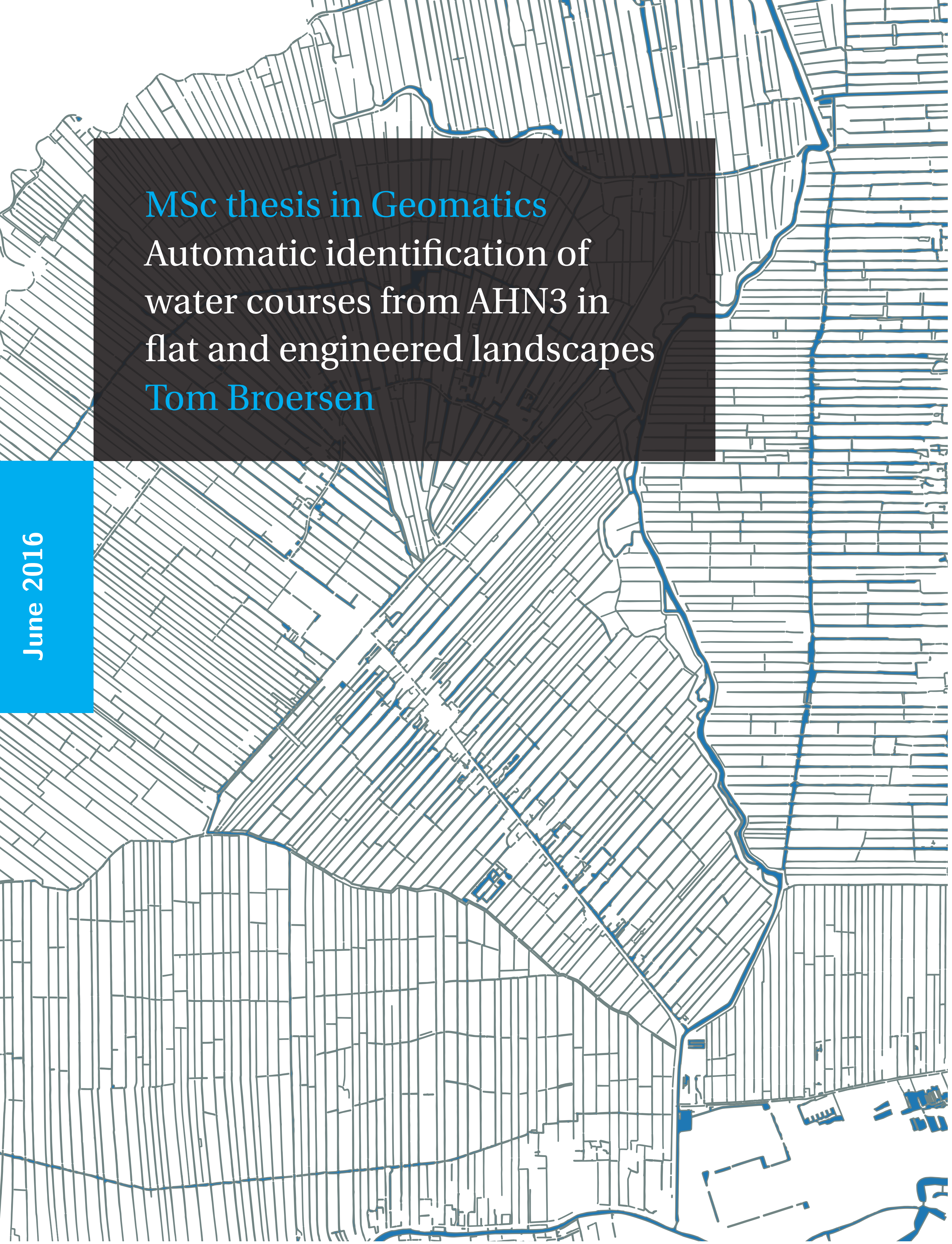


MSc thesis in Geomatics

Automatic identification of water courses from AHN3 in flat and engineered landscapes

Tom Broersen

June 2016

TU Delft
Delft University of Technology

On cover:

A top-down view of a network of water courses nearby the village Zegveld, in the Netherlands. The dataset of water courses was obtained from 'Hoogheemraadschap De Stichtse Rijnlanden' (HDSR).

AUTOMATIC IDENTIFICATION OF WATER COURSES FROM AHN3 IN FLAT AND ENGINEERED LANDSCAPES

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

Tom Broersen

June 2016

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: Dr. Hugo Ledoux
Ravi Ylan Peters
Co-reader: Jinhu Wang

ABSTRACT

The Netherlands is characterised by mostly low lying, flat, and engineered agricultural lands, which are sensitive to flooding. To protect against floods, a correct network characterisation is of the utmost importance. A Dutch water resources management company, Hoogheemraadschap De Stichtse Rijnlanden (HDSR), wishes to have a highly automated method to characterise the water course network. In this thesis, I investigate the possibilities of automatically identifying water courses in flat and engineered landscapes, using the raw points of the Algemeen Hoogtebestand Nederland 3 (AHN3) Light Detection and Ranging (LiDAR) dataset.

I found that there are many methods described in literature which identify channel-like features, and some which identify water courses in engineered landscapes, but none of these are suitable for this application. Thus, I designed a new methodology which is based on two concepts; (1) concave hulls, and (2) the Medial Axis Transform (MAT). The concave hull approach makes use of the presence of water in the water courses, while the MAT uses the concave profiles of the water courses to identify them. A workflow was implemented which uses the raw AHN3 LiDAR point cloud to identify for every water course the polygons of the water surface, and the geographical position of the water surface's center lines. The implemented prototype was used for four different areas to test its applicability to different environments; a clay, peat, urban, and sand area. The water course characteristics in terms of water surface width and surface concavity, differ between these areas. The resulting datasets were validated to obtain mapping and positional accuracies.

The experiments performed in this thesis show the potential of the designed methodology. The concave hull method is very robust to errors in the identification; there are relatively few errors of commission. However, the method does not perform well for high vegetation coverage or low water surface width. It is particularly suited for use in areas where relative water levels are high, water courses are wide, and vegetation coverage is low. The MAT is able to operate well when water levels are low, or even when water courses are dry, and it is relatively insensitive to vegetation coverage. However, it does not perform well for water courses which show little surface curvature, and is prone to errors of commission caused by local non-watercourse convexities and concavities. The combined prototype provides a strong and promising approach for the automatic identification of water courses in flat and engineered landscapes from the raw AHN3 point cloud. When the methods are combined, they manage to identify 98% of all water courses for the clay area, 97% for the peat area, 95% for the urban area, and 76% for the sand area. Clearly, the identification rates profit from the combination of methods. However, the relatively high error of commission of the MAT also radiates into the combined method. The error of commission is then 8% for the clay and peat area, 47% for the urban area, and 17% for the sand area. A number of possible improvements are identified which could elevate the identification rate for the sand area, but specifically lower the presented commission rates. Although the methods currently require a small amount of calibration when applied to new areas, they can in principle be fully automated.

ACKNOWLEDGEMENTS

I would like to express my gratitude to all those people involved in helping me write this thesis, and doing the associated research. Many thanks to my main supervisor Hugo Ledoux, and second supervisor Ravi Peters for their excellent guidance, helpful suggestions, and comments on my work. Furthermore, I would like to thank my co-reader, Jinhu Wang, for his useful comments and suggestions in the final stages of my thesis. Finally, my thanks go to Roger de Crook for providing me with the required data and information from the HDSR, which have allowed me to perform this research and write the thesis.

CONTENTS

ACRONYMS

AHN Algemeen Hoogtebestand Nederland

AHN2 Algemeen Hoogtebestand Nederland 2

AHN3 Algemeen Hoogtebestand Nederland 3

BPA Ball Pivoting Algorithm

CRS Coordinate Reference System

DEM Digital Elevation Model

DEMs Digital Elevation Models

DTM Digital Terrain Model

EPSG European Petroleum Survey Group

GIS Geographical Information System

GRASS Geographic Resources Analysis Support System

HDSR Hoogheemraadschap De Stichtse Rijnlanden

LiDAR Light Detection and Ranging

MAT Medial Axis Transform

NAP Normaal Amsterdams Peil

PDOK Publieke Dienstverlening Op de Kaart

QGIS Quantum GIS

RD Rijksdriehoekstelsel

SAGA System for Automated Geoscientific Analysis

VD Voronoi Diagram

1 INTRODUCTION

The Netherlands is characterised by mostly low lying, flat, and engineered agricultural lands, of which approximately 55% is sensitive to flooding [Parry et al., 2007]. To protect against floods, and to maintain an optimal water balance, a good drainage system is thus of the utmost importance. Artificial drainage networks in agrarian landscapes consist of connected linear features such as channels, culverts, and reshaped gullies [Bailly et al., 2011]. Together these linear features (hereafter referred to as 'water courses') form a network of structures (Figure 1.1 and Figure 1.2) which transit water from the fields into larger canals [Bouldin et al., 2004]. An up-to-date and detailed recognition of the network of water courses is crucial for landscape management issues such as water resources management [Cavalli et al., 2013]. The storage capacity within the network plays an important role in designing drainage channels and pumping stations [Malano and Hofwegen, 1999], and the assessment of network storage capacity is a crucial tool in flood management as it can identify areas which are potentially vulnerable to floods [Cazorzi et al., 2013]. A correct network characterisation containing large-scale and up-to-date positioning and geometry of the water courses is beneficial for the programming of measures, which can guarantee safety from flooding [Cazorzi et al., 2013].

In the Netherlands there are 24 water boards, 'waterschappen' in Dutch, which each are responsible for the water resources management in their designated areas. One of these water boards, the Hoogheemraadschap De Stichtse Rijnlanden (HDSR), uses the SOBEK software suite by Deltares for hydrologic modelling in their working area. A correct and up-to-date characterisation of the water course network in terms of position and dimensions, is beneficial for the correctness of the model output. Current methods employed by the HDSR to characterise the drainage network are based on Algemeen Hoogtebestand Nederland (AHN) Light Detection and Ranging (LiDAR) measurements and stereo-imaging. The AHN dataset is updated every 6 years, while the stereo images are updated yearly. The methods employed by the HDSR are labour-intensive, and subjective[1]. The HDSR wishes to have a highly automated method to characterise the water course network, allowing faster and less subjective updates in the future. A LiDAR dataset, the Algemeen Hoogtebestand Nederland 3 (AHN3)[2], is available for the Netherlands which can be used to develop the new technique. In this thesis, I investigate the possibilities of automatically identifying water courses in flat and engineered landscapes, using the AHN3 dataset.

1 Information obtained through personal communication with Roger de Crook, on January 26th 2016.
2 AHN: www.AHN.nl.

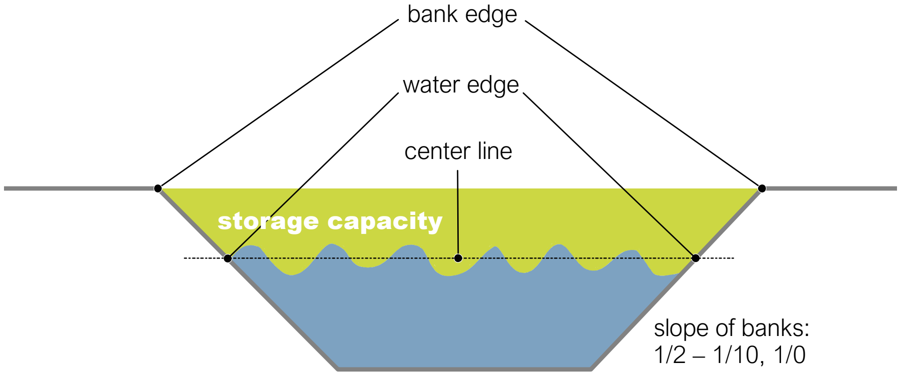


Figure 1.1: An exemplary cross section of a drainage channel. The slopes of the channel banks are typically between 1/2 - 1/10, but can be vertical in some cases, and the slopes of the opposite banks are not always equal (image courtesy of HDSR).

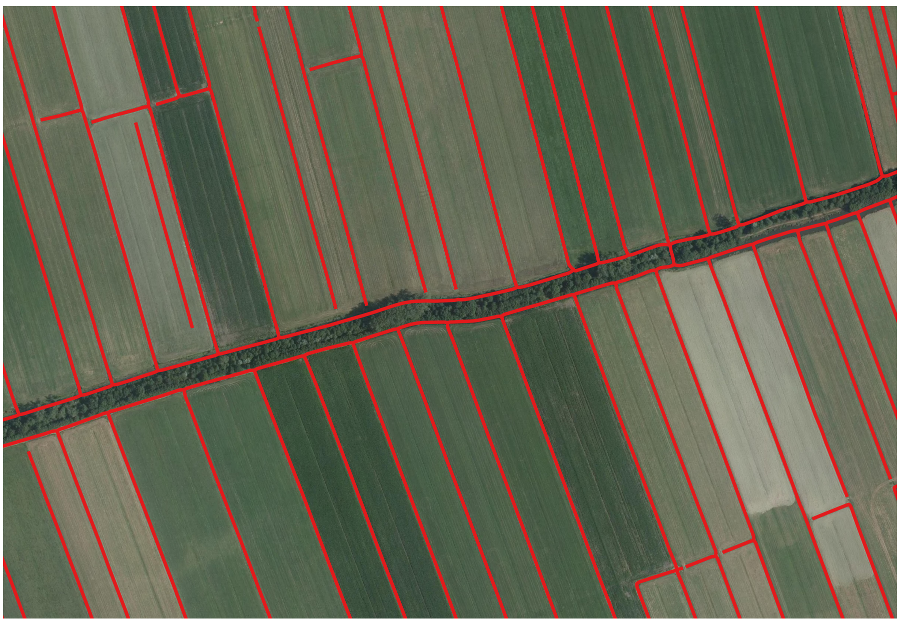


Figure 1.2: Together, water courses form an artificial drainage network (dataset of water courses courtesy of HDSR, background aerial photo courtesy of PDOK).

1.1 THE AHN3 LIDAR DATASET

Manual field observations and photo-interpretations are traditional methods for the delineation of the drainage network. Manual field observations are limited by human resources and money constraints [Gandolfi and Bischetti, 1997], and involve the subjective judgement of the observer [Gandolfi and Bischetti, 1997]. Network detection from aerial images is subject to the same subjectivity. Furthermore, this technique has problems with obscuration and misleading effect of the canopy, the image scale, and the presence of distortions and shadows. Good quality large scale images, and a large amount of work, are needed to correctly delineate the drainage network using aerial images [Gandolfi and Bischetti, 1997]. LiDAR has become an accepted means of acquiring topographic data because of short data acquisition and processing times, relatively high accuracy and point density, and reductions in acquisition costs [Flood and Gutelius, 1997; Hill et al., 2000; Charaniya et al., 2004; van der Zon, 2013]. The width and height of water courses are often larger than the resolution of topographic data provided by LiDAR sources, thus this kind of data has high potential for the use in mapping of water courses [Bailly et al., 2008].

The AHN[3] is a freely available dataset provided by the Dutch government agency 'Rijkswaterstaat', containing detailed and precise elevation measurements covering the entire Netherlands. These elevation measurements were obtained using laser altimetry (LiDAR) from airplane or helicoptre. The first version of the AHN (AHN1) was acquired between 1996 and 2003, and was mainly meant for water system management purposes. The second version of the AHN (Algemeen Hoogtebestand Nederland 2 (AHN2)) was acquired between 2008 and 2013, and is an improvement relative to the original AHN because of higher point density and precision. AHN2 is characterised by a systematic and stochastic error of maximum 5 cm, with an average point density varying between 6-10 points per m^2 [van der Zon, 2013]. The AHN2 point cloud is available in an unfiltered form, and a filtered form only containing the points at field level. The AHN3 is a continuation of the AHN2, retaining the characteristics of the AHN2 with respect to systematic and stohastic error and average point density. During every leafless season of spring (between 2014-2019), a new part of the Netherlands is acquired. Apart from the year of measurement, the AHN3 differs from the AHN2 in terms of the added classification of points. The contractors responsible for acquiring the measurements have classified every point in AHN3 into one of the five classes: vegetation, ground surface, buildings, water, and artificial. Detailed information about the classification process is not specified, but very high quality standards and the use of advanced algorithms is mentioned[4], which are based on surface relief, slope, and the mixed occurrence of high and low reflections. Next to the classification, every point is provided with the additional attributes; scan angle, flightline ID, the return number and number of returns, GPS time, and LiDAR signal intensity. The AHN3 is provided in the compound Coordinate Reference System (CRS) EPSG:7415, which uses Amersfoort / RD New (European Petroleum Survey Group (EPSG):28992) for (x,y) coordinates and Normaal Amsterdams Peil (NAP) height (EPSG:5709) for the (z) values.

3 See footnote 1.
4 http://www.ahn.nl/pagina/het-ahn/inwinning-en-productie.html

1.2 WATER COURSES IN THE FLAT AND ENGINEERED LANDSCAPES AROUND UTRECHT

The working area of the HDSR is situated around the city of Utrecht, the Netherlands. It consists for the most part of flat (elevation typically ranges between -2 m to +10 m) and engineered landscapes, which can be divided into four different dominant environments. Three of these four environments are largely agricultural, and can be classified according to their subsoils; clay, peat, and sand, while urban areas form the fourth environment. While the water courses in these environments differ in some aspects (described in detail in Section 5.2), the environments share several important and defining characteristics. These areas are all very flat, with altitude differences limited to a few m per km^2. But above all, they are all human engineered; they either consist of urban centers, or are used by intensive agricultural practices outside urban centers. Since all of these areas are intensively used (and large parts reside below sea level), drainage is not left to chance; the network is almost entirely artificial. In rural areas, it consists of smaller -often regularly spaced- ditches intersecting the meadows, which eventually drain into larger canals and rivers. In urban centers, sewers represent a large part of the drainage network, and water courses are mostly present in the form of canals. In all these environments, water levels are strictly regulated.

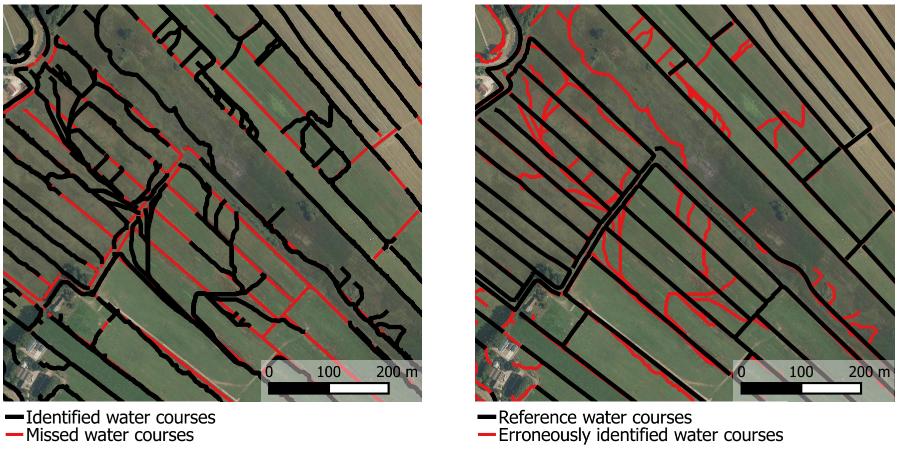
1.3 THE STATE-OF-THE-ART FOR AUTOMATIC IDENTIFICATION OF WATER COURSES

The vast majority of methods developed in interdisciplinary studies have not been applied to the identification of (artificial) water courses in flat, human-engineered areas, however they may still be relevant to the cause of this thesis. Thus, this overview (see Chapter 2 for an extensive literature review) is not limited to the state-of-the-art in extraction of artificial water course networks, but it also considers methods designed for mapping of channel-like features and extraction of hydro break lines[5].

Many methods have been designed to identify channel-like features, of which only the local depression-based method by Liu et al. [2015] may have potential for the identification of artificial water courses in this study. However, their method has never been applied to such an environment, and was only tested with a relatively course 5 m gridded Digital Elevation Model (DEM) for the identification of tidal channels[6]. Höfle et al. [2009] and Toscano et al. [2014] present methods to identify hydro break lines. However, the method by Höfle et al. [2009] requires significant pre-processing [Toscano et al., 2014], and both methods are unable to identify dry water courses. Only a few authors have actually tried to map artificial water course networks in agricultural areas using LiDAR data, and none of these methods were identified as suitable for this study. I tested the GeoNet toolbox [Passalacqua et al., 2010, 2012] for the identification of water courses

5 Hydro break lines can be defined as the edges of water bodies. They indicate the land-water boundary, which is used to generate more accurate gridded Digital Elevation Models (DEMs) [Toscano et al., 2014].

6 Tidal channels are linear depressions embedded in coastal landscapes [Liu et al., 2015], and are formed by the repeated advancement and retreat of the tides [Vandenbruwaene et al., 2012].



(a) Error of omission

(b) Error of commission

Figure 1.3: Identification of water courses by Passalacqua et al. [2012] for an area with peat soil near Utrecht. The dataset is compared to a reference dataset provided by the HDSR (background aerial photo courtesy of PDOK).

to a clay and peat soil area around Utrecht, and obtained poor results for low-relief water courses (Figure 1.3).

What most of these methods have in common, is that they use derivatives of the original LiDAR datasets. They require the generation of a gridded DEM, which is an inherent problem with such datasets since they contain missing data where the water is located. This is due to the absorption of LiDAR signals by water, thus generation of gridded DEMs of these parts is inherently difficult and prone to error. Furthermore, the usage of such gridded DEMs infers a certain decrease of accuracy due to the necessary conversion and interpolation process of the raw LiDAR points [Gold and Edwards, 1992; Fisher, 1997; Brzank et al., 2008]. For this reason, I prefer methods that use the raw LiDAR points. I was unable to identify any suitable method for the identification of wet and dry artificial water courses in the flat and engineered landscapes around Utrecht from raw LiDAR. There is thus the need for the development of a new method.

1.4 THE DISCERNING PROPERTIES OF WATER COURSES AROUND UTRECHT

The artificial water courses around Utrecht have in common that they are human-engineered, and thus mostly have regular shapes. Furthermore, since these are low-elevation areas, water is widely present in these landscapes, and water courses are often -if not permanently- filled with water. Based on the provided description, three characteristics of the water courses in these areas can be provided, which can potentially be used to discern them from the rest of the landscape:

1. *Low elevation in the landscape*: Since the landscapes show very little relief, it is plausible that the water courses are the landscape features with the lowest elevation.

2. *Concave profiles*: Since almost all of the water courses are artificial, especially the regularly spaced ditches in rural areas will have distinct concave profiles.

3. *Presence of water*: Due to the low elevation of the landscape, and constant regulation of water levels, it is likely that most of the water courses always contain water.

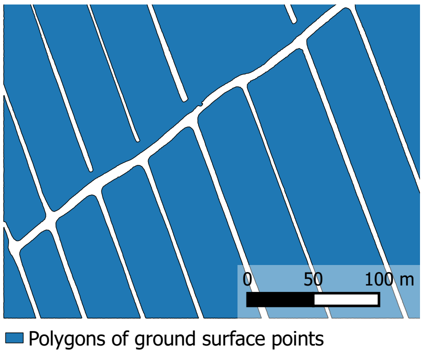
In LiDAR point clouds such as the AHN3, the first two characteristics of water courses are well reflected, since elevation values of point measurements are recorded. Furthermore, the third characteristic is reflected in the AHN3 LiDAR point cloud by voids in the data, since the red laser signal is almost entirely absorbed by water, thus no -or a very weak- reflected signal reaches the sensor. In practice, only the second and thirds characteristic can be effectively used to discern water courses from the rest of the landscape. In these human-engineered landscapes, it is very possible that water levels are artificially kept higher in some parts than in others, thus water courses may not always be the lowest feature in the landscape. Thus, the identification of water courses should be done based on their concave profiles or presence of water.

1.5 MY HYPOTHESIS FOR THE AUTOMATIC IDENTIFICATION OF WATER COURSES IN FLAT AND ENGINEERED LANDSCAPES

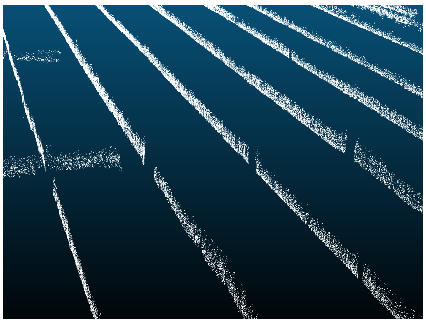
I identified two properties of water courses which can be used to extract them from the AHN3 LiDAR dataset; their presence of water, and concave profiles. Using only one of these properties would negatively affect the identification, since not every water course strongly displays both properties. Obviously, dry water courses cannot be identified by presence of water, and shallow water courses (i.e. with low-curvature or vertical banks) cannot be identified by surface concavity. Thus, I propose to utilize both properties to design a more robust methodology. Two concepts are identified which each utilize one of these properties:

- *Concave hull*: The concave hull is similar to the convex hull, but allows interior angles of the hull to be concave, and therefore to be less than 180° Thus, it can give a good approximation of non-convex distributions of points. LiDAR point clouds of a landscape which contains a network of water courses (and thus a network of voids in the data), can also be seen as a pronounced non-convex distribution of points. Therefore, I suggest that the concave hull can be useful for the identification of water courses, since it can envelope the groups of ground surface points in the landscape, while the water courses remain empty space (see Figure 1.4a).

- *MAT*: The Medial Axis Transform (MAT) is a skeleton-like shape-descriptor that models objects as a union of balls, it essentially gives a lower dimensional representation of an object [Ma et al., 2012]. Approximating the MAT for a typical water course would result in three medial sheets; two inner medial sheets, and one outer medial sheet. The outer medial sheet of a water course forms a 'center plane' (see Figure 1.4b), which can be used to extract the water course's center lines.



(a) Plan view of polygons formed by estimating the concave hull for ground surface points in an AHN3 subset. The water courses (white space) border these polygons.



(b) Perspective view of outer medial sheets approximated by the MAT for the water courses.

Figure 1.4: Examples of the base datasets generated by the concave hull and MAT concepts.

For a more extensive description of these concepts (figures included), the reader is referred to Chapter 3 of this thesis. Both concepts can use raw (albeit classified) LiDAR point clouds as input. The concave hull approach makes use of the presence of water in the water courses, and should thus be able to identify water courses with a sufficiently large water surface, irrespective of their degree of concavity. In contrast, the MAT uses the concave profiles of the water courses to identify them, and can work irrespective of water presence, and should thus additionally be able to identify dry water courses. My hypothesis is that a combination of these concepts can form a robust methodology for the identification of water courses from AHN3 in the flat and engineered landscapes around Utrecht.

1.6 OBJECTIVES & RESEARCH QUESTION

This thesis identifies the possibilities for the characterisation of the water course network in the flat and engineered landscapes around Utrecht, from the AHN3 LiDAR point cloud dataset, thereby striving for full automation. I use two concepts to identify the water courses; the concave hull, and the MAT, the combination of which forms a robust methodology according to my hypothesis. The result of the thesis is a workflow that takes an AHN3 LiDAR point cloud as input, and accordingly identifies for every water course the polygons of the water surface, and the geographical position of the water surface's center line. The workflow thus results in the creation of two datasets: (1) A 2D dataset of water surface polygons, and (2) a 2D dataset of water surface center lines. The methodology is used with different areas to test its applicability to different environments. The resulting datasets are validated to obtain mapping and positional accuracies. The main research question of this thesis is defined as follows:

"To what extent can the position and planimetric geometry of the drainage networks in flat, engineered landscapes be automatically identified from the AHN3 LiDAR point cloud?"

To answer this main research question, the following sub-questions are answered:

1. What are the typical properties of water courses in the flat and engineered landscapes around Utrecht, that make them different from the rest of the landscape?

2. How (well) are these typical properties reflected in the AHN3 dataset?

3. Which methods can use these properties to identify the water courses in the AHN3 dataset?

4. How can the polygons of the water surfaces, and the water course center lines be extracted from the AHN3 using these methods?

5. How well do the extracted datasets compare to reference data?

6. To what extent can the approach be fully automated?

To answer these questions, (1) a literature study is performed to identify the current state-of-the-art in automatic identification of water courses, (2) a methodology is designed and a prototype implementation of the methodology is developed to serve as a proof of concept, and (3) the prototype is tested and validated for different areas to obtain mapping and positional accuracies for water courses identification in different environments.

1.7 SCOPE OF RESEARCH

There is limited time to fulfil the objectives set in this thesis, thus the scope of the current research has to be strictly defined. The following restrictions are specified:

1. By absence of any meta data on point classifications in the AHN3 dataset, it is assumed that the classification is error-free. This thesis specifically does not deal with the classification of points itself.

2. A methodology is designed for the identification of water courses in flat and engineered landscapes, such as the Netherlands. Although the methodology may work well for other landscapes, these are not evaluated in this thesis.

3. The methodology is designed such that it can be (nearly) fully automated, but the prototype does not necessarily reach the same level of automation. The implemented prototype serves as a proof of concept of the proposed methodology only. It is not in any way aimed to be production ready.

4. The methodology and implementation are designed with topology in mind, but the thesis does not deal with topological repair.

1.8 THESIS OUTLINE

The chapters in this thesis are structured as follows:

- In Chapter 2 a literature study is presented which introduces the reader to the current state-of-the-art in identification of water courses.
- In Chapter 3 a theoretical background is given of several concepts which are important for the reader to understand when reading this thesis.
- Chapter 4 presents the methodology developed in this thesis for the identification of water courses in flat and engineered landscapes.
- Chapter 5 subsequently describes how the proposed methodology is implemented, which datasets are applied to the implementation, and how the error metrics are computed. Finally, it resents the experimental results and performance of the implemented prototype for each of the datasets.
- Chapter 6 reflects on the performance and implementation of the methodology, and identifies its strengths and weaknesses.
- Finally, Chapter 7 summarises the most important findings and conclusions, gives an answer to the research questions, and lists recommendations for future work.

2 RELATED WORK

The use of LiDAR mapping to extract water course-like features has been applied to many problems: e.g. for the detection of tidal channels[1] [Lohani and Mason, 2001; Mason et al., 2006; Liu et al., 2015], gullies[2] [Baruch and Filin, 2011], streams [Cho et al., 2011], and coastal structural lines[3] [Brzank et al., 2008]. The vast majority of the methods developed in these interdisciplinary studies have not been applied to the mapping of artificial water course networks in agricultural areas, however they may still be relevant to the cause of this thesis. Furthermore, the identification of hydro break lines can be a promising method, since it essentially provides a dataset of water course outlines. These water course outlines can then be used to create the required datasets of water course polygons and center lines specified in Section 1.6, through center line approximation procedures [Haunert, 2008; Zervakis, 2015]. Thus, this literature study will not be limited to purely the extraction of artificial water course networks, rather it will also consider methods designed for mapping of other water course-like features and the identification of hydro break lines. This chapter provides an overview of relevant methods for the identification of water course-like features in Section 2.1, identification of hydro break lines in Section 2.2, and the identification of artificial water course networks in Section 2.3. The GeoNet toolbox [Passalacqua et al., 2010, 2012], developed for the identification artificial water course networks in flat and engineered landscapes, is applied in Section 2.4 to two separate low-relief areas around Utrecht. Finally, Section 2.5 gives an overview of the most important findings in this literature study.

2.1 IDENTIFICATION OF WATER COURSE-LIKE FEATURES

A wide range of methods exist which try to identify water course-like features. To provide structure, I divided these methods into three categories: (1) flow-routing models, (2) thresholding methods, and (3) geometry and geomorphology-based methods. These categories are loosely based on Liu et al. [2015], and they are not mutually exclusive. All flow-routing and thresholding methods are essentially based on the geometry or geomorphology of landscape features, and would thus fit equally well under 'Geometry and geomorphology-based methods'. I chose to define separate categories for flow-routing and thresholding methods, since they represent distinct ways of working, which can have a significant influence on their applicability to the identification of water courses in artificial water course networks.

1 Tidal channels are linear depressions embedded in coastal landscapes [Liu et al., 2015], and are formed by the repeated advancement and retreat of the tides [Vandenbruwaene et al., 2012].

2 Gullies are depressions embedded within the terrain, formed by the erosive action of water flowing downhill [Baruch and Filin, 2011].

3 Brzank et al. [2008] identify coastal structural lines, which are essentially the edges of tidal channels.

All the methods described in these categories are based on the use of gridded DEMs.

2.1.1 Flow-routing models

Many methods extract water course networks from gridded DEMs using flow accumulation algorithms (e.g. Jenson and Domingue [1988]; Costa-Cabral and Burges [1994]; Lohani and Mason [2001]). Such algorithms assume that slopes along the entire water course are positive, and consider flow-routes to begin near ridges and places with relatively high surface curvature [Baruch and Filin, 2011]. However, these assumptions do not necessarily apply for artificial water course networks in flat landscapes. Studies on the detection of artificial water course networks in agrarian landscapes have shown that flow-routing models are not suitable for anthropogenic networks [Bailly et al., 2008].

2.1.2 Thresholding methods

An often used technique to extract water course-like features, is by setting thresholds values. For example, Meisels et al. [1995] extract channels by identifying pixels with curvature values higher than a threshold. Brzank et al. [2005] select local height minima below an empiric determined threshold, which is then used for region growing. Subsequently, they check whether the mean intensity value of a region is below an empirically determined intensity value, to extract water bodies. Many more authors make use of similar forms of thresholding (e.g. Chorowicz et al. [1992]; Fagherazzi et al. [1999]; Rutzinger et al. [2006]). Usage of such thresholds negatively affects the ability to detect features which display significant form diversity [Baruch and Filin, 2011], and may limit the methods to use low-resolution DEMs [Liu et al., 2015]. Multi-scale thresholds can potentially overcome these limitations. Lohani and Mason [2001] use an adaptive height threshold to locate tidal channels, which enables them to identify tidal channels with geometric properties. However, their approach is still limited to using low-resolution gridded DEMs, where channels are approximately one pixel wide [Baruch and Filin, 2011]. Baruch and Filin [2011] use multi-scale thresholds to detect gullies based on surface curvature, and Cavalli et al. [2013] use a curvature based and slope-dependent threshold to extract a water course network, but both methods are designed for use in strongly textured terrains.

2.1.3 Geometry and geomorphology-based methods

Cho et al. [2011] detect stream channels in very low-relief landscapes, based on local minima and maxima in elevation values from a 1 m gridded DEM, but comment that the method requires significant training and computation. Furthermore, the method may be usable with low-resolution gridded DEMs only [Liu et al., 2015]. This limits its potential for the identification of the water courses around Utrecht, of which some are close to or below 1 m in width. Mason et al. [2006]; Brzank et al. [2008]; Liu et al. [2015] developed methods for the potential extraction of tidal channels, from areas which are similarly flat to the landscapes around Utrecht. Mason et al. [2006] use a method based on edge detection, Brzank et al. [2008] additionally fit a hyperbolic tangent curve[4], while Liu et al. [2015] base their method on enhancing local depressions. Many of the tidal channels are enclosed within raised levees, which are slightly higher than the surrounding flats [Mason et al., 2006]. I assume that the edge detection algorithms works particularly well due to the presence of these levees, which are generally absent around the artificial water courses in Utrecht. Thus, the methods by Mason et al. [2006]; Brzank et al. [2008] may be less suited for the identification of such artificial water courses. Furthermore, Brzank et al. [2008] requires an additional dataset representing the underwater topography. Liu et al. [2015] claim their method to be weakly dependent on scale, robust, and automatic, and suggest that it may have potential for the identification of other types of water course features. However, their method was only tested with a course 5 m resolution gridded DEM.

2.2 IDENTIFICATION OF HYDRO BREAK LINES

Höfle et al. [2009] identify hydro break lines[5] by modeling the locations of laser shot dropouts based on timestamps of the recorded laser measurements, after which potential water regions are detected by using a region growing algorithm. The separation between water and non-water points is then determined by object-based classification. Although this is a promising method for the identification of hydro break lines (and thus for the identification of water course polygons in this thesis) in LiDAR data with laser dropouts, it requires significant pre-processing [Toscano et al., 2014]. Toscano et al. [2014] proposes using LiDAR intensity data as well as an advanced histogram analysis of LiDAR elevation data to automatically detect and delineate water bodies with very little pre-processing. They generate a gridded DEM of the LiDAR elevation data, thereby setting pixel values to 0 if LiDAR signals are missing. Then, the histogram analysis identifies peaks in the elevation data, which are identified as water bodies since these mostly have the same elevation. Histogram analysis will not suffice for smaller water bodies, since these do not generate high enough peaks in the elevation data. These water bodies are identified based on LiDAR intensity data, which is low for water bodies due to high absorption of the LiDAR signal. Both Höfle et al. [2009] and Toscano et al. [2014] are unable to classify dry water courses, since their methods are specifically designed with the properties of the water surfaces in mind.

2.3 IDENTIFICATION OF ARTIFICIAL WATER COURSE NETWORKS

Very few authors have tried to map artificial water course networks in agricultural areas using LiDAR data. Bailly et al. [2008] propose a three-step procedure: (1) estimation of elevation profiles from raw LiDAR points on a set of pre-located sites perpendicular to field plot boundaries, (2) curve

4 The edges of the tidal channels can be extracted from the hyperbolic tangent curve using either maximum slope, or maximum curvature, depending on the shape of the water course banks [Brzank et al., 2008].

5 Hydro break lines is synonymous to the edge of a water body, they thus give the distinction between water and non-water areas, which can be used for the generation of gridded DEMs from LiDAR points [Toscano et al., 2014].

shape analysis to derive concavity indicators from the elevation profiles, and (3) a boolean classification procedure that discriminates ditches from 'non-ditches' on the basis of the concavity indicators. They achieved ditch omissions of around 50%, and ditch commissions of around 15%. They attribute the poor performance of the method for a major part to insufficient density of LiDAR points, and vegetation coverage of the ditches at LiDAR survey time. Their LiDAR point density averaged 10 points per m^2, which is comparable to the point density of the AHN3. Cazorzi et al. [2013] identifies two important limitations of Bailly et al. [2008]; (1) ditches can only be identified if they are located exclusively at field boundaries, and (2) a geographic database of plot boundaries must be available. I add to these limitations, that the method of Bailly et al. [2008] may not work well for water courses which are largely filled with water, since this will hinder the estimation of elevation profiles and subsequent derivation of concavity indicators.

Passalacqua et al. [2010] designed a methodology which involves a preprocessing step to remove noise and enhance features that are critical to the network extraction. Water courses can typically be characterised by positive curvature, and by high values of flow accumulation. Therefore, the subsequent network extraction is based on these properties, which distinguish water courses from the rest of the landscape. Passalacqua et al. [2010] define the concept that, if many possible curves are created connecting point a to point b, then the curve with the largest overall positive curvature and flow accumulation would be the actual water course. This concept is mathematically expressed in a cost function, which is used to extract the water course network. Passalacqua et al. [2012] extended Passalacqua et al. [2010] for the use in flat and engineered landscapes, by using Laplacian instead of geometric curvature to more effectively distinguish water courses in these environments. They successfully extracted the water course network using a 3 m gridded DEM for the low-relief human-impacted landscape of Le Sueur River Basin, Minnesota. However, their study area has elevation differences of up to 60 m, and therefore seems to be less flat than the area around Utrecht used in this study. Furthermore, the use of surface curvature to distinguish water courses may provide insufficient results if these water courses display little surface curvature, and the use of flow accumulation may not be suitable for anthropogenic networks [Bailly et al., 2008]. However, the combination of these distinguishing properties, and the enhancement of features during pre-processing, may mitigate these downsides to some extent. Passalacqua et al. [2010] have released the presented methodology to the community as a toolbox called GeoNet, which is freely available for download[6], and Passalacqua et al. [2012] extended GeoNet with their solution for flat and engineered landscapes. In Section 2.4 I test the applicability of the standard [Passalacqua et al., 2010] and extended [Passalacqua et al., 2012] GeoNet package for the (very) low-relief landscape around Utrecht.

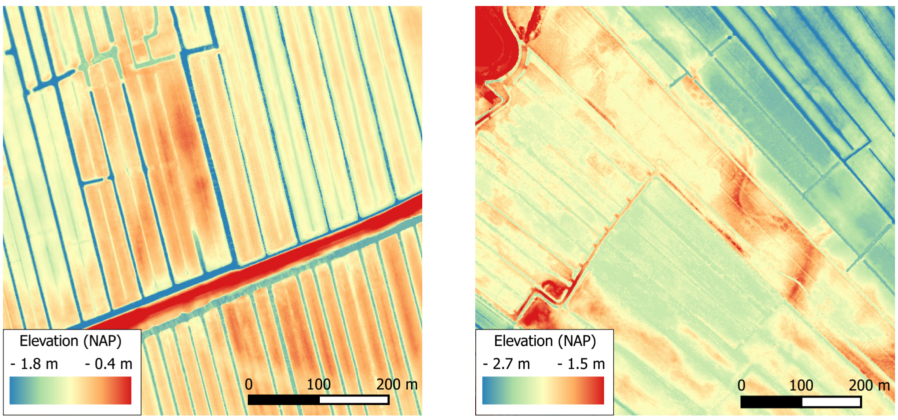
Cazorzi et al. [2013] propose a methodology based on the extraction of local small-scale low-relief features from a 1 m gridded Digital Terrain Model (DTM), by elimination of the large-scale landscape forms from the data. A filter is applied to the DTM, which results in a smoothed elevation model that approximates the large-scale landscape forms. Then, by subtracting this smoothed model from the DTM, an approximation of the local relief is obtained. The water course network is then detected by labeling of peak values in the local relief map through the use of a threshold value, which is taken as the standard deviation of the local relief. Their results proved to

6 GeoNet: https://sites.google.com/site/geonethome/source-code

be more reliable than their outdated cartography-based reference data, and a median distance of reference points to the extracted water course network was registered to be about 1 m. The usage of a threshold on the local relief, can have implications on the ability of the method to identify water courses of different forms (Section 2.1.2), but this could be mitigated by using multiple or adaptive thresholds. Cazorzi et al. [2013] tested their method for an area with elevation differences of maximum 30 m within 1km², which can be considered low-relief, but is somewhat more textured than the area around Utrecht used in this study. Furthermore, I suspect that the use of local relief is less suitable for low-relief water courses, which are especially present in some of the water-rich areas around Utrecht.

2.4 GEONET TESTED FOR UTRECHT

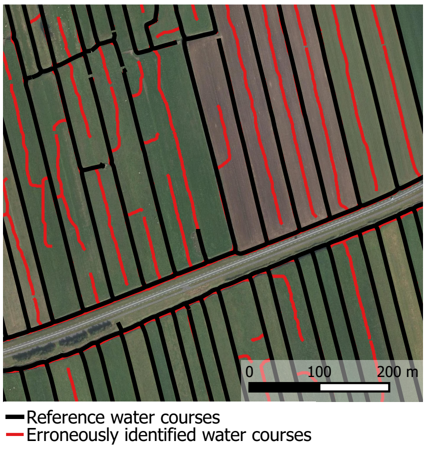
Since GeoNet was shown to work well in flat and engineered landscapes [Passalacqua et al., 2012], and is freely available for download with instructions provided, I tested the standard [Passalacqua et al., 2010] and extended [Passalacqua et al., 2012] GeoNet package for the (very) low-relief landscape around Utrecht. The methods were applied to two separate areas; one area with clay soil, and the other with peat soil. The area with clay soil has water courses which are clearly distinguishable from the surrounding meadows by elevation (Figure 2.1a), while the area with peat soil has much less distinguishable water courses (Figure 2.1b). The gridded DEMs for both areas were constructed using LAStools (see Section 5.1.1), using the AHN3 LiDAR point cloud as source, from which the vegetation, building, and artificial classes were filtered out. For the clay soil area, the GeoNet methods were performed using the standard values for the user-defined parameters. For the area with peat soil, the standard and extended GeoNet methods were both performed with 'flowThresholdForSkeleton' set to 300. Additionally, for the extended GeoNet method, 'nFilterIterations' was set to 10. The results of the methods were compared to a reference dataset of water course center lines (see Section 5.2) supplied by the HDSR.



(a) Area with clay soil (b) Area with peat soil

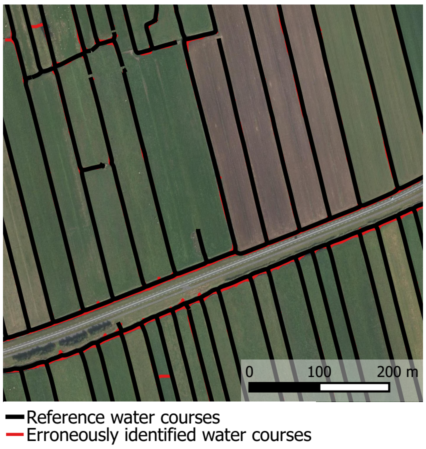
Figure 2.1: DEMs generated for two areas near Utrecht, using LAStools, from AHN3 source. Vegetation, building, and artificial classes were filtered out. Elevation is given in meters relative to the NAP.

(a) Error of omission for standard GeoNet.



(b) Error of commission for standard Geonet.

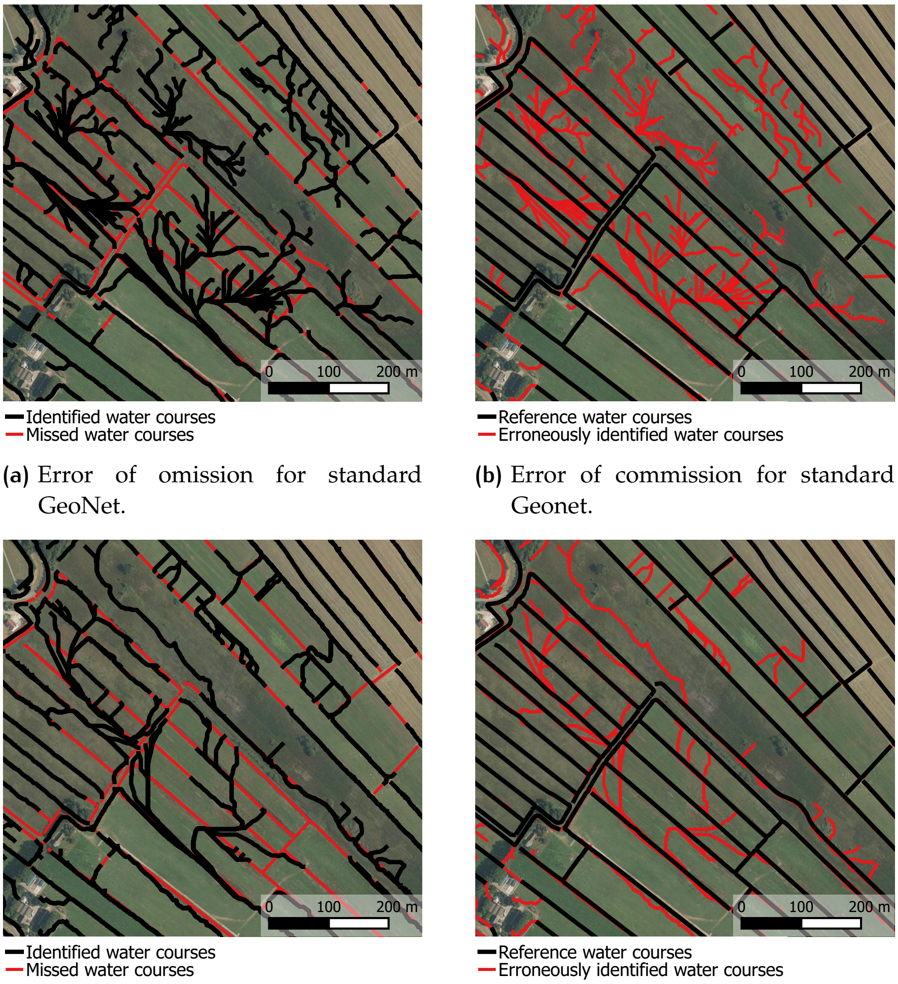
(c) Error of omission for extended GeoNet.



(d) Error of commission for extended GeoNet.

Figure 2.2: Water courses extracted by standard [Passalacqua et al., 2010] and extended [Passalacqua et al., 2012] GeoNet, for the area with clay soil (Figure 2.1a). The dataset is compared to a reference dataset provided by the HDSR (background aerial photo courtesy of PDOK).

RESULTS For the clay area, standard GeoNet performed poorly, displaying many errors of omission (Figure 2.2a) commission (Figure 2.2b), but extended GeoNet performed well (Figure 2.2c and Figure 2.2d). For the peat area, both methods performed poorly, showing extensive errors of omission (Figure 2.3a and Figure 2.3c) and commission (Figure 2.3b and Figure 2.3d). As expected, the GeoNet method extended for flat and engineered landscapes overall performed better than standard GeoNet, but still had major difficulties identifying the water courses in places with very low relief (Figure 2.3 and Figure 2.1b). Furthermore, both GeoNet methods lack the accuracy needed for the current research, since the lines drawn for the water courses often do not correspond to the actual centre lines of said water courses. The illustrated deficiencies indicate that the GeoNet package is not well suited for the identification of water courses in regions such as Utrecht, which are characterised by very low relief and water courses which are hard to distinguish from their surrounding meadows based on elevation.



(a) Error of omission for standard GeoNet.

(b) Error of commission for standard Geonet.

(c) Error of omission for extended GeoNet.

(d) Error of commission for extended GeoNet.

Figure 2.3: Water courses extracted by standard [Passalacqua et al., 2010] and extended [Passalacqua et al., 2012] GeoNet, for the area with peat soil (Figure 2.1b). The dataset is compared to a reference dataset provided by the HDSR (background aerial photo courtesy of PDOK).

2.5 SUMMARY

Many methods have been designed to identify water course-like features (Section 2.1), which I have classified into three categories: (1) flow-routing models, (2) thresholding methods, and (3) geometry and geomorphology-based methods. Flow-routing models may not be suitable for anthropogenic networks [Bailly et al., 2008], and thresholding methods can have difficulties detecting features which display significant form diversity [Baruch and Filin, 2011], and may be limited to use low-resolution DEMs [Liu et al., 2015]. The use of multi-scale thresholds can potentially overcome these limitations, but no suitable multi-scale thresholding methods were identified for the use in the flat and engineered landscapes around Utrecht. The geometry and geomorphology-based methods presents a broad category, of which only the local depression-based method by Liu et al. [2015] may have potential for the identification of artificial water course channels in this study. However, their method has never been applied to such an environment, and was only tested with a relatively course 5 m gridded DEM for the identification of tidal channels. Höfle et al. [2009] and Toscano et al. [2014] present methods to identify hydro break lines (Section 2.2), from the water course center lines could be extracted. However, the method by Höfle et al. [2009] requires significant pre-processing [Toscano et al., 2014], and both methods are unable to identify dry water courses. Only a few authors have actually tried to map artificial water course networks in agricultural areas using LiDAR data (Section 2.3), and none of these methods were identified as suitable for this study. I tested the GeoNet toolbox [Passalacqua et al., 2010, 2012] for the identification of water courses in a clay and peat soil area around Utrecht, and obtained poor results for low-relief water courses.

What most of the identified methods have in common is that they use derivatives of the original LiDAR datasets. They require the generation of a gridded DEM, which is an inherent problem with such datasets since they contain missing data where the water is located. This is due to the absorption of LiDAR signals by water, thus generation of gridded DEMs of these parts is inherently difficult and prone to error. Furthermore, the usage of such gridded DEMs infers a certain decrease of accuracy due to the necessary conversion and interpolation process of the raw LiDAR points [Gold and Edwards, 1992; Fisher, 1997; Brzank et al., 2008]. For this reason, I prefer methods that use the raw LiDAR points. I was unable to identify any suitable method for the identification of wet and dry artificial water courses in the flat and engineered landscapes around Utrecht from raw LiDAR. Thus, there is a need for the development of a new method.

3 THEORETICAL BACKGROUND

This chapter provides an overview of the relevant theory for this thesis. Section 3.1 describes the concept of the Voronoi Diagram (VD), and how they can be used to identify polygon centre lines. In Section 3.2, an introduction is given of convex and concave hulls. Furthermore, it gives an example of the usage of the concave hull for a point cloud with water courses. Lastly, Section 3.3 presents the MAT. It shows an example of approximating the medial axis of water courses, explains how the medial axis can be segmented into medial sheets using the medial bisector, and introduces the Ball Pivoting Algorithm (BPA) for the triangulation of medial sheets.

3.1 VORONOI DIAGRAMS FOR POLYGON CENTRE LINES

A VD [Aurenhammer, 1991] partitions a plane into convex cells (see Figure 3.1), which is done based on distance to a set of generating points. The partitioning is performed such that each cell contains exactly one generating point, and all other points in the cell are closer to this generating point than to any other. Every edge of a Voronoi cell forms the boundary between two adjacent cells.

VDs can be used to approximate the centre line of a polygon, by using the points on the polygon boundary as generating points. A line-in-polygon test can then select the VD's edges which are completely inside the polygons[1], these edges form the polygon centre line (see Figure 3.2). The degree to which the centre line is correctly identified depends on the density of the generating points. If this density is high enough (see Figure 3.2b), then the method will approximate the polygon's medial axis, which is similar to the polygon's centre line [Gold, 1999; Haunert, 2008].

3.2 THE CONCAVE HULL OF A SET OF POINTS

The convex hull of a set of points in the Euclidean plane is defined as the smallest convex set that contains the points. For example, if P is a subset of points in the plane, then the convex hull can be visualised as the shape obtained when stretching a rubber band around P [de Berg et al., 2000]. However, if the set of points has a pronounced non-convex distribution, then the convex hull of these points will not provide a good characterisation of this distribution of points (see example in Figure 3.3b) [Duckham et al., 2008].

1 Method obtained through personal communication with Martijn Meijers (TU Delft) on Monday, February 15, 2016.

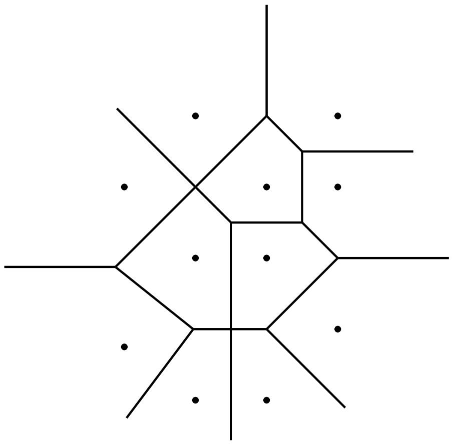
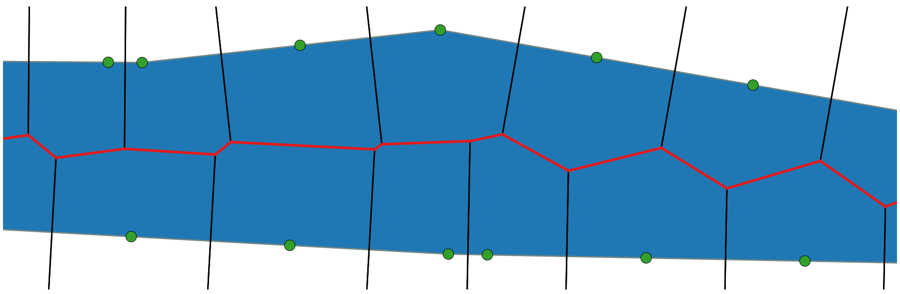
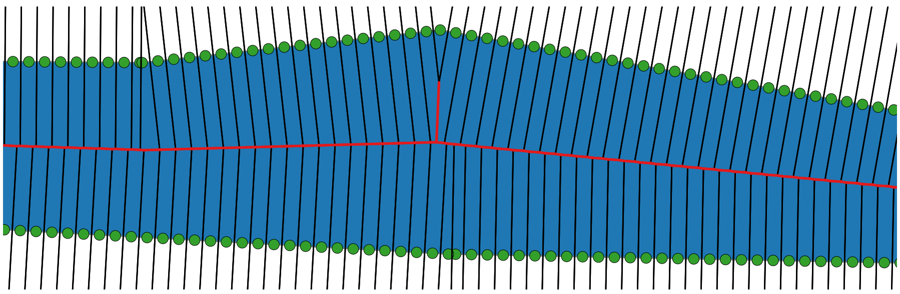


Figure 3.1: A VD [Aurenhammer, 1991] created on a set of generating points.

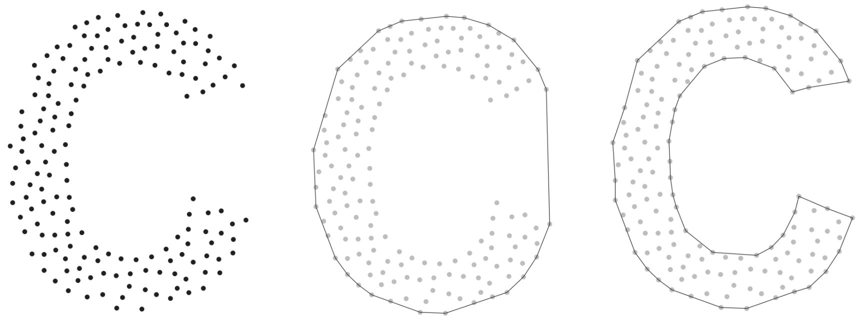


(a) A sparse distribution of generating points leads to a poorly approximated centre line.



(b) A dense distribution of generating points leads to a well approximated centre line.

Figure 3.2: Approximation of a water course (blue polygon) centre line (red) using a VD [Aurenhammer, 1991] created on the generating points (green) on the water polygon boundary, for two different generating point densities.



(a) A non-convex set of points.

(b) The convex hull of the points in (a).

(c) The concave hull of the points in (a).

Figure 3.3: The concepts of the convex hull and concave hull for a set of points (image courtesy of Duckham et al. [2008]).

Conversely to the convex hull, the concave hull allows interior angles of the hull to be concave, and therefore to be less than 180°. The concave hull can therefore minimize the area of the containing shape, giving a better approximation of the non-convex distribution of points (as shown in Figure 3.3c). The concept of the concave hull was introduced in Galton and Duckham [2006] as 'non-convex footprints', and developed in Moreira and Santos [2007] as concave hulls. Similar solutions to this problem were developed by Edelsbrunner et al. [1983] in the form of alpha-shapes, and by Duckham et al. [2008] in the form of characteristic shapes. While there is only one convex hull for every set of points, there can be many different concave hulls, the shape of which depends on the threshold distance between two points that is regarded as 'open' space [Moreira and Santos, 2007; Duckham et al., 2008]. There is no 'correct' or 'best' characteristic shape, this depends on the final application [Moreira and Santos, 2007; Duckham et al., 2008].

3.2.1 Algorithm for computing the concave hull

It is possible to compute the concave hull of a set of points in the (x,y) plane by computing a TIN of all the points based on their (x,y) values, and subsequently removing all triangles whose edge-length is above a chosen 'concavity' value, by order of the longest edge length[2]. This effectively grows the convex hull inwards, and results in a concave hull of the set of points.

3.2.2 The concave hull of a point cloud with water courses

A LiDAR point cloud of a landscape which contains a network of water courses (and thus a network of voids in the data), can also be seen as a pronounced non-convex distribution of points. Therefore, to obtain a good characterisation of the distribution of this set of points, the concave hull is to be preferred above the convex hull. By specifying a sufficiently low threshold distance (concavity value), the concave hull can envelope the groups of ground surface point in the landscape, while the water courses remain

2 This algorithm description for the computation of the concave hull was obtained through personal communication with Martin Isenburg (rapidlasso) on June 1st 2016.

empty space[3] (see Figure 3.4). This can be useful for the identification of water courses.

3.3 MEDIAL AXIS TRANSFORM (MAT)

The MAT is a skeleton-like shape-descriptor that models objects as a union of balls. The MAT is formally defined as the set of maximal balls that are tangent to the surface of a shape at two or more points. The centres of these maximal balls, also called medial balls, form a medial skeletal structure of the object [Peters et al., 2015]. The medial axis thus gives a lower dimensional representation of an object [Ma et al., 2012].

3.3.1 Approximating the MAT

Ma et al. [2012] proposed a method to produce approximate medial axis points given a set of surface sample points and their corresponding normal vectors. The method works as follows (and is visualised in Figure 3.5) [Peters et al., 2015]:

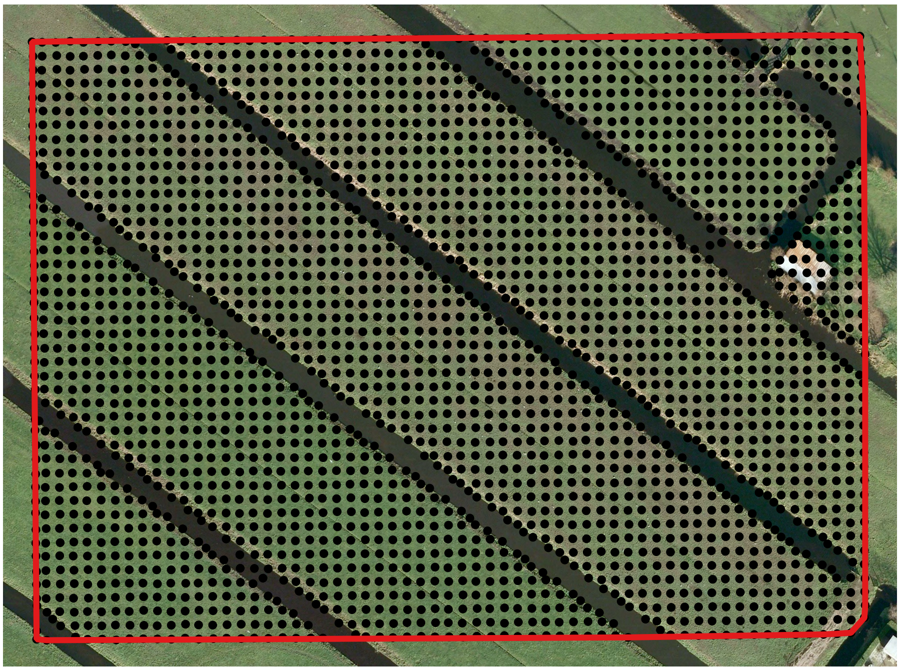
- Medial balls are found for each point p in the point cloud by iteratively shrinking a very large ball that is centred along the point's normal n.
- At each of these iterations, a point q is found that is nearest to the ball's centre. At the next iteration, the ball is accordingly shrunk such that it touches both p and q, and remains centred along n.
- The iteration continues until the ball's interior is empty, and there are no closer points to its centre than p and q.

This algorithm results in the creation of two products: the *interior* and *exterior* MAT. The interior MAT is obtained when the normals point outward (such as in Figure 3.5). The exterior MAT is obtained by flipping the normals, essentially resulting in the complement of the space that is occupied by the interior MAT [Peters et al., 2015]. In three-space, the medial axis is represented by a set of medial sheets; these are manifolds with boundaries that meet along Y-intersection curves. Together, these medial sheets form a skeleton-like structure (see [Siddiqi and Pizer, 2008] and [Peters and Ledoux, 2016]).

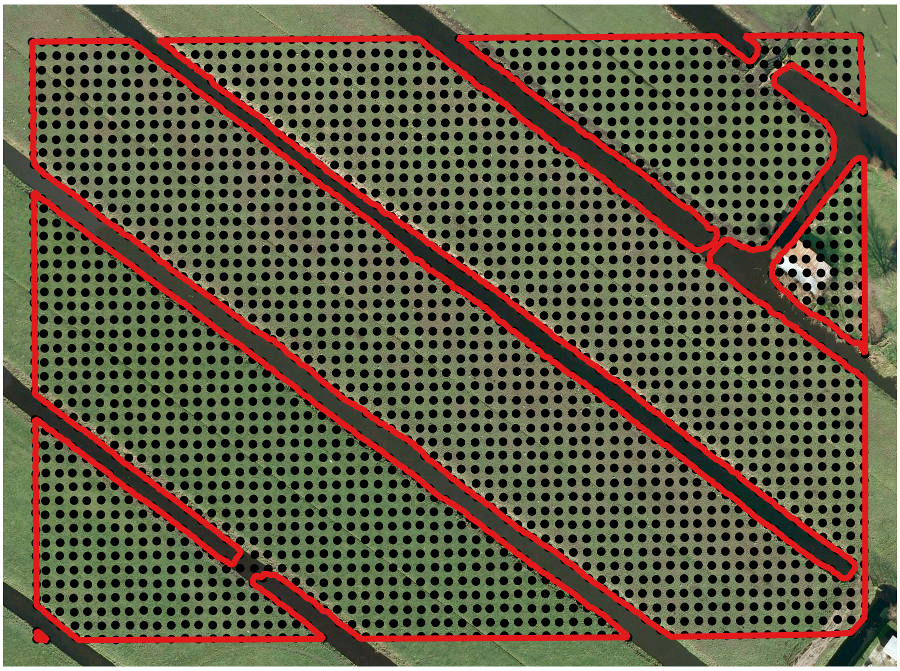
NOISE HEURISTICS Peters et al. [2015] improved the performance of the shrinking ball algorithm by Ma et al. [2012] for LiDAR point clouds which contain significant noise. The algorithm was extended with heuristics that can prematurely stop the shrinking of the medial balls. These heuristics are based on the progression of the separation angle, which is the angle pcq, where c is the ball's centre. Two heuristics were proposed [Peters and Ledoux, 2016]:

1. *Stable ball preservation*: Whenever the separation angle drops below a threshold t_{preserve}, the ball-shrinking process is stopped and the medial ball for p is selected as the last ball which did not violate the threshold. This heuristic prevents small-scale surface roughness from polluting the medial sheets.

3 Note that this method only works if any points classified as water are first removed from the point cloud. This also means that a solid classification of the points needs to be available.



(a) The convex hull.



(b) The concave hull.

Figure 3.4: The convex hull and concave hull computed for a subset of AHN3 points. Displayed is a 3 m uniformly spaced subset of AHN3 points, but the convex hull and concave hull were computed for the full subset of points (background aerial photo courtesy of Google Maps).

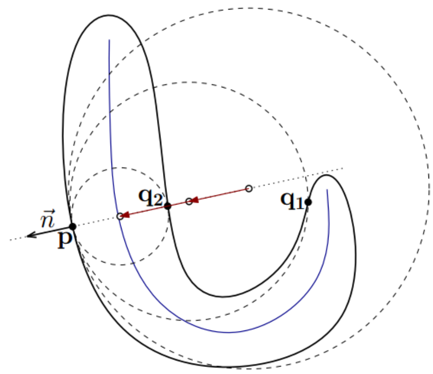


Figure 3.5: The shrinking ball algorithm (Image courtesy of [Peters et al., 2015]).

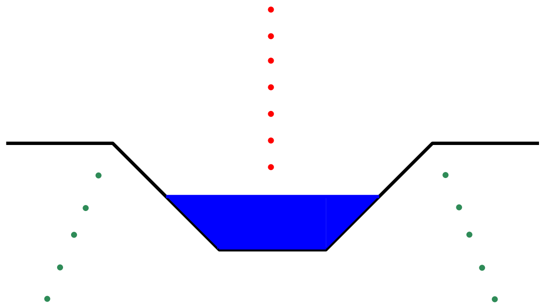


Figure 3.6: Cross section of a water course. The two inner (green points) and one outer medial sheets (red points) are displayed of a medial axis approximated for a set of points on the water course banks.

2. *Plane detection*: When the separation angle of the initial medial ball is lower than a threshold t_{planar}, no medial ball is assigned to the point p. This heuristic prevents small-scale surface roughness of planar features from polluting the dataset of medial sheets.

These heuristics exploit the information that is captured in the sequence of medial balls of a surface point, and provide a more robust way of approximating the medial balls.

APPROXIMATING THE MAT FOR WATER COURSES Approximating the MAT for a typical water course would result in three medial axes (see Figure 3.6); two inner medial axes, and one outer medial axis. This is visualized in 3D in Figure 3.7, which shows the inner and outer medial axes approximated for a subset of the ground surface points in the AHN3 LiDAR point cloud. The outer medial axis of water courses forms a 'centre plane', which essentially describes the location of the channel centre line depending on the height of the water surface. The lower points in this outer medial axis describe the centre line of the water surface at the time of point cloud measurement.

3.3.2 Segmentation into medial sheets using the medial bisector

Approximating the outer medial axis for the water courses in a subset of the AHN3 dataset, results in a dataset without a distinction between separate water courses (such as in Figure 3.7). However, for further processing of the

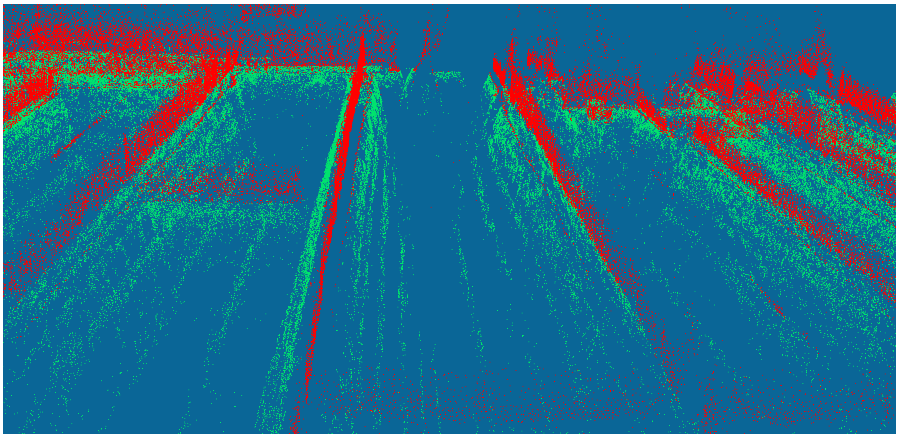


Figure 3.7: Perspective view of inner (green points) and outer (red points) medial sheets of a medial axis approximated for a subset of ground surface points in the AHN3 dataset.

approximated medial axis, it is beneficial that it is segmented into separate medial sheets as much as possible.

The medial bisector can be used to segment the medial axis into separate medial sheets. Any medial ball generated by the shrinking ball algorithm has two points where the ball touches the ground surface, and the vectors from the medial ball centre to these surface points are called spoke vectors. The bisector of these two spoke vectors is called the medial bisector, which is by definition tangent to the corresponding medial sheet, and points in the direction of decreasing ball radius on the medial sheet (see Figure 3.8). Different medial sheets will have different bisector orientations, while these orientations are very similar within sheets, thus the medial bisector can be used to successfully differentiate between the medial sheets[4].

3.3.3 Triangulation of medial sheets using the Ball Pivoting Algorithm

The BPA can triangulate a set of points using the following principle (described in [Bernardini et al., 1999]); If a manifold M is the surface of a three-dimensional object, and S is a subset of M, then the algorithm starts by selecting an initial seed triangle. Subsequently, a ball with radius p is placed in contact with the three vertices of the seed triangle. Then, while keeping the ball in contact with two of these vertices, the ball is pivoted until it touches another point in the subset. This pivoting operation is performed around each edge of the mesh boundary. Every time the pivoting ball contacts three vertices, a new triangle is added to the mesh.

4 The method of segmenting the medial axis into medial sheets by using the medial bisector was obtained through personal communication with Ravi Peters (TU Delft) on February 15th 2016 .

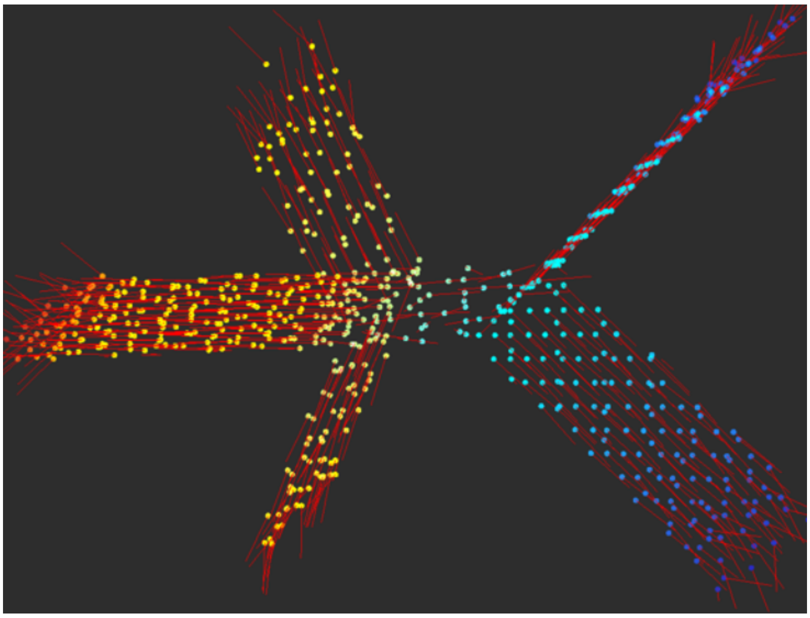


Figure 3.8: Shown are the medial bisectors corresponding to the medial points of a medial axis, which are tangent to the medial sheet to which these medial points belong. The medial bisectors point in the direction of decreasing ball radius on the medial sheet (image courtesy of Ravi Peters, TU Delft).

4 A NEW METHODOLOGY FOR THE AUTOMATIC IDENTIFICATION OF WATER COURSES

The literature review in Chapter 2 did not identify any suitable methods for the automatic identification of artificial water courses from the raw AHN3 points for flat and engineered landscapes. Thus, I introduce a new methodology in this chapter, which is based on two complementary concepts: the concave hull (see Section 3.2), and the MAT (see Section 3.3). This chapter thoroughly explains the methodology which I designed around the concave hull concept in Section 4.1, and the MAT in Section 4.2. Lasty, Section 4.3 describes how the concave hull and MAT methods are combined to form one single, robust, methodology which identifies the water course polygons and centre lines.

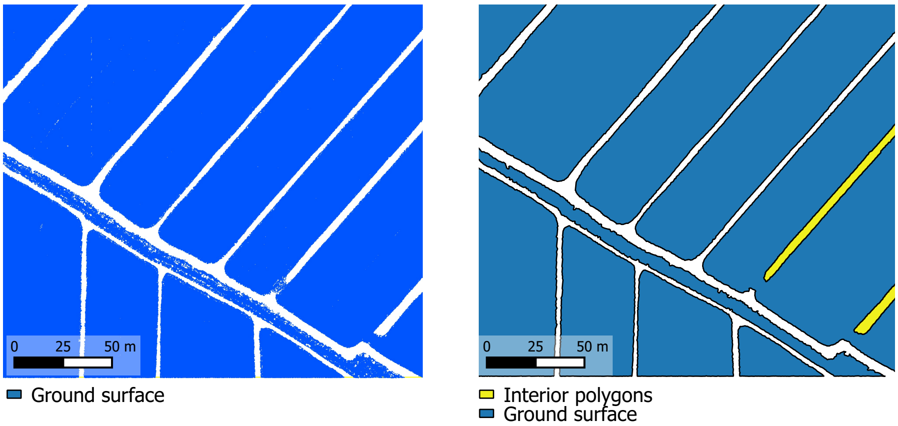
4.1 THE CONCAVE HULL METHOD

The concave hull method takes advantage of a key property of AHN3 measurements (and of any other red laser-based LiDAR dataset) above open water bodies; the LiDAR signal emitted above water bodies is almost completely absorbed, only LiDAR signals emitted at or near nadir are reflected strong enough to be detected by the sensor. Since the points in the AHN3 dataset are classified into five classes (see the description of AHN3 in Section 1.1), among which a water class, the few LiDAR measurements which did reflect on the water bodies can be filtered out. What remains is a dataset with separate disconnected groups of ground surface points, with voids in between these groups which represent the water bodies, from which *visual* detection of the water courses is relatively easy (see Figure 4.1a). These groups of ground surface points can be captured in polygons by generating their concave hulls (as described in Section 3.2), which results in multiple disconnected ground surface polygons (shown in Figure 4.1b). The majority of water courses are represented by the space between these ground surface polygons, while a minority is represented by interior polygons. From such datasets, automatic derivation of the water courses is possible.

This principle forms the basis of what I refer to as the 'concave hull method'. The conceptual workflows in Figure 4.2 and Figure 4.8 list the subsequent procedures required to identify the water course polygons and centre lines from the AHN3 dataset using this method. By hand of these workflows, Section 4.1.1 explains the procedures in detail for the identification of the water course polygons, and Section 4.1.2 explains the procedures required to generate the water course centre lines.

4.1.1 Identification of water course polygons

To identify the water course polygons, two datasets are generated; (1) the concave hulls of the ground surface points, and (2) the concave hulls of the



(a) Subset of AHN3 ground surface points.

(b) Polygonal dataset of the concave hulls of the groups of ground surface points.

Figure 4.1: This figure shows the base principle of the concave hull method. The concave hulls are computed for the groups of ground surface points in the AHN3 dataset. The majority of the water courses are represented by the space between these concave hulls, while some are interior polygons of the concave hulls.

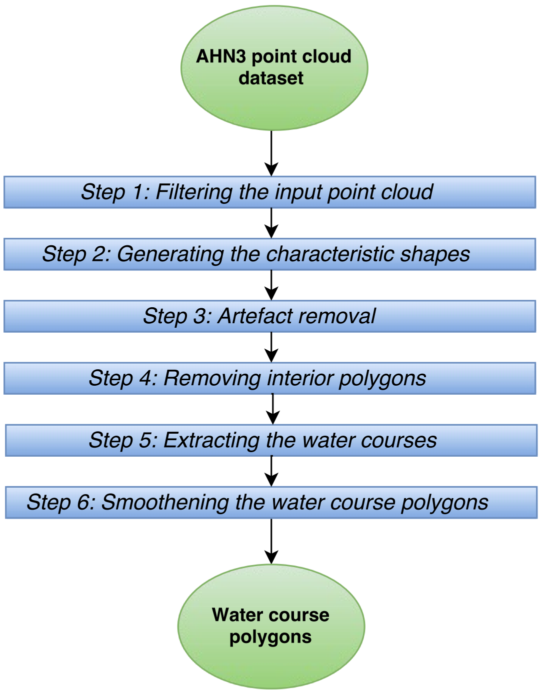


Figure 4.2: Conceptual workflow for the identification of the water course polygons using the concave hull method.

vegetation points. The first dataset is used to identify the water courses, while the second is used to remove artefacts from the first. The required procedures (see workflow in Figure 4.2) are elaborated upon in the following paragraphs.

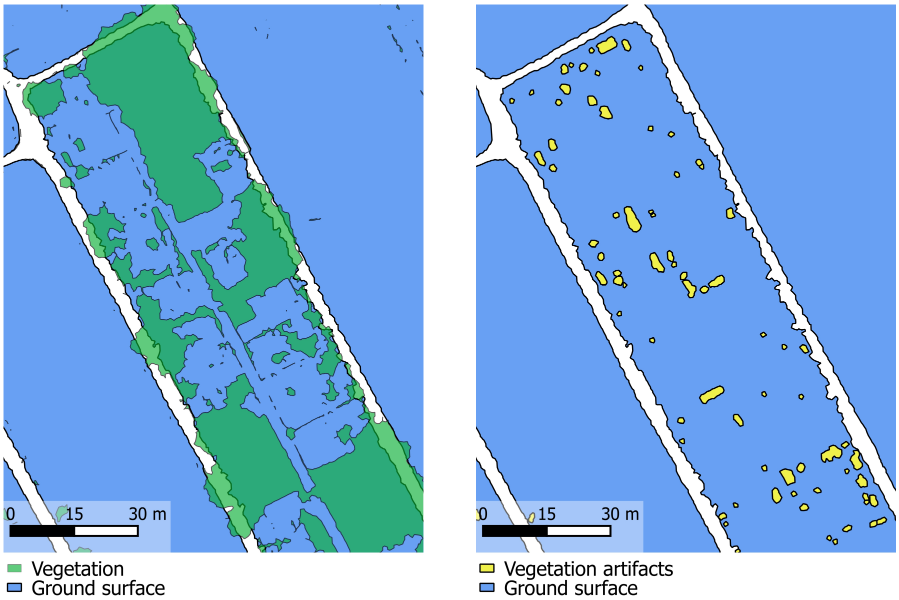
STEP 1: FILTERING THE POINT CLOUD To compute a concave hull of the ground surface points, it is important that some of the non-ground surface points are removed from the dataset. Naturally, all points classified as water should be removed, but this is also true for the points classified as vegetation and artificial objects. Vegetation can occlude parts of the water courses, either by growing on the water surface, or by overhanging canopies from vegetation on the channel banks. If these points are not removed from the dataset, then this influences the generated concave hulls. Furthermore, artificial objects such as bridges can also occlude parts of the water courses, and cause similar problems. Thus, in this first step, all points classified as water, vegetation, and artificial objects are removed from the dataset. Furthermore, in a parallel filtering operation, all points except the vegetation points are removed from the dataset. At the end of this first step, two datasets are formed: (1) a dataset of ground surface and building points, and (2) a dataset of vegetation points.

STEP 2: GENERATING THE CONCAVE HULLS Concave hulls are generated for both filtered point clouds. This results in two separate datasets[1]; (1) the concave hulls of the clusters of ground surface points (such as in Figure 4.1b), and (2) the concave hulls of the clusters of vegetation points. Water courses which are contained in the ground surface point clusters (e.g. they are not part of the connected network of water courses) are represented in the output dataset of concave hulls as interior polygons (see Figure 4.1b), which is done similarly for clearings in patches of vegetation.

STEP 3: ARTEFACT REMOVAL By removing vegetation from the AHN3 point cloud in step 1 of the workflow, the voids otherwise occluded by these vegetation points are exposed, thus easing the identification of the water courses. Unfortunately, there is a trade-off in removing the vegetation points. In most cases, the red laser used for the AHN3 acquisition partially penetrates through the vegetation canopy, and are either absorbed by the water surface or reflected by the ground surface below the canopy. However, in some cases the vegetation canopy is so dense that the laser signal cannot penetrate through, thus any ground surface below the canopy is not detected. If these vegetation points are removed from the AHN3 point cloud, then voids in the data remain, and if large enough these voids are erroneously identified as water bodies (see the artefacts in Figure 4.3b). This is undesirable, since the concave hull method is based on the assumption that all voids in the point cloud are water courses, which they are not in this case.

To remove these artefacts, the previously generated dataset with the concave hulls of the vegetation point clusters is used. This removal procedure is based on the following assumption: if the artefact is indeed caused by the removal of vegetation points, then this artefact should be completely covered by the concave hull of the specific patch of vegetation points that induced the void. Thus, to identify and remove these artefacts, all polygons in

1 The concave hulls are generated for the groups of points in the ground surface and vegetation datasets. The minimum point density that a group of points is required to have for the generation of concave hulls, depends on the value of the 'concavity' parameter specified in the concave hulls algorithm (see Section 3.2.1).



(a) The concave hulls of ground surface and vegetation points.

(b) The interior polygons are artefacts caused by vegetation removal.

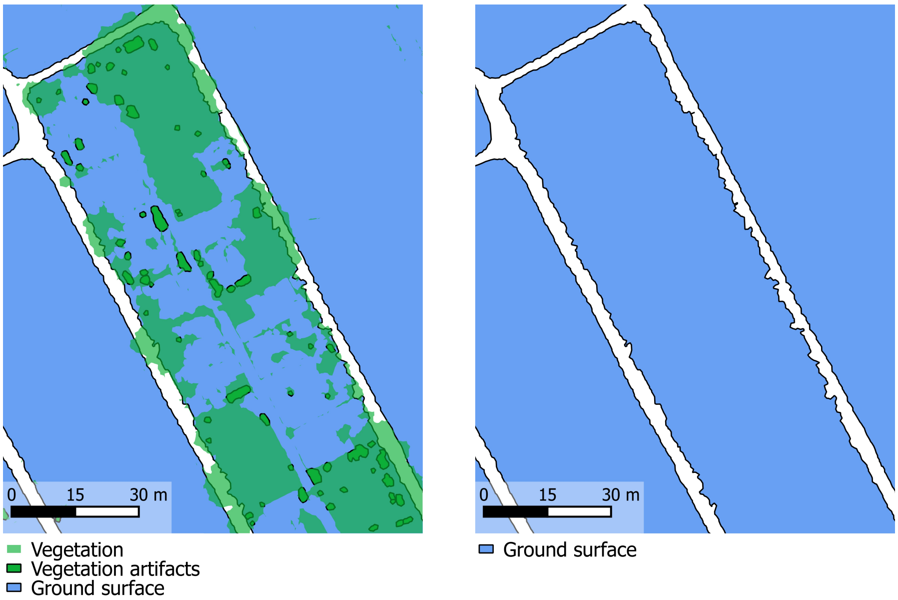
Figure 4.3: The removal of the vegetation points from the AHN3 dataset can leave voids within the groups of ground surface points. These voids then lead to artefacts represented by interior polygons in the dataset of concave hulls.

the dataset of ground surface concave hulls which are within the vegetation concave hulls are selected and subsequently dissolved with the ground surface polygon that has the largest shared common border (see Figure 4.4 for an example of this procedure). In practice, some of the artefacts are slightly larger than the concave hulls of the vegetation points, due to the manner of concave hull formation (see the example in Figure 4.5). Thus, before identification of the artefacts, the concave hulls of the vegetation patches are slightly buffered[2]. This ensures that the majority of these artefacts are successfully identified and removed.

STEP 4: REMOVING INTERIOR POLYGONS After eliminating the artefacts, the remaining interior polygons are removed. This results in a dataset such as in Figure 4.1b, with the concave hulls of the ground surface points, and the water courses in the space space between these polygons. The water courses enclosed by the ground surface points are now represented by holes in the concave hulls.

STEP 5: EXTRACTING THE WATER COURSE POLYGONS To obtain a dataset of the water course polygons, essentially the inverse of the concave hull dataset of ground surface points is required. Instead of polygons of the ground surfaces, polygons of the water courses are needed. To get this inverse dataset, a polygon can be generated which is as large as the total extent of the disjoint hull dataset. Then, the difference between this polygon and the concave hull dataset creates the inverse dataset, which thus gives the polygons of the water courses (see Figure 4.6).

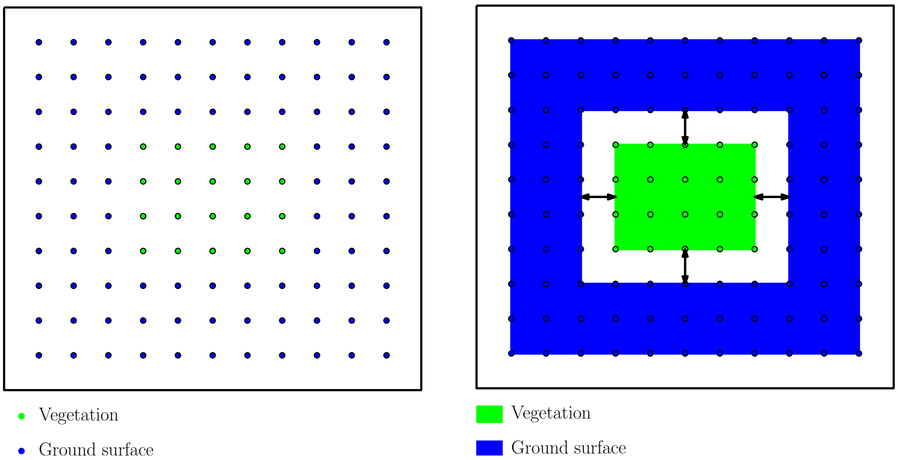
2 Twice the average point spacing of the point cloud provides a good measure for the required buffer value.



(a) The concave hulls of the vegetation points cover the vegetation artefacts.

(b) A clean dataset of ground surface concave hulls is formed by removing the vegetation artefacts.

Figure 4.4: The vegetation artefacts can be removed by identifying the interior polygons which are within the concave hulls of the patches of vegetation points, and accordingly dissolving these with the ground surface polygon that has the largest common boundary.



(a) A group of vegetation points contained by a group of ground surface points.

(b) The concave hulls of the groups of ground surface and vegetation points. There is empty space between them.

Figure 4.5: The vegetation canopy can be so dense that no ground surface points are measured underneath, causing sharp transitions between vegetation and ground surface points. If the concave hulls of these ground surface and vegetation points are formed, then there will be empty space between them, while the vegetation artefacts will have the size of the vegetation concave hulls plus the empty space. If these vegetation concave hulls are accordingly used to filter the vegetation artefacts, then they will not fully cover the artefacts, and the filtering result will be suboptimal.

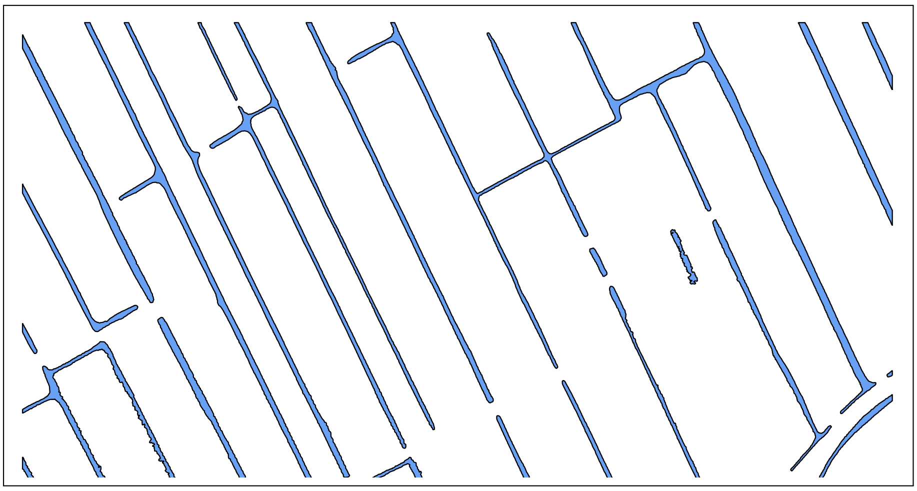


Figure 4.6: A dataset of water course polygons generated by the concave hull method.

STEP 6: SMOOTHENING THE WATER COURSE POLYGONS The water courses generated by the procedure above can have many irregularities, caused by perturbations in the water course banks[3]. These irregularities are largely smoothened through the use of three procedures; (1) a buffer-debuffer operation[4], (2) Boyle's Forward-Looking Algorithm[5] [Boyle, 1970], and (3) polygon simplification using the Douglas-Peucker algorithm [Douglas and Peucker, 1973]. Figure 4.7 shows how these three procedures significantly smoothen the polygon irregularities.

4.1.2 Identification of water course centre lines

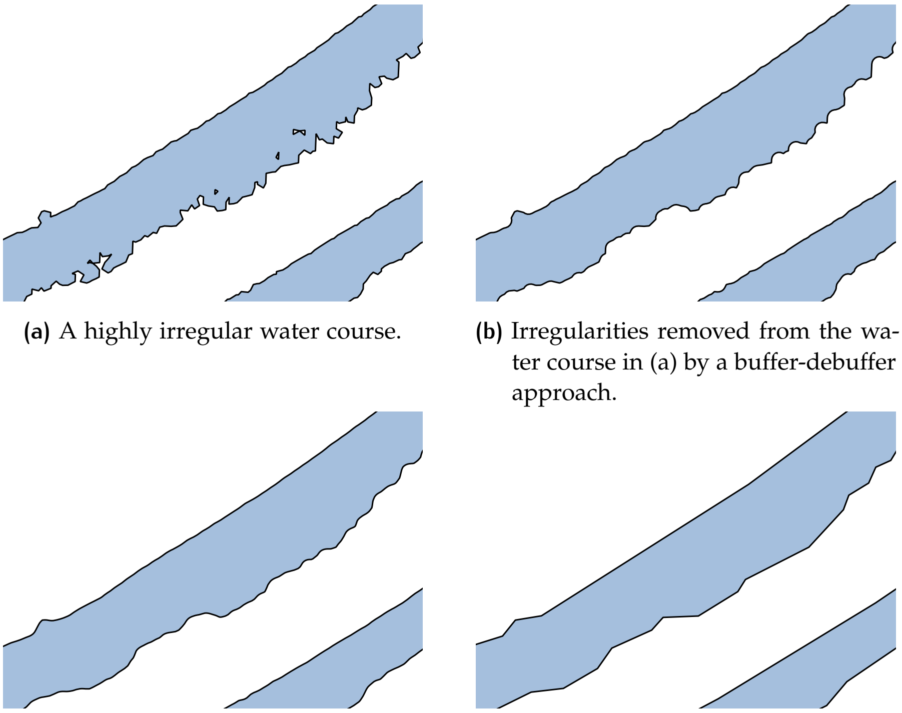
The method described in Section 4.1.1 can be used to obtain the polygons of the water courses, but needs to be extended for extracting the water course centre lines. Extracting polygon centre lines is an ongoing topic of research, and multiple techniques are available for the cause. A solid description of several procedures is given in Haunert [2008] and Zervakis [2015], of which a short summary is presented here:

- *Rasterisation and iterative collapse*: Polygons can be rasterised and iteratively thinned until the rasterised shape is at maximum one cell wide (described in Zervakis [2015]). The gridded skeleton can then be converted to a polyline, after which any unwanted branches should be removed. The accuracy of the resulting centre line approximation is highly dependent on the raster resolution, and increasing the raster resolution will also increase the required computation time. Furthermore, the rasterisation and subsequent vectorisation inherently infer

3 Note that many of these perturbations occur where vegetation is present on the water course banks. Possibly the vegetation deforms these banks through some physical processes, or the irregularities are caused by the vegetation removal in step 1 of this workflow.

4 A nice example of a buffer-debuffer operation is shown here: `http://blog.cleverelephant.ca/2010/11/removing-complexities.html`.

5 Boyle's Forward-Looking Algorithm is a "forward looking" interpolation method introduced by [Boyle, 1970]. A fictive line is drawn between a starting point and a point, n points ahead. Then, $1/n^{th}$ of the fictive line is stored, ending in point p, and a new fictive line is drawn from point p to the next point, n points ahead. This procedure is repeated until the fictive line connects to the end point. The algorithm does not change the number of points on the lines, it only translates them.



(a) A highly irregular water course.

(b) Irregularities removed from the water course in (a) by a buffer-debuffer approach.

(c) The water course in (b) smoothened using Boyle's Forward-Looking Algorithm.

(d) The water course in (c) simplified using the Douglas-Peucker algorithm.

Figure 4.7: A highly irregularly shaped water course generated by the concave hulls method. Such water courses can be smoothened by sequentially using a buffer-debuffer approach, Boyle's Forward-Looking Algorithm [Boyle, 1970], and the Douglas-Peucker algorithm [Douglas and Peucker, 1973].

a loss in positional accuracy [Liao and Bai, 2010], especially at lower raster resolutions.

- *Medial Axis Transform*: The MAT (see Section 3.3 and Peters et al. [2015]) can be computed to approximate the medial axis of any given polygon. This method gives a solid approximation of the polygon centre line provided the polygon has sufficient point density [Haunert, 2008], and denoising heuristics [Peters and Ledoux, 2016] can prevent formation of unwanted branches. However, a major disadvantage of this method is that it generates a set of medial points, and as such lacks topology.

- *Voronoi Diagram*: VDs (see Aurenhammer [1991]) can be constructed on the points of polygon boundaries, after which a line-in-polygon test can select the VD's edges which are completely within the polygons. These lines form the polygon centre line (see Section 3.1). This method gives a solid approximation of the medial axis [Haunert, 2008] provided the polygon has sufficient point density. McAllister and Snoeyink [2000] successfully approximated the medial axis of river networks with VDs.

- *Triangulation-based skeleton*: Triangulation-based skeletons can be based on constrained or conforming Delaunay triangulations. Polygons are triangulated, after which the triangles that share one edge with the polygon are selected. Skeleton edges are constructed by connecting the centres of the two other triangle edges. Triangulation-based skeletons have the disadvantage that small disturbances in the polygon boundary have a significant effect on the skeleton, producing a spike [Haunert, 2008].

- *Straight skeleton*: The straight skeleton was introduced by Aichholzer et al. [1995]. It can be constructed by shrinking the polygon in a stepwise fashion, which is performed by simultaneous parallel offsetting of the polygon edges. The resulting skeleton mostly consists of long straight lines. A large disadvantage of the straight skeleton is that it is sensitive to reflex angles close to 360°, in which cases the skeleton will diverge from the centre of the polygon [Haunert, 2008]. It is possible to solve this issue and approximate the medial axis with the straight skeleton, but additional procedures are required.

Obtaining the polygon centre lines by rasterisation and iterative collapse has the major disadvantage that rasterisation and subsequent vectorisation procedures are required, making this the less preferred option. Furthermore, triangulation-based skeletons are highly affected by disturbances in polygon boundaries, and approximation of the medial axis with the straight skeleton requires a relatively complex procedure. The MAT and VD approach require a less complex procedure to approximate the medial axis, but the MAT generates sets of points which lack topology. Based on this comparison, using VDs to approximate the polygon centre lines is the best option for this thesis. The required procedures (see workflow in Figure 4.8) are elaborated upon in the following paragraphs.

STEP 7: CREATING THE VORONOI DIAGRAM FROM POINTS The VD can be used to generate centre lines of the water course polygons. To generate the VD, first the polygons of the water course polygons have to be converted to points, since VDs are constructed from points[6]. Point density is an impor-

6 It is also possible to generate VDs from lines, but this is more complex (see Held [2001]).

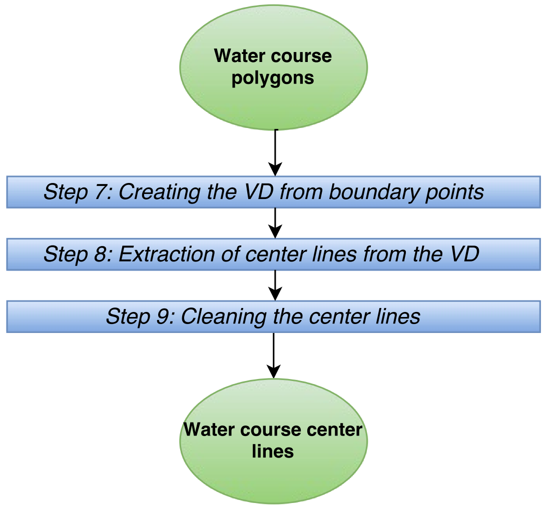


Figure 4.8: Conceptual workflow for the identification of the water course centre lines using the concave hulls method.

tant factor here, if point density is too low, then the generated centre lines will not approximate the medial axis well. Thus point density should be increased if needed, by inserting additional points on the polygons of the polygon, in between the original line vertices. The VD is then created on these points.

STEP 8: EXTRACTING CENTRE LINES FROM VORONOI DIAGRAM After creating the VD, the edges which make up the centre lines need to be selected. This is done by finding all edges that make up the VD, which are completely inside the water course polygons. Thus, the VD should first be deconstructed into it's edges, after which the appropriate edges can be selected.

STEP 9: CLEANING THE CENTRE LINES When using the VD of polygon boundary points to approximate the polygon medial axis, a number of *unwanted* branches are generated (Figure 4.9a), next to the actual *wanted* centre lines (Figure 4.9b). Branches are here defined as any edge with a 3-connected vertex on one end, and a 1-connected vertex on the other end. These branches can be removed by manual cleaning, but doing this for the entire extracted river network would involve a lot of manual labour, and defeats the purpose of developing an automated centre line extraction method. These branches can be automatically pruned using a number of different methods:

- *Shortest path between start and end point*: Zervakis [2015] describes a method which cleans the centre line by manually selecting a start and end point. The shortest path between the start and end point then gives the cleaned centre line.

- *Dissolving the Voronoi diagram*: This method presents an alternative to the line-in-polygon test described in Section 3.1 for the selection of centre lines from the generated Voronoi polygons. The line-in-polygon test is used to select the VD's edges which are completely within the water course polygons (se Figure 4.10a). Instead of using the line-in-polygon test, the Voronoi polygons can be dissolved with the channel bank polygons which they intersect[7]. In the case of Figure 4.10a, this creates two large polygons on either sides of the water course, the separation line of which is the centre line of the water course (see Figure 4.10b). This method essentially does not clean the centre line, but avoids unwanted branches being added to the centre line in the first place.

- *Removal of 1-connected vertices* To remove the unwanted branches, any vertices which are connected to only one edge can be removed. This can be done iteratively, until all unwanted branches have been pruned. A threshold on the number of iterations needs to be set to avoid removal of the entire centre lines.

- *Removing branches based on branch length*: A threshold branch length can be set, below which branches are automatically be removed. This can be done iteratively so that sub-branches are also removed.

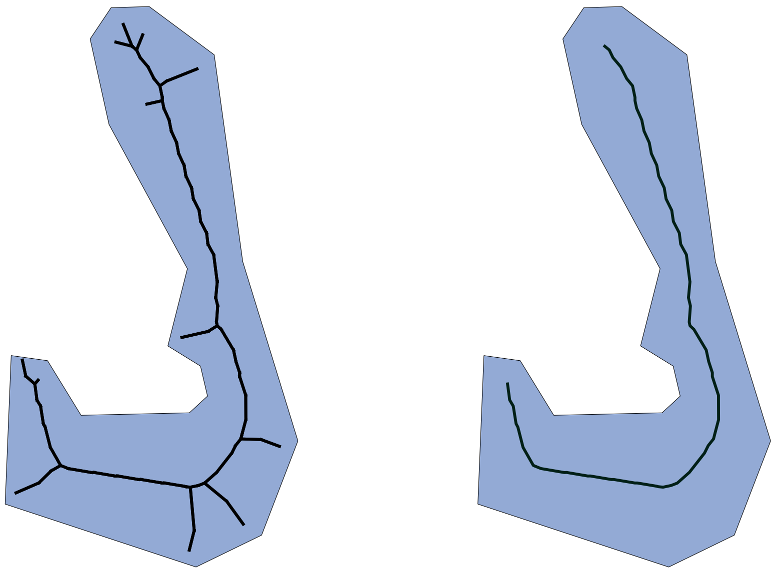
Pruning the unwanted branches by finding the shortest path from a start to an end point still requires the manual selection these points, which is not practical. Automatic selection of these points is difficult, especially for irregularly shaped water courses such as Figure 4.9. Extraction of a cleaner centre line by dissolving the VD presents a more elegant method of centre line extraction, but entirely depends on opposite river banks being represented by separate polygons. This will not be the case for many water courses (multiple water courses in Figure 4.1 have opposite channel banks represented by the same ground surface polygon), thus the method cannot be used here. This leaves two possible methods: (1) removal of 1-connected vertices, and (2) removing branches based on branch length. Both methods can be applied to the case presented here, of which the removal of branches based on branch length is slightly more elegant since it can remove all small unwanted branches, without the needs of setting a threshold on the number of iterations[8]. Consequently, this method was selected for use in this thesis.

Removing the branches based on branch length could be done by creating a graph network of the centre line network. Accordingly, all vertices with degree 1 (thus only 1 connected edge) are selected. Then for every of these vertices, their corresponding neighbouring vertices are traversed, until a vertex is encountered with degree 3 (thus having 3 connected edges, marking the start/end of a branch). The edges in between these vertices belong to the same branch, and are used to compute the branch length. If branch length is lower than a threshold, the vertices are removed.

4.2 MAT APPROXIMATION

The MAT approximation provides a method complementary to the concave hull (see Section 4.1) for the identification of water course centre lines. Whereas the concave hull method works in 2D, and is based on the absorption of the red LiDAR signal by water surfaces, the MAT uses the 3D shape of the landscape to identify water courses. The MAT complements the concave hull

7 Method obtained from: `http://www.ian-ko.com/resources/howto.htm`.
8 Note that a threshold is still needed for the branch length, but an appropriate length is more straightforward to estimate than an appropriate number of iterations.



(a) A water course polygon with a centre line with several unwanted branches.

(b) A water course polygon with a cleaned centre line.

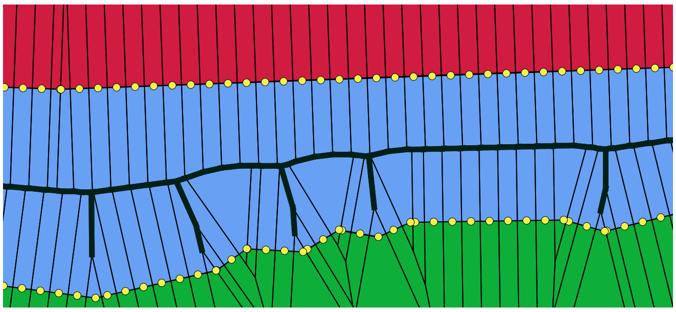
Figure 4.9: The extraction of centre lines from the VD (described in step 8 of Section 4.1.2), can lead to the formation of multiple *unwanted* branches. The cleaning method identified in step 9 of Section 4.1.2 removes the *unwanted* branches, and leaves the *wanted* centre line.

method by its ability to additionally identify dry water courses[9]. The underlying principle of the MAT, and an example of approximating the MAT for water courses is described in Section 3.3. Similarly to the concave hull method, visual derivation of water courses from an approximated outer medial axis in plan view (see Figure 4.12a) is relatively simple. This dataset of the outer medial axis, generated from the AHN3 ground surface points, forms the basis of my proposed MAT methodology for the identification of water course centre lines. Following from the subsequent required procedures listed in the conceptual workflow in Figure 4.11, the following paragraphs describe how the MAT method can be used to identify the water course centre lines.

STEP 1: FILTERING THE POINT CLOUD The MAT should approximate the outer medial axis of the landscape. Thus, all points except for the ground surface points are filtered from the AHN3 dataset (see Section 1.1 for a description) before approximating the medial axis, otherwise it will also be approximated for the objects in the other classes.

STEP 2: APPROXIMATING THE OUTER MEDIAL AXIS Subsequently, the outer medial axis is approximated for the filtered point cloud, for which the shrinking ball algorithm (see Section 3.3.1 and Peters et al. [2015]; Peters and Ledoux [2016]) can be used. The output will be a point cloud containing the outer medial axis of the landscape (see Figure 4.12a).

9 Note that the MAT has the potential to identify both wet and dry water courses, as long as relative water levels are not too high (i.e. there should be a sufficiently large portion of the water course banks visible to approximate the MAT).



(a) A VD of boundary points used to approximate the centre line (black) of a water course polygon (blue), in between two river banks (red and green). The centre line is defined as all edges of the VD which are completely within the water course polygons.

(b) Using the same VD as in (a), but the Voronoi polygons are dissolved with their corresponding river banks. This generates a centre line without unwanted branches.

Figure 4.10: Two different methods of centre line extraction using VD's.

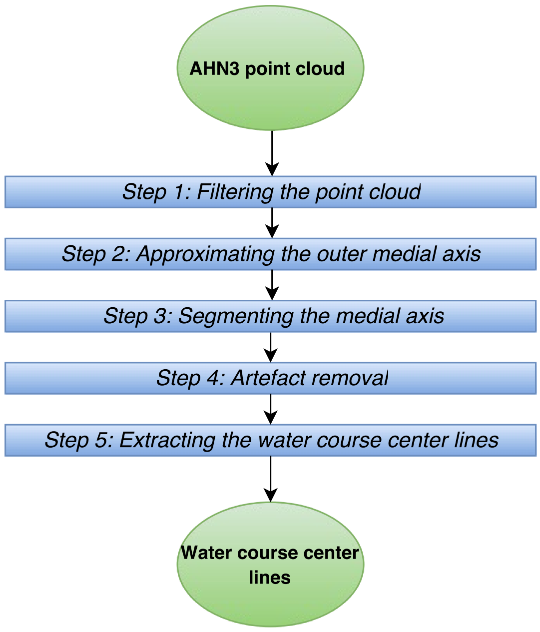


Figure 4.11: Conceptual workflow for the identification of the water course centre lines using the MAT method.

STEP 3: SEGMENTING THE MEDIAL AXIS The base dataset of outer medial axes is an unsegmented dataset, i.e. there is no distinction between individual water courses. For subsequent procedures it is preferable to have a segmented dataset of medial sheets, thus to distinguish as much as possible the individual water courses, which is done based on the medial bisector method described in Section 3.3.2. Figure 4.12b shows a dataset of medial sheets, resulting from the segmentation of the outer medial axis shown in Figure 4.12a.

STEP 4: ARTEFACT REMOVAL The MAT of the landscape (i.e. the ground surface points) approximates the medial axis for the water courses, but also for other landscape features reflected in the terrain. It is desirable to remove these artefacts as much as possible from the dataset of medial sheets before extraction of centre lines, to avoid erroneous results. This procedure removes artefacts from the segmented dataset, while keeping the medial sheets of the water courses intact as much as possible.

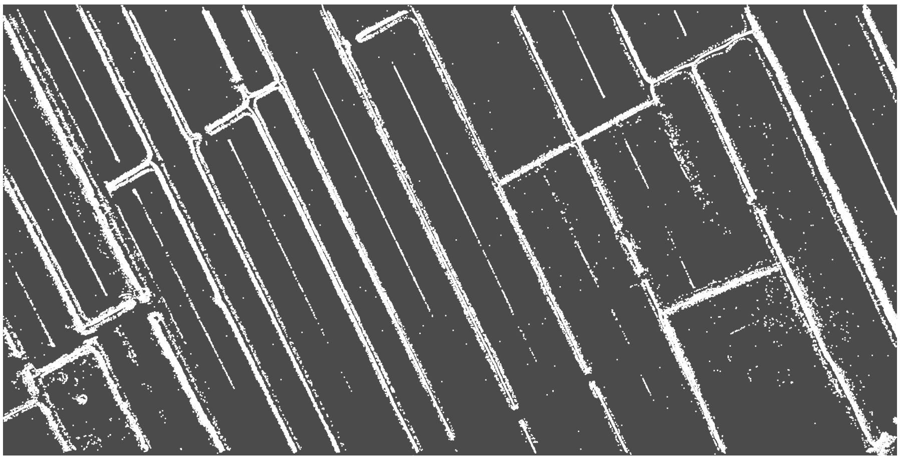
The medial sheets of the water courses have in common that they form large collections of points (see Figure 4.12b). However, some of the segmented medial sheets in the dataset are very small, meaning that they most likely do not belong to the network of water courses. By setting a threshold on the minimum number of points per segment, such smaller segments can be removed. This is used as a first step in the procedure.

A second step in the artefact removal design is based on point density in octree cells[10][11]. Octrees are spatial data structures which recursively subdivide space into smaller octants (see Figure 4.13), and are commonly used to segment and structure point clouds [Wang and Tseng, 2011; Zhou et al., 2011; Schön et al., 2013]. Octrees [Samet, 1990] can be constructed such that, as long as an octant contains a certain number of points, the octant is again subdivided into eight equally sized octants [Broersen et al., 2016]. This process can be continued until a predefined maximum number of subdivisions is reached. This predefined number of subdivisions then controls the size of the smallest possible octant, and thus the resolution of space segmentation. Broersen et al. [2016] describe an efficient way to segment a point cloud by using a linear octree and locational codes. The generated octree structure can be used to remove artefacts from the segmented dataset. It is possible to check for every cell in the octree, how many points are contained in this cell. If the number of points in the cell is lower than a specified threshold number, then these points are removed from the dataset. Since every cell in the octree which contains points is of the same size, this procedure in essence filters the point cloud based on point density. Since the water courses in the dataset consist of points clustered into dense medial sheets (see Figure 4.12b), point density in these respective cells will be higher than the threshold number, which should be set accordingly. The dataset resulting from the described procedure is much cleaner (see Figure 4.12c), making identification of individual water courses easier, while at the same time reducing the number of erroneously identified water courses[12].

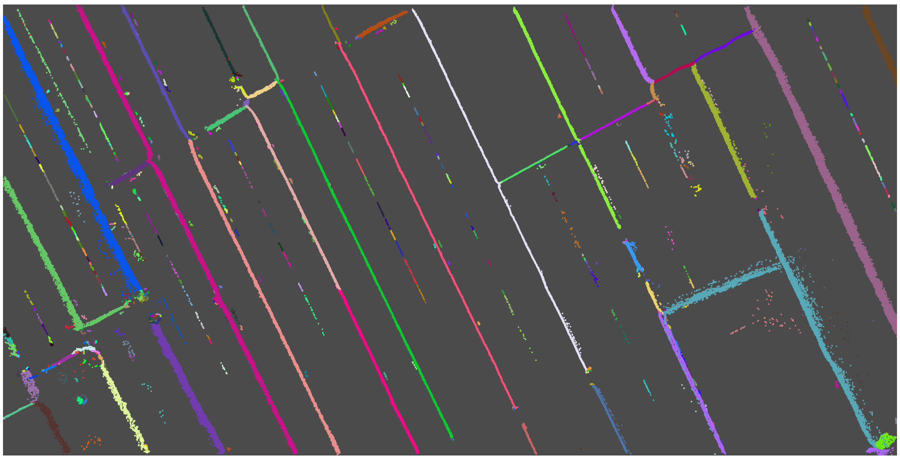
10 The Statistical Outlier Removal filter in CloudCompare (see `http://www.cloudcompare.org/doc/wiki/index.php?title=SOR_filter`) is also a good option for artefact removal, but I did not achieve sufficiently good results.

11 This second artefact filtering procedure is introduced since the first procedure does not filter artefacts which belong to any of the larger medial sheets.

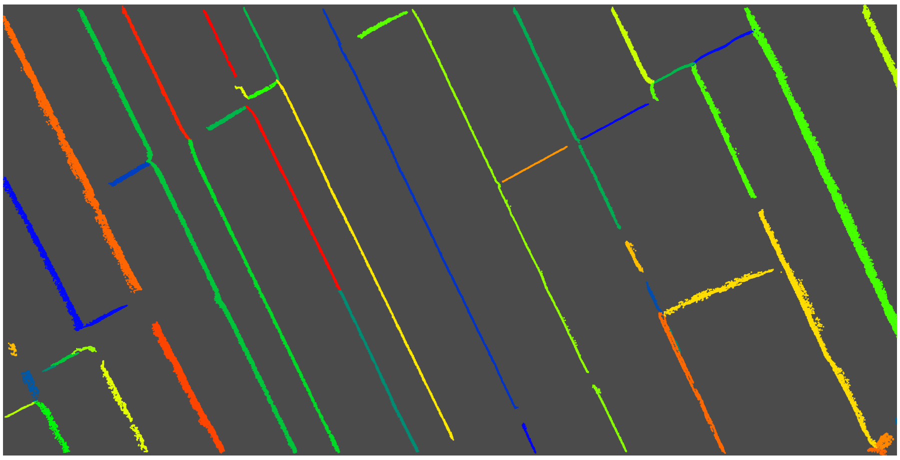
12 A sufficiently low threshold number prevents the vast majority of the 'good' water courses from being removed, but such removal can not be fully avoided using the presented denoising method. Future research should look into a more sophisticated artefact removal procedure.



(a) Unsegmented outer medial axis of the landscape.



(b) Outer medial axis in (a) segmented into medial sheets.



(c) Artefacts removed from the medial sheets in (b). Colours are randomly assigned.

Figure 4.12: Plan view of the outer medial axis approximated for the ground surface points in a subset of the AHN3 dataset, using the MAT method described in Section 4.2. The outer medial axis can be segmented into medial sheets using the method described in step 3 of Section 4.2. Subsequently, artefacts can be removed from these medial sheets using the procedure described in step 4 of Section 4.2.

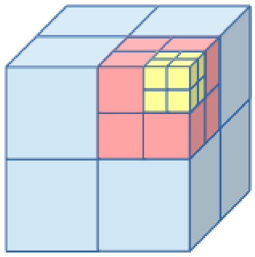


Figure 4.13: An example of a cubic segment of space, captured in an octree [Samet, 1990]. The space is recursively subdivided into eight equal-sized octants, until a specified maximum subdivision level is reached. Image courtesy of `https://geidav.wordpress.com`.

STEP 5: EXTRACTING THE WATER COURSE CENTRE LINES Approximating the outer MAT and subsequent segmentation and denoising, results in a relatively clean dataset of individual water courses (see Figure 4.12c). However, these water courses are still merely sheets of outer medial axis points, while the goal is to obtain the water course centre lines. There are multiple ways possible to come from this dataset to a dataset of centre lines:

- *Quadtree averaging*: A quadtree can be generated for the denoised dataset of outer medial axes, by minor alteration of the octree generation method described in Broersen et al. [2016]. Then, segment by segment, the (x,y) coordinates of the points in the quadtree cells can be averaged to form one averaged 2D point for every cell. This results in a 2D dataset of averaged points, where ideally the water courses are represented by a series of points, which can be connected using nearest neighbour searches to form centre lines.

- *Plane fitting*: Planes can be fit through the denoised medial sheets (e.g. using the RANSAC algorithm by Schnabel et al. [2007]), after which the channel centre lines can be obtained by intersecting each plane with a horizontal plane at the elevation of the water surface.

- *Polynomial or spline fitting*: It is possible to fit polynomials [Alexa et al., 2003] or splines [Fabio, 2003; Wang et al., 2006] to the medial sheets (see e.g. Deng and Han [2013] for a RANSAC splines fitting algorithm). Then, the channel centre lines can be obtained by intersecting each polynomial or spline with a horizontal plane [Lee and Fredricks, 1984] at the elevation of the water surface.

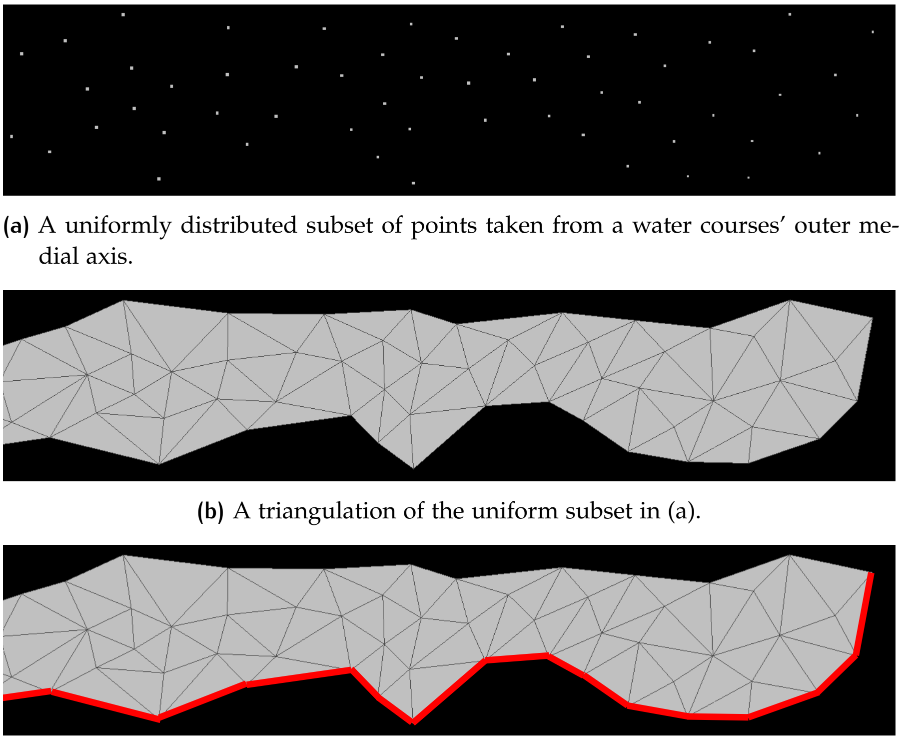
- *Triangulation*: The medial sheets can be triangulated to approximate the surface (see example in Figure 4.14), e.g. by using the BPA (see Section 3.3.3). The triangulation works best when performed on a uniformly distributed subset of points. A centre line of the water surface can then be derived by intersecting the triangulation with a horizontal plane, or by selecting the lower edge segments.

Quadtree averaging may work well if water courses are represented by straight lines only, intersections are simple, and the water courses have symmetrical cross sections (such as Figure 3.6). Unfortunately, there are many exceptions to these situations, and in these cases the presented method will not work well. Connecting neighbouring points will be particularly difficult near the intersections of medial sheets, and where multiple sheets are parallel and close to each other. Plane fitting can work well in situations where channels are straight, and can be applied to both symmetrical and asymmetrical water courses. But, some water courses are curved, and fitting planes through curved water courses will lead to suboptimal results. Fitting polynomials or splines to the medial sheets alleviates this problem, but requires a great deal of scripting, and intersecting these with planes is no trivial task. Approximation of the medial sheets by triangulation can be applied to any straight, tilted, or curved surface. Furthermore, intersection of a triangulation with a horizontal plane is relatively easy, and implementations of the BPA are readily available. For these reasons, triangulation is used in this thesis to extract the centre lines.

Theoretically, the lower points of the medial sheets are formed by the medial balls which touch the ground surface points that are closest to the water surface, thus the connection between these points should represent the centre line of this water surface. After triangulating the medial sheets (see Figure 4.14), these centre lines have to be extracted. One possibility is to intersect a horizontal plane with the triangulation, and accordingly calculate the line of intersection between the two shapes. However, this would only work for very regularly formed water courses, where the triangulation approximates a rectangular shape. For the water course in Figure 4.14, the shape is highly irregular, and intersecting a plane with the lower parts of the triangulation would lead to a discontinuous centre line. A second possibility involves extracting all the edges on the lower half of the triangulation, which together should form a continuous centre line (see Figure 4.14c). To extract these, I designed a simple algorithm: first the outlines of the triangulation have to be identified by selecting the edges which are in the triangulation only once. Then, the lower edges can be extracted from these polygons using the following principle (see Figure 4.15). Every edge on the outline of the triangulation has two vertices, v_1 and v_2, and is connected to a third vertex, v_3, which completes the triangle. To test whether the edge belongs to the upper or lower part of the outline, vertex v_3 is projected onto the corresponding edge at a right angle. Then, if the projected point v_3' is below vertex v_3 in terms of elevation, the edge is on the lower half of the outline. Otherwise, it belongs to the upper half.

4.3 COMBINED CONCAVE HULL – MAT METHOD

The concave hull method (presented in Section 4.1) and the MAT method (presented in Section 4.2) are based on different water course properties. While the concave hull method identifies water courses through the presence of water, the MAT method identifies water courses through their 3D shape (i.e. they typically exhibit a concave surface curvature). This is a potentially very useful combination, since the two properties can complement each other well in certain situations. In the case where a large part of the water course banks are under water due to high water levels, the MAT will have difficulties approximating the medial axis due to the low availability of points on the water course banks. But in the same situation the concave hull method may perform well due to the abundant availability of water, and thus high absorption of the LiDAR signal. Conversely, the MAT can perform better for low-water level situations, and has the potential to detect dry water courses. Furthermore, the concave hull method adds the ability



(a) A uniformly distributed subset of points taken from a water courses' outer medial axis.

(b) A triangulation of the uniform subset in (a).

(c) The selection of all lower edges of the triangulation in (b).

Figure 4.14: Medial sheets can be triangulated to approximate the surface, using the implementation of the Ball Pivoting Algorithm [Bernardini et al., 1999], after which the lower edges can be selected to represent the water course centre line.

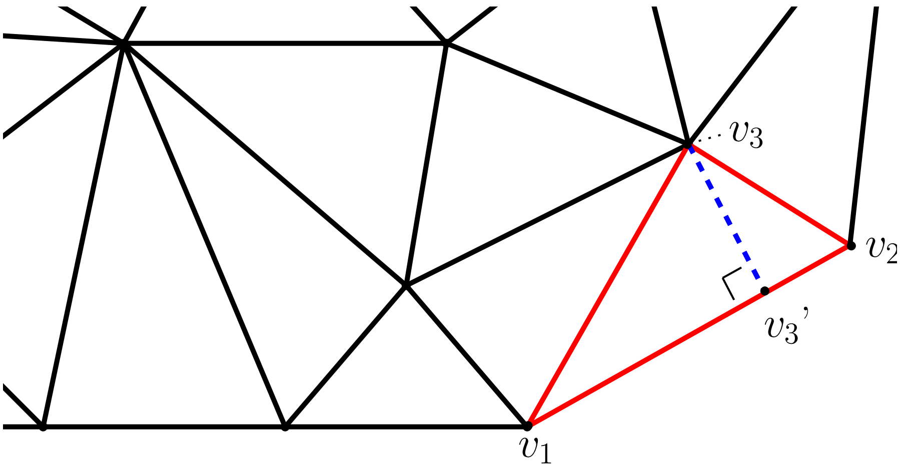


Figure 4.15: This figure shows a method to identify whether a boundary edge is on the lower or upper half of the triangulated surface. Vertex v_3 of the triangle is projected onto edge (v_1, v_2). Then, if the z coordinate of the projected point v_3' is lower than that of v_3, the edge belongs to the lower half.

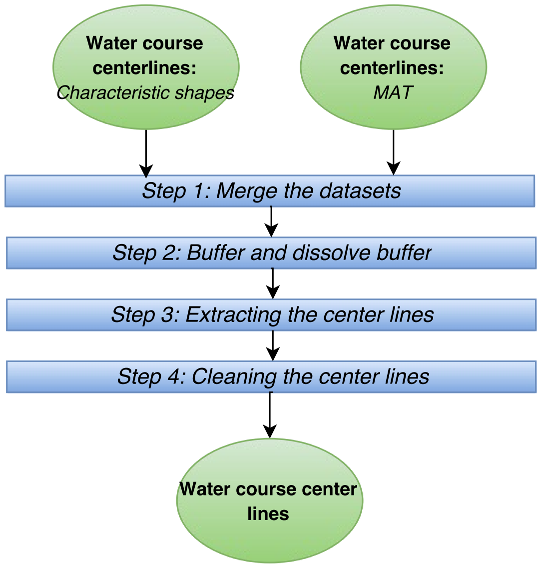
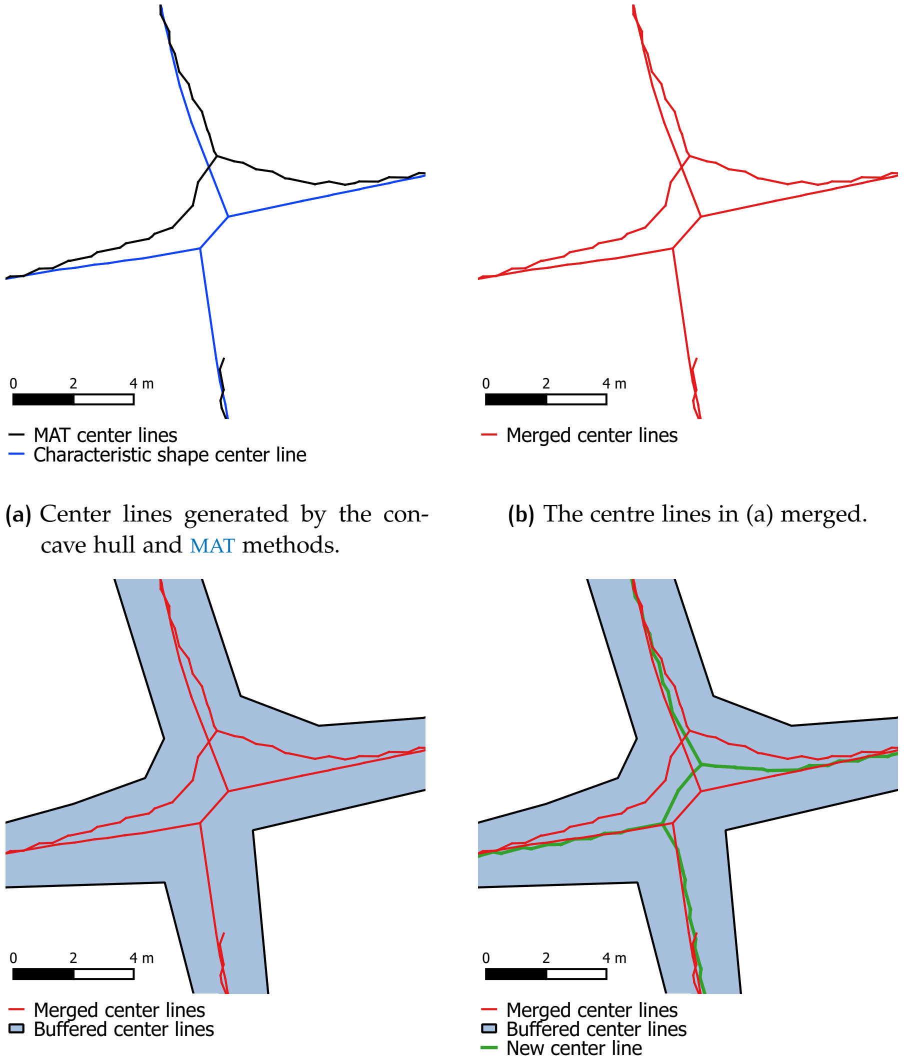


Figure 4.16: Conceptual workflow for the identification of the water course centre lines using the combined concave hull - MAT method.

to identify the water course polygons[13]. The combination of these methods can mitigate the weaknesses of both, extends their possibilities, and can make the methodoly as a whole more robust.

Both the concave hull and MAT method generate separate datasets of the water course centre lines. To combine these (see the conceptual workflow in Figure 4.16), the datasets are merged, thus forming a dataset with two centre lines for every water course[14]. By buffering these centre lines[15], and accordingly dissolving the buffers, one polygon remains for every water course. Then, the centre lines of these polygons are generated and cleaned by using the techniques described in Section 4.1.2 for the extraction of polygon centre lines for the concave hulls. See Figure 4.17 for a visualisation of the designed procedure.

13 In theory, this is also possible with the MAT method, but requires a more extensive procedure (see Section 7.3).
14 This method is easy to implement, and works sufficiently well for this proof of concept, but more sophisticated approaches are possible (see Section 7.3).
15 To estimate an appropriate buffer value (see also Section 7.3 about this topic), it would be a good measure to compute the average (or median) positional deviation of both datasets with respect to each other, and set the buffer value accordingly. For this, the 'point matching' procedure described in Section 5.4.2 can be used.



(a) Center lines generated by the concave hull and MAT methods.

(b) The centre lines in (a) merged.

(c) The merged centre lines of (b) buffered.

(d) New centre line generated for the buffer polygon.

Figure 4.17: The centre line generated by the concave hull (see Section 4.1.2) and MAT (see Section 4.2) methods are combined (see Section 4.3) by merging the centre lines, subsequently buffering them, and lastly by generating a new centre line using the method outlined in Section 4.1.2.

5 IMPLEMENTATION AND EXPERIMENTS

This chapter describes the implementation of the methodology presented in Chapter 4, and presents and briefly analyses the results of the application of this methodology to four different areas. First, Section 5.1 lists and describes all the tools that are used in the implementation, and Section 5.2 describes the areas and datasets which are used in the later experiments. Then, Section 5.3 describes in detail the prototype implementation of the methodology presented in Chapter 4, and also mentions to what degree the implementation is automated. A procedure for the validation of the datasets generated by the implemented prototype is presented in Section 5.4, and finally, the results of the experiments and error metrics are described in Section 5.5.

5.1 TOOLS USED

A large number of tools are used in this thesis to implement the methodologies described in Chapter 4. Section 5.1.1 lists and shortly describes the programs which are used in thesis for processing and visualisation of the datasets. A number of algorithms are used in this thesis to compute e.g. the concave hulls, the MAT, and triangulations of medial sheets. These algorithms and their implementations were not designed by me, and are therefore described separately in Section 5.1.2. Furthermore, a number of Python packages are used for convenience and automation, which are described in Section 5.1.3.

5.1.1 Programs

In this thesis, I made use of the following programs for processing, conversion, and visualisation purposes:

- Quantum GIS (QGIS)[1]: Geographical Information System (GIS) software used for many elementary GIS operations and visualisations, also in combination with System for Automated Geoscientific Analysis (SAGA) and Geographic Resources Analysis Support System (GRASS) GIS implementations.

- CloudCompare[2]: 3D point cloud processing software, used for file conversions, elementary point cloud processing, and visualisation.

- LAStools[3]: Software suite used for many point cloud operations, such as tiling and filtering.

1 QGIS: `http://www.qgis.org`.
2 CloudCompare: `http://www.danielgm.net/cc/`.
3 LAStools: `https://rapidlasso.com/lastools/`.

- MeshLab[4]: Software used for file conversions, visualisation, and triangulation (see Section 5.1.2) of point clouds.
- PostgreSQL[5]: Open source database used for storage and retrieval of the point clouds and generated octree.

5.1.2 Algorithm implementations

This section describes the implementations which are used in this thesis, but not designed by me.

LASBOUNDARY Lasboundary is a function available in LAStools. It reads LiDAR data and computes a boundary polygon for the points. It can compute the convex hull of points, but is also able to compute a boundary polygon to which it refers as the 'concave hull', which is based on the algorithm described in Section 3.2.1. This is currently the only freely available and practical implementation of the concave hull concept (see Section 3.2), thus I use lasboundary in this thesis to compute the required concave hulls.

MASBCPP This is a C++ implementation[6] of the shrinking ball algorithm (see Ma et al. [2012] and Section 3.3.1) developed by Peters et al. [2015] to approximate the MAT of a point cloud, which includes the noise heuristics (see Peters and Ledoux [2016] and Figure 3.3.1) to deal with noisy input data. It requires NumPy binary files as input, and is thus used in conjunction with Pointio.

POINTIO Pointio[7] is a small utility for managing point clouds as NumPy binary files (.npy). It is used to convert the point clouds in LAS format to NumPy binary files for input in masbcpp.

MEDIAL BISECTOR The medial axis is segmented into medial sheets based on medial bisector difference (see Section 3.3.2), for which a Python-based region growing algorithm is used[8], which utilizes the NumPy and pykdtree packages (see Section 5.1.3. The algorithm has two controlling parameters; a threshold on the bisector angle difference (°) , and the k-number in the k-nearest neighbours search. The bisector angle threshold determines how large of a difference in medial bisector angle is allowed between neighbourhoods, if the difference is above the specified threshold, then the neighbourhoods are regarded as separate segments. Thus, lower values of the bisector angle threshold result in more and smaller medial sheet segments. The k-number in the k-nearest neighbours search determines how many points are used to form the neighbourhoods for region growing. Here too, lower values of the k-number result in more and smaller medial sheet segments.

The values of the required bisector angle threshold and k-number depend on the properties of the medial axis. If the distribution of medial points is patchy, then high values of the k-number can prevent excessive segmentation on straight sections, but this also decreases the ability to segment the medial sheets at intersections. The required bisector angle largely depends

4 MeshLab: http://meshlab.sourceforge.net.
5 PostgreSQL: https://www.postgresql.org.
6 masbcpp: https://github.com/tudelft3d/masbcpp
7 Pointio: https://github.com/Ylannl/pointio
8 The implementation of the medial bisector method was obtained from Ravi Peters (Delft University of Technology) through personal communication, on February 15th 2016.

on the angle which the water courses make at their intersections. If they are at right angles, then higher values of the bisector threshold may be sufficient to segment them. However, if the water courses are nearly parallel oriented or the changes in bisector angle are very gradual, then lower values of the bisector threshold are required to segment them.

BALL PIVOTING ALGORITHM An implementation of the BPA (see Section 3.3.3) is available in MeshLab and is used in this thesis to triangulate the medial sheets. Three parameters need to be specified, which are specific to the implementation of the algorithm in MeshLab[9]:

- Pivoting ball radius (in meters): This is the radius of the ball which is pivoting over the set of points. Gaps that are larger than the ball radius will not be filled.
- Clustering radius (% of ball radius): To avoid the creation of too small triangles, if a vertex is found too close to a previous one, it is merged with it.
- Angle Threshold (in degrees): If an angle between two triangles is encountered that is too large, then the ball stops pivoting.

The triangulation of the medial sheets is significantly affected by the settings of these parameters. The value of the pivoting ball radius influences to what extent gaps in the medial sheets are allowed, i.e. if there are gaps in a medial sheet and the ball pivoting radius is smaller than the gaps, then only part of the medial sheet will be triangulated. Similarly, also a very high setting of the pivoting ball radius will negatively affect the triangulation; the triangulation will be too generalized, and smaller details will not be captured in the triangulation. The value of the clustering radius controls how much detail is retained in the triangulation. Low values of this parameter will lead to many smaller triangles being included in the resulting mesh, while high values remove many of the triangles and result in a very generalised result. The correct setting of this parameter depends mostly on the required detail of the mesh. Lastly, the angle threshold controls to what extent curves in the medial sheets are tolerated. If a low angle threshold is used, then the algorithm may stop the triangulation when a curve or corner is encountered in the medial sheet. If the value is set too high, then artefacts can be introduced into the mesh since any angle is allowed.

IMPLEMENTATION OF THE OCTREE GENERATION This thesis uses an octree generated on the points in the medial sheets to remove artefacts (see step 4 in Section 4.2). The methodology behind this specific octree generation algorithm is described in Broersen et al. [2016], and their implementation was slightly adjusted to the purpose of this thesis[10]. This implementation is entirely scripted in Python, and stores the resulting octree and the processed point cloud in a PostgreSQL database. The script makes use of the psycopg2 package (see Section 5.1.3) to communicate with the PostgreSQL database, and uses the libLAS package (see Section 5.1.3) to read the input point cloud in LAZ format.

9 The parameter descriptions are taken from the ‘help’ section of the BPA implementation in MeshLab.

10 A detailed description of the implementation of the octree generation algorithm of Broersen et al. [2016] can be found here: `http://repository.tudelft.nl/islandora/object/uuid:c9e55f6a-c874-4aee-9ceb-6bbcf34d9dc7/?collection=research`.

5.1.3 Python packages

The Python scripting language is used in this thesis for data conversion and automation of processes. A number of packages have been used, which are shortly listed and described here:

- pyshp[11]: Provides read/write support for the ESRI Shapefile format.
- NetworkX[12]: A Python software package that allows creation and manipulation of graph networks. Supports reading the ESRI Shapefile format natively.
- psycopg2[13]: A package that is used to access and manipulate the PostgreSQL database.
- libLAS[14]: A C/C++ library for reading and writing the ASPRS LAS format.
- pykdtree[15]: A package with kd-tree implementation for fast nearest neighbour searches. It is used in this thesis for the medial bisector implementation (see Step 3 of Section 5.3.3), and validation of the datasets (see Section 5.4).).
- Fiona[16]: OGR's[17] API for Python. Reads and writes data using multi-layered GIS formats. Used in this thesis to read shapefiles for the dataset validation (see Section 5.4).

5.2 AREAS AND DATASETS

The working area of the HDSR is located in the centre of the Netherlands (see Figure 5.1). Four areas of 3x3 km were selected which are together representative for the diversity of environments in the HDSR working area; an area with clay, peat, and sandy soils, and an urban area (see Figure 5.2 for their locations). The water course characteristics differ between these areas due to their different subsoils, geographical location, and land use, which makes them a good test case for the methods designed in this thesis. Datasets of AHN3 LiDAR points were obtained for each of these areas[18]. Per area, the total number of LiDAR points in the resulting datasets can exceed 200 million. To make these datasets more manageable, they were each divided into nine square tiles of 1x1 km each, while adding a buffer of 10 m to enable the designed methods to work well around the seams. The acquisition dates of the AHN3 LiDAR datasets for these areas all fall in winter 2014[19]. The AHN3 datasets for each of these four areas are used in this thesis to test the implementation of the concave hull (see Section 5.3.1), MAT (see Section 5.3.3), and combined concave hull - MAT method (see Section 5.3.4) for the identification of water course polygons and centre lines. Existing datasets of water course polygons and centre lines were obtained from the HDSR, which cover all four

11 pyshp: `https://pypi.python.org/pypi/pyshp`.
12 NetworkX: `https://networkx.github.io`.
13 psycopg: `http://initd.org/psycopg/`.
14 libLAS: `http://www.liblas.org`.
15 pykdtree: `https://github.com/storpipfugl/pykdtree`.
16 Fiona: `https://pypi.python.org/pypi/Fiona`.
17 OGR: `http://gdal.org/1.11/ogr/`.
18 The AHN3 datasets can be downloaded from: `https://www.pdok.nl/nl/ahn3-downloads`.
19 Acquisition dates of AHN3 were obtained from the header data in the AHN3 LAZ files.

Figure 5.1: Working area of HDSR (outlined in red) in the Netherlands.

identified areas, and are used to serve as a reference dataset to validate the datasets generated by the designed methods. All datasets involved in this survey use the CRS EPSG:28992. The areas and their general characteristics are listed in Table 5.1 and shortly described in the following paragraphs.

CLAY The clay area (see Figure 5.3a) is located near the small village of Cabauw, and has a rural setting with the majority of the land surface being used for agricultural purposes. Building coverage is low, and vegetation is present in patches around buildings, but is very limited in the meadows. The water courses can be visually readily distinguished from the landscape, since they often have water surfaces of 2 m in width or more. The elevation of the clay area varies between -2.5 m to +3 m NAP, and topographic relief is very low. There is very little relief, but water courses display enough local surface curvature to visually distinguish them from the rest of the landscape.

PEAT This area (see example in Figure 5.3b) has a rural setting, and likewise to the clay area, the majority of the land surface is used for agricultural purposes. It is located near the small village of Zegveld, where building coverage is equally low to the clay area. Similarly, vegetation is present in patches around buildings, but is very limited in the meadows. Water is omnipresent in this landscape. The water courses are often (much) more than 3 m wide, and it is thus relatively easy to distinguish them from the rest of the landscape based on water coverage. The elevation of the area varies between -2.5 m to +1.5 m NAP, and topographic relief is very low. The relative water levels in the water courses of this area is high, and some of the water

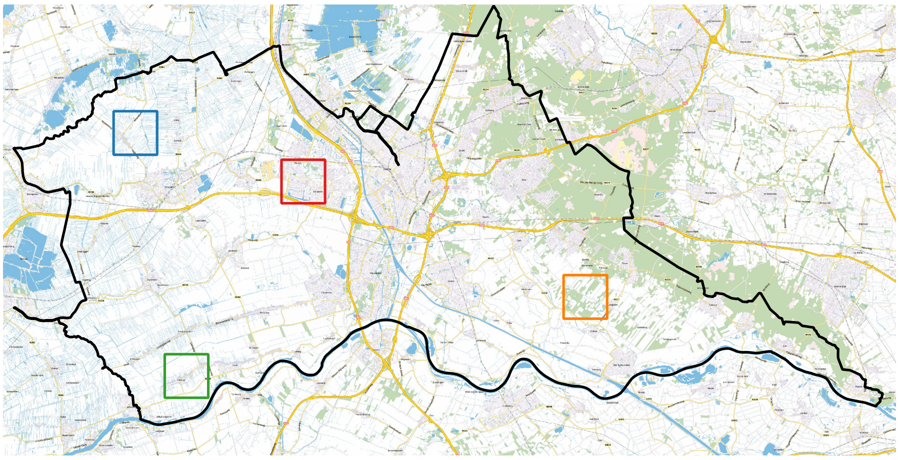


Figure 5.2: The four selected study areas, which are all 3x3 km in size. The HDSR working area is outlined in black. The extents of the selected study areas have been marked by: orange (sandy soils), blue (peat soils), green (clay soils), and red (urban area).

Table 5.1: This table lists some of the characteristics of the areas specified in Section 5.2. The specified location in EPSG:28992 indicates the (x,y) position of the lower left coordinate of the area (they are all 3x3 km in size). The percentage of building coverage is computed by dividing the total number of building points in every dataset by the total number of points in the dataset, which was done in a similar fashion for the vegetation coverage. The percentage of water coverage is computed by taking the total surface area of all water polygons in the HDSR reference dataset (see Section 5.2), and dividing it by the total surface area of the corresponding area.

	Area			
Characteristic	Clay	Peat	Urban	Sand
Location (in EPSG 28992)	(120279 , 440768)	(116785 , 457391)	(128271 , 454058)	(147565 , 446180)
Location (city / village)	Cabauw	Zegveld	de Meern	Langbroek
Setting (rural/urban)	Rural	Rural	Urban	Rural
Building coverage (in %)	2	2	14	0.5
Vegetation coverage (in %)	5	8	27	47
Water coverage (in %)	9	14	8	5
Elevation range (in m)	-2.5 to +3	-2.5 to +1.5	-2 to +10	+1.5 to +6

courses display very little local surface curvature, which makes it difficult to distinguish them visually from the rest of the landscape based on this property.

SAND This area (see Figure 5.3c) is located near the small village of Langbroek, and has a rural setting. Most of the land is used for agricultural purposes, but recreational and nature areas are also widely present. Building coverage is very low here, but vegetation is plentiful, with large patches of forest spread throughout the landscape. Open water is much less present in this area than in any of the other described areas, i.e. water surfaces are often around or below 1 m in width. Elevation in the area varies between +1.5 m to +6 m NAP, and relief is low, but water courses display enough local surface curvature to visually distinguish them.

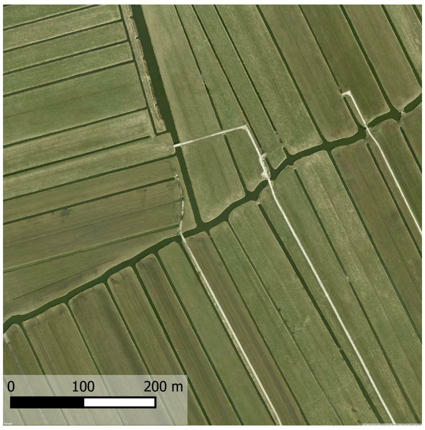
URBAN The urban area (see Figure 5.3d) is located near the large village of 'de Meern', where major construction works have been undertaken in the last five years. The area largely has a low-density urban setting, with building coverage naturally being the highest of the four selected areas. Vegetation is also plentiful here, especially near buildings and in human-engineered parks. There are many water courses in this area, often in the form of canals. In contrast to the other areas, water courses here are very variable in surface width, shape, and spacing. Elevation varies between -2 m to +10 m NAP, and relief is low, but local human-engineered surface convexities are often present (e.g. in the form of noise-cancelling levees).

HDSR REFERENCE DATASETS The HDSR has kindly provided me with a dataset of water course polygons, and water course centre lines. These products are the result of manual work[20]; they are created by visually identifying the extent and centre lines of the water courses from stereo aerial photos stemming from winter 2014, thereby striving for a 96% identification rate. After this procedure, the dataset is further modified after any possible mutations recorded by the HDSR. The aerial photos used for the water course identification, have an accuracy of about 0.1 m. The datasets of water courses resulting from the described manual procedure, have an accuracy of 0.15 to 0.2 m at best.

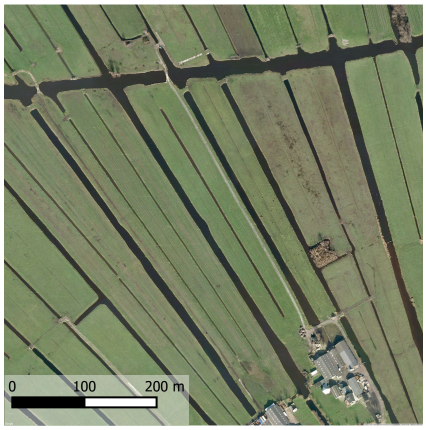
5.3 THE IMPLEMENTED PROTOTYPE

A methodology was designed in Chapter 4 which identifies water course polygons and centre lines in flat and engineered landscapes from AHN3. This section describes a prototype implementation of this methodology. The prototype is described in four separate subsections; (1) Implementation of the concave hull method to identify the water course polygons (Section 5.3.1), (2) Implementation of the concave hull method to identify the water course centre lines (Section 5.3.2), (3) Implementation of the MAT method (Section 5.3.3), and (4) Implementation of the combined concave hull - MAT method (Section 5.3.4). Finally, Section 5.3.5 describes the extent to which the implemented prototype is automated, and provides an estimate of required processing time.

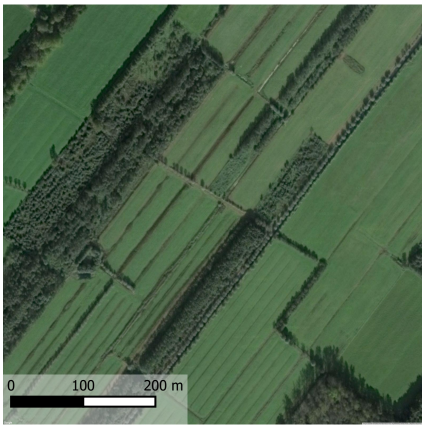
20 The information about the procedure and specifications of the reference datasets were obtained via personal communication with Roger de Crook (HDSR), and René van Ginkel (Arcadis).



(a) The clay area: little vegetation and clearly recognisable water courses.



(b) The peat area: little vegetation and very wide water courses.



(c) The sand area: a lot of vegetation and narrow water courses.

(d) The urban area: water courses are very variable in shape and size.

Figure 5.3: Characteristic top-down views of the selected areas used for testing the implemented methods (background aerial photos courtesy of Google Maps).

5.3.1 Implementation of the concave hull method to identify water course polygons

This section describes how the concave hull method was implemented to identify the water course polygons (see Section 4.1.1 for the methodology). A detailed workflow of the implementation is visualised in Figure 5.4, which shows the entire process from the unfiltered AHN3 point cloud to a shapefile of water course polygons. The following paragraphs provide a stepwise description of the implementation, including the required parameter values.

STEP 1: FILTERING THE INPUT POINT CLOUD The point cloud was filtered using LAStools' las2las function, which supports removal of points based on their classification. Two separate filtering procedures are performed, which result in the creation of two separate point clouds; one point cloud with ground surface and building points, and another point cloud with only vegetation points. The point cloud provided as input to this step should be in LAZ or LAS format, and similarly for the output.

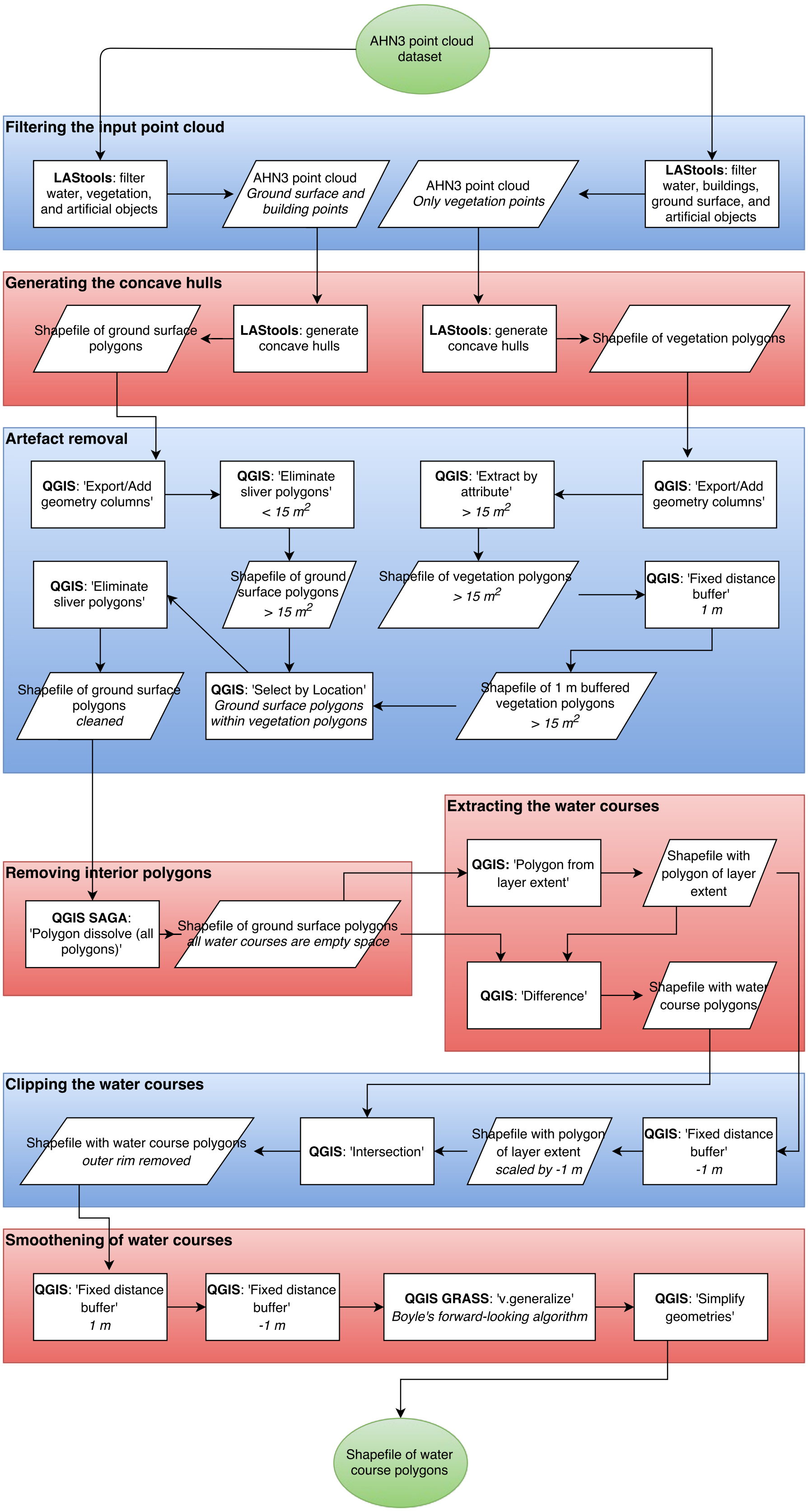


Figure 5.4: A detailed workflow of the implementation designed for the identification of water course polygons using the concave hull method.

STEP 2: GENERATING THE CONCAVE HULLS LAStools provides the lasboundary function which takes a LiDAR point cloud and computes a concave hull for the points. By default, the function generates connected concave hulls of the groups of points, but a '-disjoint' flag can be specified which leads to the creation of a separate concave hull for every group of points. Furthermore, the '-holes' flag can be given, which enables the computation of interior polygons in the data, i.e. the algorithm (see Section 3.2.1) also removes inner triangles instead of just the convex boundary. By using this flag, separate polygons are generated for water courses contained within other ground surface concave hulls (as shown in Figure 4.1b). Lastly, a value for the 'concavity' needs to be specified, which controls the threshold distance for voids to be considered as part of the exterior (or part of an interior polygon). For this thesis, I used a value of 1 m, meaning that water courses need to be at least 1 m wide to be identified by the procedure. Lower values of the concavity are possible, but introduce considerably more noise in the output. LAStools' lasboundary function is used to generate both the concave hulls of the ground surface and buildings point cloud, as well as of the vegetation point cloud. This results in the generation of two shapefiles with concave hulls.

STEP 3: ARTEFACT REMOVAL A chain of processes is used to remove artefacts from the two concave hull shapefiles. For both shapefiles, the QGIS function 'Export/Add geometry columns' is used to add geometrical information to the polygons. After this, the process for both shapefiles differs. For the shapefile of vegetation concave hulls, first the QGIS 'Extract by attribute' function is used to extract all vegetation polygons larger than 15 m^2, and save them to a new shapefile. The polygons are then buffered by 1 m using QGIS's 'Fixed distance buffer' function. For the shapefile of ground surface concave hulls, the QGIS function 'Eliminate sliver polygons' is used to dissolve all polygons smaller than 15 m^2. This function merges all selected polygons with neighbouring polygons based on the largest common boundary. A value of 15 m^2 is used since this removes many small artefacts, while leaving the vast majority of water courses intact, since their surface areas commonly exceed the threshold size. It is important to re-evaluate this value when applying the implementation to different environments and different settings of the concavity parameter. The previous step already filters many of the artefacts, but still leaves some larger patches. These could be removed by increasing the area threshold, but this increases the risk of removing the water courses themselves. Thus, an additional procedure is implemented which selects all ground surface polygons which are within the buffered vegetation polygons using QGIS's 'Select by Location' function. The QGIS 'Eliminate sliver polygons' function is then used to merge these polygons with the neighbouring polygon that shares the largest common boundary. The result is a relatively clean shapefile of ground surface concave hulls.

STEP 4: REMOVING INTERIOR POLYGONS The shapefile now consists of the concave hulls of ground surface points, while most of the water courses are represented by voids. However, there are still some interior polygons contained in the concave hulls, which also represent water courses. To convert these to voids, the SAGA function in QGIS 'Polygon dissolve (all polygons)' is used to dissolve all polygons, without keeping inner boundaries. This results in a dataset with one large multi-polygon of all concave hulls, with interior polygons removed. All water courses are now represented by voids.

STEP 5: EXTRACTING THE WATER COURSES In this step, the shapefile output by the previous step is inversed to obtain polygons of the water courses, instead of polygons of the ground surface points. First, the QGIS function 'Polygon from layer extent' is used to generate a polygon with size of the extent of shapefile with ground surface concave hulls. Subsequently, the QGIS 'Difference' function is used to obtain the difference between both shapefiles, resulting in a shapefile with polygons of the water courses.

STEP 6: CLIPPING THE WATER COURSES The procedure in step 5 takes the difference between a rectangular layer extent, and the concave hulls. However, these concave hulls have the property that they are not straight nor rectangular. Thus, by taking the difference between these layers, a narrow polygon is introduced along the edges of the dataset. To remove this artefact, the shapefile has to be clipped. First the QGIS function 'Fixed distance buffer' is used with the layer extent polygon as input. By specifying a buffer size of -1 m, and dissolving the result, a polygon is generated which is negatively scaled by 1 m in every dimension. Subsequently, the QGIS 'Intersection' function is used to intersect the scaled layer extent with the water courses shapefile, thereby generating a new water courses shapefile where the outer 1 m is cut off. The effectively removes the generated artefact.

STEP 7: SMOOTHENING OF WATER COURSES The polygons of the water courses are still highly irregular in shape, and need to be smoothened and simplified. First a buffer-debuffer procedure is executed, by first buffering all water courses by 1 m, and subsequently using a buffer of -1 m. For this the QGIS 'Fixed distance buffer' function is used. Secondly, Boyle's Forward-Looking Algorithm [Boyle, 1970] is used to further smoothen the water courses. This algorithm is available in the GRASS QGIS function 'v.generalize', and is used with a look ahead parameter of 5[21], which sufficiently smoothens the polygons, but does not change the polygon shape too much. Increase the value of this parameter if more intensive smoothening is required. Lastly, the polygons are simplified to remove more of the irregularities, using the QGIS 'Simplify Geometries' function (which uses the algorithm by [Douglas and Peucker, 1973], with a tolerance threshold of 0.5 m. The tolerance threshold controls the minimum distance between vertices, the higher the value of the threshold, the intenser the simplification. A value of 0.5 m was selected since it sufficiently simplifies the polygons, without changing too much of their original shape. This concludes the current implementation of the concave hull method to identify the water course polygons, and results in a shapefile of cleaned, smoothened, and simplified polygons of the water courses.

5.3.2 Implementation of concave hull method to identify water course centre lines

The following section describes how the concave hull method was implemented to identify the water course centre lines, after the methodology de-

21 The look ahead parameter controls the number of points that the algorithm uses to 'look forward'. The larger this number, the more intense the translation of points, resulting in more smoothening.

scribed in Section 4.1.2. A workflow of the implementation is visualised in Figure 5.5, which shows the entire process from the shapefile of water course polygons to a shapefile of water course centre lines. The following paragraphs provide a stepwise description of the implementation, including the required parameter values. The specified procedure follows logically on the procedure specified in Section 5.3.1 for the identification of water course polygons.

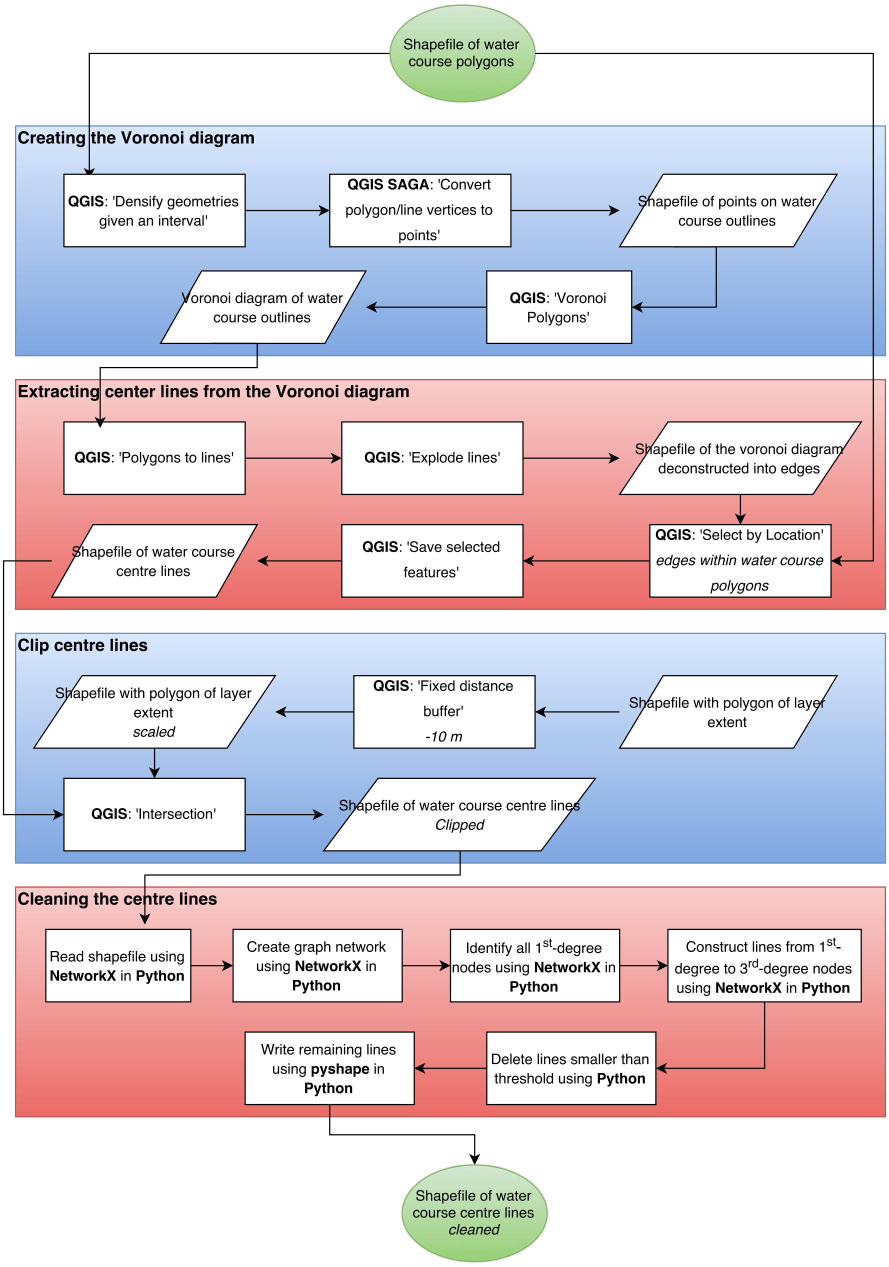


Figure 5.5: A detailed workflow of the implementation designed for the identification of water course centre lines using the concave hull method.

STEP 8: CREATING THE VORONOI DIAGRAM This step creates a VD on the boundary points of the water course polygons. The polygons are first densified to provide sufficient point density to generate a dense VD. For this the

QGIS 'Densify geometries given an interval' function is used, with an interval of 1 m[22]. The geometries are subsequently converted to a point dataset, using the SAGA QGIS function 'Convert polygon/line vertices to points'. Then, the QGIS function 'Voronoi Polygons' is used to generate a shapefile of the VD created on the densified boundary points.

STEP 9: EXTRACTING CENTRE LINES FROM THE VORONOI DIAGRAM To extract the centre lines of the water courses using the VD, first the diagram is converted to lines using the QGIS function 'Polygons to Lines', and subsequently the QGIS function 'Explode lines' is used to deconstruct the lines into edges. Then, the QGIS 'Select by Location' function allows to select the edges of the VD which are inside the water course polygons. The QGIS function 'Save selected features' is subsequently used to extract the selected edges to a new shapefile, which thus contains the approximated centre lines of the water courses.

STEP 10: CLIP CENTRE LINES The shapefile with centre lines has to be clipped to remove any buffer space added when tiling the AHN3 dataset (see Section 5.2). First, the centre lines are clipped by generating a clipping layer. This clipping layer is generated by using the QGIS function 'Fixed distance buffer', and using the previously created water course extent polygon as input, while specifying a buffer of -10 m, and dissolving the result. This generates a polygon which is negatively scaled in every dimension by 10 m. Then, this layer is intersected with the shapefile of centre lines using the QGIS function 'Intersection', which generates a clipped shapefile of water course centre lines.

STEP 11: CLEANING THE CENTRE LINES The generated shapefile of water course centre lines still contains many unwanted branches (see Section 4.1.2). A methodology was described in Section 4.1.2, which can be used to prune these unwanted branches. The methodology is implemented in Python, using the pyshp and NetworkX packages. Hereby, the NetworkX package is used to read the shapefile of centre lines, and accordingly generate a graph network of the edges in this shapefile. All 1^{st}-degree nodes (nodes which only connect to a single edge) are selected, after which their connecting edges are traversed until a 3^{rd}-degree node (which connects to three edges) is encountered. These edges then form a single branch. NetworkX provides functions to find the 1^{st}- and 3^{rd}-degree nodes. Subsequently, the line length of all these branches is checked, and the branch is removed if the length is lower than a specified threshold. A threshold of 20 m was used in this implementation, which removes many of the unwanted branches, but leaves longer -wanted- branches intact. The pyshp package is subsequently used to write the remaining edges to a shapefile, which forms the output centre lines of the concave hulls implementation.

5.3.3 Implementation of the MAT method to identify water course centre lines

This section describes how the MAT method (see Section 3.3 for the methodology) was implemented to identify the water course polygons. A workflow

22 The interval of 1 m means that additional points are inserted on the polygon boundaries in between the original vertices, until the euclidean distance between points or vertices is 1 m at maximum.

of the implementation is visualised in Figure 5.6, which shows the entire process from the unfiltered AHN3 point cloud to a shapefile of water course centre lines. The following paragraphs provide a stepwise description of the implementation, including the required parameter values.

STEP 1: FILTERING THE POINT CLOUD The AHN3 point cloud is filtered using LAStools' las2las function, leaving only the ground surface points. The point cloud provided as input should be in LAZ or LAS format, and the output should be LAS format.

STEP 2: APPROXIMATING THE OUTER MEDIAL AXIS To approximate the medial axis of the water courses (see Section 3.3 for a theoretical description), the proven and ready-to-use C++ implementation of the Shrinking Ball Algorithm, masbcpp (see Section 5.1), is used. Since masbcpp requires NumPy binary files (.npy) as input, Pointio (see Section 5.1) is used to convert the LAS file to NumPy Binaries. During this conversion, the original coordinates are translated to coordinates more suitable for masbcpp's procedures. Thus, additionally to converting the LAS file to NumPy binaries, Pointio generates a NumPy binary file which stores the translation parameters. After conversion, masbcpp is used to estimate the normals of the points in the point in the point cloud, and subsequently approximate the medial axis of these points. Parameter values of respectively 30° and 32° were used for the stable ball preservation and plane detection noise heuristics (see Section 3.3.1 and Peters and Ledoux [2016] for a description of these heuristics). These values were selected using trial and error, and proved to provide an optimal approximation of the water course's outer medial axis with the least amount of noise. The output of masbcpp consists of a NumPy binary with the approximated outer medial axis.

STEP 3: SEGMENTING THE MEDIAL AXIS The medial axis is segmented based on the difference in medial bisector (see Section 3.3.2), using a Python-based region growing algorithm (see Section 5.1). Using trial and error, segmentation proved optimal using a value of 6° for the bisector angle threshold, and a value of 9 for the k-number. For most of the water courses, the individual medial sheets are identified properly (see Figure 5.7a) using these parameter values. However, in some of the more complicated areas, segmentation is insufficient (see Figure 5.7b). This can have negative consequences during the later triangulation procedure, which works best if individual medial sheets are given as input, else part of the sheets are not triangulated. For these complex areas, a value of 7 for the k-nearest neighbours results in a more segmented dataset (see Figure 5.7c), and provides better results in the triangulation procedure for these areas. A downside of the lower k-number is that more smaller medial sheet segments are formed, some of which are so small that they are removed in the later artefact removal procedure. To combine the strengths of the two settings, and mitigate their weaknesses, the segmentation is performed separately with both parameter values. This produces two separate datasets of medial sheets, which are combined at a later stage. The segmentation script outputs a NumPy array with a segment id for every MAT point, which are subsequently converted to the Object File Format (.off) using a Python-based script with the NumPy package.

STEP 4: ARTEFACT REMOVAL The Object File Format files which form the output of the segmentation procedure, are converted by Meshlab to Poly-

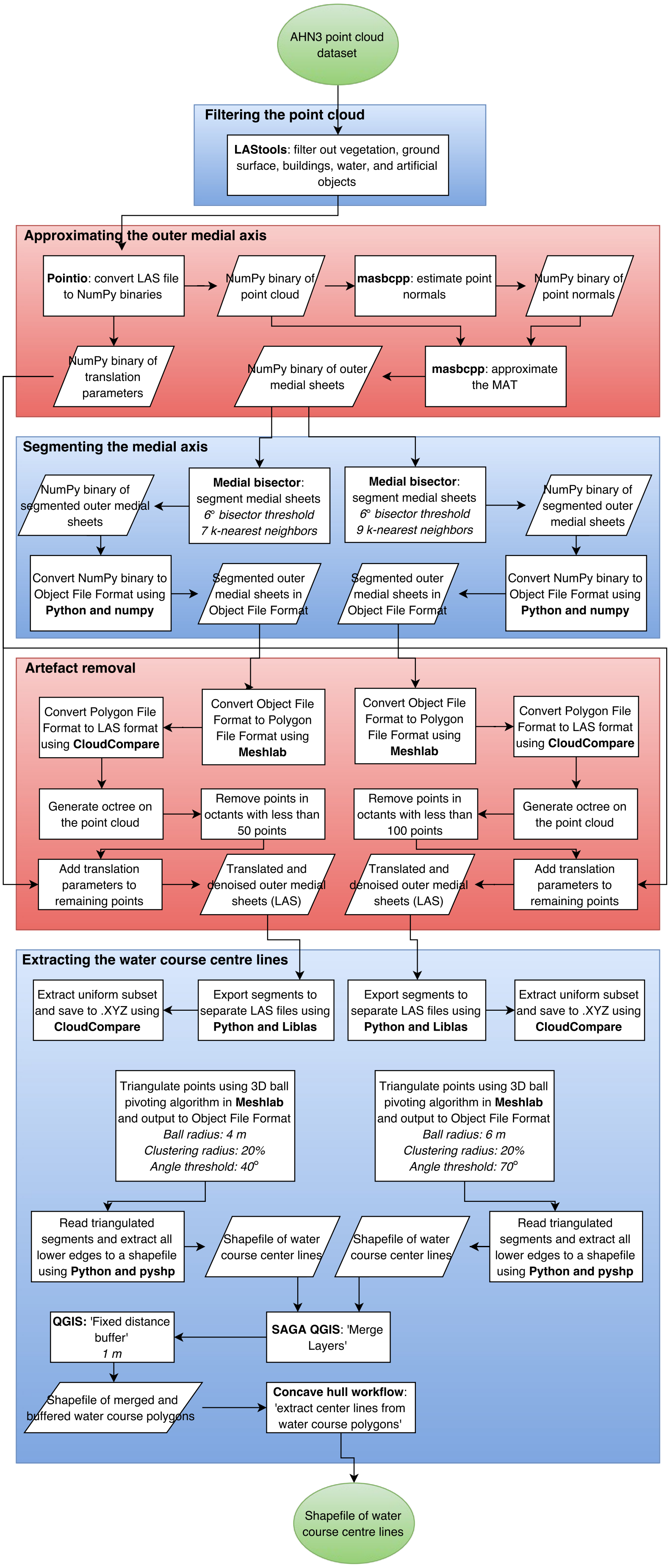
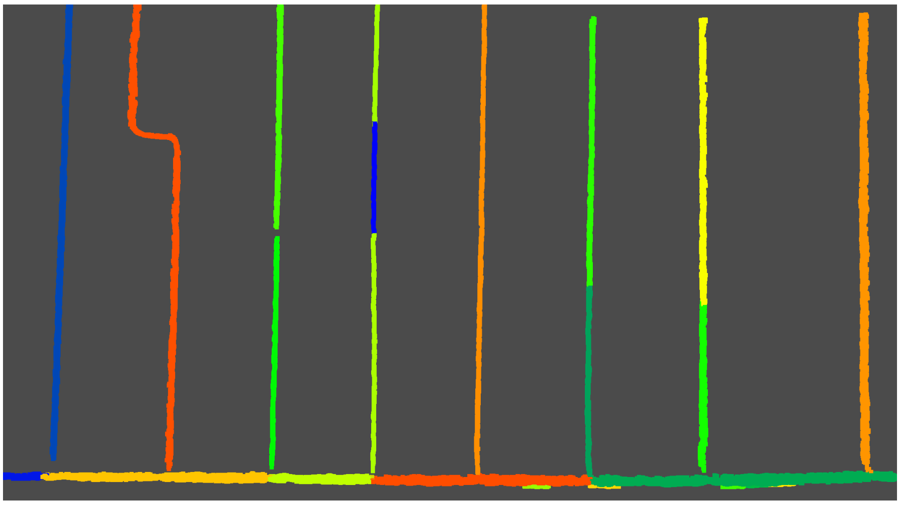
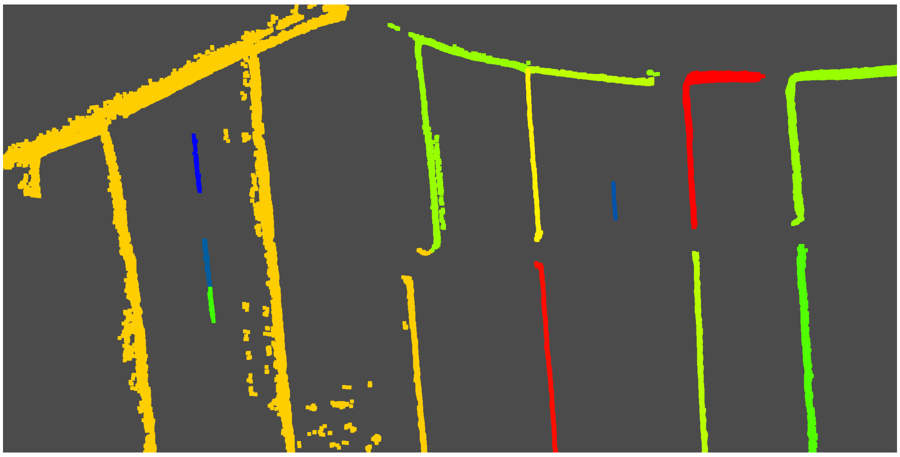


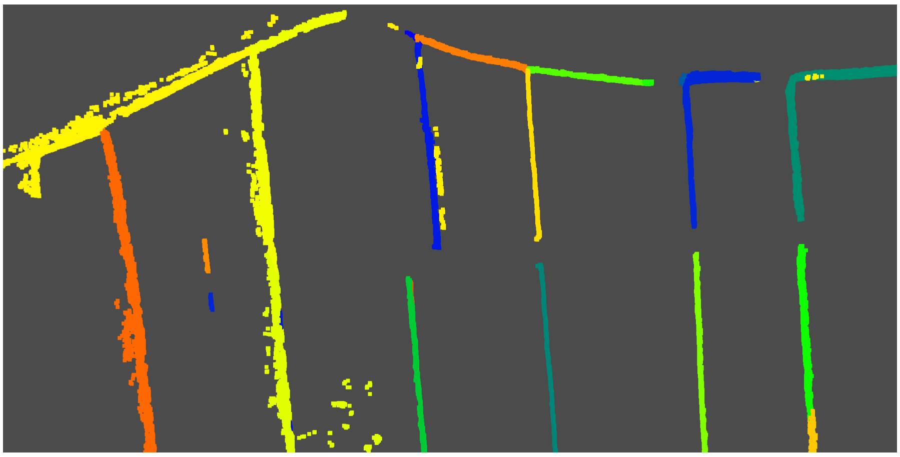
Figure 5.6: A detailed workflow of the implementation designed for the identification of water course centre lines using the MAT method.



(a) Medial axis segmented into medial sheets for a section with straight water courses. Using the parameter values 6° for the bisector angle threshold, and a value of 9 for the k-number. The segmentation performs well.



(b) Medial axis segmented into medial sheets for a section with curved water courses. Using the parameter values 6° for the bisector angle threshold, and a value of 9 for the k-number. The segmentation is suboptimal.



(c) Medial axis segmented into medial sheets for a section with curved water courses. Using the parameter values 6° for the bisector angle threshold, and a value of 7 for the k-number. The segmentation is slightly optimised with respect to (b).

Figure 5.7: The medial axis of the landscape as approximated by the MAT, segmented into medial sheets, using different parameters for the segmentation process. Note that the colours are randomly assigned to the segments.

gon File Format (.ply), and subsequently converted to ASPRS LAS format by CloudCompare. An octree structure is then created on the points in the segmented medial sheets, using the octree generation script (see Broersen et al. [2016] and Section 5.1). The octree is constructed such that all black leaf nodes (thus nodes which contain points) have the same geometrical size, which depends on the specified number of octree subdivisions. For the tile size of 1x1 km used here (see Section 5.2), the number of subdivisions was set to 8, which equates to an octant size of approximately 4x4 m. The octree generation script stores the entire point cloud and octree structure in a PostgreSQL database, thus a SQL query can be used to remove artefacts from the point cloud based on point density in the octants. I slightly modified the Python-based script by Broersen et al. [2016] to select only those points from the PostgreSQL database in octants with more than 5 points. This implementation uses the psycopg2 package to connect to the PostgreSQL database from Python. Since all octants have the same cell size, this algorithm essentially removes artefacts based on point density.

Additional artefacts are filtered by removing very small medial sheets segments, which are likely to be collections of artefacts. The threshold size for medial sheet segments to be removed depends on the segmentation parameters used in the segmentation algorithm (see step 3 of this section). Optimal thresholds (see Figure 4.12b and Figure 4.12c for the result) were obtained by trial and error, and are set to 50 points for the (6° bisector angle, 7 k-number) configuration, and 100 points for the (6° bisector angle, 9 k-number) configuration. This filtering procedure was performed in sequence to the previous filtering procedure, in the same Python script.

After the denoising procedures, the point cloud is corrected in a separate Python script for the offset introduced during the earlier medial axis approximation (see step 2 of this section), by translating the point cloud using the translation parameters stored in the corresponding NumPy Binary File.

STEP 5: EXTRACTING THE WATER COURSE CENTRE LINES The next step in the workflow involves the extraction of the water course centre lines from the medial sheets, which is done based on the implementation of the BPA (see [Bernardini et al., 1999] and Section 3.3.3) in MeshLab (see Section 5.1). The triangulation works best if it is performed separately for each medial sheet, using a uniform subsection of the points in the sheet. First, the package liblas is used in a Python script to load the LAS file with segmented medial sheets, and accordingly outputs a separate LAS file for each of the segments. Subsequently, CloudCompare is used to take a uniform subset of each of these segments, and export the segment to .xyz file. The uniform subset is taken such that the points are spaced approximately 2 m apart, which is dense enough for a solid representation of the medial sheets. Denser subsets can be used, but this significantly slows down the procedure. The .xyz files are then loaded into Meshlab one by one, and the medial sheet is triangulated using the Meshlab implementation of the BPA. The algorithm requires three parameters values (see Section 5.1): (1) pivoting ball radius (m), (2) clustering radius (% of ball radius), and (3) angle treshold (°).

The algorithm was tested for multiple different configurations of these parameters, and overall performed well using a 4 m ball pivoting radius, 20% clustering radius, and 40° angle threshold. However, it is difficult to find a configuration that is suitable for the successful triangulation of every medial sheet. I found that, especially for the dataset created with 9 k-nearest neighbours, this parameter configuration did not always perform well. This dataset is less segmented, and there may thus be larger gaps between groups of points. In these cases it is better to use a larger pivoting ball radius, and a higher angle threshold. Thus, for the 9 k-nearest neighbours dataset, the triangulation was performed using 6 m for the ball pivoting radius, 20% clustering radius, and 70° for the angle threshold. The combination of these two datasets is able to successfully triangulate nearly all medial sheets. All the triangulations are stored in the Object File Format (.obj). They are accordingly read by a Python script, and all the outer edges on the lower half of the triangulation outline are selected (using methods outlined in Section 4.2), and accordingly written to a shapefile using pyshp. This shapefile then contains the centre lines of the water courses.

The procedures above result in the creation of two different shapefiles with centre lines. These centre lines overlap in many places and do not connect well. To create one single dataset of centre lines, the same methodology (see Section 4.3) and implementation (see Section 5.3.4) is used as described for the combined concave hull - MAT centre lines. The result is a dataset with one centre line for every medial sheet.

5.3.4 Implementation of the combined concave hull - MAT method

To combined the datasets of water course centre lines stemming from the implementation in Section 5.3.2 for the concave hull method, and the implementation in Section 5.3.3 for the MAT method, the methodology described in Section 4.3 is followed. First, both centre line datasets are merged using the SAGA QGIS function 'Merge Layers', after which they are buffered by 1 m and dissolved, using the QGIS function 'Fixed distance buffer'. This results in the creation of one water course polygon for every water course, after which new centre lines can be derived for these polygons using the implementation used for generating the centre lines in the concave hulls implementation (see Section 5.3.2). The output is a shapefile of centre lines generated by the combined concave hull - MAT prototype.

5.3.5 Degree of automation and processing time

As stated in Section 1.7, I designed the methodology for water course identification with full automation in mind, but full automation of the implementation is not a requirement. However, processes are automated in the current implementation where possible, for practical purposes. The degree of automation is shortly described for the implemented prototype in the following paragraphs. Additionally, an indication of the required processing time is provided.

AUTOMATION OF THE CONCAVE HULL IMPLEMENTATION The concave hull implementation (see Section 5.3.1) is automated to a large degree. A batch file calls the LAStools program to perform Step 1 and 2 of the procedure; the point cloud filtering and concave hulls generation. Subsequently, the batch file starts QGIS, and a plugin is executed which performs all the processes described in Step 3 through to 10. Lastly, this plugin calls a Python script to perform Step 11. In principle, the method can be fully automated, although the current implementation requires the selection of appropriate parameter values. The processing time required for the current implementation varies between 10 to 15 minutes for a 1x1 km area[23].

AUTOMATION OF THE MAT IMPLEMENTATION Execution of the MAT implementation (see Section 5.3.3) is somewhat more cumbersome than for the concave hulls implementation. The first step in the process, filtering the point cloud, currently is done separately from the rest of the implementation. Step 2 and 3, approximation and segmentation of the medial axis are performed in Ubuntu which runs in a VirtualBox. These steps are automated by calling the required programs from a Bash script. Step 4 and 5 of the implementation are performed in the native Windows environment, and have been automated using a batch script. The CloudCompare procedures are executed by using its command line environment, and Meshlabserver is used to automate the MeshLab procedures. Similarly to the concave hull method, the method has the potential for full automation, although the current implementation requires the selection of appropriate parameter values. The processing time required for the current implementation varies between 30 to 45 minutes for a 1x1 km area[24].

AUTOMATION OF THE COMBINED CONCAVE HULL – MAT METHOD The combined implementation (see Section 5.3.4) is fully captured in a QGIS plugin, which automatically performs all required procedures to generate a new centre line. However, similarly to the other implementations, it requires the selection of appropriate parameter values. The processing time required for the current implementation varies between 5 to 10 minutes for a 1x1 km area[25].

5.4 VALIDATION OF RESULTS

The generated datasets are validated to provide an indication of the performance of the implementations, i.e. to indicate to what extent the generated datasets can be considered valid. The following error metrics are defined [Lillesand et al., 2008], for which reference datasets obtained from the HDSR (see Section 5.2) serve as reference data:

- *Error of omission*: The error of omission is defined as the percentage of water courses in the reference dataset, which are *not* identified by the respective method.
- *Error of commission*: The error of commission is defined as the percentage of water courses in the generated dataset, which are *not* in the reference dataset. Thus, this metric gives the percentage of water courses on the generated map which were erroneously identified as such.
- *Positional accuracy*: While the above metrics refer to the mapping accuracy, thus to the correctness of the map in terms of the percentage of omitted or committed water courses, the positional accuracy refers to the extent to which the actual position of the water courses is correctly

23 The indicated processing time is obtained by testing the implementation on a Windows 10 machine with an Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz, 8 GB DDR3 RAM, and a 7200 rpm hard drive.
24 See footnote 24.
25 See footnote 24.

indicated. It can be estimated by calculating the average positional deviation for multiple water courses in the generated dataset with respect to the reference dataset.

These metrics can give a solid indication of the mapping (errors of omission and commission) and positional accuracy of the generated datasets, but it has to be taken into account that the reference datasets can also contain errors (see a description of the reference dataset specifications in Section 5.2), which affects the computed error metrics. This complication has to be taken into account when evaluating the metrics[26].

Section 5.4.2 and Section 5.4.1 describe the possible methodologies to estimate these metrics for the datasets of water course polygons and centre lines, and the implementations are described in Section 5.4.3 for the water course polygons, and Section 5.4.4 for the centre lines. Note that there can be distinct differences between the validation results of the two datasets. Whereas the validation of the water course polygons indicates the extent to which the identification was correct in terms of *water surface area*, the validation of water course centre lines indicates the extent to which the identification was correct in terms of *the number of water courses*.

5.4.1 Validation of water course polygons

The mapping accuracies can be estimated by intersecting the generated water course polygons with the reference water course polygons. To estimate the error of omission, the polygon areas of the reference dataset which do not intersect with the generated dataset are summed, and accordingly divided by the total polygon area of the reference dataset. Similarly, the error of commission can be estimated by summing the polygon areas of the generated dataset which do not intersect with the reference dataset, and subsequently dividing these by total polygon area in the generated dataset. These mapping accuracies give an estimate of the extent to which the generated and reference datasets agree with respect to the total identified water course surface area, but not of positional accuracy. I currently do not have an adequate way of computing the positional accuracy for the generated dataset. The centre lines generated by the concave hull method are derived from the water course polygons, thus the positional accuracy computed for the centre line dataset (see Section 5.4.2) can also give an estimate of the positional accuracy for the water course polygons. However, it has to be kept in mind that some additional positional error may have been introduced during centre line generation.

5.4.2 Validation of centre lines

I identified three methods which can provide in validation of the generated centre lines; (1) manual selection of points of interest, (2) buffer operation, and (3) point matching. These methods are briefly described here, and a best method is selected based on weighing of their advantages and disadvantages.

- *Manual selection of points of interest*: Points of interest can be selected manually in the generated and reference centre line dataset, for exam-

26 Manual identification of water courses from aerial imagery or the AHN3 point cloud could provide an alternative reference dataset (which is also listed as possible future work in Section 7.3), but requires significant labour.

ple at well recognisable places such as intersections. The positional deviation of these points in the generated dataset with respect to the reference dataset, can then be measured and averaged, giving an indication of the positional accuracy for the generated dataset.

- *Buffer operation*: A buffer operation can be performed on the centre lines in the generated dataset, thus forming a set of polygons of which the size depends on the specified buffer. Then, the reference centre line dataset can be intersected with these polygons, and the line segments which do not intersect with any of the polygons are selected. By accordingly calculating the cumulative length of these lines, and relating this to the total centre line length of the reference dataset, an estimate of the error of omission is obtained. A similar procedure can be used to obtain the error of commission, for which the buffer operation should be performed on the reference dataset, and the centre lines in the generated dataset should be intersected with the resulting polygons. This validation method can give estimates of the mapping accuracy, but does not directly evaluate the positional accuracy of the generated dataset. However, some measures of the positional accuracy can be derived from the buffer operations and identified mapping accuracies. For example, if the centre lines are buffered by 0.5 m, and the error of omission is estimated at 15%, then this means that 85% of the water courses must have been positioned within 0.5 m distance of their position on the reference map. These metrics can be repeated for different buffers sizes, giving a better indication of the positional accuracy of the generated dataset.

- *Point matching*: Another method is to take a subset of points on the generated centre lines, and accordingly identify for these points what is the shortest euclidean distance to the centre line in the reference dataset. By averaging these distances, and estimate is obtained of the generated dataset's positional accuracy. Furthermore, it is possible to obtain the mapping accuracies by setting threshold distances. For example, if a threshold distance of 2 m is set, and the distance between a point on the generated centre line and the reference centre line is larger than this distance, then this point counts as an error of commission. Similarly, points can be selected on the generated centre line to find the error of omission. The metrics are computed by taking the number of points omitted or committed, relative to the total number of points. All the identified error metrics can be computed using random or uniform point matching.

From the identified methods, manual selection of points of interest is by far the most subjective, and only provides the positional accuracy and not the mapping accuracies. By using buffer operations, all metrics describing the mapping accuracy can be directly estimated, but a direct measure of the positional accuracy is not available and has to be derived from the mapping accuracy. Random or uniform point matching provides directly both the mapping and positional accuracy in an objective manner, and is thus selected as the most suitable validation method for the centre line datasets generated in this thesis.

5.4.3 Implementing the water course polygons validation procedure

Validation of the water course polygons generated by the concave hull implementation (see Section 5.3.1) was done after the methods described in Section 5.4.1. The following procedures were all performed in QGIS, manually for each of the four areas to which the concave hull implementation was applied. The dataset of water course polygons supplied by the HDSR (see Section 5.2) was used as reference data. First, the nine tiles (see Section 5.2) with generated water course polygons for an area are merged using the QGIS SAGA function 'Merge Layers'. The polygons in both the generated and reference datasets are then dissolved, forming essentially one large polygon for each of the datasets, using the QGIS SAGA function 'Polygon dissolve (all polygons)', while keeping the inner boundaries. The QGIS function 'Export/Add geometry columns' is then used to find the total surface area of the water courses in each of these datasets. The QGIS function 'Difference' is then used to generate a difference layer for both the reference and generated datasets, the resulting datasets of which respectively indicate the omitted and committed surface areas. The polygons in both datasets are then dissolved using the QGIS SAGA function 'Polygon dissolve (all polygons)', while keeping the inner boundaries, after which the QGIS function 'Export/Add geometry columns' is used to compute the total surface areas for the omitted and committed water surfaces. By dividing the omitted surface area by the total surface area in the reference dataset, and dividing the committed surface area by the total surface area in the generated dataset, an indication is obtained of the percentage of the water surface areas which were omitted and committed.

5.4.4 Implementing the water course centre lines validation procedure

The generated datasets are validated using the methods described in Section 5.4.2 for the water course centre lines. The methodology was implemented in a Python script, using the Fiona, NumPy, and pykdtree packages (see Section 5.1.3). The error metrics are computed separately for each of the four areas (see Section 5.2), and for each of the three implementations (Section 5.3.2, Section 5.3.3, and Section 5.3.4), thereby using the datasets supplied by the HDSR as reference data (see Section 5.2). The following paragraphs describe how the validation procedures were implemented.

PREPARING THE DATASETS The water course centre line datasets output by the concave hull and MAT implementations, as well as the HDSR centre line reference dataset, need to be prepared before metrics can be computed. First, the nine tiles (see Section 5.2) with centre lines are merged for every area using the QGIS SAGA function 'Merge Layers'. Then, the QGIS SAGA function 'Convert lines to points' is used to convert the reference and generated centre line datasets to points, while inserting additional points every 0.1 m for the reference dataset, and every 1 m for the generated datasets. The additional insertion of points every 1 m for the generated datasets is needed to ensure that the validated points are approximately evenly spaced, and that points are present in the entire network. Points were inserted every 0.1 m for the reference dataset, to ensure that these points are spaced densely enough to make possible an accurate estimate of the approximate distances between the centre lines.

COMPUTING THE ERROR METRICS To compute the metrics, I wrote a small Python script that uses the Fiona package to read the shapefiles containing the generated and reference points, which result from the data preparation procedure. These points are stored in two separate NumPy arrays, after which pykdtree is used to generate kd-trees on the arrays of points, which enables swift nearest neighbour finding. The errors of omission and commission are then computed in two separate procedures:

- *The error of omission* is computed by using nearest neighbour searches for the points in the reference dataset, thereby identifying for every point, the distance to the closest point in the generated dataset. Then, if the distance between the reference point and its closest neighbour in the generated dataset is larger than 5 m, then the reference point is counted as an error of omission. The number of reference points identified as error of omission is then divided by the total number of points in the reference dataset, which gives the error of omission for the generated dataset.
- *The error of commission* is computed by using nearest neighbour searches for the points in the generated dataset, thereby identifying for every point, the distance to the closest point in the reference dataset. Then, if the distance between the generated point and its closest neighbour in the reference dataset is larger than 5 m, then the generated point is counted as an error of commission. The number of generated points identified as error of commission is then divided by the total number of points in the generated dataset, which gives the error of commission for the generated dataset.

The positional accuracy of the generated dataset is obtained by taking the average of the distances from the generated points, to their closest neighbour in the reference dataset, but only for those generated points which have not been identified as error of commission. The described validation procedure is done separately for every dataset of centre lines (thus for the different areas and different methods).

5.5 EXPERIMENTS

This section describes the performance of the prototype implementations for the identification of water course polygons (see Section 5.3.1) and centre lines (see Section 5.3.2, Section 5.3.3, and Section 5.3.4) for all four areas. The metrics for the different areas are displayed in Table 5.2 for the water course polygons, and Table 5.3 for the centre lines. Section 5.5.1 describes the mapping accuracies, and Section 5.5.2 describes the positional accuracies. For a description of the areas used in these experiment, the reader is referred to Section 5.2.

5.5.1 Mapping accuracies

CLAY Performance of both the concave hull and MAT method is good for the clay area (see the virtually absent errors of omission in Figure 5.8a and Figure 5.8b). Both methods manage to identify more than 95% of all water courses, while the two methods combined even identify more than 98%. Furthermore, the error of commission is below 1.5% for the concave hull

Table 5.2: This table lists the metrics which were computed (after Section 5.4.1) for the water course polygons generated by the concave hull, for the clay, peat, sand, and urban areas.

	dataset			
Error metric	Clay	Peat	Urban	Sand
Error of omission (%)	13	10	6	43
Error of commission (%)	6	8	11	15

Table 5.3: This table lists the metrics which were computed (after Section 5.4.2) for the centre lines generated by the concave hull, MAT, and combined concave hull - MAT, for clay, peat, sand, and urban areas.

		dataset			
Error metric		Clay	Peat	Urban	Sand
Positional accuracy (m)	Concave hull	0.5	0.7	0.7	0.6
	MAT	0.6	0.8	1	0.8
	Combined concave hull - MAT	0.6	0.7	0.8	0.9
Error of omission (%)	Concave hul l	5	5	9	58
	MAT	4	15	15	26
	Combined concave hull - MAT	2	3	5	24
Error of commission (%)	Concave hull	1	2	17	4
	MAT	8	8	46	17
	Combined concave hull - MAT	8	8	47	17

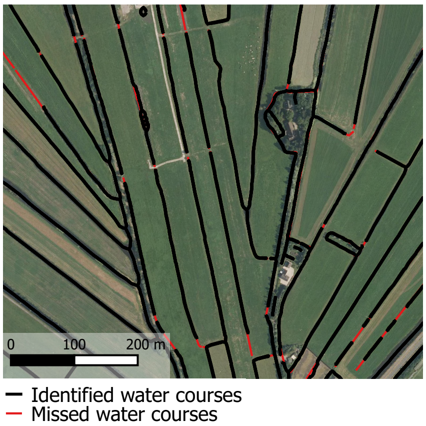
and 8% for the MAT and combined concave hull - MAT. Also for the omission and commission of the total water surface area (13% and 6%), the concave hull method performs relatively well[27]. The clay area is in many ways an ideal area for both methods to identify water courses. This is because the water course surfaces are wide, easing identification using the concave hull, and also the water courses have sufficient concavity to be identified by the MAT. Furthermore, vegetation in the area is mostly limited to small patches near built environments, and does not cover much of the water courses.

PEAT For the peat dataset, the concave hull performed equally well as for the clay dataset, with 95% of all water courses identified. The total percentage of water surface area identified also scored similarly to the clay area, with 90%. However, the MAT performed less (see the difference between Figure 5.8c and Figure 5.8d), with only 85% of water courses identified. The concave hull method performs equally well here since water surfaces are even wider than in the clay area (see also Section 5.2), and can thus by identified based on this characteristic, and also there is little vegetation covering these surfaces. However, the water courses display less clear concave profiles, since relative water levels are higher here (thus many water courses have only very small banks), which impedes classification by the MAT in multiple cases. Although the MAT performs less for this area, it still manages to identify some of the water courses which were not identified by the

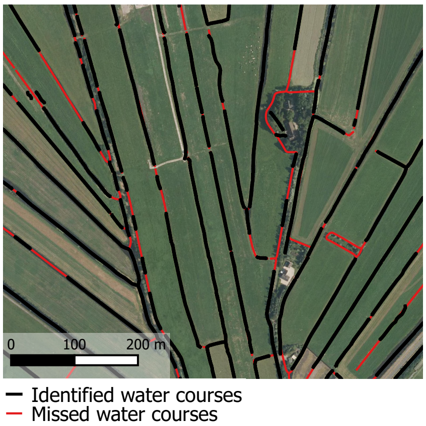
27 Note that the differences of water course outline identification between the concave hull method (Section 5.3.1) and the HDSR (Section 5.2) are markedly different, thus there will always be some error of omission of commission for every water course polygon.

(a) Error of omission of the clay area for the concave hull.

(b) Error of omission of the clay area for the MAT.



(c) Error of omission of the peat area for the concave hull.

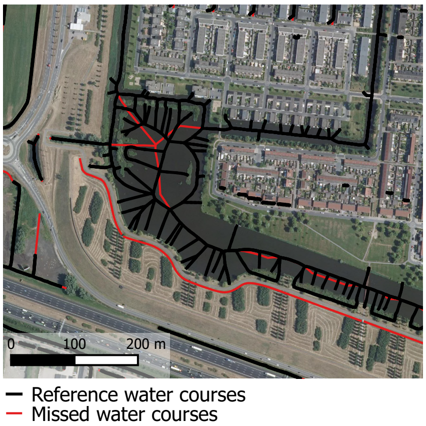


(d) Error of omission of the peat area for the MAT.

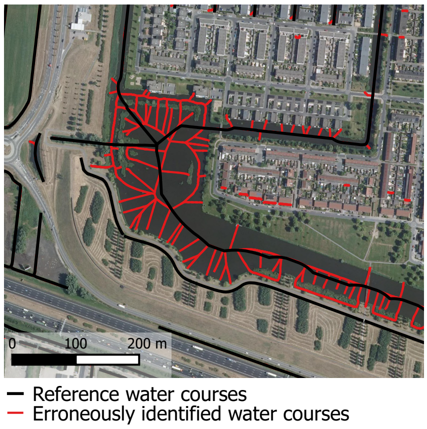
Figure 5.8: The errors of omission for the generated centre lines shown for the concave hull and MAT for the clay and peat areas (background aerial photo courtesy of PDOK).

concave hull method. This is indicated by the fact that the combined concave hull - MAT method identifies roughly 97% of the water courses, which is more than the concave hull method does by itself. The errors of commission are comparable to the clay dataset.

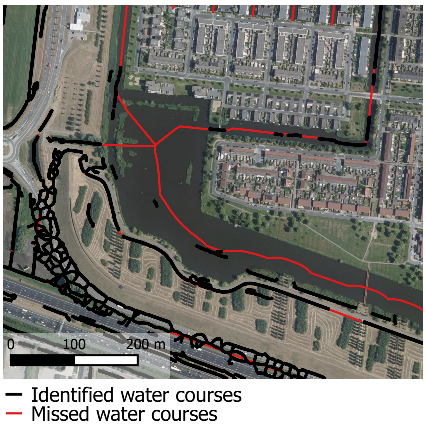
URBAN For the urban dataset, the concave hull identified more than 90% of all water courses, while the MAT identified approximately 85%. This area is characterized by water courses which are very variable in shape of the water surface and concavity of their banks. Although the concave hull method has no problems identifying the open water surfaces (as indicated by the 6% error of omission in terms of water surface area), since they are mostly wide enough here, it does not do a good job estimating the centre lines for larger water surfaces. In these cases, many branches are formed (see Figure 5.9b), which are not removed in the branch pruning process since they are generally longer than the threshold for branch removal. This causes many of the errors of commission for the concave hulls approach (which is also indicated by the fact that the error of commission of the water surface area is somewhat lower with only 11%), but also increases the error of omission since the centre lines may be out of place and thus counted as error of omission. Furthermore, there are relatively many dry water courses in the urban environment, which are not identified by the concave hull method (see Figure 5.9a). The MAT has relatively high error of omission (see Figure 5.9c) due to the irregular channel banks (e.g. vertical or very small banks), which lead to highly fragmented or no medial sheets. However, what is more important is the high error of commission for the MAT, with more than 45% of the dataset being erroneous. Many of these errors are caused by the many human-engineered concavities and convexities (see Figure 5.9d) which are present in this urban environment, many of which are not water courses. The combined concave hull - MAT identifies 95% of all water courses, but also contains all the errors propagated from the MAT.



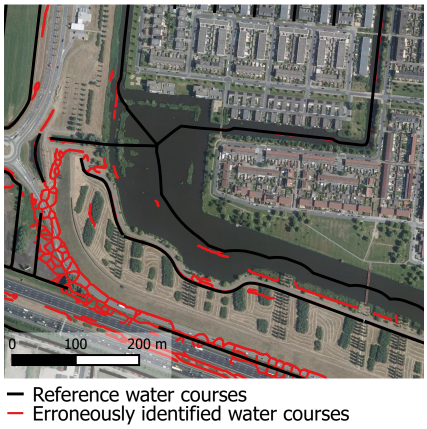
(a) Error of omission of the urban area for the concave hull.



(b) Error of commission of the urban area for the concave hull.



(c) Error of omission of the urban area for the MAT.



(d) Error of commission of the urban area for the concave hull.

Figure 5.9: The errors of omission and commission for the generated centre lines shown for the concave hull and MAT for the urban area (background aerial photo courtesy of PDOK).

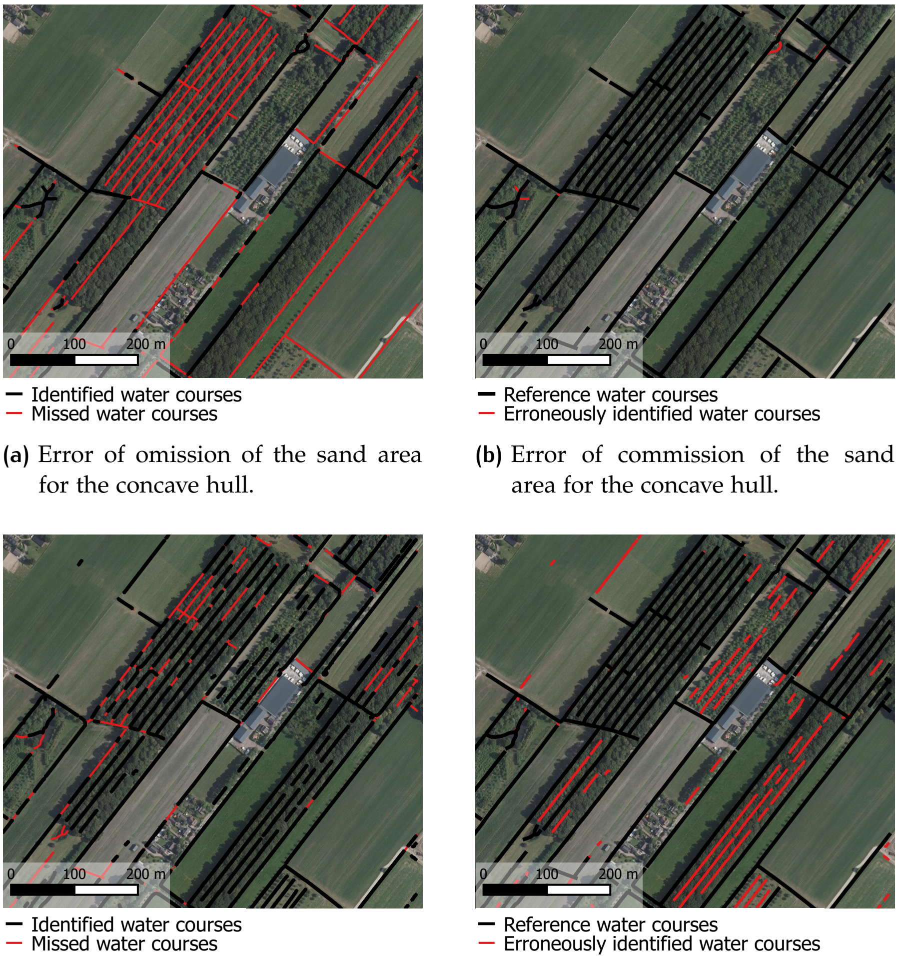
SAND The concave hull method does a poor job in identifying the water courses in the sand area (see Figure 5.10a), it finds just over 40% of all water courses. This is caused by the fact that water is not well visible in this landscape, water surfaces are often narrower than 1 m, and many patches of forest are present. The method performs somewhat better in terms of the water surface area which is identified (57%), but this difference in is mainly caused by the fact that the larger (thus more surface area) water courses in the area are correctly identified, which thus have a major weight in the computation of this percentage. The MAT does a better job (see Figure 5.10c) with more than 74% of water courses identified, but similarly struggles with the identification of the narrower water courses. The concave hull method displays very little error of commission (see Figure 5.10b), which is typical for this method in all the four areas (again, the error of commission in terms of water surface is higher with 15% since several large water courses weigh heavily on this percentage). The MAT shows an error of commission of more than 17%, however I estimate that at least half of these are not actually errors, they are water courses which are not present in the reference dataset (see Figure 5.10b), but are actually existing water courses. Combining the concave hull and MAT methods for this area only raises the identification rate of the water courses to roughly 76%, indicating that the MAT identified almost all of the water courses identified by the concave hull, and is clearly the better performing method (in the current implementation) for this area.

5.5.2 Positional accuracy

Average positional accuracy of the different methods is in the order of 0.5 to 1 m, which slightly varies per area. For regular, straight water courses, the positional accuracy is generally below or close to 0.5 m, but deviations are often larger near intersections. This is presumably also the reason that the positional accuracy is highest for the clay area, since this area is the most characterised by straight water courses. In general, the concave hulls and MAT methods agree quite well in terms of position, while the HDSR reference dataset often deviates from the two.

5.5.3 The quality of reference and input data: impact on error metrics

The analysis of the identification results in Section 5.5.1 has indicated that the reference dataset provided by the HDSR (see Section 5.2) does not always present the 'true' situation, it contains errors in the identification and positioning of water courses. Multiple situations were encountered where a water course was present in the reference dataset, which was clearly not present in the AHN3 data and the resulting generated datasets. These situations were encountered in each of the four areas (see Section 5.2), and an example is shown in Figure 5.11 for such a situation encountered in the urban area. Figure 5.11a shows that a water course was not identified, which is present in the reference dataset. However, this water course is also not present in the displayed background aerial photo of spring - summer 2014 (obtained from PDOK), and since both the AHN3 as well as the HDSR reference datasets stem from winter 2014 (thus a time discontinuity between the data is not of influence), this appears to be an identification error in the HDSR reference dataset. Interestingly, the water course is visible in an aerial photo of 2007 (obtained from Google Maps, see Figure 5.11b), which indicates that this may be an artefact which stems from an earlier version of the HDSR reference dataset. Of course, other identification errors may be present in the dataset, which are solely caused by the human interpreter. Another identified error of the HDSR reference dataset, is that some water courses identi-



(a) Error of omission of the sand area for the concave hull.

(b) Error of commission of the sand area for the concave hull.

(c) Error of omission of the sand area for the MAT.

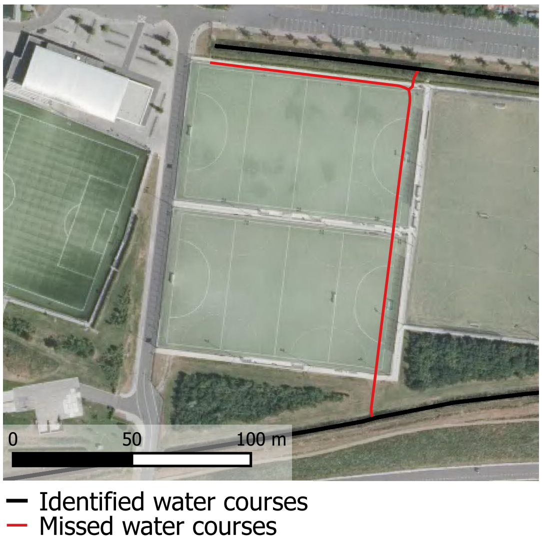
(d) Error of commission of the sand area for the MAT.

Figure 5.10: The errors of omission and commission for the generated centre lines shown for the concave hull and MAT for the sand area (background aerial photo courtesy of PDOK).

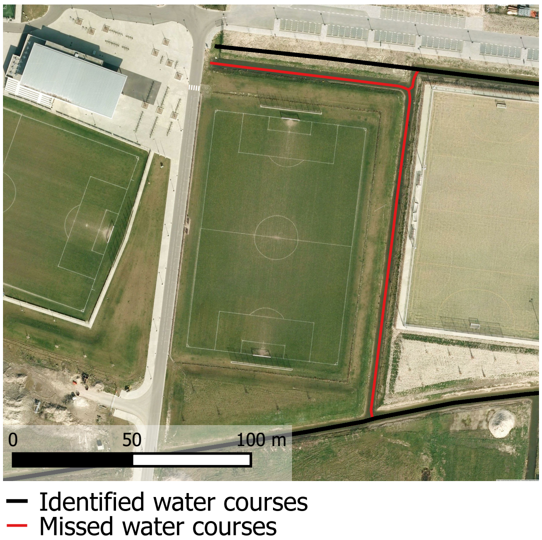
fied by the concave hull and MAT methods are not present in the reference dataset, while they clearly appear to be correctly identified water courses. These situations occur mostly in the sand area, of which Figure 5.10d shows an example. Here a large number of water courses are identified, which are not present in the reference dataset, but which based on shape and size clearly seem to be valid water courses. These water courses are hard to identify, since they are covered by forest, thus this likely indicates an incompleteness of the reference dataset.

Next to identification errors, also positional errors may be present in the reference dataset. Many cases were identified where, based on comparison to aerial photos, the generated products by the concave hull and MAT methods seemed to be closer to the 'true' situation than the HDSR reference dataset. However, in about equally many situations the HDSR datasets performed better than the generated datasets. As described in Section 5.5.2, the concave hull and MAT results agree well in most cases, while the HDSR dataset deviates from the two. This suggests that either the environmental conditions were different for the AHN3 as opposed to the HDSR reference datasets (for example a difference in water level, which can significantly change within days), or the subjectivity of the human interpreter of the HDSR reference datasets is of influence (thus the difference in procedures). The difference in input data and used procedures to create the reference and generated datasets, is also very clearly visible by comparing the metrics for the identification of water courses (Table 5.3), to the metrics for the identification of water surface area (Table 5.2). These computed values can differ significantly for the same area, which is caused by the fact that the metrics for the identification of water surface area are more sensitive to the differences between the methods. Although the centre line may be correctly identified (and thus the water course is identified), there may still be significant differences between the polygons of the water courses, and thus their surface areas.

Next to the problems described above, it also has to be taken into account that there is some specified inaccuracy in the AHN3 (see Section 1.1) and HDSR reference datasets (see Section 5.2), and my methods depend strongly on the correct classification of point in the AHN3. Concluding from the problems identified above, it can be said that there is no way of knowing which of the datasets represents the 'true' situations, without synchronized input data and solid ground truth. Thus, the computed error metrics should be used as only an indication of the 'true' mapping and positional accuracies of the designed methods.



(a) The error of omission with a background aerial photo of PDOK, stemming from 2014.



(b) The error of omission with a background aerial photo of Google Maps, stemming from 2007.

Figure 5.11: The error of omission shown for the centre lines generated by the concave hull for the urban area, displayed with two different background photos. This shows that the area has changed in the time period between the two photos.

6 REFLECTION ON THE PERFORMANCE AND IMPLEMENTATION OF THE CURRENT PROTOTYPE

Chapter 5 described the implementation of the methodology, and presented the results of the experiments. Based on these results, this chapter describes the strengths and weaknesses of the concave hull method in Section 6.1, and of the MAT method in Section 6.2. Additionally, these sections will give insight into why the methods performed as they did, and possibilities for future improvement of the implementations are suggested.

6.1 THE CONCAVE HULL METHOD

STRONG PERFORMANCE IN FLAT AND WATER ABUNDANT LANDSCAPES The concave hull method identifies water courses based on the presence of voids in the LiDAR measurements, caused by the absorption of the LiDAR signal by water, and adequate filtering of the remaining water points from the point cloud. Therefore, the method can operate completely independent of landscape topography; i.e. the presence or lack of topographic relief of the landscape is of no influence, as long as the water courses have significantly large water bodies. The ability of the method to identify water courses in virtually flat landscapes is showcased by its solid performance for the low-relief peat area (see Section 5.5.1), whereas the MAT suffers in this area from the lack of surface curvature of the water courses. Thus, the concave hull method is particularly suited for use in areas where relative water levels are high and water is a predominant feature of the landscape.

ROBUSTNESS TO ERRORS In rural areas, the concave hull method is characterized by low error of commission; the chance that a water courses identified by this method is not actually a water course in reality, is very small. This is due to the fact that the concave hull method makes use of the voids in the LiDAR measurements, and in rural areas there is only one major factor which induces such voids; the presence of water bodies. The removal of vegetation points from the dataset in the filtering procedure (see Step 1 of Section 5.3.1) can also induce such voids, but the resulting artefacts are largely removed by using an effective cleaning method (see Step 3 of Section 5.3.1). Furthermore, the error of omission of the concave hull method is equally low to the error of commission, for water abundant and low-vegetation areas. This is due to the fact that the procedure used to identify the water course polygons is relatively simple and concise (see Section 5.3.1), thus there is little which can go wrong in the identification procedure. In other words; if a significantly wide and uncovered water course is present in the landscape (granted that water points are adequately classified, and red laser is used for the LiDAR measurements), then it is highly likely that the concave hull method will correctly identify it. The robustness to errors is a major strength of the concave hull method, and makes it a safe solution for combination with other methods, since it adds few additional errors.

SENSITIVITY TO WATER SURFACE WIDTH The current implementation of the concave hull method is limited to the identification of water courses with a water surface width of at least 1 m, which is due to the fact that a value of 1 m was specified for the 'concavity' parameter (see Section 3.2.1 for an explanation of this parameter). As a consequence of this concavity value, if a water surface - thus the void in the point cloud - is narrower than 1 m, then it is not identified by the concave hull algorithm (see Section 3.2.1) as a void. Thus, the water course will not be identified in this case. In the sand area, water surfaces are not seldom less than 1 m wide (see Section 5.2), which partially explains why the method performs poorly for this area (see Section 5.5.1). Furthermore, this sensitivity also means that the concave hull method is very sensitive to the water level at time of LiDAR measurements. If water levels are low, then water surfaces may not be wide enough, or they may even be dry, and the water courses can not be identified.

SENSITIVITY TO VEGETATION COVERAGE The removal of vegetation points from the point cloud (see Step 1 of Section 4.1.1) can introduce artefacts into the concave hull dataset, which are effectively filtered using a cleaning method (see Step 3 of Section 4.1.1). This cleaning method works well when vegetation is present in small patches only (such as in the clay and peat area), but it causes problems when vegetation coverage is extensive (such as in the partially forested sand area, see Section 5.5.1). In these situations, the vegetation may cover entire water courses, which are accordingly removed from the dataset in the cleaning method. This presumably explains partially (next to the sensitivity to water surface width) the low performance of the concave hull method for the sand area, and indicates that the current implementation is not suited for use in areas with high vegetation coverage.

THE DIFFICULTY OF WATER COURSE IDENTIFICATION IN URBAN AREAS Urban environments display a wide variety of different types of water courses, which are very variable in their water surface width, vegetation coverage, and water levels. For the urban area, the error of commission by the concave hull method is around 17% (see Section 5.5.1). This has two reasons, the most important of which is that the branch pruning procedure (see Step 9 of Section 4.1.2) requires setting a threshold on the branch length. This works fine for rural environments, where water courses are relatively homogeneous in surface width, but poses problems for urban environments where water surface width varies significantly. For such variable water courses, part of the branches will be longer than the specified threshold length, and will thus not be removed from the dataset of centre lines, and are accordingly identified as errors of commission in the validation procedure (see Section 5.4.4). A second, somewhat less influential, factor that affects the error of commission in urban areas is the shadowing effect of tall buildings (which introduces voids into the LiDAR dataset). Lastly, the urban area has relatively many dry water courses, which are not identified by the concave hull method, and increase the error of omission for this area.

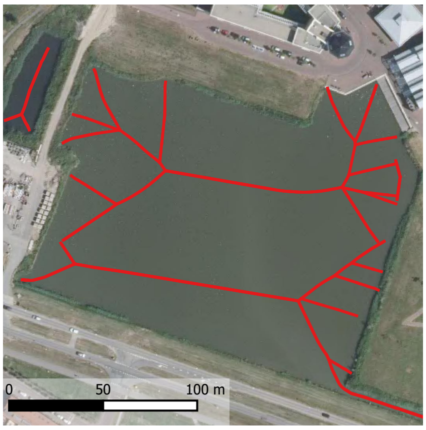
IMPROVING THE CURRENT IMPLEMENTATION OF THE CONCAVE HULL METHOD Based on the results in Section 5.5.1 and the discussion in this section, a number of possible future improvements to the current implementation of the concave hull method (see Section 5.3.1 and Section 5.3.2) are identified:

- *Lower concavity value:* This value controls the threshold distance for voids to be considered part of the exterior (Section 3.2.1). In the current implementation, this value is set to 1 m, which limits the identification of smaller water courses. I expect that lowering this threshold to sub-meter values will significantly increase identification of these smaller water courses, and will thus improve the metrics obtained for the sand area. However, lowering this value significantly increases the number of artefacts (e.g. those caused by vegetation, see Step 3 of Section 4.1.1), thus more extensive artefact filtering procedures will be required.

- *Filtering vegetation artefacts*: Currently, a filtering procedure removes vegetation-induced artefacts from the concave hull dataset (see Step 3 of Section 4.1.1). However, if water courses are entirely covered by vegetation, then they may be identified as artefacts and removed from the dataset. I suggest to extend the current filtering procedure by taking into account the typical shapes of water courses; they are typically extensive elongated features, which is not true (in the vast majority of cases) for vegetation artefacts (see e.g. Figure 4.3b). A decision rule should be formed based on this property, which decides whether to remove the artefact or not.

- *A dynamic threshold for branch removal*: The current method to generate and clean centre lines (see Section 4.1.2) works well for water courses in rural areas, which are relatively homogeneous in shape and size. However, as indicated by the error of commission for the urban area (see Section 5.5.1), the removal of unwanted branches from the centre lines does not currently work well for water courses with highly irregular size and shape. The problem is that a static threshold is used for branch removal (see Step 11 of Section 5.3.2), which is not sufficient when datasets contain variable-width water courses, since branch lengths generally become longer with water course width (see Figure 5.9b for an example of a water course with variable width, for which the unwanted branches are not automatically removed). This may be mitigated by using a dynamic threshold, which depends on water course width. Furthermore, the generation of a correct centre line is ambiguous for water bodies which are not shaped like a typical water course (e.g. for lakes, ponds such as Figure 6.1a). A solution may be to detect such shapes, and exclude them from centre line generation.

- *Shadowing effect of buildings*: In urban areas, the shadowing effect of buildings (i.e. they block the LiDAR signal in case of non-vertical incidence angles due to overhang) introduces additional voids into the dataset, which lead to vegetation-like artefacts (see Figure 4.3b), for which centre lines are generated (see Figure 6.1b). However, since there is no vegetation on these locations, the current artefact removal procedure (see Step 3 of Section 4.1.1) does not remove them. To filter these artefacts, concave hull could be generated of the building points in the AHN3 dataset. By slightly buffering the resulting polygons (e.g. a few m), and subsequently identifying which of the artefacts are contained within these polygons, the corresponding artefacts could be



(a) The generation of a correct centre line for non-elongated water bodies is ambiguous (background aerial photo courtesy of PDOK).

(b) The shadowing effect of buildings causes voids in the LiDAR data, leading to artefact centre lines (background aerial photo courtesy of Google Maps).

Figure 6.1: This figure shows two problems which commonly occur with the centre line generation by the concave hull, especially in urban areas.

removed. This is a relatively safe procedure, since water courses are typically located further away from buildings.

- *Artificial objects*: Artificial objects (e.g. bridges) are removed from the point cloud (see Step 1 in Section 4.1.1) since they obstruct water courses. In some cases, bridges are removed which do not cross water courses, but simply cross another road. This leaves additional voids in the point cloud, and thus leads to artefacts in the concave hull data, which are not filtered similarly to the previously mentioned shadowing effect artefacts. The centre lines generated for these artefacts have in common that they are parallel to the removed objects, while the centre lines of water courses are by definition not parallel to the objects (e.g. bridges) that cross them. This can be a simple, yet effective, characteristic to the artefact centre lines.

6.2 THE MAT METHOD

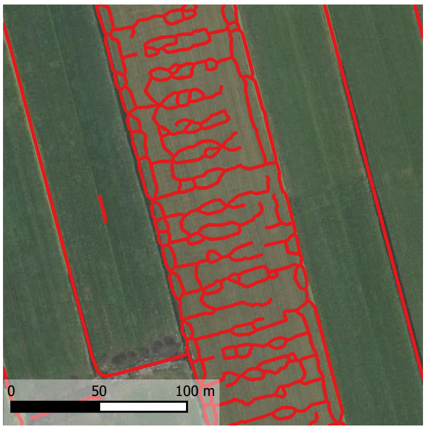
SENSITIVITY TO SURFACE CURVATURE While the MAT does not specifically compute any surface curvature values, it does depend on this landscape characteristic through the size of its medial balls (see Section 3.3). Thus, if the landscape is flat, and has no curved features, then no skeleton of the landscape can be derived. Similarly, if the banks of the water courses in the landscape show no or very little curvature, then the MAT will be unable to identify these channels. This fact explains why the MAT performs better for the clay area than for peat, although both areas are similar with respect to the dominant role of water in the landscape. In the clay area, the water courses have a clear concave profile, while this profile is much less distinguishable in the peat area (see Section 5.2). The identification of water courses in the sand area is problematic for similar reasons. In the sand area, water courses are much less dominant landscape features than in the clay and peat areas; there are a few larger water courses which display a pronounced curvature, but the majority of water courses here are so narrow that there are not enough LiDAR points on the surface banks to approximate the MAT from (this is of course also caused by the current point spacing of the AHN3). It can be said that the clay area forms a nearly ideal environment for the identification of water courses by the MAT, with clear concave profiles and enough LiDAR points on the surface banks. Furthermore, the concave features in this area are almost all water courses, so there is relatively little pollution from other sources (this would be different in high-relief terrain). An advantage of the dependence on surface curvature is the ability to detect both wet and dry water courses, since surface curvature is theoretically independent from water level (unless the water course is entirely flooded, covering the banks). This feature is a very strong point of the MAT method, which distinguishes it from the concave hull method, and makes it a valuable addition.

INSENSITIVITY TO VOIDS IN THE POINT CLOUD Contrary to the concave hull method, the MAT method is not sensitive to voids in the input LiDAR point cloud. The input dataset is allowed to contain voids, which makes it possible to filter out any unwanted classes such as buildings or vegetation. Consequently, the MAT can theoretically perform equally well in vegetated areas, as it can in areas without vegetation. This is also true in practice, as long as the canopy is not too dense and allows some returns of the LiDAR signal. If the vegetation coverage is too dense, then the water course banks can be devoid of LiDAR measurements, and the water courses cannot be identified. However, such situations are not wide spread, and in general the insensitivity of the MAT to vegetation coverage can be presented as a strong point.

PRONE TO ERRORS The sensitivity of the MAT method to surface curvature allows it to identify (dry) water courses, but this also makes it prone to identification errors. Water courses generally have a concave profile, which is reflected in the skeleton of the landscape. However, there can also be other features in the landscape with a concave profile, which are not necessarily part of the drainage network (e.g. small inundations or rows of crops such as potatoes see Figure 6.2a). Furthermore, convex features such as levees or piles of earth or dirt are also present in the landscape. All these examples introduce errors into the identification results, leading to such features being identified as water courses (see Figure 6.2, while they are different features in reality. Furthermore, the current implementation of the MAT method is quite convoluted; e.g. relatively many procedures are required (see Section 5.3.3) to identify the centre lines from the input AHN3 point cloud. This further heightens the chance of error inclusion, since there are simply more processes where an error can be introduced.

THE DIFFICULTY OF WATER COURSE IDENTIFICATION IN URBAN AREAS Similarly to the disjoint hull method, also the MAT has a high error of commission in urban environments (see Section 5.5.1). Urban areas are especially susceptible to noise in the MAT, due to the human modifications of the original landscape, which introduces many local convexities and concavities (which are not all water courses). Furthermore, many of the water courses are asymmetrical, and the edge selection algorithm (explained in Step 5 of Section 4.2) does not perform well in these cases (see Figure 6.3 and Figure 6.4), leading to very messy centre lines and many errors of commission. Next to

(a) Rows of freshly planted potatoes, which add local relief to the terrain (image courtesy of Hilda Schuitema, fotoo.nl).



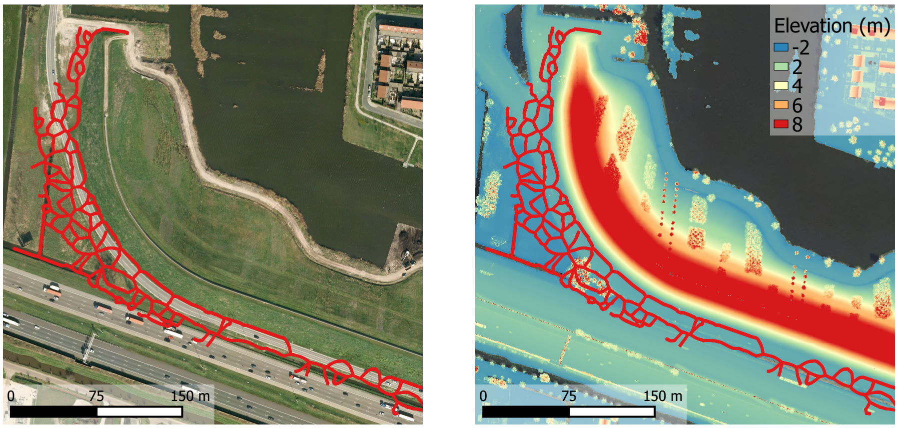
(b) Artefact centre lines caused by local relief (background image courtesy of PDOK).

Figure 6.2: Local forms of topographic relief, for example caused by farming activities, can potentially lead to artefacts in the set of centre lines generated by the MAT.

a high error of commission, the MAT method also shows are relatively high error of omission for the urban area. Water courses in urban areas are very variable with respect to their surface width, which is also true for the nature of their banks. In many cases, houses border the water courses, replacing any form of a natural bank. Furthermore, these water courses can be very wide and sometimes have vertical banks. The shrinking ball algorithm (see Section 3.3.1) then returns a very patchy or even no medial axis, thus the water course are not identified, causing errors of omission.

IMPROVING THE CURRENT IMPLEMENTATION OF THE MAT Similarly to the concave hull method, based on the results in Section 5.5.1 and the discussion in this section, a number of possible future improvements to the current implementation of the MAT method (see Section 5.3.3) are identified:

- *Improving noise filtering:* The current MAT implementation performs an artefact filtering procedure based on points density (see Step 4 in Section 4.2). However, as indicated earlier; although the landscape in the tested areas (see Section 5.2) is generally low-relief, there are still irregularities present in the form of local concavities or convexities (e.g. levees, inundations). The current MAT implementation does not foresee in filtering artefacts caused by such irregularities. This leads to high errors of commission compared to the concave hull method, which is especially true for the urban area. Part of these artefacts are caused by convex features. Since water courses are by definition concave features, a possibility would be to detect convex features, and prevent the MAT from approximating the medial axis for such features. A second procedure should be implemented which detects local concavities which do not have the typical shape of a water course, which can then also be filtered out. The two procedures together should filter a good deal of the generated artefacts, and error of commission should be closer to the concave hull values.



(a) A messy centre line, which crosses many features such as roads (background aerial photo courtesy of Google Maps).

(b) The centre line is generated next to a high convex feature on one side of the water course, which thus has an asymmetrical cross section (shown is a gridded version of the AHN3 DTM).

Figure 6.3: The current implementation of the MAT sometimes generates very messy centre lines for asymmetrical water courses.

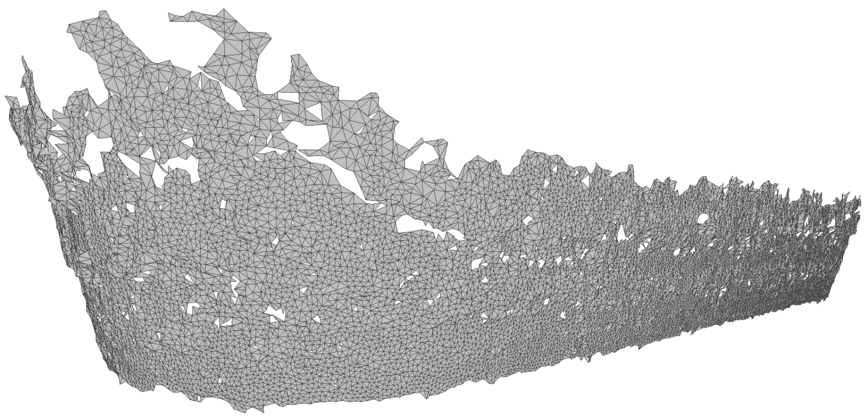


Figure 6.4: The asymmetrical water course shown in Figure 6.3, leads to the generation of a large, tilted medial sheet. Furthermore, the sheet has a patchy distribution of medial points, which leads to holes in the corresponding triangulation. The edge selection algorithm (explained in Step 5 of Section 4.2) extracts all the lower edges, which includes the lower edges in these holes, which explains the messy centre line generated in Figure 6.3.

- *Improving water course segmentation:* The current implementation of the medial axis segmentation (see Step 3 of Section 5.3.3) performs well in most situations, but not when water courses have less pronounced concavity. In these cases, the medial axis can be patchy, and the segmentation algorithm segments these into many separate sheets. Sometimes this results in the creation of many very small segments, which are currently automatically filtered out (see Step 4 in Section 5.3.3). Ideally, these segments should be joined so that they are not filtered out, and can still be used to represent parts of the water courses[1]. Additionally, medial sheets that are not parallel to each other, or do not belong to the same water course, should be represented by separate segments. This additional segmentation rule would decrease the need for a second segmentation procedure as is used now (with different segmentation parameters, see Step 3 of Section 5.3.3). This leads to a less convoluted implementation, and decreases processing time.

- *Triangulation and edge selection of medial sheets with gaps*: The BPA (see Section 3.3.3) currently used for the triangulation of medial sheets works well in most cases, but fails to perform when sheets need to be triangulated which contain gaps larger than the size of the pivoting ball. In such cases, only a segment of the sheet is triangulated (see Figure 6.5), and thus parts of the water course centre lines are not generated. Currently, the implementation uses the BPA implementation of MeshLab (see Section 5.1.2). My suggestion is to rewrite this algorithm, either in MeshLab, or a different environment, and enable it to work with such gaps. The method should detect these gaps[2] and generate triangulations for the sub segments separately.

  Another problem with gaps occurs with the current implementation of the edge selection algorithm (see Step 5 in Section 4.2). centre lines are extracted from the triangulation by selecting all the lower edges. This works well in case of straight water courses with vertical medial sheets and closed triangulations. However, some water courses are represented by large tilted medial sheets with sparse and patchy distribution of points. The triangulation of such sheets can have interior holes (see Figure 6.3 and Figure 6.4), the edges of which are also selected by the edge extraction algorithm. Especially for the urban area this happens often, and this introduces many artefact centre lines and raises the error of commission significantly. Such interior holes should be excluded by the selection algorithm, generating a much cleaner dataset.

1 This is already possible to some extent in the current segmentation script, but is not yet applied in this thesis.
2 The detection of these gaps could be done based on properties of the MAT, such as the medial bisector.

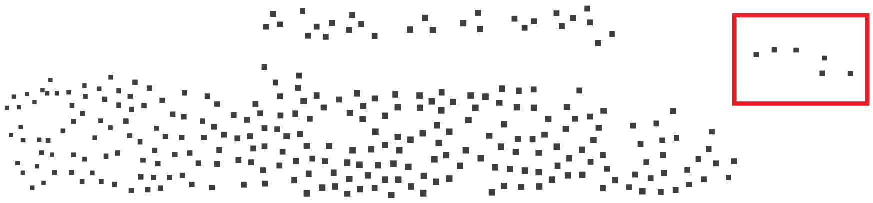


Figure 6.5: This figure shows a medial sheet for a water course, which contains gaps between groups of points. The BPA implemented in MeshLab (see Section 5.1.2) first selects a 'seed triangle', from which it performs the triangulation. But if the seed triangle is in the group of points in the red rectangle, and the gap between these groups is larger than the specified ball radius, then the algorithm only triangulates the points in the red rectangle.

7 CONCLUSION AND RECOMMENDATIONS

In this thesis, I investigated the possibilities of automatically identifying water courses in flat and engineered landscapes, using the raw (albeit classified) LiDAR points of the AHN3 dataset. I found that there are many methods described in literature which identify channel-like features (see Section 2.1), and some which identify water courses in engineered landscapes (see Section 2.3), but none of these are suitable for this application. Thus, this identifies the need for the development of a new method. I formulated the hypothesis (see Section 1.5) that a combination of two concepts would form a robust methodology for the automatic identification of water courses from AHN3 in the flat and engineered landscapes around Utrecht; (1) the concept of concave hulls (see Section 3.2), and (2) the MAT (see Section 3.3). A workflow was implemented which takes an AHN3 LiDAR point cloud as input, and accordingly identifies for every water course the polygons of the water surface, and the geographical position of the water surface's centre lines. The implemented prototype was used for different areas to test its applicability to different environments. The resulting datasets were validated to obtain mapping and positional accuracies. The following sections conclude the most important findings of this thesis. Section 7.1 answers the research questions posed in Section 1.6, and thereby describes the potential of the designed prototype for the application of this thesis. Subsequently, Section 7.2 states what the scientific value of the research performed in thesis is, and how it contributes to the scientific community. Lastly, Section 7.3 presents possibilities for future work on the designed prototype.

7.1 THE POTENTIAL OF THE DESIGNED PROTOTYPE

This section answers the research questions posed in Section 1.6, starting from the sub-questions, and ending with the main research question. Through answering these questions, the potential of the designed prototype for the automatic identification of water courses in flat and engineered landscapes from AHN3, will be made clear.

What are the typical properties of water courses in the flat and engineered landscapes around Utrecht, that make them different from the rest of the landscape?

The water courses around Utrecht have in common that they are almost entirely artificial, and thus mostly have regular shapes. In rural areas, the network of water courses consists of smaller -often regularly spaced- ditches intersecting the meadows, which eventually drain into larger canals and rivers. In urban centers, sewers represent a large part of the drainage network, and water courses are mostly present in the form of canals. Furthermore, since these are low-elevation areas, water is widely present in these landscapes, and water courses are often -if not permanently- filled with water. Three essential characteristics were identified, which can be used to discern the water courses from the rest of the landscape:

1. *Low elevation in the landscape*: Since the landscapes show very little relief, it is plausible that the water courses are the landscape features with the lowest elevation.

2. *Concave profiles*: Since almost all of the water courses are artificial, especially the regularly spaced ditches in rural areas have distinct concave profiles.

3. *Presence of water*: Due to the low elevation of the landscape, and constant regulation of water levels, it is likely that many of the water courses always contain water.

How (well) are these typical properties reflected in the AHN3 dataset?
In LiDAR point clouds such as the AHN3, the low elevation of water courses in the landscape, and their concave profiles are well reflected, since elevation values of point measurements are recorded. Furthermore, the presence of water is reflected in the AHN3 dataset as voids in the data, since the red laser signal is almost entirely absorbed by water, thus no -or a very weak- reflected signal reaches the sensor. Since the AHN3 provides a classification of points, amongst which also a water category, any signal which does reflect on the water surface can be filtered out.

In practice, only the concave profiles and presence of water can be effectively used to discern water courses from the rest of the landscape. In these human-engineered landscapes, it is very possible that water levels are artificially kept higher in some parts than in others, thus water courses may not always be the lowest feature in the landscape. Thus, the identification of water courses can be done based on their concave profiles or presence of water.

Which methods can use these properties to identify the water courses in the AHN3 dataset?
Two concepts were identified which each utilize one of these properties: (1) the concave hull, and (2) the MAT. Both concepts can use raw (albeit classified) LiDAR point clouds as input. The concave hull approach makes use of the presence of water in the water courses, and should thus be able to identify water courses with a sufficiently large water surface, irrespective of their degree of concavity. In contrast, the MAT uses the concave profiles of the water courses to identify them, and can work irrespective of water presence, and should thus additionally be able to identify dry water courses. I proposed the hypothesis that a combination of these concepts can form a robust methodology for the identification of water courses from AHN3 in the flat and engineered landscapes around Utrecht.

How can the polygons of the water surfaces, and the water course centre lines be extracted from the AHN3 using these methods?
The polygons can be extracted by using the concave hull method, with the following workflow; First, the input point cloud is filtered, creating two separate point clouds: (1) a point cloud of ground surface and building points, and (2) a point cloud with only vegetation points. The concave hulls for both datasets are then generated, after which artefacts in the dataset are removed by selecting all artefacts which are contained within the concave

hulls generated for the vegetation points dataset, and subsequently dissolving these. This results in a relatively clean shapefile with polygons of the concave hulls of the ground surface points. In this dataset, most of the water course are represented by the space in between the concave hull, but some are contained in the shapes as interior polygons. All these interior polygons are then removed, and the inverse of the dataset is taken to obtain a shapefile of water course polygons (which thus represent the polygons of the water courses). These polygons are additionally smoothened to remove irregularities.

The water course centre lines can be extracted by using the following workflows:

1. *concave hull*: This procedure follows directly on the procedure to identify the water course polygons. The dataset of water course polygons is converted to points, subsequently densified by inserting additional points on the polygons, after which a VD is created on these points. All edges of the VD that are within the water course polygons are then selected and extracted to a new shapefile. These form the centre lines of the water courses. Any unwanted branches on these centre lines are pruned, to obtain a relatively clean dataset of water course centre lines.

2. *MAT*: The input point cloud is filtered, only leaving ground surface points. The outer medial axis of the landscape is then approximated, after which it is segmented into separate medial sheets based on the medial bisector difference. An octree is then constructed on the points in the medial sheets, which subsequently is used to remove artefacts from the sheets based on point density. Additionally, the smallest sheets are removed, since they likely represent artefacts. From each of the remaining sheets, a uniform subset of points is then extracted, which are subsequently triangulated. The edges on the lower half of the triangulated sheets are then extracted, and written to a new shapefile. These represent the water course centre lines.

Both workflows are used, which leads to the creation of two datasets of centre lines for every input point cloud. These centre line datasets are combined by merging the shapefiles, and accordingly buffering all lines in the dataset. The buffers are then dissolved, which results in one polygon for every water course. Then, again a VD approach (similarly to the concave hull) is used to extract the centre lines.

How well do the extracted datasets compare to the reference data?

The concave hull method identifies approximately 95% of all water courses for the clay and peat area, 90% for the urban area, and 42% for the sand area. The error of commission is 1% for the clay area, 2% for the peat area, 17% for the urban area, and 4% for the sand area. The results are somewhat different for the extent to which the method identified the water surface areas, which is due to the fact that the computed surface area is more sensitive to differences in the methodology of water course identification (see Section 5.5.3). The concave hull method manages to identify 87% of all water surface area for the clay area, 90% for the peat area, 94% for the urban area, but only 57% for the sand area. Also for the water surface area, the error of commission is relatively low, with 6% for the clay area, 8% for the peat area, 11% for the urban area, and 15% for the sand area. The MAT method identifies 96% of all water courses for the clay area, 85% for the peat and urban area, and 74% for the sand area. The error of commission is 8% for the clay and peat area, 46% for the urban area, and 17% for the sand area. As can be seen, both methods generate many artefacts for the urban area, and have difficulties identifying the water courses in the sand area. When the methods are combined, they manage to identify 98% of all water courses for the clay area, 97% for the peat area, 95% for the urban area, and 76% for the sand area. Clearly, the identification rates profit from the combination of methods. However, the relatively high error of commission of the MAT also radiates into the combined method. The error of commission is then 8% for the clay and peat area, 47% for the urban area, and 17% for the sand area.

To what extent can the approach be fully automated?

The current implementation is not fully automated, although relatively little manual action is required to generate the datasets. However, the methodology was designed with full automation in mind, and both concepts (the concave hull and MAT) can in theory be fully automated when developed further. In practice, some manual calibration of parameters will likely be required when applying the method to a new area.

To what extent can the position and planimetric geometry of the drainage networks in flat, engineered landscapes be automatically identified from the AHN3 LiDAR point cloud?

The experiments and analysis performed in this thesis have successfully shown the potential of the presented prototype for the automatic identification of water courses from AHN3. Both methods included in this prototype, the concave hull and MAT, proved to be important components. The concave hull method can operate independently of landscape topography, and depends only on the presence of water in the water courses (and on a solid classification to filter water points). Furthermore, it is very robust to errors in the identification; there are relatively few errors of commission. However, the method is sensitive to vegetation coverage and water surface width (the current implementation does not perform well for water courses less than 1 m wide), and cannot identify dry water courses. Thus, the concave hull method is particularly suited for use in areas where relative water levels are high, water courses are wide, and vegetation coverage is low. Due to the robustness to errors, this is a relatively safe method to combine with other methods.

Contrary to the concave hull method, The MAT method depends on landscape topography; it identifies water courses by their concave surface curvature. This has the advantage that the MAT is able to operate well when water levels are low, or even when water courses are dry. Furthermore, it is relatively insensitive to vegetation coverage, and can theoretically operate even in lightly forested areas. However, the MAT does not perform well for water courses which show little surface curvature, and is prone to errors of commission caused by local non-watercourse convexities and concavities. Both methods currently do not perform well for the generation of water course centre lines in the urban and sand area. For the urban area, both methods generate relatively many artefacts (thus errors of commission), but improvements have been suggested which may well mitigate the generation of these artefacts substantially. Furthermore, the concave hull method performs well in the urban area with respect to the identification of water surface area.

Similarly to the urban area, improvements are also proposed for the sand area, but I expect that the identification of water courses here will remain difficult (unless point density in a future version of the AHN is improved) since they are relatively narrow and thus difficult to identify.

Through the combination of the strengths of both methods, a more robust approach is obtained than could be achieved for any of the methods separately, since part of the weaknesses of both methods are mitigated. The combined prototype provides a strong and promising approach for the automatic identification of water courses in flat and engineered landscapes from the raw AHN3 point cloud. In its current form, it is able to identify above 95% of all water course centre lines for the clay, peat, and urban areas, and above 75% of all water courses for the sand area. Though the methods currently require a small amount of calibration when applied to new areas, they can in principle be fully automated. It has to be stressed however, that the current implementation does not do any form of point classification, thus it is strongly dependent on the extent to which the classification of the input point cloud was performed accurately.

7.2 THE SCIENTIFIC VALUE AND CONTRIBUTIONS OF THIS THESIS

The research performed in this thesis successfully proved the validity of my proposed hypothesis; a combination of the concave hull and MAT concepts forms a robust methodology for the identification of water courses from AHN3 in the flat and engineered landscapes around Utrecht. The design I implemented performs favourably compared to GeoNet [Passalacqua et al., 2010, 2012], which shows poor results in the low-relief peat area. In contrast to most solutions presented in the present day literature (see Chapter 2), my design does not require the troublesome generation of a gridded DEM, and does not suffer from any decrease in accuracy due to the associated conversion and interpolation processes of LiDAR points. To the contrary, my design uses the most detailed form of input data possible for this application, which is the raw data in the AHN3 LiDAR point cloud. Furthermore, the design does not require the interference of a human operator, and is thus very objective compared to manual procedures such as those used by the HDSR. The present day scientific literature provides no other suitable solutions for the identification of water courses in flat and engineered landscapes using raw LiDAR data. Thus, the methods designed in this thesis fill a scientific gap, and thereby provide a valuable contribution to the scientific community.

Additionally, I add that both Höfle et al. [2009] and Toscano et al. [2014] try to identify hydro break lines, but require relatively complicated procedures. The methodology which I developed around the concept of concave hull, provides a simple way to identify these hydro break lines, which can be defined as an additional scientific contribution of this thesis.

7.3 RECOMMENDATIONS FOR FUTURE WORK

Section 6.1 and Section 6.2 already listed and described a number of possible improvements to the current implementations of the concave hull and MAT methods, which can potentially lower the error of omission and commission for both. This section does not repeat these improvements, rather a number of additional suggestions for future work are provided, which could extend the designed prototype with new functionalities and can enable it to be used for different applications and other locations.

AUTOMATION AND OPTIMISATION An obvious extension to the current implementation would be to further automate the required procedures. Ideally, the prototype should be automated to such an extent that only the specification of an input point cloud is sufficient, after which the required products are generated automatically. Additionally, the procedures should be optimised. Currently, the implementation relies heavily on the use of procedures in LAStools, QGIS, CloudCompare, and MeshLab. This was convenient for the development of the prototype since it saves development time, but also this provides less control over the algorithms used, and is less optimised in terms of speed. Writing the required procedures by hand can allow further optimisation and automation. This eventually can make the prototype ready for use to other, less specialised users.

APPLICATION TO DIFFERENT ENVIRONMENTS The prototype was tested only for the use in the flat and engineered landscapes around Utrecht, but it would be interesting to see how the method performs in other environments. These methods may perform equally well in engineered landscapes with more relief, and I think they also have potential for the identification of streams in natural environments.

DIFFERENT POINT CLOUD DENSITIES To test the limits of both methodologies, the implementation should be tested with other point clouds of different densities. I assume the methods will also work for less dense point clouds, albeit only for the larger water courses. Similarly, using a denser point cloud may significantly improve the ability for identification of smaller (i.e. narrower) water courses.

GENERATION OF HYDRO BREAK LINES Höfle et al. [2009] and Toscano et al. [2014] identify hydro break lines, but require relatively complicated procedures. The concave hull method presented here provides a relatively simple solution to generate these hydro break lines. It would be interesting to know how well this method performs compared to Höfle et al. [2009] and Toscano et al. [2014], and whether it can accordingly be used to generate more accurate gridded DEMs.

TOPOLOGICAL REPAIR The topic of topological repair was considered outside the scope of this thesis, but it is an important topic nonetheless. The current prototype generates datasets which are unoptimised in terms of topology; i.e. there can be gaps in the network of centre lines, lines may not be connected at intersections, and there may be lines which are completely disconnected from the network. A form of edge snapping is performed for the MAT in Step 5 of Section 5.3.3, since two datasets are combined by buffered the centre lines, which additionally closes small gaps in the centre lines. However, no procedure was implemented with the specific goal of topological repair in mind. Ideally, the water course identification should result in a connected network of water courses, which can then be used for hydrologic modelling purposes. In reality, water courses are not always continuous, they are often intersected by land bridges, allowing farmers to cross to adjacent meadows. These water courses may or may not be connected underground by tubes. Edge snapping in such situations may introduce more error into the dataset.Thus, the process of topological repair in water course networks is not simple, since it is often ambiguous whether two disconnected lines should be connected or not. However, it is an important topic, and is therefore recommended as future work.

AESTHETIC ENHANCEMENT The generation of an aesthetically attractive dataset of water course polygons and centre lines is important when these datasets are used for visual display purposes. Such visual enhancement of the datasets is in the current implementation only performed for the water course polygons, in a light smoothening procedure (see Step 6 of Section 4.1.1). This process could be improved. Furthermore, the generated centre lines appear as jagged lines when zoomed in, which is due to the use of VD's for centre line extraction in the concave hull (see Step 7 of Section 4.1.2), MAT method (see Step 5 of Section 5.3.3), and combined method (see Section 5.3.4). These lines should be straightened to improve visual attractiveness.

USING THE MAT TO IDENTIFY THE 3D GEOMETRY OF WATER COURSES The MAT is a potentially very versatile method, which is not nearly used to its fullest extent in this thesis. The method uses the points reflected on the landscape to form medial balls which approximate the medial axis, and the points which are used to form the medial balls, and associated medial points, can be stored. This is potentially very interesting, since if we know that a certain collection of medial points represents the medial sheet of a water course, then it is possible to reconstruct the banks of this water course through the earlier storage of the points. In this way, a 3D geometry of the water course (above the water surface) could be obtained. Such a 3D geometry can be useful for the estimation of storage capacity in the water course, but can also be used to derive other products from. For example, it could be used to extract the polygons and centre lines of the water surface for any specific water level, or to generate cross sectional profiles (of the part of the water course above the water surface at the time of measurement). In my opinion, this is the most interesting topic of future work, and I recommend this topic to be further explored. At the time of writing, there is not yet any method available in scientific literature which manages to extract 3D geometries of water courses, and it could thus provide a solid contribution to the scientific community.

MANUAL COLLECTION OF REFERENCE DATA The current verification procedure of the datasets generated by the prototype implementation developed in this thesis, is based on using reference datasets supplied by the HDSR (see Section 5.2). As described in Section 5.5.3, the validity of these datasets is questionable in many cases, thus it would be good to test the designed implementation with other reference data. This could be done by manual generation of such reference datasets from aerial photography, or by collecting ground truth in the field. Although these procedures can be labour intensive, they may be the only way to accurately test the validity of the generated datasets.

BIBLIOGRAPHY

Aichholzer, O., Aurenhammer, F., Alberts, D., and Gärtner, B. (1995). A novel type of skeleton for polygons. *Journal of Universal Computer Science*, 1(12):752–761.

Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. (2003). Computing and rendering point set surfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 9(1):3–15.

Aurenhammer, F. (1991). Voronoi diagrams – a survey of a fundamental geometric data structure. *AACM Computing Surveys (CSUR)*, 23(3):345–405.

Bailly, J. S., Lagacherie, P., Millier, C., Puech, C., and Kosuth, P. (2008). Agrarian landscapes linear features detection from lidar: application to artificial drainage networks. *International Journal of Remote Sensing*, 29(12):3489–3508.

Bailly, J. S., Levavasseur, F., and Lagacherie, P. (2011). A spatial stochastic algorithm to reconstruct artificial drainage networks from incomplete network delineations. *International Journal of Applied Earth Observation and Geoinformation*, 13(6):853–862.

Baruch, A. and Filin, S. (2011). Detection of gullies in roughly textured terrain using airborne laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(5):564–578.

Bernardini, F., Mittleman, J., Rishmeier, H., Silva, C., and Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):349–359.

Bouldin, J., Farris, J., Moore, M., and Cooper, C. (2004). Vegetative and structural characteristics of agricultural drainages in the mississippi delta landscapes. *Environmental Pollution*, 132(3):403 – 411.

Boyle, A. (1970). The quantised line. *The Cartographical Journal*, 7(2):91–94.

Broersen, T., Fichtner, F. W., Heeres, E. J., de Liefde, I., Rodenberg, O. B. P. M., Verbree, E., and Voûte, R. (2016). Project pointless. identifying, visualising and pathfinding through empty space in interior point clouds using an octree approach. In *AGILE 2016; 19th AGILE Conference on Geographic Information Science, 14-17 June, 2016; Authors version*.

Brzank, A., Göpfert, J., and Lohmann, P. (2005). Aspects of lidar processing in coastal areas. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36 (Part 1/W3):6.

Brzank, A., Heipke, C., Goepfert, J., and Soergel, U. (2008). Aspects of generating precise digital terrain models in the wadden sea from lidar-water classification and structure line extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(5):510–528.

Cavalli, M., Trevisani, S., Goldin, B., Mion, E., Crema, S., and Valentinotti, R. (2013). Semi-automatic derivation of channel network from a high-resolution dtm: the example of an italian alpine region. *European Journal of Remote Sensing*, 46:152–174.

Cazorzi, F., Fontana, G. D., De Luca, A., Sofia, G., and Tarolli, P. (2013). Drainage network detection and assessment of network storage capacity in agrarian landscapes. *Hydrological Processes*, 27:541–553.

Charaniya, A., Manduchi, R., and Lodha, S. (2004). Supervised parametric classification of aerial lidar data. In *CVPRW'04, Proceedings of the IEEE 2004 Conference on Computer Vision and Pattern Recognition Workshop*, volume 3, pages 1–8. Baltimore, Md.

Cho, H., Slatton, K. C., Krekeler, C. R., and Cheung, S. (2011). Morphology-based approaches for detecting stream channels from alsm data. *International Journal of Remote Sensing*, 32(24):9571–9597.

Chorowicz, J., Ichoku, C., Riazanoff, S., Kim, Y., and Cervelle, B. (1992). A combined algorithm for automated drainage network extraction. *Water Resources Research*, 28(5):1293–1302.

Costa-Cabral, M. C. and Burges, S. J. (1994). Digital elevation model nnetwork (demon): A model of flow over hillslopes for computation of contributing and dispersal areas. *Water Resources Research*, 30(6):1681–1692.

de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O. (2000). *Computational Geometry: AAlgorithm and Applications*, volume Third Edition. Springer.

Deng, J. and Han, Y., editors (2013). *A real-time system of lane detection and tracking based on optimized RANSAC B-spline fitting*, RACS '13 Proceedings of the 2013 Research in Adaptive and Convergent Systems. ACM New York, NY, USA.

Douglas, D. and Peucker, T. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122.

Duckham, M., Kulik, L., Worboys, M., and Galton, A. (2008). Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10):3224–3236.

Edelsbrunner, H., Kirkpatrick, D., and Seidel, R. (1983). On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, IT-29(4):551–558.

Fabio, R. (2003). From point cloud to surface: the modmodel and visualization problem. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(5):W10.

Fagherazzi, S., Bortoluzzi, A., Dietrich, W. E., Adami, A., Lanzoni, S., Marani, M., and Rinaldo, A. (1999). Tidal nnetwork 1. automatic network extraction and preliminary scaling features from digital terrain maps. *Water Resources Research*, 35(12):3891–3904.

Fisher, P. F. (1997). The pixel: A snare and a delusion. *International Journal of Remote Sensing*, 18(3):679–685.

Flood, M. and Gutelius, B. (1997). Commercial implications of topographic terrain mapping using scanning airborne laser radar. *Photogrammetric Engineering and Remote Sensing*, 63(4):327–366.

Galton, A. and Duckham, M. (2006). What is the region occupied by a set of ponts? *GIScience*, pages 81–98.

Gandolfi, C. and Bischetti, G. B. (1997). Influence of the drainage network identification method on geomorphological properties and hydrological response. *Hydrological Processes*, 11:353–375.

Gold, C. (1999). Crust and anti-crust: A one-step boundary and skeleton extraction algorithm, acm. *Proceedings of the fifteenth annual sumposium on Computational geometry*, pages 189–196.

Gold, C. M. and Edwards, G. (1992). The Voronoi spatial model: Two- and three-dimensional applications in image analysis. *ITC Journal*, 1:11–19.

Haunert, J.-H. (2008). *Aggregation in Map Generalisation by Combinatorial Optimization*. Number 626 in Reihe C. Deutsche Geod"atische Kommission, München.

Held, M. (2001). vroni: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. *Computational Geometry—Theory and Applications*, 18(2):95–123.

Hill, J., Graham, L., and Henry, R. (2000). Wide-area topographic mapping and applications using airborne light detection and ranging (lidar) technology. *Photogrammetric Engineering and Remote Sensing*, 66(8):908–914.

Höfle, B., Vetter, M., Pfeifer, N., Mandlburger, G., and Stötter, J. (2009). Water surface mapping from airborne laser scanning using signal intensity and elevation data. *Earth Surface Processes and Landforms*, 34:1635–1649.

Jenson, S. K. and Domingue, J. O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Protogrammetric Engineering & Remote Sensing*, 54(11):1593–1600.

Lee, R. and Fredricks, D. (1984). Intersection orf parametric surfaces and a plane. *IEEE Computer Graphics and Applications*, 4(8):48–51.

Liao, S. and Bai, Y. (2010). A new grid-cell-based method for error evaluation of vector-to-raster conversion. *Computational Geosciences*, 14(4):539–549.

Lillesand, T., Kiefer, R., and Chipman, J. (2008). *Remote Sensing and Image Interpretation*. John Wiley & Sons, Ltd, 6th edition edition.

Liu, Y., Zhou, M., Zhao, S., Zhan, W., Yang, K., and Li, M. (2015). Automated extraction of tidal creeks from airborne laser altimetry data. *Journal of Hydrology*, 527:1006–1020.

Lohani, B. and Mason, D. C. (2001). Application of airborne scanning laser altimetry to the study of tidal channel geomorphology. *ISPRS Journal of Photogrammetry and Remote Sensing*, 56:100–120.

Ma, J., Bae, S., and Choi, S. (2012). 3d medial axis point approximation using nearest neighbors and the normal field. *The Visual Computer*, 28(1):7–19.

Malano, H. and Hofwegen, P. V. (1999). *Management of Irrigation and Drainage Systems: A Service Approach*. A. A. Balkema Publishers, Rotterdam, Netherlands.

Mason, D. C., Scott, T. R., and Wang, H. (2006). Extraction of tidal channel nnetwork from airborne scanning laser altimetry. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61:67–83.

McAllister, M. and Snoeyink, J. (2000). Medial axis generalization of river networs. *Cartography and Geographic Information Science*, 27(2):129–138.

Meisels, A., Raizman, S., and Karnieli, A. (1995). Skeletonizing a dem into a drainage network. *Computers & Geosciences*, 21(1):187–196.

Moreira, A. and Santos, M. (2007). Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. *GRAPP 2007: Proceedings of the International Conference on Computer Graphics Theory and Applications*.

Parry, M., Canziani, O., Palutikof, J., van der Linden, P., and Hanson, C. (2007). Contribution of working group ii to the fourth assessment report of the intergovernmental panel on climate change. *IPCC Fourth Assessment Report: Climate Change 2007*.

Passalacqua, P., Belmont, P., and Foufoula-Georgiou, E. (2012). Automatic geomorphic feature extraction from lidar in flat and engineered landscapes. *Water Resources Research*, 48(3).

Passalacqua, P., Do Trung, T., Foufoula-Georgiou, E., Sapiro, G., and Dietrich, W. E. (2010). A geometric framework for channel network extraction from lidar: Nonlinear diffusion and geodesic paths. *Journal of Geophysical Research: Earth Surface*, 115(F1):n/a–n/a. F01002.

Peters, R. and Ledoux, H. (2016). Robust approximation of the medial axis transform of lidar point clouds as a tool for visualisation. *Computers & Geosciences*, 90(Part A):123–133.

Peters, R., Ledoux, H., and Biljecki, F. (2015). Visibility analysis in a point cloud based on the medial axis transform. In Biljecki, F. and Tourre, V., editors, *Eurographics Workshop on Urban Data Modelling and Visualisation*. The Eurographics Association.

Rutzinger, M., Höfle, B., Pfeifer, N., Geist, T., and Stötter, J. (2006). Object-based analysis of airborne laser scanning data for natural hazard purposes using open source components. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36.4/C42.

Samet, H. (1990). *The design and analysis of spatial data structures*. Addison-Wesley Publishing Company, Reading, Massachusetts, USA.

Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient ransac for point-cloud shape detectiion. *Computer graphics forum*, 26(2):214–226.

Schön, B., Mosa, A., Laefer, D., and Bertolotto, M. (2013). Octree-based indexing for 3d point ccloud within an oracle spatial dbms. *Computers & Geosciences*, 51:430–438.

Siddiqi, K. and Pizer, S. (2008). *Medial Representations: Mathematics, Algorithms and Applications.*, volume 37. Springer, New York, NY, USA.

Toscano, G. J., Gopalam, U. K., and Devarajan, V. (2014). Auto hydro break line generation using lidar elevation and intensity data. *conference*.

van der Zon, N. (2013). Kwaliteitsdocument ahn2 (versie 1.3).

Vandenbruwaene, W., Meire, P., and Temmerman, S. (2012). Formation and evolution of a tidal channel network within a constructed tidal marsh. *Geomorphology*, 151-152:114–125.

Wang, M. and Tseng, Y. (2011). Incremental segmentation of lidar point ccloud with an octree-structured voxel space. *The Photogrammetric Record*, 26(133):32–57.

Wang, W., Pottmann, H., and Liu, Y. (2006). Fitting b-spline curves to point clouds by curvature-based squared distance minimization. *AACM Transactions on Graphics(ToG)*, 25(2):214–238.

Zervakis, D. (2015). Combining a physics-based model and spatial interpolation of scarce bed topography data in meandering alluvial rivers. *Faculty of Architecture and the Built Environment, Delft University of Technology, the Netherlands*.

Zhou, K., Gong, M., Huang, X., and Guo, B. (2011). Data-parallel octrees for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):669–681.

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. The figures and diagrams in this thesis were mostly drawn using IPE. Maps were generated with QGIS, and point cloud visualisations were made with CloudCompare. Flow charts were created using https://www.draw.io.