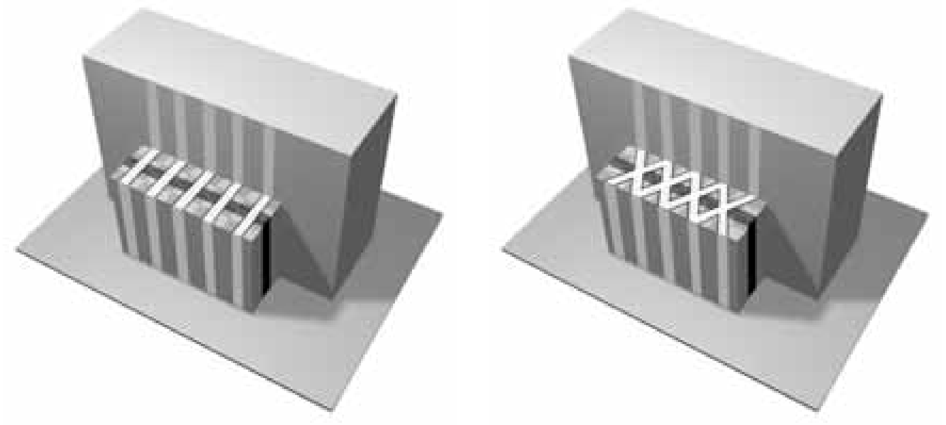

(a) (b)

Figure 6.7: Shape from orientation. Two profile directions (a), three profile directions (b)

$$f(\psi(s) = \mu) = \sum_{i=1}^{|D|} \frac{|M_{s,\mu,\varphi_i}|}{|M_{s,\varphi_i}|} \frac{1}{|D|} \tag{6.3}$$

where $|M_{s,\varphi_i}|$ is the cardinality of the set of line segment shapes with orientation φ_i in surface segment s, and $|M_{s,\mu,\varphi_i}|$ is the cardinality of the set of line segments with shape μ and orientation φ_i in surface segment s.

In figure 6.7 the surface segment's raised grade is now $\frac{1}{2}\frac{1}{1} + \frac{1}{2}\frac{0}{5} = \frac{1}{2}$ and $\frac{1}{3}\frac{1}{1} + \frac{1}{3}\frac{0}{5} + \frac{1}{3}\frac{0}{5} = \frac{1}{3}$ for (a) and (b) respectively. The shape grades obtained from this expression do not diminish as rapidly as those from equation 6.1. More importantly by aggregating the shape grades along each orientation the bias of the segmentation procedure is largely removed from the shape determination.

Having determined the shape grade for a surface segment the next step is to classify the surface segment as object or bare earth based on the shape grade, i.e., the shape of the surface segment. The concept behind the classification proceeds as follows. Each shape is associated with the class (ϕ) bare earth, object or both. The degree of association, $\beta_{\phi,\mu}$ (value between 0 and 1), is chosen by a visual examination of the character of objects and the bare earth in a landscape. For example raised surfaces are mainly found in objects and as such $\beta_{object,raised}$ is set equal to 1 and $\beta_{bareearth,raised}$ is set equal to 0. Similarly lowered surfaces are mainly found in the bare earth and as such $\beta_{object,lowered}$ is set equal to 0 and

μ	$\beta_{object,\mu}$	$\beta_{bareearth,\mu}$	Reason
raised	1	0	objects are typically raised
lowered	0	1	the bare earth is lowered
high	1	0	objects have high edges
low	0	1	the bare earth has low edges
terrace	0.5	0.5	terraces are found in both the bare earth and objects
no shape	0	1	objects are typically surrounded by bare earth or other objects so objects cannot be no shape

Table 6.1: Example of a set of associations.

$\beta_{bareearth,lowered}$ is set equal to 1. A sample of associations is given in table 6.1. The collection of associations $\beta_{\phi,\mu}$ over all the classes and shapes can be thought as an individual's topological characterization of a landscape.

Using the shape grades and the associations a class grade, $g(\phi(s) = object)$ (value between 0 and 1), is determined for each surface segment, given the class ϕ (object or bare earth). The class grade g is given by:

$$g(\phi(s) = \phi) = \sum_{\substack{\mu = raised,\\ lowered,\\ high,\\ low,\\ terraced,\\ noshape}} \beta_{\phi,\mu} f(\psi(s) = \mu) \tag{6.4}$$

This is essentially a weighted mean of the associations where the weights are given by the corresponding shape grades. During classification a surface segment is deemed to be an object if its class grade, exceeds the threshold, ϵ_{object}, set equal to or greater than 0.5.

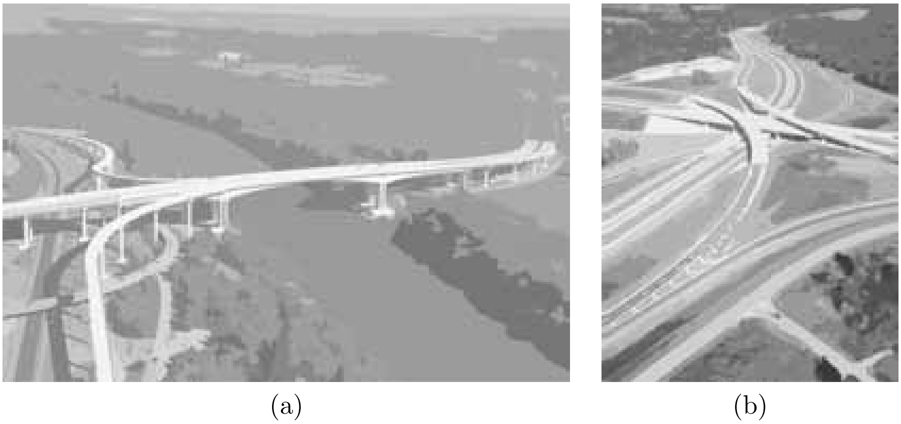

(a) (b)

Figure 6.8: Conventional bridges, interchanges and flyovers, (a) a conventional bridge with interchange (b) an interchange and flyover

6.3 Detecting bridges

By definition, a bridge is a structure that spans the gap between two land masses. This definition will be understood here to include structures like conventional bridges, interchanges, flyovers and ramps. Literature on the detection of bridges is thin, and the work done is based on data obtained by satellite imagery and radar. Some approaches to bridge detection are now briefly discussed.

Wang and Zheng (1998) propose a technique for detecting bridges in Synthetic Aperture Radar (SAR) imagery. Based on the principle of SAR (Rodriguez and Martin, 1992) a range image is obtained for a landscape. The authors argue that because of the speckle in SAR images they avoid edge detection and object recognition based on contours. Instead, they favour an approach based on segmentation and Hough transformations to detect the bridges. The approach is heavily based on the measurement characteristics of SAR, and shall not be elaborated further because it is beyond the scope of the discussion. However, the principle on which bridges are detected is noteworthy in that it assumes that every bridge is covered by the same material as roads and every bridge has a parapet. The segmentation step identifies all roads and by association all bridges. The authors note that because the backscattering from parapets is strong, this allows them to isolate these features. Once the parapet points are identified, a Hough transformation is used to identify parallel parapets. The points between parallel parapets are then treated as being part of a bridge. Houzelle and Giraudon (1992) also detect bridges in SAR data but first reduce their search space by identifying water bodies using SPOT imagery.

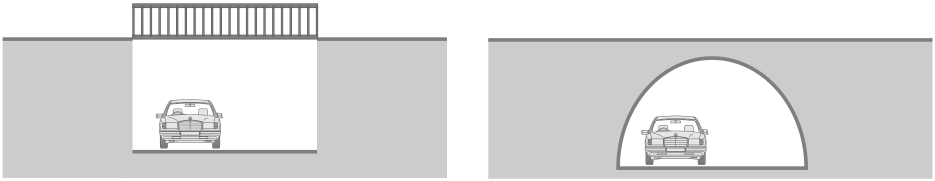

(a) (b)

Figure 6.9: Bridges (a) and tunnels (b).

Trias-Sanz and Lomenie (2003) propose an approach for detecting bridges in high resolution satellite imagery. The procedure starts by segmenting an image and then classifying the generated segments using a neural network. Next, rules are defined to classify segments as bridge. Some example rules are:

1. large regions of *water* or *railway yard* are separated by a narrow and long strip. This strip is a bridge.
2. a small gap between two regions that have been identified as road and are aligned is a bridge. The definition of small is user defined.
3. a small gap between two regions that have been identified as canal and are aligned is a bridge. The definition of small is user defined.

Lomenie et al. (2003) augment the above approach with a geometric technique based on edge detection. The method makes use of high resolution imagery. First, an edge detector is applied to an image. Next, edges smaller than a user set value are removed. The remaining edges are tested to find aligned and parallel edges separated by a given maximum distance. Once these parallel edges are found the region between them is treated as being potentially bare earth. Using the rules employed by Trias-Sanz and Lomenie (2003) these bare earth regions are tested to determine if they are indeed bridges. Another example of bridge detection using classified regions in satellite imagery is provided by Ritter et al. (1986).

Although novel, all these algorithms are designed to detect simple bridges and flyovers (i.e., straight parallel sided tracks). They would have problems with the bridges shown in figure 6.8. Figure 6.8(a) shows a bridge that branches into three and figure 6.8(b) shows a flyover running under another flyover. A geometric method is proposed here that detects both simple and complex bridges.

It is argued that the means to detect a bridge can be derived from the purpose a bridge serves in a landscape. A bridge is designed to span bare earth segments. Hence, the assumption that a bridge is connected to the bare earth on at least two

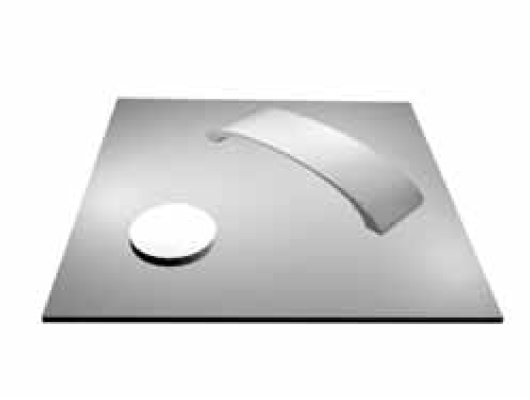


Figure 6.10: Bridge and small object

sides. To span land masses a bridge will necessarily have be raised above the bare earth (otherwise it would not be a bridge). Hence the assumption that along the length of a bridge, diametrically opposite points on its perimeter are raised above the bare earth (except where it connects to the bare earth). Additionally, because of economic constraints a bridge typically has to be longer than it is wider. Hence the assumption that a bridge is greater in length than in width.

Before proceeding to explain how bridges are detected based on the above assumptions, for the sake of completeness it is necessary to briefly distinguish between bridges and tunnels. Topologically a bridge and a tunnel are not dissimilar, as shown in figure 6.9. However, materially they differ in that the ceiling of tunnels is part of the bare earth. Because of their design purpose, most tunnels are typically long in length and narrow in width. In this respect they are different from bridges which as already stated are assumed to be greater in length than in width.

Algorithm for bridge detection

The algorithm for detecting macro objects is based on the shape of line segments. To explain how the algorithm works the bridge and small object in figure 6.10 shall be used. Before the actual detection can begin, the point cloud has to be preprocessed. In this preprocessing additional geometric information is obtained for each point. Before proceeding the reader is encouraged to have a look at the example in appendix D. It will help in understanding the algorithm.

Algorithm 6.2: Detecting bridges: Pre-processing

```
Data: Point cloud, $V$
Result: Bare earth point cloud, $V_{BE}$, smoothness and discontinuity
        measure $\forall v \in V_{BE}$
begin
6.2.1   Remove from $V$ all $v$ labeled as object.
        $V_{BE} = V$ - $\{v | v \in V, \quad \phi(v) = object\}$
        foreach point $v$ in $V_{BE}$ do
6.2.2       Fit a plane to $K$ nearest planimetric neighbours of $v$
            Determine the standard deviation of the residuals
            Keep this standard deviation as a property (smoothness) of $v$
        end
        foreach point $v$ in $V_{BE}$ do
6.2.3       Determine largest height difference to $K$ nearest planimetric
            neighbors of $v$
            Keep this height difference as a property (discontinuity) of $v$
        end
end
```

Step 6.2.1: Bridges are detected in the bare earth. Therefore, before a bridge can be detected all macro objects have to be removed from the point cloud.

Step 6.2.2: Part of the bridge detection requires the determination of points where a bridge makes a smooth transition into the bare earth, i.e., where a bridge connects to the bare earth. This step aims to estimate the smoothness of the bare earth at a point. The smoothness is estimated by fitting planes at a point. The smaller the standard deviation of the residuals in the fit the smoother the surface is deemed to be.

Step 6.2.3: Part of the bridge detection also requires the determination of points where a bridge is discontinuous to the bare earth, i.e., the edges of a bridge. Importantly the size of the discontinuity has to be known. The discontinuity at a point is estimated by determining the largest absolute height difference to its neighborhood. In figure 6.11(a) points on the edge of the bridge and the small object show strong discontinuity.

The next step is to identify points that are likely from a bridge structure. These points will serve as seeds in the detection of bridge structures.

Algorithm 6.3: Detecting bridges: Identifying possible bridge points

```
Data: Bare earth point cloud, $V_{BE}$
Result: Points that are likely off bridges, $V_{Raised}$
begin
6.3.1   Segment $V_{BE}$. $\Theta_{PI,Consecutive-Slope} V_{BE} \Rightarrow S$
        foreach point $v$ in $V_{BE}$ do
6.3.2       Get line segments, $L_v$ passing through $v$
            foreach line segment $l_v$ in $L_v$ do
6.3.3           Get the shape of $l_v$. $\psi(l_v)$
                Count the shape
            end
6.3.4       Label $v$ using the shape with the largest count
6.3.5       Remove from $V_{BE}$ all $v$ not labeled as Raised.
            $V_{Raised} = V_{BE} - \{v | v \in V_{BE}, \quad \phi(v) \neq Raised\}$
        end
end
```

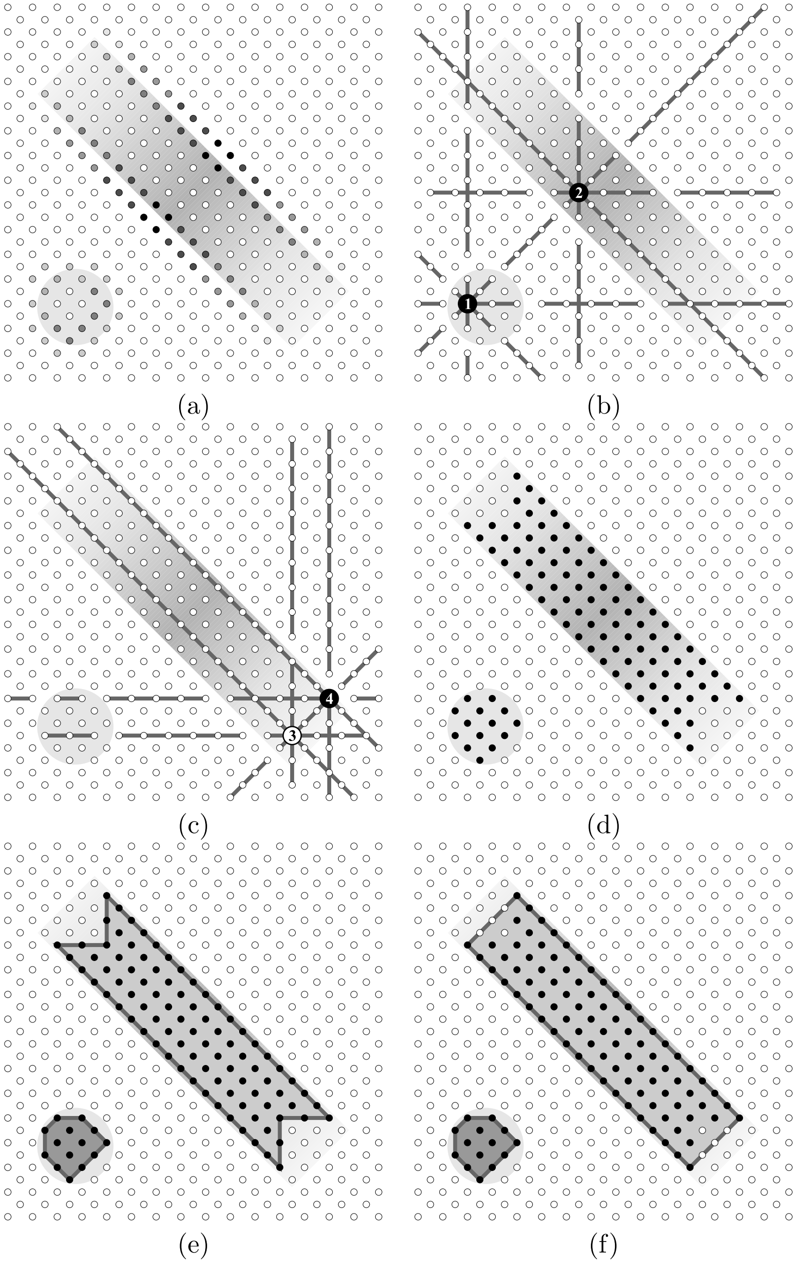
Step 6.3.1: The shape of line segments along a cross section of the bare earth has to be known. Because diametrically opposite points on the perimeter of a bridge are raised, it is therefore likely that a *raised* line segment is from a bridge. Therefore, the point cloud is segmented by profile intersection using consecutive labeling with slope. The rationale for using this method of segmentation has been stated in section 5.4 example 2.

Step 6.3.2: The line segments passing through a point are identified.

Step 6.3.3: The shape of the line segments passing through every point is determined.

Step 6.3.4: A point is labeled using the shape with the highest count. Points on a bridge should have many *raised* line segments passing through them. In figure 6.11(b) and (c) the points marked v1, v2 and v4 are labeled as *raised* (large black circle) because most of the line segments passing through them are *raised.* Point v3 is labeled as *high* (large white circle) because most of the line segments passing through it are *high.* In this manner all points in the cloud are labeled as shown in figure 6.11(d).

Step 6.3.5: The next step of the algorithm is to determine the boundaries of bridges. These boundaries are determined using the *raised* points. For this reason *raised* points are separated from the *bare earth* point cloud.



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

Figure 6.11: Bridge detection.

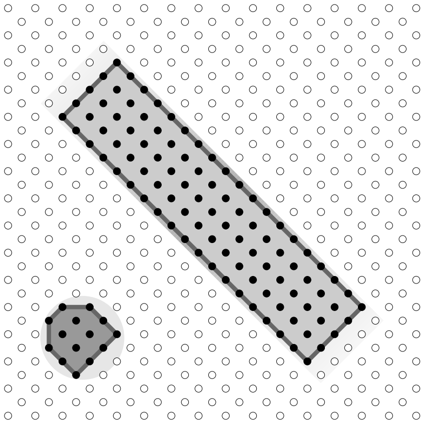


Figure 6.12: Bridge detected.

In the final part of the bridge detection algorithm the boundaries of bridges are determined and the points within the boundaries are classified as bridge.

Step 6.4.1: To determine the closed boundaries (figure 6.11(e)) the *raised* points are first triangulated using Delaunay triangulation. Next, the triangulated points are segmented by proximity (see section 5.2). Each segment represents a potential bridge.

Step 6.4.2: Next, the points within each segment are triangulated using Delaunay triangulation. A 2D convex hull is determined for each segment. All long edges are removed from the triangulation. A boundary is then traced along the outside edge of the triangulation.

Step 6.4.3: Once the closed boundary around a segment has been obtained, the next step is to determine likely boundary points where the bridge connects to the bare earth. Those parts of a bridge that connect to the *bare earth* are called *Across edges* and those that don't are called *Along edges*, because they extend across and along the length of a bridge respectively. To find *Across* and *Along* edges the boundary is first broken up by connecting adjacent points (on the boundary) if both their smoothness is greater than or smaller than a chosen threshold. This partitioning of the boundary yields a series of edges.

Step 6.4.4: In *across* edges points have smoothness values smaller than or equal to the smoothness threshold. Furthermore, the length of an *across* edge is greater or equal to the expected minimum bridge width.

Step 6.4.5: In *along* edges points have smoothness values greater than the smoothness threshold. Furthermore, in an *along* edge there should be at least one point whose discontinuity exceeds the minimum bridge height. A minimum bridge height is defined to avoid incorrectly detecting micro objects as bridges.

Step 6.4.6: A bridge must connect to the bare earth at least twice. Therefore, a bridge must have at least two *Across* edges and two *Along* edges. Those segments that meet this criteria are treated as being *bridge* segments.

Step 6.4.7: In figure 6.11(e) the *across* edge causes the extent of the bridge to be underestimated. This is a side effect of the bridge seed point detection (algorithm 6.3). Therefore, from each *bridge* boundary the points from *across* edges are removed. After this removal points from the along edges are merged into a single closed boundary, figure 6.11(f).

Step 6.4.8: Next all points inside the planimetric *bridge* boundary are collected.

Step 6.4.9: Segment by proximity all points that fall within a planimetric *bridge* boundary.

Step 6.4.10: If a segment touches a *bridge* boundary then the points within the segment are labeled as *bridge*, figure 6.12. This step is necessary to avoid *bare earth* points being incorrectly labeled as *bridge*, because In cases where bridges split into two or more parts, step 6.4.8 can result in the bare earth being included in the bridge boundary.

This procedure takes advantage of the information contained in the line segments generated by the profile intersection technique. Furthermore, as opposed to other algorithms a bridge need not be of uniform width and additionally a bridge can split along its length.

If a bridge is wider than it is long, then most of the line segments passing through a point on the bridge will not be *raised*, and the bridge will not be detected. For the same reason tunnels will not be detected. Therefore, this aspect of the algorithm naturally accommodates the assumption that a bridge is longer than it is wide.

Where the algorithm will face difficulties is when a bridge crosses a deep water body. In ALS data deep water bodies are not represented, because laser pulses are absorbed rather than reflected. Therefore, in an ALS point cloud where there are deep water bodies there are also data gaps. This makes it difficult or near impossible to establish if a bridge is raised along its length. One possibility of getting around this problem is to alter step 6.3.5, and search for *no shape* line segments instead of *raised* line segments.

Algorithm 6.4: Detecting bridges: Final identification

```
Data: $V_{BE}$, $V_{Raised}$, Smoothness threshold, Minimum bridge width,
      Minimum bridge depth
Result: Point cloud, $V$, classified as objects, bridge or bare earth
begin
6.4.1    Segment $V_{Raised}$. $\Theta_{Proximity} V_{Raised} \Rightarrow S_{Prox}$
         foreach $s$ in $S_{Prox}$ do
6.4.2        Determine the closed boundary, $c$, around each segment, $s$
6.4.3        Segment, $c$, based on the smoothness threshold
             foreach boundary segment, $c_i$ in $c$ do
                 if $\exists v \in c_i$ with smoothness $\leq$ Smoothness threshold then
6.4.4                if length of $c_i >$ minimum bridge width then
                         Label $c_i$ as across edge
                         Count across edge
                     end
                 else
6.4.5                if $\exists v \in c_i$ with discontinuity $\geq$ minimum bridge height then
                         Label $c_i$ as along edge
                         Count along edge
                     end
                 end
6.4.6            if $|acrossedges| \geq 2$ AND $|alongedges| \geq 2$ then
6.4.7                Remove from $c$ all $c_i$ labeled as across edge
                     $c = c - \{c_i | c_i \in c, \quad \phi(c_i) = acrossedge\}$
6.4.8                $\forall v in V_{BE}$, get all points, $v$ inside planimetric projection of $c$
                     $\Rightarrow V_c$
6.4.9                Segment $V_c$. $\Theta_{Proximity} V_c \Rightarrow S_c$
6.4.10               foreach $s$ in $S_c$ do
                         if $s \cap c \neq \emptyset$ then
                             foreach point $v$ in $s$ do
6.4.11                           Label $v$ as bridge
                             end
                         end
                     end
                 end
             end
         end
end
```

6.4 Detecting micro objects

Micro objects reveal themselves as low amplitude high frequency variations in the surface of the *bare earth*, or in other words roughness. Because roughness is a localized phenomenon, local surface fits are applied to search for points that contribute to the roughness of a surface.

Algorithm for detecting micro objects

The fittings are used to approximate a smooth surface and the height variations on the surface approximate the roughness. Typically, a plane is used. The algorithm proceeds as below.

Step 6.5.1: *Micro objects* are detected in the *bare earth*, therefore *object* points are first removed from the point cloud.

Step 6.5.2: *Bridges* are also removed from the point cloud.

Step 6.5.3: At every point a plane is fit to the K nearest neighbors, and the standard deviation of the residuals is computed.

Algorithm 6.5: Detecting micro objects

```
Data: Point cloud, $V$, Smoothness threshold
Result: Point cloud, $V$, classified as object, bridge or bare earth
begin
6.5.1   Remove from $V$ all $v$ labeled as object
        $V_{BE} = V$ - $\{v|v \in V, \quad \phi(v) = object\}$
6.5.2   Remove from $V_{BE}$ all $v$ labeled as bridge
        $V_{BE} = V_{BE}$ - $\{v|v \in V_{BE}, \quad \phi(v) = bridge\}$
        foreach point $v$ in $V_{BE}$ do
6.5.3       Fit a plane to $K$ nearest neighbours of $v$
            Determine, $\sigma$, the standard deviation of the residuals
            if $\sigma$   Smoothness threshold then
                if $v$ is above the fitted plane then
6.5.4               Label $v$ as object
                end
            end
        end
end
```

Step 6.5.4: If the standard deviation of the residuals at a point is greater than the smoothness threshold and the point is above the fitted plane, then the point is deemed to be a *micro object*. In other words, if the surface about a point is rough and the point is above the surface then the point must be contributing to the roughness and hence must be an object.

To avoid mis-detections the point neighborhoods have to be kept relatively small, particularly in low resolution point clouds. This is one reason why planes and not polynomials of degree two or higher are used. Nonetheless, pavement edges maybe lost but this is considered an acceptable trade off. The choice of smoothness threshold depends on two factors, (a) the accuracy of the points and (b) the resolution of the point cloud. It should be greater than the accuracy of the point cloud and smaller than the resolution of the point cloud. How much it should be greater or smaller than these quantities requires further study. In tests, the smoothness threshold was chosen on intuition.

6.5 Detection of man made and natural objects

The points that have been classified as objects (sections 6.2 and 6.3) are next to be classified as man made or natural objects. So far discrimination between segments has been done solely based on topology. For the discrimination of man made and natural objects this is not suitable because the topological relation between these objects is indeterminate. Here geometric and radiometric characteristics are much more favourable.

Algorithm for detecting man made and natural objects

To make the detection problem more specific man made objects shall be limited to mean buildings and natural objects every other object in the landscape. Because of this separation, small objects like motor vehicles, trains, street lights and other such features shall be treated as natural objects. This is not a satisfactory separation, but it is convenient since typically after the bare earth, buildings are the next feature sought in a landscape.

The algorithm for detecting natural and man made objects is divided into three parts. The first part of the algorithm seeks vegetation islands in the bare earth.

Step 6.6.1: A small cluster of *object* points in the middle of a sea of *bare earth* points is assumed to belong to vegetation on the rational that small objects cannot be buildings. Examples of such islands are hedges and bushes[1]. To detect these islands the point cloud is first triangulated using Delaunay triangulation. Next, the triangulation is segmented by using the labels of the points . Points on and edge are placed in the same segment if they share the same label.

Step 6.6.2: If the number of points in a segment is less than or equal to the maximum island size, then the points in the segment are labeled as *vegetation.* The maximum island size is chosen taking into account the resolution of the point cloud and the expected size of the largest island object in the landscape.

Algorithm 6.6: Detecting man made and natural objects: Vegetation Islands in a sea of bare earth

```
Data: Point cloud, $V$, max island size
Result: Point cloud, $V$, points labeled as bare earth,
        bridge, object, vegetation

        begin
6.6.1       Segment $V$. $\Theta_{Label} V \Rightarrow S$
            foreach $s$ in $S$ do
                if $|s| \leq$ maximum island size AND Label of $s$, $\mu_s =$ object then
                    foreach point $v$ in $s$ do
6.6.2                   Label $v$ as vegetation
                    end
                end
            end
        end
```

The second part of the algorithm classifies the buildings and larger vegetation.

Step 6.7.1: The algorithm begins by computing characteristics for points in the cloud. A plane is fit to the K nearest neighbors of a point. The standard deviation of the residuals to the plane estimates the roughness of the surface at a point. For convenience, the roughness is bounded and scaled. This is necessary for comparisons with other characteristics such as the RGB triplets.

Step 6.7.2: To classify objects all *bare earth* and *bridge* points are removed from the cloud.

---

[1] While objects like cars and lamp posts are man made they are included in the vegetation class. This contradiction is tolerated because the correct detection of buildings is given higher priority.

Algorithm 6.7: Detecting man made and natural objects: Buildings and Vegetation

```
Data: Point cloud, $V$, minimum object size, training data
Result: Point cloud, $V$, points labeled as bare earth, bridge, building,
        vegetation
begin
6.7.1   Fit plane to $K$ nearest neighbors of $v$
        Compute the standard deviation of the residuals
        Scale the standard deviation of the residuals
6.7.2   Remove from $V$ all $v$ labeled as bare earth, bridge or vegetation
        $V_{object} = V - \{v|v \in V, \quad \phi(v) = bareearth \text{ OR } \quad \phi(v) = bridge \text{ OR } \phi(v) = vegetation\}$
6.7.3   Segment $V_{object}$ by profile intersection, $\Theta_{PI,Proximity}V \Rightarrow S$
6.7.4   Select $n$ features to use in the classification
        foreach $s$ in $S$ do
6.7.5       Compute the median of the roughness of all $v \in S$
6.7.6       Compute the median of the reflectance of all $v \in S$
6.7.7       Compute the median of the RGB of all $v \in S$
            if $|s| \geq$ min segment size then
6.7.8           Classify by KNN in the $n$-dimensional feature space $\Rightarrow$ class label
                foreach point $v$ in $s$ do
                    Label $v$ as class label
                end
            end
            foreach point $v$ in $s$ do
6.7.9           Label $v$ as none
            end
        end
end
```

Step 6.7.3: The object points are segmented by profile intersection with proximity. Segmentation by proximity is chosen on the strength that building points are closer to each other than they are to vegetation points and that vegetation points are closer to each other than they are to building points.

Step 6.7.4: The features to be used in the classification are selected. The features used are *roughness*, *reflectance*(if available) and *RGB triplet* (if available).

Step 6.7.5: For each segment, the median of the roughness values for all points is computed.

Step 6.7.6: If *reflectance* values are available, then for each segment the median of the reflectance values for all points is computed.

Step 6.7.7: If *RGB triplet* values are available, then for each segment the median of each RGB value for all points is computed.

Step 6.7.8: An n-dimensional feature space is defined by training data (the determination of the training data is described at the end of this section). If the number points in a segment is greater or equal to the minimum object size, then the characteristics of a segment are mapped into the n-dimensional feature space. The K nearest neighbors of the segment in the feature space are determined. All points in the segment are labeled using the feature class with the highest count.

Step 6.7.9: If the number of points in a segment is less than the minimum object size, then all points in the segment are labeled as *none*.

Algorithm 6.8: Detecting man made and natural objects: Walls

```
Data: Point cloud, $V$, wall threshold
Result: Point cloud, $V$, points labeled as bare earth, bridge, building,
        vegetation
begin
6.8.1   Remove from $V$ all $v$ labeled as bare earth OR bridge $V_{object} = V$ -
        $\{v|v \in V, \quad \phi(v) = bareearth$ OR $\phi(v) = bridge\}$
        foreach point $v$ in $V_{object}$ do
6.8.2       Determine $K$ nearest planimetric neighbors of $v$
            Determine PCA for the neighborhood
            Project the neighbors of $v$ onto the eigenvector with the smallest
            eigenvalue
            Compute, $\sigma_{wall}$, the standard deviation of the projections along this
            eigenvector
            if $\sigma_{wall} \leq$ wall threshold then
6.8.3           Label $v$ as none
            end
        end
        foreach point $v$ in $V_{object}$ do
            if Label of $v$ = none then
6.8.4           Classify using $K$ nearest neighbors of $v \Rightarrow$ class label
                Label $v$ as class label
            end
        end
end
```

Tests conducted using the above algorithm showed that the largest classification errors arose from misclassification of wall points. The reason for this is the strong roughness values for wall points. To overcome this problem walls need to be classified using the labels of roof lines. The last part of the algorithm classifies wall points and points that were classified as *none* in the previous steps.

Step 6.8.1: All *bare earth* and *bridge* points are removed from the point cloud.

Step 6.8.2: At each point a plane is fit to the K nearest planimetric neighbors of v The principal component of the neighborhood is determined. The neighborhood is transformed onto the principal components and the standard deviation of the coordinate along the smallest component is determined.

Step 6.8.3: If the standard deviation is less than or equal to the wall threshold then the point is labeled as *none*. This is a preparation for the next step.

Step 6.8.4: If in the cloud a point has the label *none*, then its K nearest neighbors are determined. In the neighborhood the label with the highest count is determined, and the point is assigned this label. In this manner, points below roof edges will be detected as building.

Training the classifier

In step 6.7.8 of the algorithm training data is used to perform the classification. This training data is obtained in the following manner:

1. A representative subset of the point cloud is manually classified. To speed up the classification the macro object, bridge and micro object detection algorithms can be run to identify objects. From here, the objects can be separated into buildings and vegetation.
2. Next, the subset is segmented as in step 6.7.3. For each segment the same characteristics in steps 6.7.5, 6.7.6 and 6.7.7 are determined.
3. Finally a majority classification using the labels from the manual classification is used to label each segments.
4. Representative segments and their n characteristics are then chosen to build the n-dimensional feature space.

The manual classification need only be coarse and if done correctly should take about one hour for a cloud of 300000 points.

6.6 Discussion

It was argued that macro objects should be detected first because they are predominant in the landscape. The micro object detection is a more expensive algorithm, hence this rationale is valid. In the performance tests (chapter 7) the macro objects were detected first to avoid the overheads of micro object detection.

However, there is an argument for performing micro object detection before macro object detection. Doing the micro object detection first may remove vegetation next to buildings and hence reduce the possibility of buildings being merged into bare earth segments in macro object detection. However, in practice this does not work because vegetation that is removed leaves behind holes in the data that complicate the macro object detection.

A better approach to avoid the merging of objects into bare earth segments is to re-segment the segments obtained in the macro object detection. In other words, during the macro object detection individual segments can be further segmented to improve the separation between bare earth and objects.

Chapter 7

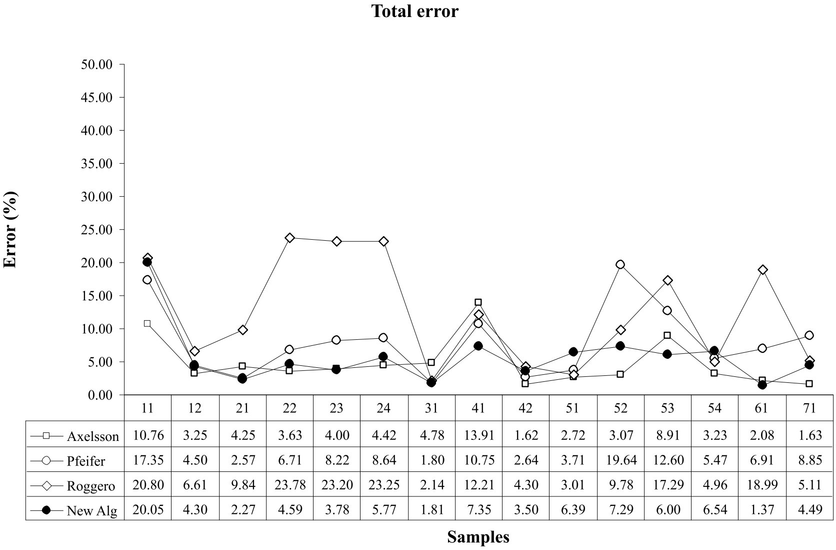
Results and Quality Analysis

7.1 Introduction

The proposed filtering algorithm was tested on real data. This chapter examines and discusses the results of the tests. Three different data sets were used in the test.

1. The first set is the ISPRS data set described in chapter 3.
2. The second set comes from the city of Wijhe in the east of the Netherlands. The data was acquired by Fugro-Inpark using the helicopter mounted FLI-MAP system. The point spacing of the data is about 0.5m and associated with each point is a positive integer representing the strength of the returned pulse, i.e., the reflectance strength.
3. The third data set comes from the city of Nijmegen (Netherlands). The data was acquired by TerraImaging using the airplane mounted ALTM system by Optech. The point spacing of the data is approximately 1.0m, and associated with each point is an RGB triplet. RGB triplets for the points are extracted from images taken during scanning.

For purposes of comparison the last two data sets were manually classified into three classes, *bare earth*, *building* and *vegetation*. The ISPRS data sets were also manually classified but only into the classes *bare earth* and *object*.



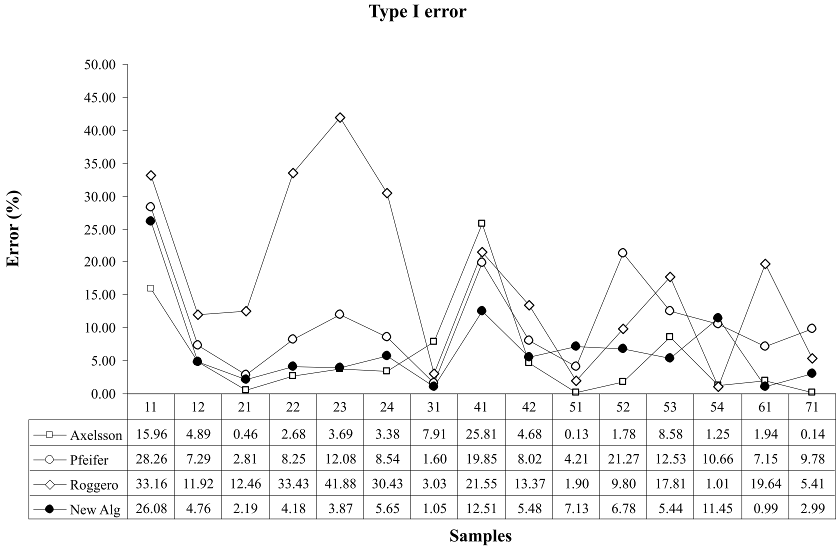
	11	12	21	22	23	24	31	41	42	51	52	53	54	61	71
—□— Axelsson	10.76	3.25	4.25	3.63	4.00	4.42	4.78	13.91	1.62	2.72	3.07	8.91	3.23	2.08	1.63
—○— Pfeifer	17.35	4.50	2.57	6.71	8.22	8.64	1.80	10.75	2.64	3.71	19.64	12.60	5.47	6.91	8.85
—◇— Roggero	20.80	6.61	9.84	23.78	23.20	23.25	2.14	12.21	4.30	3.01	9.78	17.29	4.96	18.99	5.11
—●— New Alg	20.05	4.30	2.27	4.59	3.78	5.77	1.81	7.35	3.50	6.39	7.29	6.00	6.54	1.37	4.49

Figure 7.1: Total errors(%) over the 15 samples

7.2 Assessment against ISPRS filter test data

This section presents the results of the tests of the new algorithm against the ISPRS data set, with a view to determining if the problems noted in chapter 3 have been solved. The results from the tests and the full comparisons against the other tested algorithms are presented in appendix C. The results from the developed algorithm are compared against two of the better performing algorithms (Axelsson and Pfeifer) in the ISPRS test. The algorithm is also compared against another one of the algorithms that showed average performance (Roggero).

The total errors committed is shown in figure 7.1. The total error presents the number of misclassified points in a sample as a percentage of all points in the sample. Overall, the new algorithm does as well or better than most of the algorithms tested. Importantly because of the multi-step classification approach (macro and micro object detection) the algorithm is able to control the reduction of both type I and type II errors. The Axelsson and Pfeifer algorithms also use a multi-step approach. In their iterative approach, they start by seeking large objects and then with each iteration they seek ever smaller objects in a landscape. The significant difference in the multi-step concept of the developed algorithm and that in the algorithms of Axelsson and Pfeifer is that the developed algorithm uses different algorithms to detect macro and micro objects. The algorithms of Axelsson and



	11	12	21	22	23	24	31	41	42	51	52	53	54	61	71
—□— Axelsson	15.96	4.89	0.46	2.68	3.69	3.38	7.91	25.81	4.68	0.13	1.78	8.58	1.25	1.94	0.14
—○— Pfeifer	28.26	7.29	2.81	8.25	12.08	8.54	1.60	19.85	8.02	4.21	21.27	12.53	10.66	7.15	9.78
—◇— Roggero	33.16	11.92	12.46	33.43	41.88	30.43	3.03	21.55	13.37	1.90	9.80	17.81	1.01	19.64	5.41
—●— New Alg	26.08	4.76	2.19	4.18	3.87	5.65	1.05	12.51	5.48	7.13	6.78	5.44	11.45	0.99	2.99

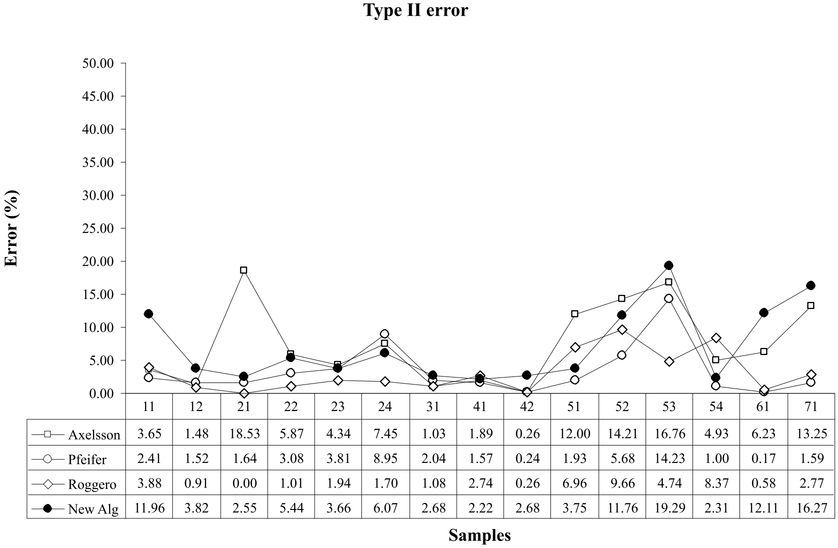
Figure 7.2: Type I errors over the 15 samples

Pfeifer apply the same algorithm at each iteration but at a different scale. Notwithstanding the differences in their multi-step approaches the good performance of the new algorithm and the algorithms of Axelsson and Pfeifer reinforce the belief that iterative approaches do better than non iterative approaches.

The type I errors committed are shown in figure 7.2. The type I error presents the number of misclassified bare earth points in a sample as a percentage of all bare earth points in the sample. The developed algorithm does not exhibit large error variations in type I errors between the sample sites. Considering that the parameters used in all the tests were nearly the same, this is encouraging because it indicates that the algorithm is more robust to different landscape types and hence is more reliable.

The type II errors committed are shown in figure 7.3. The type II error presents the number of misclassified object points in a sample as a percentage of all object points in the sample. The type II errors obtained were relatively small, except for a few sites where the prevalence of low vegetation and large point spacing led to higher errors. Typically, there are more bare earth points then there are object points hence the impact of type II errors on the total error is small.

The results for each site shall now be discussed in more detail.



	11	12	21	22	23	24	31	41	42	51	52	53	54	61	71
—□— Axelsson	3.65	1.48	18.53	5.87	4.34	7.45	1.03	1.89	0.26	12.00	14.21	16.76	4.93	6.23	13.25
—○— Pfeifer	2.41	1.52	1.64	3.08	3.81	8.95	2.04	1.57	0.24	1.93	5.68	14.23	1.00	0.17	1.59
—◇— Roggero	3.88	0.91	0.00	1.01	1.94	1.70	1.08	2.74	0.26	6.96	9.66	4.74	8.37	0.58	2.77
—●— New Alg	11.96	3.82	2.55	5.44	3.66	6.07	2.68	2.22	2.68	3.75	11.76	19.29	2.31	12.11	16.27

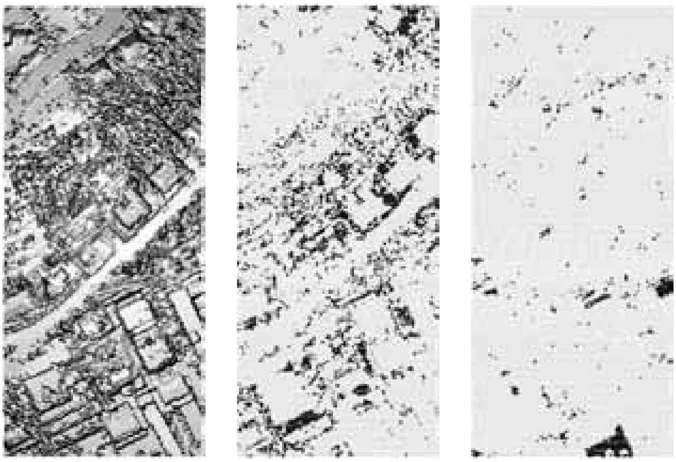
Figure 7.3: Type II errors over the 15 samples

- **Sample 11**: Of all the 15 samples, this was the most difficult (for all the algorithms tested) because of the steeply vegetated slopes and the tiered buildings. Figure 7.4 shows that most of the Type I errors occur on the slopes because of over segmentation on the slopes. The increased point density in the overlap regions leads to under segmentation thus causing the macro object detection to fail. Hence type II, errors in the overlap regions.

- **Sample 12**: The algorithm performed very well on this site (figure 7.5). The flat bare earth and well elevated buildings make this a relatively simple landscape to filter. Type I errors arise from sparse bare earth points beneath the dense vegetation canopy segmenting unfavorably. As in sample 11 the increased point density in the overlap regions is the cause of most of the type II errors.

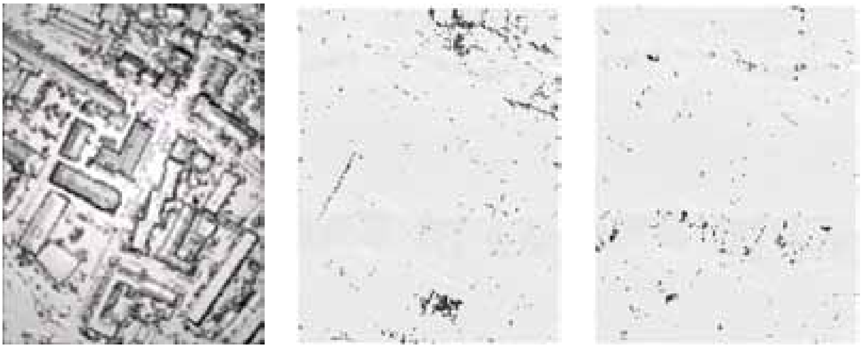
- **Sample 21**: The special characteristic of this site is the bridge. In the filtering, the bridge was successfully detected (using the bridge detection algorithm, section 6.3) and later classified as object. The ends of the bridge were slightly under determined, hence The Type II errors at the ends of the bridge. Type I errors are relatively few in number and mainly result from the misclassification of pavement edges in the micro object detection. Despite this the error is kept small and importantly for an urban scene discontinuities are preserved, see figure 7.6.

- **Sample 22**: The special characteristic of this sample (figure 7.7) are the



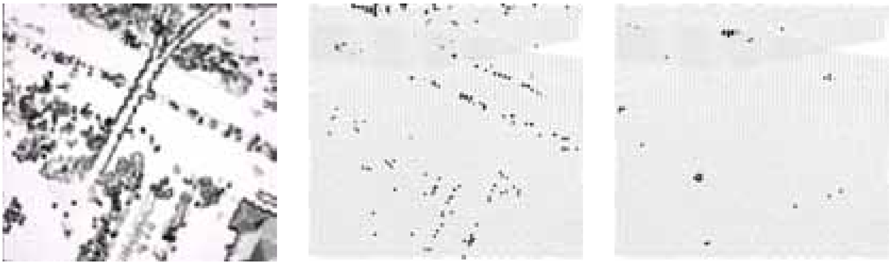
Original Type I Type II

Figure 7.4: Sample 11



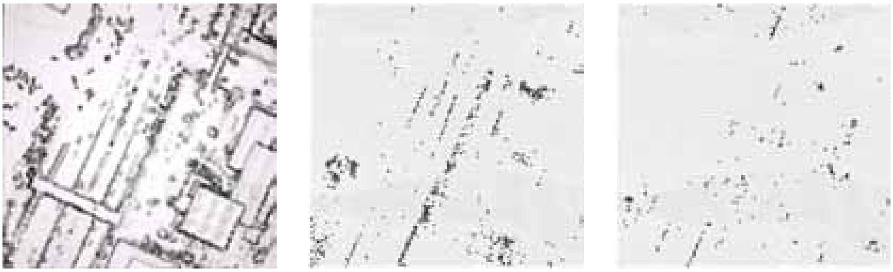
Original Type I Type II

Figure 7.5: Sample 12



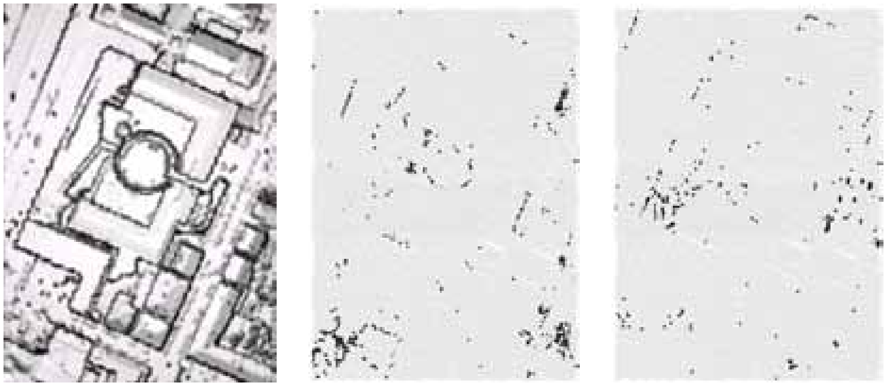
Original Type I Type II

Figure 7.6: Sample 21



Original Type I Type II

Figure 7.7: Sample 22

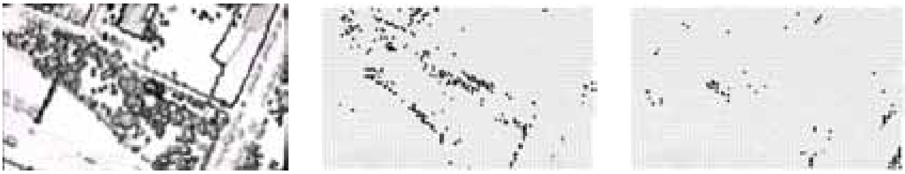


Original Type I Type II

Figure 7.8: Sample 23

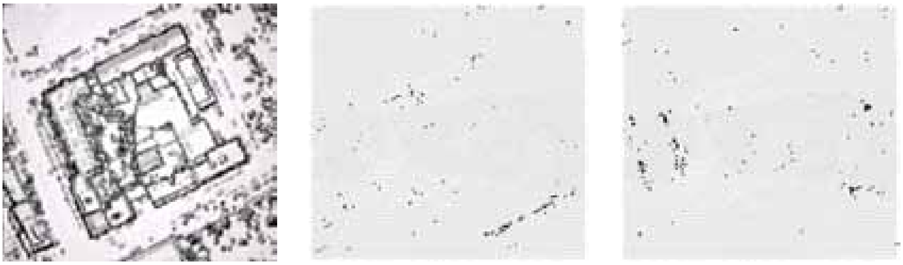
two gangways and the large buildings. The macro object had no problems detecting the large buildings. After the macro object detection the bridge detection algorithm was run and the gangways were detected. The gangways were slightly over detected, hence leading to type I errors. Other type I errors were caused by the micro object detection which mis-classifies low pavement edges. Type II errors were caused by low vegetation.

- **Sample 23**: This site (figure 7.8) is used to test filters in a complex urban scene. It is difficult to define what is and is not bare earth within the building complex. For example, there is a stairway that rises from the bare earth to a raised platform and this makes the classification of the platform ambiguous. For simplicity, all surfaces that connect to the bare earth were also classified as bare earth. The new algorithm does very well on this site. A major reason for this is the algorithm's ability to detect large buildings and preserve discontinuities. The inability to preserve discontinuities was



Original Type I Type II

Figure 7.9: Sample 24



Original Type I Type II

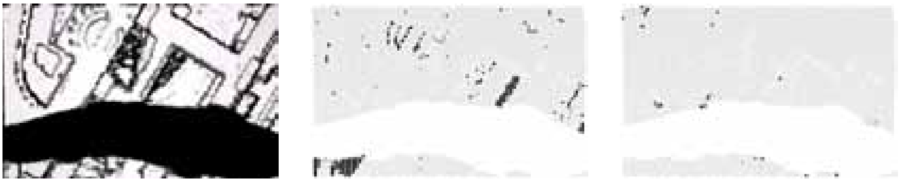
Figure 7.10: Sample 31

identified as one of the reasons why current filter algorithms fare poorly in complex urban scenes.

- **Sample 24**: This small sample shows the effect of a ramp in the landscape, see figure 7.9. Ramps are part of the bare earth, i.e., they are not bridges. Nonetheless, bridge detection was applied to determine if it would mis-classify the ramp. The minimum bridge width used in detecting the ramp was 3m. The width of the ramp is about 3m and hence it was not detected. Most of the type I errors result from misclassification of pavement edges in the micro edge detection. This is also the cause of the errors at the base of the ramp. Type II errors are few and most of them occur in the overlap areas and at the edges of the data.

- **Sample 31**: This sample (figure 7.10) is relatively simple except for the very low outlying point (which cannot be seen). Because of the underlying segmentation concept this site proved unproblematic. In the macro object detection the outlying points are detected as single point segments. Segments containing three points or less are classified as object. This makes the algorithm robust to outliers.

- **Sample 41**: This sample demonstrates the effect of gaps in the data, see figure 7.11. There is insufficient context information to detect the building in the south west corner of the data. Nonetheless, the algorithm outperformed



Original Type I Type II

Figure 7.11: Sample 41

Original Type I Type II

Figure 7.12: Sample 42

all the other algorithms.

- **Sample 42**: The sample is that of a railway station platform (figure 7.12). The aspects of the data that make it interesting are the railway roof, platform and track. In general, the algorithm manages to correctly classify the roofs and platform as object, except for two platforms.The cause of this error is insufficient points on the railway tracks along these platforms. As a result the segments formed for the platforms are determined as strongly *no shape*, and hence classified as bare earth. A roof segment in the south east corner of the data is also misclassified because of insufficient information. The type I errors are caused by high railway tracks that are filtered by the micro object detection and the sides of pavements that are filtered by the micro object detection. In the manual classification, all railway tracks were classified as bare earth.

- **Sample 51**: The succeeding samples are of a lower resolution. The difficulty in this sample (figure 7.13) is the vegetated slopes. The macro object detection did well in removing the buildings above the slopes, but because of under segmentation not all the low vegetation on the slopes were detected. The micro object detection was applied to remove the low vegetation but the balance between Type I and II errors was non-optimal hence the large Type I errors. Even so the total error is relatively small.

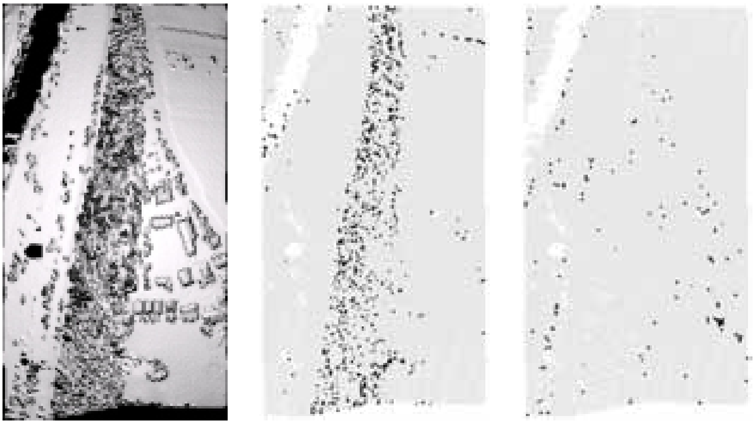

Original Type I Type II

Figure 7.13: Sample 51

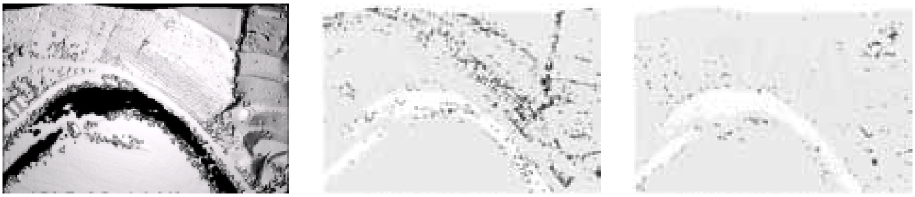
- **Sample 52**: The main cause of error in this site are the low vegetation on the slopes, see figure 7.14. As in sample 51 the non-optimal balance between Type I and II errors is the cause of most of the Type I errors.

- **Sample 53**: Sample 53 is of a quarry and tests an algorithm's ability to preserve discontinuities. The new algorithm out performs all the algorithms. The preservation of discontinuities (figure 7.15) confirms the algorithm's robustness in discontinuous landscapes. Some points on the faces of quarry edges are detected as objects but the cost of this error in a DTM generation should be negligible. Although not implemented the misclassified points on the faces of the quarry could be corrected using the wall detection algorithm. The type II errors are relatively large because of the prevalence of low vegetation.

- **Sample 54**: Sample 54 (figure 7.16) tests the performance of an algorithm in a low density point cloud of an urban scene. The macro detection does well in removing the buildings even though they are poorly defined. In removing the base of buildings the micro object detection also removes bare earth points, hence The type I errors.

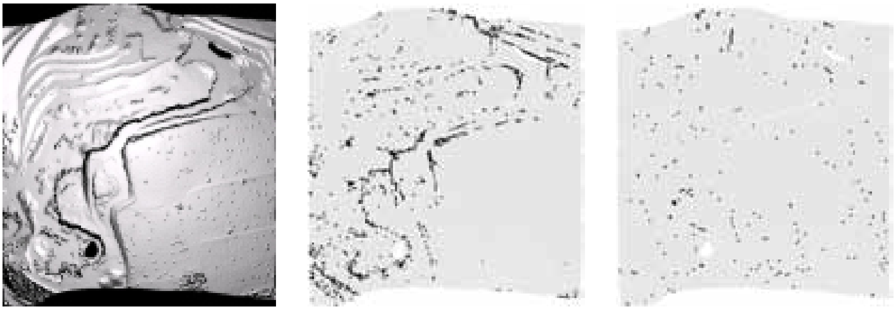
- **Sample 61**: Sample 61 (figure 7.17) contains a data gap, road embankments, ridges and ditches. The challenge in this data set is to filter a landscape that contains relatively few objects. In this the algorithm does very well with a type I error of only 1%. The type II error is caused by low vegetation points on the road embankments.

- **Sample 71**: This sample (figure 7.18) tests the removal of a bridge in a low density data set. It is a relatively difficult sample. The reason for this



Original Type I Type II

Figure 7.14: Sample 52



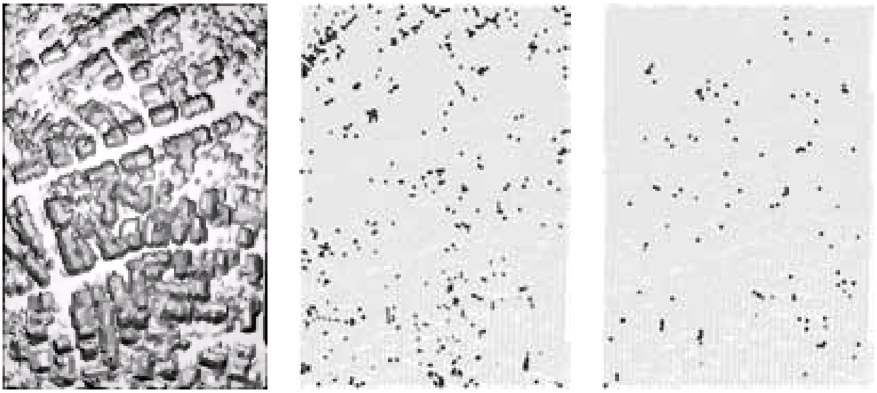
Original Type I Type II

Figure 7.15: Sample 53

difficulty was the overestimation of the bridge. The cause of this error is the steep slopes combined with the low point density. In the bridge detection a slope threshold of 45° was used. The slopes are a little steeper than this therefore the bridge is detected further into the slopes. The solution for this was to increase the slope threshold. Although a better solution was obtained, the bridge was slightly overestimated leading to type I errors. Other type I errors are caused by misclassification of steep slopes and ridges in the bare earth. The type II errors are caused by low vegetation.

One of the primary objectives of the research was to develop a filter algorithm that performs equally well in all landscapes. In this respect and based on the results of the ISPRS test the developed algorithm is deemed to succeed. It performs better or as well as the other algorithms.

The algorithm is able to do this mainly because, (i) segmentation allows better discrimination of large objects, (ii) it targets specific features (e.g., macro objects, bridges, etc.,) in a landscape, (iii) it is better at preserving discontinuities, and (iv) the multi-step nature of the algorithm allows both type I and II errors to be reduced. This is unlike most of the current algorithms that have to make a trade



Original Type I Type II

Figure 7.16: Sample 54

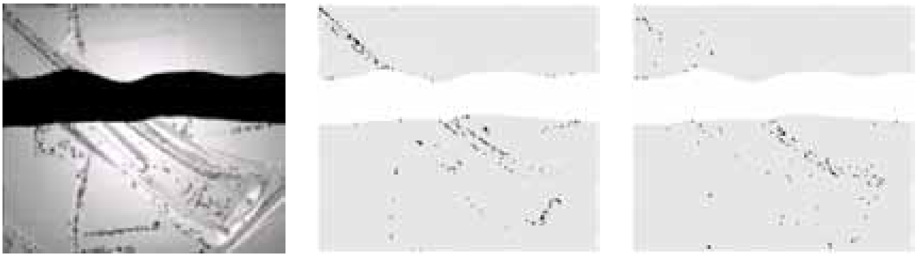
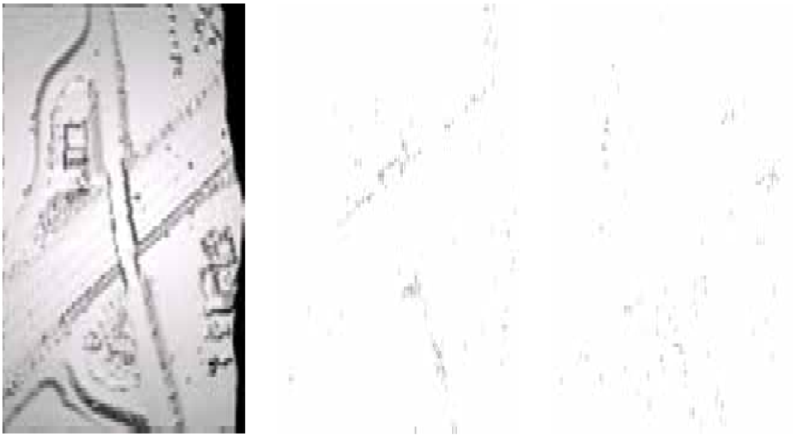


Figure 7.17: Sample 61



Original Type I Type II

Figure 7.18: Sample 71

Cross-matrix

	Bare earth	Man made object	Natural object	Total
Bare earth	124948	14	365	125327
Man made object	272	22988	1529	24789
Natural object	1339	1551	17677	20567
Total	126559	24553	19571	170683

Error-matrix: bare earth, object

	Bare earth	Object	
Bare earth		379 (0.3%)	
Object	1611 (3.6%)		

Error-matrix: bare earth, man made object, natural object

	Bare earth	Man made object	Natural object
Bare earth		14 (0.0%)	365 (0.3%)
Man made object	272 (1.1%)		1529 (6.2%)
Natural object	1339 (6.5%)	1551 (7.5%)	

Table 7.1: Summary of Wijhe results

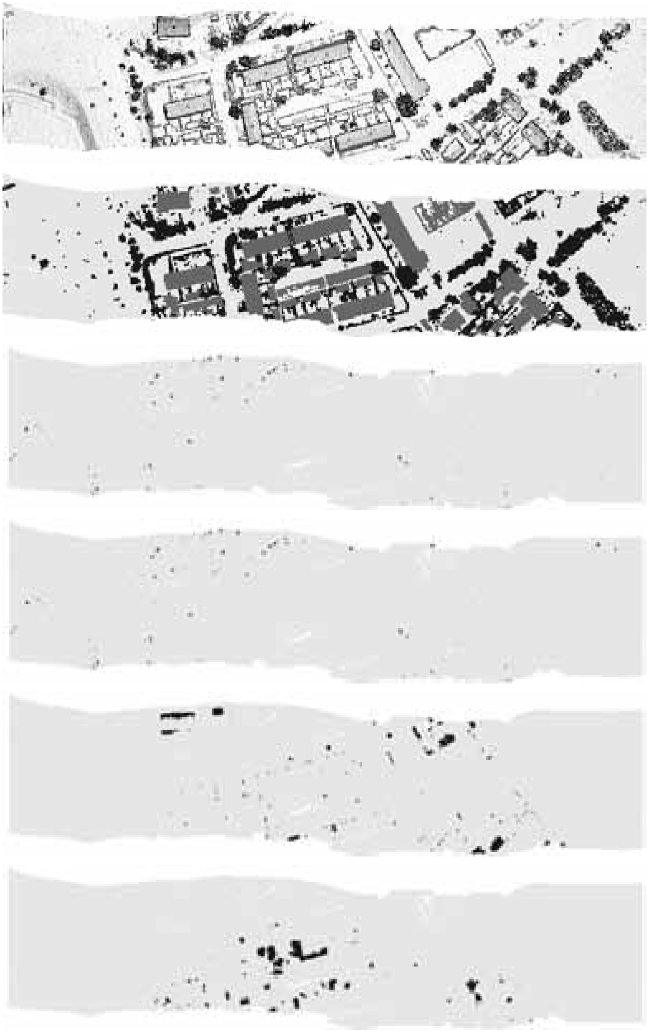
off.

In the following sections the performance of the developed algorithm in respect to the classification of man made objects, natural objects and bridges is tested.

7.3 Wijhe - High density data with reflectance

This data set was chosen because it is a high resolution sample of a residential area, figure 7.19. The bare earth in the landscape is flat except for a road embankment to the south west of the data set. The buildings are relatively small and are surrounded by both high and low vegetation.

The results after application of the algorithms are summarized in figure 7.19 and table 7.1. The results are discussed on two levels, (i) the separation between the bare earth and objects, and (ii) the separation between man made and natural objects. Overall, the errors are small and the algorithm can be said to have performed well. Particularly in the bare earth and object separation.



Shaded relief

Manual classification: bare earth (light gray), buildings (dark gray), vegetation(black).

Bare earth misclassified as object

Object misclassified as bare earth

Building misclassified as vegetation

Vegetation misclassified as building

Figure 7.19: Wijhe test. Depiction of errors in the cross-matrix

Errors from bare earth and object separation

The error in the misclassification of bare earth points (Type I error) is 0.3%. This error is very small. The error arises from sparse points beneath dense vegetation canopies. Sparse points do not segment well, leading to very small segments. In the classification, very small segments (arbitrarily chosen as segments with 5 points or less) were classified as object. Some other causes of bare earth misclassification included:

- Some bare earth points at the base of buildings are misclassified as building in the wall detection.
- Sparsely spaced points lead to small segments that are in turn classified as vegetation. For example this happens beneath dense trees.
- In the micro detection some hillocks (depending on their size) are detected as object.

The misclassification of object points (Type II error) is about 3.6%. The main cause of this error is the misclassification of very low vegetation points. This is evidenced by the misclassification of vegetation points as bare earth, which is 6.5%. In the micro object detection the fit threshold used was 0.2m. Therefore, vegetation below this threshold was not detected. Some other causes of object misclassification included:

- If the bare earth gradually blends into the base of a building macro or micro object detection will fail to detect some object points.
- Glass houses and buildings covered by material that absorb laser pulses lead to sparse point coverage for such buildings. Sparse coverage can lead to islands in the macro object detection. Because these islands are shaped as *no shape* they are classified as bare earth.

The total error in the bare earth object separation is about 1.1%. The flatness of the bare earth combined with the fact that the object misclassification arises from low vegetation means that the cost of this error in a DEM generation will be small.

Errors from building (man made object) and vegetation (natural object) separation

These classifications show a relatively larger error than those of the bare earth - object classification. However, considering that only two features were used, i.e., roughness and reflectance, this result is deemed good. Most of the errors result from the misclassification of vegetation. Part of the cause for this large error is the strength of the discriminators used in the classification. Surface segment roughness and reflectance were used in a 2-dimensional feature space to separate between buildings and vegetation. As explained in appendix A most of the strength of the discrimination is in the roughness characteristic of a surface segment. Therefore, for objects for which roughness is a poor discriminator the reflectance discrimination may likely fail to compensate. An example of this is hedges whose segments have small roughness and low reflectance, and are therefore classified as building. Other causes of vegetation misclassification included:

- Hedges or small trees next to buildings are merged into the same segment as the buildings. If the roughness values of the building dominate those of the vegetation then the vegetation is classified as building.
- Hedges have fairly smooth surfaces and hence the segments formed by them have small roughness.

Small sheds with steep sloped roofs can have strong roughness. If they are covered by vegetation or reflective material then the points from them will posses strong reflectance signatures. Such buildings are misclassified as object. Other causes of building misclassification included:

- Where wall detection is unsuccessful, wall points are classified as vegetation because they possess strong roughness.
- Chimneys on roofs maybe classified as vegetation if their points are planimetrically tightly packed.
- Sparsely sampled buildings will yield small segments. The roughness value computed for small segments maybe overestimated, leading to a classification as building.
- Small buildings may be merged in the same segment as large vegetation causing the small building to be classified as vegetation.
- Small buildings (shacks) may be covered by material whose spectral characteristics are closer to that of vegetation. This leads to a classification as vegetation.

- The roughness at a point is computed using a fairly large neighborhood (about 15 to 20 points). Hence for small buildings with slanted roofs, a neighborhood will invariably contain the roof apex and points from various faces of the roof. This results in a large roughness for the building segment and hence a classification as vegetation.

7.4 Nijmegen - Medium density data with RGB attributes

This data set was chosen because it is a medium resolution sample of an urban area, figure 7.20.

The results after application of the algorithms are summarized in figure 7.20 and table 7.2. Overall errors are relatively small except for the mis-classification of vegetation. This is a satisfactory result considering that in this data set there are more object points than there are bare earth points. As in the Wijhe test the results shall be discussed on two levels, (1) the separation between the bare earth and objects, and (2) the separation between man made and natural objects.

Errors from bare earth and object separation

The error in the misclassification of bare earth points (Type I error) is about 3.8%. This is a very small error and the source is mainly sparse points beneath dense vegetation canopies. Sparse points segment unfavorably and this increases the chance of misclassification. Other causes of bare earth misclassification included:

- Errors at platform edges caused by the micro detection algorithm (not critical)
- Errors beneath trees caused by point sparsity. Point sparsity leads to small segments. Small segments are classified as vegetation.
- Errors at the edge of buildings caused by open drainage systems and pavements.
- Some errors because of edge effects.

The misclassification of object points (Type II error)is about 3.6%. The major source of this error is the misclassification of very low vegetation points. In the

Cross-matrix

	Bare earth	Man made object	Natural object	Total
Bare earth	39781	401	1184	41366
Man made object	58	33588	2409	36055
Natural object	624	2698	19585	22907
Total	40463	36687	23178	100328

Error-matrix: bare earth, object

	Bare earth	Object	
Bare earth		1585 (3.8%)	
Object	682 (1.2%)		

Error-matrix: bare earth, man made object, natural object

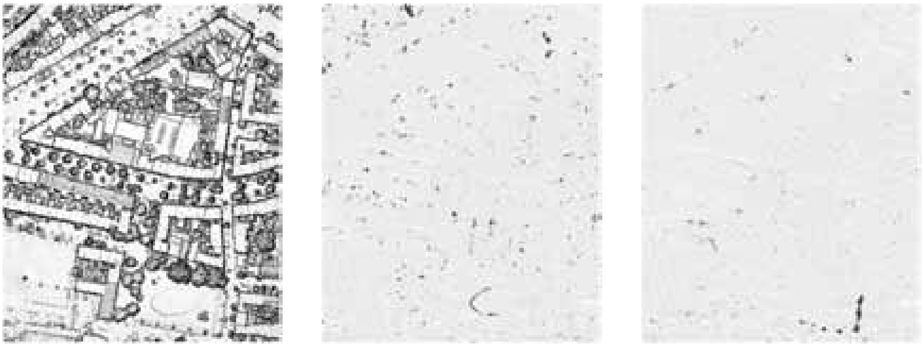
	Bare earth	Man made object	Natural object
Bare earth		401 (1.0%)	1184 (2.9%)
Man made object	58 (0.2%)		2409 (6.7%)
Natural object	624 (2.7%)	2698 (11.8%)	

Table 7.2: Summary of Nijmegen results

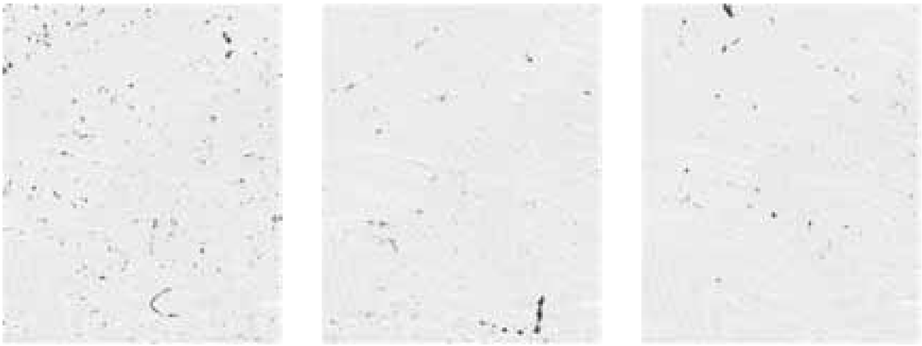
micro object detection the fit threshold used was 0.2m. Therefore, vegetation below this threshold was not detected. The cost of the misclassification of low vegetation in a DEM generation should therefore be small. Other causes of object misclassification included:

Causes of object misclassification

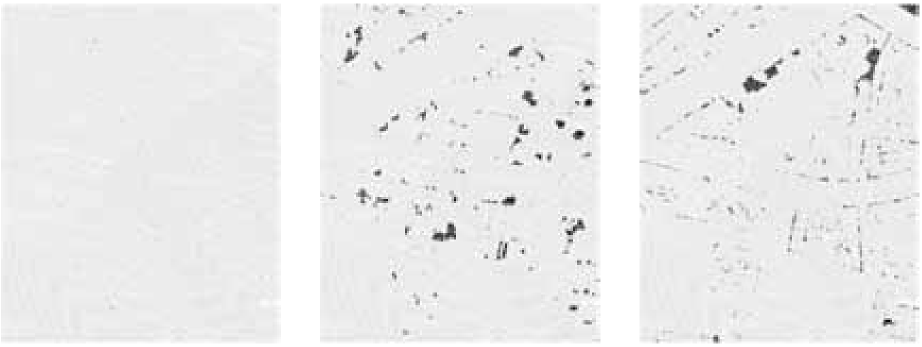
- Very low vegetation are not detected by the micro detection algorithm.
- Overlapping regions create an area of fuzziness in the bare earth. In the manual classification the highest points, despite being bare earth were classified as vegetation. In the automatic classification these points, because they are very low are not detected by the micro detection algorithm.
- Very few building points (57) are misclassified, because most buildings are fairly well off the ground. Those points that have been misclassified are at the base of buildings.



Shaded Bare earth classified object Object classified bare earth



Bare earth classified veg. Veg. classified bare earth Bare earth classified veg.



Build. classified bare earth Veg. classified build. Build. classified veg.

Figure 7.20: Nijmegen test. Depiction of errors in the cross-matrix

Errors from building (man made object) and vegetation (natural object) separation

The errors generated from these classifications were much larger. The total error in this classification was about 8.8% (5107 points). However, this is comparable to results typically obtained in the classification of landscapes in rasterized laser data. Considering that classifying the raw data poses greater difficulties than it's raster counterpart, this result is deemed acceptable.

As in the Wijhe data part of the cause for this large error is the strength of the discriminators used in the classification. The strong mixing (appendix A) of the point and surface segment characteristics weakens the partitioning of the feature spaces. To overcome this mixing, a KNN classifier was used. The drawback of using this classifier is that the need for a representative training set becomes more critical, as oversampling of buildings or vegetation on the boundaries of the building and vegetation clusters can dramatically affect the classification results. This can be appreciated by examining the scatter plots in figure A.4.

Another strong contributor to the misclassification is the surface segmentation itself. From figure 7.20 it can be observed that most of the building-vegetation errors occur where there is vegetation adjacent to buildings. In these regions, vegetation is sometimes merged into building segments and hence vegetation is classified as building.

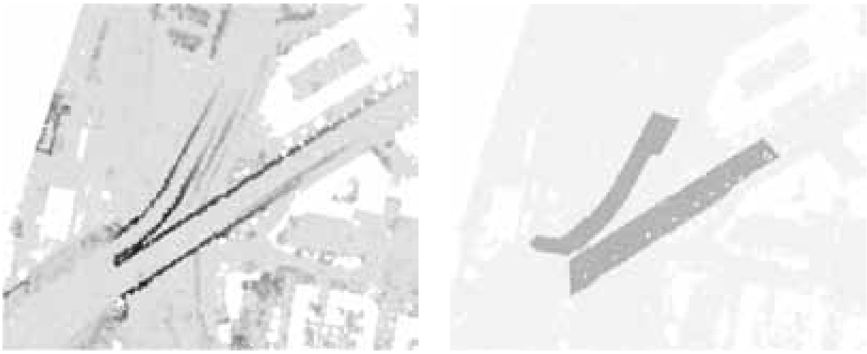
Other causes of vegetation misclassification included:

- Some vegetation are merged into a large building segments and then classified as building
- Elongated but thin vegetation segments are misclassified as wall points
- Stray vegetation in the vicinity of buildings are misclassified in the K nearest neighbour classification
- The uncertainty in the mixed regions of the feature space combined with insufficiency in the training can in a few instances lead to a misclassification of building segments. To avoid bias, in the training data the number o building and vegetation samples are kept nearly the same. Vegetation shows much more variation than buildings and therefore the balancing of samples can lead to the poor training of vegetation. This is a bigger problem for vegetation than it is for buildings.

Buildings that are over segmented (as happens with steeply roofed buildings) are classified as vegetations. This is because as a rule small segments are classified as vegetation on the assumption that small segments cannot be buildings.

Other causes of building misclassification included:

- Not all wall points classified correctly, because of over segmentation. Over segmentation is necessary to reduce the effect of vegetation next to buildings



Landscape minus macro objects Detected overpass

Figure 7.21: Bridge detection

- Complex roof segments possess strong roughness values because many of the neighborhoods used in the local roughness determinations will contain discontinuities. This is also because the point spacing is insufficient to counter this problem.
- Some buildings are merged into a large vegetation segment and then classified as vegetation
- Some motor vehicles and other non vegetation materials because of their smoothness and spectral characteristics are classified as building
- Some building segments are mapped into mixed areas in the feature space. The uncertainty in the mixed regions combined with insufficiency in the training can in a few instances lead to a misclassification of building segments.

7.5 Nijmegen data - overpass

A sample from another part of the city of Nijmegen, figure 7.21, was used to test the performance of the bridge detection algorithm. In this landscape, there is an overpass that starts from an elevated position and divides into two as it descends to the bare earth below. In the sample, macro objects were first removed from the landscape leaving behind potential bare earth.

The landscape was then segmented using profile intersection with labeling by slope. Five profile directions were used in the segmentation. This number is greater than that used in the macro object detection segmentation. Because bridge detection is primarily a point based classification it is important to gather sufficient profiles. The slope threshold used in the profile labeling controls the extension of the bridge into the bare earth. The smaller the threshold the more the bridge extends into the bare earth. A slope threshold of 45° degrees was found to work well. The overpass was successfully detected, but as can be seen in figure 7.21 one is also over estimated at its higher end. The cause of this is the steep slopes ($> 45°$) of the embankments. Because a threshold of 45° degrees was used the algorithm incorrectly detects the bare earth in this area as an extension of the overpass. Correcting this problem requires a modification of the slope labeling algorithm so that it accounts for variations in slope. Unfortunately, there was insufficient time to effect this modification.

The fact that the branching overpass is detected indicates that the foundation of the bridge detection concept is sound.

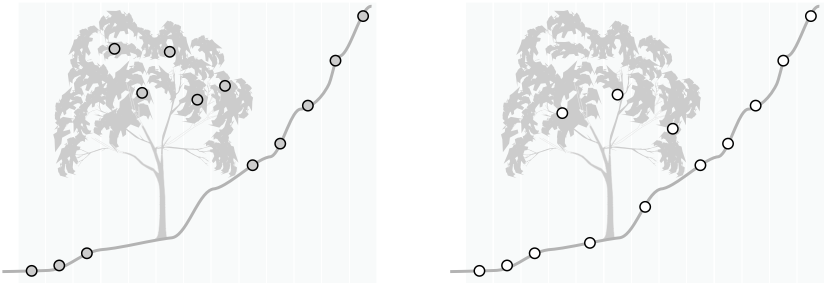
7.6 Using first pulse returns to detect natural objects in last pulse returns

In section 5.5 it was argued that first pulse returns were of marginal value in the segmentation, but were useful for the classification. In vegetated areas first pulse returns typically come from the top of vegetation, and the last pulse from the bare earth or some point within the vegetation canopy.

Because of this, in vegetated areas there is a separation between corresponding first and last pulses returns. A large separation indicates that the first pulse is from vegetation. Thresholding this separation (in the first pulse data) using the accuracy of the point measurement yields potential *vegetation* points and *unclassified* points.

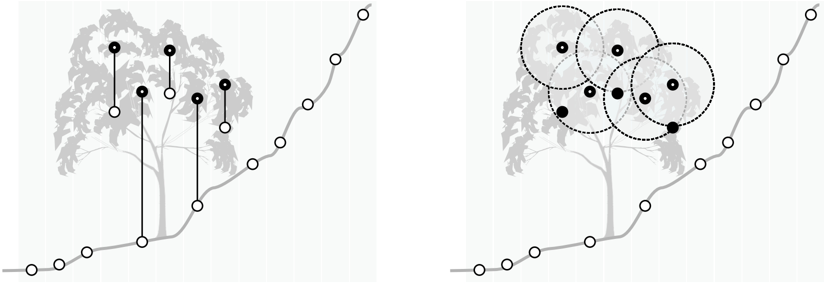
Necessarily vegetation points can only be identified in the first pulse returns. Recalling that for bare earth detection the last pulse returns are preferred, then some method has to be found to use the detected vegetation points to aid in the classification of last pulse returns. The method proposed here is to classify every last pulse return in the vicinity of a vegetation point (first pulse) as vegetation. This method works on the assumption that any point in the vicinity of a *vegetation* point must itself also be a vegetation point (figure 7.22).

There was insufficient time to include this approach into an actual classification algorithm, but visual examination of a simple implementation indicates that it is viable and holds promise for better classifications. An example of the approach is shown in figure 7.23. First pulse points that are 0.5m above their corresponding second pulse were labeled vegetation. The 5 (intuitively chosen based on the



First pulse returns

Last pulse returns



Distance between first and last pulse returns thresholded. First pulse points with distances above the threshold are classed as vegetation (black points).

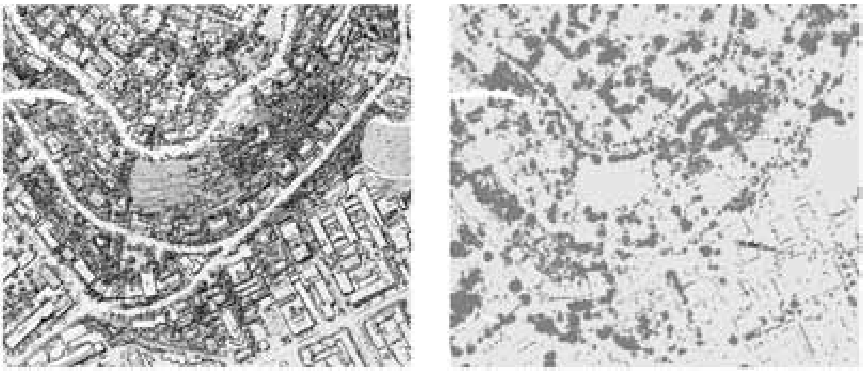
Last pulse returns in the neighborhood of vegetation points classed as vegetation.

Figure 7.22: Using first pulse returns to classify vegetation in last pulse returns.

resolution of the point cloud) nearest second pulse neighbors of these vegetation points were then also labeled vegetation. As can be seen in the figure a considerable number of vegetation points on the slopes are detected. This offers a better starting point for the other algorithms. One drawback of the approach is that points on the edges of some buildings can be misclassified as vegetation.

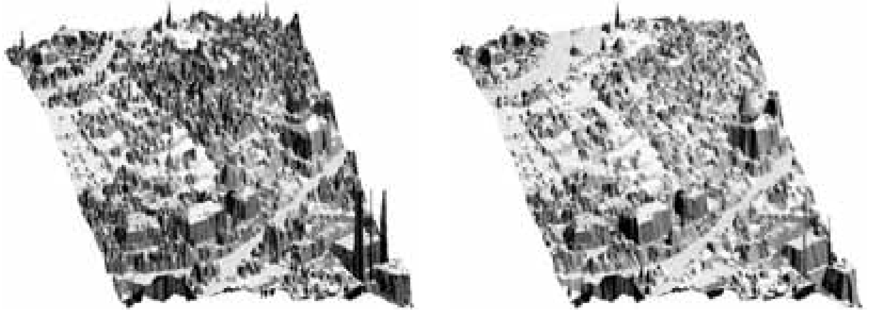
7.7 Discussion

Eventually most filtered ALS point clouds are used to generate DTMs. The preservation of discontinuities is important in reducing the cost of errors. Because of this even though the developed algorithm has generated higher errors on some samples, the cost of the error in generating a DTM maybe lower.



Steep slopes with vegetation

First pulse points (dark gray) that are strongly separated from the last pulse points (light gray)



Steep slope with vegetation (last pulse)

Last pulse points in the neighborhood of first pulse points removed

Figure 7.23: A possible use of first pulse data.

Despite the performance gains, the algorithm has difficulties in classifying features at the edges of the data, and difficulties in filtering low resolution point clouds (even more than other algorithms). At first glance, this may appear to be a weakness. The algorithm is designed to classify a surface segment and if there is insufficient topological or contextual information, the class of the surface segment will be uncertain. This problem is similar to that faced by a human operator confronted with a surface at the edge of the data. The smaller the surface the greater the uncertainty. The fewer contextual information available, the greater the uncertainty. The solution in this case is to trust only the classification of those points away from the edge of the data.

The man made - natural object classification generated total errors of less than 10%. This is similar to that generated by other documented algorithms. However, most other algorithms work with raster data where wall points are not a problem. A large contributor to errors in this classification is the merging of vegetation and buildings into the same segment. This suggest that an alternative method of segmentation needs to be sought or segments have to be split in a pre processing step before classification.

In conclusion, the errors obtained in the classifications (by most of the algorithms tested) have mostly been less than 10%. Compared to classical land use classification from imagery this is extremely good. This shows that the geometric information contained in a point cloud (i.e., the position of points) is a strong basis for discriminating different features in a landscape. The challenge now is to develop sound concepts that combine both the geometric information in ALS point clouds and radiometric information from other sources to improve the classification of digital landscapes. In the developed algorithm, radiometric information has been used to aid in distinguishing between buildings and vegetation. Good results have been obtained but these can be further improved.

Chapter 8

Conclusion and future work

8.1 Conclusion

An experimental study of eight different filtering algorithms was done to gauge the design strengths and weakness of different classification algorithms. The major findings of the study were that (i) surface based filters performed better than structure based filters, (ii) most of the current algorithms are ill suited to preserving discontinuities in the bare earth (a major problem in urban landscapes), (iii) most of the algorithms are landscape-type specific, (iv) type I and type II errors cannot be minimized using a single algorithm, (v) features like bridges that are extensions of the bare earth need to be detected separately, and that (vi) segmentation based filtering approaches have the potential to offer greater reliability of classification.

To solve some of the above problems a new algorithm was developed. The new algorithm departs from most of the current algorithms in that it is segmentation based, and bridges are detected explicitly. Unlike most of the current algorithms, the new algorithm works based on the relationship of surfaces in a landscape as opposed to the relationship of points in a landscape. Moreover, it is a multi-scale algorithm in that it applies a different algorithm to classify points objects at different scales (not to be confused with using the same algorithm at different scales).

A new segmentation algorithm was developed. The segments obtained from the segmentation of the point cloud are made from sets of parallel profiles. These profiles have the property that they implicitly code the topological relationship of a segment to neighboring segments. The topological information contained in the profiles is aggregated in a novel way to to classify a segment as either object or bare earth. Current segmentation methods classify segments based on the outlines of segments. This can pose difficulties in urban environments where the arrangement of objects can complicate classification. The new segmentation-classification method does not require the outline of an object to be known and because of this it does well in complex environments. Tests on point clouds of complex urban landscapes have shown that the algorithm works well and that its ability to preserve discontinuities reduces many of the errors experienced by other algorithms.

In a landscape, bridges are ambiguous features in that they are extensions of the bare earth but they are also man made. For this reason they have to be detected explicitly. The novel bridge detection algorithm also makes use of the new segmentation algorithm. The topological information contained in the profiles is used to identify seed bridge points. The seed bridge points are then used to detect actual bridges. The bridge detection algorithm is novel in that the algorithm is blind to the shape of a bridge, i.e., bridges need not have parallel sides and bridges can branch out. The bridge detection algorithm was found to work well in most cases. However, the algorithm was found to experience problems where a bridge extends from an embankment and in low resolution data.

An algorithm was also setup to classify man made and natural objects in a point cloud. The algorithm employs the roughness of a surface segment and the reflectance of its constituent points. If available the RGB triplet of segment points are also used. Classification accuracies in the order of 5-10% were obtained. This is comparable or better than the accuracy achievable with other data sources, e.g., satellite imagery. However, with the additional use of other data (e.g., infra red imagery) the accuracies can be further improved.

The algorithm was tested on data sets of different landscape types and resolutions. In general, it is an improvement on current algorithms. However, problems remain on steep slopes, particularly those that are vegetated. Other aspects of the algorithm that can be improved are outlined in the next section. It is envisaged that with the inclusion of external data (e.g., thematic maps, existing digital elevation models, infra red imagery, etc.) and the use of waveform ALS data the classification accuracy of ALS point clouds will only improve.

8.2 Future work

New segmentation and classification procedures have been proposed. The algorithms, particularly the segmentation algorithm were designed to be extendible.

Because of insufficient time it was not possible to fully extend and test them. Therefore, some aspects of the research require further attention.

- The new algorithm does not work as well as expected in steep slopes. This is partly because of the labeling algorithm. A further segmentation of the detected line segments by curve fitting may help prevent objects merging into bare earth segments.
- Many of the parameters in the algorithm were chosen intuitively or by trial and error. They may not necessarily have been optimal parameters. During tests, it was observed that there is a link between the choice of certain segmentation parameters and the point spacing. This link requires further investigation.
- The classification of buildings and vegetation posed one of the greatest problems. The problems mainly arose from the merging of buildings and vegetation into the same segment. It has been found that segmentation by profile intersection can lead to loosely connected segments, as exists between vegetation and buildings. Because of this after segmentation a post processing step is necessary to remove the loose connections. A possible means of detecting loose connections is a *strongly connected components* analysis.
- In the ISPRS filter test a quantitative assessment was possible because of manually generated reference data. The results showed that filters are not foolproof and performance can vary from one type of landscape to another. Therefore, while testing a filter against reference data is a good measure of gaining an appreciation of the filters performance, it is not a guarantee that a filter will always perform as expected. If the type of environment being filtered is untested, then unpredictable results can be and should be expected. In these circumstances, it would be advantageous to an operator (doing quality control) if filters could be designed to report the anticipated quality of the filtering and/or flag where the filter may have encountered difficulties.

  One possible way of identifying uncertain regions is to apply a filter on a point cloud using different parameter values. The areas whose classification changes with a variation of the parameter values are flagged as uncertain. Gooch and Chandler (2001) use such a scheme in the prediction of failure in automatic DEM generation. However, some features (such as large buildings) are invariant to changes in filter parameters. Therefore, another approach is proposed here. In the simplest approach an uncertainty band is placed on the class grade in equation 6.4. Those surface segments that have a class grade within the uncertainty band (e.g., 0.5-0.6) can be flagged as uncertain segments and set aside for classification by a human operator. In this research, this flagging method was tried with some success, but it still requires more investigation.

- In the concept of the segmentation by profile intersection, profiles are not allowed to overlap. This is partly because it is desired that parallel profiles only connect through cross profiles. However, if the point spacing is large, then there may be some merit in allowing profiles to overlap.

Bibliography

Abdou, I. and W. Pratt (1979). Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE*, pp. 753–763.

Ackermann, F. (1999). Airborne laser scanning - present status and future expectations. *ISPRS JPRS vol. 54*(2-3), pp. 64–67.

Airborne1 (2004). lidar industry directory 2004. Website. `http://www.airborne1.com/downloads/2004LIDprev.pdf` - last accessed May 2004.

Akel, N. A., O. Zilberstein, and Y. Doytsher (2003). Automatic DTM extraction from dense raw lidar data. *Proceedings of FIG working week 2003. April 13-17, Paris, France.*, 10 pages.

Amenta, A. B., M. W. Bern, and D. Eppstein (1998). The crust and the β-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing vol. 60*(2), pp. 125–135.

Axelsson, P. (1999). Processing of laser scanner data - algorithms and applications. *ISPRS JPRS vol. 54*, pp. 138–147.

Axelsson, P. (2000). DEM generation from laser scanner data using adaptive tin models. *IAPRS vol. 33*(B4/1), pp. 110–117.

Axelsson, P. (2001). Ground estimation of laser data using adaptive tin-models. *Proceedings of OEEPE workshop on airborne laser scanning and interferometric SAR for detailed Detailed Digital Elevation Models. March 1-3, Stockholm, Sweden. Official Publication No. 40. CD-ROM.*, 11 pages.

Baltsavias, E. (1999a). Airborne laser scanning - relations and formulas. *ISPRS JPRS vol. 54*, pp. 199–214.

Baltsavias, E. (1999b). Airborne laser scanning: existing systems and firms and other resources. *ISPRS JPRS vol. 54*, pp. 164–198.

Baltsavias, E. (1999c). A comparison between photogrammetry and laser scanning. *ISPRS JPRS vol. 54*, pp. 83–94.

Bose, P., W. Lenhart, and G. Liotta (1993). Characterizing proximity trees. pp. pp. 9–11. Centre D'Analyse et de Mathématique Sociales, Paris Sorbonne.

Briese, C. and N. Pfeifer (2001). Airborne laser scanning and derivation of digital terrain models. *Proceedings of the 5th conference on optical 3D measurement techniques, Vienna, Austria*, pp. 80–87.

Briese, C., N. Pfeifer, and P. Dorninger (2002). Applications of the robust interpolation for DTM determination. *IAPRS vol. 34*(3A, September 9 - 13, Graz, Austria), pp. 55–61.

Brovelli, M., M. Cannata, and U. Longoni (2002). Managing and processing lidar data within GRASS. *Proceedings of the Open Source GIS-GRASS users conference, 11-13 September, Trento, Italy*, 29 pages.

Brovelli, M., M. Cannata, and U. Longoni (2004). Lidar data filtering and DTM interpolation within grass. *Transactions in GIS vol. 8*(2), pp. 155–174.

Davis, L. (1975). A survey of edge detection techniques. *Computer graphics and image processing. vol. 4*, pp. 248–270.

Elmqvist, M. (2001). Ground estimation of laser radar data using active shape models. *Proceedings of OEEPE workshop on airborne laser scanning and interferometric SAR for detailed Detailed Digital Elevation Models, March 1-3, Stockholm, Sweden. Official Publication No. 40. CD-ROM.*, 8 pages.

Elmqvist, M. (2002). Ground surface estimation from airborne laser scanner data using active shape models. *IAPRS vol. 34*(3A, September 9-13, Graz, Austria), pp. 114–118.

Elmqvist, M., E. Jungert, F. Lantz, A. Persson, and U. Soderman (2001). Terrain modelling and analysis using laser scanner data. *IAPRS vol. 34*(3/W4, October 22-24, Annapolis (MD), USA.), pp. 219–227.

Filin, S. (2002). Surface clustering from airborne laser scanning data. *IAPRS vol. 34*(3A, September 9-13, Graz, Austria), pp. 119–124.

Flood, M. (1999). Commercial development of airborne laser altimetry: a review of the commercial instrument market and its projected growth. *IAPRS vol. 32*(3W/4, WG III/5 and WG III/2. November 9-11, La Jolla (CA), USA), pp. 13–20.

Flood, M. (2001a). Laser altimetry: From science to commercial lidar mapping. *PE&RS vol. 67*(II), pp. 1209–1217.

Flood, M. (2001b). Lidar activities and research priorities in the commercial sector. *IAPRS vol. 34*(3W/4, October 22-24, Annapolis (MD), USA), pp. 678–684.

Fram, J. and E. Deutsch (1975). On the quantitative evaluation of edge detection schemes and their comparison with human performance. *IEEE Transactions on Computers vol. C-24*, pp. 616–628.

Fu, K. and J. Mui (1981). A survey on image segmentation. *Pattern Recognition vol. 13*, pp. 3–16.

Gooch, M. and J. Chandler (2001). Failure prediction in automatically generated digital elevation models. *Computers & Geosciences vol. 27*(8), pp. 913–920.

Gorte, B. (2002). Segmentation of tin-structured surface models. *ISPRS WG IV/6, Joint Conference on Geo-spatial theory, Processing and Applications. July 8 - 12, Ottawa, Canada*, 5 pages.

Haralick, R. (1983). Image segmentation survey. *Fundamentals in Computer Vision O.D. Faugeras(ed) Cambridge University Press*, pp. 209–224.

Haralick, R. (1985). Survey-image segmentation techniques. *Computer Vision Graphics and Image Processing vol. 29*, pp. 100–132.

Hartsfield, N. and G. Ringel (1994). *Pearls in Graph Theory: A Comprehensive Introduction.* Revised and Augmented. Academic Press Inc. San Diego, United States of America.

Haugerud, R. and D. Harding (2001). Some algorithms for virtual deforestation (vdf) of lidar topographic survey data. *IAPRS vol. 34*(3W/4, WG IV/3, October 22-24, Annapolis(MD), USA.), pp. 211–218.

Heath, M., S. Sarkar, T. Sanocki, and K. Bowyer (1996). Comparison of edge detectors: A methodology and initial study. *CVPR'96 vol. 21*, pp. 143–148.

Hoffman, R. and A. Jain (1987). Segmentation and classification of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 9*(5), pp. 608–620.

Hoover, A., G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher (1996). An experimental comparison of range image segmentation algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence vol. 18*(7), pp. 673–689.

Houzelle, S. and G. Giraudon (1992). Automatic feature extraction and localization using data fusion of radar and infrared images. *AGARD, Radio location Techniques (SEE N93-23598 08-32)*, 9 pages. Provided by the NASA Astrophysics Data System.

Huising, E. and L. M. G. Pereira (1998). Errors and accuracy estimates of laser altimetry data acquired by various laser scanning systems for topographic applications. *ISPRS JPRS vol. 53*(5), pp. 245–261.

Hyyppa, H. and J. Hyyppa (1999). Comparing the accuracy of laser scanner with other optical remote sensing data sources for stand attribute retrieval. *Photogrammetric Journal of Finland, 1999 vol. 16*(2), pp. 5–15.

Jain, A. and R. Dubes (1988). *Algorithms for clustering data.* Prentice Hall International Inc, New Jersey, USA. ISBN 0-13-022278-X.

Jain, A., R. Duin, and J. Mao (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence vol.22*(1), pp. 4–37.

Jain, A., M. Murty, and P. Flynn (1999). Data clustering: a review. *ACM computing surveys vol. 31*(3), pp. 264–324.

Jiang, X. and H. Bunke (1994). Fast segmentation of range images into planar regions by scanline grouping. *Machine vision and applications vol. 7*(2), pp. 115–122.

Kasser, M. and Y. Egels (2002). *Digital Photogrammetry.* Taylor and Francis, London.

Katzenbeisser, R. (2003). About the calibration of lidar sensors. *IAPRS vol. 34*(3/W13, October 8-10, Dresden, Germany), 6 pages.

Kilian, J., N. Haala, and M. Englich (1996). Capture and evaluation of airborne laser scanner data. *IAPRS vol. 31*(B3, Vienna, Austria), pp. 383–388.

Klein, J. and G. Zachmann (2004). Proximity graphs for defining surfaces over point clouds. In *Symposium on Point-Based Graphics*, pp. 8 pages.

Kraus, K. and N. Pfeifer (1998). Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS JPRS vol. 53*, pp. 193–203.

Kraus, K. and N. Pfeifer (2001). Advanced DTM generation from lidar data. *IAPRS vol. 34*(3W/4, WG IV/3. October 22-24, Annapolis(MD), USA), pp. 23–30.

Lee, I. and T. Schenk (2001). 3d perceptual organization of laser altimetry data. *IAPRS vol. 34*(3W/4, WG IV/3, October 22-24, Annapolis(MD), USA), pp. 57–65.

Lee, I. and T. Schenk (2002). Perceptual organization of 3d surface points. *IAPRS vol. 34*(3A, September 9-13, Graz, Austria), pp. 193–198.

Lindenberger, J. (1991). Methods and results of high precision airborne laser profiling. *Proceeding of the 43rd Photogrammetric Week at Stuttgart University, September 9 -14, Stuttgart, Germany*, pp. 83–92.

Lindenberger, J. (1993). *Laser-Profilmenssungen zur topographischen Gelandeaufnahme.* Ph. D. thesis.

Lohmann, P. (2000). Segmentation and filtering of laser scanner digital surface models. *IAPRS vol. 34*(2, WG II/2, August 22-23, Xi'an, China), pp. 311–315.

Lohmann, P. and A. Koch (1999). Quality assessment of laser-scanner data. *Proceedings of the ISPRS Joint workshop "Sensors and Mapping from space", University of Hanover, Institute for Photogrammetry and Engineering Surveys, WG I/1 & I/3 &IV/4, on CD-ROM*, 9 pages.

Lomenie, N., J. Barbeau, and R. Trias-Sanz (2003). Integrating textural and geometric information for an automatic bridge detection system. *Proceedings of the 3rd International Conference on Computer Vision Systems (ICVS 03). Lecture Notes in Computer Science. April 1-3, Graz, Austria vol. 2626*, pp. 172–181.

Masaharu, H. and K. Ohtsubo (2002). A filtering method of airborne laser scanner data for complex terrain. *IAPRS vol. 34*(3B, September 9-13, Graz, Austria), pp. 165–169.

Patane, G. and M. Spagnuolo (2002). Multi-resolution and slice-oriented feature extraction and segmentation of digitized data. *Proceedings of the seventh ACM symposium on Solid modeling and applications. June 17-21, Saarbrcken, Germany*, pp. 305–312.

Peli, T. and D. Malah (1982). A study of edge detection algorithms. *Computer graphics and image processing vol. 20*, pp. 1–21.

Petzold, B. and P. Axelsson (2001). Results of the oeepe wg on laser data acquisition. *Proceedings of OEEPE workshop on airborne laser scanning and interferometric SAR for detailed Detailed Digital Elevation Models. March 1-3, Stockholm, Sweden. Official Publication No. 40. CD-ROM.*, 6 pages.

Petzold, B., P. Reiss, and W. Stossel (1999). Laser scanning - surveying and mapping agencies are using a new technique for the derivation of digital terrain models. *ISPRS JPRS vol. 54*, pp. 95–104.

Pfeifer, N., A. Kostli, and K. Kraus (1998). Interpolation and filtering of laser scanner data - implementation and first results. *IAPRS, Columbus vol. 32*(3/1), pp. 153–159.

Pfeifer, N., A. Kostli, and K. Kraus (1999). Restitution of airborne laser scanner data in wooded area. *GIS Geo-Information-Systems vol. 12*, pp. 18–21.

Pfeifer, N., T. Reiter, C. Briese, and W. Rieger (1999). Interpolation of high quality ground models from laser scanner data in forested areas. *IAPRS, WG III/5 and WG III/2. 9-11 Nov. 1999. La Jolla(CA), USA vol. 32*(3W/4), pp. 31–36.

Pfeifer, N. and P. Stadler (2001). Derivation of digital terrain models in the scop environment. *Proceedings of OEEPE workshop on airborne laser scanning and interferometric SAR for detailed Detailed Digital Elevation Models. March 1-3, Stockholm, Sweden. Official Publication No. 40. CD-ROM.*, 13 pages.

Ritter, G., P. Gader, and J. Davidson (1986). Automated bridge detection in flir images. *Proceedings of 8th International Conference on Pattern Recognition. October 27-31, Paris, France*, pp. 862–864.

Rodriguez, E. and J. Martin (1992). Theory and design of interferometric synthetic aperture radars. *Radar and Signal Processing, IEEE Proceedings-F vol. 139*(2), pp. 147–159.

Roggero, M. (2001). Airborne laser scanning: Clustering in raw data. *IAPRS vol. 34*(3W/4, WG IV/3. October 22-24, Annapolis(MD), USA), pp. 227–232.

Roggero, M. (2002). Object segmentation with region growing and principal component analysis. *IAPRS vol. 34*(3A, September 9-13, Graz, Austria), pp. 289–294.

Sahoo, P., S. Soltani, A. Wong, and Y. Chen (1988). A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing vol. 41*, pp. 233–260.

Sampath, A. and J. Shan (2003). Building segmentation from raw lidar data. *ASPRS annual conference, Anchorage, AK, May 5-9. CD-ROM*, 5 pages.

Sampath, A. and J. Shan (2004). Urban modelling based on segmentation and regularization of airborne lidar point clouds. *IAPRS vol. 35*(B3, WG III/3, July 12-23, Istanbul, Turkey), pp. 937–941.

Schachter, B. and A. Rosenfeld (1978). Some new methods of detecting step edges in digital pictures. *Commun. ACM vol. 21*(2), pp. 172–176.

Schenk, T. (1999). Photogrammetry and laser altimetry. *IAPRS vol. 32*(3W/4, WG III/5 and WG III/2. November 9-11, La Jolla(CA), USA), pp. 3–12.

Schickler, W. and A. Thorpe (2001). Surface estimation based on lidar. *Proceedings ASPRS conference April 23-27, St. Louis Missouri. CD-ROM*, 11 pages.

Shan, J. and A. Sampath (2004). Urban dem generation from raw lidar data: a labeling algorithm and its performance. *Photogrammetric engineering and remote sensing.*. In print.

Sithole, G. (2001). Filtering of laser altimetry data using a slope adaptive filter. *IAPRS vol. 34*(3/W4, WG IV/3. October 22-24, Annapolis (MD),USA), pp. 203–210.

Sithole, G. and G. Vosselman (2002a). Filtering strategy: working towards reliability. *Proceedings of the Photogrammetric Computer Vision, ISPRS Commission III, Symposium 2002 September 9-13, 2002 vol. 34*(3A), pp. 330–335.

Sithole, G. and G. Vosselman (2002b). ISPRS filter test. Website. `http://www.geo.tudelft.nl/frs/isprs/filtertest` - last accessed November 2004.

Sithole, G. and G. Vosselman (2003a). Automatic structure detection in a point-cloud of an urban landscape. *Proceedings of 2nd Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (Urban 2003), May 22-23, Berlin, Germany*, pp. 67–71.

Sithole, G. and G. Vosselman (2003b). Comparison of filtering algorithms. *IAPRS vol. 34*(3/W13), pp. 71–78.

Sithole, G. and G. Vosselman (2003c). Isprs comparison of filters. pp. 62 pages. `http://www.geo.tudelft.nl/frs/isprs/filtertest/AppendixB05082003.pdf` - Last accessed November 2004.

Sithole, G. and G. Vosselman (2004). Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS JPRS vol. 59*(1-2), pp. 85–101.

Sohn, G. and I. Dowman (2002). Terrain surface reconstruction by the use of tetrahedron model with the mdl criterion. *Proceedings of the Photogrammetric Computer Vision, ISPRS Commission III, Symposium 2002 September 9-13, 2002 vol. 34*(3A), pp. 336–344.

Tao, C. and Y. Hu (2001). A review of post-processing algorithms for airborne lidar data. *Proceedings ASPRS conference April 23-27, St. Louis Missouri. CD-ROM*, 14 pages.

ThuyVu, T. and M. Tokunaga (2002). Designing of wavelet-based processing system for airborne laser scanner segmentation. *Proceedings of IAPRS and Spatial Information Science, February 2002 vol. 34*(5/W3), pp. 1682–1777.

Trias-Sanz, R. and N. Lomenie (2003). Automatic bridge detection in high-resolution satellite images. *Proceedings of Computer Vision Systems: Third International Conference, ICVS 2003 vol. 2626, April 1-3, Graz, Austria*, pp. 172–181.

Voegtle, T. and E. Steinle (2003). On the quality of object classification and automated building modelling based on laser scanning data. *Workshop on 3-D reconstruction from airborne laser scanner and InSAR data vol. 34*(3/W13, October 8-10, Dresden, Germany), pp. 149–155.

Vosselman, G. (2000). Slope based filtering of laser altimetry data. *IAPRS, WG III/3, Amsterdam, The Netherlands vol. 33*(B3), pp. 935–942.

Vosselman, G. and H. Maas (2001). Adjustment and filtering of raw laser altimetry data. *Proceedings of OEEPE workshop on airborne laser scanning and interferometric SAR for detailed Detailed Digital Elevation Models. March 1-3, Stockholm, Sweden. Official Publication No. 40. CD-ROM.*, 11 pages.

Wack, R. and A. Wimmer (2002). Digital terrain models from airborne laser scanner data - a grid based approach. *IAPRS vol. 34*(3B, September 9-13, Graz, Austria), pp. 293–296.

Wang, Y. and Q. Zheng (1998). Recognition of roads and bridges in sar images. *Pattern recognition vol. 31*(7), pp. 953–962.

Wehr, A. and U. Lohr (1999). Airborne laser scanning - an introduction and overview. *ISPRS JPRS vol. 54*, pp. 68 –82.

Wever, C. and J. Lindenberger (1999). Experiences of 10 years of laser scanning. *Photogrammetric Week, Fritsch/Spiller (Eds.)*, pp. 125 –132.

Zahn, C. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers vol. C-20*, pp. 68–86.

Zhan, Q., M. Molenaar, and K. Tempfli (2002). Building extraction from laser data by reasoning on image segments in elevation slices. *IAPRS vol. 34*(3B, September 9-13, Graz, Austria), pp. 305–308.

Zhang, Y. (1997). Evaluation and comparison of different segmentation algorithms. *Pattern Recognition Letters vol. 18*, pp. 963–974.

ACM	Association for Computing Machinery
ASPRS	American Society for Photogrammetry and Remote Sensing
CVPR	Conference on Computer Vision and Pattern Recognition
IAPRS	International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences
ICVS	International Conference on Computer Vision Systems
ISPRS	International Society for Photogrammetry and Remote Sensing
JPRS	ISPRS Journal of Photogrammetry and Remote Sensing
PE&RS	Photogrammetric Engineering & Remote Sensing

Appendix A

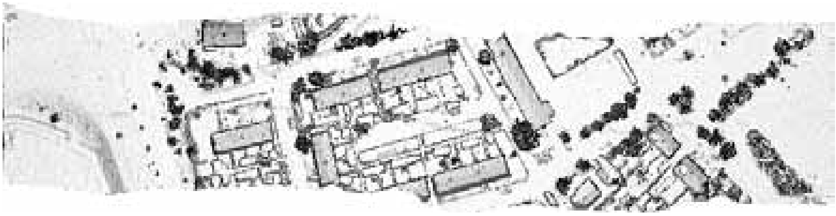
Analysis of point and segment attributes

A.1 Wijhe Point Attribute Characteristics

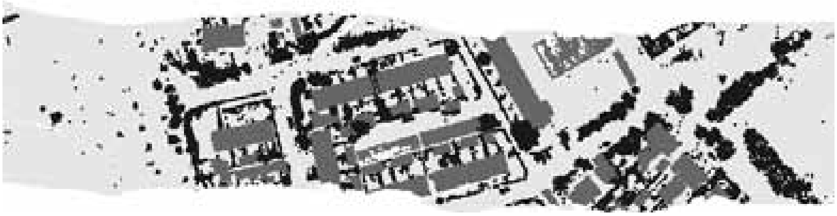
Because the Wijhe data was manually classified the distribution of bare earth, building and vegetation characteristics could be studied. Figure A.1 shows the distribution of the roughness and reflectance values in the data. In general the *bare earth* and *buildings* show weak roughness (darker points) while *vegetation* (lighter points) shows strong roughness. Mid-level roughness (light gray) is apparent on grass patches, slopes and roof apexes. The distinctions become even more apparent when the cloud is segmented and the roughness values aggregated for segments. On the strength of this observation, it can be said that roughness is a good measure for separating between vegetation and buildings.

For reflectance, the distinction between features is less apparent. Most *buildings* are mid-level gray as are most of the vegetation. However, a considerable amount of gray is also found in the *bare earth.* The reflectance does not appear to be a convincing attribute for separating between features. For this reason the scatter plots for the different classes were compared to determine the interaction between roughness and reflectance.

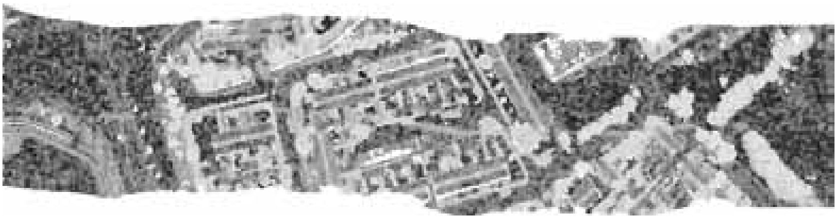
In figure A.2 building points are concentrated in a small cluster near the origin, indicating that buildings have small roughness and reflectance values. This is as expected because the roof of buildings are relatively smooth. Most roofs are also covered by the same material hence the reflectance values fall in a narrow



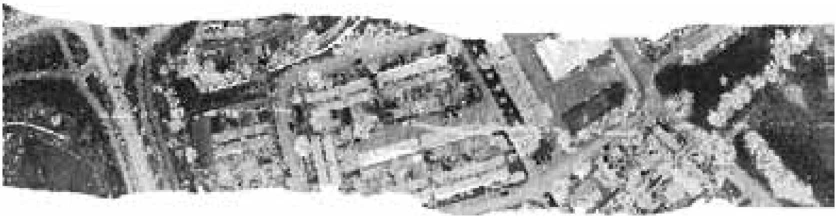
Shaded relief



Manual classes: bare earth (light gray), buildings (dark gray), vegetation (black)



Roughness (dark values < light values)



Reflectance (dark values < light values)

Figure A.1: Wijhe point characteristics.

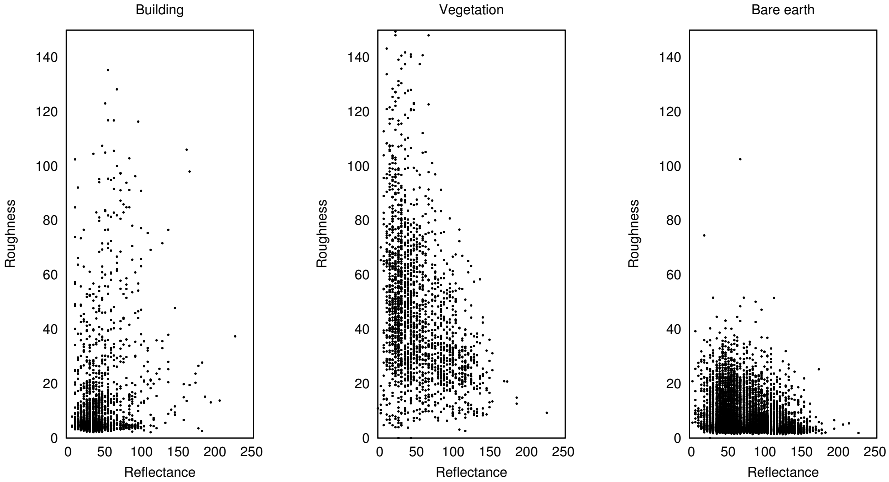


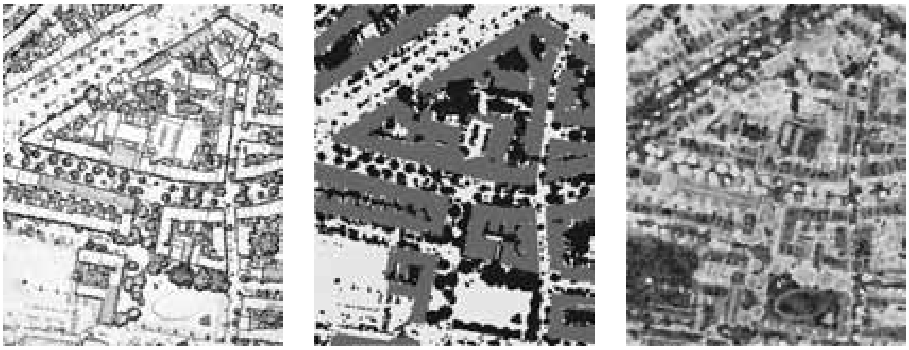
Figure A.2: Scatterplots - Wijhe.

banded. The outlying building points are from walls and roof apexes that have strong roughness.

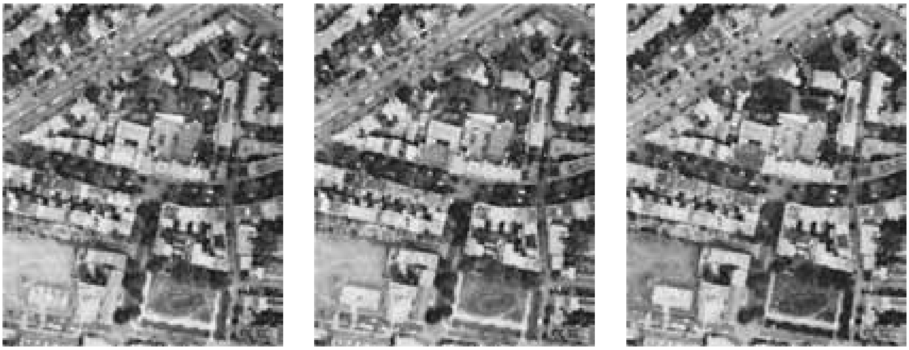
As expected vegetation shows a wider range of roughness and reflectance values, Because of the variation in the height, density and material content of vegetation (for convenience cars, lamp posts were included in the vegetation class). Importantly though the vegetation cluster does not appear to greatly overlap building cluster.

The bare earth cluster is less compact than the building cluster but more compact than the vegetation cluster. This is because the bare earth shares some qualities from both buildings and vegetation. The bare earth is covered by vegetation (e.g., grass), which gives it the radiometric qualities of vegetation. It also contains discontinuities, which give it the geometric characteristics of buildings.

Because the detection of the bare earth from objects is purely geometric, therefore, roughness offers a good means for separating between *buildings* and *vegetation.* The reflectance is not as strong a discriminator as the roughness.



Shaded relief Manual classes Roughness (dark values < light values)



Red (0-255) Green (0-255) Blue (0-255)

Figure A.3: Nijmegen data characteristics.

A.2 Nijmegen Point Attribute Characteristics

Unlike the Wijhe data, in the Nijmegen data, associated with each point is an RGB triplet instead of a reflectance value. Figure A.3 shows the distribution of the roughness and reflectance values in the data. As with the Wijhe data the *bare earth* and *buildings* show weak roughness (darker points) while *vegetation* (lighter points) shows strong roughness. Some buildings in the north of the data have a strong roughness. The cause of this is the narrowness of the buildings and the steepness of their roofs. This is because the neighborhood used in the roughness computation at each point will contain points from opposing roof facets. Some large trees in the south show small roughness because of their size. On such trees the roughness is smaller at their center (i.e. top) than at their edges.

For the RGB triplets the separation between the classes is less clear. While the vegetation generally appears dark and the buildings light, there are also exceptions.

In figure A.4 the roughness and RGB triplets are plotted against each other based on the classes *building* and *vegetation*. The buildings and vegetation in the scene have a yellowish tinge, which can be partly explained by the strong correlation between the red and green components. The exception to this is the orange roofed buildings. From the scatter plots it can be said that the red and green components are useful for separating the brightest buildings. For a better separation between buildings and vegetation the red-blue and green-blue component pairings offer better discrimination, because in these cases the vegetation clusters are more compact even though the components are still strongly correlated. The use of roughness improves the discrimination of buildings and vegetation. In particular, the roughness-blue pairing appears to separate the clusters well.

A classification of segments (after a segmentation) is intended to improve the discrimination of vegetation and buildings by reducing the mixing of the clusters. For example, if wall points can be included in building segments then the small roughness values of the many roof points will suppress the large roughness values of the wall points. This leads to the problem of combining the point characteristics to obtain a single value for each segment. Particularly with a view to improving the discriminating power of the geometric and radiometric characteristics. Figure A.5 shows some ways of obtaining a single value for each segment. To generate the figures segments were manually classified. Single values were obtained by taking the n^{th}-percentile, mean and standard deviation.

For the roughness the worst combination is the standard deviation. Because of the wall points buildings have a large standard deviation which causes the building and vegetation clusters to mix. Using the 10^{th} percentile suppresses the large roughness values and hence compressing the building and vegetation clusters. This compression also leads to shift of the clusters toward the origin. The vegetation cluster is shifted more than the building cluster and hence there is mixing. The reverse process happens when the 90^{th} percentile is used. The median and mean show little difference, but eventually the median was used to obtains single values because of its robustness to outliers.

Because the material characteristics of most objects are fairly uniform, the radiometric characteristics of object segments are more stable, in particular for vegetation segments. As can be seen the clusters do not shift dramatically between percentiles. Building clusters because of their greater material variability expand with higher percentiles. As with the roughness, the median was used to obtain a single RGB values for the surface segments.

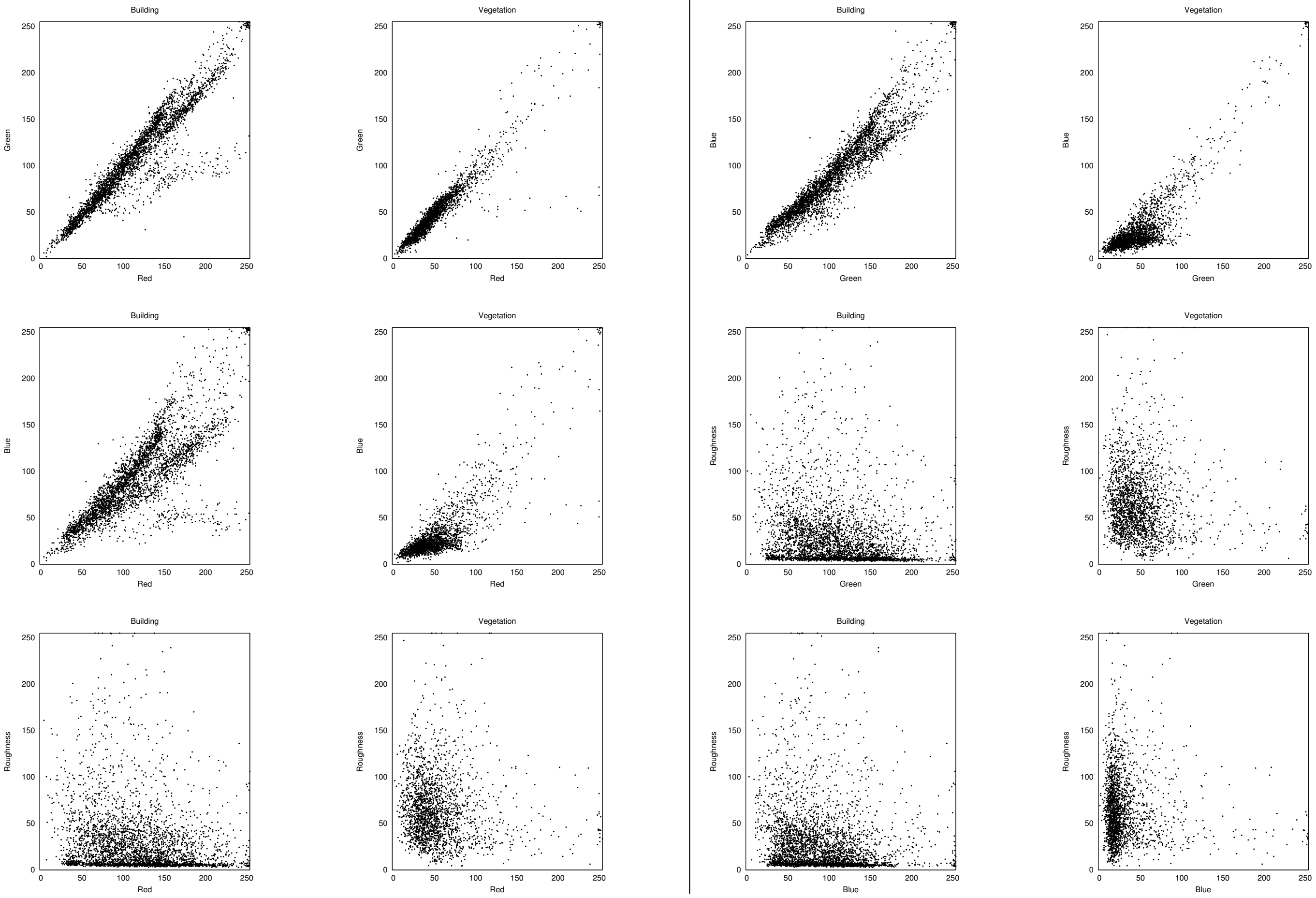


Figure A.4: Scatterplots - Nijmegen.

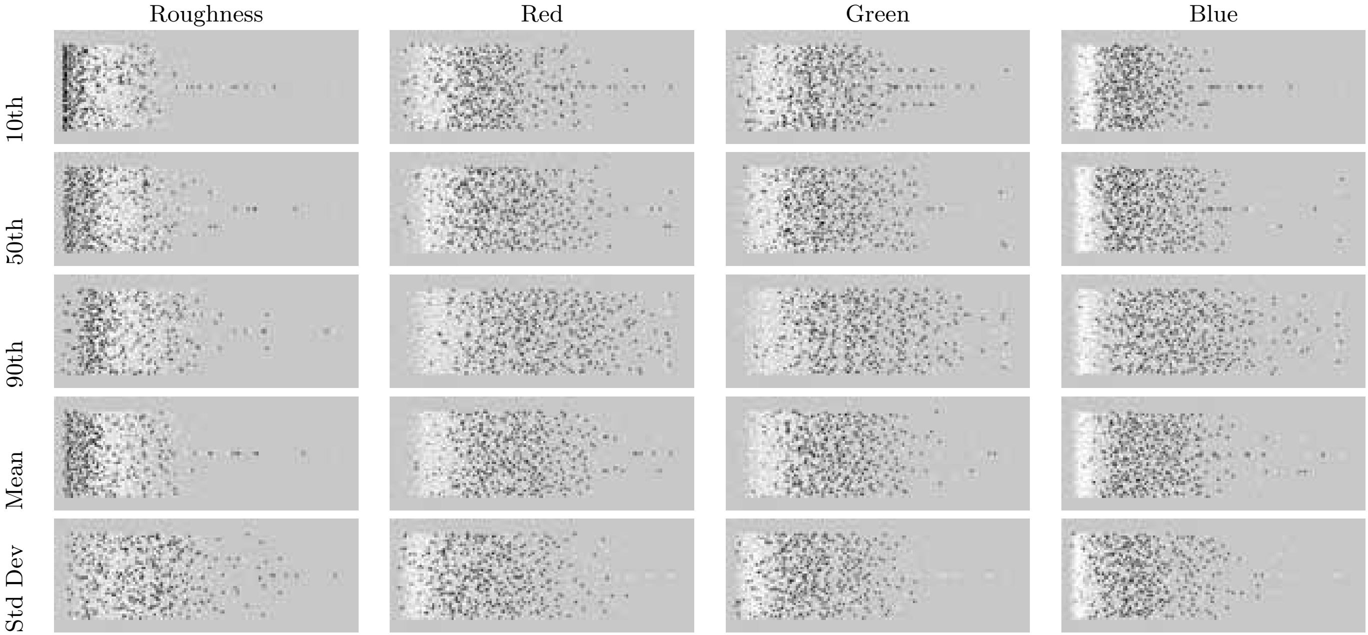


Figure A.5: 1D Plots for segment characteristics - Nijmegen. Values are jittered to emphasise the distribution. Buildings are shown in black and vegetation in white.

Appendix B

Algorithm Parameters

The purpose of this appendix is to explain how the parameters in the different segmentation and classification algorithms are chosen. Although a total of 37 parameters have to be set for all the algorithms combined it will be shown that most of these are related to the point spacing (e.g., segmentation parameters) and some others independent of the whatever point cloud is used (e.g., man made and natural object detection parameters). Therefore, during classification only a few parameters need to be set.

It will also be shown that the choice of values for the parameters is intuitive and easy for an operator to relate to the characteristics of a point cloud and its corresponding landscape.

B.1 Macro object detection parameters

Fifteen parameters are used in the macro object detection algorithm, see table B.1. Eight of these relate to the segmentation algorithm, and seven to the classification algorithm.

Segmentation parameters

The segmentation algorithm starts by profiling a point cloud in a given number of directions. Each profile is then further segmented to yield line segments. Surface segments are finally obtained by stitching together line segments that pass through the same point. The concept of the algorithm is elaborated in section 5.3. The parameters to be set relate to the thickness of the profiles, the rule used to segment the profiles (labeling rule), and the thresholds used in the labeling rules (height and proximity thresholds).

In the macro object detection the segmentation process is iterated several times (section 6.2, algorithm 6.1). After each iteration the profile range is incremented and the height (or proximity) threshold is decremented. In the macro object detection this iteration simulates the gradual stripping away of objects from the bare earth (starting with the most elevated and working down toward the bare earth). Typically, three iterations are required for a good classification, but it depends on the height threshold, and the profile range.

In the labeling rules too small a height threshold can result in over segmentation. For example, artifacts on roofs will be treated as separate objects but also small bare earth segments can be misclassified as object. Conversely too large a threshold leads to under segmentation, which can protract the macro object detection and necessitates more iterations. To offset the effects of over and under segmentation, in each iteration the threshold is decremented. This can be likened to removing ever smaller objects with each iteration. The optimal choice of a threshold has not been studied, but experiments have showed that typically a starting value equal to one third of the point spacing and a decrement of factor 0.8 works well.

The profile range controls the segmentation algorithms ability to span planimetric gaps in the data. The greater the range, the greater the gap spanned. Too small a range leads to over segmentation. Too large a range leads to under segmentation, as large gaps between objects are spanned. The choice of range depends on the resolution of the data and distance between objects in a landscape. Experiments have shown that in relatively flat landscapes the optimal range is about twice the point spacing. In steeper landscapes a smaller range is chosen to prevent objects from merging into the bare earth because of reduced lateral separations (see figure 4.3(c)). Like the height threshold, the range is varied with each iteration. As objects are stripped from the landscape, they leave gaps behind. Because of these gaps, to correctly determine the shape of new line and surface segments, small gaps have to be spanned. This requires the range to be increased in each iteration, hence the range increment. A range increment of 1.5 has been found to work well.

Line segment shapes and surface segment classification parameters

In figure 6.4(a) it was shown that an object segment is composed of both *raised* and *high* line segments. Similarly, a potential bare earth segment is composed of both *lowered* and *low* line segments. *Terraced* line segments can belong to either objects or the bare earth. To this can be added that line segments that have *no shape* are indeterminable. For this reason objects are not associated with segments that have no shape, i.e. $\beta_{object,noshape} = 0$ (see table B.1).

Segmentation

Parameter	Typical values	Comment
Profile width	$1 * \text{point spacing}$	The profile width
Labeling rule	PI-MST	Profile intersection (PI) by minimum spanning tree (MST)
Profile range	$2 * \text{point spacing}$	Largest planimetric separation between adjacent points in a line segment
Height Threshold	$1/3 * \text{point spacing}$	Largest height difference between two adjacent points in a line segment
Number of profiles	3	Number of profile directions
Iterations	3	Number iterations in the classification
Range increment	1.5	The factor by which the profile range is increased with each iteration
Threshold decrement	0.8	The factor by which the height threshold is decreased with each iteration

Line segment shapes and surface segment classification (see section 6.2)

Parameter	Typical values	Comment
raised	1	$\beta_{object,raised}$
lowered	0	$\beta_{object,lowered}$
terraced	0.5	$\beta_{object,terraced}$
low	0	$\beta_{object,low}$
high	1	$\beta_{object,high}$
no shape	0	$\beta_{object,noshape}$
ϵ_{object}	0.6	Threshold on the class grade $g(\phi(s) = object)$ (equation 6.4)

Table B.1: Algorithm Parameters: macro object detection

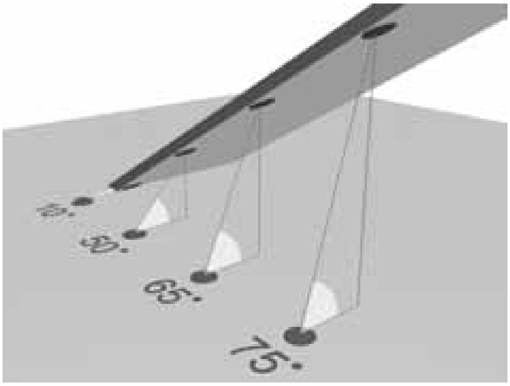


Figure B.1: The choice of slope threshold

B.2 Bridge detection parameters

Ten parameters are used in the bridge detection algorithm, see table B.2. Two, five, and three parameters are involved in the roughness computation, segmentation and classification respectively.

Roughness and discontinuity parameters

The purpose of roughness is to determine places where a bridge connects to the bare earth. Roughness is determined as the standard deviation of heights above and below a fitted plane. To obtain the roughness at a point a minimum area of the landscape around a point is sampled (K-nearest neighbours, KNN). For this reason the higher the resolution of the point cloud the greater the number of points that are sampled. The discrimination of a point as a possible bridge-bare earth connection is based on a fit threshold. Because bridge surfaces are generally smooth and their transition into the bare earth is shallow, this threshold is chosen to be small. In experiments, a value of 0.20m was chosen. This value may seem to be large, but it is chosen to avoid the detection being influenced by errors in the laser points measurements, which are in the order of 0.10-0.15m.

The purpose of the discontinuity parameter is to detect the presence and size of discontinuity around a point. It is determined as the largest absolute height difference in the neighborhood of a point (K-nearest neighbours, KNN). If the height difference is negative, the discontinuity is assigned a negative value.

Segmentation parameters

The segmentation is done using profile intersection with slope labeling. Figure B.1 shows the section of a bridge. Shown in the figure are four points running along the bridge and another four corresponding points along the base. Traveling up the bridge the slope between the base and the bridge edge is seen to increase. In the segmentation the choice of slope threshold determines where the bridge edge tears from the bare earth. In experiments a slope threshold of 45° has been found to work well.

Classification parameters

A bridge has been characterised by its width and its headroom. The design width is chosen based on the type of traffic that it will carry. A small bridge designed to carry a single motor vehicle will have a width of about 3m. Therefore, this value was chosen as the minimum width of a bridge. Motor vehicles have a height of about 2m or less. The headroom of a bridge has to greater than this. Here a minimum headroom of 3m was chosen.

B.3 Micro object detection parameters

The micro detection algorithm consists of only two parameters, table B.3. Firstly, the size of the point neighborhood for which roughness values are computed and secondly the threshold on the roughness. The same rationale is used in selecting the neighborhood size and threshold as in the bridge detection.

Roughness and Discontinuity

Parameter	Typical values	Comment
Roughness	10 nearest points	3D neighborhood
Discontinuity	5 nearest points	2D neighborhood - planimetric

Segmentation

Parameter	Typical values	Comment
Profile width	$1 *$ point spacing	The profile width
Labeling rule	PI-Slope	Profile intersection (PI) with consecutive slope labeling
Profile range	$2 *$ point spacing	Largest planimetric separation between adjacent points in a line segment
Slope Threshold	45°	Largest slope between two adjacent points in a line segment
Number of profiles	5	The number of profile directions

Classification

Parameter	Typical values	Comment
Fit threshold	0.20m	Smoothness threshold used when searching for bridge ends
Bridge height	3.00m	Minimum bridge height
Bridge width	3.00m	Minimum bridge width

Table B.2: Algorithm Parameters: bridge detection

Parameter	Typical values	Comment
Roughness	10 nearest points	3D neighborhood
Fit threshold	0.25m	Smoothness threshold used when searching for low object points

Table B.3: Algorithm Parameters: micro object detection

B.4 Man made and natural object detection parameters

Ten parameters are used in the man made and natural object detection algorithm, see table B.4. One, five, one, one and two parameters are involved in the islands classification, segmentation, roughness computation, main classification and walls classification respectively.

Island classification

Cars, lamp posts, clumps of vegetation and similar objects can be thought of as islands in a sea of bare earth. As explained already islands are treated as natural objects. Removing islands before proceeding to the main classification improves the final accuracy in the separation between man made and natural objects. To avoid classifying building blocks as islands, the maximum size of and island is limited to a few square meters. In experiments, using a maximum area of between 10 to $20m^2$ was found to work well. Depending on the resolution of a point cloud this equates to a few points.

Segmentation

There is strong cohesion between points from the same object. For this reason a proximity based segmentation is chosen. The choice of segmentation parameters has been explained previously and shall not be elaborated.

Roughness

Before the main classification can be done a roughness value has to computed for every point. This computation is done in the same manner as in the bridge detection algorithm, except this time a limit is placed on the size of the roughness values and the value is scaled.

$$roughness = \left\{ \begin{array}{ll} \sigma \geq 0.5 & 255.0 \\ \sigma < 0.5 & 255\sigma/0.5 \end{array} \right\} \tag{B.1}$$

Where σ is the standard deviation of residuals in a plane fit in the neighborhood of a point. The thresholding (in this case 0.5m) and scaling (0-255) is necessary for comparison with other attributes that are used in the feature space classification (i.e., reflectance or RGB). As desired the thresholded and scaled roughness is a measure that is deterministic for all landscapes.

Main classification

The main classification is done in an n-dimensional feature space. In the feature space a point is classified using the majority class of its K nearest neighbors. In experiments a $K \geq 5$ points was used successfully. For the experiments all available features were used, i.e., position (x,y,z), reflectance (if available) and RGB triplet (if available). Segments with fewer than 5 points were classified as unlabeled.

Walls and unlabeled points classification

After the main classification walls are detected. Building segments are partly distinguishable from vegetation segments in that they are smooth. This however does not hold true for wall segments (which posses strong roughness values). Similar to islands, walls are therefore treated separately to improve the accuracy of classification. In experiments walls in the point cloud were assumed to have a maximum depth of 0.15m[1].

The points from these segments and the wall points are now classified using the majority label of their nearest K points. In experiments the 10 nearest points were used.

---

[1] The fact that a maximum depth of 0.15m can be successfully used to detect wall points demonstrates that the relative planimetric accuracy of lidar points is much better than the published absolute accuracy of 0.5m

Islands classification

Parameter	Typical values	Comment
Max. island size	10 to $20m^2$	Object islands in the bare earth, e.g., small trees, bushes, cars, etc.,

Segmentation

Parameter	Typical values	Comment
Profile width	$1 *$ point spacing	The profile width
Labeling rule	PI-Proximity	Algorithm used for segmenting profiles
Profile range	$2 *$ point spacing	Largest planimetric separation between adjacent points in a line segment
Height Threshold	$1/3 *$ point spacing	Largest height difference or slope between two adjacent points in a line segment
Number of profiles	3	The number of profile directions

Roughness

Parameter	Typical values	Comment
Roughness	10 nearest points	3D neighborhood

Main Classification

Parameter	Typical values	Comment
Partitioning scheme	KNN	Feature space partitioning scheme K nearest neighbours

Walls and unlabeled points classification

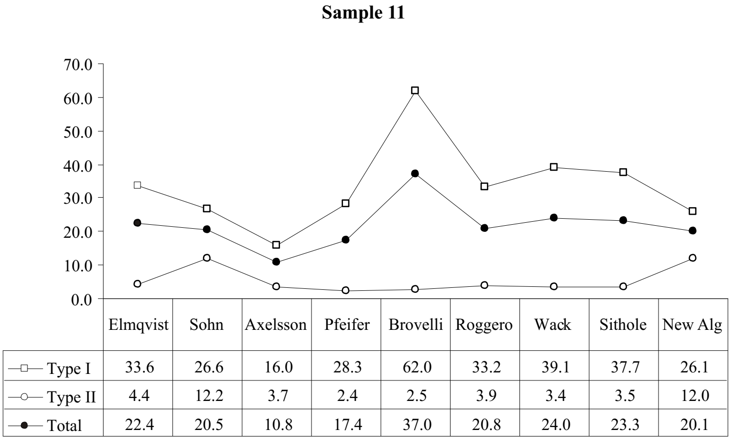
Parameter	Typical values	Comment
Wall threshold	0.15m	Maximum depth of a wall. Threshold on the smallest eigenvalue for a planimetric neighborhood of points
KNN (Unlabeled)	10 nearest points	Number of points in the neighborhood of an unlabeled point

Table B.4: Algorithm Parameters: man made and natural object detection

Appendix C

ISPRS Filter Test Results

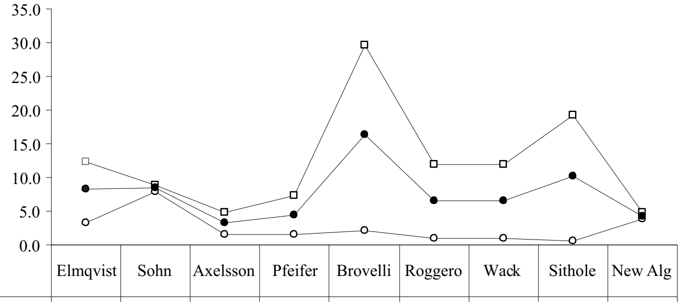
In section 7.2 the performance of the new algorithm was compared against three other algorithms studied in the ISPRS test. This appendix presents a comparison of the performance of the new algorithm against all the algorithms studied in the ISPRS test. The comparison is done over the 15 samples used in the test. A description of the algorithms can be found in section 2.3.



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	33.6	26.6	16.0	28.3	62.0	33.2	39.1	37.7	26.1
—○— Type II	4.4	12.2	3.7	2.4	2.5	3.9	3.4	3.5	12.0
—●— Total	22.4	20.5	10.8	17.4	37.0	20.8	24.0	23.3	20.1

	Bare earth	Object	Total	Error
Bare earth	16104	5682	21786	26.1%
Object	1940	14284	16224	12.0%
Total	18044	19966	38010	20.1%

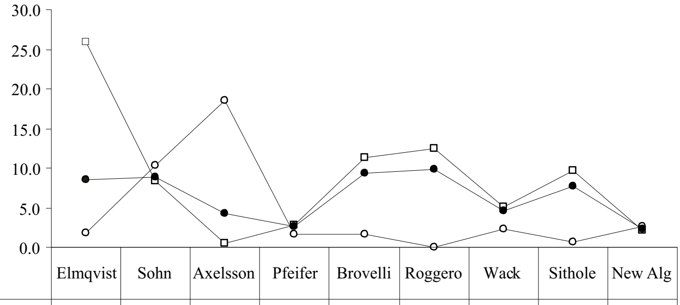
Sample 12



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
□ Type I	12.4	8.9	4.9	7.3	29.6	11.9	11.9	19.2	4.8
○ Type II	3.3	7.9	1.5	1.5	2.0	0.9	0.9	0.6	3.8
● Total	8.2	8.4	3.3	4.5	16.3	6.6	6.6	10.2	4.3

	Bare earth	Object	Total	Error
Bare earth	25420	1271	26691	4.8%
Object	972	24456	25428	3.8%
Total	26392	25727	52119	4.3%

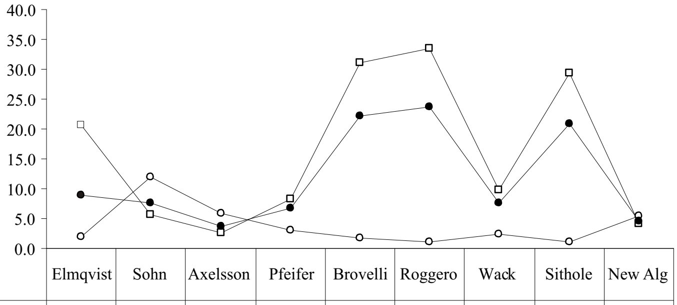
Sample 21



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
□ Type I	25.9	8.4	0.5	2.8	11.4	12.5	5.2	9.6	2.2
○ Type II	1.8	10.4	18.5	1.6	1.6	0.0	2.3	0.7	2.6
● Total	8.5	8.8	4.3	2.6	9.3	9.8	4.6	7.8	2.3

	Bare earth	Object	Total	Error
Bare earth	9877	221	10098	2.2%
Object	73	2789	2862	2.6%
Total	9950	3010	12960	2.3%

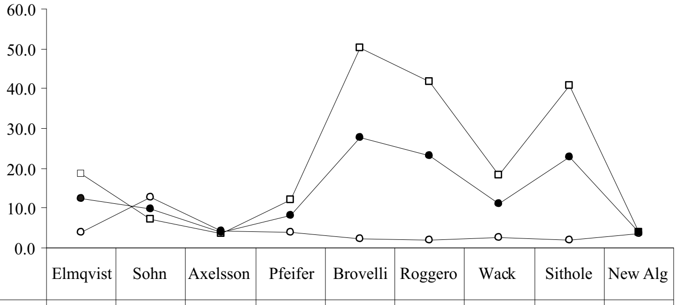
Sample 22



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
□ Type I	20.6	5.7	2.7	8.3	31.2	33.4	9.7	29.3	4.2
○ Type II	1.9	11.9	5.9	3.1	1.8	1.0	2.4	1.0	5.4
● Total	8.9	7.5	3.6	6.7	22.3	23.8	7.5	20.9	4.6

	Bare earth	Object	Total	Error
Bare earth	21134	923	22057	4.2%
Object	579	10070	10649	5.4%
Total	21713	10993	32706	4.6%

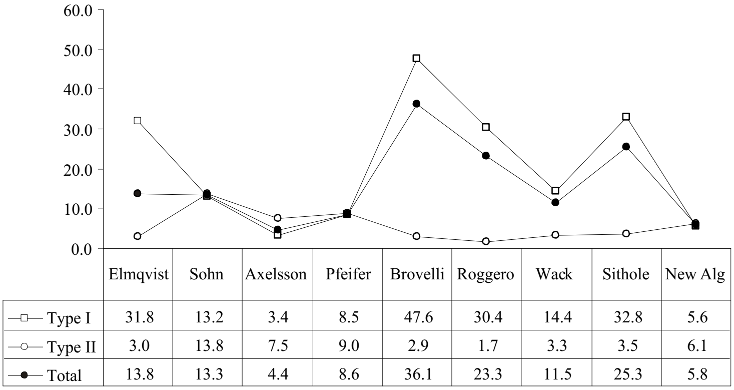
Sample 23



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
□ Type I	18.7	7.3	3.7	12.1	50.3	41.9	18.4	40.9	3.9
○ Type II	4.0	12.8	4.3	3.8	2.4	1.9	2.6	2.1	3.7
● Total	12.3	9.8	4.0	8.2	27.8	23.2	11.0	22.7	3.8

	Bare earth	Object	Total	Error
Bare earth	13024	525	13549	3.9%
Object	423	11120	11543	3.7%
Total	13447	11645	25092	3.8%

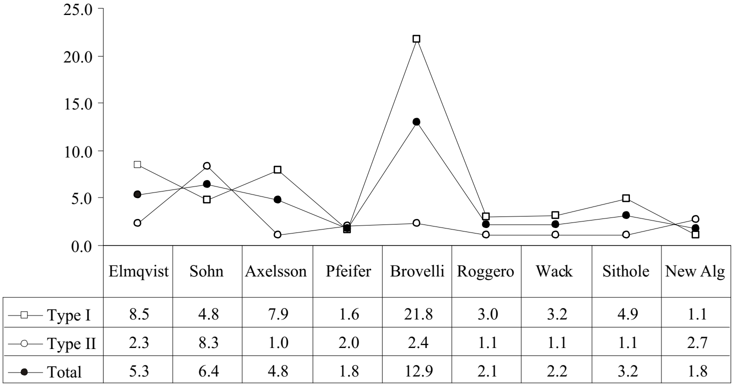
Sample 24



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	31.8	13.2	3.4	8.5	47.6	30.4	14.4	32.8	5.6
—○— Type II	3.0	13.8	7.5	9.0	2.9	1.7	3.3	3.5	6.1
—●— Total	13.8	13.3	4.4	8.6	36.1	23.3	11.5	25.3	5.8

	Bare earth	Object	Total	Error
Bare earth	5127	307	5434	5.6%
Object	125	1933	2058	6.1%
Total	5252	2240	7492	5.8%

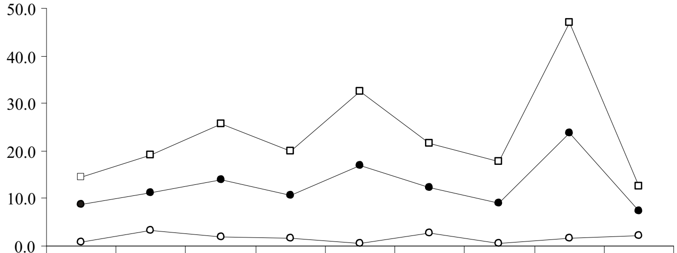
Sample 31



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	8.5	4.8	7.9	1.6	21.8	3.0	3.2	4.9	1.1
—○— Type II	2.3	8.3	1.0	2.0	2.4	1.1	1.1	1.1	2.7
—●— Total	5.3	6.4	4.8	1.8	12.9	2.1	2.2	3.2	1.8

	Bare earth	Object	Total	Error
Bare earth	15392	164	15556	1.1%
Object	357	12949	13306	2.7%
Total	15749	13113	28862	1.8%

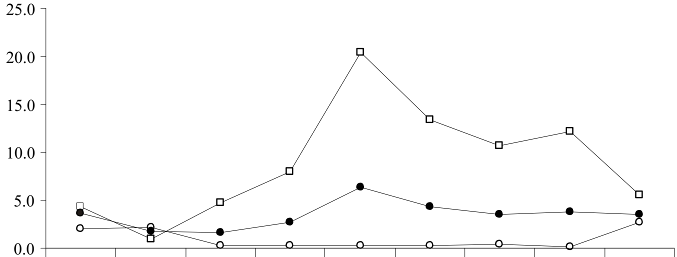
Sample 41



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	14.4	19.3	25.8	19.9	32.4	21.6	17.6	47.1	12.5
—○— Type II	0.9	3.2	1.9	1.6	0.5	2.7	0.5	1.7	2.2
—●— Total	8.8	11.3	13.9	10.8	17.0	12.2	9.0	23.7	7.4

	Bare earth	Object	Total	Error
Bare earth	4901	701	5602	12.5%
Object	125	5504	5629	2.2%
Total	5026	6205	11231	7.4%

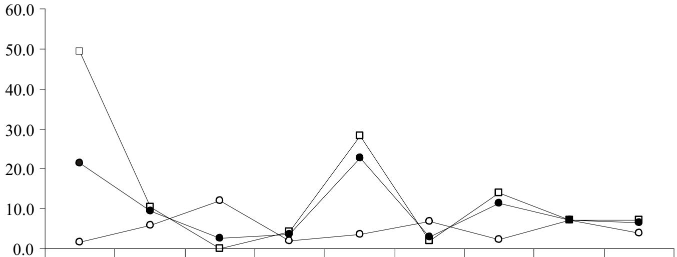
Sample 42



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	4.3	1.0	4.7	8.0	20.4	13.4	10.7	12.2	5.5
—○— Type II	2.0	2.1	0.3	0.2	0.3	0.3	0.4	0.2	2.7
—●— Total	3.7	1.8	1.6	2.6	6.4	4.3	3.5	3.9	3.5

	Bare earth	Object	Total	Error
Bare earth	11761	682	12443	5.5%
Object	804	29223	30027	2.7%
Total	12565	29905	42470	3.5%

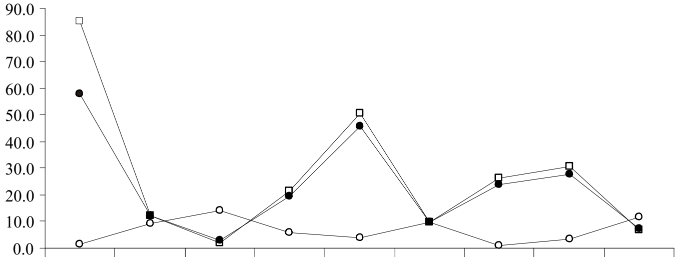
Sample 51



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	49.3	10.3	0.1	4.2	28.2	1.9	14.0	7.0	7.1
—○— Type II	1.6	5.7	12.0	1.9	3.6	7.0	2.2	7.0	3.7
—●— Total	21.3	9.3	2.7	3.7	22.8	3.0	11.5	7.0	6.4

	Bare earth	Object	Total	Error
Bare earth	12955	995	13950	7.1%
Object	146	3749	3895	3.7%
Total	13101	4744	17845	6.4%

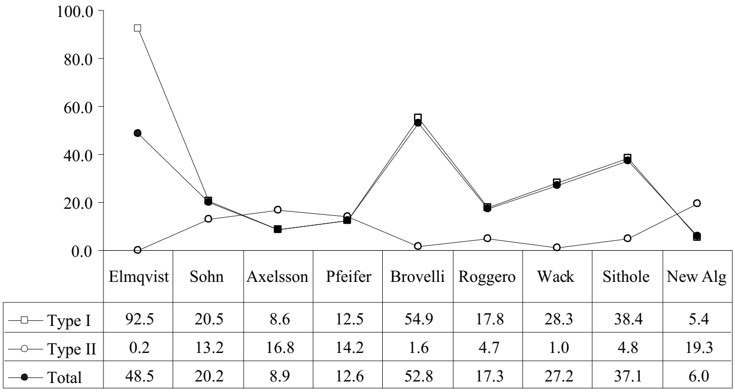
Sample 52



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	85.1	12.3	1.8	21.3	50.4	9.8	26.5	30.4	6.8
—○— Type II	1.3	9.5	14.2	5.7	3.8	9.7	1.0	3.6	11.8
—●— Total	58.0	12.0	3.1	19.6	45.6	9.8	23.8	27.5	7.3

	Bare earth	Object	Total	Error
Bare earth	18796	1366	20162	6.8%
Object	272	2040	2312	11.3%
Total	19068	3406	22474	7.3%

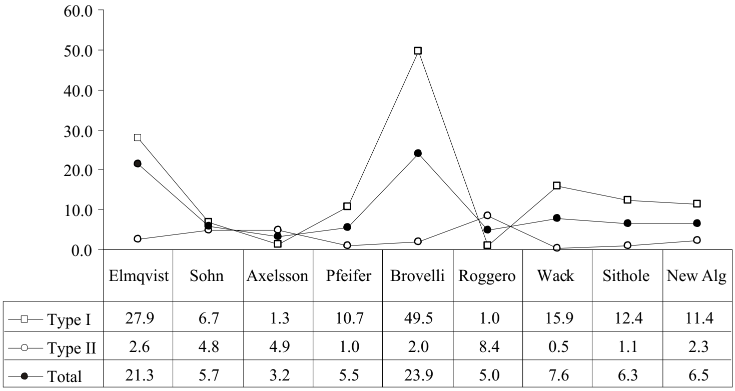
Sample 53



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	92.5	20.5	8.6	12.5	54.9	17.8	28.3	38.4	5.4
—○— Type II	0.2	13.2	16.8	14.2	1.6	4.7	1.0	4.8	19.3
—●— Total	48.5	20.2	8.9	12.6	52.8	17.3	27.2	37.1	6.0

	Bare earth	Object	Total	Error
Bare earth	31196	1793	32989	5.4%
Object	268	1121	1389	19.3%
Total	31464	2914	34378	6.0%

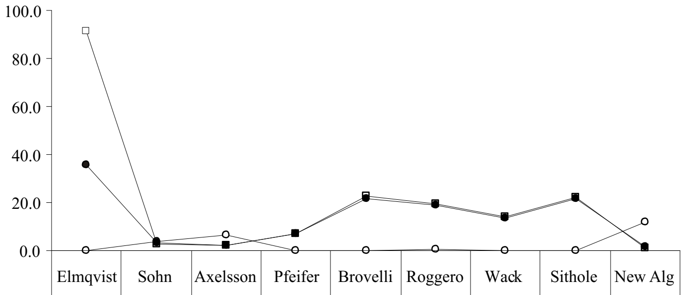
Sample 54



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	27.9	6.7	1.3	10.7	49.5	1.0	15.9	12.4	11.4
—○— Type II	2.6	4.8	4.9	1.0	2.0	8.4	0.5	1.1	2.3
—●— Total	21.3	5.7	3.2	5.5	23.9	5.0	7.6	6.3	6.5

	Bare earth	Object	Total	Error
Bare earth	3527	456	3983	11.4%
Object	107	4518	4625	2.3%
Total	3634	4974	8608	6.5%

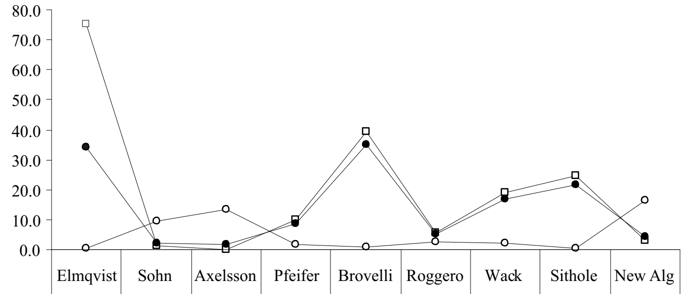
Sample 61



	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	91.3	3.0	1.9	7.2	22.5	19.6	13.9	22.4	1.0
—○— Type II	0.1	3.7	6.2	0.2	0.0	0.6	0.3	0.3	12.1
—●— Total	35.9	3.0	2.1	6.9	21.7	19.0	13.5	21.6	1.4

	Bare earth	Object	Total	Error
Bare earth	33518	336	33854	1.0%
Object	146	1060	1206	12.1%
Total	33664	1396	35060	1.4%

Sample 71



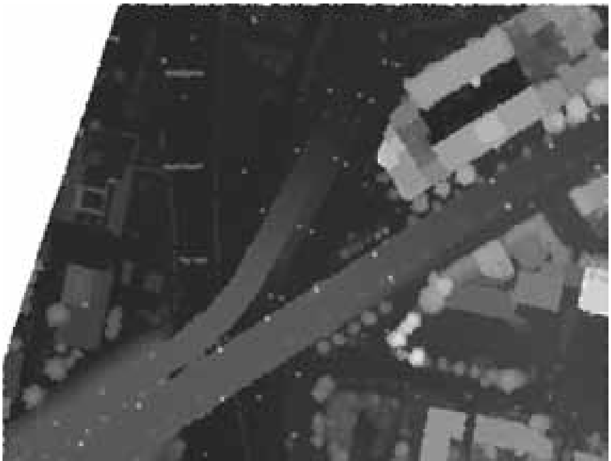
	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	New Alg
—□— Type I	75.2	1.3	0.1	9.8	39.4	5.4	18.9	24.6	3.0
—○— Type II	0.2	9.5	13.3	1.6	0.8	2.8	2.0	0.3	16.3
—●— Total	34.2	2.2	1.6	8.9	35.0	5.1	17.0	21.8	4.5

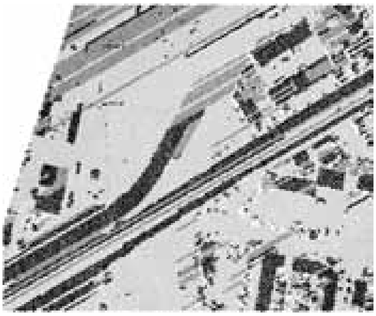
	Bare earth	Object	Total	Error
Bare earth	13460	415	13875	3.0%
Object	288	1482	1770	16.3%
Total	13748	1897	15645	4.5%

Appendix D

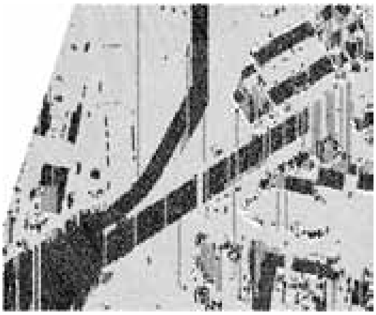
Example of macro object and bridge detection

The purpose of this appendix is to provide a step-by-step example of a macro object and bridge detection. The example uses a cloud of about 130000 points. The point cloud is that of an urban landscape in which there are large buildings, vegetation (both low and high), traffic, and two overpasses. In the figure below the point cloud is colored by height, black is low and white is high.

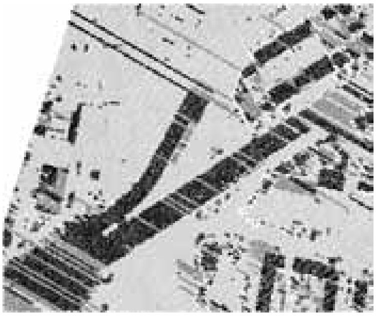




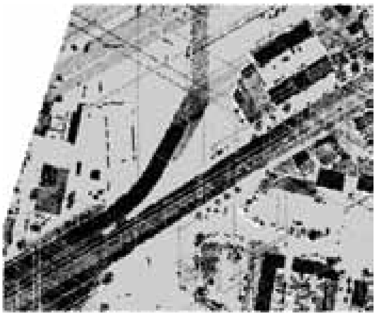
The point cloud is profiled. This figure shows the profiles at an orientation of 30°. Once done each profile is segmented (in this case using a minimum spanning tree labeling). After the profiles have been segmented the shape of each line segment is determined.



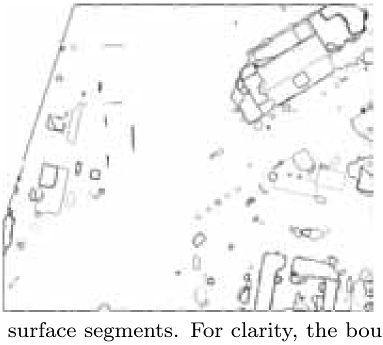
Raised and high line segments are shown in black, lowered and low line segments are shown in light gray and terraced and no shape line segments are shown in dark gray.
The point cloud is profiled in several directions. This figure shows the profiles at 90°.



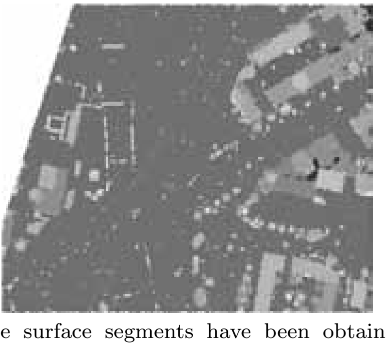
This figure shows the profiles at 150°.



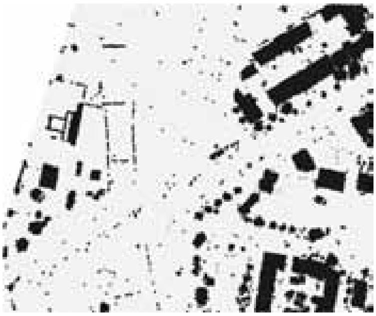
Once all profiles have been segmented they are overlaid. Line segments that have points in common are linked. This linking is continued until surface segments are obtained. As can be seen objects like buildings are conspicuous by the fact that they contain many raised and high line segments.



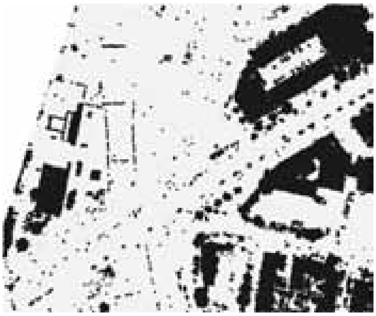
The surface segments. For clarity, the boundaries of the segments are shown.



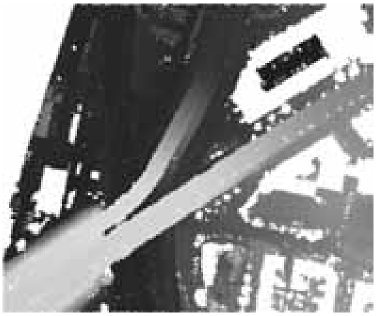
Once surface segments have been obtained, class grades are computed for them. Surfaces with a light color have a higher class grade.



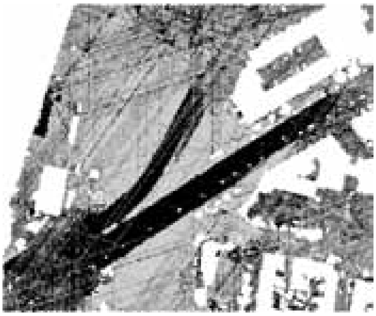
The class grades are thresholded. All surfaces with a class grade greater than 0.6 are labeled as object (shown in black). The remaining surfaces are labeled as possible bare earth.
The first classification does not catch all objects.



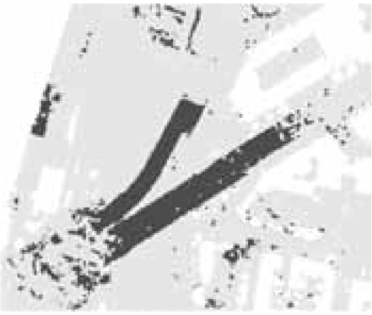
Therefore, the detected objects are removed, the shape of the line segments recomputed and new class grades determined for the remaining surface segments. The class grades are again thresholded. This process is repeated several times. The figure above shows the result after the third repetition.



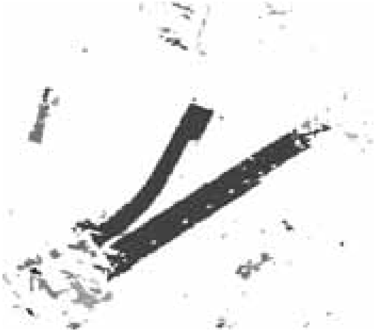
After the macro objects in the landscape have been identified, they are removed from the point cloud (colored by height). This is necessary for the bridge detection, recalling that bridges are detected in the bare earth.



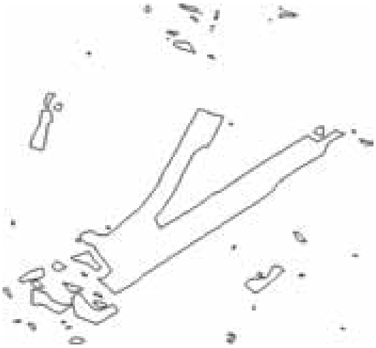
The bare earth point cloud is segmented by profile intersection with slope labeling. In the case above, five directions are used. It will be noticed that on the bridge there are many raised line segments (shown in black).



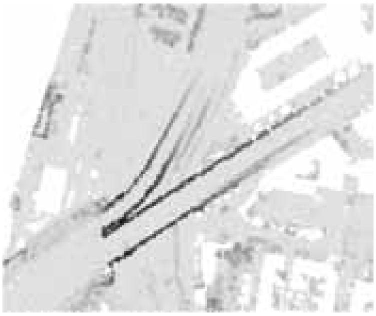
Points that have a majority of raised line segments passing through them are isolated. These are seed bridge points.



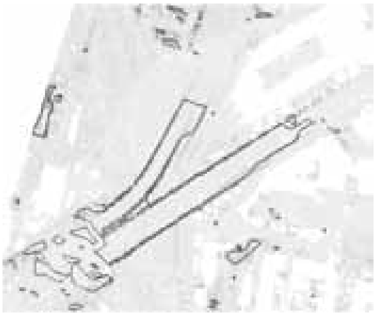
The isolated seed points are segmented by proximity (3D).



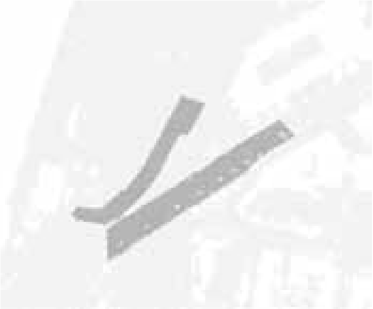
Boundaries are determined for the bridge segments.



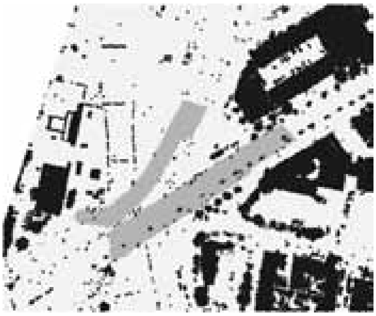
Roughness values are computed for each point in the cloud. Roughness is determine by fitting planes to point neighborhoods and determining the standard deviation of the distance of points from the fitted planes. In the figure above points with a high roughness are shown in dark gray.



The bridge boundaries are now trimmed (not shown in the figure) using the roughness values. Points that have a small roughness value are removed from the boundaries. In this way the edges along the length of the bridge and across the length of the bridge are identified.



If a bridge segment boundary contains more than one across edge it is labeled as bridge. In the figure above the two overpasses have been identified.



After the bridges are identified the macro objects (detected earlier) are put back into the point cloud.

Curriculum Vitae

George Sithole was born in 1968 in the city of Tehran, Iran. In 1989 he enrolled at the University of Zimbabwe to do a BSc degree in Surveying. After four years he graduated in 1992 as the best student in his year, and immediately joined a land survey practice (Saich Land Surveyors). Over the next two years he worked as a land surveyor in training on a variety of cadastral and engineering surveys. At the beginning of 1995 he earned his professional land surveyors licence.

At about the same time he was approached by his alma mater with an offer to return to them as a staff development fellow. In 1995 he joined the University of Zimbabwe assisting with service courses and practicals, and administering the departmental computer systems. In 1996 he received a scholarship to undertake a Master of Science degree in Integrated Geoinformation and Map Production at the International Institute for Geo-Information Science and Earth Observation (ITC) in Enschede (The Netherlands). Graduating with distinction in late 1997 (with a Master's thesis entitled "Digital photogrammetry for automatic photo-texture extraction") he returned to the University of Zimbabwe as a lecturer in GIS, Photogrammetry, and Computer applications.

In 2001 an opportunity arose to do Ph.D. research in the filtering of airborne laser scanner data with the Photogrammetry and Remote Sensing group at the Delft University of Technology. The culmination of this research is the material presented in this thesis.