AUTOMATIC GENERATION OF RASTER-BASED HEIGHT DATA FOR THE NETHERLANDS BASED ON THE AHN2 DATA SET

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of

Master of Science in Geomatics

by

Kees Jonker

January 2016

Kees Jonker: *Automatic generation of raster-based height data for the Netherlands based on the AHN2 data set* (2016)
This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit
http://creativecommons.org/licenses/by/4.0/.

The work in this thesis was made in the:



3D geoinformation group
Department of Urbanism
Faculty of Architecture & the Built Environment
Delft University of Technology

Supervisors:	Prof. dr. Jantien Stoter
	Dr. Hugo Ledoux
Co-reader:	Dr. ir. Ben Gorte

ABSTRACT

The recent emergence of Light Detection And Ranging (LiDAR) scanning technology has resulted in the availability of very large three dimensional point cloud data sets. These LiDAR data sets have become a main source for the modeling and reconstruction of both ground surface as well as above-ground objects such as buildings and vegetation.

For the Netherlands, the second version of the Algemeen Hoogtebestand Nederland (AHN) is a country-wide high-resolution point cloud data set comprising the nation's terrain by measuring heights, obtained by LiDAR, using Airborne Laser Scanning (ALS). None of the currently available raster-based products based on the AHN2 data set can be considered as being good: height maps contain holes, unintentional dynamic objects are present and more. Causes for errors can be found in deviations during the collection of the point cloud data as well as the applied methodology to process the data.

This thesis will identify the possibilities to improve the quality of raster-based height map products based on the AHN2 data set with respect to currently available products. Quality will be determined with respect to the *Geographic Information Quality principles* standard, introduced by the International Organization for Standardization (ISO).

A five-step methodology is proposed in order to generate raster-based height data from massive LiDAR point cloud samples from the AHN2 data set. In the first step, the massive LiDAR point cloud data will be split up in overlapping tiles in order to pipeline subsets of data in order to feed data sequentially to the computers' main memory. In the second step filtering of different classes takes place in order to filter specific information within the point cloud data. In the third step, for each filtered class specific interpolation methods will be applied in order to achieve raster-based data from the point cloud data that fits best for a certain class of data. In the fourth step some post-processing steps will be applied in order to optimize the raster data and a composition of the tiles that were decomposed in the first step. In the fifth step raster visualization will be applied in order to support visual inspection of the data.

Quality assessment shows that the methodology proposed within this thesis is capable to process data with a completeness rate of 100%. Positional accuracy is determined with respect to currently available raster-based height maps since no reference data is available. For Digital Elevation Model (DEM)s low positional errors are measured where for Digital Surface Model (DSM)s higher positional errors are measured due to erroneous data within current DSM products. Thematic accuracy is defined for a larger amount of classes in comparison with currently available raster-based height maps with a sensitivity rate between 56$\sim$100% $\pm$ 5%.

ACKNOWLEDGEMENTS

As first I want to thank my supervisor Hugo Ledoux. You did not only advice me to postpone my graduation up to two times in order to let me reach the current level of quality of my work; your help and advices were always helpful at the moments I took a wrong exit, which happened multiple times. I think no other supervisor within the department of Geomatics would have been able to stimulate me in doing my work as you did. In the year you were my supervisor you inspired me a number of times with your view on the world, also sometimes when the topic of the conversation was not relevant for my research.

As second, many thanks to Jantien Stoter for being my graduation professor, despite we did not see each other that much, your comments were helpful and always stimulating me to focus on what I was doing exactly. Also many thanks to Ben Gorte for being willing to become my co-reader.

I want to thank Rijkswaterstaat for providing me the high resolution photography. Some of the employees support the concept of open data more than the official restrictions that exist within this governmental department. Also many thanks to Henk Kramer from Alterra for providing me samples of the OHN data set which is used for quality assessment and also for his view on dealing with the same problems that I had while writing this thesis.

I want to thank the fellow Geomatics students, my friends and my loved one for always supporting me and telling me many times that I am really capable to finish my thesis. I also want to thank my colleagues from Stadsradio Delft, the local radio station of Delft. My colleagues told me many times that finishing my thesis is way more important than investing my time in the radio.

Last but not least I want to thank my parents. Despite they did not understand what my research was about at all, they always tried to support me in many ways.

CONTENTS

1 INTRODUCTION 1
- 1.1 Motivation . . . 1
- 1.2 Research question . . . 3
- 1.3 Scope . . . 4
- 1.4 Outline . . . 4

2 STATE OF THE ART IN AHN2 PRODUCT DEVELOPMENT 5
- 2.1 Digital terrain modeling . . . 5
- 2.2 LiDAR . . . 7
- 2.3 The AHN2 data set . . . 9
- 2.4 Existing products . . . 12

3 RELATED WORK 21
- 3.1 Related projects . . . 22
- 3.2 Pipeline generation . . . 23
- 3.3 Filtering . . . 26
- 3.4 Spatial interpolation . . . 30
- 3.5 Post processing . . . 38
- 3.6 Raster visualization . . . 39
- 3.7 Quality assessment . . . 42
- 3.8 Discussion . . . 44

4 METHODOLOGY 45
- 4.1 Pipelining . . . 46
- 4.2 Filtering . . . 48
- 4.3 Spatial interpolation . . . 56
- 4.4 Post processing . . . 74
- 4.5 Raster visualization . . . 78

5 IMPLEMENTATION & RESULTS 81
- 5.1 Implementation . . . 81
- 5.2 Test data sets . . . 81
- 5.3 Validation of individual classes . . . 84
- 5.4 Quality assessment . . . 93
- 5.5 Evaluation . . . 101

6 CONCLUSION, DISCUSSION AND FUTURE WORK 103
- 6.1 Conclusion . . . 103
- 6.2 Discussion . . . 106
- 6.3 Future work . . . 107

A REFLECTION 117

B OPEN 2D GEODATA SETS IN THE NETHERLANDS 119

C POINT CLOUD ANALYSIS 123

D THEMATIC ACCURACY 127

LIST OF FIGURES

Figure 1 Digital Surface Model [Tukay Mapping] 1
Figure 2 Faculty of Architecture and the build Environment, Delft 3
Figure 3 A digital terrain model stored as a triangular irregular network [Pudelko, 2007]. 6
Figure 4 The difference between a DSM, DEM and a nDSM. . 7
Figure 5 Visualization of two LiDAR collection concepts . . . 8
Figure 6 Example of a public header block and Variable Length Records information of an AHN2 tile 9
Figure 7 Visualization of AHN2 point cloud products 11
Figure 8 Tiles of the AHN2 data set covering the Netherlands [PDOK, 2014] 11
Figure 9 Difference between the not-filled and filled digital elevation models 12
Figure 10 Inverse distance weighting as applied for the generation of raster-based height maps from AHN2 point cloud data [Rijkswaterstaat Meetkundige Dienst]. . . 14
Figure 11 Errors in PDOK digital elevation model 14
Figure 12 Errors in PDOK digital elevation model 14
Figure 13 Errors in PDOK digital surface model 15
Figure 14 Interpolation based on a TIN 16
Figure 15 ESRI raster-based height maps 16
Figure 16 Wrongly filling of no-data Raster cells 18
Figure 17 Dynamic Holland Shading Map (own image, based on Geodan [2014]) 19
Figure 18 Work flow for the generation of a raster-based height map from massive point cloud data. 21
Figure 19 Danish raster-based height map 24
Figure 20 Object hierarchy [Hug et al., 2004] 29
Figure 21 The influence of different power values regarding inverse distance weighting 32
Figure 22 The concept of (constrained) Delaunay triangulation 34
Figure 23 Digital terrain model generated by interpolation based on a triangulated irregular network 34
Figure 24 A regular grid overlain on a Delaunay surface to produce a raster file of height values [Pearlstone, 2010] . 35
Figure 25 Hill shading 40
Figure 26 Low pass pyramid with four levels for an image of 4 096 x 4 096 pixels. The tile size is 256 x 256 pixels [McInerney and Kempeneers, 2015]. 41
Figure 27 Work flow for the processing methodology as introduced in this chapter. 45
Figure 28 Decomposition and interpolation of overlapping point cloud tiles [Guan and Wu, 2010]. 47
Figure 29 Points classified as non-ground points with LASground 50
Figure 30 Points classified as non-ground points after application of the LASground tool 50

Figure 31 A digital elevation model stored as a triangular irregular network containing classified ground point records. A number of point records are wrongly classified as ground while being reflected on buildings. . 51
Figure 32 Points classified as building points by using the LASclassify tool 53
Figure 33 Correlation between the standard deviation and the amount of LiDAR points being classified as building for a sample data set. 53
Figure 34 Points classified as vegetation points by using the LASclassify tool 55
Figure 35 Correlation between the standard deviation and the amount of LiDAR points being classified as vegetation for a sample data set. 55
Figure 36 Influence of different cut-off threshold values after rasterization a triangular irregular network constructed from point records classified as ground 58
Figure 37 Correlation between sparse point record distributions and the presence of building footprints and water bodies 59
Figure 38 Calculating a slope-based raster file from a digital elevation model 60
Figure 39 Visualization of disjoint polygons after the application of multiple concavity values using the LASboundary tool 65
Figure 40 Adding holes within the interior of a building polygon with the LASboundary tool 66
Figure 41 Gathering years of the AHN2 point cloud data [Van der Zon, 2011] 67
Figure 42 Visual comparison of different methods of building boundary extraction 68
Figure 43 The potential of point records classified as noise during the generation of a digital building model. (a) Aerial photograph 69
Figure 44 Point cloud manipulation 70
Figure 45 Comparison of interpolated point cloud products applying interpolation based on a triangular irregular network. 71
Figure 46 Comparison of raster data of different digital building models 71
Figure 47 Diagram of a pit-free algorithm methodology [Khosravipour et al., 2014]. 72
Figure 48 Canopy height model generation 73
Figure 49 Digital elevation model generated by gridded interpolation based on a triangular irregular network . . . 74
Figure 50 Indirect interpolation. 75
Figure 51 Difference between directly interpolated raster data and indirect interpolated data after resampling. . . . 76
Figure 52 Distribution of height differences between direct interpolation and indirect interpolation using different resampling methods. 76
Figure 53 Virtual raster generation 78

Figure 54 Calculating a slope-based raster file out of a digital elevation model . . . 78
Figure 55 Calculating a slope-based raster file out of a digital elevation model . . . 79
Figure 56 Aerial photographs covering the test data set areas . 82
Figure 57 Digital elevation models . . . 83
Figure 58 Digital surface models . . . 83
Figure 59 Height differences between PDOK raster-based height data and the digital elevation model generated according to the methodology described in this thesis . 84
Figure 60 Visualization of the areas covered by the test data sets 85
Figure 61 Detected water bodies . . . 86
Figure 62 Falsely detected water bodies . . . 86
Figure 63 Filling building footprints and local deviations within a digital elevation model . . . 87
Figure 64 Filling of buildings within a digital surface model . . 88
Figure 65 Missing building data within a digital building model 89
Figure 66 Absent building data within a digital building model 89
Figure 67 Filling of no-data areas within vegetation . . . 90
Figure 68 Height differences between the raster-based height data of PDOK and the canopy height model generated according to the methodology described in this thesis. . . . 91
Figure 69 Electricity poles and cables within a canopy height model . . . 91
Figure 70 Digital elevation model generated with point cloud data from four input tiles. . . . 92
Figure 71 Errors on the edge between sub-projects . . . 92
Figure 72 Digital surface model generation . . . 93
Figure 73 Vector-based map based on the *TOP10NL* data set. . 120
Figure 74 Comparrison of TOP10NL and BGT . . . 122
Figure 75 Comparrison of TOP10NL and BAG . . . 122
Figure 76 Point density of the filtered AHN2 point cloud product 123
Figure 77 Histogram of the average point density per square meter of the filtered AHN2 point cloud product . . . 124
Figure 78 Point density of the filtered + unfiltered AHN2 point cloud product . . . 125
Figure 79 Histogram of the average point density per square meter of the filtered + unfiltered AHN2 point cloud product . . . 126

LIST OF TABLES

Table 1 Points classified as ground and non-ground for both filtered and filtered + unfiltered AHN2 point cloud data for a random sample of 128 928 866 filtered points and 204 737 885 unfiltered points. 49
Table 2 Points classified as building for a random sample of 967 987 filtered points and 2 066 287 unfiltered points from the AHN2 data set. 52
Table 3 Points classified as vegetation for a random sample of 967 987 filtered points and 2 066 287 unfiltered points from the AHN2 data set. 54
Table 4 Details of test data sets, heights are with respect to the Dutch geodetic datum. 82
Table 5 Completeness of digital elevation models expressed in commission (C) and omission (O) for test data sets. 94
Table 6 Completeness of digital surface models expressed in commission (C) and omission (O) for test data sets. . 94
Table 7 Accuracy assessment of digital elevation models for test data sets (in meters). 97
Table 8 Accuracy assessment of digital surface models for test data sets (in meters). 97
Table 9 Confusion matrix for not-filled digital elevation model (Dronten). 98
Table 10 Confusion matrix for filled digital elevation model (Dronten). 98
Table 11 Confusion matrix for the OHN digital elevation model (Dronten). 99
Table 12 Confusion matrix for my digital elevation model (Dronten). 99
Table 13 Confusion matrix for not-filled digital surface model (Dronten). 100
Table 14 Confusion matrix for OHN digital surface model (Dronten). 100
Table 15 Confusion matrix for not-filled digital surface model (Dronten). 100
Table 16 Quality of digital surface models for test data sets. . . 101
Table 17 Quality of digital surface models for test data sets. . . 101
Table 18 Comparison between the TOP10NL, BAG and BGT data sets. 121
Table 19 Confusion matrix for not-filled digital elevation model (Kerkrade). 127
Table 20 Confusion matrix for filled digital elevation model (Kerkrade). 127
Table 21 Confusion matrix for the OHN digital elevation model (Kerkrade). 127
Table 22 Confusion matrix for my digital elevation model (Kerkrade). 127
Table 23 Confusion matrix for not-filled digital surface model (Kerkrade). 128
Table 24 Confusion matrix for OHN digital surface model (Kerkrade). 128

Table 25 Confusion matrix for my digital surface model (Kerkrade).128
Table 26 Confusion matrix for not-filled digital elevation model (Leiderdorp). 129
Table 27 Confusion matrix for filled digital elevation model (Leiderdorp). 129
Table 28 Confusion matrix for the OHN digital elevation model (Leiderdorp). 129
Table 29 Confusion matrix for my digital elevation model (Leiderdorp). 129
Table 30 Confusion matrix for not-filled digital surface model (Leiderdorp). 130
Table 31 Confusion matrix for OHN digital surface model (Leiderdorp). 130
Table 32 Confusion matrix for my digital surface model (Leiderdorp). 130
Table 33 Confusion matrix for not-filled digital elevation model ('s-Gravenhage). 131
Table 34 Confusion matrix for filled digital elevation model ('s-Gravenhage). 131
Table 35 Confusion matrix for the OHN digital elevation model ('s-Gravenhage). 131
Table 36 Confusion matrix for my digital elevation model ('s-Gravenhage). 131
Table 37 Confusion matrix for not-filled digital surface model ('s-Gravenhage). 132
Table 38 Confusion matrix for OHN digital surface model ('s-Gravenhage). 132
Table 39 Confusion matrix for my digital surface model ('s-Gravenhage). 132

LIST OF ALGORITHMS

4.1 Rewriting array algorithm 62
4.2 Breadth-first search 62

ACRONYMS

AHN Algemeen Hoogtebestand Nederland iii
ALS Airborne Laser Scanning iii
BAG Basisregistraties Adressen en Gebouwen 17
BFS Breadth-first search 61
B-REP Boundary Representation 60
BGT Basisregistratie Grootschalige Topografie 121
BRT Basisregistratie Topografie 85
CHM canopy height model 6
DT Delaunay Triangulation 33
DBM Digital Building Model 6
DEM Digital Elevation Model iii
DSM Digital Surface Model iii
DTM Digital Terrain Model 1
GDAL Geospatial Data Abstraction Library 38
GIS Geographic Information Systems 5
GPS Global Positioning System 7
IDW Inverse Distance Weighting 2
INS Inertial Navigation System 7
ISO International Organization for Standardization iii
LAS LASer 8
LiDAR Light Detection And Ranging iii
nDSM normalized Digital Surface Model 6
NDVI Normalized Difference Vegetation Index 17
NAP Normaal Amsterdams Peil 81
NIR Near Infrared 7
NNI Natural Neighbor interpolation 31
RGB Red, Green and Blue 22
RMSE Root Mean Square Error 43
SDI Spatial Data Infrastructure 2
SD Standard deviation 54
TIN Triangular Irregular Network 5
TLS Terrestial Laser Scanning 18
VLR Variable Length Record 8
VRT Virtual Raster 39
XML Extensible Markup Language 39

1 INTRODUCTION

1.1 MOTIVATION

The recent emergence of LiDAR scanning technology has resulted in the availability of very large three dimensional point cloud data sets [Wouda, 2011]. These LiDAR data sets have become a main source for the modeling and reconstruction of both ground surface as well as above-ground objects such as buildings and vegetation. In case of a reconstruction of a ground surface often is referred to the terms Digital Terrain Model (DTM) (vector-based) or DEM (raster-based), where in case of a reconstruction of both ground surface as well as above-ground objects is often referred to a DSM (raster-based, see Figure 1). Both models provide high resolution information about a terrain's surface and can be used as input for many applications such as 3-dimensional visualization [Döllner and Hinrichs, 2000], modeling water flows [Li et al., 2008], precision farming [Senay et al., 1998], forestry [Akay et al., 2009], intelligent transportation systems [Li et al., 2014] and many more.

In comparison with traditional land surveys and photgrammetry it is possible with LiDAR technology to obtain data with centimeter level accuracy [Hu, 2001] in a cost-effective way [Lohr, 1998]. The production of DEM and DSM products using LiDAR is faster, can be more automated and coupled with the high density of point measurements it can offer greater definition of urban features. These factors encouraging research into the automated extraction and characterization of surface features [Priestnall et al., 2000].

However, the effective processing of LiDAR data and the generation of efficient and high-quality DEMs and DSMs remain big challenges [Liu, 2008a]. Compared with developed LiDAR hardware techniques for capturing data, LiDAR data processing techniques such as modeling of systematic errors, filtering, feature detection and thinning are very important to application [Sithole and Vosselman, 2005].



Figure 1: Digital Surface Model [Tukay Mapping]

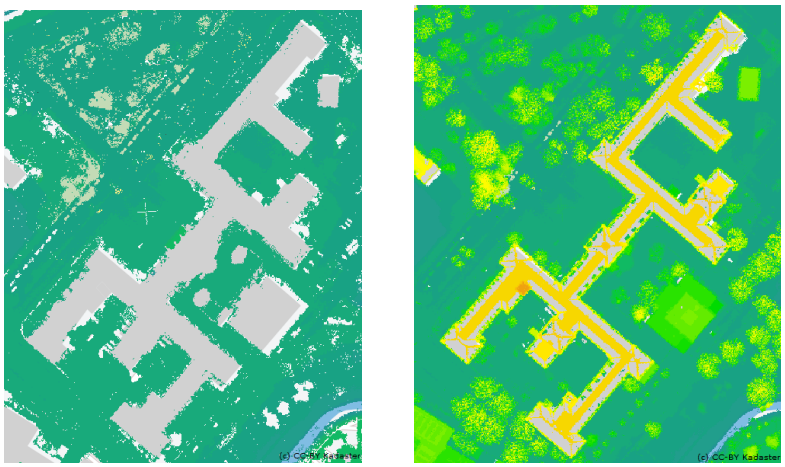
DEMs and DSMs can be stored both in a vector-based and a raster-based file format, scope of this thesis is focused on the generation of raster-based DEM and DSM. Podobnikar and Vrečko [2012] state that the process of generating raster-based DEMs and DSMs from LiDAR point cloud data is complex: results depend on chosen methods, algorithms, parameters and on different aspects of data quality. During the rasterization process, a conversion from 3-dimensional LiDAR data into 2.5-dimensional raster-based data will take place: data is stored in a 2-dimensional grid of raster cells, better known as a height map. By storing a height value for each raster cell a 2.5-dimensional representation is achieved, the model is embedded in 3-dimensional space but is not able to represent all 3-dimensional shapes. Only one height value is stored for a raster cell, e.g. the highest or lowest height value.

For the Netherlands, the second version of the AHN is a country-wide high-resolution point cloud data set comprising the nation's terrain by measuring heights, obtained by LiDAR, using ALS. The average point density of the data set is on average between 6 and 10 points per square meter [Van der Zon, 2011]. Such a high point density makes the AHN2 data set large; it contains approximately 640 billion point records [Kadaster, 2014c]. The initiative for the AHN2 data set is taken by the 26 Dutch water boards and Rijkswaterstaat, the executive directorate general for public works and water management of the Dutch government. The collection of the AHN2 data set was finished in 2012 and the point cloud data is publicly available as open data since 2014 via the Dutch Spatial Data Infrastructure (SDI) PDOK. Also raster-based height maps, generated by interpolation of the point cloud data by Inverse Distance Weighting (IDW), are available at 0.5 and 5.0 meter resolution.

Van der Zon [2011] indicates that the Dutch water boards and Rijkswaterstaat make use of the AHN2 raster-based height maps for nearly all water management tasks. In general the 0.5 meter resolution data is used, for some applications data is processed further. Most water boards do not use the point cloud data because most users are far more familiar with working with raster-based grids and the hard- and software environment do not always accommodate the convenient use of point data. In addition a lack of knowledge, communication and documentation hampers the potential use of point cloud data.

The usage of raster-based products is surprising since none of the currently available raster products based on the AHN2 data set can be considered as being good: height maps contain 'holes' (raster cells with a no-data values), unintentional dynamic objects are present within the data, wrongly applied interpolation methods and more (Figure 2). Cause for these errors are deviations during the collection of the point cloud data as described by Van der Zon [2011] and the applied methodology to process the LiDAR data.

Kramer et al. [2014] introduce a methodology in order to fill holes within the raster-based height maps from PDOK by making use of external open 2-dimensional geodata sets. This approach is somehow limited in its possibilities: filtering of erroneous data that should not be present within the raster-based height maps cannot not take place. In order to solve the errors within the raster-based height maps from PDOK just filling the holes within the raster data is not enough; additional steps in the processing of LiDAR data are essential before generating raster data. This leaves space for an approach where it could be possible to solve some or even all of the problems introduced in this section that occur within currently available raster-based height maps.



(a) (b)

Figure 2: Faculty of Architecture and the build Environment, Delft (a) A digital elevation model, containing holes located at building envelops, water surfaces, and small urban objects. (b) A digital surface model, containing holes within buildings (gray) and water (blue) surfaces, presence of small urban objects which temporarily perturb the scene (white) [PDOK, 2014].

1.2 RESEARCH QUESTION

This thesis will identify the possibilities to improve the quality of raster-based height map products based on the AHN2 data set with respect to currently available products. The main research question of this thesis is:

What quantitative degree of quality can be achieved for raster-based height maps generated from AHN2 point cloud data by the application of an automated process?

To answer the main research question the following sub-questions are relevant:

- What kinds of errors are most common within currently existing raster-based height maps based on the AHN2 data set?
- What strategy is appropriate for the processing of large amounts of point cloud data to raster-based height maps?
- Given a point cloud sample, which algorithm or methodology is best filter out different classes of information?
- What interpolation technique is most appropriate to estimate a height at a given location for different classes of objects?
- To which extent can external 2D geodata sets help to improve the quality of raster-based height maps?

This research will investigate the possibilities to develop an automated method for the generation of raster-based height maps from high-resolution LiDAR point cloud data. The advantages and disadvantages of different steps within the process will be discussed and a validation and quality assessment of the output will be performed.

1.3 SCOPE

Vosselman and Maas [2010] distinguish four main steps within the processing procedure of point cloud data to raster-based height maps:

1. Data acquisition
2. Registration of strips and geo-referencing of the point cloud
3. Filtering the point cloud
4. Interpolation and also sometimes smoothing of output data

For the AHN2 data set the data acquisition, registration of strips and geo-referencing of the LiDAR data already has taken place. One assumption in this thesis is that both steps are executed properly. Nevertheless, Van der Zon [2011] indicates possibilities to increase the quality during the data acquisition of the AHN2 data set. In Sande et al. [2010] it has been proved that the quality registration of strips and geo-referencing of the AHN2 data set leaves space for improvement.

This research will mainly focus on the possibilities to improve the other steps as introduced by Vosselman and Maas [2010]; filtering, interpolation and smoothing of the data. Focus will be on quality of the outcome of the algorithm, not on performance of the algorithm itself. Computation time of the algorithm is something that comes secondary.

1.4 OUTLINE

In chapter 2 basic concepts and terminology related to digital terrain modeling and LiDAR will be introduced and defined. In the second part of chapter 2 the AHN2 data set and its derivative products will be introduced and discussed.

In chapter 3 an overview of related work for the generation of DEMs and DSMs will be introduced. Related work with respect to five components within the work flow will be introduced: pipeline generation, filtering, spatial interpolation, post processing and raster visualization. Parameters will be introduced in order to assess quality of geographic information. In the final part of this chapter a discussion will take place where the components introduced in this section which are interrelated with each other in order for usage within an integral strategy.

In chapter 4 a methodology for the automatic generation of raster-based height maps is proposed which consist of the generation of three classes of data: ground, buildings and vegetation. For all classes a description is provided of the applied steps, specific issues for each class and the decisions made in order to solve them.

In chapter 5 an validation and assessment of the outcome of the methodology proposed in chapter 4 will take place. Therefore a number of test data sets will be introduced. They will be validated and assessed with respect to currently available raster-based height maps in order to measure the degree of improvement.

In chapter 6 main- and sub-research questions will be answered, the developed methodology will be discussed and suggestions for future work will be given.

2 STATE OF THE ART IN AHN2 PRODUCT DEVELOPMENT

In this chapter a overview will be given of the different AHN2 products. First, in section 2.1 related terminology with respect to digital terrain modeling will be introduced. As second, an introduction of the basic concepts of LiDAR will be introduced in section 2.2. In section 2.3 a deeper introduction will be given of the AHN2 data set. Finally, in section 2.4 derivative products based on the AHN2 data set will be introduced and discussed.

2.1 DIGITAL TERRAIN MODELING

In order to make the modeled data usable for Geographic Information Systems (GIS) conversion of the point cloud datasets into a geographical data format is needed. In the context of GIS there exist two file formats to store geographical data:

- Raster-based
- Vector-based

DEMs and DSMs can be represented both in a vector-based as well in a raster-based way.

In case of a vector-based method data is stored in a Triangular Irregular Network (TIN). This is a 2.5-dimensional triangulation based on the work of Peucker et al. [1978].

In case of a raster-based method data is stored in a grid of squares, better known as a height map. By storing a height value for each raster cell separately the data can be qualified as 2.5-dimensional, where the term 2.5-dimensional refers to a model that is embedded in 3-dimensional space, but is not able to represent all 3-dimensional shapes but only one height value for each raster cell. The resolution of a raster cell is the size related to a real world width in ground units.

Both file formats have their advantages and disadvantages with respect to each other. By conversion raster-based data can be transformed to vector-based data and vice versa; within this conversion process a loss of data quality needs to be taken into account. The vector-based TIN DEM data set is also referred to a primary and measured DEM, whereas a raster-based DEM is referred to a secondary and computed DEM [Toppe, 1987].

In section 1.1 the relevance of DEMs and DSMs is introduced. Since these models could be used for multiple applications, not all information is relevant for all applications: in case of an application as forestry information about buildings is probably irrelevant. For this reason it might happen that not all the information is necessarily needed within a model; different kinds of models can be generated that contain different kinds of information.

There are no official standards that describe the format of digital terrain models. This has led to a situation where terms are used synonymously and different models containing different kinds of information are called similarly.

Behrendt [2012] distinguish three different digital models, Wichmann [2012] adds a fourth one:

DTM

The first model is a DTM, a vector-based bare-earth representation with irregular spaces between points stored in a 2.5D TIN (Figure 3).

DEM

The second model is a DEM, a raster-based representation of the DTM (see Figure 2a). The conversion from vector-based to raster-based data is performed by the application of spatial interpolation (section 3.4).

DSM

The third model is a raster-based DSM representing the first echo/return the laser received for each laser pulse sent out. It represents the building roofs, treetops, and tops of other objects or the ground, if unobstructed (Figure 4).

nDSMs, CHMs and DBMs

The fourth model is a normalized Digital Surface Model (nDSM) containing above-ground objects as buildings and vegetation. Height data is determined with respect to the underlaying ground (see Figure 4). A canopy height model (CHM) is subset from the nDSM that only contains information about vegetation, a Digital Building Model (DBM) is a subset from the nDSM that only contains information with respect to buildings.

These definitions will be applied in the remainder of this thesis in order to express different models.

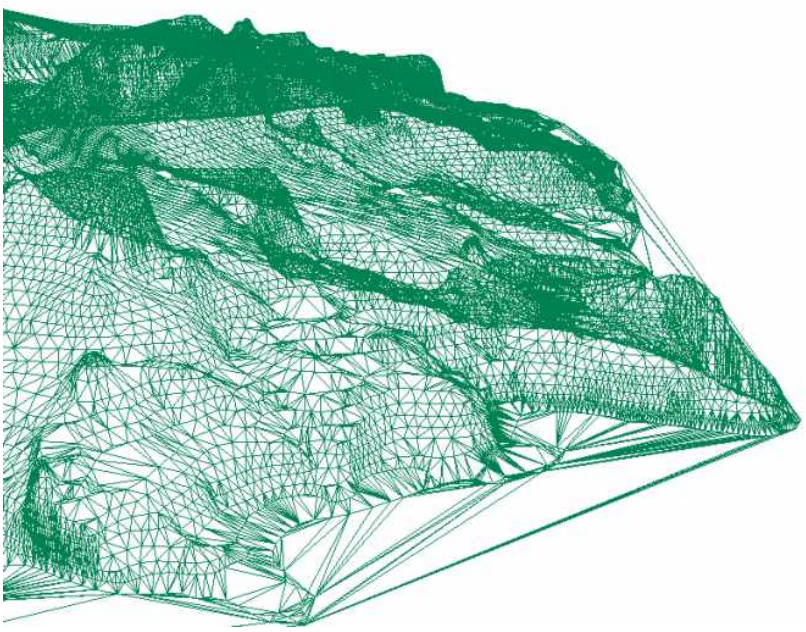


Figure 3: A digital terrain model stored as a triangular irregular network [Pudelko, 2007].

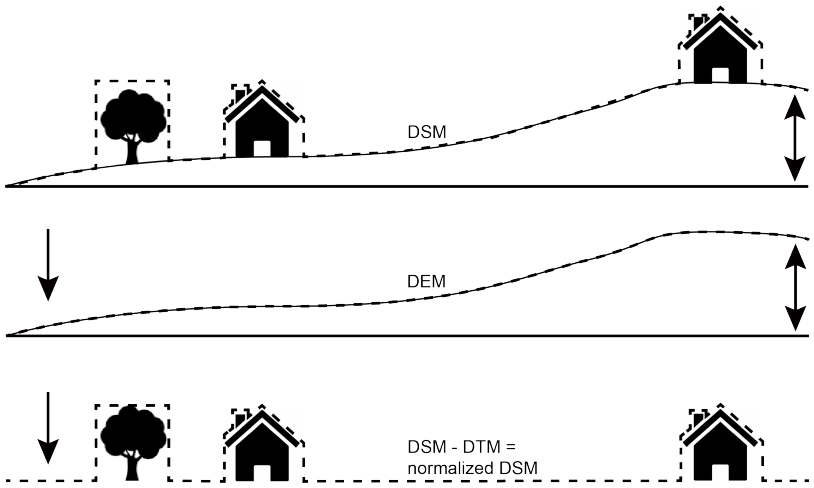


Figure 4: The difference between a DSM, DEM and a nDSM.

2.2 LIDAR

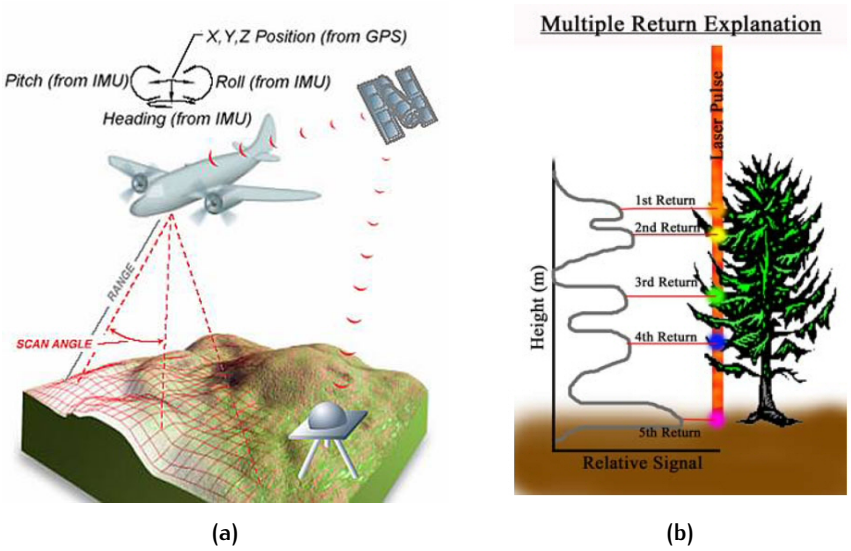
LiDAR is an active remote sensing technology that measures distance by illuminating a target with a laser and analyzing the reflected light. These light pulses combined with other data recorded by the airborne system generate precise 3-dimensional point-based information with respect to the shape of the Earth and its surface characteristics. Wehr and Lohr [1999] distinguish four main components within LiDAR systems:

- The laser
- The scanner and optical system
- The receiver and processing system
- Inertial Navigation System (INS) and Global Positioning System (GPS) for the correction and geo-referencing of the collected data

INS measures roll, pitch, and heading of the LiDAR system. GPS measures the position and time that the recording took place (Figure 5a). The raw data is post-processed after the data collection survey into high-accurate geo-referenced 3-dimensional coordinates by analyzing the laser time range, laser scan angle, GPS position and INS information.

Wehr and Lohr [1999] distinguish two different types of LiDAR: topographic and bathymetric LiDAR. Topographic LiDAR typically uses a Near Infrared (NIR) laser to map the land, while bathymetric LiDAR uses water-penetrating green light to also measure sea floor and riverbed elevations. Both systems are complementary; due to the characteristics of topographic LiDAR water is penetrated much more, so not reflected, the amount of returns is low on water surfaces.

One emitted laser pulse can return to the LiDAR sensor as one or many returns but LiDAR systems commonly record multiple returns; these returns are separate measurements for the light returning from discrete elevation layers (earth, vegetation, buildings) [Fancher, 2012]. Figure 5b shows that any emitted laser pulse that encounters multiple reflection surfaces as it travels toward the ground is split into as many returns as there are reflective surfaces.



(a) (b)

Figure 5: Visualization of two LiDAR concepts. (a) Different components of a LiDAR system [Fancher, 2012]. (b) When a LiDAR beam hits vegetation this can lead to multiple returns of point data [Fancher, 2012].

2.2.1 The LAS format

Main goal of this section is to introduce relevant parts of the LASer (LAS) format. For a complete description of the LAS format see [ASPRS, 2013]. The LAS format is a public and industry-standard file format for the storage, distribution and interchange of 3-dimensional LiDAR data between users. A LAS file consists out of three parts:

- A public header block;
- Variable Length Record (VLR) and;
- Point data records.

The public header block contains information about the name of generating software, version number, and statistics like minimum and maximum values of XYZ are stored in the public header block. The VLR defines the content of a LAS file:

- Number of variable length records
- Offset to the start of the points
- Type and size of each point
- Number of points
- Offsets and scale factors for point coordinates
- Bounding box describing the XYZ extends of all point records within the LAS file

For point data records the following meta data can be stored:

- XYZ coordinate
- Intensity
- Return Number
- Number of Returns of Pulse
- Scan Direction Flag
- Edge of Flight Line
- Classification
- Scan Angle Rank
- User data
- Point Source ID

Figure 6 shows an example of a public header block and a VLR.

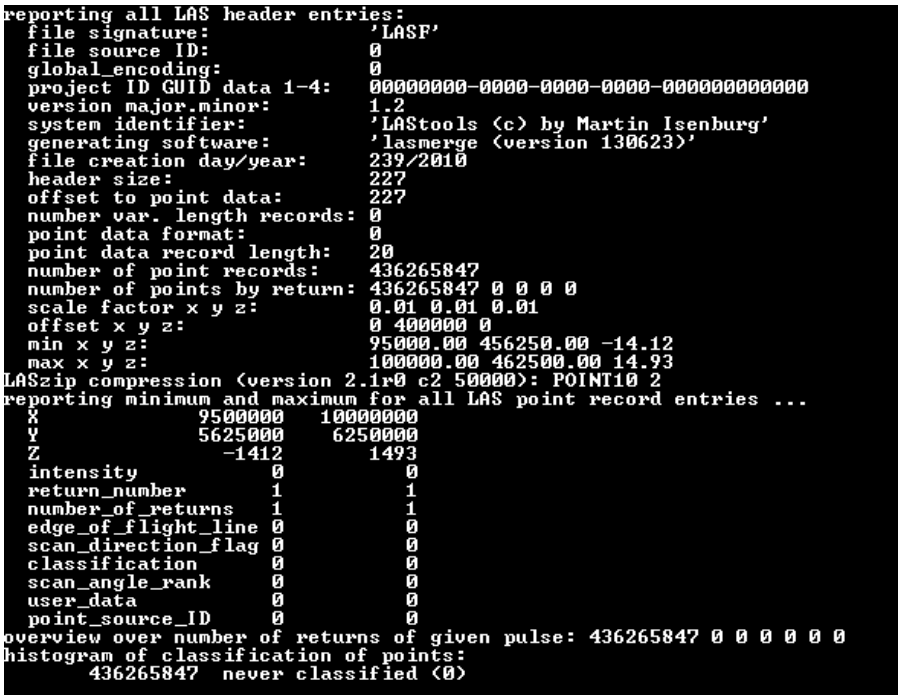


Figure 6: Example of a public header block and Variable Length Records information of an AHN2 tile

2.3 THE AHN2 DATA SET

In this section the AHN2 data set will be introduced. In the first part of this section different point cloud products will be introduced and linked to the different models as introduced in section 2.1. In the second part of this section the distribution of the AHN2 point cloud data will be treated. In the third part of this section a deeper look in the point cloud data will take place in order to see what information is available within the AHN2 point cloud data.

2.3.1 AHN2 point cloud products

The AHN2 point cloud data is available in two products:

- A *filtered* product containing ground points
- A *unfiltered* product containing all remaining non-ground points

The reason for this classification is because the original purpose for collecting the AHN2 point cloud data was to collect information about the earth's surface for purposes such as dike management and mapping [Swart, 2010]. Rijkswaterstaat and the Dutch water boards were originally only interested in point data related to the ground.

AHN2 filtered

For the generation of the *filterded* AHN2 point cloud a filtering procedure is developed based on the height, slope and multipath in order to filter out all points that are classified as non-ground points [Swart, 2009]. Automatically separating ground and non-ground points from LiDAR point clouds has proven to be difficult, especially for large areas of varied terrain characteristics [Liu, 2008b]. For that reason the filtering procedure is done only partly automated for the AHN2 data set. Additional manual filtering is applied in order to deliver a product that meets the requirements defined by the Dutch water boards and Rijkswaterstaat [Van der Zon, 2011].

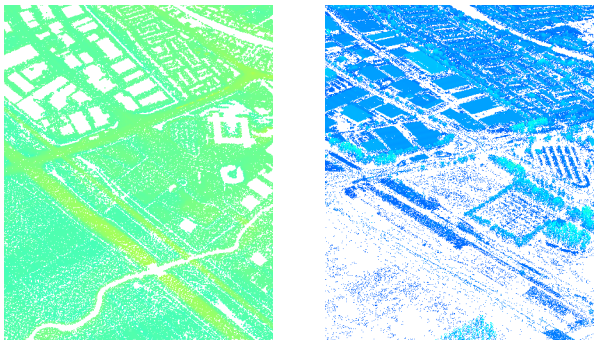
This point cloud product can be classified being a DTM. The product could be used as input for the generation of a DEM; a raster-based representation of a terrain's surface (see section 2.1). Figure 7a shows that the AHN2 data set contains bare ground points including slope-based objects like infrastructural dikes. Above-ground objects such as vegetation and buildings are filtered out properly.

AHN2 unfiltered

The *unfiltered* AHN2 point cloud product contains all remaining LiDAR points. Figure 7b shows clearly that all above-ground objects like buildings, vegetation and infrastructural objects like bridges are present within this product. Also outliers are included, these are points which have significantly higher or lower elevations with respect to elevations expected for ground, buildings and vegetation.

Points reflected on thin clouds, birds and other (dynamic) above-ground objects will result in measured point elevations that are significantly higher with respect to other neighboring points. Multiple returns from structures and vegetation (multipath) can result in returns with an excessively longer travel time and thus point elevations that are significantly lower with respect to other neighboring points.

After the generation of a DEM it is possible to calculate the normalized height of the above-ground points with respect to the DEM. In that way it is possible to use this product as input for the generation of a nDSM (section 2.1).



(a) (b)

Figure 7: Visualization of AHN2 point cloud products. (a) Filtered AHN2 point cloud. (b) Unfiltered AHN2 point cloud.

Distribution of the AHN2 data set

The AHN2 data set is available as open data via the web portal of the Dutch SDI PDOK[1]. The whole AHN2 data set contains in total approximately 640 billion points. This makes the data set is large in file size. In order to improve the accessibility and distribution of smaller parts of the AHN2 data set it has been decided to split up the AHN2 data set into non-overlapping tiles. The point cloud products are distributed in 1 372 tiles each covering an area of 5 x 6.25 kilometers (Figure 8). The naming system for the tiles is based on the *TOP10NL* data set (Appendix B).



Figure 8: Tiles of the AHN2 data set covering the Netherlands [PDOK, 2014]

Available meta data within the AHN2 data set

As introduced in section 2.2, the LAS format has the capability to store meta data for a point record. Hence, potential relevant meta data such as intensity, number of returns, flight line information and classification are not available within the AHN2 data set (Figure 6). According to the product specifications, defined by the Dutch water boards and Rijkswaterstaat, there was no need to share or publish these meta data within the LiDAR files. As acquisition and quality control is performed by other contractors then the company that collected the LiDAR data quality has to define explicitly by the companies that collected the data. After acceptance of the point cloud data by the clients the quality control products are archived; they are not distributed and cannot be ordered [Swart, 2010].

1 https://www.pdok.nl/nl/producten/pdok-downloads/atomfeeds/a

2.4 EXISTING PRODUCTS

In this section relevant products based on the AHN2 data set will be introduced and analyzed in order to get more insight in their advantages and disadvantages. In subsection 2.4.1 and subsection 2.4.2 two raster-based height maps based on the AHN2 data set will be introduced. In subsection 2.4.3 an oversight of further research with respect to the generation of raster-based products from the AHN2 data set is given.

2.4.1 PDOK raster-based height maps

Based on the AHN2 point cloud data also raster-based height maps derived from the AHN2 data set are available as open data via the Dutch SDI PDOK[2]. The *filtered* AHN2 point cloud data is processed in order to generate two types of DEM:

1. The not-filled DEM contains height data only determined by point records located within a raster cell.

2. The filled DEM fills incidental holes in case of a combination of local low point density and distribution [Van der Zon, 2011].

Figure 9 shows the difference between the two types of DEMs.

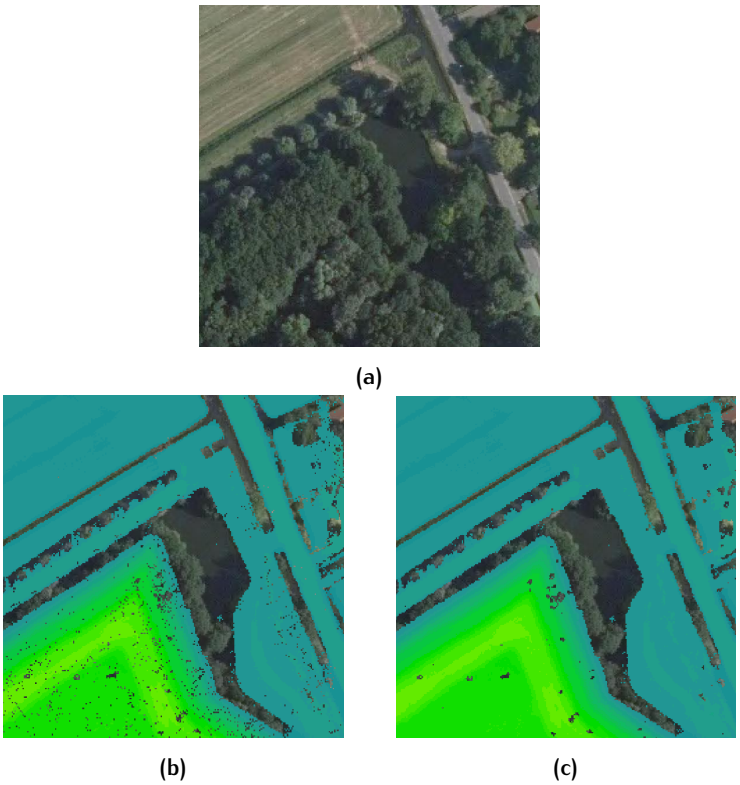


Figure 9: Difference between the not-filled and filled digital elevation model. (a) Aerial photograph. (b) Not-filled digital elevation model (own image, based on PDOK [2014]). (c) Filled digital elevation model (own image, based on PDOK [2014]).

2 https://www.pdok.nl/nl/producten/pdok-downloads/atomfeeds/a

A raster-based DSM is generated similarly as the not-filled DEM; a raster cell does only have a height value when there is at least one point record located within the raster cell.

Both the raster-based DEM products as well the DSM are available in two spatial resolutions:

- 0.5 meter
- 5.0 meter

Height values within the raster data at 0.5 meter resolution are determined using IDW interpolation the point records. For the estimation of a height value for a raster cell only those points are involved located within the raster cell (Figure 10).

The application of IDW interpolation results in the visibility of small objects like speed bumps and curbs. Van der Zon [2011] states that it is advantageous using IDW interpolation because it averages the stochastic error, by averaging the number of LiDAR points for each raster cell. For the AHN2 data set the stochastic error is calculated at 0.05 meter. The concept of IDW will be introduced more extensive in section 3.4.

For the 5 meter grid the height will be calculated by averaging the 0.5 meter grid within each 5 meter grid cell [Van der Zon, 2011].

Errors and holes within the raster-based DEM products from PDOK

Given a point density between 6 and 10 points/m^2 for the AHN2 data set Arcadis [2012] calculated that the average distance between two points at land even in the most pessimistic situation is 0.46 meter at maximum; theoretical a height value should be available for each raster cell covering land. In practice the theory does not work out; Figure 11 shows a random sample of the not-filled DEM product, a number of holes can be detected. Multiple causes for their occurrence can be distinguished:

- No ground point records are available for raster cells covering buildings. The methodology does not provide any solution to estimate a height value for such raster cells.
- Within the applied filtering procedure dynamic objects (e.g. cars) are most often not present within the *filtered* AHN2 point cloud data. In case that no point record hitting the ground is available this results in raster cells having no height data covering streets and parking lots.
- Almost no points records are available for raster cells covering water due to the characteristics of topographic LiDAR (see section 2.2). This results in raster cells covering water having no height data.
- It happens that LiDAR beams cannot penetrate through vegetation resulting in the unavailability of point records that hit the ground in case of dense vegetation (Figure 12).

During the generation of raster-based height data from AHN2 point cloud data no solution is provided for the estimation of a height value for a raster cell in the above described scenarios within the raster-based not-filled DEM and the filled DEM height map.

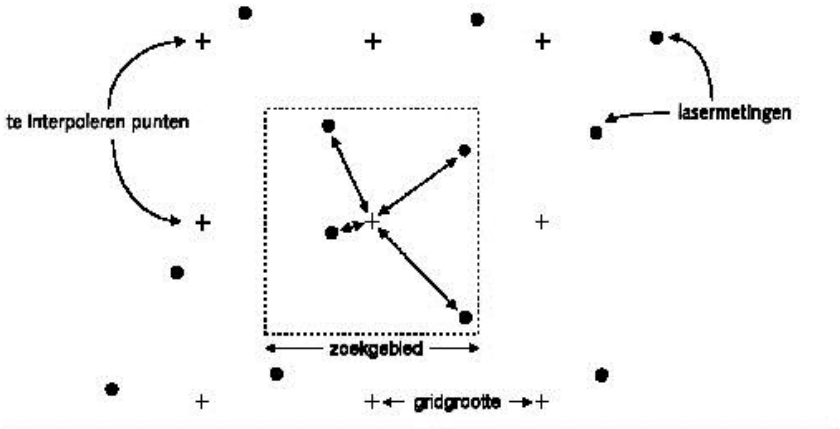
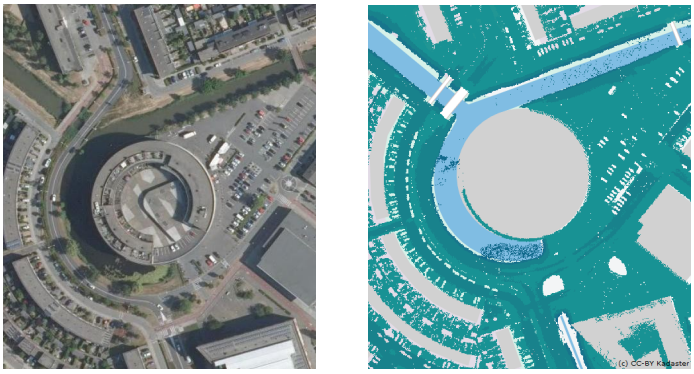
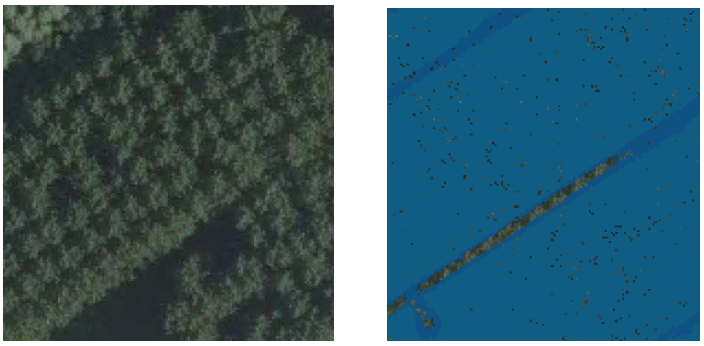


Figure 10: Inverse distance weighting as applied for the generation of raster-based height maps from AHN2 point cloud data [Rijkswaterstaat Meetkundige Dienst].



(a) (b)

Figure 11: Errors in PDOK digital elevation model. (a) Aerial photograph. (b) Not-filled digital elevation model containing holes near water, buildings and cars (own image, based on PDOK [2014]).



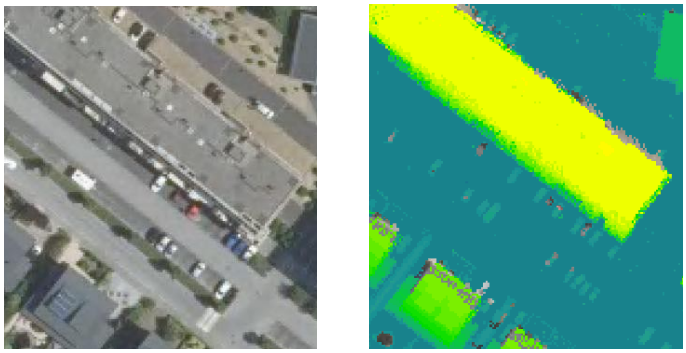
(a) (b)

Figure 12: Errors in the PDOK digital elevation model. (a) Aerial photograph. (b) Not-filled digital elevation model containing holes due to dense vegetation (own image, based on PDOK [2014]).

Errors and holes within the raster-based DSM product from PDOK

In case of the generation of a DSM both the filtered and unfiltered point cloud products are simply merged before applying *local* IDW interpolation. Within the raster-based DSM product the following errors occur:

- For vegetation, it can be assumed that a LiDAR pulse will have multiple returns from multiple branches and eventuelly even from the ground (section 2.2). Estimating the height from these returns by IDW interpolation will determine an average weighted height: this value is lower than the first return and higher then the last return which will be somewhere in between the ground and a treetop.
- In urban areas multiple records might return from vertical structures like building facades and balconies. These records are obtained due to the angle of the emitted laser beam with respect to the lower world over time. Interpolation of these point records using IDW results in fuzzy height data that is the weighted average of these 'unwanted' point records. Similar as for vegetation, the determined height will be somewhere in between the ground and a building roof (Figure 13).
- On the other side, shadows within the LiDAR data will result in missing raster data besides building structures (Figure 13). These shadows are caused due to angle of the emitted laser beam with respect to the lower world when the laser beam does not look 'backwards'.
- Multipath occurs when the laser reflects from a wall to the ground (outside the building) before it reflects back to the sensor. This multipath effect results in a point record with a measured distance as if it is directly to an object but based on the direction of the point it is located inside an object (subsection 2.3.1). Interpolation of these point records using IDW results in raster data with lower height values with respect to the true height and adjacent raster cells within buildings.
- No LiDAR point records within the AHN2 data set are excluded for the generation of a DSM. This leads to the presence of noise within the raster-based DSM: small urban objects which temporarily perturb the scene such as cars, roof antennas, cranes and other objects (Figure 13).



(a) (b)

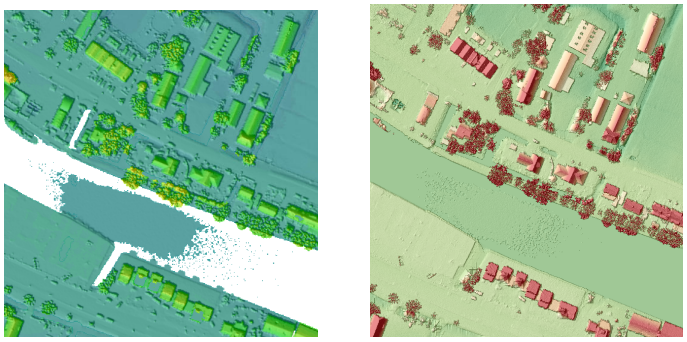
Figure 13: Errors in PDOK digital surface model. (a) Aerial photograph. (b) Digital surface model containing cars, fuzzy height data near building edges and LiDAR shadowing (own image, based on PDOK [2014]).

It can be concluded that the applied methodology for the generation of both DEM as well the DSM is erroneous and leaves space for improvement.

2.4.2 ESRI raster-based height maps

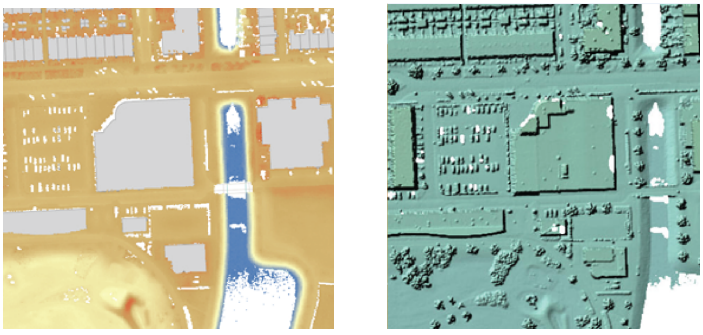
The Dutch department of the GIS company ESRI provides a number of raster-based height maps based on the AHN2 data set. Two maps are generated by interpolating a TIN that is constructed with AHN2 point cloud data. Application of the same methodology creates near-similar output (Figure 14). Interpolation based on a TIN will be introduced in subsection 3.4.2.

Figure 15 shows two samples of a raster-based DEM (Figure 14a) and DSM height maps (Figure 14b). Similar problems can be determined with respect to the height maps from PDOK (subsection 2.4.1). For this reason it can be assumed that the AHN2 point cloud data is interpolated directly without filtering of the point cloud data. In order to improve the visualization of the height maps a hill shading effect is added, a visual effect that provides an optical relief for cartography (section 3.6).



(a) (b)

Figure 14: Comparison between ESRI [2014] digital surface model and a raster-based height map generated by interpolation based on a triangular irregular network that is constructed from AHN2 point cloud data. (a) ESRI digital surface model (own image, based on ESRI [2014]). (b) Digital surface model generated by the interpolation of a triangular irregular network that is constructed from AHN2 point cloud data.



(a) (b)

Figure 15: Impression of two raster-based height maps from ESRI [2014]. (a) Digital elevation model (own image, based on ESRI [2014]). (b) Digital surface model (own image, based on ESRI [2014]).

2.4.3 Related work

In section 1.1 it has been introduced how DEMs and DSMs can be used as input for many applications. The quality of the output for these applications is dependent on the quality of the generated input DEMs and DSMs. In this subsection different research projects will be introduced that are related to improvement and/or the extraction of features from the AHN2 data set.

Object Hoogte Nederland

Kramer et al. [2014] propose a methodology for the filling of holes within the raster-based DEM and DSM height maps from PDOK (subsection 2.4.1). A set of rules are designed to fill holes by making use of external 2D geodata sets (*TOP10NL* and Basisregistraties Adressen en Gebouwen (BAG)) and aerial photographs. For the filling of holes within the raster-based DEM height maps the following rules are defined:

- Water bodies are detected with polygon data from the *TOP10NL* data set (see Appendix B). The minimum height within each polygon is determined and assigned to all raster cells within the polygon. Smaller water that is stored as line in the *TOP10NL* data set is not taken into account in the methodology.
- Building footprints are detected by polygonal data[3] in the BAG data set (see Appendix B). For each building footprint an (unknown) buffer is determined and the average height is estimated using all the known height values of raster cells located within the buffered polygon.
- All other holes are filled using IDW interpolation (see section 3.4).

For the raster-based DSM height maps the following rules are defined:

- For holes within build objects the average height is estimated from neighboring raster cells located within the representing building polygon from the BAG data set.
- Vegetation is detected by calculating a Normalized Difference Vegetation Index (NDVI) using aerial photography. Holes within vegetation are filled based on the average height of the detected vegetation.

A nDSM is generated by subtraction of the raster based DEM height map from the raster-based DSM height map.

This approach is somehow limited in its possibilities: filtering of erroneous data that are present in one of the raster-based products is not possible. Just filling holes within the raster-based height maps is not enough, processing of LiDAR data is essential before generating raster data.

Another problem is the temporal difference when combining different geodata sets. When two geodata sets have a different source date the information within both data sets might differ. Additional, a difference in definitions and the positional accuracy between geodata sets could be a source for errors. Figure 16d shows an example where do to a misunderstanding of the definitions of external 2D geodata sets the height value of raster cells is adjusted wrongly because they are located within a polygon representing water from the *TOP10NL* data set.

3 The smallest functional and architectural-constructive, self contained unit that is directly and permanently connected to the ground which is enter-able and lockable [BAO, 2013].

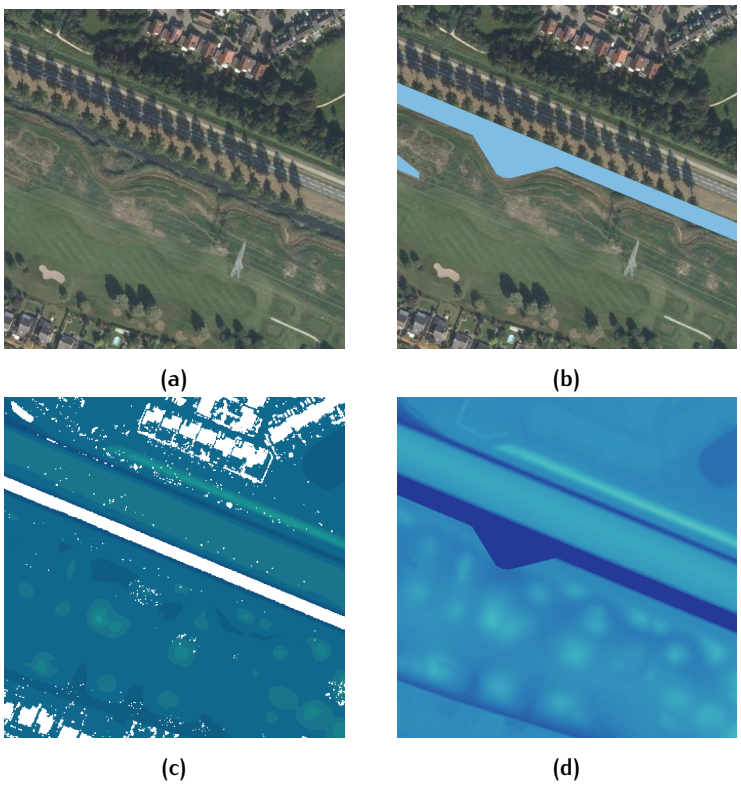


Figure 16: Errors within the methodology of Kramer et al. [2014]. (a) Aerial photograph. (b) *TOP10NL* polygon data (blue) defining water bodies. (c) Not-filled digital elevation model containing holes near water and buildings (own image, based on PDOK [2014]). (d) Digital elevation model containing raster cells having a height value that is overwritten by a false height value because they are located within a polygon representing water from the *TOP10NL* data set. (own image, based on Kramer et al. [2014], copyright Alterra, Wageningen UR).

Tree detection

Volkova [2014] developed a tree database using the AHN2 data set. This database included information about tree locations, tree crown projection parameters and several tree shape parameters. Aim of the study was to find a new way of delineating trees and deriving their parameters using the AHN2 data set. The quality of tree parameters derived from raster-based height maps generated from the AHN2 data set and point data was assessed using parameters derived from Terrestial Laser Scanning (TLS).

The positional accuracy of determined tree locations is 0.23 meter. Result are not very much reliable since only 63.07% of the tree locations derived from the AHN2 data set was correctly predicted. It has been concluded that the positional accuracy of the raster data is low due too low point densities in order to create raster data at a 0.5 meter resolution.

Shadow analysis

Geodan [2014] developed a raster-based map using the AHN2 data set called *Dynamic Holland Shading Map*[4]. Goal of this map is to detect and visualize the amount of direct sunlight through the year from day to day for individual raster cells at a spatial resolution of 0.5 meter.

Heights of above-ground objects are extracted from the BAG data set for buildings (see Appendix B) and information with respect to the height of vegetation is derived from the tree database of Volkova [2014].

Shadows are determined by a technique called *dynamic* hill shading. Calculation of hill shade data using different input parameters for the vector representing the illumination direction of the sun generates shadow information for each moment of the day (Figure 17). Hill shade calculations take place at a local scale where slope is determined based on adjacent raster cells (see subsection 3.6.2). For that reason hill-shading does not simulate real shadows.

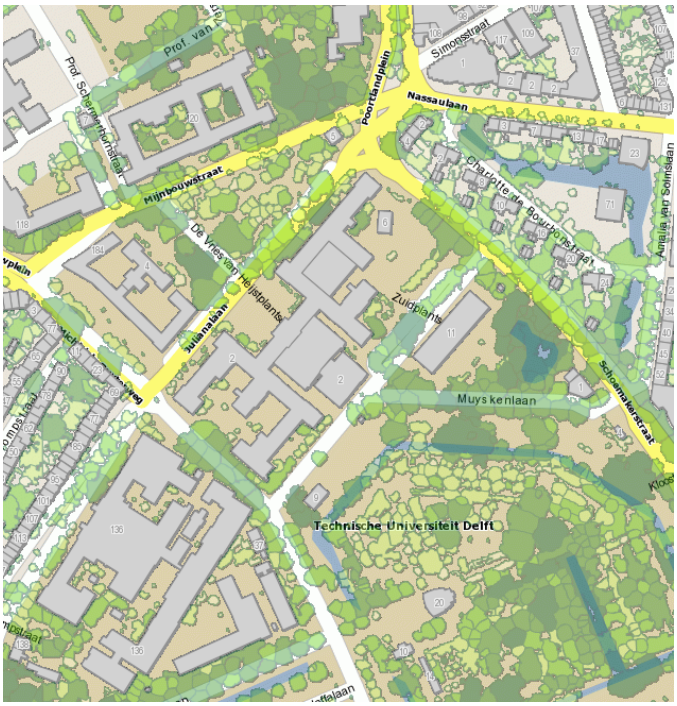


Figure 17: Dynamic Holland Shading Map (own image, based on Geodan [2014])

Usage of AHN2 products by Dutch water boards and Rijkswaterstaat

Van der Zon [2011] indicates that the Dutch water boards and Rijkswaterstaat use the AHN2 raster-based products for almost all water management tasks. In general the full 0.5 meter resolution is used but for some applications data is processed to a higher resolution. The point cloud data is used only for mapping purposes and this is mainly left to external contractors.

Most water boards do not use the point cloud data, mainly because most users are far more familiar with working with grids. Also the software and hardware environment do not accommodate the convenient use of the point data. A lack of knowledge, communication and documentation hampers the potential use of point cloud data.

4 http://research.geodan.nl/sites/ahn/

2.4.4 Evaluation

After the introduction of a number of raster-based products based on the AHN2 data set it can be concluded that all of them contain errors and that proposed solutions solve problems partly or even not at all. None of the products can be considered as being good: data contains holes, unintentional dynamic objects, wrongly applied interpolation methods and more. Cause for these errors can be found in deviations during the collection of the point cloud data as described by Van der Zon [2011] and the applied methodologies to process the point cloud data.

This thesis will identify the possibilities to improve the quality of raster-based height maps based on the AHN2 data set. In order to do so, it is necessarily needed to go back to point cloud level in order to solve current errors that occur on raster level.

3 RELATED WORK

In this chapter related work regarding the generation of raster-based DEMs and DSMs from point cloud samples will be introduced. It is not impossible to process the AHN2 data set at once for current computers; pipelining is needed first in order to feed massive point cloud data data sequentially to the computers' main memory [MacDonald et al., 2004]. Once that step is taken it is possible to filter and interpolate the point cloud data. Final steps are post processing and visualization in order to improve the quality and visibility of the raster data. Figure 18 shows a flowchart of the different processing steps that will be introduced in this chapter:

- Pipeline generation (section 3.2)
- Filtering of the point cloud (section 3.3)
- Spatial interpolation (section 3.4)
- Post processing (section 3.5)
- Raster visualization (section 3.6)

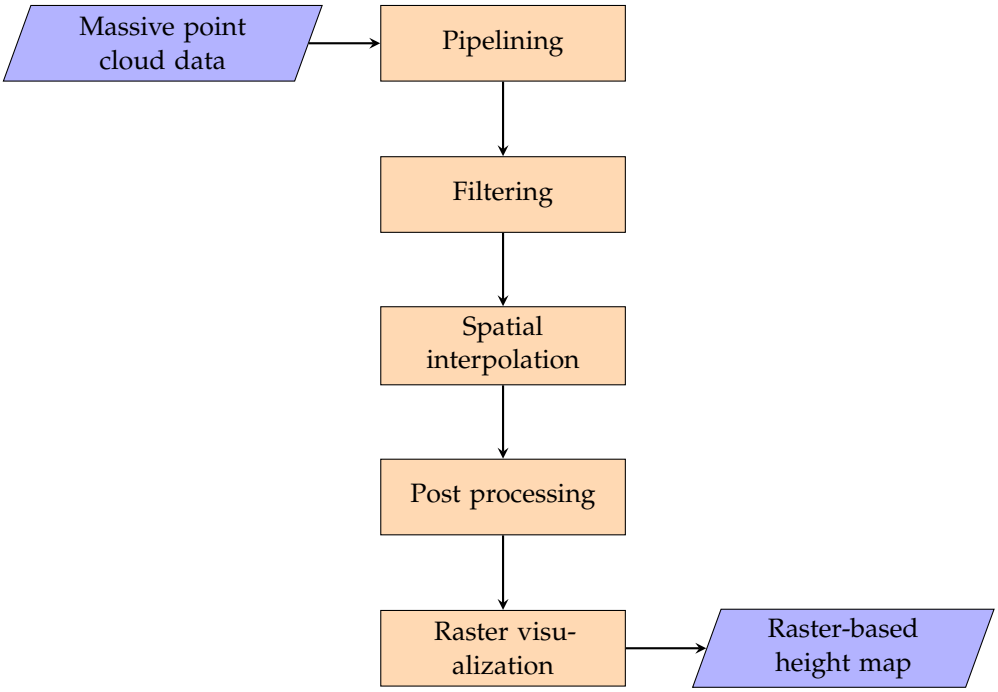


Figure 18: Work flow for the generation of a raster-based height map from massive point cloud data.

3.1 RELATED PROJECTS

In section 2.4 the state of the art in product development regarding the AHN2 data set has been introduced. The Netherlands is not the only country in the world developing a national point cloud data sets. In this section related projects in other countries will be introduced.

Switzerland

Luethya and Stengeleb [2005] describe the different steps (e.g airborne data acquisition, preprocessing, filtering) that were taken in order to generate a national DEM and DSM for Switzerland. The average point density is 1 point per square meter with a vertical accuracy of approximately 0.3 meter.

For preprocessing, the different products need special attention because of their size, their importance for subsequent processes or a combination of both. Before interpolation point cloud data is merged from different flight lines and split up in tiles, smaller subsets of the massive point cloud. In order to reduce artifacts along the borders of these tiles a buffer of 30 to 50 meters is applied when processing the data.

For data filtering an automatic classification algorithm based on adaptive triangulation (see Axelsson [1999]) was applied to filter ground points from non-ground points. The error rate differs between 0% and 10% of all points, where zero percent of defects are often detected in flat areas. For the generation of a DSM only the first returns are used for a classification into ground, vegetation and buildings points. A higher degree of manual classification and editing was necessary: due to the specification only permanent objects were allowed in the data set which means that recognizable objects like trains or annually changing vegetation had to be removed [Luethya and Stengeleb, 2005]. A combination of an automatic classification algorithm (based on adaptive triangulation, see Axelsson [1999]) and making use of external 2D geodata regarding building footprints from cadastral surveying are used in order to distinguish buildings. No further details regarding the applied interpolation methods and post processing are provided.

Denmark

Geodatastyrelsen, the Danish geodata agency, is in the process of gathering LiDAR point cloud data set for the generation of a new and better elevation model since spring 2014 [Geodatastyrelsen, 2013]. The data set does have a point density of about 4 points/m^2 which is lower than the point density of the AHN2 data set. Nevertheless, this data set contains information that is not available for the AHN2 data set:

- For each point is the Red, Green and Blue (RGB) colors are detected of the returned laser signal for each LiDAR point. The registered RGB colors can be used for enhanced visualization and improving the automation process for object identification and point classification.
- Full waveform data is available, not only the first and last return of the laser pulse is registered, but the full waveform. Currently, full waveform data is mainly used in scientific studies such as forestry [Geodatastyrelsen, 2013].

Septima [2014] used a part of this data set in order to produce a raster-based height map with a raster resolution of 0.4 meter applying interpolation based on a TIN (Figure 19). Claimed is that the map is just a representation of raw point cloud data: errors are not filtered [Septima, 2014] and no constraints are applied within the interpolation process. This resulted in a number of errors like holes due to noise that is present within the data (Figure 19b). Since no constraints are applied this strategy leads to the occurrence of artifacts where point density is low or in areas where point data is missing, for example near water bodies and building footprints in DEMs (Figure 19c).

Interesting about the raster-based height map generated by Septima [2014] is the applied methodology for visualization. With respect to the raster-based height maps based on the AHN2 data set a higher raster resolution is applied and the visualization is more clear for human eye while the input data set does have a lower point density.

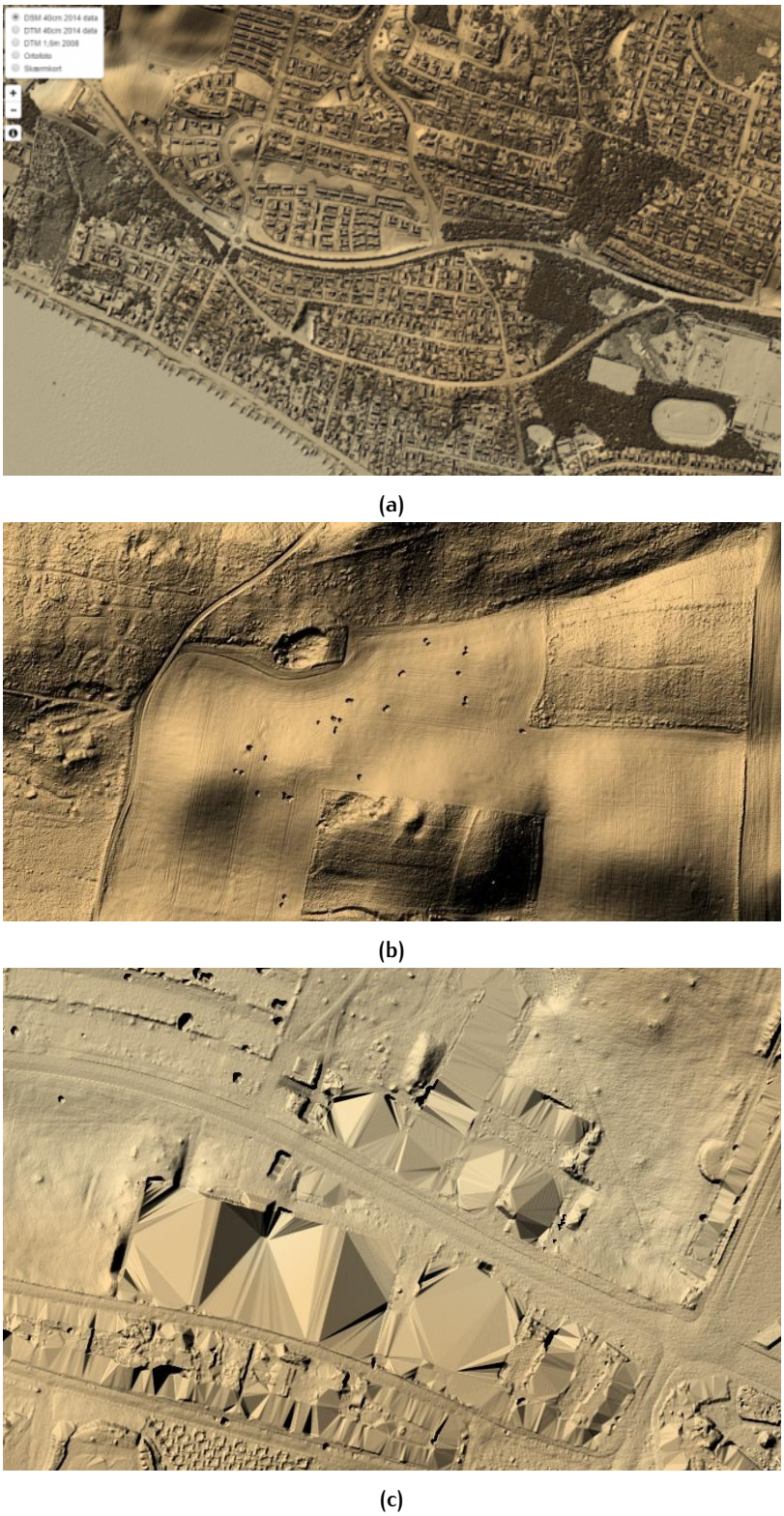
Developments worldwide in relation to the Netherlands

Most development regarding the collection of national LiDAR data sets take place within Europe and North America currently. When looking at developments regarding the collection of nationwide LiDAR point cloud data at a country level, no other country provides or will provide a LiDAR data set with such a high point density as the Netherlands in the near future.

The AHN2 data set does not contain much meta data (section 2.3), where the Danish point cloud data provides interesting futures such as RGB colors and full waveform data. The successor of the AHN2 data set, the AHN3 data set, is expected to be published as open data starting from 2015 and will contain more meta data (see section 6.3). Nevertheless, it can be concluded that the situation regarding the collection of a national LiDAR data set for the Netherlands is unique currently. For that reason it is interesting to research the possibilities to generate derivative products such as raster-based DEMs and DSMs from the AHN2 data set.

3.2 PIPELINE GENERATION

The developments in LiDAR data acquisition technologies resulted in an explosive increase in volume of spatial data. This create new challenges in the design of algorithms to process point clouds. Point clouds can be defined as massive when the data size of the point cloud is larger then a computers' main memory; the transfer of the data between external storage and main memory becomes a performance bottleneck. In order to deal with this problem pipelining is the most common strategy for feeding data sequentially into the memory [Guan and Wu, 2010]. A pipeline is a stream of data that flows consecutively through all of the stages and can be processed step-by-step [MacDonald et al., 2004]. Isenburg et al. [2006a] detects different types of algorithms to process massive LiDAR point cloud data subsequently. All types of algorithms try to exploit or create spatial coherence; a correlation between the proximity in space of geometric entities and the proximity of their representations in the data.



(a)

(b)

(c)

Figure 19: Danish raster-based height map. (a) Preview of Denmark's digital surface model (own image, based on Septima [2014]). (b) Holes in the ground due to noise in the data (own image, based on Septima [2014]). (c) Occurrence of artifacts at building locations in a digital elevation model (own image, based on Septima [2014]).

Divide-and-conquer algorithms

Divide-and-conquer algorithm recursively breaking down a complex problem into multiple sub-problems (divide) until these become simple enough to be solved directly (conquer). The solutions to the sub-problems are then combined to give a solution to the original problem.

Tiling is the common term for the process of decomposing LiDAR point cloud data into smaller subsets of data. In section 2.3 it has been introduced that the AHN2 point cloud data is distributed in tiles in order to increase the accessibility of smaller parts of the data set. When processing tiled data sets directly after dividing the data, errors will appear near tile edges due to spatial decomposition. In order to give a solution to the original problem a buffered version of the tiled data can be generated. By generating some degree of overlap the results of the divided data are equal to that of the original data set.

The buffered divide-and-conquer strategy including is a often applied strategy for the generation of large-scale raster-based height maps from massive LiDAR point cloud data sets. Selection of a proper buffer size is key: the buffer size should be large enough in order to guarantee that the solution of the divided tiles are similar to the solution of the original data set. The overlap should not be too large since a larger overlap means more redundant memory usage [Guan and Wu, 2010]. Mitas and Mitasova [1999] discovered that from the perspective of empirical statistics that the number of points involved in interpolation is around a maximum of 10–30 meters in order to converge the final estimated value, so a proper buffer size should be around this size. For the generation of a 0.5 meter resolution raster-based height map for Switzerland, tiles are processed with a buffer size between 30-50 meters based on an input point cloud data set with an average point density of 1 point/m^2 [Luethya and Stengeleb, 2005]. Khosravipour et al. [2014] apply a buffer size of 25 meter for the generation of a 0.5 meter resolution raster-based CHM height map based on an input point cloud data set with an average point density of 160 points/m^2.

External memory algorithms

External memory algorithms use disks for temporary storage of data structures that do not fit in the computers' main memory and explicitly control data movement and data layout on disk with the goal of minimizing the number of disk accesses [Vitter, 2001]. Agarwal et al. [2006] presents a main memory efficient algorithm for the construction of DEMs, in which an out-of-core sorting algorithm is designed to minimize the total main memory time. The applied methodology consist of three steps. First, a quad-tree is constructed based on a set of points to partition the point set into a set of non-overlapping tiles. As second, for each tile and all adjacent tiles the set of points is calculated. Finally each segment is interpolated independently using points within the segment and its neighboring segments.

Basically this strategy uses adjacent tiles in a similar way as the buffers in the buffered divide-and-conquer strategy described in the previous paragraph. Comparable external memory algorithms are applied by Agarwal et al. [2005] for the construction of massive TINs and by Vitter [2001] for multiple geometric operations on massive data sets.

Streaming algorithms

Streaming algorithms are akin to external memory algorithms but do no swap at all in the meantime as they only use the external memory for input and output [Isenburg et al., 2006a]. Instead of loading a complete tile and all its neighboring tiles into the computers' main memory, only a part of the data is loaded into the main memory and when that part is finished it is written to the output file and removed from the computers' main memory. Therefore it is important to know what part of the point cloud data are finished and which ones not.

Isenburg et al. [2006a] applies the concept of spatial finalization for this as part of a three-step methodology for the generation of DEMs and DSMs via streaming TINs. First, bounding information is detected within the public header block of the input LAS file (see section 2.2). As second, the input file is decomposed in non-overlapping tiles and the points is count within each tile. This count is used as a finalization tag in order to indicate if all points are triangulated within a tile. As third, a triangulation algorithm certifies triangles as being Delaunay (see section 3.4) when the finalization tag shows it is safe to do so. In this way it is possible to write triangles to the output stream and so they can be removed from the computers' main memory in order to read more from the input stream. Only not-finalized parts within the triangulation process are resident in memory and for that reason, the memory footprint remains relatively small.

3.3 FILTERING

LiDAR records are more or less evenly distributed over the reflected surface, but no direct information about what type of surface was hit by each shot is available. In order to derive elevation models that only represent certain types of surfaces, it is needed that the different surfaces hit by LiDAR pulses are recognized and distinguished with respect to each other. Filtering is the process of assigning individual LiDAR records to surface classes so that in subsequent processing surface and object modeling may be based only on the points from relevant surfaces [Hug et al., 2004]. Effective and precise filtering of a point cloud is crucial to achieve high quality DEMs and DSMs [Liu, 2008b].

3.3.1 Point classification

First step in the filtering process is the definition of a classification system. The LAS standard provides the possibility for the storage of 32 filtered classes: 10 predefined, 22 reserved for future definitions [ASPRS, 2013]. A basic classification system often consist of the following classes:

- Ground
- Buildings
- Vegetation
- Noise

The *ground* class refers to all points which are related to bare ground, where the *building* class refers to all points which are related to buildings. In the *vegetation* class all trees with a non negligible size at a city scale (i.e. with a height of several meters) will be represented. All remaining points corresponds with outliers in the data and are classified as *noise*. Such points are reflected on small urban objects which temporarily perturb the scene (e.g. cars, roof antennas, cranes) and vertical structures like facades.

3.3.2 Filtering methods

Charaniya et al. [2004] qualifies two categories for the filtering of LiDAR point data:

- Filtering of LiDAR point cloud data into ground and non-ground points.
- Filtering of non-terrain LiDAR point cloud data into features as vegetation and buildings.

Current AHN2 point cloud products are filtered according to the first filtering category.

For both filtering categories different approaches exist. Most of them use geometrical relations between neighboring points in order to assign a classification. A number of tests and comparisons of different filtering algorithms have been performed but not many appropriate measures for the quality of filtering algorithms have been invented yet [Vosselman and Maas, 2010]. A comprehensive comparison of point cloud filters is compiled by Sithole and Vosselman [2005], more recent ones are made by Meng et al. [2010], Tinkham et al. [2011] and Podobnikar and Vrečko [2012]. General conclusion is that no point cloud filtering algorithm scores significantly best in general, different filtering algorithms score better in the filtering of certain objects and/or circumstances. It has to be taken into account that there does not exist any method that can guarantee a 100% correct classification of LiDAR points and also that results of classification algorithms strongly differ because of the characteristics of the point cloud data and the characteristics of the terrain.

Podobnikar and Vrečko [2012] concludes that the software package LAStools, containing multiple LiDAR processing tools gave appropriate results in general. In the remainder of this section two classification algorithms will be introduced that are part of LAStools; in subsection 3.3.3 an algorithm will be introduced for the classification of LiDAR point data into terrain and non-terrain points. In subsection 3.3.4 an algorithm will be introduced for feature classification of non-terrain LiDAR points.

3.3.3 Classification into terrain and non-terrain points

For the extraction of a bare-earth model a classification algorithm is needed that distinguish ground points from non-ground points. In this subsection a discrimination will be given of such a classification algorithm used by the LASground tool, part of the software package LAStools. Since there is no proper documentation available of the LASground classification algorithm no description can be provided.

Evaluation

In the technical description of the LASground tool it is stated that this algorithm has proven to work very well with a large variety of surfaces including mountainous areas with steep slopes and sharp ridges that often pose challenges to conventional approaches.

Podobnikar and Vrečko [2012] tested the LASground algorithm using the standard settings. The provided results were better compared to other filtering algorithms, the LASground algorithm performs worse near river beds in the used test data sets.

3.3.4 Feature classification of non-terrain points

Hug et al. [2004] provide a theoretical description of an automatic contour/segmentation based object-oriented classification algorithm for the filtering of point cloud data into above-ground features as vegetation and buildings. This algorithm is implemented within the LASclassify tool, as part of LAStools.

Object-oriented contouring

Main concept of this algorithm is the creation of horizontal segments by contouring; within a top-down approach contour lines are defined every 0.5 meter and for each elevation level new closed contours are searched.

Starting from each new found closed contour, a segment is defined: a coherent planimetric area delineated by one closed contour. For each defined segment, searched will be for segments at lower levels until the ground is reached. On subsequent (lower) levels, the number of segments will grow and grouped in objects, Hug et al. [2004] distinguished two kinds of objects:

- Primitive objects are defined by the lowest contour and all contours above that contain at most one contour at the next higher level.
- Complex objects are defined when multiple segments merge on a higher level, e.g. a building with two towers (Figure 20). Within this concept complex segments are the parents of their primitive children.

At the lowest elevation (root) level, one complex parent segment exists that contains all objects (both primitive and complex) within the entire point cloud (Figure 20). The objects can be defined as abstract entities representing hierarchies of enclosed contours. Although they do not have any real meaning with respect to the real-world objects they represent, the hierarchical description of objects facilitates searching for real-world objects significantly.

Object analysis

The next step is a top-down analysis of the object family tree from top level to root in order to determine if an object is part of a terrain structure or if it is an artificial structure that should be considered as belonging to terrain (e.g. dams). This analysis takes place by analyzing the shape of the objects and the growing behavior of its segments from level to level.

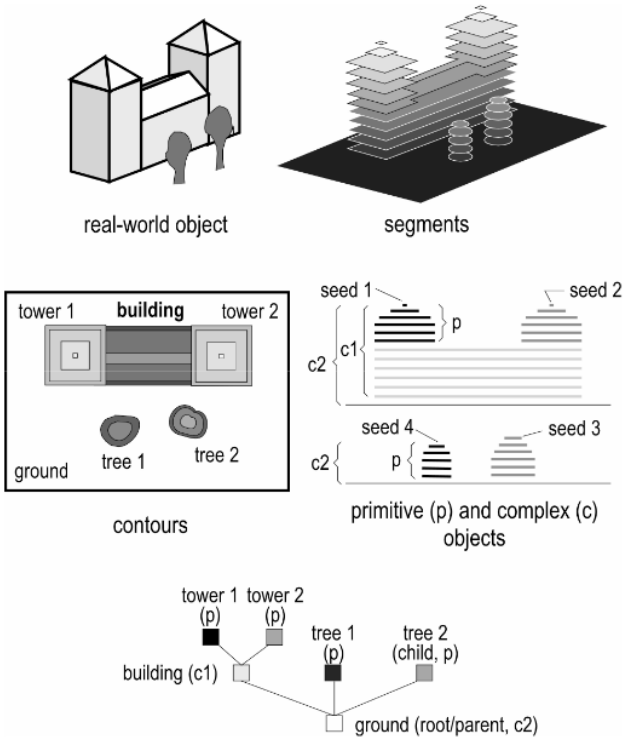


Figure 20: Object hierarchy [Hug et al., 2004]

In the same way a similar analysis is applied on above-ground objects. For example, a simple rectangular real-world house could be analyzed as: a small contour is detected that grows quadratically in an area from level to level into a rectangular shape. The object stops growing for several levels and suddenly grows again in large but random steps. The analysis is completed at root level where the segment starts growing randomly. Hug et al. [2004] introduce a number of potential criteria to determine real-world objects and on which elevation level they start:

- Object geometry
  - Area growth
  - Contour/segment shape
  - Relationship of area size and contour length
  - Relationship of volume (height) and area size
- Object context
  - Shape
  - Sizes
  - Growth behavior of adjacent objects and the parents and grandparents of multiple objects in the object tree
- Object attributes
- Several others

A fuzzy-logic-based classification that can be parameterized by training is used in order to detect real-world objects based on these criteria.

Evaluation

Hug et al. [2004] indicate that discrimination of artificial structures like dams and ramps that are usually considered as belonging to the ground class are reliable classified such while other structures of almost any size and shape are reliably identified as non-ground objects.

Besides distinguishing surface objects from ground, contour based object detection generates comprehensive information about the surface objects that can readily be used for further classification. Different types of above-ground objects can be identified and geometrical object descriptions (e.g. building geometry, roof shape, ridge orientation) can be derived with little additional effort by just evaluating the geometry of the abstract object. Complex buildings, for example are represented in the object tree as a parent object with multiple child objects representing building primitives. The geometries of parent and child objects structure the building and describe its geometry in detail.

3.4 SPATIAL INTERPOLATION

Spatial interpolation in digital elevation modeling is used to determine heights of neighboring locations where no height information is available. Two implicit assumptions here are that the terrain surface is continuous and smooth and that there is a high spatial coherence between the neighboring data points [Li et al., 2004].

LiDAR point clouds are not acquired on a uniform grid, they can be seen as a set S of n arbitrary points in 2-dimensional raster R^2 with an associated elevation function $h : S \rightarrow R$. To construct a raster-based DEM h has to be extended via interpolation to a uniform grid $G \subset R^2$ at the desired resolution [Beutel, 2011]. The height value for each raster cell represents the height in the middle of a raster cell.

3.4.1 Raster resolution

Before interpolation can take place it is important to determine an appropriate raster resolution for the output of interpolated data. The term resolution is often used for a pixel count in digital imaging. In case of DEMs and DSMs the term resolution refers to the grid size of the model with respect to the ground distance. The smaller this ground distance the higher is the resolution, representing a surface in more detail.

According to Liu [2008b] an appropriate raster resolution depends on source data density, terrain complexity and applications. McCullagh [1988] suggests that the number of grid cells should be correlated to the number of data points of an area. Hu [2004] introduced a formula to estimate the raster resolution as:

$$S = \sqrt{\frac{A}{n}} \tag{1}$$

Where n is the number of data points and A is the covered area. In this scenario each grid cell should contain one point on average. An appropriate grid size is ≈ 0.41 meter, based on an average minimal point density of at least 6 points/m^2 for the AHN2 data set.

Hengl [2006] introduced a formula for calculating the raster resolution based on the terrain complexity. According to this concept the raster resolution should be at least half the average spacing between the inflection points:

$$S = \frac{L}{2 * N_p} \tag{2}$$

Where L is the length of the transect and N_p is the number of inflection points as observed. Arcadis [2012] has calculated that the theoretical distance between two points for the AHN2 data set is 0.46 meter in the most pessimistic situation, based on a Voronoi method. In this scenario an appropriate grid size should be $\approx$ 0.23 meter.

Another criterion for the selection of an appropriate raster resolution is the application [Liu, 2008b]. A high raster resolution might significantly improve the predictive ability of terrain attributes, the choice of raster resolution for terrain based environmental modeling depends on the output of interest.

3.4.2 Spatial interpolation methods

A wide range of interpolation methods exist; in the next paragraphs the most often used interpolation methods in GIS will be introduced. Li et al. [2004] defines two classes of interpolation methods for the generation of DEMs or DSMs from LiDAR point cloud data:

- Deterministic interpolation methods
- Geo-statistical interpolation methods

Deterministic interpolation methods assume that each LiDAR point has a local influence. Values at different unsampled points are computed by functions with different parameters and the condition of continuity between these functions is defined only for some approaches. The method of point selection used for the computation of the interpolator differs among the various methods and their concrete implementations. The following deterministic interpolation methods will be introduced in this subsection:

- IDW
- Natural Neighbor interpolation (NNI)
- Interpolation based on a TIN
- Splines

Kriging is a geo-statistical interpolation method taking both the distance and the degree of auto-correlation (the statistical relationship among the sample points) into account.

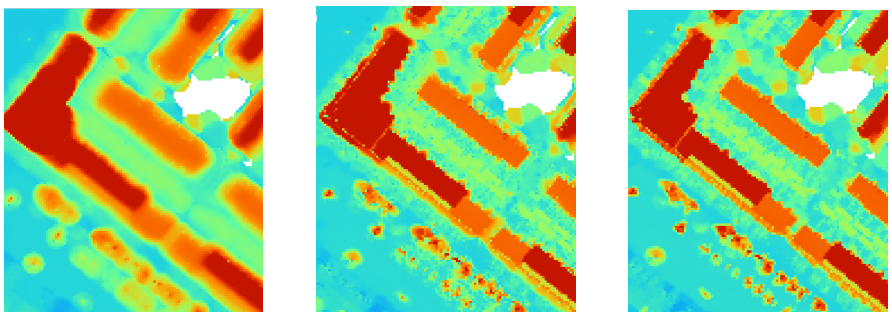
Inverse Distance Weighting

IDW explicitly implements the assumption that things located close to each other are more alike than those that are farther apart. This assumption is better known as the first law of geography as introduced by Tobler [1970].

In case of IDW interpolation the assigned values for unknown points are calculated with a weighted average where the applied weight is based on the distance to a known point; measurements close to the prediction location will have a higher influence on the predicted value than those farther away. Given a set of known LiDAR points IDW is a deterministic method for multivariate interpolation; interpolation takes place with more than one variable in a function:

- Power function
- Search radius
- Maximum number of points

The weight value of LiDAR points is proportional to the inverse distance raised to the power function: as the distance increase the weight decrease. The extend to which that takes place is dependent of the value of the power function. In case of a power value 1 there is no decrease in weight when distance increases and the calculated value will be the average (smoothed) of all measured points (Figure 21a). Figure 21 shows that an increasing power value the influence of farther away located points the weighting value decreases. A typical power value $u = 2$ [Watson, 2013].



(a) (b) (c)

Figure 21: The influence of different power values regarding inverse distance weighting. (a) Power value = 1. (b) Power value = 2. (c) Power value = 3.

Increasing point densities will increase the processing time. When distance d_i increases, the weight value of a point will have a lower relationship with the prediction location. A search radius d_i can select only those points that are within a Cartesian distance $d_{i,max}$, having a significant influence on the prediction value f_1P at point P. Another variable is a maximum number of i LiDAR points that might be taken into account for the prediction of the value f_1 at location $f_1(P)$; only the closest i points will be used. After determining these variables a height value f_1 at location P is calculated using the following expression:

$$f_1(P) = \begin{cases} \frac{\sum_{i=1}^{N} (d_i)^{-u} z_i}{\sum_{i=1}^{N} (d_i)^{-u}} & \text{if } d_i \neq 0 \text{ for all } D_i (u > o) \\ z_i & \text{if } d_i = 0 \text{ for some } D_i \end{cases} \tag{3}$$

Where f_1 is the value for point P, u is the power function and z_i is the value at data point D_i. d_i is the Cartesian distance between P and D, $d[P, D_i]$ [Shepard, 1968].

Natural Neighbor interpolation

NNI applies a weighted average from local data (neighbors) based on the creation of Thiessen (or Voronoi) polygons out of a discrete set of spatial points and assign a height value from each point to its corresponding Thiessen polygon.

Estimation of a height value at unsampled locations takes place by calculating the weighted average of nearby values where the weight of each value is determined by the area of the Thiessen polygon that is covered. Where IDW uses distance as a parameter for allocating a weight to a local value, NNI determines the weight of influence of nearby points based on their corresponding Thiessen polygon area. The basic equation for a bivariate 2D NNI function is:

$$G(x,y) = \sum_{i=1}^{N} w_i f(x_i, y_i) \tag{4}$$

Where $G(x,y)$ is the estimate at location (x,y), $f(x_i,y_i)$ are values nearby location (x,y) and w_i are weights of the nearby values based on their corresponding Thiessen polygons [Sibson, 1981]. Interpolation functions such as IDW and NNI might take into account the influence of close by points resulting in a different interpolated height, even at the location of a known point.

Interpolation based on a Triangulated Irregular Network

As introduced in section 2.1 it is possible to generate a secondary computed DEM of a primary measured DTM by gridding a TIN DTM.

First step in this process is to generate a TIN; a 2.5D triangulation based on the work of Peucker et al. [1978]. A common method for the construction of triangles within TINs is based on Delaunay Triangulation (DT), named after Delaunay [1934] for his work on this topic. Basic principle of DT is that given a set of P points in a plane, a triangular mesh, surface or triangular planes connecting the data points in a triangulation $DT(P)$ so that there will be no point P inside the circumcircle of any triangle in $DT(P)$ (see Figure 22b). For any set of points in two dimensions a DT is possible. A DT is always unique as long as no four points in the point set are co-circular.

DT is considered being a desirable approach for creating natural-looking surfaces because minimum interior angles of all triangles are maximized and triangles are as equiangular as possible, thus avoiding long and thin triangles [Pearlstone, 2010]. In this way the determined height at a certain point will be calculated by height sample points that are relatively close by. Given no other information but the sample points and assuming that the height at the sample points is correct, all triangulations can be equally good.

In situations with (steep) vertical elements (e.g. building walls) a TIN representation might give artifacts having no LiDAR points available both on top and down the vertical element. Due to the characteristics of topographic LiDAR near water bodies less points will be detected (section 2.2). Also in case of a DEM it might happen that locally less points are available where non-ground points are filtered (e.g. buildings). This might lead to the creation of triangles with long edges.

For both introduced problems a *constrained* DT might be a solution by generalizing the DT that forces certain required segments into the triangulation.

Given a set of points P and a set of segments S a *constrained* DT tries to achieve a triangulation of into the triangulation [Chew, 1989]. Given a set of points P a *constrained* DT is a triangulation that is as close to a DT under the constraint that all line segments in S become part of the DT as edges of the triangulation Figure 22c. Whereas a *constrained* DT contains edges that do not meet the Delaunay condition a *constrained* DT is often not a DT itself.

By subdividing a segment in multiple edges by adding extra vertices (Steiner points) to the original segment it is possible to construct a *conforming* DT. A *conforming* DT contains constraints and does also meet the condition of being Delaunay.

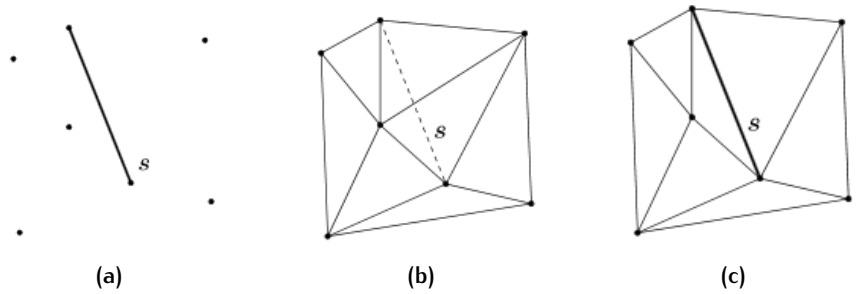


Figure 22: (a) A set of points P and a segment S. (b) A Delauney triangulation of point set P. (c) A *constrained* Delaunay triangulation of point set P and segment S [Agarwal et al., 2005].

Based on one of the TIN a grid can be derived using a bivariate function for each triangle in order to estimate height values at unsampled locations. Linear interpolation fits planar faces to each triangle individual. This might give a jagged appearance where it is visually possible to distinguish individual triangles. This is caused by discontinuous slopes at the triangle edges and sample data points (Figure 23).

Non-linear blended functions (e.g. polynomials) use additional information in first order (or both first- and second order) to derive a more smooth connection of triangles. After the generation of a TIN it is possible to rasterize the TIN into a regular grid regardless of the grid cell size or grid placement (Figure 24).

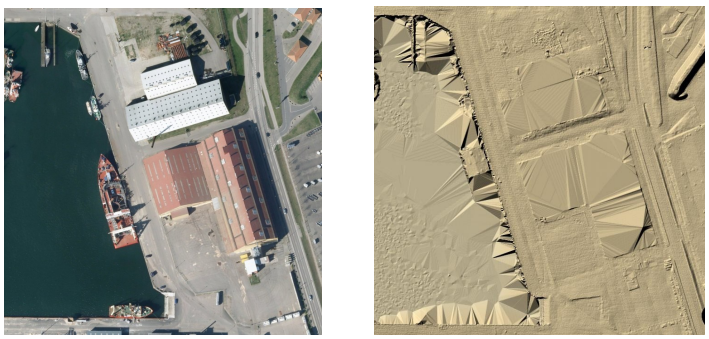


Figure 23: Digital terrain model generated by interpolation based on a triangulated irregular network. (a) An aerial photograph representing a harbor area (own image, based on Septima [2014]). (b) A digital terrain model representing the same area with clearly visually distinguishable triangles (own image, based on Septima [2014]).

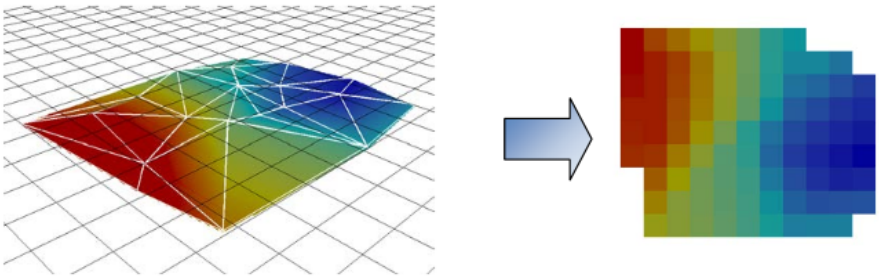


Figure 24: A regular grid overlain on a Delaunay surface to produce a raster file of height values [Pearlstone, 2010]

Kriging

Kriging estimates height values for locations with an unknown height using geo-statistical interpolation, derived from statistics. The interpolated values are modeled by a Gaussian process governed by prior covariances, as opposed to a piecewise-polynomial spline chosen to optimize smoothness of the fitted values. Explanation of this interpolation method is complex, information can be found in [Agterberg, 1974; Cressie, 1990].

3.4.3 Selection of a proper interpolation method

After introducing multiple interpolation methods, the question is what interpolation method is most appropriate in different contexts. Mitas and Mitasova [1999] poses several challenges for the selection of an appropriate spatial interpolation method:

- The modeled fields are usually very complex
- Data are spatially heterogeneous and often based on far from optimal sampling
- Significant noise or discontinuities can be present
- Data sets can be very large

Additional Mitas and Mitasova [1999] introduce a set of demands for a reliable interpolation tool should satisfy, suitable for GIS applications:

- Accuracy and predictive power
- Robustness and flexibility in describing various types of phenomena
- Smoothing for noisy data
- D-dimensional formulation
- Direct estimation of derivatives (gradients, curvatures)
- Applicability to large data sets
- Computational efficiency
- Ease of use

There does not exist any product that satisfy all these conditions for a wide range of geo-referenced data. Therefore the selection of a good interpolation method with appropriate parameters is the best possible.

3.4.4 Digital Elevation Model interpolation

Evaluation of performance of different interpolation methods has been assessed on DEM accuracy by Zimmerman et al. [1999]; Ali [2004]; Blaschke et al. [2004]; Lloyd and Atkinson [2002]; Chaplot et al. [2006]; Podobnikar [2005]. Zimmerman et al. [1999] showed that Kriging yielded better estimations of elevation than IDW did, especially when sampling density of points becomes sparse [Lloyd and Atkinson, 2002]. The result is probably due to the ability of Kriging to take into account the spatial structure of data [Chaplot et al., 2006]. However, if the sampling density is high, there is no significant difference between IDW and Kriging methods [Chaplot et al., 2006]. Ali [2004]; Blaschke et al. [2004]; Podobnikar [2005] pointed out that the IDW method performs well if sampling data density is high, even for more complex terrains. Liu et al. [2007] states that LiDAR data does have high sampling density in general, and so the IDW approach is suitable for the generation of DEMs from point data.

It is inappropriate to generate a high resolution DEM with sparsely distributed LiDAR data: any surface generated in such a way is more likely to represent the shape of the specific interpolator used than that of the target terrain because interpolation artifacts will abound [Liu and Zhang, 2008], this might lead to erroneous interpolated data. Kraus and Otepka [2005] showed the benefits of using a hybrid model for DEMs. This approach employed a vector-based TIN model for complex geomorphologic areas and a raster-based model for simple areas. The degree of complexity of the terrain could be determined based on the length of the edges within the TIN model. Isenburg et al. [2006b] introduce with the las2dem tool, as part of LAStools, a method to generate a raster-based DEMs via TIN streaming, this hybrid methodology combines the advantages of both vector-based and raster-based methods to store and process information.

Sink filling

In flat areas the accuracy of the single point is critical for water management and flood risk modeling. But due to the characteristics of topographic LiDAR a correlation can be detected between water bodies and sparser distributions of LiDAR data. Hydrological conditioned DEMs are required in situations in which a sound representation of the flow network for calculating flow-related quantities is necessary [Bailly et al., 2006; Davies et al., 2008]. Mark and Aronson [1984] performed a moving average window to remove small depressions. This approach, fails in eliminating larger sinks because it alters the entire DEM and may even generate new sinks along drainage pathways with steep side walls [Reuter et al., 2009].

A variety of different preprocessing techniques to ensure coherent networks of water have been developed termed *hydrological conditioning*. One of the hydrological conditioning techniques is stream burning which uses mapped stream locations to artificially lower stream cells in a DEM. Stream burning is particularly useful for the precise location of streams in low gradient landscapes such as coastal areas [Maidment, 1996] but it requires digitized stream maps which may often be unavailable at the desired map scale.

The most common hydrological conditioning technique is sink filling. Sink filling elevates pixel values in topographic depressions so that each pixel in a DEM has at least one neighboring pixel with the same or lower elevation. Sink filling can create large and contiguous areas of flat water bodies.

3.4.5 Digital Surface Model interpolation

A common method to develop a DSM is by interpolating all first returns representing a surface elevation [Vögtle and Steinle, 2003; Bartels et al., 2006]. In subsection 2.4.1 and subsection 2.4.2 it was introduced how a similar method based on the average of all points for each raster cell is applied for the generation of a raster-based DSM based on the AHN2 data set.

A common error of such a strategy is the presence of data pits within the generated raster data. These pits are visible as dark holes that are digitally represented by exceptionally lower digital height values than their neighbors. It is believed that these artifacts are caused by a combination of factors, from data acquisition to post-processing, though no specific cause has been defined in the literature [Ben-Arie et al., 2009].

As introduced in subsection 3.4.4, interpolation is based on the assumption that the terrain surface is continuous and smooth, and that there is a spatial coherence between the neighboring data points [Li et al., 2004]. Where urban features and vegetation have specific characteristics in terms of both elevation and slope, spatial interpolation will introduce errors in a DSM. Priestnall et al. [2000] shows that errors near building surfaces are practically the same for different interpolation methods. For that reason, a common method is to process above-ground features as building and vegetation objects separately after the classification of point records into these classes (subsection 3.3.1). Research on the generation of DBM from LiDAR data is been done by Palmer and Shan [2002]; Cho et al. [2004]; Alexander et al. [2009]. Research on the generation of CHMs from LiDAR data has been done by Clark et al. [2004]; Popescu et al. [2002]; Khosravipour et al. [2014].

3.4.6 Digital building model interpolation

Different then DEMs which require continuous interpolation, buildings have irregular shapes and there is no correlation between different buildings; interpolation should take place on building level. Most work on the extraction of building information is based on the extraction of building data from raster-based data after interpolation of the point cloud data. Palmer and Shan [2002]; Cho et al. [2004] state that such methods introduce unwanted errors into the data by creating incorrectly smoothed heights for the building edges. Cho et al. [2004] introduces a concept of pseudo-grid (or binning) into raw laser scanning data to avoid a loss of information and accuracy due to interpolation, but such a methodology is not capable to distinguish trees from buildings. Alexander et al. [2009] states that the use of building footprint polygons offers a potential solution to these problems; an open data set regarding buildings is used to determine the presence of buildings and use them as breaklines.

Interpolation based on a TINs are widely is considered as highly suitable interpolation method for the interpolation of buildings since data is not smoothed by definition; this interpolation method is able to represent multiple segments in a correct way. Another advantage of interpolation based on a TINs is the possibility to add constraints. Within a *constraint* DT it possible to remove redundant edges, a part of which are outside the building boundary [Hu, 2004].

3.4.7 Canopy height model interpolation

Similar as for the generation of DBM standard procedures for the generation of a CHM are focused on the subtraction of a DTM from a DSM. Such a procedure is described in Clark et al. [2004]; Popescu et al. [2002]. The disadvantage of this method is that it does not solve the problems regarding data pits.

The depth and the distribution of pits in a CHM depend on the crown structure and the diameter of the laser beam as well the sensitivity of the system processing the returning waveform [Gaveau and Hill, 2003]. Instead of hitting the highest point of the canopy, the laser pulses may produce their first return when they hit a lower branch or even after they penetrate all the way through the crown to the ground [Khosravipour et al., 2014]. Hence, the depth of different canopy pits varies greatly, making it impossible to use a fixed threshold to define and potentially remove them [Ben-Arie et al., 2009].

Khosravipour et al. [2014] proposes a pit-free methodology that height-normalize filtered vegetation LiDAR points first and then generated a raster-based CHM by applying interpolation based on a TIN. In order to solve the problem regarding data pits partial CHMs are created to determine the shape of the canopy at different heights and merge them finally.

3.5 POST PROCESSING

Smoothing and resampling of raster-based data are both commonly performed on surfaces before they are suitable for analysis [Bater and Coops, 2009].

3.5.1 Raster resampling

Resampling is the process of transforming a discrete image which is defined at one set of coordinate locations to a new set of coordinate points [Parker et al., 1983]. Downsampling is the process of transforming a discrete image which is defined at one set of coordinate locations to a new set of coordinate points with a lower resolution.

McInerney and Kempeneers [2015] compare and illustrate different resampling methods by using open source software package Geospatial Data Abstraction Library (GDAL). Concluded is that the resampling methods bilinear and cubicspline result in smoother results for resampling as well as for downsampling.

3.5.2 Raster smoothing

Smoothing is the process of approximating the capture of important patterns in the data and removing noise. It is often an iterative process, comparing a point with nearby points and adjusting its elevation [Tao and Hu, 2001]. Usually a best-fit facet model is computed for a group of points and the elevation of the center point is adjusted to better match the facet [Wang et al., 2001]. Since un-autocorrelated errors are the major cause of the numerous small depressions in LiDAR data it appeared that some degree of smoothing is beneficial [Li et al., 2011].

Smoothing does not only remove errors but it modifies potentially every height value. Excessive smoothing could lead to the modification and elimination of real topographic features within a DEM or DSM and should be avoided. This can be achieved by setting appropriate area and depth thresholds. No single values of area and depth thresholds are best in all cases [Li et al., 2011].

3.5.3 From small-scale raster data to large-scale raster data

In section 3.2 different methods of spatial decomposition are introduced. After processing of these smaller subsets they can be composed in order to regenerate one single raster image, two approaches are distinguished:

- Mosaic images
- Virtual Raster (VRT)

Mosaic Images

Mosaics use two or more input images to create a single output image.

Virtual raster

VRTs do not contain actual pixel values of the raster cells. Instead, these virtual files describe in a Extensible Markup Language (XML) format the characteristics of the input raster images. Characteristics that are stored in a VRT are:

- Name and path of the input raster file
- Number of bands
- Lines and columns
- Projection information

McInerney and Kempeneers [2015] describe the benefits of VRTs over mosaic images. VRTs can easily be edited to modify mappings, add attributes such as color tables and meta data or perform raster operations. In case of consecutive raster operations, the actual writing of the pixel values can be postponed until the end. This avoids reading and writing of temporary files, which increases efficiency. Virtual rasters can also be useful when you need to access raw binary raster files for which no GDAL driver exists: it is needed to describe the structure of the binary raster in the VRT file, such as: length of the header in bytes, data type, band encoding and byte order (most or least significant bit first). Virtual formats in XML support the description of algorithms to be applied to the raster data. Finally, a VRT also saves considerable disk space in comparison to mosaic images.

3.6 RASTER VISUALIZATION

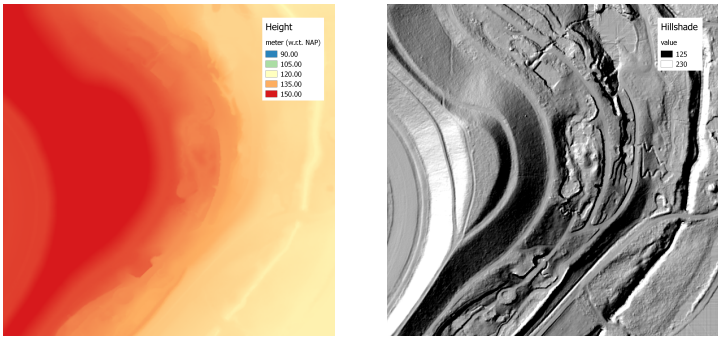
Visualization in GIS, better known as Geographic Visualization or Geovisialization, is a set of tools and techniques which support geospatial data analysis through the use of interactive visualization.

3.6.1 Hypsometric tinting

Hypsometric tints are colors used to indicate elevation. Ranges of elevation are indicated by bands of a color, often gradually, or as a color ramp to contour lines. A typical scheme progresses from dark greens for lower elevations up through yellows/browns, and on to grays and white at the highest elevations. Regarding information about underwater heights it is possible to apply *bathymetric* tinting in order to visualize depths; lighter shades of blue indicate shallow water and deeper water are tinted darker. Figure 25a shows an example of hypsometric tinting.

3.6.2 Hill shading

Hill shading is a hypothetical illumination of a surface according to a specified azimuth and altitude for the sun based on the work of Horn [1981]. It creates an effect that provides an optical relief for 2D cartography of terrain in a three dimensional 3D appearance [Robinson, 1960]. Figure 25b shows an example of a hill shade map.



(a) (b)

Figure 25: The optical effect of adding hill shade to a raster-based height map (a) A raster-based digital elevation model visualized using hypsometric tinting. (b) A hill shade map calculated from the digital elevation model.

Definition of the gray value of each surface unit is determined as the ratio between the cosine of the angle between a surface normal vector and a vector representing the illumination direction [Horn, 1981]. Burrough [1986] provides an explanation to perform a hill shade calculation.

Hill shade values are partially calculated based on the height difference of adjacent raster cells; for this reason the hill shade effect is determined on a local scale. No real shadows are determined within a hill shade map. In order to extend the hill shade to provide real shadows both the effects of local illumination angle and height data of farther located raster cells should be considered.

3.6.3 Image overviews and pyramids

In subsection 3.5.3 two approaches have been introduced to compose one single output raster image from smaller tiles. When the size of raster data increases the efficiency of data for purposes such as visualization decrease. Image overviews and pyramids are techniques to view images more efficiently [McInerney and Kempeneers, 2015].

Image overviews

An image overview is a downsampled version of an original raster image. The overviews can be located in external files or, for some image formats, be included within the image file itself. Image overviews are typically used to display reduced resolution overviews more quickly than could be done by reading the full resolution data followed by downsampling [McInerney and Kempeneers, 2015].

Image pyramids

Building image pyramids is a technique that combines image overviews with tiling for raster images. In this way image pyramids are a predecessor to scale-space and multi resolution analysis. In Figure 26 an image pyramid is presented: it contains a raster data set of 4 096 x 4 096 pixels at four scales. At each scale, the image is divided in a number of tiles, where each tile has the same number of pixels, in this example 256 x 256 pixels. At the original spatial resolution (level 0), a maximum of tiles is needed to represent the entire image. At the coarsest resolution (level 4), the image can be represented by a single tile.

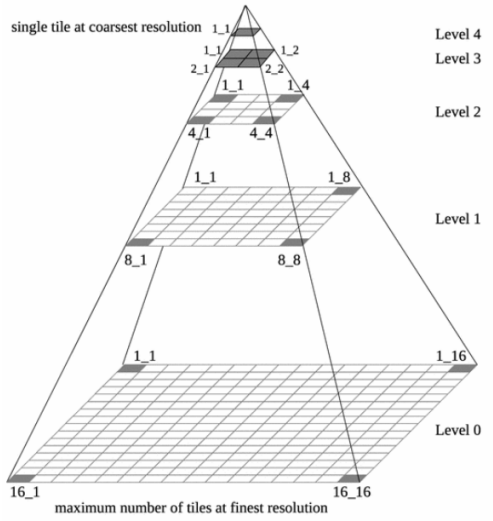


Figure 26: Low pass pyramid with four levels for an image of 4 096 x 4 096 pixels. The tile size is 256 x 256 pixels [McInerney and Kempeneers, 2015].

Pyramid generation kernels

When the viewing level (the scale) increase raster data is displayed on a lower level on the screen it is not needed to show all available details at once [McInerney and Kempeneers, 2015]. Therefore it is recommended to apply smoothing kernels during the generation of image pyramids; multiple height values will be resampled to one new height value by interpolation.

Bilinear interpolation is a good and fast method for continuous data, such as elevation. This interpolation method uses the weighted average of four nearest raster cells in the source image to estimate new height values for the destination image. Where the application of raster pyramids is meant for alternative representation of the raster data, the data at base level (level 0) will not be changed.

3.6.4 Raster visualization for quality assessment

Luethya and Stengeleb [2005] explain the role of raster visualization within quality assessment after the generation of a countrywide DEM for Switzerland. Various raster-based maps were generated for a visual inspection. Quality control regarding DEMs was done by the generation of multi-resolution point density maps, contour lines, hill-shaded DEMs, slope grids and the difference between DSMs and DEMs.

3.7 QUALITY ASSESSMENT

Data quality is a pillar in any GIS implementation and application of reliable data are indispensable to allow the user to have meaningful results [Srivastava, 2008]. Since a DEM or DSM is an approximation of the reality, based on a nominal ground [Podobnikar, 2009], all spatial data are at different levels, vague, incorrect, old or incomplete [Devillers and Jeansoulin, 2006]. Data quality refers to the performance of the dataset given the specification of the data model [Haining, 2003] or the degree of data excellency that satisfy the given objective [Srivastava, 2008]. Quality is the totality of characteristics of a product that bear on its ability to satisfy stated and implied needs [ISO, 2013]. In *ISO19157: Geographic Information Quality principles* five elements for data quality are described:

- Completeness
- Logical consistency
- Positional accuracy
- Temporal accuracy
- Thematic accuracy

All elements provide quantitative quality information about a data set.

Completeness

Completeness expresses the presence and absence of data, their attributes and relationships. There are two sub elements: commission (excess data present) and omission (data absent) [ISO, 2013]. This definition requires a precise description of the abstract universe since the relationship between the dataset and the abstract universe cannot be ascertained if the objects in the universe cannot be described. The abstract universe can be defined in terms of a desired degree of abstraction and generalization (i.e. a concrete description or specification for the database). This leads to the realization that there are in fact two different types of completeness [Veregin, 1999].

'Data completeness' is a measurable error of omission observed between the database and the specification. Data completeness is used to assess data quality, which is application-independent. Even highly generalized databases can be complete if they contain all of the objects described in the specification [Veregin, 1999].

'Model completeness' refers to the agreement between the data set specification and the abstract universe that is required for a particular database application [Guptill and Morrison, 2013]. Model completeness is application-dependent and therefore an aspect of fitness-for-use. It is also a component of 'semantic accuracy' [Salgé, 1995].

Additional distinctions are required. The definitions of completeness given above are examples of 'feature or entity completeness'. In addition we can identify 'attribute completeness' as the degree to which all relevant attributes of a feature have been encoded. A final type of completeness is 'value completeness' which refers to the degree to which values are present for all attributes [Guptill and Morrison, 2013].

Logical consistency

The logical consistency is a degree of adherence to logical rules of data structure, attribution and relationships [ISO, 2013]. For geospatial data the term is used primarily to specify conformance with certain topological rules [Salgé, 1995]. Spatial relations describe the spatial integrity of a geospatial data set. Spatial integrity constraints are a tool for improving the internal quality of spatial data [Devillers and Jeansoulin, 2006].

The identification of an inconsistency does not necessarily imply that it can be corrected or that it is possible to identify which attribute is in error. Note also that the absence of inconsistencies does not imply that the data are accurate. Thus consistency is appropriately viewed as a measure of internal validity. Despite the potential to exploit redundancies in attributes, tests for logical consistency are almost never carried out [Veregin, 1999].

Positional accuracy

The ISO [2013] describes accuracy as a closeness of agreement between a test result and the accepted reference value, in case of positional accuracy it refers to the accuracy of the spatial component of a dataset. There are three sub elements:

- absolute or external accuracy
- relative or internal accuracy
- gridded data position accuracy

In case of raster data the latter one, gridded data position accuracy, the closeness of provided data position values to values accepted as or being true is the component of interest. Expression of this accuracy is by calculating the Root Mean Square Error (RMSE). The RMSE is not the same as the standard deviation of a statistical sample, because the value of the RMSE is calculated from a set of check measurements [Huisman and Rolf, 2009]. RMSE is commonly used to document vertical accuracy for DEM. RMSE is a measure of the magnitude of error but it does not incorporate bias since the squaring eliminates the direction of the error [Veregin, 1999].

Temporal accuracy

Temporal accuracy refers to the agreement between encoded and 'actual' temporal coordinate system [Veregin, 1999]. It is the discrepancy between the actual attributes value and coded attribute value. A value is actual if it is correct in spite of any possible time-related changes in value. Thus currentness refers to the degree to which a database is up to date [Redman, 1992]

Another impediment to the measurement of temporal accuracy is that time is often not dealt with explicitly in geospatial databases. Temporal information is often omitted, except in databases designed for explicitly historical purposes. This assumes that observations are somehow 'timeless' or temporally invariant. The implications of this omission are potentially quite significant, especially for features with a high frequency of change over time [Veregin, 1999].

Thematic accuracy

Thematic (or attribute) accuracy compares the classes assigned to a feature or their attributes to a reference dataset or ground truth [ISO, 2013]. Quantitative attributes can be conceived as statistical surfaces for which accuracy can be measured in much the same way as for elevation [Veregin, 1999].

The check can be done by making use of a *confusion matrix* or *error matrix*. The matrix contains additional information on the frequency of various types of misclassification, e.g. which pairs of classes tend most often to be confused. In addition, the matrix permits assessment of errors of omission (omission of a location from its 'actual' class) and errors of commission (assignment of a location to an incorrect class) [Veregin, 1999].

3.8 DISCUSSION

In this chapter related projects and work are identified and introduced with respect to the different steps for the generation of a raster-based height map from massive point cloud data.

In section 3.2 different methods for pipelining were introduced. The main matter is that the different pipelining methods are focused on an efficient usage of a computers' main memory. As long the processed data size does not exceed the size of a computers' main memory there is no problem regarding the processing of massive point cloud data within the scope of this thesis. Chosen is to adopt a divide-and-conquer tiling algorithm.

In section 4.2 the concepts of filtering and classification of point records are treated; most packages have capabilities for the filtering of ground, buildings and vegetation classes. No filtering algorithm can guarantee a 100% correct classification. For one software package, LAStools, two algorithms for the filtering and classification of point records are presented and evaluated.

In section 3.4 theory about the determination of a proper raster resolution and spatial interpolation methods are introduced. Different objects require different methods of interpolation; for the generation of DBMs and CHMs interpolation based on a TIN is a reasonable method. In case of a raster-based DEM, a hybrid method using interpolation based on a TIN models in combination with methods for sink filling and IDW interpolation is indicated as a proper strategy.

In section 3.5 relevant work regarding resampling and smoothing are provided. In subsection 3.5.3 the advantages of composing tiles into a VRT are described. In section 3.6 related concepts for the visualization of raster data are treated. Proper multi-scale raster visualization can be achieved with the generation of image pyramids, a combination of tiles and image overviews. In this way the concept of tiling can be applied integrally within the whole processing chain.

4 METHODOLOGY

In this chapter a methodology will be proposed for the generation of a raster-based height map from massive point cloud data. Within this methodology, three models will be generated:

- DEM
- DBM
- CHM

Each model does have an individual procedure, Figure 27 shows the work flow of the methodology that will be proposed within this chapter. The different models can be used in an interchangeable way in order to combine only data of interest.

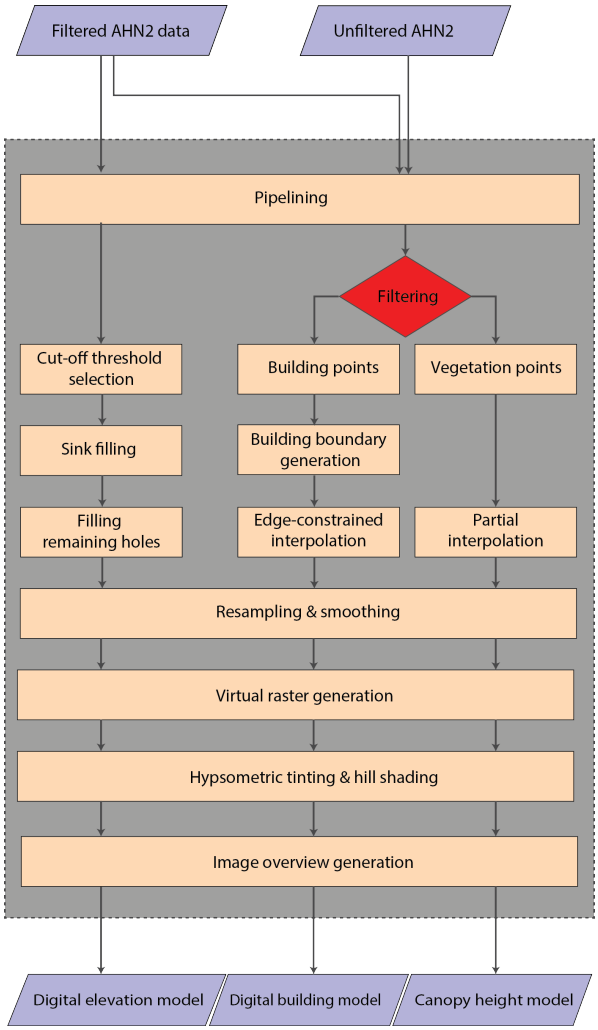


Figure 27: Work flow for the processing methodology as introduced in this chapter.

4.1 PIPELINING

The first step in the generation of a raster-based height map is the collection of the input data and the definition of a pipeline in order to feed data sequentially to the computers' main memory. Related work with respect to pipelining is introduced in section 3.2. In section 3.8 it has been evaluated that the adaption of a divide-and-conquer tiling algorithm is advantageous.

4.1.1 Obtaining the input data

Before pipelining the point cloud data, in this subsection relevant information will be provided with respect to the collection and optionally merging of the input tiles covering the target area and clipping them with respect to the target area boundary.

Collection and merging the input tiles

First step is the collection of input point cloud tile(s) covering the target area. One or more tiles might be needed in order to cover the target area which can be downloaded from the website of the Dutch SDI PDOK[1] as open data. In case of multiple tiles covering a target area the tiles can be merged with LASmerge, as part of LAStools, via a command prompt line:

```
$ lasmerge -i input1.las input2.las -o merged.las
```

Where flag *-i* defines the input tiles of interest and the flag *-o* the name and location defines where the combined tiles will be stored. This step needs to be applied twice; one time for the filtered tiles and another time for the unfiltered tiles. In case of a scenario where the target area can be covered with only one tile this step can be skipped.

Clipping the target area

After collecting and merging of the point cloud data covering the target area, next step is to clip the point cloud data with respect to the target area. LASclip, as part of LAStools, can clip all point located outside a predefined shapefile and store surviving points to a new point cloud file via a command prompt line:

```
$ lasclip -i merged.las -poly convexhull.shp  -o clipped.las
```

Where flag *-i* defines the input point cloud data, the flag *-poly* defines a polygon representing the outer boundary of the target area and the flag *-o* the name and location defines where the clipped point cloud data will be stored. Similar as for merging of the tiles, this step needs to be applied twice; one time for the merged filtered point cloud data and another time for the unfiltered point cloud data.

It needs to be taken into account that in order to achieve correct data for the target area, data can be better processed for a larger area rather then only the target area in order to deal with artifacts near the borders of the target area. Relevant information with respect to the selection of a proper buffer size will be introduced in subsection 4.1.2.

1 https://www.pdok.nl/nl/producten/pdok-downloads/atomfeeds/a

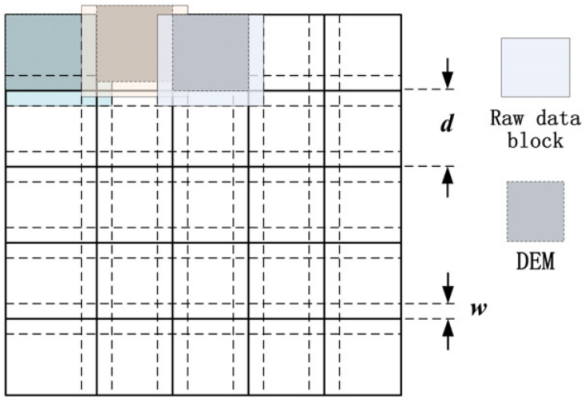


Figure 28: Decomposition and interpolation of overlapping point cloud tiles [Guan and Wu, 2010].

4.1.2 Tiling

Tiling is the reorganization and storage of LiDAR point cloud data into contiguous regular tiles. Data is split up into a sequence of discrete overlapping tiles (Figure 28). While processing a tile there is no involvement of information located outside the tile within the remainder of the pipeline. Two variables have to be determined within the tiling process:

- a tile size t
- a buffer size b

Where tile size t is used to control the granularity of the parallel pipelines and buffer size b is used in order to guarantee that the results of the spatial decomposition are equal to that of the massive non-tiled data set. LAStile, as part of LAStools, is an automatic tiling algorithm that can run via a command prompt line:

```
$ lastile -i clipped.las -tile_size t -buffer b -o tile.las
```

Where flag *-i* defines the input point cloud data, the flag *-tile_size* defines a tile size t, the flag *-buffer* defines a buffer size b and the flag *-o* the name and location defines where the tiled point cloud data will be stored.

Tile size

For the selection of a proper tile size just considering the file size of the point cloud data is not enough. During different steps within the pipeline the file size might increase, resulting in the possibility that a file size might exceed the computers' main memory size in a later stage of the processing pipeline. Some steps that increase the data size within the processing procedure are:

- Meta data that will be added to the point records (e.g. classification)
- (Post-)processing of raster-based height data

Selection of a proper tile size is strongly dependent on the size of the computers' main memory and the process that will make use of the computers' main memory most intensively within the pipeline. For this reason it is impossible to define a value for the tile size based on the file size of the input point cloud data. A trial and error method seems to be the best possible strategy for the selection of a proper tile size.

Buffer size

In section 3.2 the need for a buffer within divide-and-conquer strategy is mentioned; some degree of overlap within the divided point cloud will result in output data that is equal to that of the original data set.

Considering the buffer distances applied in related work (see section 3.2), a buffer distance of 25 meters is supposed to provide proper results. When testing, it can be considered that a buffer distance of 25 meter results in output data that is near-similar with respect to the original data set, after application of the remainder of the processing pipeline. Height differences with respect to raster data interpolated from the original data set are below millimeter level, both for ground as for above-ground points.

4.2 FILTERING

Filtering is the process of distinguishing individual 3-dimensional point records and the assignment to predefined surface classes so that in subsequent processing surface and object modeling may be based on points from relevant classified surfaces [Hug et al., 2004].

In section 3.3 relevant work regarding filtering has been introduced. Main conclusion of this section is that no method exist that can guarantee a 100% correct and automatic classification of LiDAR points. The results of filtering strongly differ based on the characteristics of input data and terrain characteristics.

In this section the potential of an automated filtering algorithm will be tested; in subsection 4.2.1 an automated filtering procedure for the classification of ground points is tested. In respectively subsection 4.2.3 and subsection 4.2.4 an automated filtering procedure for the classification of buildings and vegetation is tested.

4.2.1 Ground

For the filtering of ground points it is necessary to apply a classification of LiDAR point cloud data into ground and non-ground points. The LASground tool, as part of LAStools, has been introduced in subsection 3.3.3. This automatic classification algorithm filters ground point records from non-ground point records via a command prompt line:

```
$ lasground -i input.las -o classified_terrain.las -step_size
```

Where flag *-i* defines the input point cloud data and flag *-o* defines the name and location defines where the classified point cloud data will be stored. For the *-step_size* flag different parameters can be selected representing different search distances:

- Forest - 5 meters
- Town - 10 meters
- City - 25 meters
- Metropolis - 35 meters

The LASground tool will be tested for two point cloud products based on the AHN2 data set:

- Filtered AHN2 point cloud data
- Merged filtered and unfiltered AHN2 point cloud data

For both point cloud products the LASground tool will be tested using multiple search distances.

Classification of filtered AHN2 point cloud data into ground and non-ground points

Application of LASground using the forest parameter results in a small percentage of point records that are classified as non-terrain, mainly randomly distributed over the terrain (Figure 29a).

Increasing the step size (town parameter) results in an increase of the number of point records classified as non-terrain; distribution of these point records is mainly near areas with some degree of slope (e.g. water and tunnels, see Figure 29b).

A further increase of the step size (city parameter) will increase the number of point records classified as non-terrain further. The distribution of detected non-terrain point records clusters near areas having some degree of slope in such a way that large parts of dikes are classified as non-ground points (Figure 29c).

A maximization of the step size (metropolitan parameter) increases the number of point records classified as non-terrain again. Classified non-terrain point records cluster further in a way that complete (infrastructural) dikes are classified as non-ground point records (Figure 29d). Table 1 provides an oversight of the statistical results for all scenarios.

Classification of filtered and unfiltered AHN2 point cloud data into terrain and non-terrain points

Combining the filtered and unfiltered AHN2 point cloud data, at first sight the results appear to be nearly similar with respect the filtering of only the filtered AHN2 point cloud data: most above-ground point records are filtered out properly (Figure 30).

A big difference for all scenarios is an increase of the amount of points classified as ground with approximately 60% (Table 1). When looking at a higher scale it appears that point records reflected on small buildings and larger buildings with flat roofs are often falsely classified as ground points (Figure 31).

	Filtered			Filtered + unfiltered		
	Ground	Non-ground	%	Ground	Non-ground	%
Forest	128 912 113	16 753	$0.1 * 10^{-3}$	205 230 896	128 435 855	0.38
Town	128 890 519	38 347	$0.3 * 10^{-3}$	203 882 641	129 784 110	0.39
City	128 632 683	296 183	$2.3 * 10^{-3}$	202 717 837	130 948 914	0.39
Metropolis	127 315 041	1 613 825	$12.5 * 10^{-3}$	201 320 742	132 346 009	0.40

Table 1: Points classified as ground and non-ground for both filtered and filtered + unfiltered AHN2 point cloud data for a random sample of 128 928 866 filtered points and 204 737 885 unfiltered points.

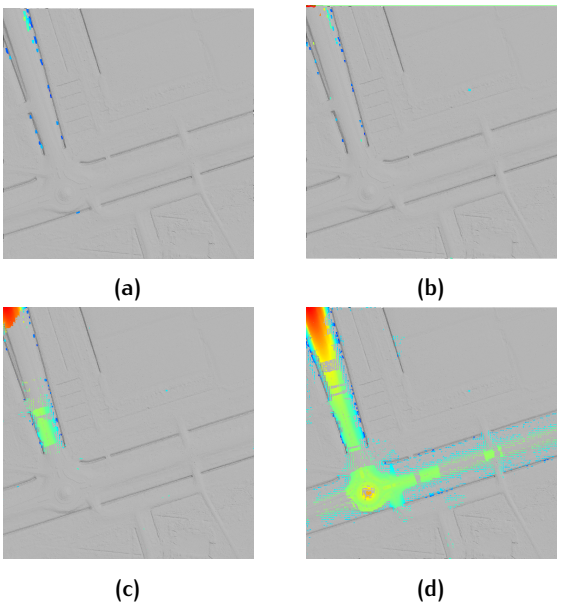


Figure 29: Points classified as non-ground points by LASground for different step_size parameters. Blue colored points indicate non-ground points with a relative lower Z-value and red colored points indicates non-ground points with a relative higher Z-value. The gray colors represent a gridded interpolation of a triangular irregular network of classified ground points. (a) Forest parameter. (b) Town parameter. (c) City parameter. (d) Metropolis parameter.

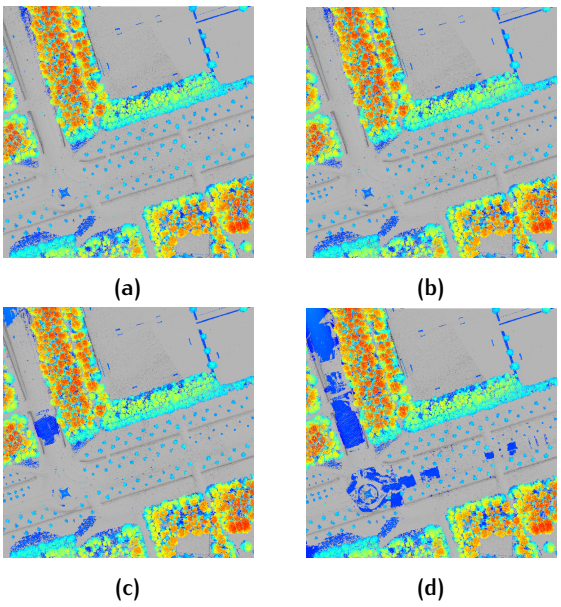


Figure 30: Points classified as non-ground points by LASground for different step_size parameters. Blue colored points indicate non-ground points with a relative lower Z-value and red colored points indicates non-ground points with a relative higher Z-value. The gray colors represent a gridded interpolation of a triangular irregular network of classified ground points. (a) Forest parameter. (b) Town parameter. (c) City parameter. (d) Metropolis parameter.

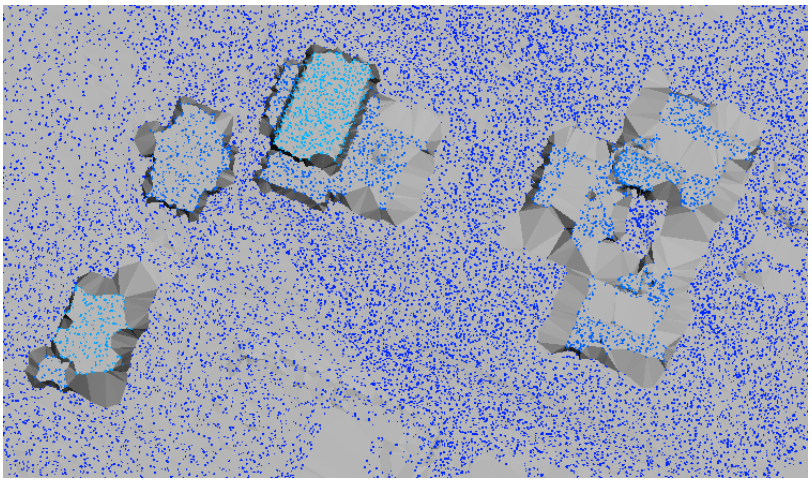


Figure 31: A digital elevation model stored as a triangular irregular network containing classified ground point records. A number of point records are wrongly classified as ground while being reflected on buildings.

Evaluation

Application of LASground after combining filtered and unfiltered AHN2 point cloud data does not provide a proper filtering of ground points; LiDAR points reflected on small buildings and larger buildings with flat roofs are often classified as ground points (Figure 31).

When using only the filtered AHN2 point cloud data basically two extreme scenarios can be distinguished:

1. The forest scenario leads to a small amount of points that are classified as non-ground points without a loss of significant features and;

2. The metropolis scenario leads to a larger amount of points that are classified as non-ground points; a loss of significant features such as dikes is the result.

For both extremes there is no significant improvement within the data set; the forest scenario will lead to the loss of a relatively small amount of LiDAR points and the metropolis scenario will lead to a loss of significant features in the filtered AHN2 point cloud data. Two other scenarios (town and city) are gradual variants in between both extreme scenarios.

LASground provides additional parameters in order to improve *local* filtering capabilities. It can be expected that additional parameterization can help in order to improve filtering capabilities at a local scale. The opposite might happen at a lower scale; it can be expected for an one size fits all-algorithm application of additional parameters might lead to worsened results.

It can be concluded that none of the filtering procedures provide a significant better product representing ground point records. For that reason chosen is to use the filtered AHN2 point cloud data without filtering as input point cloud data for the remainder of the processing pipeline for the generation of a raster-based DEM.

4.2.2 Preprocessing of above-ground objects

Preprocessing is needed in order to normalize of above-ground points with respect to the underlaying ground before it is possible to classify above-ground points. First step is the definition of the underlaying ground. With the las2las tool, as part of LAStools, it is possible to classify the point records within the filtered data set as ground (classification = 2):

```
$ las2las -i [filtered.las] -set_classification [2] -o [ground.las]
```

After assigning a classification to the ground point records, the second step merges the classified ground point records with the still unclassified unfiltered AHN2 point cloud data with LASmerge, as part of LAStools:

```
$ lasmerge -i [ground.las][unfiltered.las] -merged -o [merged.las]
```

With LASheight, as part of LAStools, a TIN will be generated representing the underlaying ground and for all above-ground point records the normalized height will be determined with respect to the TIN representing the underlaying ground. This normalized height will be added to the meta data, the original Z value of the point record will not be changed:

```
$ lasheight -i [merged.las] -o [normalized.las]
```

After preprocessing the point cloud data individual methodologies for the classification of buildings and vegetation will be introduced in respectively subsection 4.2.3 and subsection 4.2.4.

4.2.3 Buildings

Classification of point records as building takes place using the LASclassify algorithm, as part of LAStools:

```
$ lasclassify -i [normalized.las] -planar [standard deviation]
  -ground_offset [height] -o [classified.las]
```

As introduced in subsection 3.3.4, this classification algorithm generates contours in order to define above-ground objects. The user can define which point are part of each planar contour with the *planar* flag; this parameter describes the standard deviation point records can have from the planar region they share.

Table 2 shows the statistical results for multiple planar settings. Figure 32 shows the correlation between the number of points classified as buildings and different values of the *-planar* flag.

	Building points	Relative increase (%)
Standard deviation = 0.05	21 481	-
Standard deviation = 0.1	28 630	33.0
Standard deviation = 0.2	29 662	3.5
Standard deviation = 1.0	30 353	2.3

Table 2: Points classified as building for a random sample of 967 987 filtered points and 2 066 287 unfiltered points from the AHN2 data set.

Figure 33 shows that the correlation between the number of building points and the applied standard deviation is comparable with a infinite function $f(n) = \frac{1}{-n}$. After calculation of the limit $L = \lim_{x \to \infty} f(n)$ for the infinite function $f(n)$, L is determined as $L = 30\,353$ by using a logarithmic approximation.

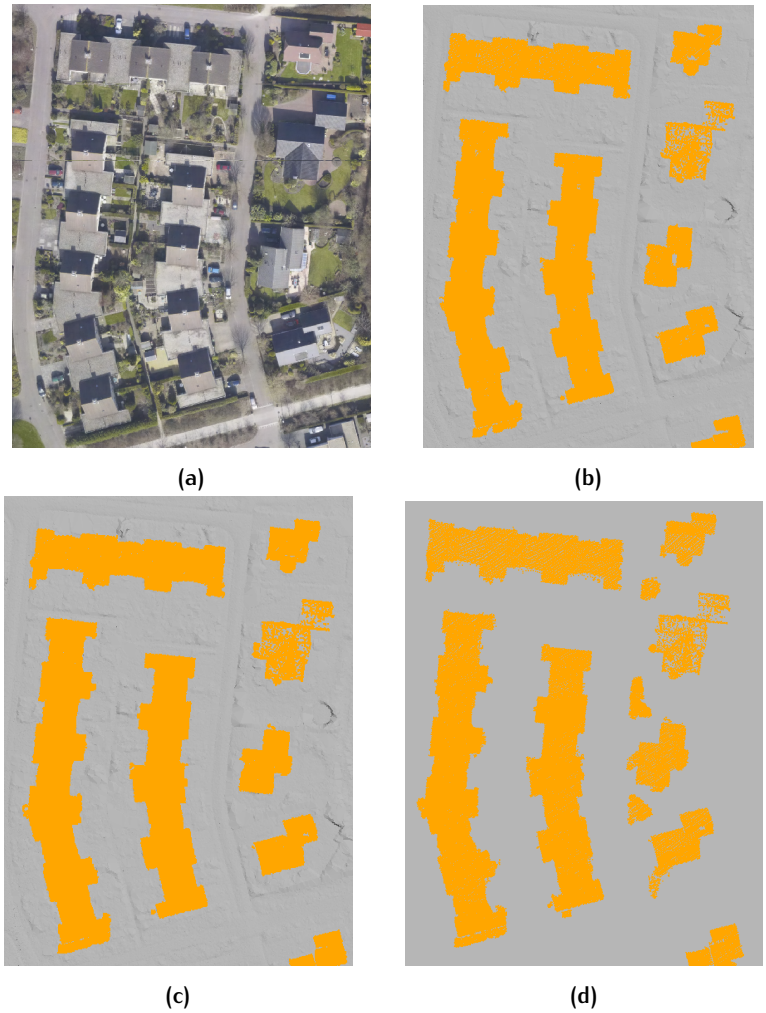


Figure 32: Points classified as building points (yellow) by using the LASclassify tool applying different parameters. (a) Aerial photograph. (b) Standard deviation = 0.1. (c) Standard deviation = 0.2. (d) Standard deviation = 1.0.

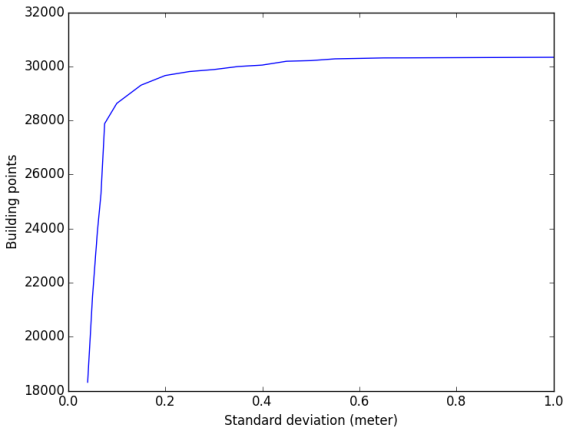


Figure 33: Correlation between the standard deviation and the amount of LiDAR points being classified as building for a sample data set.

After calculation of the accessory *planar* value it appears that vegetation is falsely classified as building (Figure 32d). Therefor, a 0.95 percentile value of L is defined as an acceptable value for the classification of building point records. A Standard deviation (SD) value of 0.1152 meter classifies this amount of building point records. This planar value is nearly the same as the standard planar value of the LASclassify algorithm, which is 0.1000 meter.

With the flag *-ground_offset* the user can define an offset with respect to normalized height from which point records are classified as building. Within LASclassify this parameter is standard set to 2.0 meter what is an appropriate height in most cases.

4.2.4 Vegetation

Similar as for the classification of point records as building, the classification of point records as vegetation takes place using the LASclassify algorithm, as part of LAStools. A description of this algorithm is given in subsection 3.3.4.

```
$ lasclassify -i [normalized.las] -rugged [standard deviation]
 -ground_offset [height] -o [classified.las]
```

Introduced in subsection 4.2.3, main concept of the classification algorithm is the generation of contours in order to define objects. Contours are defined with the *-planar* flag describing the standard deviation points can have from the planar region they share.

Table 2 shows the statistical results for multiple planar settings. Figure 32 shows the correlation between the number of points classified as vegetation and different values of the *-planar* flag.

	Vegetation points	Relative increase (%)
Standard deviation = 0.3	1 208 072	-
Standard deviation = 0.4	1 208 071	$-1.8 * 10^{-6}$
Standard deviation = 0.5	1 208 068	$-2.5 * 10^{-6}$
Standard deviation = 1.0	1 207 242	$-1.7 * 10^{-3}$

Table 3: Points classified as vegetation for a random sample of 967 987 filtered points and 2 066 287 unfiltered points from the AHN2 data set.

Figure 34 shows that there are no big visual differences observable between the different classified point cloud data products. Table 3 confirms this; the amount of points is more or less similar using different SD values. Plotting the correlation between the number of point records classified as vegetation for different planar values shows that the amount of points is quite similar up to a standard deviation of 0.8 meter (Figure 35). For larger values of the standard deviation, the amount of points classified as vegetation drops relatively more. Chosen is to adopt the standard value of 0.4 meter for the planar parameter for the classification of vegetation.

With the flag *-ground_offset* the user can define an offset with respect to normalized height from which points are started to be classified as vegetation. Within LASclassify this parameter is standard set to 2.0 meter what is an appropriate height in most cases.

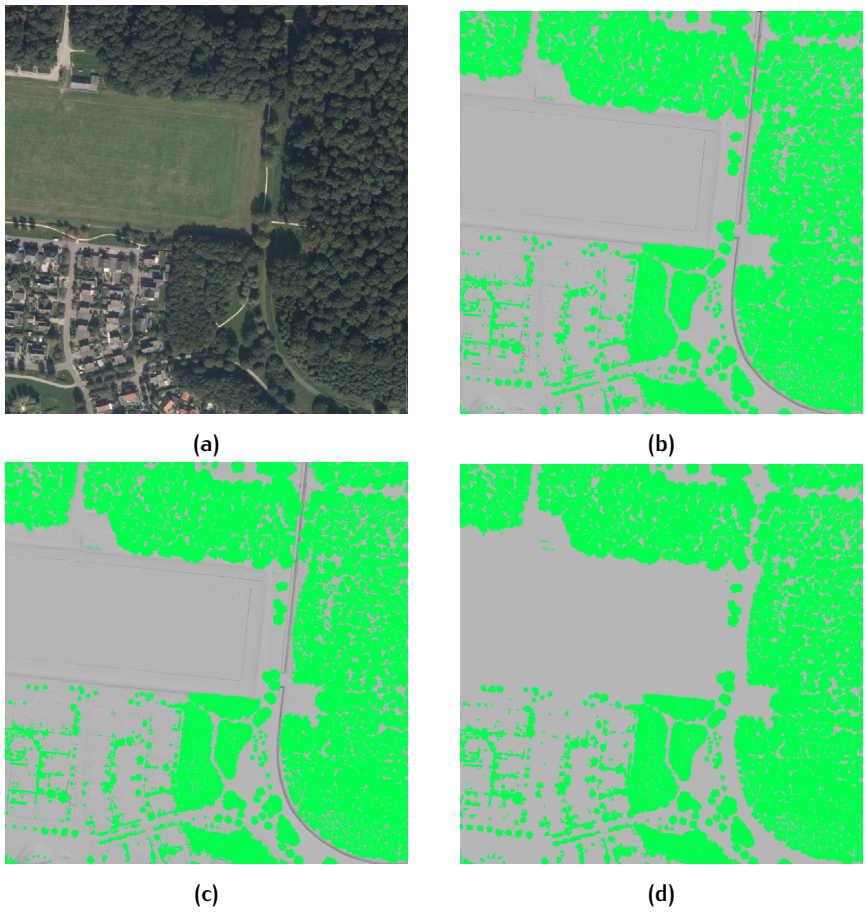


Figure 34: Points classified as vegetation points (in green) by using the LASclassify tool applying different parameters. (a) Aerial photograph. (b) Standard deviation = 0.4. (c) Standard deviation = 0.5. (d) Standard deviation = 1.0.

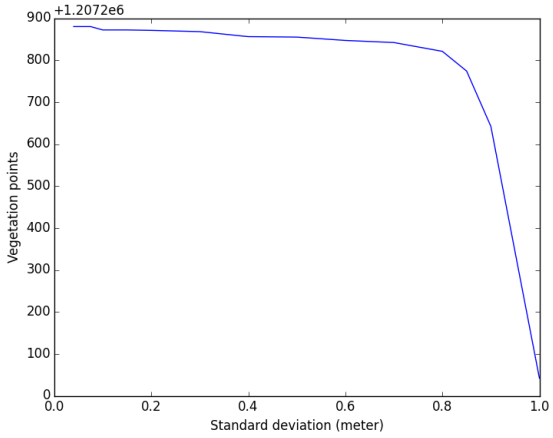


Figure 35: Correlation between the standard deviation and the amount of LiDAR points being classified as vegetation for a sample data set.

4.3 SPATIAL INTERPOLATION

In section 3.4 related work with respect to spatial interpolation was introduced. In this section methods for spatial interpolation will be introduced for the individual classes defined within the filtering and classification procedure in the previous section.

In subsection 4.3.2 a combination of interpolation methods based on a TIN, sink filling and IDW will be applied for the generation of a raster-based height map representing a DEM.

In subsection 4.3.5 an interpolation method based on interpolation of a TIN will be introduced for the generation of a DBM, this is a model that contains normalized height data with respect to buildings.

In subsection 4.3.6 an interpolation method based on interpolation of a TIN will be introduced for the generation of a CHM, this is a model that contains normalized height data with respect to vegetation.

4.3.1 Raster resolution

In subsection 3.4.1 related work has been introduced regarding the selection of an appropriate raster resolution; theoretically a spatial resolution between 0.23 and 0.41 meter is possible, based on the point density and the average distance between two points within the AHN2 data set. This is higher then the spatial resolution of all current raster-based height maps which have a spatial resolution of 0.5 meter (see section 2.4).

Within the remainder of this section, interpolation of point cloud data to raster data will take place with an output resolution of 0.25 meter.

4.3.2 Ground

First step for the spatial interpolation of ground data is the generation of a TIN as described in subsection 3.4.2. The blast2dem tool can generate raster-based height data by rasterizing a TIN derived from point cloud data via a command prompt line:

```
$ blast2dem -i [normalized.las] -step [output resolution] -kill [cut-off
  threshold] \\ -o [dem.tif]
```

Where flag *-i* defines the normalized point cloud data, the flag *-step* defines the output resolution and the flag *-o* the name and location defines where the output raster will be stored. The flag *-kill* defines an optional threshold value based on the longest edge for the triangle; when the longest edge transcends the value defined for this parameter the triangle will not be rasterized. Selection of a proper cut-off threshold is essential when rasterizing the TIN in order to prevent the occurrence of artifacts. The not filling of certain raster cells will lead to new holes in the raster-based height map. In subsection 4.3.3 and subsection 4.3.4 additional procedures will be introduced in order to estimate a height for certain holes.

Cut-off threshold selection

In subsection 3.4.4 it has been introduced that is inappropriate to generate a high resolution DEM with sparsely distributed LiDAR data: any surface generated in such a way is more likely to represent the shape of the specific interpolator used than that of the target terrain because interpolation artifacts will abound [Liu and Zhang, 2008].

Rasterization of a vector-based TIN is derived by a bivariate function for each triangle in order to estimate height values at unsampled locations. Linear interpolation fits planar faces to each triangle individual leading to a jagged appearance where it is visually possible to distinguish individual triangles (Figure 36f). This is caused by discontinuous slopes at the triangle edges and sample data points.

A cut-off threshold could be introduced that will not rasterize triangles if the length of the longest edge for that triangle transcends a predefined threshold value. Selection of a proper cut-off threshold value is important in order to prevent the occurrence of artifacts when applying interpolation based on a TIN.

Figure 36 shows samples of raster data taking into account different cut-off threshold values. A too large cut-off threshold leads to interpolation near building corners (see Figure 36d), where the selection of a too small cut-off threshold leads to the presence of many small holes within the rasterized data (see Figure 36a). It cannot be expected that other interpolation methods will provide a better height approximation of these smaller holes.

Chosen is to apply a cut-off threshold value of 2 meter (Figure 36c). This cut-off threshold value leads to the disappearance of most artifacts that would be present without the implementation of a cut-off threshold value (Figure 36e). The holes that appear when selecting a smaller threshold value are holes do not have clear artifacts in comparison with a scenario using no threshold value.

Filling remaining holes

The implementation of a proper cut-off threshold solves the artifacts that appear by interpolation based on a TIN. After introduction and implementation of a cut-off threshold value, rather than the occurrence of artifacts, the interpolated raster-based height maps contains holes (no-data values). In general, three kinds of sources for the occurrence of holes can be defined within a point cloud data set of ground points:

- Water bodies
- Building footprints
- Local deviations

Figure 37 shows the correlation between sparser LiDAR point densities located near building footprints and water bodies becomes clearly visible. Van der Zon [2011] describes multiple reasons for local deviations within the AHN2 data set:

- Missing data next to high buildings and under trees
- Lower point densities due to the reflection properties of LiDAR beams with respect to on black surfaces like asphalt and roof tiles

These errors are not specific for the AHN2 data set but can be expected for all point cloud data sets obtained by *topographic* LiDAR. Within the remainder of this section methods will be proposed for the filling of holes for water bodies using sink filling (subsection 4.3.3) and for building footprints and local deviations using IDW interpolation (subsection 4.3.4).

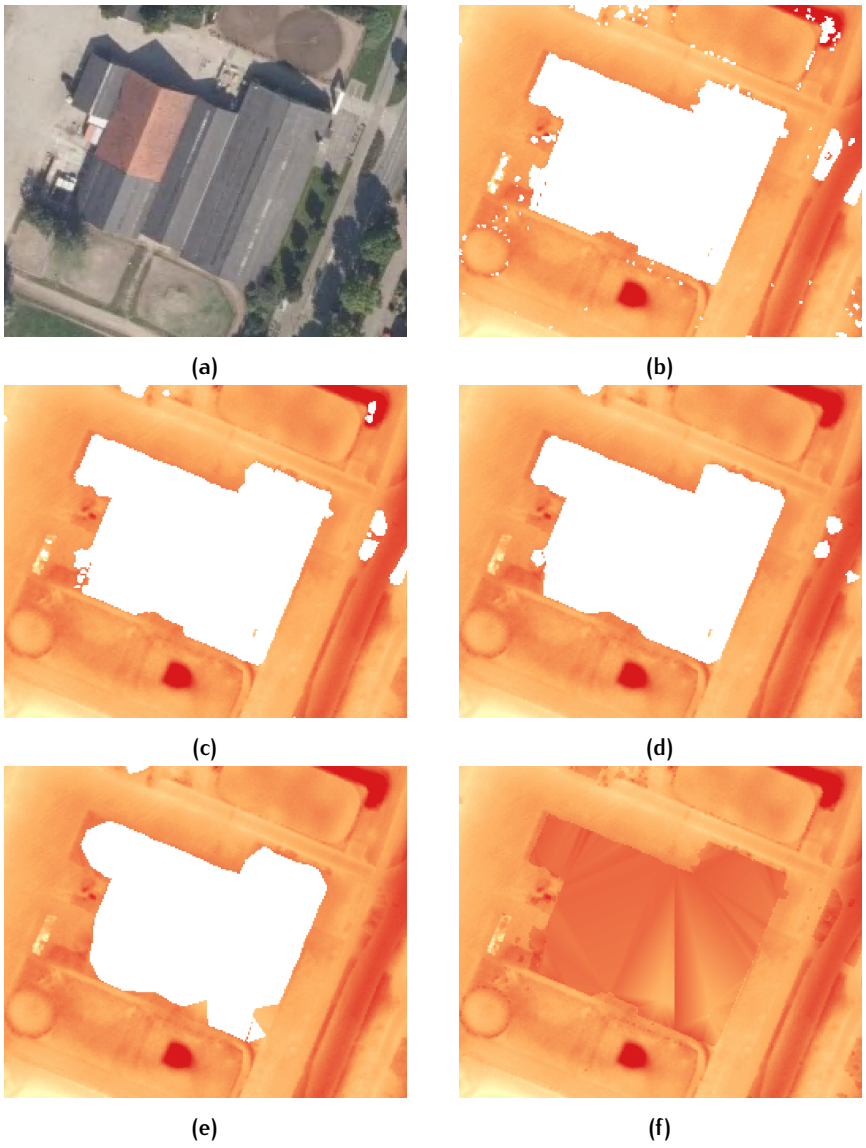


Figure 36: Influence of different cut-off threshold values after rasterization a triangular irregular network constructed from point records classified as ground. (a) Aerial photograph (b) Cut-off threshold = 0.5 meter. (c) Cut-off threshold = 1 meter. (d) Cut-off threshold = 2 meter. (e) Cut-off threshold = 4 meter. (f) No cut-off threshold.



(a) (b)

Figure 37: Correlation between sparse point record distributions and the presence of building footprints and water bodies. (a) Aerial photograph. (b) Point records classified as ground.

4.3.3 Sink filling

Due to the characteristics of *topographic* LiDAR it is not possible to extracting water bodies directly (see section 2.2). A method is applied where potential water bodies are derived based on determination of slope within the partly interpolated DEM generated in subsection 4.3.2. This method does not detect water itself, it detects areas with a high probability of the presence of water.

Slope polygon generation

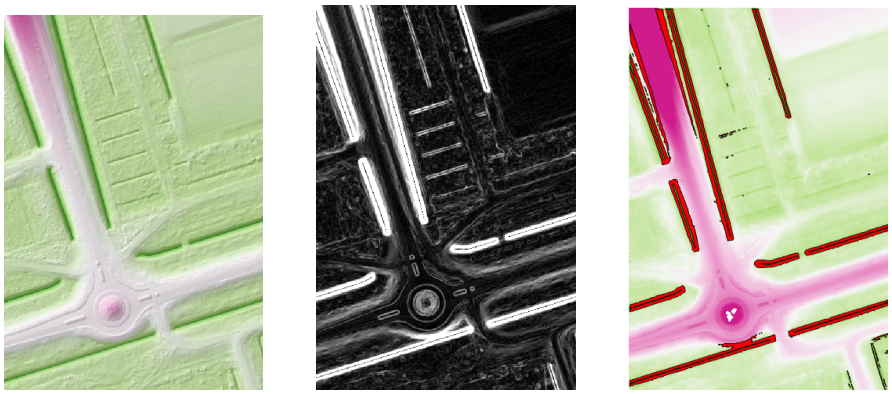
First step is to generate a slope map based the input DEM generated in subsection 4.3.2; this is done with the the GDAL command gdaldem:

```
$ gdaldem slope [dem.tif] [slope.tif]
```

Slope is an analysis method to measure the *local* steepness of a terrain by comparing the elevation for each pixel with respect to adjacent pixels. Within this methodology the slope is calculated for each raster cell with a kernel size of 3 x 3 pixels. The output is a raster file where each pixel value represents the degree of steepness with its neighboring cells. Figure 38b shows that slope is a promising parameter that correlates clearly with the presence of a water body. In order to detect potential areas for the presence of water three criteria need to be fulfilled:

- An area does a minimum degree of slope
- An area does have a certain size
- An area does have a certain ratio between size and perimeter

For the Netherlands, which can be considered as flat in general, a slope of 15 degrees can be considered as minimal threshold for the detection of water bodies. Areas that satisfy all these criteria will be polygonized into individual *slope polygons*.



(a) (b) (c)

Figure 38: Calculating a slope-based raster file from a digital elevation model. (a) A digital elevation model. (b) A slope-based raster file extracted from the digital elevation model. White colors represents raster cells with high slope, black raster cells represent areas with lower degrees of slope. (c) Slope polygons (red).

Boundary representation

The slope polygons that are created in the previous step will be used to detect the location and value of the raster cell containing the lowest height value within each individual slope polygon. A Boundary Representation (B-REP) will be generated; this is a geometric object that is represented by the union of a *topological* model, which describes the topology of the modeled object and an *embedding* model, which describes the embedding of the object in 2D Euclidean space. Application of a B-REP based on the boundaries of a polygon can be approximated within a raster file by 'masking'; this is a process that transforms polygon coordinates to pixel locations. Given a masked raster it is possible to determine the location and value of the raster cells of interest. It can happen that multiple raster cells have the same value; if so, then all raster cell locations with such a value will be stored.

When masking raster data, the masked data will be stored in a it local coordinate system. Where the locations of the raster cells of interest are in real world coordinates it is needed to store extra parameters for each mask:

- Scale
- Grid dimension
- Raster resolution
- Translation values

The scale defines the size of the grid which is determined by the largest length of the model in either the XY direction, the grid dimension defines the number of raster cells needed to fill the length of the model scale and the raster resolution defines the dimensions of a raster cell. The translation values describe the distance of the upper left corner of the mask its bounding box with respect to the upper left corner of the raster file.

After determination of the location and value of raster cells with the lowest value within a mask, a reverse conversion is needed in order to define their real world coordinates. Equation 5 describes how to obtain these real world coordinates:

$$x = scale_x * \frac{i + 0.5}{dimension_x} + translate_x \tag{5a}$$

$$y = scale_y * \frac{j + 0.5}{dimension_y} + translate_y \tag{5b}$$

Where *translate* describe the distance the distance of the upper left corner of the mask its bounding box with respect to the upper left corner of the raster file. *Dimension* defines the resolution of the grid. Scale dimensions can be determined by:

$$scale_x = dimension_x * Raster\ cell\ size \tag{6a}$$

$$scale_y = dimension_y * Raster\ cell\ size \tag{6b}$$

A list of real world raster cell locations and values within a mask is obtained now. Based on this information, the $\Delta height_{max}$ within a mask can be determined. This information is the starting point for detecting areas with a high probability of the presence of water.

Breadth-first search

After obtainment of the initial locations for the detection of areas with a high probability for the presence of water it is needed to search for other raster cells that do have similar characteristics, in order to increase the scale of information from cell level to area level.

A method in order to achieve this is to make the data relational within a tree or graph data structure. These graph data structure consist of a finite and possibly mutable set of nodes. A method to store raster data in a graph structure is by translating the data into a 2D array; a systematic arrangement of nodes in rows and columns. The array stores basically information with respect to the value for each node. In case it is needed to store additional information besides a (height) value each node needs to be rewritten so that it is capable to store extra information. algorithm 4.1 shows a methodology to rewrite an array in order to store additional data for each node.

Pu and Zlatanova [2005] introduce an overview of different search algorithms. One search algorithm is Breadth-first search (BFS) which was initially invented as a method to find the shortest path out of a maze [Skiena, 1998] and later expanded as a wire routing algorithm [Lee, 1961]. The BFS algorithm starts at the tree root or some arbitrary node of a graph and explores the neighbor nodes first, before moving to the next level neighbors, and then their successors, and so on till it finds the goal node. Rather than for example a depth-first algorithm a BFS explores more close by nodes first before exploring farther located nodes.

Algorithm 4.1: Rewriting array algorithm

Input: An array A_m representing height values from raster file for each node within the array

Output: An array A_n representing height, positional values for each node within the array and also information whether the node is visited by a search algorithm

```
1 create array A_n of 0s with equal shape as A_m
2 for i, row in enumerate(A_m) do
3     for j, z in enumerate(row) do
4         A_n.value = A_m[i][j];
5         A_n.position = A_m(i,j);
6         A_n.visited = False
```

Within this context there is no goal node, the BFS algorithm will search for adjacent nodes that have certain characteristics that they can be classified as comparable. When there are no more adjacent nodes that needs to be tested the algorithm is finished; algorithm 4.2 shows the pseudo code for a BFS algorithm.

Algorithm 4.2: Breadth-first search

Input: An array A_n described in algorithm 4.1, a starting Raster cell c and a height difference Δh

Output: A list of raster cell coordinates reachable from c labeled as water

```
1  create empty queue Q;
2  create empty list L
3  initial height H = c.value
4  Q.append(c)
5  while Queue is not empty do
6      u = Q.pop(0);
7      L.append(u.coordinate)
8      for each node n that is adjacent to u do
9          if n < H + 0.5 * Δh then
10             n.visited = True
11         if n.visited = False then
12             n.visited = True;
13             Q.append(n)
```

Another criterium that algorithm 4.2 will test is whether the height of each adjacent node does not transcend the $\Delta height_{max}$ that is determined within the mask. If so, it is assumed that this node does have a higher probability being land rather then water. By lowering $\Delta height_{max}$ of percentage (e.g. 50%) smaller parts of water courses surrounding the water body will be classified as areas with a high probability of the presence of water. Application of this parameter assumes that there is a correlation between the presence of water and height within the masked area.

Further testing

Where the input DEM contains many holes, it is possible to use this characteristic for the classification of two classes of water bodies:

- Smaller water bodies that mainly consist out of data raster cells
- Larger water bodies that mainly consist out of no-data raster cells

After determining areas with a high probability of the presence of water by applying a BFS algorithm additional testing is needed; the BFS algorithm detects many areas falsely. For further testing, some statistics will be calculated for each output of the BFS algorithm:

- Coverage (with respect to the raster file);
- Percentage of raster nodes with a no-data value within the output of algorithm 4.2
- Standard deviation of the output of algorithm 4.2

These statistics will provide the tools to test each output of algorithm 4.2 deeper. All output is tested if they meet the following conditions:

- The lowest value that is detected by the BFS algorithm should not be significant lower than the initial detected *lowest* height
- If the output does have a certain coverage (e.g. $> 10\%$), the standard deviation is supposed to have a certain value (e.g. < 0.1 meter)

Small water bodies

Small water bodies are classified as water bodies where a maximum of 5% of the output of algorithm 4.2 are raster nodes with a no-data value. If so, the output is tested if it meets the following condition:

- The areas with a high probability of the presence of water should not have a too high standard deviation (e.g. > 0.5 meter) when the *coverage* of a water body exceeds a certain value (e.g. $> 5\%$)

If the output meets this condition all raster nodes will be qualified as nodes with a high probability of the presence of water.

Large water bodies

Large water bodies are classified as water bodies where at least 5% of the output of algorithm 4.2 are raster nodes with a no-data value. For these data tested is if it meets the following conditions:

- The areas with a high probability of the presence of water should have a certain minimum size
- The lowest point that is detected by the BFS algorithm should not be significant lower than the initial detected 'lowest' height
- In case of data with a high ($> 50\%$) percentage of raster nodes with a no-data value the standard deviation is should not be too high.

If the output meets all of these conditions only the raster nodes with a no-data value will be qualified as nodes with a high probability of the presence of water.

4.3.4 Filling building footprints and local deviations

After the filling of holes identified as water bodies, holes remain within the rasterized ground data that are caused by building footprints and local deviations. These holes will be filled by using IDW interpolation. *gdal_fillnodata.py* is a Python script that fills holes within raster files by estimating heights using IDW interpolation using the heights of surrounding raster cells with a known height value:

```
$ gdal_fillnodata.py [input.tif] [output.tif] -md [value] -si [value]
```

Holes are identified as raster cells with a no-data value within the input raster using an image mask. Valid raster cells containg height data that can be used for the interpolation are searched in four directions (up, down, left, right). With the flag *-md* it is possible to define a maximum search distance.

With the flag *-si* an average filter can be applied on the interpolated pixels to smoothen potential artifacts in the raster output. The average filter has a kernel size of 3 x 3 pixels and is applied iteratively. However, due to a problem in the current code this option should be avoided until a fix has been provided [McInerney and Kempeneers, 2015].

4.3.5 Buildings

For the generation of a DBM interpolation should be limited to the interior of the building outline. This makes the concept of edge-constrained interpolation relevant. A method for designing these 'constraints' is by determinion of a buildings' outer boundary. In this subsection two methods for the extraction of building boundaries will be compared:

- Extracting building boundaries by LiDAR data
- Extracting building boundaries by external 2D geodata sets

The first method extracts building boundaries directly from classified point cloud data where the second method extracts building boundaries indirectly using external 2D geodata sets.

Extracting building boundaries by LiDAR data

For the determination of building boundaries from point cloud data only those point records are relevant that are classified as building (see subsection 4.2.3). A tool to extract building boundaries is LASboundary, as part of LAStools. LASboundary reads LiDAR point records and computes a boundary polygon for the points:

```
$ lasboundary -i [classified.las] -keep\_class [class_num] -holes
  -disjoint -concavity [value] -o [buildings.shp]
```

The flag *-i* defines the input point cloud data, the flag *-keep_class* defines the class for which the boundary should be determined (buildings = class 6). The flag *-holes* defines whether a polygon can have interior holes. Adding the flag *-disjoint* will produce multiple hulls in order to generate disjoint polygons for individual buildings. Definition of the flag *-concavity* defines a maximum search distance for each separate hull; for example, a *concavity* value of 1 meter, meaning that voids with a distances of more than 1 meter are considered as the exterior.

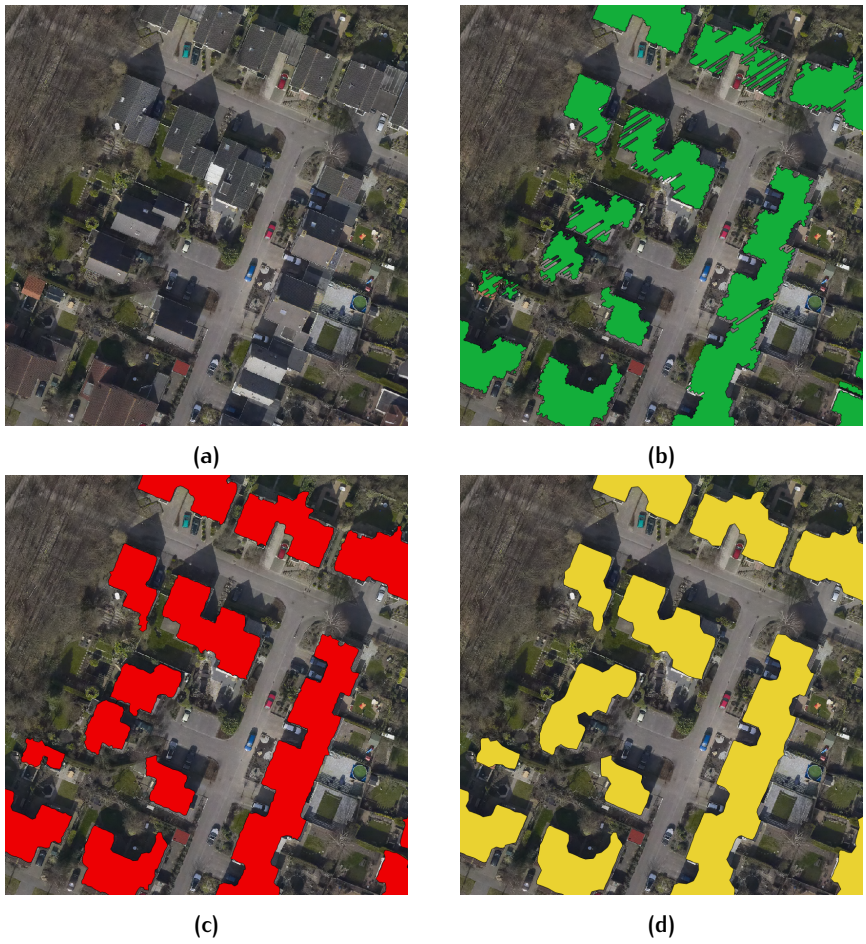


Figure 39: Visualization of *disjoint* polygons after the application of multiple *concavity* values using the LASboundary tool. (a) Aerial photograph. (b) Concavity value = 0.5 meter. (c) Concavity value = 1.0 meter. (d) Concavity value = 2.0 meter.

Figure 39 shows the output of LASboundary using different *concavity* values. Selection of a relative small *concavity* value will lead to gaps and spikes within building polygons due to a too low point density (Figure 39b). Selection of a too large *concavity* values will lead to a large generalization of polygons or even generate polygons covering multiple buildings, if located close to each other (Figure 39d). Chosen is to adapt a *concavity* value of 1.0 meter, this value will prevent most of the above introduced problems (Figure 39c).

Besides the detection of a building its outer boundary it can also happen that a building contains a non-build area within its interior (Figure 40a). In such a situation there should also be a hole within the interior of the building polygon. With LASboundary it is possible to add these holes to a building polygon as defined in the previous step (Figure 40c). Hole definition applies the same *concavity* value, no separate value can be defined.



(a) (b) (c)

Figure 40: Adding holes within the interior of a building polygon with the LAS-boundary tool. (a) Aerial photograph. (b) A polygon representing the outer boundary of a building. (c) A polygon representing the outer boundary, including an interior hole within a building polygon.

Extracting building boundaries by external geodata sets

For the Netherlands, there exist two open data sets containing building information: *TOP10nl* and BAG. Details about both data sets and a qualitative comparison are described in Appendix B. Chosen is to make use of the BAG data set because it does have a higher positional accuracy. A method how to obtain a recent version of the BAG data set and how to process and store the data set in a spatial database is described at `http://www.nlextract.nl/home`.

The BAG and the AHN2 data set are obtained at different moments; there is a temporal difference between the dates of collection of the data sets. In order to have information about buildings that were only present at the moment of data acquisition of both data sets it is needed to know the temporal accuracy for both data sets.

The BAG data set contains information about the year of construction ('bouwjaar') resulting in a temporal accuracy of 1 year.

For the AHN2 data set it is possible to check the public header block information of the point cloud data in order to obtain the file creation day and year (see section 2.2). After checking this file creation day for four data samples it appears that three of them are created within two days. It can be assumed that the time stamp is probably not related to the data collection date, but to the date that data was processed. Date of collection for the AHN2 data set is differs over multiple areas (Figure 41) and since only the year of collection is known theoretically the temporal accuracy is between 1 day and 1 year. Practically the AHN2 data set provide information with a temporal accuracy of 1 year, assuming no errors within the data set with respect to the year of collection.

After considering the temporal accuracy of both data sets, it is possible to extract building information from the BAG data set. All buildings with a year of construction that is older than the year of collection for the AHN2 data set for the particular area are clipped from the database using a SQL-query:

```
$ ogr2ogr -skipfailures -clipsrc [xmin] [ymin] [xmax] [ymax] [output.shp]
"PG:dbname=[localhost] user=[user] password=[pw] dbname=[dbname]"
-sql "SELECT * FROM pandactueel WHERE bouwjaar < [AHN2]"
```

The output contains polygons representing individual houses (Figure 42b). In order to interpolate the building points it is needed to have the outer boundary of each building block, for that reason a dissolve operation is applied in order to obtain polygon data representing the outer boundary of individual building blocks (Figure 42c).



Figure 41: Gathering years of the AHN2 point cloud data [Van der Zon, 2011]

Comparison

When looking at the building boundaries extracted from the classified LiDAR data most boundaries are detected well (Figure 42b). Nevertheless, detected building boundaries can appear fuzzy, contain false holes within building structures and some buildings are detected only partly or sometimes even completely not at all (Figure 42e). This in line with the conclusion of Taylor et al. [2007], who states that LiDAR data is not dense enough to model accurately sharp surface discontinuities like building boundaries.

Despite this, LiDAR data could be used as an indicator for the detection of building boundaries in case this data is wrongly present or missing within external 2D geodata sets. When taking a closer look at the BAG data set it appears that not all building boundaries are detected completely or sometimes even completely (Figure 42e).

Overlaying building boundary data extracted from LiDAR data with building boundaries extracted from the BAG data set shows that LiDAR data detects roofs of building structures such as carports and barns that are not present within the BAG data set. These objects are not included because of the definitions of the BAG data set[2] (see Appendix B). LiDAR data also detects inaccurate and missing data within the BAG data set (Figure 42f).

Doing the opposite shows that the BAG data is capable to detect building boundaries that are not detected by LiDAR data (Figure 42f). Where building boundaries obtained by LiDAR can be wobbly due to the characteristics of LiDAR, the building boundaries from the BAG data set are represented with a higher positional accuracy. Also falsely holes detected within building structures when using LiDAR data are not present in the BAG data set.

It can be concluded that both methods are complementary with respect to each other. For this reason both methods will be combined for the detection of building outlines.

2 The smallest functional and architectural-constructive, self contained unit that is directly and permanently connected to the ground which is enter-able and lockable [BAO, 2013].



Figure 42: Visual comparison of different methods of building boundary extraction. (a) Aerial photograph. (b) Building polygons extracted from the BAG data set. (c) Dissolved building polygons. (d) Building polygons extracted from LiDAR data. (e) Dissolved building polygons extracted from the BAG data set overlaid by building polygons extracted from LiDAR data. (f) Building polygons extracted from LiDAR data overlaid by dissolved building polygons.

Edge-constrained interpolation

After extraction of building boundaries in twofold both shapefiles will be dissolved. In order to deal with falsely detected holes during the extraction of building boundaries from LiDAR data, only holes within the BAG data set will be used to indicate holes within buildings (e.g. patio's).

Next step is a clipping operation on the classified point cloud data that has been defined in subsection 4.2.3. Within this process all points classified as *vegetation* are dropped. Point data that is classified as *noise* is kept; many points are falsely classified as noise where these points should have been classified as points reflected on buildings. For this reason there might be potential information within the noise points (Figure 43). Noise points will be further processed when a point is located within the interior of a building polygon from the BAG data set. This operation takes place with LASclip, as part of LAStools:

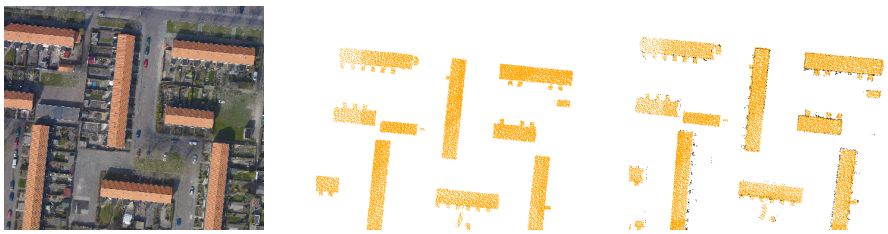
```
$ lasclip -i [classified.las] -poly [dissolved.shp] -drop_class
  [class_number] -o [clipped.las] -v
```

Where the flag *-i* indicates the input point cloud that will be clipped, the flag *-poly* indicates the shapefile that will be used to clip the point data. The flag *drop_class* indicates the point cloud classes that needs to be dropped during the clipping procedure and the flag *-o* indicates the location and name of the output file. The flag *-v* indicates that the interior of the shapefile defined with the flag *-poly* needs to be clipped.

After clipping the point cloud data it is needed to normalize the height of the point data with respect to the underlaying DEM in order to generate a DBM. For that reason LASheight will be used another time:

```
$ lasheight -i [clipped.las] -replace_z -o [normalized.las]
```

Application of LASheight differs from the method described in subsection 4.2.2, adding the flag *-replace_z* will overwrite the original Z-coordinate and replace it with the normalized height for each above-ground point record.



(a) (b) (c)

Figure 43: The potential of noise points during the generation of a digital building model. (a) Aerial photograph. (b) Clipped building point data (yellow). (c) Clipped building (yellow) and noise (black) point data located within geometry of the BAG data set.

In subsection 2.3.1 a number of sources for errors within the AHN2 point cloud data have been introduced, resulting in raster cells containing height values that are mostly lower with respect to the real height. A two-step manipulation of the point cloud data will be performed in order to remove point records that cause these errors:

- Thickening the point cloud
- Thinning the point cloud

The first step is a thickening of the point cloud data. In order to remove point records with a lower value with respect to the real height, information from neighboring points can be used in order to eliminate such lower point records. For all points within a building polygon, each point will be duplicated 8 times in a discrete circle with a small radius around every original point. In this way outliers are not filtered, they are covered by artificial point records that originally reflected on the nearby surface of the building.

The second step is a thinning of the thickened point cloud. A virtual grid is defined and for each cell the highest point record will be selected. This record will replace all points within the raster cell by one located in the center of the virtual grid cell (Figure 44b). Both these steps can be applied with LASthin, as part of LAStools:

```
$ lasthin -i [normalized.las] -step [step size] -highest -subcircle
  [radius] -o [manipulated.las]
```

Where the flag *-i* defines the input point cloud, the flag *-highest* indicates that the highest point record should remain for each virtual grid cell with a cell size that is defined with the flag *step*. The flag *-subcircle* defines the radius of the artificial duplicates for each point record and the flag *-o* determines the name and location of the cleaned output point cloud file.

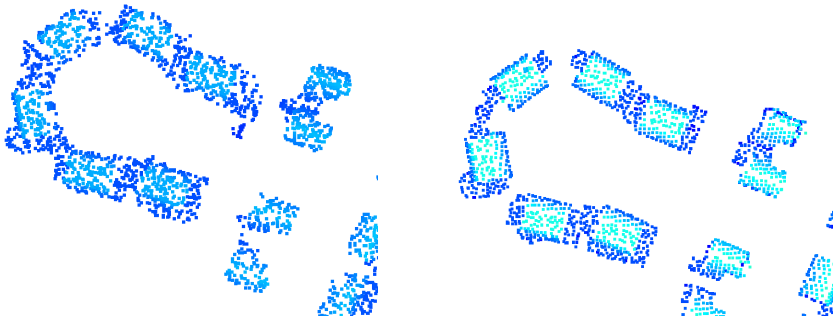


Figure 44: Point cloud manipulation. (a) Original point cloud data. (b) Point cloud data after, thickening and thinning by selecting the highest LiDAR for each raster cell.

The point cloud data after manipulation will provide better input data for spatial interpolation. In section 3.4 it is introduced that interpolation based on a TIN has the potential to generate height data with respect to buildings in a proper way because there will not take place any smoothed by definition. For this reason interpolation based on a TIN has the capability to represent complex building segments in a correct and non-smoothed way. By using the shapefile data representing building boundaries it possible to generate height data for only those raster cells that are located within the interior of a building polygon. In this way a strategy is applied that is comparable with the construction of a *constrained* DT.

Application of interpolation based on a TIN using point records that are classified as building and noise (Figure 45b) accomplish a higher coverage with respect to the usage of only those point records that are classified as building point records (Figure 45a).



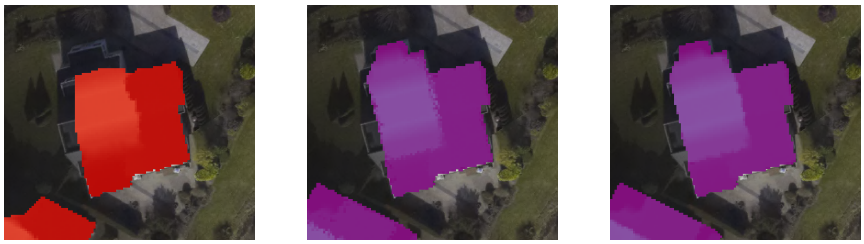
(a) (b)

Figure 45: Application of interpolation based on a triangular irregular network. (a) Using only point records classified as building points. (b) Using both point records classified as building and noise points located within building geometry of the BAG data set.

By increasing the scale of the DBM some more interesting comparisons can be made. The applied classification algorithm described in subsection 4.2.3 lacks to determine buildings 100% correctly. This results in an increasing degree of misclassified point records when the complexity of a building increases. For this reason, point records reflected on small building components such as bay windows, dormers and small veranda's are often not classified as building. Figure 46a shows that the interpolation of only building point records, without the application of point cloud manipulation results in a poor representation of height data within a DBM.

When expanding the input data with point records classified as noise located within building geometry of the BAG data set it appears that a higher coverage of raster data for buildings is achieved. Despite the higher coverage, the height data does have a fuzzy appearance, especially along building boundaries (Figure 46b). This height data is similar to the data related to buildings within the currently existing raster-based height maps introduced in subsection 2.4.1. The presence of fuzzy height data is related to errors during the collection of the AHN2 data set.

The addition of point cloud manipulation steps as introduced on the previous page leads to raster data having a smoother appearance and less artifacts (Figure 46c). Fuzzy height data within the previous examples caused by unwanted point records are mostly covered by artificial point records (thickening of the point data) and a removal of the original erroneous point records (thinning of the point data).



(a) (b) (c)

Figure 46: Comparison of raster data of different digital building models. (a) Using classified building point cloud data. (b) Using point records classified building and noise points located within geometry of the BAG data set. (c) Using classified building and noise point cloud data located within geometry of the BAG data set combined with point cloud manipulation.

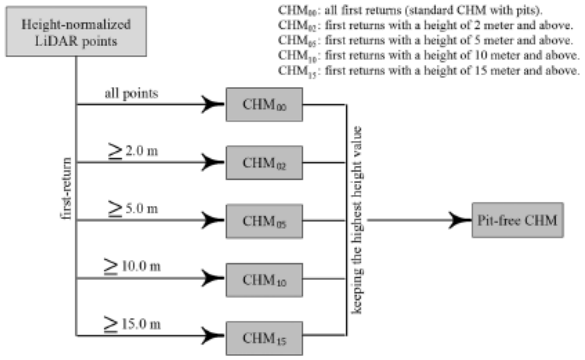


Figure 47: Diagram of a pit-free algorithm methodology [Khosravipour et al., 2014].

4.3.6 Vegetation

The first step in order to generate a DBM is to normalize the height of the point data with respect to the underlaying DEM in order to generate a DBM, within this procedure all points being classified different then vegetation are dropped. For that reason LASheight will be used again:

```
$ lasheight -i [clipped.las] -replace_z -keep_class [class_number]
-o [normalized.las]
```

Application of LASheight differs from the method described in subsection 4.2.2, by adding the flag *-replace_z* will overwrite the original Z-coordinate for each above-ground point record with the normalized height with respect to an underlaying DEM. Addition of the flag *-keep_class* keeps only point records classified as vegetation.

In subsection 3.4.3 it is introduced that the presence of data pits is a big challenge within the generation of CHMs. In order to address this issue, Khosravipour et al. [2014] introduce the concept of partial CHMs; in an iterative process multiple CHMs are generated excluding all returns above an increasing height above an underlaying DEM (Figure 47). Each partial CHM represents only some higher parts of the vegetation. ASPRS [2013] distinguish three classes of vegetation:

- Low vegetation (0.5 m <height $\leq$ 2.0 m)
- Medium vegetation (2.0 m <height $\leq$ 5.0 m)
- High vegetation (5.0 m <height)

A similar layering system can be used for the construction of partial CHMs after rasterization of a TIN constructed from point records classified as vegetation. In order to interpolate vegetation point records reflected on the same tree crown, a cut-off threshold value should be larger then the average point spacing, but smaller than the space that separates individual trees. A cut-off threshold value of 1.5 meter is applied based on the LiDAR point density:

```
$ blast2dem -i [normalized.las] -step [step_size] -kill -o
  [interpolated.tif]
```

Where flag *-i* defines the normalized point cloud data, the flag *-step* defines the output resolution and the flag *-o* the name and location where the output data will be stored. The flag *-kill* defines a threshold value based on the longest edge for each triangle; when the longest edge of the triangle transcend the kill-value the interior of the triangle will not be rasterized.

For the first partial CHM (CHM_{00}), all LiDAR point records classified as vegetation are used for construction (Figure 48b); this is a standard CHM that other researchers typically generate from first return LiDAR point records [Hyyppä et al., 2008]. As second, CHM(CHM_{02}) is constructed by including all point records classified as vegetation having a normalized heights of 2 meters or more with respect to the underlaying ground (Figure 48c). Point records with a normalized height higher then 5 meters are used in order to generate the third CHM (CHM_{05}, Figure 48d). The fourth CHM (CHM_{10}) and fifth acchm (CHM_{15}) are constructed from point records having normalized heights of respectively at least 10 (Figure 48e) and 15 meter. This process is continued iteratively with threshold intervals of 5 meter until the highest normalized vegetation points are lower then the threshold value. After generation of the partial CHMs they are merged into one CHM (Figure 48f). This CHM preserves the morphological structure of individual tree crowns better having less data pits in comparison with the first-return CHM (Figure 48b).

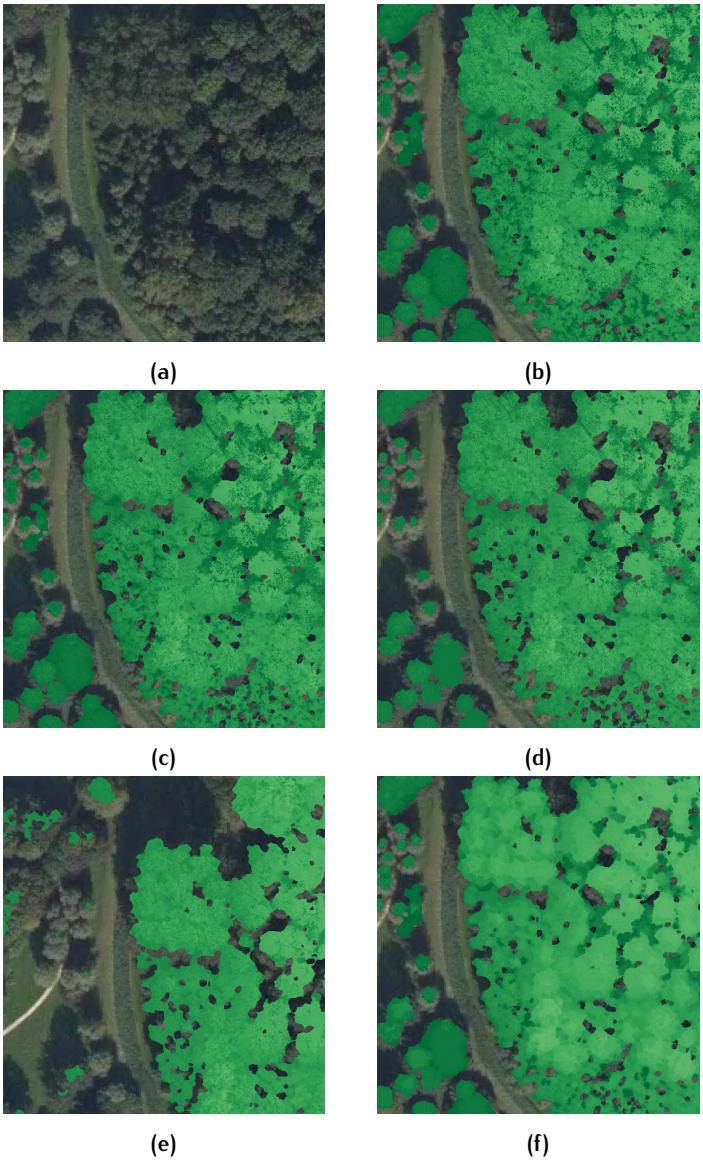


Figure 48: Canopy height model generation. (a) Aerial photograph. (b) CHM_{00}. (c) CHM_{02}. (d) CHM_{05}. (e) CHM_{10}. (f) Merged canopy height model.

4.4 POST PROCESSING

In section 3.5 related work regarding to post processing is introduced. In a raster-based context, post processing is a set of operations that can be applied on the generated raster data in order to improve its quality and useability. Processing steps that will be applied in this section are resampling in subsection 4.4.1, smoothing of the data in subsection 4.4.2 and merging of the tiles generated in section 4.1 in subsection 4.4.3.

4.4.1 Raster resampling

Final application of this thesis is a comparison of the output data generated according to the methodology described within this chapter with currently existing raster-based height maps introduced in section 2.4. For a proper comparison it is required that the raster data does have an equal raster resolution. Where it has been shown in subsection 4.3.1 that it is possible to generate data at a higher resolution, two strategies can be applied in order to achieve an equal resolution as currently existing height maps do have:

- Direct interpolation of data at the required output resolution
- Indirect interpolation of data at the highest possible resolution and resampling of the data to the required output resolution

Within this subsection both methods will be applied and compared.

Direct interpolation

When directly interpolating the LiDAR data at the required output resolution, topographic features smaller then the DEM resolution will be suppressed and smoothened during the interpolation process (Figure 36a).

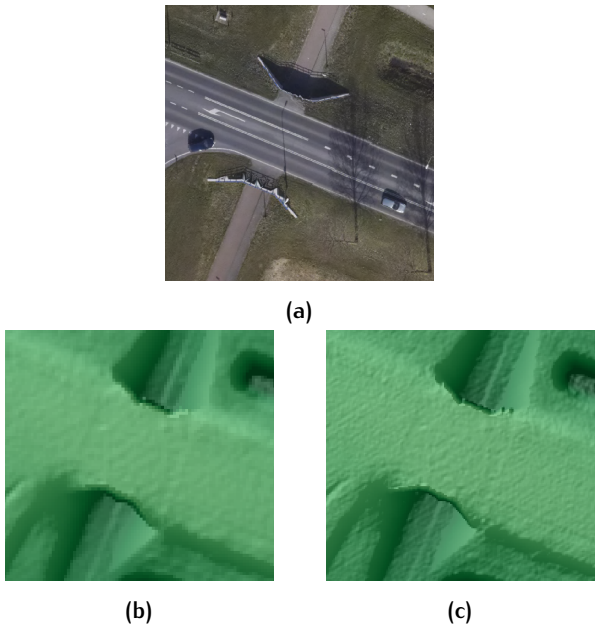


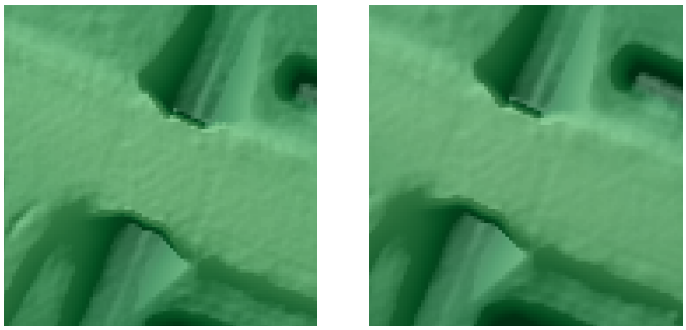
Figure 49: Digital elevation model generated by gridded interpolation based on a triangular irregular network. (a) Aerial photograph. (b) 0.5 meter resolution. (c) 0.25 meter resolution.

Indirect interpolation

In subsection 4.3.1, the second strategy is applied for the generation of a raster-based height data having a spatial resolution of 0.25 meter (Figure 49c), resampling can be one with gdalwarp, as part of GDAL:

```
$ gdalwarp -s_srs "EPSG:28992" -tr [xres} [yres] -dstnodata [val]
-r [resampling method] [input.tif] [output.tif]
```

Where the flag *-s* indicates the spatial reference system and the flag *tr* the resolution in the x- and y-axis. The flag *-dstnodata* indicates the value that needs to be assigned to new raster cells for which no height can be determined and the flag *-r* indicates the resampling method. In subsection 3.5.1 it is described that the resampling methods *bilinear* and *cubicspline* result in smoother results when resampling the raster data.



(a) (b)

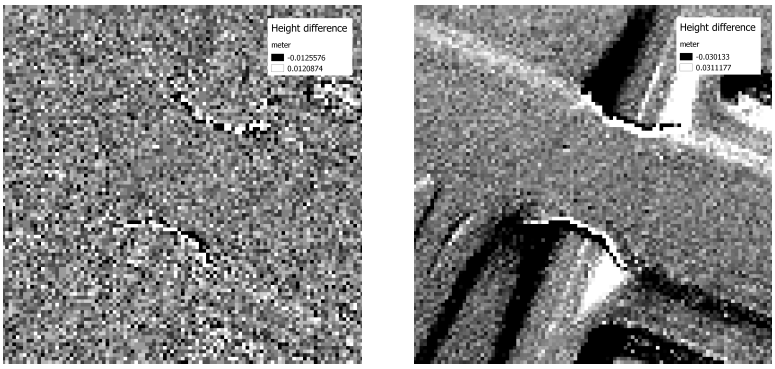
Figure 50: Indirect interpolation. (a) 0.5 meter resolution raster data after bilinear resampling. (b) 0.5 meter resolution raster data after cubicspline resampling.

Figure 50 shows the down-sampled raster data after applying a bilinear resampling (Figure 50a) and cubic spline resampling procedure (Figure 50b). Visually there does not seem to be that much difference at first instance; some degree of smoothness is perceptible near the overpass of the bikepath.

Comparison

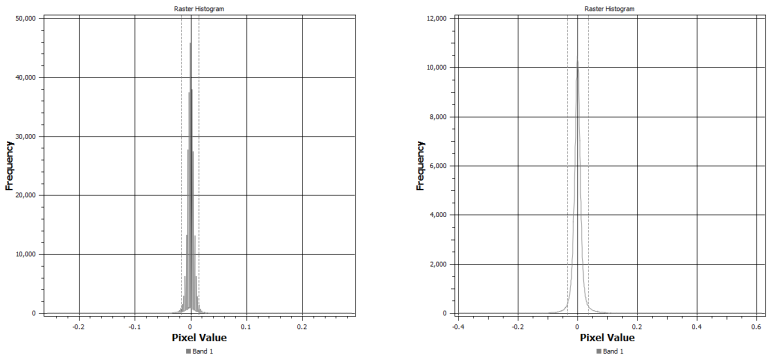
When subtracting the bilinear resampled raster data from the direct interpolated data, the difference is near-random distributed applying a bilinear resampling method (Figure 52). The differences in height is at centimeter level ($-0.012 < \Delta h_{max} < 0.012$ meter). Figure 51a indicates that the height of raster cells representing the overpass are higher value (white), where part that belong to the tunnel have a lower value after resampling (black). It can be concluded that a better representation of the situation is achieved by bilinear resampling in comparison with direct interpolation.

When subtracting the cubicspline resampled data from the direct interpolated data a correlation is visible between the height difference and both models (Figure 51b). Features can be distinguished in the subtracted data, this indicates that, rather than smoothing, this resampling method also applies a geometrical shift. Height differences are at centimeter level but larger in comparison to bilinear resampling ($-0.030 < \Delta h_{max} < 0.031$ meter) which indicates a higher degree of smoothing in comparison to bilinear resampling.



(a) (b)

Figure 51: Difference between directly interpolated raster data and indirect interpolated data after resampling. (a) Difference between directly interpolated raster data and indirect interpolated data after bilinear resampling. (b) Difference between directly interpolated raster data and indirect interpolated data after cubic spline resampling.



(a) (b)

Figure 52: Distribution of height differences between direct interpolation and indirect interpolation using different resampling methods. (a) Bilinear resampling. (b) Cubicspline resampling.

The distribution of the height difference between the direct interpolated model and both bilinear as well as cubic spline resampling is Gaussian (Figure 52)

It has been shown that some degree of improvement can be achieved by applying by interpolation of the data at the highest possible resolution and resampling of the data to the required output resolution. Especially for complex situations, where direct interpolation provides less smooth looking raster data, it has been shown in this subsection that resampling is capable to process the data in a smooth way. This is interesting, especially for the application of complex objects such as houses.

Both methods perform some degree of raster smoothing and for that reason, the selection of the resampling method will take place in subsection 4.4.2.

4.4.2 Raster smoothing

In subsection 3.5.2 it has been introduced how smoothing of raster data is beneficial, whereas smoothing also can lead to the loss of real tpographic information within a DEM or DSM.

In the previous subsection it has been proved that resampling has the potential to generate smoother data; a generalization of raster data with a higher resolution takes place in order to retrieve raster data at the required (lower) resolution. Within this method, some degree of smoothing is will take place.

Selection of a best resampling method is impossible; [Li et al., 2011] states that no single values of area and depth thresholds are best in all cases. For this reason it is chosen to apply the bilinear resampling method: a lower degree of smoothing is applied in order to generate resampled data with a relative lower degree of smoothness. By selecting bilinear resampling there will not be a geometrical shift, which has been indicated when applying cubicspline resampling. Application of bilinear resampling makes it is possible to smooth un-autocorrelated errors such as numerous small depressions. It is expected that the degree of smoothing will not lead to excessive smoothing.

4.4.3 Virtual raster generation

Final post-processing step is to merge the overlapping tiles that are created by pipelining the data as first step within the processing methodology as treated in section 4.1. In order to merge the small-scale data, a VRT will be generated from all overlapping tiles, in subsection 3.5.3 the advantages of the creation of a VRT above the creation of mosaic images are explained.

Besides the composition of large-scale data from small scale data, the following steps will be applied additional within the generation of a VRT:

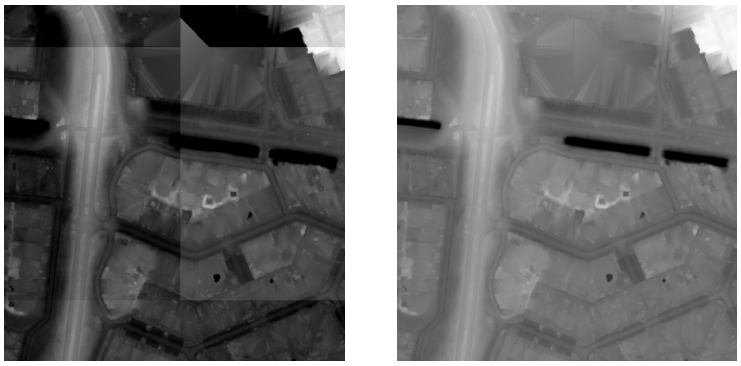
- Removal of buffers
- Geo referencing of the raster data

This can be applied with gdalbuildvrt, as part of GDAL:

```
$ gdalbuildvrt -a_srs "EPSG:28992" -te [xmin] [ymin] [xmax] [ymax]
  [output.tif] [input.tif]
```

Where the flag *-a_srs* indicates the spatial reference system and the flag *-te* the boundary of the original convex hull in order to remove the buffer.

Figure 53 shows the difference between a set of non-overlapping tiles (Figure 53a) and the same raster data represented in a overlapping VRT after removal buffered data(Figure 53b); near the tile boundaries jumps in height/color are clearly visible, both because coloring takes place for each tile separately, but also because of edge effects due to bad interpolation near tile boundaries. After the creation of a VRT, visualization of raster data will be applied in a uniform way.



(a) (b)

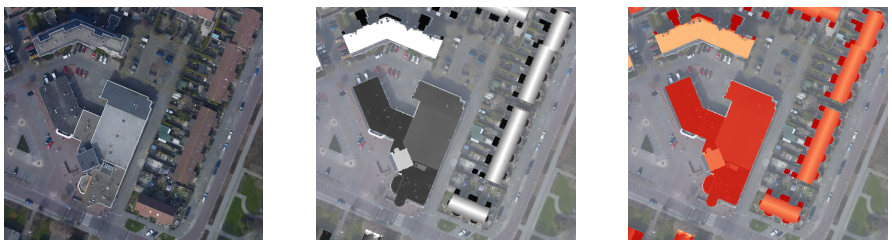
Figure 53: Virtual raster generation. (a) A virtual raster consisting of non-overlapping tiles. (b) A virtual raster consisting of overlapping tiles after the removal of buffered data.

4.5 RASTER VISUALIZATION

In the final part of the methodology visualization of the raster data will be treated. In section 3.6 related work was introduced with respect to raster visualization. In the first part of this section a methodology will be proposed in order to improve the visualization of the raster data. Within the scope of this thesis raster visualization will be applied in order to support a visual inspection of the raster data, similar as applied by Luethya and Stengeleb [2005] introduced in subsection 3.6.4. In the second part of this section a method regarding multi-scale representation of raster-data is proposed.

4.5.1 Hypsometric tinting

Hypsometric tints are colors used to indicate elevation. Standard raster data is visualization based on singleband gray values; high values are near-white and lower values are near-black (Figure 54b). Applying hypsometric tinting makes it possible to apply another color schema and specify which color need to represent certain heights. In this way a producer can control the color schema and produce maps with better accessible data (Figure 54c).



(a) (b) (c)

Figure 54: Calculating a slope-based raster file out of a digital elevation model. (a) An aerial photograph. (b) A raster image visualized with singleband gray colours. (b) A raster image visualized with hypsometric tints.

4.5.2 Hill shading

Hill shading creates an effect that provides an optical relief for cartography. With *gdaldem*, as part of GDAL, it is possible to generate hill shadings based on raster-based height data:

```
$ gdaldem hillshade [input.tif] [output.tif] -z [value] -s [value]
-az [value] -alt [value] -of GTiff
```

Where the flag *-z* indicates the vertical exaggeration used to pre-multiply the elevations and the flag *-s* the ratio of vertical units to horizontal. The flags *-az* and *-alt* indicate respectively the azimuth and altitude of the light, in degrees. The flag *-of* indicates the output format.

Figure 55b shows an hill shade image based on the height data generated within this thesis. Figure 55c shows how a combination of hypsometric tinting and a hillshade image is capable to improve the visualization of raster data.



(a) (b) (c)

Figure 55: Calculating a slope-based raster file out of a digital elevation model. (a) An aerial photograph. (b) A hill shade image generated from raster-based height data. (b) A hill shade image overlain with hypsometric tining based on raster-based height data.

4.5.3 Image overview generation

In subsection 3.6.3 related work has been introduced with respect to the visualization of large-scale raster-based data. Where the scale increases the efficiency of the raster-based data decreases. Image overviews are down-sampled versions of the original high scale raster data. The concept of image pyramids combines these image overviews with the generation of tiles.

Due to the selection of a tiling-based method for the creation of a pipeline as introduced in section 4.1 implementing image overviews is preferred with respect to the generation of an image pyramid since tiles are already available. Image overview generation will take place on the individual tiles, rather than the virtual raster described in subsection 4.4.3

With *gdaladdo*, as part of GDAL, overview images can be built. No new files need to be built, the overviews can be included in the current available tiles:

```
$ gdaladdo -r [resampling method] [input.tif] [overview_level_2^1]
  [overview_level_2^2] [overview_level_2^X]
```

Where the flag *-r* indicates the resampling method. A number of image overview levels can be added by the user in order to define the number of image overviews and their level, typical image overview levels can be 2 4 8 16. In subsection 3.6.3 it is introduced that bilinear interpolation is a good and fast method for continuous data, such as elevation.

5 IMPLEMENTATION & RESULTS

In section 5.1 the implementation of the methodology that is introduced in chapter 4 will be described. Test data sets that will be introduced in section 5.2 and in section 5.3 a validation of the output of this methodology will be provided for these test data sets. In section 5.4 a quality assessment will take place by assessing the output of the methodology within this thesis with current raster-based height maps introduced in chapter 2.

5.1 IMPLEMENTATION

Implementation of the methodology that is described in chapter 4 is applied in a number of Python scripts. Additional, there is made use of the Python packages *GDAL*, *OGR*, *OSR*, *Numpy*, *Image* and *subprocess*. All work related to point cloud data is done with *LAStools* and all work with respect to visualization is done with *QGIS*.

5.2 TEST DATA SETS

Four sample areas of 2 x 2 kilometer with different terrain characteristics are selected in order to validate and assess the quality of the developed methodology under different circumstances:

- Dronten, a rural town in the late 1950's drained province Flevoland. The landscape in this area is famous for its straight lines (Figure 56a).
- Kerkrade, a city in the province of Limburg which can be considered as a mountainous area by Dutch standards (Figure 56b).
- Den Haag, the third largest city of the Netherlands and the city having the highest average citizen density (Figure 56c).
- Leiderdorp, a semi-rural area in the western part of the Netherlands, the landscape is a typical moorland with small height differences between land and water (Figure 56d).

In Table 4 some details about the input point clouds are shown; in Appendix C a deeper analysis of the LiDAR data sets is provided. Heights are with respect to the Dutch geodetic datum, the Normaal Amsterdams Peil (NAP). Figure 57 depicts the generated DEMs for the test data sets and Figure 58 depicts the generated DSMs for the test data sets.

	Data set			
	Dronten	Kerkrade	Leiderdorp	's-Gravenhage
Point records (*filtered*)	35 854 955	41 756 071	59 706 251	33 089 491
Point records (*unfiltered*)	56 056 266	70 546 031	6 127 630	32 049 810
Min Z (*filtered*)	-5.91	98.21	-3.73	-10.25
Max Z (*filtered*)	-0.63	208.15	9.00	7.43
Min Z (*unfiltered*)	-5.82	98.18	-3.00	-10.99
Max Z (*unfiltered*)	38.85	218.67	84.15	133.87
Scale factor XYZ	0.01/0.01/0.01	0.01/0.01/0.01	0.01/0.01/0.01	0.01/0.01/0.01
File creation day/year	240/2010	142/2013	239 2010	239/2010
Number of point returns	1	1	1	1
Classification	0	0	0	0

Table 4: Details of test data sets, heights are with respect to the Dutch geodetic datum.

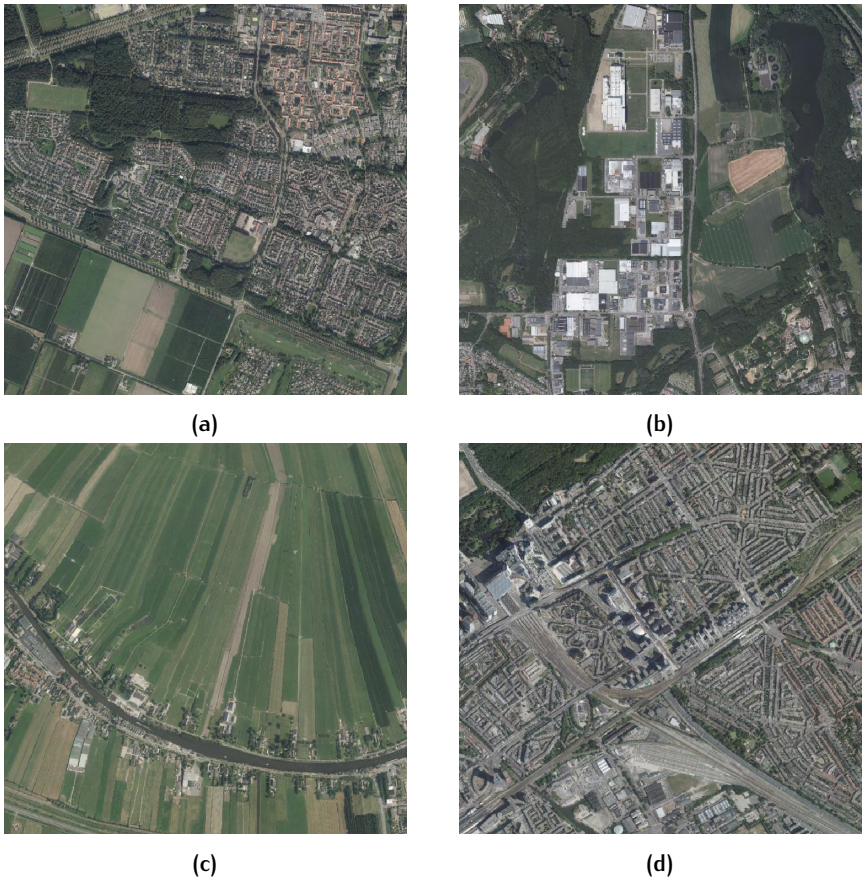


Figure 56: Aerial photographs covering the test data set areas. (a) Dronten. (b) Kerkrade. (c) Leiderdorp. (d) s-Gravenhage.

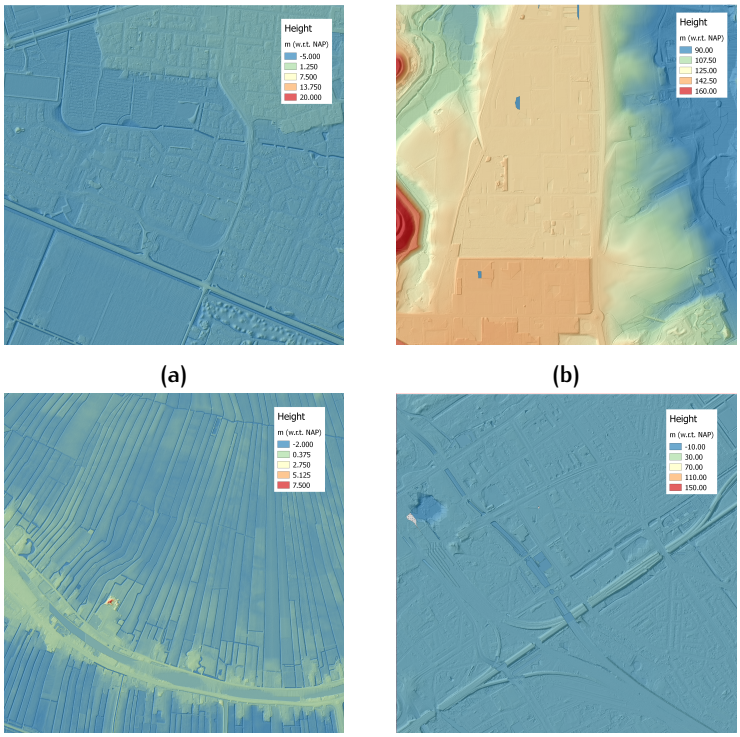


Figure 57: Digital elevation model. (a) Dronten. (b) Kerkrade. (c) Leiderdorp. (d) s-Gravenhage.

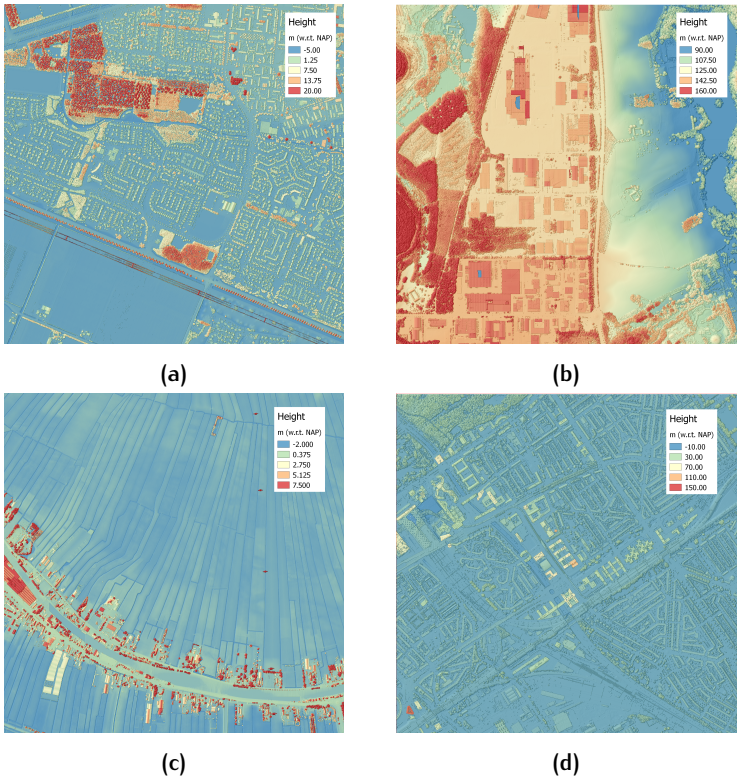


Figure 58: Digital surface model. (a) Dronten. (b) Kerkrade. (c) Leiderdorp. (d) s-Gravenhage.

5.3 VALIDATION OF INDIVIDUAL CLASSES

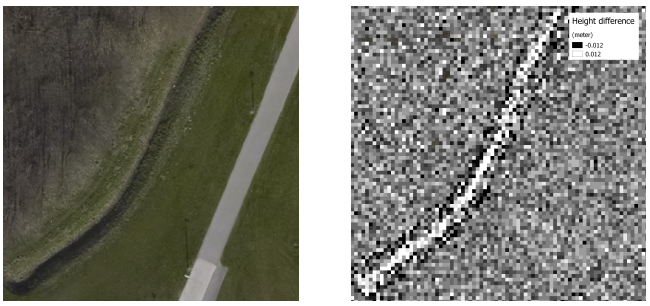
In subsection 2.4.1 and subsection 2.4.2 an oversight is given of the weaknesses of current raster-based height maps. In this section a visual comparison will take place in order to compare in which extend the methodology as proposed in chapter 4 is capable to solve these errors. The performance of the individual classes ground, vegetation and buildings generated according to the methodology proposed in chapter 4 will be validated.

5.3.1 Ground

PDOK provides raster-based height data that is generated with the application of IDW interpolation of the AHN2 point cloud data (see subsection 2.4.1). Kramer et al. [2014] assumes the height data of PDOK being correct, for that reason this data is used directly within the remainder of his method.

For the methodology described in subsection 4.3.2, high resolution raster-based height data is generated by gridding (indirect, see subsection 3.5.1) a vector-based TIN constructed from the AHN2 point cloud data. This methodology provides raster-based data that is nearly similar with respect to IDW interpolation that is applied by PDOK for the generation of their raster-based height map; height differences are measured at centimeter level ($-0.012 < \Delta h_{max} < 0.012$ meter) with a SD value of $5.9 * 10^{-3}$ meter with respect to the raster-based height map for the Dronten test data set. Height differences are near-random distributed, the biggest height differences are determined near edges of water bodies (Figure 59b).

Application of direct interpolation (subsection 3.5.1), height differences are achieved at sub-milimeter level ($-3.5 * 10^{-5} < \Delta h_{max} < 3.5 * 10^{-5}$ meter) with a SD value of $1.2 * 10^{-3}$ meter with respect to the raster-based height map for the Dronten test data set.



(a) (b)

Figure 59: Height differences between PDOK raster-based height data and the digital elevation model generated according to the methodology described in this thesis. (a) Aerial photograph. (a) Positive height differences (white) and negative height differences (black).

The conclusion is that interpolation based on a TIN generates raster-based height data with nearly similar heights in comparison with IDW interpolation, as applied for the generation of the raster-based height map of PDOK. The degree of smoothing when applying interpolation is the cause for the biggest height differences between both raster-based height maps. Indirect interpolation generates raster-based height data wit a higher degree of smoothness with respect to direct interpolation, resulting in better looking raster-based height maps.

Sink filling

Due to the characteristics of topographical LiDAR on water surfaces, point records are sparsely distributed on water bodies (see section 2.2). In subsection 2.4.1 it has been introduced that this leads to no-data values for raster cells covering water when applying IDW interpolation for the generation of a raster-based height map. Figure 60b shows a sample of the raster-based height map as generated by PDOK.

In subsection 2.4.3 a method has been introduced for the filling of no-data areas near water bodies by Kramer et al. [2014]. This method fills no-data areas by using the *TOP10NL* data set; for each polygon representing a water body the raster cell with the lowest height value is determined and the value is assigned to all raster cells within the polygon. Figure 60c shows a sample of this method; proper height data is transformed in wrong height data due to a poor positional accuracy of the used external data set. The quality of the methodology of Kramer et al. [2014] is dependent on the quality of the external data set that is used. For this reason, the usage of a more accurate data set, such as the Basisregistratie Topografie (BRT) does give a similar problem: missing or inaccurate data cannot be used or could lead to wrong output.

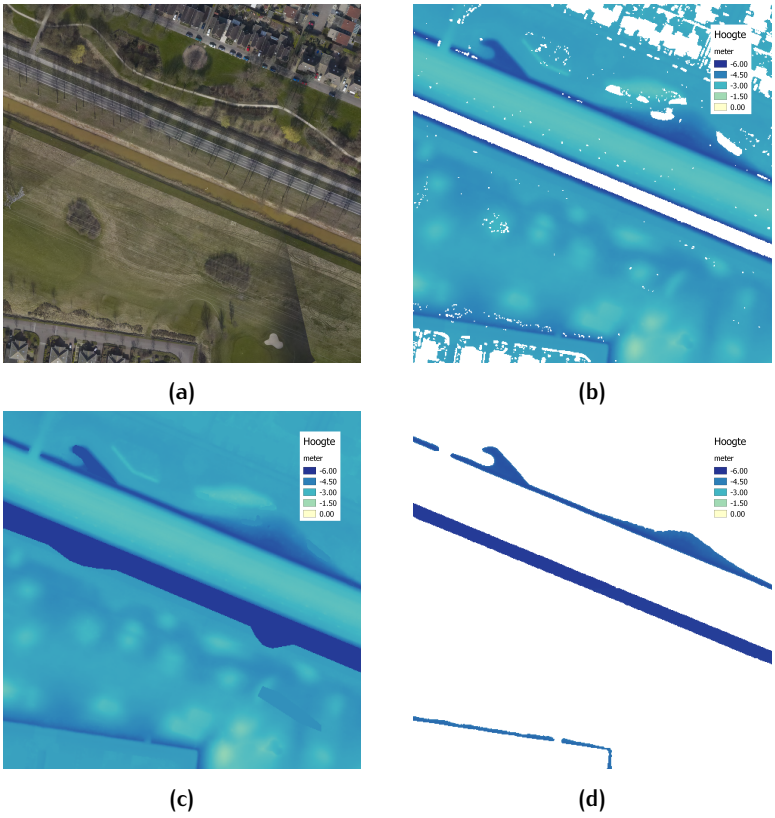


Figure 60: Visualization of the areas covered by the test data sets. (a) Aerial photograph. (b) PDOK. (c) Kramer et al. [2014], copyright Alterra, Wageningen UR. (d) The slope-based water detection algorithm described within this thesis.

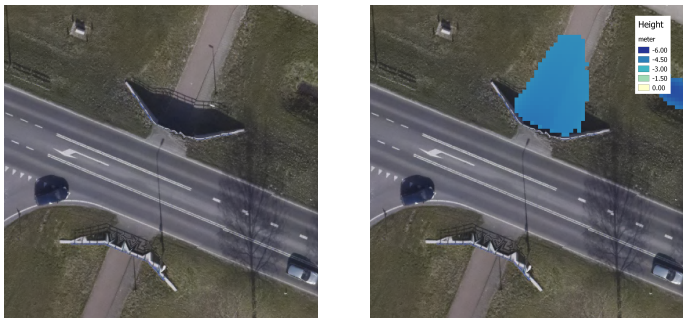
The slope-based water detection algorithm introduced in subsection 4.3.3 detects areas with a high potential on the presence of water. This algorithm is based on the concept that water always flows from a location with a higher height to a location with a lower height. The partly gridded DEM product is used as input in order to detect potential water bodies. The algorithm is capable to detect different kinds of water bodies, e.g. canals, (dry) ditches and small lakes. For areas that do not have slope near water bodies water is not detected. This leads to missing detection of water in wetlands, urban canals and quays. Although, for the Dronten test data set, the algorithm is capable to detect a higher amount of water bodies in comparison to the BRT data set (Figure 61).



(a) (b)

Figure 61: Detected water bodies. (a) Polygons (blue) from the BRT data set representing water bodies. (b) Water bodies (blue) detected with the slope-based water detection algorithm.

The slope-based water detection algorithm detects area with a high potential on the presence of water. Some areas could be classified as such while not being a real water body. For example, tunnels are within the applied parameterization detected and classified as a location with a high possibility on the presence of water (Figure 62), but due to sewerage systems this is not the case under normal circumstances.



(a) (b)

Figure 62: Falsely detected water bodies. (a) Aerial photograph. (b) Water bodies (blue) detected with the slope-based water detection algorithm near a tunnel.

Another disadvantage of slope-based water detection algorithm is that it is executed for each tile separately. For that reason it is possible that a larger water body, which is distributed over multiple tiles, could be detected in one tile but not in another. This will result in partly detected and classified water bodies.

Filling building footprints and local deviations

Another category of holes within the raster-based height data of PDOK that has been detected in subsection 4.3.2 are building footprints and local deviations. Figure 63b shows these holes clearly within the DEM near buildings, dynamic objects and dense vegetation.

Kramer et al. [2014] detects holes with respect to building footprints within the DEM from PDOK using of the BAG data set; the average height is calculated for each buffered building polygon separately and no-data raster cells within the buffered building polygon are assigned with the calculated height. Where the applied buffer size is unknown, the influence of the selected buffer size is high; not only the building footprint is filled; up to 20 meters outside the building footprint the calculated height is assigned to no-data raster cells (Figure 63c). In case that a building is located in a hilly area it can be expected that the application of a smaller buffer size would provide a rough estimation of the earth's surface near a buildings' footprint. The influence of the applied strategy is clearly visible.

In order to fill all remaining holes within the DEM of PDOK, Kramer et al. [2014] applies IDW interpolation. It can be assumed that this is a better strategy, also for the filling of building footprints. Using IDW interpolation based on neighboring raster cells containing a height value local deviations are taken into account in order to perform a more smooth reconstruction, even within larger no-data areas (Figure 63d).

In case of larger buildings, IDW interpolation could lead to artifacts within the building footprint. The applied *gdal_fillnodata.py* script, as part of GDAL, provides a flag to apply a predefined number of smoothing iteration based on a 3 x 3 average filter. Switching on this flag will lead to failures when running the algorithm; for this reason smoothing cannot be applied.

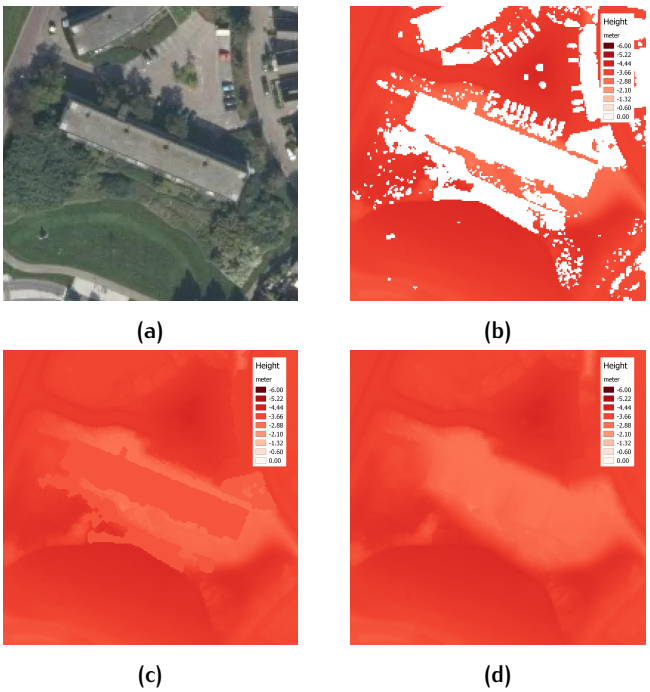


Figure 63: Filling building footprints and local deviations within a digital elevation model. (a) Aerial photograph. (b) PDOK. (c) Kramer et al. [2014], copyright Alterra, Wageningen UR. (d) Outcome of the methodology described in this thesis.

5.3.2 Buildings

In subsection 2.4.1 it is introduced that the raster-based height map of PDOK contains noisy height data near building edges. There are also holes present in the raster-based height data for raster cells covering buildings (Figure 64b). These holes are often correlated to building edges in combination with black surfaces or glass (Figure 64a).

The methodology of Kramer et al. [2014] fills these holes within the raster-based height map of PDOK by assigning the average height of nearby raster cells containing a height value within the corresponding building polygon from the BAG data set. This method based on an average value does not seem to be a proper approach in order to define a value for the filling of holes within buildings where building often have complex structures and many height differences. Figure 64c shows that the output of this methodology leads to appearance of new erroneous data. Additional, the methodology of Kramer et al. [2014] does not provide an answer with respect to the noisy height data that has been indicated within the raster-based height map of PDOK.

The methodology described within this thesis goes deeper with respect to the methodologies described above. As introduced in subsection 4.3.5, individual buildings are detected and the point data within each building is interpolated individually in order to develop a DBM. Rather then the definition of a height with respect to sea level (NAP) the normalized height with respect to the underlaying ground is defined for the DBM. Figure 64d shows the output of this methodology.

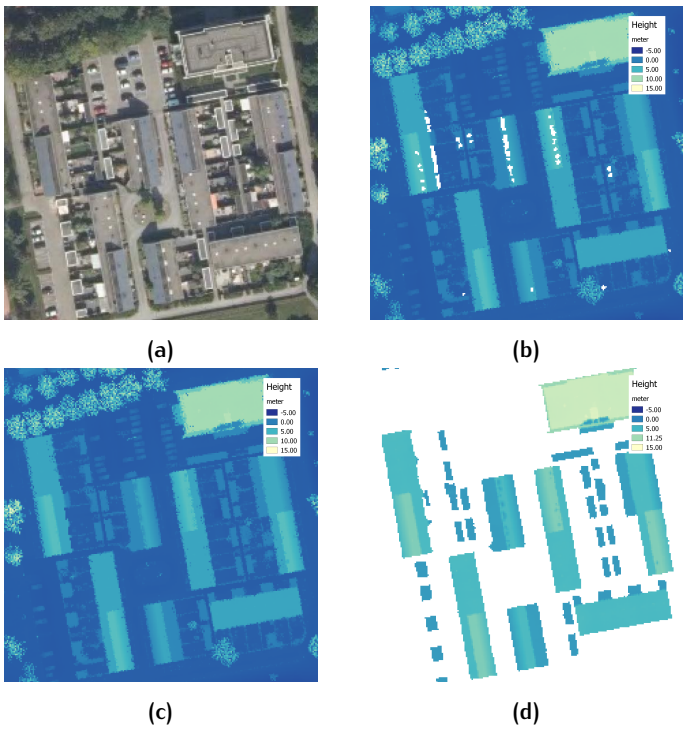
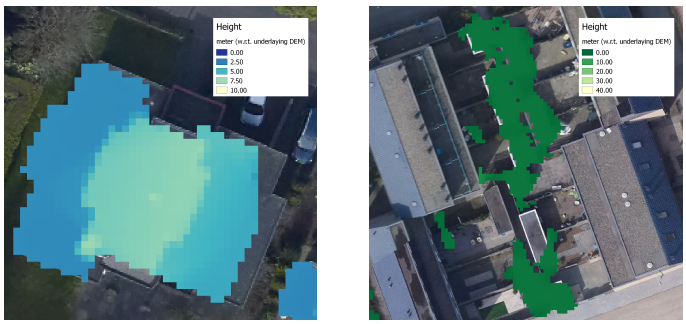


Figure 64: Filling of buildings within a digital surface model. (a) Aerial photograph. (b) PDOK. (c) Kramer et al. [2014], copyright Alterra, Wageningen UR. (d) Digital building model generated according to the methodology described in this thesis.

For larger buildings a high degree of data completeness is achieved; coverage is achieved for large parts of buildings. Main errors can be found near (complex) building edges. Due to missing point records or too low point densities it is not possible to generate a TIN covering the complete building (Figure 65a).

For smaller buildings (e.g. barns, garages) the data completeness is lower. The cause for this can be found in the classification step within the processing procedure; LiDAR points reflected on smaller buildings are most often classified as vegetation when using the chosen parameterization described in subsection 4.2.3. Within the method for DBM generation these points are not further processed, although these wrongly classified LiDAR points will be used within the generation of a CHM resulting in the presence of building data within the CHM (Figure 65b).



(a) (b)

Figure 65: Missing building data within a digital building model. (a) Digital building model with missing height data near a building edge due to missing point records or too low point densities. (b) Canopy height model containing barns that are falsely classified as vegetation due to a misclassification of LiDAR points.

Sometimes buildings remain (partly) absent within the generated DBM (Figure 66b). This happens in case of a too low point density due to errors during the collection of the LiDAR point data or because of the reflectance characteristics of building rooftops with dark surfaces or glass; the transmitted laser pulses will reflected weakly so that many laser points were missed. For certain scenarios it is impossible to reconstruct buildings properly and it is chosen to exclude the buildings from the DBM.



(a) (b)

Figure 66: Absent building data within a digital building model. (a) Aerial photograph. (b) Digital building model with absent building data due to a too low point density of the input point cloud.

5.3.3 Vegetation

In subsection 2.4.1 it is been introduced that the raster-based height data of PDOK contains height data that is determined wrongly with respect to raster cells covering vegetation. Due to the characteristics of LiDAR multiple returns will be recorded that are reflected on vegetation. Each of these returns ($1_{st}, 2_{th}, ..{}_{th}$) will have a different height value (section 2.2). The applied IDW interpolation method calculates a weighted average height resulting in an interpolated height that is probably somewhere in between the ground and the highest point record reflected on a treetop (Figure 67b).

Kramer et al. [2014] adopts the raster-based height data of PDOK and focuses on the filling of holes within the height data with respect to vegetation. Vegetation is detected by calculation of the NDVI derived from aerial photography. Estimation of height values in order to fill a hole takes place by calculation of the average height of close by raster cells covering vegetation containing a height value (Figure 67c). Holes within vegetation can be expected at cell-level due to the distribution of LiDAR points and the shape of vegetation; for this reason it is difficult to comment this methodology.

The methodology described within this thesis propose a strategy to generate a pit-free CHM, based on the generation of multiple partial CHMs (subsection 4.3.6). A TIN is generated with point records that are classified as vegetation and this TIN is gridded into a raster-based height map. This process is repeated a number of times for all points above a stepwise increasing height with respect to the underlaying ground and in the end all partial CHMs are merged into a CHM. This results in a normalized CHM that contains a lower degree of data pits, resulting in a more smooth representation of vegetation with respect to other raster-based height maps (Figure 67d).

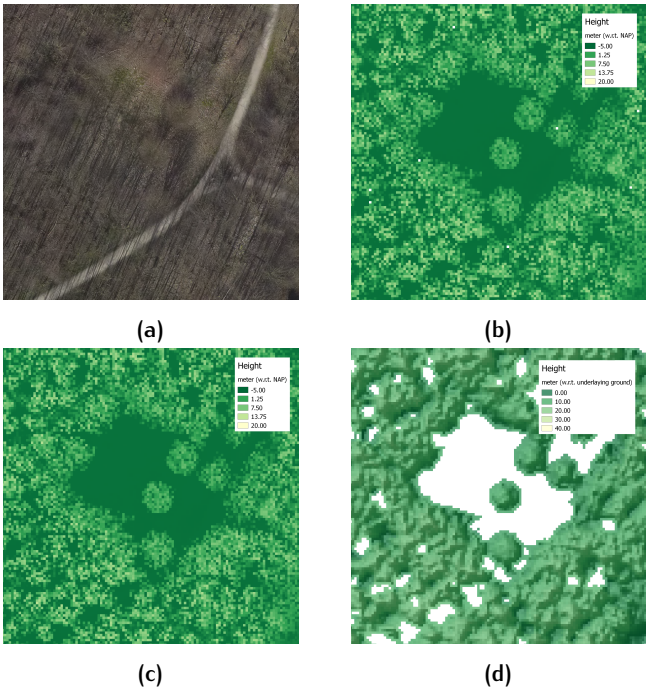


Figure 67: Filling of no-data areas within vegetation. (a) Aerial photograph. (b) PDOK. (c) Kramer et al. [2014], copyright Alterra, Wageningen UR. (d) Canopy height model generated according to the methodology described in this thesis.

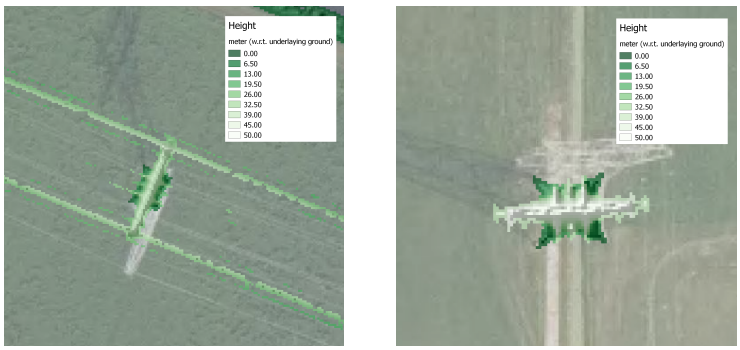
The height values stored within the CHM represents the highest point that is determined within each raster cell. When subtracting the raster-based height data of PDOK from the CHM height differences are measured up to decimeter level; $\Delta h_{max} = 25$ meter with a SD of 5.56 meter (Figure 68). These measurements prove that the applied methodology of PDOK for the generation of height data is wrong with respect to vegetation as already was indicated in subsection 2.4.1.



(a) (b)

Figure 68: Height differences between the raster-based height data of PDOK and the canopy height model generated according to the methodology described in this thesis. (a) Aerial photograph. (b) High height differences (white) and low height differences (black).

The applied classification algorithm described by Hug et al. [2004] misclassifies point data as vegetation when applying the parameterization selected in subsection 4.2.4. In subsection 5.3.2 it is already introduced that point data that is reflected on smaller buildings are mis-classified as vegetation. Also points reflected on electricity pylons and cables are often classified as vegetation. Results differ between the test data sets; in the Dronten test data set electricity cables are classified as vegetation (Figure 69a), where electricity cables are not classified as vegetation in the Leiderdorp test data set (Figure 69b). These differences in classification are probably due to differences in point cloud densities (see Appendix C).



(a) (b)

Figure 69: Electricity poles and cables within a canopy height model. (a) Dronten. (b) Leiderdorp.

5.3.4 Large-scale height map generation

Besides the validation of the individual classes it is also interesting to validate the performance of the methodology for the generation of large-scale height data. The methodology for pipelining as introduced in section 4.1 first collects, merge and clip the needed input tiles before generating overlapping tiles as input for the remainder of the processing procedure. Figure 70 shows a sample of a DEM that is generated on the intersection of four tiles. It can be concluded that the pipelining methodology that is introduced in section 4.1 is capable to generate height data without the occurrence of artifacts near the border of the original input tiles.

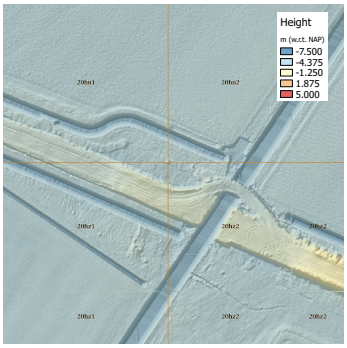
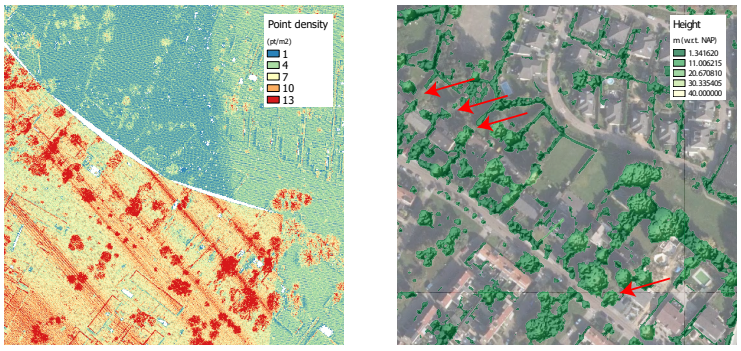


Figure 70: Digital elevation model generated with point cloud data from four input tiles.

Edges between sub-projects

Due to a temporal difference between the gathering of the different sub-projects of the AHN2 data set it might happen that there is point cloud data available with a temporal difference of one or more years within a tile or target area (see Figure 41). Figure 71a shows the edge between the different sub-projects; due to a difference in point density between the sub-projects the edge between them is easily recognizable. Alignment of the two projects is not perfect; small gaps having no point data can be detected on the edge between the sub-projects (Figure 71a). This results in missing raster-based height data when processing the point data (Figure 71b).



(a) (b)

Figure 71: Errors on the edge between sub-projects. (a) Missing point cloud data on the edge between sub projects. (b) Holes within a canopy height model due to missing point data on the edge between sub-projects.

5.4 QUALITY ASSESSMENT

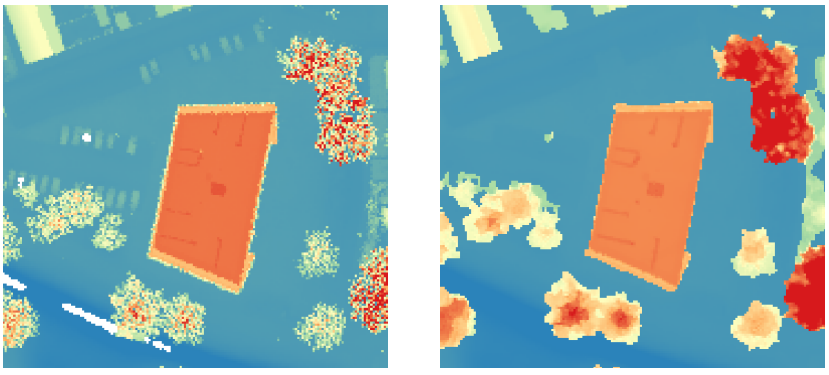
In this chapter the quality will be tested in a quantitative way for the data generated with the methodology as proposed in chapter 4. ISO19157 Geographic Information Quality principles will be used to assess data quality, introduced in section 3.7.

Assessing quality for only one data set without any reference data, is subjective and requires a high level of knowledge of the generation of processes. The operator also needs to be experienced to recognize deviations from the expected output and to predict the most useful kind of analysis [Podobnikar, 2009]. A better method for quality assessment is by comparing different data sets. In subsection 2.4.1 and subsection 2.4.2 two raster-based height maps are introduced. Where the raster-based height map produced by ESRI introduced in subsection 2.4.2 is only available in a web viewer this data cannot be used for quality assessment. For this reason, the following raster-based height maps will be used for quality assessment within the remainder of this section:

- Not filled DEM and DSM by PDOK [2014]
- Filled DEM by PDOK [2014]
- OHN DEM and DSM by Kramer et al. [2014]
- The proposed methodology in chapter 4

The raster-based height map products from PDOK and Kramer et al. [2014] are available as DEM and DSM. The methodology proposed in chapter 4 can generate a DEM but does not generate a DSM. For proper quality assessment, a DSM will be generated by merging the DEM, DBM and CHM. This method is supposed to filter out the remaining errors within raster-based DSMs as detected in subsection 2.4.1, the presence of noise: small urban objects which temporarily perturb the scene such as cars, roof antennas, cranes and other objects. A sample of this new-created DSM is depicted in Figure 72b.

Quality of the data sets will be assessed with respect to reference data. The spatial resolution of the reference data should be at least as high as expected from the tested data sets Podobnikar [2009]. For this reason, aerial photographs with a resolution of 0.07 meter will be used that are obtained during the collection of the AHN2 data set.



(a) (b)

Figure 72: Digital surface model generation. (a) A digital surface model from PDOK. (b) A digital surface model composed by merging multiple object classes.

5.4.1 Completeness

As introduced in section 3.7, completeness expresses the presence and absence of data, their attributes and relationships. There are two sub elements: commission (excess data present) and omission (data absent) [ISO, 2013].

The goal of this thesis research is to obtain universal raster-based DEM and DSM height maps that represents the earth's surface as best as possible. Final goal is a data completeness of 100% for both raster-based DEM and DSM height maps, where the output data does not have a deeper specific application.

	Dronten		Kerkrade		Leiderdorp		's-Gravenhage	
	C (%)	O (%)	C (%)	O (%)	C (%)	O (%)	C (%)	O (%)
Not-filled DEM	81.792537	18.207463	79.004762	20.995237	87.992625	12.007375	63.492169	36.507831
Filled DEM	83.659069	16.340931	80.373194	19.626806	88.632356	11.367644	65.095225	34.904775
OHN DEM	$1 * 10^2$	0	$1 * 10^2$	0	$1 * 10^2$	0	$1 * 10^2$	0
My DEM	$1 * 10^2$	0	$1 * 10^2$	0	$1 * 10^2$	0	$1 * 10^2$	0

Table 5: Completeness of digital elevation models expressed in commission (C) and omission (O) for test data sets.

Table 5 shows the completeness of raster-based DEM products for the test data sets. The not-filled DEM provided by PDOK scores commission rates between 63.492169% ('s-Gravenhage) and 87.992625% (Leiderdorp). The filled DEM is capable to have a slightly higher degree of completeness with commission rates thats scores between 0.639731% (Leiderdorp) and 1.866532% (Dronten) higher with respect to the not-filled DEMs. An explanation of these low commission rates is already provided in subsection 2.4.1.

The OHN DEM of Kramer et al. [2014] is capable to increase the commission rate for all test samples to 100% by filling all holes within the raster-based height map of PDOK. In a similar way the methodology as described in chapter 4 is capable to obtain commission rates of 100%.

	Dronten		Kerkrade		Leiderdorp		's-Gravenhage	
	C (%)	O (%)	C (%)	O (%)	C (%)	O (%)	C (%)	O (%)
Not-filled DSM	98.99411	1.00589	94.49192	5.50807	92.719375	7.280625	97.22864	2.77136
OHN DSM	$1 * 10^2$	0	$1 * 10^2$	0	$1 * 10^2$	0	$1 * 10^2$	0
My DSM	$1 * 10^2$	0	$1 * 10^2$	0	$1 * 10^2$	0	$1 * 10^2$	0

Table 6: Completeness of digital surface models expressed in commission (C) and omission (O) for test data sets.

Table 6 shows the completeness of raster-based DSM products for the test data sets. The not-filled DSM provided by PDOK scores commission rates between 92.719375% (Leiderdorp) and 98.99411% (Dronten). It can be concluded that a large part of the omission rate within the raster-based DEM height map is caused by above-ground objects that are part of the raster-based DSM height map. An explanation of the sources for omission is provided in subsection 2.4.1.

Comparable with the raster-based DEM products, the OHN DSM of Kramer et al. [2014] is capable to obtain commission rates for all test data sets of 100% by filling all holes within the raster-based height map of PDOK. In a similar way the methodology introduced in chapter 4 is capable to obtain a commission rate of 100%.

5.4.2 Logical consistency

As introduced in section 3.7, logical consistency is used primarily to specify conformance with certain topological rules [Salgé, 1995]. Spatial relations describe the spatial integrity of a geospatial dataset. Spatial integrity constraints are a tool for improving the internal quality of spatial data [Devillers and Jeansoulin, 2006].

Potentially, a grid cell can represent multiple classes: A house is build on ground and a tree is also located on the ground. A tree can also grow (partly) above a house. Vegetation can grow in the water and for the Netherlands it is also common that houses are located on the water. Given these examples there do not seem to be that many topological situations that could be defined as being illogic. For that reason, assessment of the logical consistency will not take place.

5.4.3 Positional accuracy

As introduced in section 3.7, in order to determine the positional accuracy, the closeness of provided data position values to values accepted as or being true, is the component of interest. Expression of the positional accuracy is determined by calculating the RMSE and SD. It needs to be taken into account that for data sets containing holes only raster cells with a determined height will taken into account.

No reference data is available, for that reason height differences is measured with respect to each other. RMSE is a commonly used method to document the vertical accuracy for raster-based height maps. Because a ground truth is missing when calculating the RMSE, error maps will be created in order to detect height deviations between the different raster-based height maps. Combined with reference aerial photographs it will be determined visually which raster-based height map contains the erroneous data.

Digital elevation model

In subsection 2.4.1 errors are explained within the raster-based DEM height map from PDOK. No errors related to wrongly determined height values are introduced, only errors with respect to holes within the data are determined. For this reason it can be stated that where heights are available, heights are determined in a proper way. The positional accuracy will be tested with respect to the filled DEM introduced in subsection 2.4.1 which has a slightly higher completeness with repsect to the non-filled DEM. Additional, the positional accuracy between the OHN DEM and my DEM will be determined. Table 7 shows the positional accuracy for all test data sets, some interesting things can be concluded:

- The filled raster cells within the filled DEM from PDOK are also interpolated *local* with respect to the not-filled DEM from PDOK. No differences are detected between both raster-based height maps for all test data sets when looking at the positional accuracy.
- Where the focus of the OHN DEM from Kramer et al. [2014] focuses on the filling of holes within the raster-based height data from PDOK it is surprising that height differences are measured on meter level. This is even larger than height differences measured between the filled DEM by PDOK and my DEM for most test data sets. Also RMSE values are significant higher for the OHN DEM in comparison to my DEM.

- Overall, a significant higher positional accuracy is achieved by my DEM with respect to the OHN DEM.

Also the positional accuracy between the OHN DEM and my DEM is measured. The measured parameters for this comparison are worst. This indicates that both models differ the most with respect to each other.

Digital surface model

In subsection 2.4.1 errors are explained within the raster-based DSM height map from PDOK. Both errors related to holes within the raster-based height data as well wrongly determined height values due to wrongly applied filtering and interpolation methods.

For this reason it is not possible, rather than it is for a DEM, to determine available height data as being correct. Nevertheless, the positional accuracy will be tested with respect to the not-filled DSM introduced in subsection 2.4.1. Additional, the positional accuracy between the OHN DSM and my DSM will be determined. Table 8 shows the positional accuracy for all test data sets, some interesting things that can be concluded are:

- The RMSE between the not-filled DSM and the OHN DSM does have smaller differences when calculating the RMSE and SD, rather than for my DSM. This shows that not much adjustments are done within the OHN DSM with respect to the not-filled DSM.

- In subsection 2.4.1 errors within the not-filled DSM are introduced, where in chapter 4 a methodology is proposed to solve these errors. My DSM does have the largest ΔH_{max}, RMSEs en SD with respect to the not-filled DSM for all test data sets. This indicates that many height values differ with respect to the not-filled DSM.

- The positional accuracy from my DSM with respect to the not-filled DSM and OHN DSM results in ΔH_{max}, RMSE and SD. This proves that my DSM is different with respect to both erroneous DSMs.

5.4.4 Temporal accuracy

As introduced in section 3.7, temporal accuracy refers to the agreement between encoded and 'actual' temporal coordinate system [Veregin, 1999]. It is the discrepancy between the actual attributes value and coded attribute value. A value is actual if it is correct in spite of any possible time-related changes in value. Thus currentness refers to the degree to which a database is up to date [Redman, 1992].

All raster-based height maps are based on the same data set. In section 3.4 it has been introduced that the data set does provide unrealistic information with respect to the date of collection. According to the meta data of the test data sets for 3 out of 4 of them the file creation date is within a time stamp of two days (Table 4). Based on the information provided by Van der Zon [2011] there is a temporal difference of 1-2 years between the gathering years of the AHN2 data set and the file creation date. It can be assumed that the file creation date is the date that the data is processed. The temporal accuracy can be determined as one year based on Van der Zon [2011] (see Figure 41).

	Dronten			Kerkrade			Leiderdorp			's-Gravenhage		
	ΔH_{max}	RMSE	SD	ΔH_{max}	RMSE	SD	ΔH_{max}	RMSE	SD	ΔH_{max}	RMSE	SD
Not-filled DEM	0.00	0.0000	0.0000	0.00	0.0000	0.0000	0.00	0.0000	0.0000	0.00	0.0000	0.0000
OHN DEM	2.36	0.0568	0.3121	5.59	0.0859	0.4812	2.25	0.0330	0.1981	5.63	0.0509	0.4543
My DEM	1.08	0.0054	0.0229	5.71	0.0185	0.1186	1.04	0.0154	0.0445	6.68	0.0301	0.2292
My DEM (w.r.t. **OHN** DEM)	2.87	0.0666	0.3139	5.72	0.2188	0.8078	2.48	0.1126	0.3098	10.19	0.4951	1.8893

Table 7: Accuracy assessment of digital elevation models for test data sets (in meters).

	Dronten			Kerkrade			Leiderdorp			's-Gravenhage		
	ΔH_{max}	RMSE	SD	ΔH_{max}	RMSE	SD	ΔH_{max}	RMSE	SD	ΔH_{max}	RMSE	SD
OHN DSM	40.05	3.5546	7.5510	21.22	0.1469	1.1118	76.25	0.3935	4.7111	36.58	0.1622	1.6135
My DSM	39.81	3.9822	8.1147	40.246	4.7380	7.9163	81.88	1.3701	6.33422	133.40	4.4689	13.5727
My DSM (w.r.t. **OHN** DSM)	39.81	3.9853	8.1164	40.246	4.7520	7.9275	81.88	1.3367	5.8172	132.71	4.6733	13.9945

Table 8: Accuracy assessment of digital surface models for test data sets (in meters).

5.4.5 Thematic accuracy

As introduced in section 3.7, the thematic accuracy compares the classes assigned to a feature or their attributes to a reference dataset or ground truth [ISO, 2013]. For a confidence level of 95% with a confidence interval of 5% based on a population of $16 * 10^6$ raster cells for each test data set, the thematic accuracy is tested for 384 random selected raster cells.

Digital elevation models

For measuring the thematic accuracy of DEMs a classification will be used containing the classes ground and water. Ground truth for all classes is extracted from aerial photography with a spatial resolution of 0.07 meter. In case of no-data values, raster cells will be qualified as unclassified. Within this subsection only the confusion matrices from the DEMs for the Dronten data set will be shown, outcome of the thematic accuracy from other test data sets will be showed in this subsection, the matrices from other test data sets can be found in Appendix D.

True condition

Predicted		Ground	Water	Total
	Ground	304	0	304
	Water	0	0	0
	Unclassified	74	6	80
	Total	378	6	384

Table 9: Confusion matrix for not-filled digital elevation model (Dronten).

True condition

Predicted		Ground	Water	Total
	Ground	304	0	304
	Water	0	0	0
	Unclassified	74	6	80
	Total	378	6	384

Table 10: Confusion matrix for filled digital elevation model (Dronten).

Table 9 shows the confusion matrix for the not-filled DEM for the Dronten data set. This raster-based height map does not contain information with respect to water, for that reason only a classification of ground points and unclassified points is possible. Sensitivity for the ground class is 80.42% and for the water class is 0%. Since only one class can be defined for this data set (ground), all other points are related to missing data and are for that reason classified as false positive points. For the other test data sets sensitivity rates of 67.65∼87.10 % are measured, differences are mainly do to the absence of building data in the not-filled DEMs. Water does have a sensitivity rate of 0% in all test data sets.

Table 10 shows the confusion matrix for the filled DEM for the Dronten data set. Despite an increase of the completeness of this data set of 1.87% results for the random selected points remains similar. For that reason the sensitivity remains the same; for the ground class sensitivity is 79.95% and for the water class it is 0%. For the other test data sets slightly higher sensitivity rates of 68.99∼94.82% are measured for the ground class, differences are mainly do to the absence of building data in the not-filled DEMs. Water does have a sensitivity rate of 0% in all test data sets.

		True condition		
		Ground	**Water**	**Total**
Predicted	**Ground**	377	7	384
	Water	0	0	0
	Unclassified	0	0	0
	Total	377	7	384

Table 11: Confusion matrix for the OHN digital elevation model (Dronten).

		True condition		
		Ground	**Water**	**Total**
Predicted	**Ground**	376	1	377
	Water	1	6	7
	Unclassified	0	0	0
	Total	377	7	384

Table 12: Confusion matrix for my digital elevation model (Dronten).

Table 11 shows the confusion matrix for the OHN DEM for the Dronten data set. Biggest difference with respect to previous measured DEMs is that, due to a higher degree in completeness, the data set does not contain any unclassified raster cells. The sensitivity for ground data is increased to 99.73%. The OHN DEM does not contain specified information related to water, for that reason the sensitivity for this class is 0%. For the other test data sets sensitivity rates of 88.00∼97.40% are measured for the ground class, differences are mainly do to the absence of building data in the not-filled DEMs. Water does have a sensitivity rate of 0% in all test data sets.

Table 12 shows the confusion matrix for my data set. Biggest with respect to the previous DEMs is that my DEM contains information regarding water. Sensitivity for ground data decreased slightly to 99.73% due to a raster cell that is classified as water but is ground in the real world. Where my DEM contains information regarding water, sensitivity rate for this class is 85.71%. For the other test data sets sensitivity rates of 94.41∼98.93% are measured for the ground class. Water does have a sensitivity rate of 59.38∼83.72% in all test data sets.

Digital surface models

When measuring the thematic accuracy for DSMs the confusion matrix needs to be expanded further. The methodology proposed in chapter 4 provides a differentiation for above-ground objects in buildings and vegetation, the not-filled DSM and OHN DSM do not. For those data sets, thematically accuracy will be determined whether a raster cell contains to the surface or that it is unclassified.

Table 13 shows the confusion matrix for the not filled DSM for the Dronten data set. A lower degree of points are unclassified, which is in line with the conclusion of subsection 5.4.1 that indicates a higher degree in completeness for the same raster-based height map. For the generalized surface the sensitivity is 99.07%. For the other test data sets sensitivity rates of 93.49∼97.40% are measured for the generalized surface class, differences are mainly do to the absence of building data in the not-filled DEMs.

Table 14 shows the confusion matrix for the OHN DSM for the Dronten data set. The biggest improvement of this raster-based height map is that no points are unclassified. For the generalized surface the sensitivity is 100.00%, similar sensitivity rates are achieved for all other test data sets.

Predicted \ True condition	Ground	Water	Building	Vegetation	Total
Surface	213	1	58	102	375
Ground	0	0	0	0	0
Water	0	0	0	0	0
Building	0	0	0	0	0
Vegetation	0	0	0	0	0
Unclassified	2	6	1	0	9
Total	215	7	59	102	384

Table 13: Confusion matrix for not-filled digital surface model (Dronten).

Predicted \ True condition	Ground	Water	Building	Vegetation	Total
Surface	215	7	59	102	384
Ground	0	0	0	0	0
Water	0	0	0	0	0
Building	0	0	0	0	0
Vegetation	0	0	0	0	0
Unclassified	0	0	0	0	0
Total	215	7	59	102	384

Table 14: Confusion matrix for OHN digital surface model (Dronten).

Predicted \ True condition	Ground	Water	Building	Vegetation	Total
Surface	0	0	0	0	0
Ground	210	1	0	1	212
Water	0	6	0	0	6
Building	0	0	59	1	60
Vegetation	5	0	0	101	106
Unclassified	0	0	0	0	0
Total	215	7	59	103	384

Table 15: Confusion matrix for not-filled digital surface model (Dronten).

Table 15 shows the confusion matrix for the my DSM for the Dronten data set. With respect to the previous DSMs, this data set differentiates the surface in different classes, an oversight of the sensitivity of all classes is provided in Table 17.

5.5 EVALUATION

Quality assessment has showed that a total approach, from LiDAR point cloud to raster-based height map, is capable to improve the quality of these raster-based height maps based on the AHN2 data set.

	Dronten	Kerkrade	Leiderdorp	's-Gravenhage
Completeness	$1 * 10^2$%	$1 * 10^2$%	$1 * 10^2$%	$1 * 10^2$%
Positional accuracy				
RMSE (w.r.t. not-filled DEM)	0.0054 m	0.0185 m	0.0154 m	0.0301 m
Thematic accuracy				
Ground	97.67%	99.33%	99.69%	99.48%
Water	85.71%	83.72%	56.00%	76.92%

Table 16: Quality of digital surface models for test data sets.

Table 16 shows that for all test data sets a completeness of 100% is achieved. The positional accuracy can be determined at centimeter level with respect to current DEM from PDOK, where the OHN DEMs have a lower positional accuracy due to wrong processing of data, see subsection 2.4.3. The methodology proposed in chapter 4 is capable to generate raster-based height data with a high thematic accuracy for ground with rates between 97.67∼99.69%. Thematic accuracy is lower for water bodies, with rates between 56.00∼85.71%. This indicates that a slope-based approach needs additional sources in order to detect water with a higher thematic accuracy.

	Dronten	Kerkrade	Leiderdorp	's-Gravenhage
Completeness	$1 * 10^2$%	$1 * 10^2$%	$1 * 10^2$%	$1 * 10^2$%
Positional accuracy				
RMSE (w.r.t. not-filled DEM)	3.9822 m	4.7380 m	1.3701 m	4.4689 m
Thematic accuracy				
Ground	97.67%	99.33%	99.69%	99.48%
Water	85.71%	83.72%	56.00%	76.92%
Buildings	100.00%	100.00%	94.74%	91.09%
Vegetation	98.00%	100.00%	100.00%	100.00%

Table 17: Quality of digital surface models for test data sets.

Table 17 shows that, similar as the DEM products, a completeness is achieved of 100%. Where the positional accuracy is lower with respect to currently existing DSMs this is mainly caused by the applied methodology for the generation of DBMs and CHMs as proposed in chapter 4. The Leiderdorp test data set, that contains less buildings and vegetation, has a relative higher positional accuracy proves this.

Thematic accuracy for ground, buildings and vegetation is high with accuracy rates between 91.09∼100% for all test data sets. Similar as for the DEM products, the thematic accuracy is lower for water bodies, with rates between 56.00∼85.71%.

6 CONCLUSION, DISCUSSION AND FUTURE WORK

In this last chapter the conclusions of this thesis research will be given. First in section 6.1 the research questions will be answered that are introduced in section 1.2. Secondly, in section 6.2 a discussion will take place about the chosen methodology as applied within this thesis. As third, in section 6.3 an overview will be given of future work within the field of automatic generation of raster-based height maps.

6.1 CONCLUSION

What kinds of errors are most common within currently existing raster-based height maps based on the AHN2 data set?

Two groups of currently existing raster-based height maps can be distinguished, having their own product-specific errors:

- The first group of raster-based height maps are DEMs; representing a bare earth representation without any above-ground objects.
- The second group of raster-based height maps are DSMs; representing the first echo/return the laser received for each laser pulse send out.

The first group of raster-based height maps, the DEMs, are based on a filtered version of the AHN2 data set. This product is based on semi-automatic filtering of ground points in combination with additional manual editing in order to meet the requirements that are defined by this point cloud product by Rijkswaterstaat and the Dutch water boards.

Errors within DEMs are due to no-data holes that appear after spatial interpolation of the point cloud data (subsection 2.4.1). Where LiDAR beams do not hit the ground no data is available. Identified causes for missing data with respect to the ground are filtering of above-ground objects (e.g. houses, vegetation and dynamic objects that perturb the scene) and the characteristics of *topograpic* LiDAR (e.g. water). Where raster data is available this height data is indicated being a proper representation of the real-world surface.

The second group of raster-based height maps, DSMs, are based on the complete AHN2 data set. It has been observed within this thesis that no further filtering of erroneous point records (e.g. multipath) is applied (subsection 2.4.1). Besides missing data, DSMs contain height data that is determined in a wrong way. This results in raster cells that do not represent a true height and the presence of data pits. Also dynamic objects that perturb the scene (e.g. cars, cranes) are present within currently existing DSMs.

Methodologies that fill no-data holes within both DEMs and DSMs do change height values for raster cells with a known estimated height.

What strategy is appropriate for the processing of large amounts of point cloud data to raster-based height maps?

A divide-and-conquer strategy that decomposes massive data in overlapping tiles is sufficient in general. Selection of a proper tile and buffer size is not only dependent of the size of the input point cloud data; further increase in file size during the processing within the remainder of the pipeline needs to be taken into account. A-trial-and-error method is indicated being the best method to determine proper parameterization. For the implementation of the divide-and-conquer strategy a tile size of 200 by 200 meter is applied and a buffer size of 25 meters within the methodology.

For the detection of geographic objects that exceeds the tile size (e.g. large water bodies, rivers), it can happen that an object is detected in one tile but not in any adjacent tile, due to the local presence of characteristics in order to detect a certain geographic object.

Given a point cloud sample, which algorithm or methodology is best filter out different classes of information?

No methodology or algorithm exist that can guarantee a 100% correct classification of LiDAR points automatically. Results of comparisons strongly differ based on both the characteristics of the input data and the characteristics of the terrain.

This thesis has showed that filtering of (above-)ground does perform worse results then current available point cloud products. Also, the filtering of above-ground objects as buildings and vegetation lacks performance due to a standard parameterization where point cloud densities differ in density, not only between the different test data sets, but also within test data sets. Another issue is the complexity of the terrain; there is a clear correlation between the failure of the applied point cloud classification algorithm and the complexity of the terrain characteristics.

What interpolation technique is most appropriate to estimate a height at a given location for different classes of objects?

No best method exist for the interpolation of dense 3-dimensional LiDAR point data ($1 > \text{point}/\text{m}^2$) into 2.5-dimensional raster data. Similar as for the classification of point cloud data the results of comparisons strongly differ based on both the characteristics of the input data and terrain characteristics. More simple interpolation algorithms as IDW or interpolation based on a TIN do not perform worse then more complex geo-statistical methods.

Ground

For the generation of a raster-based DEM a hybrid methodology based on the characteristics of the point cloud data and the terrain's characteristics works out well. Where point cloud data is available, interpolation based on a TIN can estimate heights at well. Selection of a proper cut-off threshold is necessary in order to prevent the occurrence of artifacts of the applied interpolator where data is sparser distributed: triangles within a TIN where the longest edge is longer then the defined cut-off threshold will not be used when converting the vector-based TIN into a raster-based height map.

Sparser distributions of point cloud data are detected near water, buildings local deviations. Sink filling is applied for areas detected as having a high probability being water, using a slope-based strategy. Where no-data holes are present within the detected water bodies, these raster cells will be filled with the minimal height value as detected within a water body. This strategy does not detect water directly, it estimates areas with a high probability on the presence of water. Remaining holes within a raster-based DEM are filled with IDW interpolation.

Buildings

Interpolation of building data, so-called DBMs, should only take place within building boundaries. Edge-constrained interpolation based on a TIN is introduced as a method in order to determine height data only within the interior of buildings. Building boundaries are detected in a complementary strategy that combines boundary extraction from point records classified as building and external 2D geodata from the BAG data set.

Data pits within building data, caused by noise within the LiDAR data is filtered out by a two-step strategy that thickens the point cloud first and thins it afterwards has showed to result in a smooth representation of raster-based building data within the DBM.

Vegetation

CHM are generated by gridding a TIN that is constructed from all point records classified as vegetation. Multiple partial CHMs are generated iteratively using point records with a normalized height above an increasing threshold until the highest normalized vegetation points are lower then the threshold value. Merging the partial CHMs results in the generation of a CHM that preserves the morphological structure of individual tree crowns better having less data pits in comparison with standard CHMs.

To which extent can external 2D geodata sets help to improve the quality of raster-based height maps?

Where there does not exist any algorithm or methodology that is capable to apply a 100% correct classification external geodata sets can help in order to identity and correct data within the processing pipeline during the classification and interpolation of point cloud data.

When using external 2D geodata a differentiation in positional and temporal accuracy with respect to the point cloud data set needs to be taken into account.

For the detection of building boundaries, within the edge-constrained interpolation of DBMs it has been proved that the BAG data set is capable to indicate building boundaries that are complementary with respect to the ones detected by point records classified as building.

What quantitative degree of quality can be achieved for raster-based height maps generated from AHN2 point cloud data by the application of an automated process?

Three parameters are indicated as usable for measuring the quality of raster-based height maps :

- Completeness
- Positional accuracy
- Thematic accuracy

For data completeness a maximum degree is achieved where the proposed methodology is capable to deliver DEMs and DSMs that do not contain any holes.

The positional accuracy has been calculated for both raster-based DEMs and DSMs. For DEMs the positional accuracy is high with respect to currently available raster-based height maps. For DSMs a lower positional accuracy is detected with respect to currently available raster-based height maps. This is due to erroneous processing of point cloud data within the processing procedure of them where the proposed methodology within this thesis solves these errors.

For thematic accuracy it has been proved that the proposed methodology is capable to classify more object with respect to currently existing raster-based height maps with a sensitivity rate that ranges between 56∼100% ± 5%.

In general it can be concluded that a higher quality is achieved with respect to currently existing raster-based height maps. Since no reference height data is available it has not been possible to measure the outcome of the proposed methodology with respect to a ground truth.

6.2 DISCUSSION

This thesis have proved that it is possible to generate raster-based height maps with a higher quality than currently available products. The potential has been showed in order to develop derivative products related to objects in the build environment such as buildings and vegetation. However, the methodology proposed within this thesis does not provide 100% correct results. This is caused both due to characteristics of LiDAR as well as the proposed methodology:

- No good insights are available in the filtering algorithm that is applied for point classification within this thesis. Despite the fact that good results are achieved it can be assumed that, when having a better insight in a used filtering algorithm, it is possible to achieve a better classification. In order to improve the proposed methodology for the generation of a raster-based height map it can be expected that the biggest improvements can be made during the filtering phase, eventually combined with manual editing.
- Application of the AHN2 data set is to obtain information with respect to the ground. For the collection of the point cloud data chosen is to obtain data early-spring when there are not much leaves on the trees. For this reason the vegetation within CHMs is represented in an order that is smaller than it is in reality. For obtainment of a CHM collection of LiDAR data should take place later in spring or during summer.
- Due to the characteristics of some building surfaces it is not possible to fully reconstruct buildings using LiDAR techniques. Examples of building surfaces can be roofing felts, or surfaces with a high reflectivity. Another issue are glass surfaces, due to its translucent characteristics LiDAR beams penetrate through glass, resulting in missing data for small skylights, but also for large glass building structures. Also in a situation where vegetation is covering parts of buildings it does not seem to be possible to reconstruct covered building parts without involvement of any structure related to vegetation.

- The results presented within this thesis are a simplified and generalized version of the data that can be created potentially within the methodology proposed within this thesis. It has been shown that there is potential for improvement, besides data quality, for trivial matters as:
  - Increased spatial resolution
  - Specific modeling for urban objects such as water, vegetation and building models

6.3 FUTURE WORK

In this section an overview of suggestions will be provided for further improvement of the methodology proposed within this theses.

Digital elevation modeling

For the detection and filling of water bodies a slope-based method is applied on tile level within this thesis. Where adjacency with respect to neighboring tiles is not taken into account, this is currently problematic for water bodies that exceed the size of a tile. Another imperfection is that not all water bodies can be detected by the proposed slope-based method. Besides this methodology, external 2D geodata sets containing information with respect to water bodies could be used in order to increase the detection of water bodies within raster-based height maps. The completeness of the BRT data set is not 100%. Nevertheless, a combined approach that makes use of a combination of both the slope-based approach and the BRT data set in order to detect (the lowest height within a) water body could lead to better results. Also the determined heights for sink filling are determined at tile-level. This approach creates unrealistic, plan form flow patterns since all topographic information within the sink is discarded. For the generation of correct water flow networks carving or breaching are techniques that can be used in order to cut into the DEM by creating a descending path from the bottom of the sink to the nearest point that is lower than that of the bottom of the sink. The objective of stream carving is to link dead-end pathways into the main network in the most realistic manner.

Digital building modeling

Building boundaries are extracted based on a concavity threshold for individual building polygons within the methodology proposed in this thesis. These building boundaries are directly stored as polygons without any further processing. The level of detail of building boundaries is limited due to factors like the availability of other data sources, the density of LiDAR data and the complexity of the scene. Alharthy and Bethel [2002] present a method in order to optimize building footprints to regular vector building shapes as connected, rectilinear line segments.

Within LASboundary, that is applied in order to extract building boundaries, it is not possible to define different concavity values for inner and outer boundaries of building polygons. This results often in small inner boundaries within building polygons due to locally lower point densities on buildings. Based on the size or/and perimeter of these holes filtering of unintentional could take place additional.

Canopy height modeling

More research is needed in order to perform better point classification. Since different strategies are proposed for different point classes a wrong classification will lead to wrongly processed output data. By performing a better point classification points are processed within a tailor-made method, resulting in better raster-based height maps.

Further optimization

Besides future work with respect to improvement of the data quality, also possibilities for speed optimization could be researched. The Python modules *subprocess* and *multiprocessing* cannot be used interchangeable resulting in inefficient usage of processing power resulting in time consuming processing of the developed scripts. Rewriting of scripts into the C++ programming language looks potentially interesting.

AHN3

After the availability of the AHN1 and AHN2 data set, the AHN3 data set will be published in several portions from 2015 until 2018. So far, there are no official documents available regarding specifications and goals when this thesis is written. The general sense from the participants within the AHN project team is focused to adjust from point cloud densification to change detection. It can be expected that there will not be a significant quality change in the AHN3 point cloud collection with respect to point density.

In section 2.2 it has been introduced that there is no more data available but the XYZ-coordinates for each point within the AHN2 data set. For the AHN3 point cloud data it more meta data is available, for example:

- number of return
- intensity
- classification
- scan angle rank
- GPS time

For this reason, meta data that has been added to AHN2 point cloud data during the processing steps within this thesis are already available within the meta data of the AHN3 data set. No further research is done with respect to the degree of correctness of the meta data for the AHN3 data set.

BIBLIOGRAPHY

Agarwal, P. K., Arge, L., and Danner, A. (2006). *From point cloud to grid DEM: A scalable approach*. Springer.

Agarwal, P. K., Arge, L., and Yi, K. (2005). I/o-efficient construction of constrained delaunay triangulations. In *Algorithms–ESA 2005*, pages 355–366. Springer.

Agterberg, F. P. (1974). *Geomathematics: mathematical background and geoscience applications*. Elsevier Scientific Publishing Company Amsterdam.

Akay, A. E., Oğuz, H., Karas, I. R., and Aruga, K. (2009). Using lidar technology in forestry activities. *Environmental monitoring and assessment*, 151(1-4):117–125.

Alexander, C., Smith-Voysey, S., Jarvis, C., and Tansey, K. (2009). Integrating building footprints and lidar elevation data to classify roof structures and visualise buildings. *Computers, Environment and Urban Systems*, 33(4):285–292.

Alharthy, A. and Bethel, J. (2002). Heuristic filtering and 3d feature extraction from lidar data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):29–34.

Ali, T. A. (2004). On the selection of an interpolation method for creating a terrain model (tm) from lidar data. In *Proceedings of the American Congress on Surveying and Mapping (ACSM) Conference*. Citeseer.

Arcadis (2012). Wat kan het AHN2 betekenen voor BGT?

ASPRS (2013). ASPRS LIDAR Data Exchange Format Standard Version 1.4.

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

Bailly, J.-S., Monestiez, P., and Lagacherie, P. (2006). Modelling spatial variability along drainage networks with geostatistics. *Mathematical Geology*, 38(5):515–539.

BAO (2013). Processenhandboek bag, basisregistraties adressen en gebouwen.

Bartels, M., Wei, H., and Mason, D. C. (2006). Dtm generation from lidar data using skewness balancing. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 566–569. IEEE.

Bater, C. W. and Coops, N. C. (2009). Evaluating error associated with lidar-derived dem interpolation. *Computers & Geosciences*, 35(2):289–300.

Behrendt, R. (2012). Introduction to lidar and forestry, part 1: a powerful new 3d tool for resource managers. *The Forestry Source*, pages 14–15.

Ben-Arie, J. R., Hay, G. J., Powers, R. P., Castilla, G., and St-Onge, B. (2009). Development of a pit filling algorithm for lidar canopy height models. *Computers & Geosciences*, 35(9):1940–1949.

Beutel, A. (2011). *From Point Cloud to 2D and 3D Grids: A Natural Neighbor Interpolation Algorithm using the GPU*. PhD thesis, Duke University.

Blaschke, T., Tiede, D., and Heurich, M. (2004). 3d landscape metrics to modelling forest structure and diversity based on laser scanning data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(8/W2):129–32.

Burrough, P. A. (1986). Principles of geographical information systems for land resources assessment.

Chaplot, V., Darboux, F., Bourennane, H., Leguédois, S., Silvera, N., and Phachomphon, K. (2006). Accuracy of interpolation techniques for the derivation of digital elevation models in relation to landform types and data density. *Geomorphology*, 77(1):126–141.

Charaniya, A. P., Manduchi, R., and Lodha, S. K. (2004). Supervised parametric classification of aerial lidar data. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 30–30. IEEE.

Chew, L. P. (1989). Constrained delaunay triangulations. *Algorithmica*, 4(1-4):97–108.

Cho, W., Jwa, Y.-S., Chang, H.-J., and Lee, S.-H. (2004). Pseudo-grid based building extraction using airborne lidar data. *International Archives of Photogrammetry and Remote Sensing*, 35(B3):378–381.

Clark, M. L., Clark, D. B., and Roberts, D. A. (2004). Small-footprint lidar estimation of sub-canopy elevation and tree height in a tropical rain forest landscape. *Remote Sensing of Environment*, 91(1):68–89.

Cressie, N. (1990). The origins of kriging. *Mathematical geology*, 22(3):239–252.

Davies, B., Biggs, J., Williams, P., Whitfield, M., Nicolet, P., Sear, D., Bray, S., and Maund, S. (2008). Comparative biodiversity of aquatic habitats in the european agricultural landscape. *Agriculture, Ecosystems & Environment*, 125(1):1–8.

Delaunay, B. (1934). Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2.

Devillers, R. and Jeansoulin, R. (2006). *Fundamentals of spatial data quality*. ISTE Publishing Company.

Döllner, J. and Hinrichs, K. (2000). An object-oriented approach for integrating 3d visualization systems and gis. *Computers & Geosciences*, 26(1):67–76.

ESRI (2014). AHN2 50cm - shaded relief.

Fancher, Z. (2012). Using ArcGIS 10.0 to develop a LiDAR to Digital Elevation Model workflow for the U.S. Army Corps of Engineers, Sacramento District Regulatory Division.

Gaveau, D. L. and Hill, R. A. (2003). Quantifying canopy height underestimation by laser pulse penetration in small-footprint airborne laser scanning data. *Canadian Journal of Remote Sensing*, 29(5):650–657.

Geodan (2014). Dynamic Holland Shading.

Geodatastyrelsen (2013). Danmarks hojdemodel bliver bedre og mere nojagtig.

Guan, X. and Wu, H. (2010). Leveraging the power of multi-core platforms for large-scale geospatial data processing: Exemplified by generating dem from massive lidar point clouds. *Computers & Geosciences*, 36(10):1276–1282.

Guptill, S. C. and Morrison, J. L. (2013). *Elements of spatial data quality*. Elsevier.

Haining, R. P. (2003). *Spatial data analysis*. Cambridge University Press Cambridge.

Hengl, T. (2006). Finding the right pixel size. *Computers & Geosciences*, 32(9):1283–1298.

Horn, B. (1981). Hill shading and the reflectance map. *Proceedings of the IEEE*, 69(1):14–47.

Hu, Y. (2001). Analysis and processing of airborne lidar data.

Hu, Y. (2004). *Automated extraction of digital terrain models, roads and buildings using airborne LiDAR data*, volume 69.

Hug, C., Krzystek, P., and Fuchs, W. (2004). Advanced LIDAR data processing with LasTools.

Huisman, O. and Rolf, A. (2009). Principles of geographic information systems.

Hyyppä, J., Hyyppä, H., Leckie, D., Gougeon, F., Yu, X., and Maltamo, M. (2008). Review of methods of small-footprint airborne laser scanning for extracting forest inventory data in boreal forests. *International Journal of Remote Sensing*, 29(5):1339–1366.

Isenburg, M., Liu, Y., Shewchuk, J., and Snoeyink, J. (2006a). Streaming computation of delaunay triangulations. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1049–1056. ACM.

Isenburg, M., Liu, Y., Shewchuk, J., Snoeyink, J., and Thirion, T. (2006b). Generating raster dem from mass points via tin streaming. In *Geographic Information Science*, pages 186–198. Springer.

ISO (2013). ISO19157:2013 Geographic information - Data Quality.

Kadaster (2014a). *Basisregistratie Topografie: Catalogus en Productspecificaties*.

Kadaster (2014b). *Controleprotocol TOP10NL*.

Kadaster (2014c). Ontwikkeling 3D-kaart van Nederland in volle gang.

Khosravipour, A., Skidmore, A. K., Isenburg, M., Wang, T., and Hussin, Y. A. (2014). Generating pit-free canopy height models from airborne lidar. *Photogrammetric Engineering & Remote Sensing*, 80(9):863–872.

Kramer, H., Clement, J., and Mucher, C. (2014). OHN: Object Hoogten Nederland, de hoogte van alles wat boven het maaiveld uitsteekt. *GeoInfo*, 3:18–21.

Kraus, K. and Otepka, J. (2005). Dtm modelling and visualization–the scop approach.

Lee, C. Y. (1961). An algorithm for path connections and its applications. *Electronic Computers, IRE Transactions on*, (3):346–365.

Li, D., Yao, Y., Shao, Z., and Wang, L. (2014). From digital earth to smart earth. *Chinese Science Bulletin*, 59(8):722–733.

Li, Q., Unger, A., Sudicky, E., Kassenaar, D., Wexler, E., and Shikaze, S. (2008). Simulating the multi-seasonal response of a large-scale watershed with a 3d physically-based hydrologic model. *Journal of hydrology*, 357(3):317–336.

Li, S., MacMillan, R., Lobb, D. A., McConkey, B. G., Moulin, A., and Fraser, W. R. (2011). Lidar dem error analyses and topographic depression identification in a hummocky landscape in the prairie region of canada. *Geomorphology*, 129(3):263–275.

Li, Z., Zhu, C., and Gold, C. (2004). *Digital terrain modeling: principles and methodology*. CRC press.

Liu, X. (2008a). Airborne lidar for dem generation: some critical issues. *Progress in Physical Geography*, 32(1):31–49.

Liu, X. (2008b). Airborne lidar for dem generation: some critical issues. *Progress in Physical Geography*, 32(1):31–49.

Liu, X. and Zhang, Z. (2008). Lidar data reduction for efficient and high quality dem generation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37:173–178.

Liu, X., Zhang, Z., Peterson, J., and Chandra, S. (2007). The effect of lidar data density on dem accuracy. In *Proceedings of the International Congress on Modelling and Simulation (MODSIM07)*, pages 1363–1369. Modelling and Simulation Society of Australia and New Zealand Inc.

Lloyd, C. and Atkinson, P. (2002). Deriving dsms from lidar data with kriging. *International Journal of Remote Sensing*, 23(12):2519–2524.

Lohr, U. (1998). Laserscanning for dem generation. *GIS technologies and their environmental applications*, pages 243–249.

Luethya, J. and Stengeleb, R. (2005). 3d mapping of switzerland - challenges and experiences. *Foot*, 3000:4900.

MacDonald, S., Szafron, D., and Schaeffer, J. (2004). Rethinking the pipeline as object-oriented states with transformations. In *High-Level Parallel Programming Models and Supportive Environments, 2004. Proceedings. Ninth International Workshop on*, pages 12–21. IEEE.

Maidment, D. R. (1996). Gis and hydrologic modeling-an assessment of progress. In *Third International Conference on GIS and Environmental Modeling*.

Mark, D. M. and Aronson, P. B. (1984). Scale-dependent fractal dimensions of topographic surfaces: an empirical investigation, with applications in geomorphology and computer mapping. *Journal of the International Association for Mathematical Geology*, 16(7):671–683.

McCullagh, M. (1988). Terrain and surface modelling systems: theory and practice. *The photogrammetric record*, 12(72):747–779.

McInerney, D. and Kempeneers, P. (2015). Virtual rasters and raster calculations. In *Open Source Geospatial Tools*, pages 163–170. Springer.

Meng, X., Currit, N., and Zhao, K. (2010). Ground filtering algorithms for airborne lidar data: A review of critical issues. *Remote Sensing*, 2(3):833–860.

Mitas, L. and Mitasova, H. (1999). Spatial interpolation. *Geographical information systems: principles, techniques, management and applications*, 1:481–492.

Palmer, T. C. and Shan, J. (2002). Comparative study on urban visualization using lidar data in gis. *URISA Journal*, 14(2):19–25.

Parker, J. A., Kenyon, R. V., and Troxel, D. E. (1983). Comparison of interpolating methods for image resampling. *Medical Imaging, IEEE Transactions on*, 2(1):31–39.

PDOK (2014).

Pearlstone, L. (2010). Interpolated surfaces using delaunay triangulation.

Peucker, T. K., Fowler, R. J., Little, J. J., and Mark, D. M. (1978). The triangulated irregular network. In *Amer. Soc. Photogrammetry Proc. Digital Terrain Models Symposium*, volume 516, page 532.

Podobnikar, T. (2005). Suitable dem for required application. In *Proceedings of the 4th International Symposium on Digital Earth*.

Podobnikar, T. (2009). Methods for visual quality assessment of a digital terrain model. *SAPI EN. S. Surveys and Perspectives Integrating Environment and Society*, (2.2).

Podobnikar, T. and Vrečko, A. (2012). Digital elevation model from the best results of different filtering of a lidar point cloud. *Transactions in GIS*, 16(5):603–617.

Popescu, S. C., Wynne, R. H., and Nelson, R. F. (2002). Estimating plot-level tree heights with lidar: local filtering with a canopy-height based variable window size. *Computers and Electronics in Agriculture*, 37(1):71–95.

Priestnall, G., Jaafar, J., and Duncan, A. (2000). Extracting urban features from lidar digital surface models. *Computers, Environment and Urban Systems*, 24(2):65–78.

Pu, S. and Zlatanova, S. (2005). Evacuation route calculation of inner buildings. In *Geo-information for disaster management*, pages 1143–1161. Springer.

Pudelko, R. (2007). TIN Model (Triangulated Irregular Networks).

Redman, T. C. (1992). *Data quality: management and technology*. Bantam Books, Inc.

Reuter, H., Hengl, T., Gessler, P., and Soille, P. (2009). Preparation of dems for geomorphometric analysis. *Developments in Soil Science*, 33:87–120.

Rietdijk, M., Ellenkamp, Y., and Wevers, R. (2008). Geometry en de BAG.

Rijkswaterstaat Meetkundige Dienst.

Robinson, A. H. (1960). *Elements of cartography*, volume 90. LWW.

Salgé, F. (1995). Semantic accuracy. *Elements of spatial data quality*, pages 139–151.

Sande, C. v. d., Soudarissanane, S., and Khoshelham, K. (2010). Assessment of relative accuracy of ahn-2 laser scanning data using planar features. *Sensors*, 10(9):8198–8214.

Senay, G., Ward, A., Lyon, J., Fausey, N., and Nokes, S. (1998). Manipulation of high spatial resolution aircraft remote sensing data for use in site-specific farming. *Transactions of the ASAE-American Society of Agricultural Engineers*, 41(2):489–496.

Septima (2014). Preview of denmark's future elevation model.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM.

Sibson, R. (1981). A brief description of natural neighbour interpolation. *Interpreting multivariate data*, 21:21–36.

Sithole, G. and Vosselman, G. (2005). Filtering of airborne laser scanner data based on segmented point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(part 3):W19.

Skiena, S. S. (1998). *The algorithm design manual: Text*, volume 1. Springer Science & Business Media.

Srivastava, R. N. (2008). Spatial data quality: An introduction.

Swart, L. T. (2010). How the up-to-date height model of the netherlands (ahn) became a massive point data cloud. *NCG KNAW*, 17.

Swart, R. (2009). Actueel Hoogtebestand Nederland (AHN): verschillen tussen AHN-1 en AHN-2.

Tao, C. and Hu, Y. (2001). A review of post-processing algorithms for airborne lidar data. In *CD-ROM Proceedings of ASPRS Annual Conference, April*, pages 23–27.

Taylor, G., Li, J., Kidner, D., Brunsdon, C., and Ware, M. (2007). Modelling and prediction of gps availability with digital photogrammetry and lidar. *International Journal of Geographical Information Science*, 21(1):1–20.

Tinkham, W. T., Huang, H., Smith, A., Shrestha, R., Falkowski, M. J., Hudak, A. T., Link, T. E., Glenn, N. F., and Marks, D. G. (2011). A comparison of two open source LiDAR surface classification algorithms. *Remote Sensing*, 3(3):638–649.

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic geography*, pages 234–240.

Toppe, R. (1987). Terrain models-a tool for natural hazard mapping. *Avalanche formation, movement and effects, IAHS Publ*, 162:629–638.

Tukay Mapping.

Van den Brink, L., Krijtenburg, D., Van Eekelen, H., and Maessen, B. (2013). *Basisregistratie grootschalige topografie Gegevenscatalogus BGT 1.1.1*.

Van der Zon, N. (2011). Kwaliteitsdocument AHN-2.

Veregin, H. (1999). Data quality parameters. *Geographical information systems*, 1:177–189.

Vitter, J. S. (2001). External memory algorithms and data structures: Dealing with massive data. *ACM Computing surveys (CsUR)*, 33(2):209–271.

Vögtle, T. and Steinle, E. (2003). On the quality of object classification and automated building modeling based on laserscanning data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(Part 3):W13.

Volkova, U. (2014). Opportunities for LIDAR to improve and validate tree data sets in the Netherlands. Master's thesis.

Vosselman, G. V. and Maas, H.-G. (2010). *Airborne and terrestrial laser scanning*. Whittles.

Wang, Y., Mercer, B., Tao, V. C., Sharma, J., and Crawford, S. (2001). Automatic generation of bald earth digital elevation models from digital surface models created using airborne ifsar. In *Proceedings of 2001 ASPRS Annual Conference*, pages 23–27.

Watson, D. (2013). *Contouring: a guide to the analysis and display of spatial data*. Elsevier.

Wehr, A. and Lohr, U. (1999). Airborne laser scanning: an introduction and overview. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):68–82.

Wichmann, V. (2012). LiDAR Point Cloud Processing with SAGA.

Wouda, B. (2011). Visualization on a Budget for Massive LiDAR Point Clouds. *Delft University of Technology*.

Zimmerman, D., Pavlik, C., Ruggles, A., and Armstrong, M. P. (1999). An experimental comparison of ordinary and universal kriging and inverse distance weighting. *Mathematical Geology*, 31(4):375–390.

A REFLECTION

This thesis proposes a methodology for the automatic generation of raster-based height data for the Netherlands based on the AHN2 data set. The research has taken place between November 2014 and January 2016. Where the initial planning supposed a literature study followed by the development of a prototype in reality both processes have been applied, in an iterative way resulting in the proposed strategy consisting of three different height models; (i) a DEM, (ii) a DBM and (iii) a CHM. Rather then the generation of a DSM it has been indicated that different strategies for different objects needs to be applied in order to generate object-specific height data with a higher quality.

The relationship between the methodical line of approach of the Master Geomatics and the method applied within this thesis consists of the obtainment, pipelining, classification and interpolation of massive LiDAR data. Another part of the research consist of the generation, manipulation and visualization. These topics are in line with the courses Sensing technologies, GIS and cartography, Python programming and Geo Datasets & Quality that are part of the Geomatics track and could not have been applied without having knowledge with respect to these information.

The relationship between the research and application of the field geomatics is that the output of the presented methodology can be used as input for a broad range of Geomatics-related applications. Height maps are often used as input in GIS and are the most common basis for digitally-produced relief maps. DSMs can be useful for applications such as landscape modeling, city modeling and visualization applications while a DTM is often required for applications such as flood or drainage modeling, land-use studies, geological applications, and other applications.

The relationship between the work presented within this thesis and the wider social context is that the applications for which raster-based height data can be used as input to predict floodings, light and shadow simulation and many more applications in a social context. The generation of higher quality raster-based height data will lead to better input data contributing to higher quality output for output applications.

B OPEN 2D GEODATA SETS IN THE NETHERLANDS

Currently most open 2D geographic vector-based data in the Netherlands is based on aerial photography and this data is generated already for many years. For that reason it can be assumed that currently available open 2D geodata sets have a degree of quality. In this chapter an oversight will be given in currently available 2D geodata sets in order to introduce their characteristics.

B.0.1 TOP10NL

TOP10NL is an object-oriented topographical vector file for the Netherlands which can be used on a scale between 1:5 000 and 1:25 000. The *TOP10NL* data model contains a collection of topographical base objects, related to a reproduction scale of 1:10 000 which have been included as object classes (Figure 73). In the main structure of *TOP10NL*, every geographical object is assigned to a specific object class. The current set of object classes consists of 'Road section', 'Railway section', 'Water section', 'Building', 'Land', 'Planimetric feature', 'Relief', 'Registrational area', 'Geographic area' and 'Functional area'. A geographical object has certain geometry (point, line or polygon) and is characterized by its attributes furthermore. The data within the *TOP10NL* data set is obtained by aerial photography [Kadaster, 2014a]. For the remainder of this section only the classes 'Water section' and 'Building' are relevant for a deeper introduction.

Water section

For the *Water section* class geometry is available both as line and polygon geometry. Where line geometry is used for water sections with a maximum width of 6 meters, the minimum area for polygon geometry is 50 m2. For both kinds of geometry a maximum positional deviation of 3 meters is allowed [Kadaster, 2014b]. An example of *TOP10NL* geometry for a water section is showed in Figure 74b.

Building

Building geometry is only available as polygons. Most important rule is that the minimum size of a building is at least 9 m2. A maximum positional deviation of 3 meters is allowed [Kadaster, 2014b]. An example of *TOP10NL* geometry for a building is shown in Figure 75b.



Figure 73: Vector-based map based on the *TOP10NL* data set.

B.0.2 BAG

The BAG is part of the basic registrations of the Dutch government. The BAG data set contains information about all buildings in the Netherlands, which are classified in five different classes:

- Buildings ('panden')
- Stay objects ('verblijfsobjecten')
- Number designations ('nummeraanduidingen')
- Public spaces ('openbare ruimtes')
- Residences ('ligplaatsen')

Attributes for the different object classes can be:

- Status
- Surface size
- Geometry
- X,Y coordinate
- Year of construction
- Purpose

Information for all object classes is provided by the different Dutch municipalities. Despite the definition of rules for the creation of data (see [BAO, 2013]), much non-uniformal data is present within the BAG data set. The minimal positional accuracy of building geometry can differ from 0.6 meter in urban areas up to even 1.2 meter in rural areas. Nevertheless, most buildings have a minimal positional accuracy of 0.28 meter for urban areas up to 0.56 meter in rural areas. More information about definitions of geometry with relation to the BAG data set can be found in Rietdijk et al. [2008]. An example of BAG geometry for a building is shown in Figure 75c.

B.0.3 BGT

The Basisregistratie Grootschalige Topografie (BGT) is a data set that is currently under construction and will be available nationwide starting 2016. The BGT data set is more or less the successor of the *TOP10NL* data set and similar to that data set the BGT contains geographic objects as 'Road section', 'Railway section', 'Water section' and more.

Where geographical object within the *TOP10NL* data set can have different geometry classes (point, line or polygon), the BGT data set defines all geographic data as polygon geometry, all data is (geometrical) stored the same. Another difference between *TOP10NL* and BGT is that different data classes are provided by different stakeholders; for example national government provides information about highways, provinces provide information with respect to the roads they maintain, while municipalities have the responsibility to provide information about remaining roads. Similar as for the BAG data set this can lead to the distribution of non-uniformal data within the BAG data set, even despite there exist definitions for the creation of data. According to these definitions defined by Van den Brink et al. [2013], the positional accuracy of data for the BGT data set depends on the need. For data that requires a high accuracy (e.g. buildings) the positional accuracy is between 0.3 meter and 0.6 meter, while for data that requires a lower positional accuracy (e.g. water) a minimal accuracy is defined at 0.6 meter.

B.0.4 Comparison between TOP10NL, BAG and BGT

In this chapter a number of open 2D geo data sets have been intro, a comparison of them is given in Table 18. It can be concluded that the BAG and BGT data sets have a higher positional accuracy with respect to the TOP10NL data set. Nevertheless, it is needed to take into account that both the BAG as well as the BGT data set have multiple contributors which might lead to a variable quality of the geo data.

	Data	Number of contributors	Positional accuracy (m)
TOP10NL	Buildings + Water	1	6.00
BAG	Buildings	multiple	0.28-1.20
BGT	Buildings + Water	multiple	0.30-0.60

Table 18: Comparison between the TOP10NL, BAG and BGT data sets.

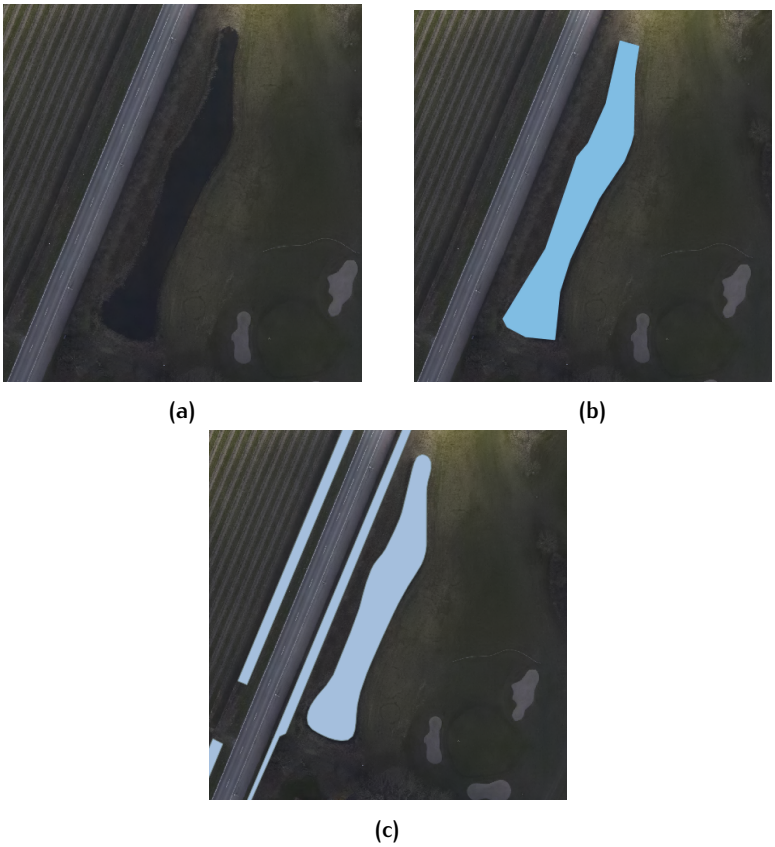


Figure 74: Comparison of TOP10NL and BGT. (a) Aerial photograph of a water section. (b) Polygon representing the water section in the TOP10NL data set. (b) Polygon representing the water section in the BGT data set.

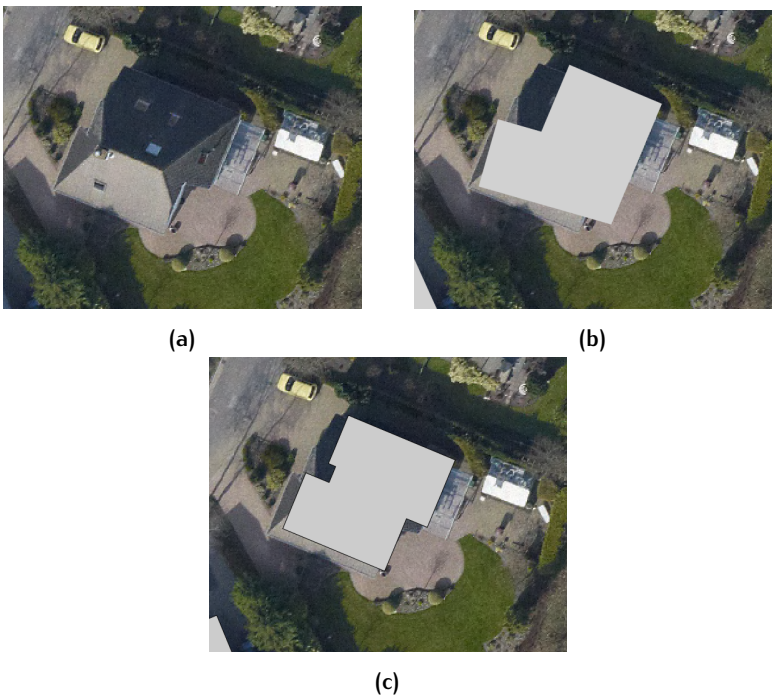


Figure 75: Comparrison of TOP10NL and BAG. (a) Aerial photograph of a house. (b) Polygon representing the house in the TOP10NL data set. (b) Polygon representing the house in the BAG data set

C POINT CLOUD ANALYSIS

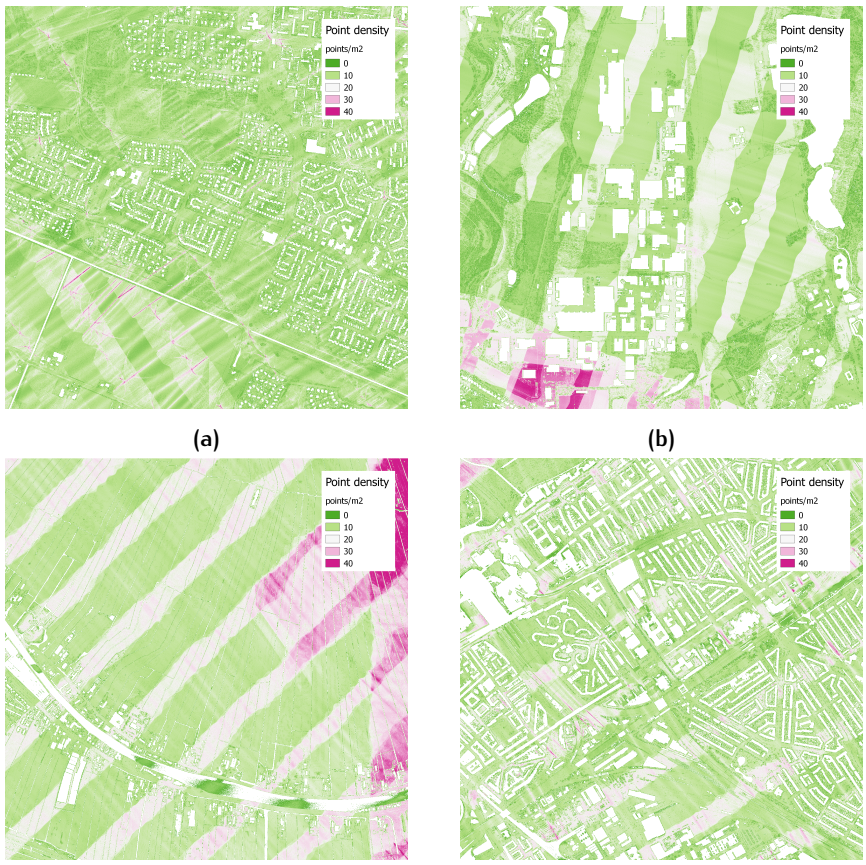


Figure 76: Point density of the filtered AHN2 point cloud product colored as dark green (0) to purple (40) per square meter. (a) Dronten. (b) Kerkrade. (c) Leiderdorp. (d) s-Gravenhage.

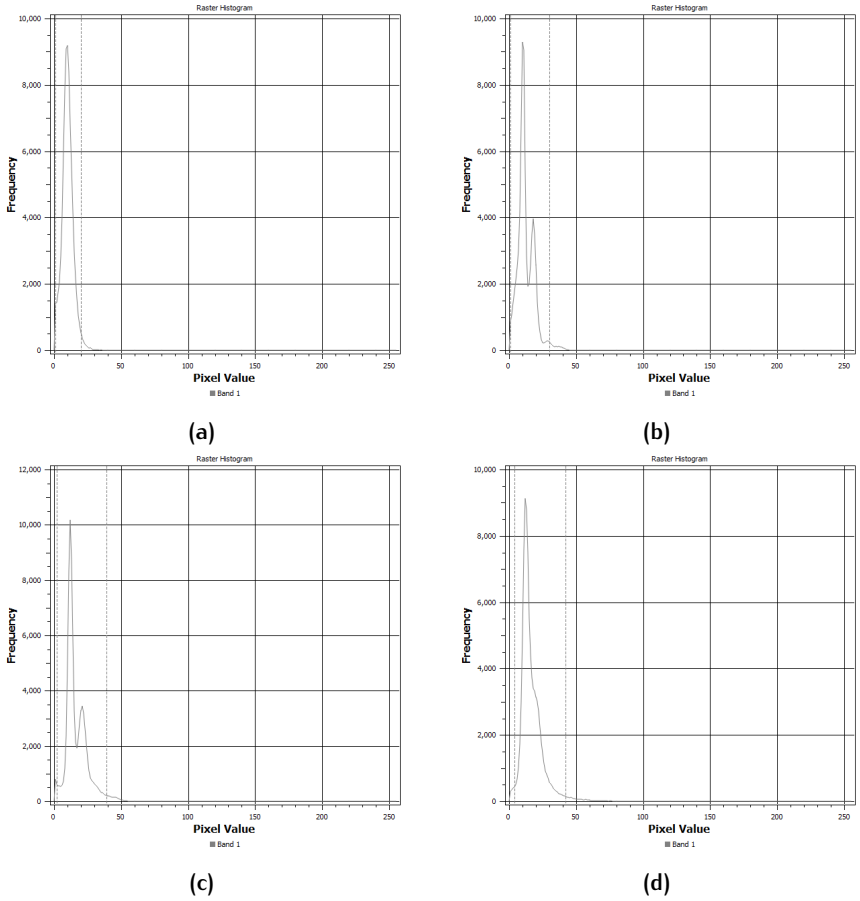


Figure 77: Histogram of the average point density per square meter of the filtered AHN2 point cloud product. (a) Dronten (average = 10.16). (b) Kerkrade (average = 12.57). (c) Leiderdorp (average = 17.42). (d) s-Gravenhage (average = 11.79).

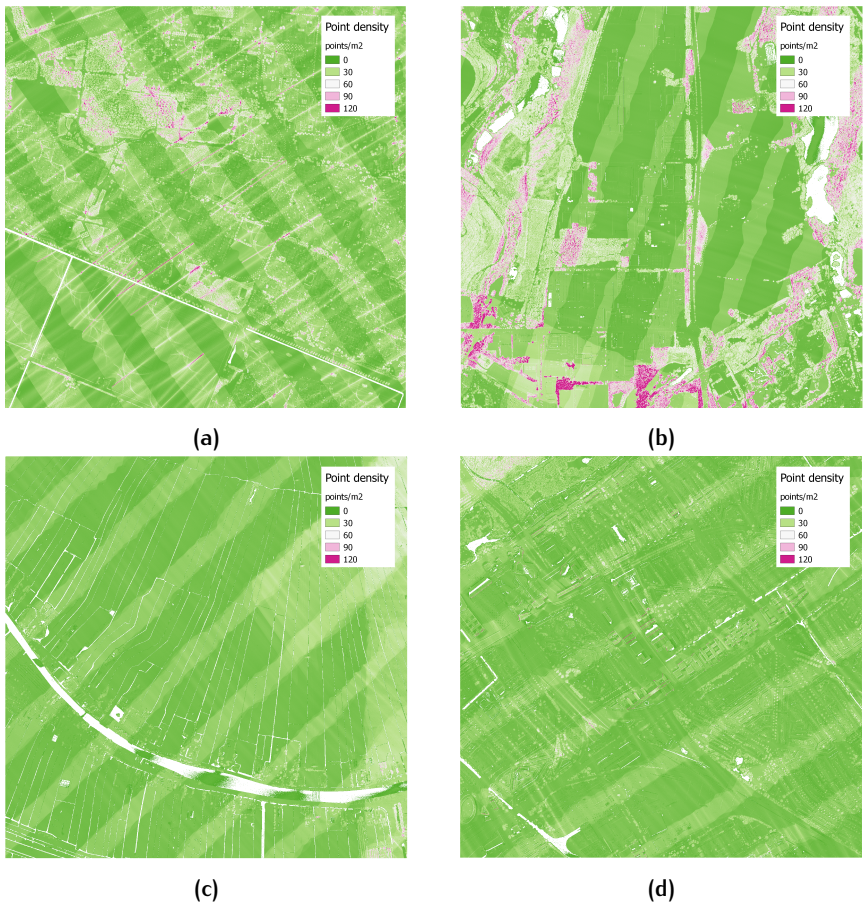


Figure 78: Point density of the filtered + unfiltered AHN2 point cloud product colored as dark green (2) to purple (104) per square meter. (a) Dronten. (b) Kerkrade. (c) Leiderdorp. (d) s-Gravenhage.

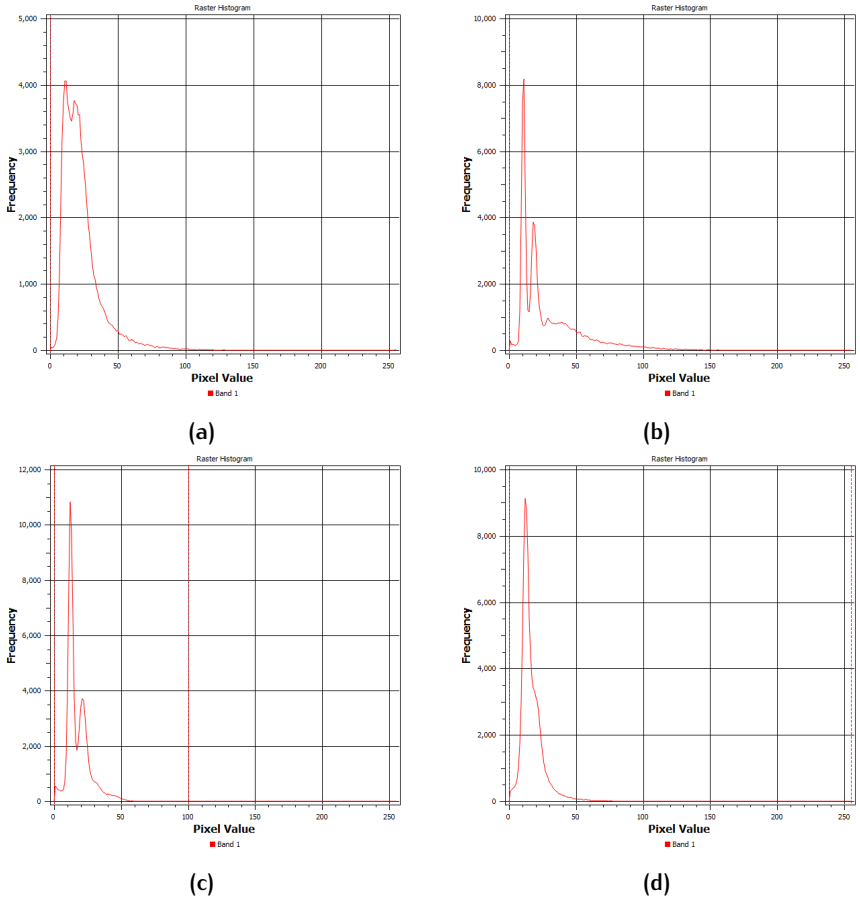


Figure 79: Histogram of the average point density per square meter of the filtered + unfiltered AHN2 point cloud product. (a) Dronten (average = 23.14). (b) Kerkrade (average = 16.43). (c) Leiderdorp (average = 29.50). (d) s-Gravenhage (average = 16.60).

D THEMATIC ACCURACY

D.0.5 Kerkrade

Digital elevation model

True condition

Predicted		Ground	Water	Total
	Ground	297	3	300
	Water	0	0	0
	Unclassified	44	40	84
	Total	341	43	384

Table 19: Confusion matrix for not-filled digital elevation model (Kerkrade).

True condition

Predicted		Ground	Water	Total
	Ground	299	3	302
	Water	0	0	0
	Unclassified	42	40	82
	Total	341	43	384

Table 20: Confusion matrix for filled digital elevation model (Kerkrade).

True condition

Predicted		Ground	Water	Total
	Ground	341	43	384
	Water	0	0	0
	Unclassified	0	0	0
	Total	341	43	384

Table 21: Confusion matrix for the OHN digital elevation model (Kerkrade).

True condition

Predicted		Ground	Water	Total
	Ground	341	7	348
	Water	0	36	36
	Unclassified	0	0	0
	Total	341	43	384

Table 22: Confusion matrix for my digital elevation model (Kerkrade).

Digital surface model

		True condition				
		Ground	**Water**	**Building**	**Vegetation**	**Total**
Predicted	**Surface**	150	23	25	164	362
	Ground	0	0	0	0	0
	Water	0	0	0	0	0
	Building	0	0	0	0	0
	Vegetation	0	0	0	0	0
	Unclassified	0	20	2	0	22
	Total	150	43	27	164	384

Table 23: Confusion matrix for not-filled digital surface model (Kerkrade).

		True condition				
		Ground	**Water**	**Building**	**Vegetation**	**Total**
Predicted	**Surface**	150	43	27	164	384
	Ground	0	0	0	0	0
	Water	0	0	0	0	0
	Building	0	0	0	0	0
	Vegetation	0	0	0	0	0
	Unclassified	0	0	0	0	0
	Total	150	43	27	164	384

Table 24: Confusion matrix for OHN digital surface model (Kerkrade).

		True condition				
		Ground	**Water**	**Building**	**Vegetation**	**Total**
Predicted	**Surface**	0	0	0	0	0
	Ground	149	2	0	0	151
	Water	0	36	0	0	36
	Building	0	5	27	0	32
	Vegetation	1	0	0	164	165
	Unclassified	0	0	0	0	0
	Total	149	43	27	164	384

Table 25: Confusion matrix for my digital surface model (Kerkrade).

D.0.6 Leiderdorp

Digital elevation model

True condition

Predicted		Ground	Water	Total
	Ground	329	2	331
	Water	0	0	0
	Unclassified	23	30	53
	Total	352	32	384

Table 26: Confusion matrix for not-filled digital elevation model (Leiderdorp).

True condition

Predicted		Ground	Water	Total
	Ground	330	2	332
	Water	0	0	0
	Unclassified	22	30	52
	Total	352	32	384

Table 27: Confusion matrix for filled digital elevation model (Leiderdorp).

True condition

Predicted		Ground	Water	Total
	Ground	352	32	384
	Water	0	0	0
	Unclassified	0	0	0
	Total	352	32	384

Table 28: Confusion matrix for the OHN digital elevation model (Leiderdorp).

True condition

Predicted		Ground	Water	Total
	Ground	352	13	365
	Water	0	19	19
	Unclassified	0	0	0
	Total	352	32	384

Table 29: Confusion matrix for my digital elevation model (Leiderdorp).

Digital surface model

		True condition				
		Ground	**Water**	**Building**	**Vegetation**	**Total**
Predicted	**Surface**	317	9	18	14	358
	Ground	0	0	0	0	0
	Water	0	0	0	0	0
	Building	0	0	0	0	0
	Vegetation	0	0	0	0	0
	Unclassified	2	23	1	0	26
	Total	319	32	19	14	384

Table 30: Confusion matrix for not-filled digital surface model (Leiderdorp).

		True condition				
		Ground	**Water**	**Building**	**Vegetation**	**Total**
Predicted	**Surface**	319	32	19	14	384
	Ground	0	0	0	0	0
	Water	0	0	0	0	0
	Building	0	0	0	0	0
	Vegetation	0	0	0	0	0
	Unclassified	0	0	0	0	0
	Total	150	43	27	164	384

Table 31: Confusion matrix for OHN digital surface model (Leiderdorp).

		True condition				
		Ground	**Water**	**Building**	**Vegetation**	**Total**
Predicted	**Surface**	0	0	0	0	0
	Ground	318	15	0	0	332
	Water	1	18	0	0	19
	Building	0	0	18	0	18
	Vegetation	0	0	1	14	15
	Unclassified	0	0	0	0	0
	Total	319	32	19	14	384

Table 32: Confusion matrix for my digital surface model (Leiderdorp).

D.0.7 's-Gravenhage

Digital elevation model

True condition

Predicted		Ground	Water	Total
	Ground	253	1	254
	Water	0	0	0
	Unclassified	121	9	130
	Total	374	10	384

Table 33: Confusion matrix for not-filled digital elevation model ('s-Gravenhage).

True condition

Predicted		Ground	Water	Total
	Ground	257	1	258
	Water	0	0	0
	Unclassified	117	9	126
	Total	375	10	384

Table 34: Confusion matrix for filled digital elevation model ('s-Gravenhage).

True condition

Predicted		Ground	Water	Total
	Ground	374	10	384
	Water	0	0	0
	Unclassified	0	0	0
	Total	374	10	384

Table 35: Confusion matrix for the OHN digital elevation model ('s-Gravenhage).

True condition

Predicted		Ground	Water	Total
	Ground	370	0	370
	Water	4	10	14
	Unclassified	0	0	0
	Total	374	10	384

Table 36: Confusion matrix for my digital elevation model ('s-Gravenhage).

Digital surface model

		True condition				
		Ground	**Water**	**Building**	**Vegetation**	**Total**
Predicted	**Surface**	191	5	97	79	372
	Ground	0	0	0	0	0
	Water	0	0	0	0	0
	Building	0	0	0	0	0
	Vegetation	0	0	0	0	0
	Unclassified	0	8	4	0	12
	Total	191	13	101	79	384

Table 37: Confusion matrix for not-filled digital surface model ('s-Gravenhage).

		True condition				
		Ground	**Water**	**Building**	**Vegetation**	**Total**
Predicted	**Surface**	191	13	101	79	384
	Ground	0	0	0	0	0
	Water	0	0	0	0	0
	Building	0	0	0	0	0
	Vegetation	0	0	0	0	0
	Unclassified	0	0	0	0	0
	Total	191	13	101	79	384

Table 38: Confusion matrix for OHN digital surface model ('s-Gravenhage).

		True condition				
		Ground	**Water**	**Building**	**Vegetation**	**Total**
Predicted	**Surface**	0	0	0	0	0
	Ground	190	3	2	0	195
	Water	1	10	0	0	11
	Building	0	0	92	0	92
	Vegetation	0	0	7	79	86
	Unclassified	0	0	0	0	0
	Total	191	11	101	79	384

Table 39: Confusion matrix for my digital surface model ('s-Gravenhage).

COLOPHON

This document was typeset using LaTeX. The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from André Miede.