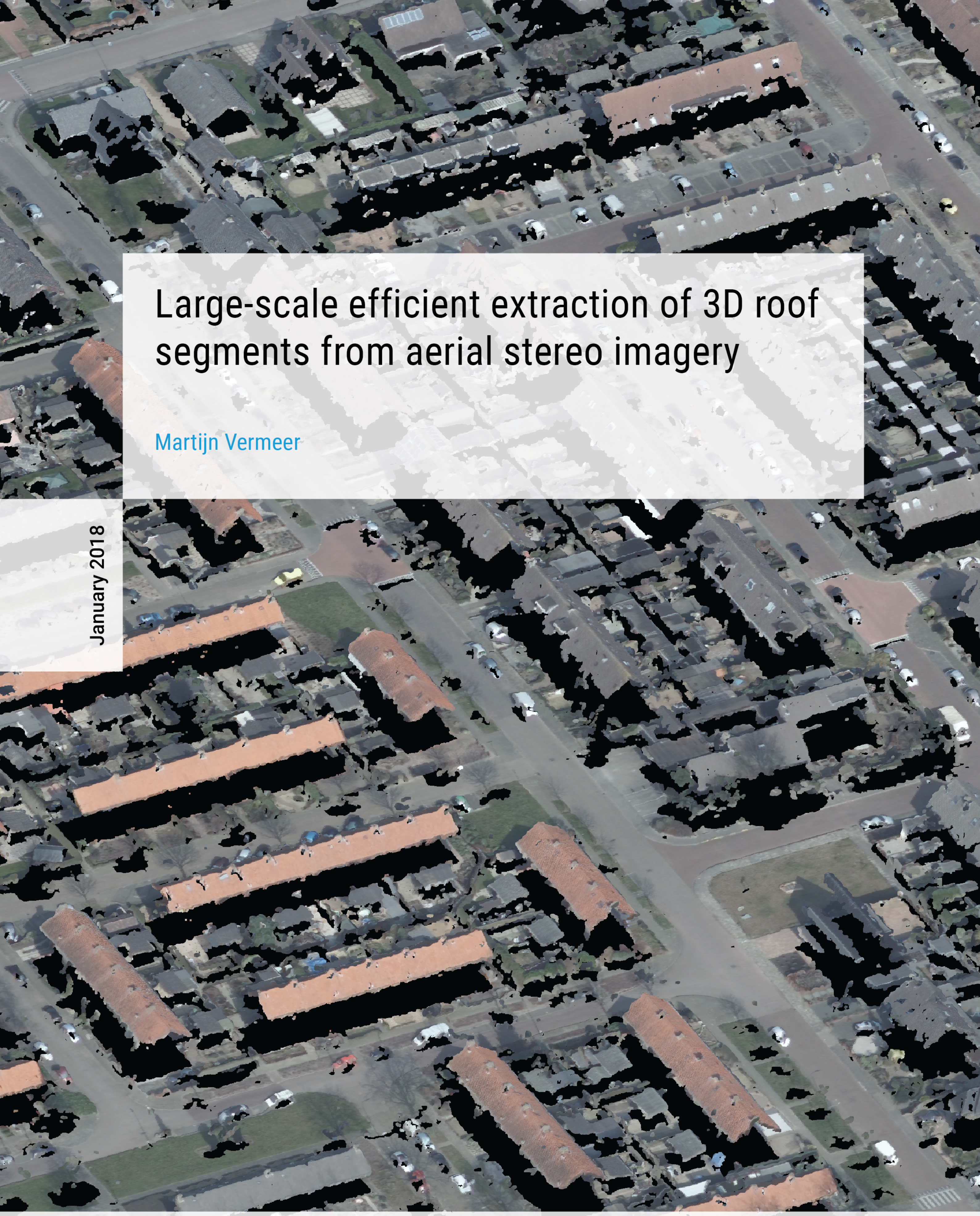


Large-scale efficient extraction of 3D roof segments from aerial stereo imagery

Martijn Vermeer

January 2018

Delft University of Technology

READAAR

MSc Thesis in Geomatics for the built environment

LARGE-SCALE EFFICIENT EXTRACTION OF 3D ROOF SEGMENTS FROM AERIAL STEREO IMAGERY

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

Master of Science in Geomatics for the Built Environment

by

Martijn Vermeer

Jan 2018

Martijn Vermeer: *Large-scale efficient extraction of 3D roof segments from aerial stereo imagery* (2018)

ISBN 999–99–9999–999–9

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
	Ir. Tom Commandeur
Co-reader:	Dr. Pirouz Nourian

ABSTRACT

3D reconstruction of the built environment is a widely studied Geomatics topic. Resulting city models are used for a great variety of purposes. The reconstruction of roofs remains challenging. These roofs are not only building blocks for the city models but are of direct interest as well. The company READAAR uses roof segments for the detection of asbestos and PV potential analysis. Typical clients for such analysis are muncipalities and provinces. Currently, READAAR extracts roof segments from the gridded LiDAR dataset: algemeen hoogtebestand nederland 2 (AHN2). The use of LiDAR data however comes with some limitations. Most importantly, outside the Netherlands, LiDAR data is not always available as it is relatively expensive to gather. Furthermore, the point density of the AHN is 6-10 points/m^2, which limits the amount of detail that can be extracted. In the Netherlands countrywide aerial stereo imagery is available at a resolution of 10cm, this potentially gives a point density of 100 points/m^2 after image matching. This research explores the possibilities of using aerial stereo imagery instead of LiDAR data for the efficient large-scale extraction of 3D roof segments. A workflow is designed in which stereo matching and extraction of segments are integrated. This makes the workflow both efficient and easily scalable. Roof segments are extracted in two steps. First, a watershed segmentation is applied to retrieved color segments. Second, these color segments are clustered based on their orientation. The resulting roof segments generally have a higher quality than the segments retrieved with READAARs LiDAR-based approach. However, problems do occur, especially in shaded areas. This could possibly be solved by integrating LiDAR data when available. Another recommendation for future research is improving the matching by using multi-view matching or possibly neural networks. Furthermore, the segmentation could potentially be improved by using multiple images from different years and processing building blocks.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisors Hugo Ledoux and Tom Commandeur for their great guidance and feedback throughout the project. Additionally, I would like to thank my co-reader Pirouz Nourian for providing me with very detailed and useful feedback.

I would like to thank Sven Briels and Matthijs van Til for giving me the opportunity to work on this project and a nice time at READAAR. Furthermore, thanks to Sven Briels, Jean-Michel Renders and Erik Heeres for their help and suggestions during the project.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AHN	actual hoogtebestand Nederland	2
BAG	basisregistraties adressen en gebouwen	41
CDF	cumulative probability density function	47
DEM	digital elevation model	13
DSM	digital surface model	11
FN	false negative	42
FP	false positive	42
PPM	perspective projection matrix	7
PDF	probability density function	16
KDE	kernel density estimation	16
LAD	least absolute deviations	17
OLS	ordinary least squares	17
PV	photovoltaic	1
PCA	principal component analysis	17
RANSAC	random sample consensus	18
RMSD	root-mean-square deviation	44
SGM	semi-global matching	9
TN	true negative	42
TP	true positive	42

1 INTRODUCTION

3D reconstruction is the process of reconstructing real world objects from 3D data. Such a virtual abstraction allows for a great variety of analysis that would otherwise not be possible or would require visiting/exploring the physical object [Moons et al., 2010]. Within Geomatics, 3D reconstruction of the built environment is a major research field. Various types of spatial data such as imagery, Light Detection and Ranging (LiDAR) and building footprints are combined to create 3D city models [Brenner, 2005]. These models are used in various domains, for example in navigation, sound and electromagnetic wave modelling, disaster management, urban planning and facility management [Kolbe, 2009]. For the majority of city models, roofs are among its components. The most common way to model roofs is as a polyhedral surface consisting of planar 3D roof segments [Rottensteiner, 2010]. Planar segments can be found by image and/or pointcloud segmentation [Haala and Kada, 2010]. An example of a pointcloud, its planar roof segments and polyhedral model, is given in Figure 1.1. Interestingly, planar roof segments are not only the building blocks for polyhedral city models, but are of direct interest as well. They can, among others, be used for the detection of asbestos and photovoltaic (PV) potential analyses. For such applications, when roof segments are used directly, there is no need to retrieve a watertight polyhedral building model.

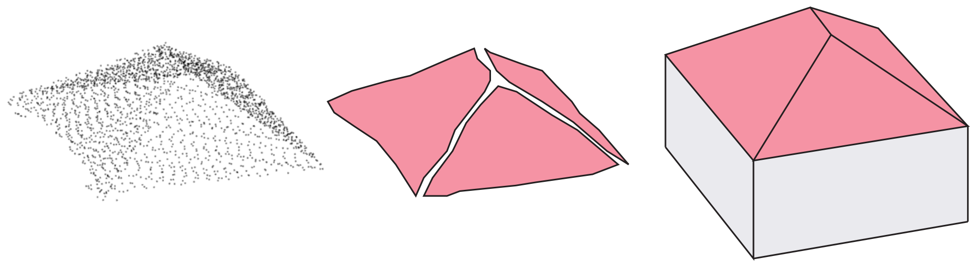


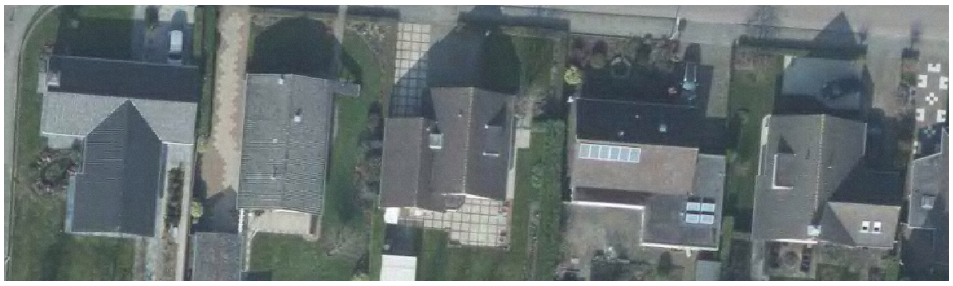
Figure 1.1: Roof pointcloud (left), planar roof segments (middle) and polyhedral building model (right)

Based on the PV potential of roof segments, the optimal configuration and yield of solar panels can be calculated. Ultimately, this allows to determine efficiently whether it is profitable to install PV panels at a particular house [Hofierka and Kaňuk, 2009]. Obviously, this promotes the use of sustainable solar energy. The detection of asbestos is highly relevant as well. As reported by SZW [2015], there is an estimated 120.000.000m^2 of asbestos in the Netherlands alone. This is causing approximately 1000 asbestos related deaths yearly. In order to deal with the problem, a law was passed by the Dutch parliament, stating that all asbestos has to be removed by 2024. For both these applications, a high degree of automation and efficiency is desired, since large numbers of buildings should be processed. Typical clients for these applications are municipalities or even provinces. However, the fully automatic efficient extraction of 3D roof segments remains far from a trivial task [Rottensteiner et al., 2014].

1.1 MOTIVATION

This problem is specifically addressed by READAAR, a company with whom this research is carried out. READAAR is based in the Netherlands and specializes in mining data from aerial- and satellite imagery and LiDAR data. They are especially active in the area of asbestos detection and PV potential analyses. Extraction of 3D roof segments is thus the core business of READAAR. Their current approach used for the extraction of 3D roof segments is LiDAR based. Inherently, the use of LiDAR data has several limitations. LiDAR datasets are relatively expensive to gather compared to aerial imagery [Novel et al., 2015]. In general, LiDAR datasets are therefore less frequently updated and not as widely available as stereo imagery. In case of the Netherlands, a countrywide LiDAR dataset, actual hoogtebestand Nederland (AHN), does exist and has been updated roughly every 7 years [van der Zon, 2013]. The main alternative data source to LiDAR is aerial stereo imagery. For the Netherlands, country wide aerial stereo imagery (Stereo10) is available at a resolution of 10cm. Imagery is retrieved yearly and offered by the company Cyclomedia. For more details on this Stereo10 dataset see § 5.2.2. With a resolution of 10cm, the pointcloud retrieved from image matching has a potential point density of 100/m^2. This is much denser than the 6-10 points/m^2 of the AHN.

Figure 1.2 shows some roof segments retrieved with READAARs current workflow. In this approach a gridded version of the AHN is used with a resolution of 0.5m. For each pixel in the grid a normal vector is determined based on its 8 neighbours. Subsequently, histogram thresholding is used (see § 2.3.1) to group pixels with a similar orientation. In this way the LiDAR grid is segmented into planar roof segments. Vectorization of the segments is done by simply taking the outer ring of pixels for each segment. Note that the resulting segments in Figure 1.2 are plotted over the aerial stereo imagery of the Stereo10 dataset. It clearly shows that the boundaries of these segments do not exactly follow the roof boundaries visible on the images. Generally the extracted roof segments do not extend all the way to the edge of the roof facets. One reason for this is that the orientation of pixels close to the edge is slightly off, since their orientation is determined partly by pixels outside the segment. Especially smaller roof segments, such as the two dormers in the middle building, are often omitted. These problems could potentially be solved using aerial stereo imagery.


(a) Aerial

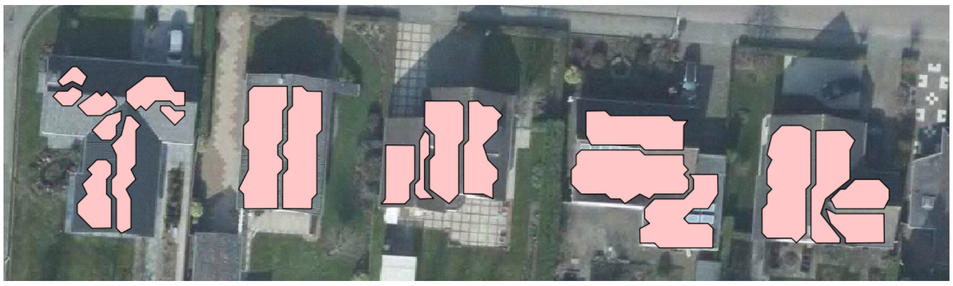

(b) Roof segments

Figure 1.2: Roof segments retrieved with READAARs LiDAR-based approach

The challenge of using only stereo imagery for 3D reconstruction is that image matching algorithms are required to retrieve a pointcloud. These algorithms are never perfect, meaning that not every pixel can be matched and matching errors are present. The resulting pointcloud will therefore be to some degree incomplete and contain errors [Nex and Remondino, 2012][Wolff et al., 2016]. Since LiDAR systems gather pointclouds directly in an active manner, they do not have these issues.

1.2 OBJECTIVES & RESEARCH QUESTIONS

This research explores the possibilities of using aerial stereo imagery instead of LiDAR data for the efficient large-scale extraction of 3D roof segments. The efficiency and scalability of the method are vital to make the approach usable, since in practice digital roof models are required mostly for large-scale applications. The main objective of this study is thus: developing a scalable method for the efficient extraction of 3D roof segments from aerial stereo imagery. The performance of the developed method is accessed in terms of completeness, accuracy and speed. Furthermore, a comparison is made with the current LiDAR based approach. The main research question is:

- *How to efficiently extract 3D roof segments from aerial stereo imagery for very large areas?*

This question incorporates the whole process from stereo imagery to roof segments. Furthermore, it deals both with the efficiency and scalability. Looking at the individual steps and requirements, the following subquestions are formulated:

- *Which stereo matching methods provide a good tradeoff between efficiency and quality?*
- *Which methods can and cannot be used for the efficient extraction of roof segments from pointclouds?*
- *How to ensure that the method is scalable to municipality level?*
- *Should radiometric (color) information be exploited?*
- *How to deal with data gaps and noise in the segmentation process?*
- *Which roof features can be ignored?*
- *What roof segment quality is achievable for an efficient stereo based method?*

In order to answer these questions: (1) a literature study is conducted on the current state of stereo matching and building extraction, (2) a workflow is designed for the efficient large scale extraction of roof segments from stereo imagery, (3) prototype software is developed to give a proof of concept, and (4) the prototype is compared to the current LiDAR based approach.

1.3 SCOPE

Regarding the scope of this research several things should be noted:

- The goal of this research is not to create new matching, clustering or segmentation algorithms. The goal is to come up with an efficient workflow for the large-scale extraction of 3D roof segments using existing algorithms. In some cases these algorithms are slightly modified.

- Which approaches and algorithms will be used is decided based upon existing literature, rather than via implementing and testing different algorithms and approaches. Explicitly, this applies to determination of the matching algorithm and segmentation/clustering approach and algorithms.

- Most studies in the area of building reconstruction aim to create watertight building models. In these studies the extraction of planar roof segments is often an intermediate step. In this study however, the retrieval of roof segments is the final goal. For this reason, boundaries of the extracted roof segments are not further optimized.

- No special attention is paid to deal with small roof object like chimneys and roof lights. Even though these features might be detected, the focus is on the extraction of the larger roof planes.

- Holes in roof segments are not reconstructed. Holes are present in the segmented image (see Figure 4.11), however, they are not vectorized. Since holes are already present in the image, functionality to deal with holes can easily be added in the future.

- Although the goal is to quantify the performance of the developed prototype, the comparison with other methods in literature will mainly be qualitative. The reason for this is that all methods deal with different input data.

1.4 THESIS OUTLINE

The remainder of this thesis is structured in the following way:

CHAPTER 2 will give the theoretical background necessary to understand the developed workflow and why certain design decisions were made. It covers the basics of computer (stereo) vision and gives a short summary of image segmentation, clustering and plane detection algorithms.

CHAPTER 3 gives an overview of the related work. The current approaches for building/roof reconstruction are discussed and compared.

CHAPTER 4 presents the designed methodology for the large-scale efficient extraction of roof segments from aerial stereo imagery. It is also argued why certain design decisions were made.

CHAPTER 5 describes how the developed method is implemented and presents the resulting roof segments and their quality.

CHAPTER 6 gives the most important conclusions that can be drawn based on this study. The research questions will be answered and the most important contributions are summarized. Furthermore, suggestions for future work are made.

2 THEORETICAL BACKGROUND

In this chapter the most important theoretical background for this thesis will be given. First, the basics of image projection (§ 2.1) and stereo vision (§ 2.2) are explained. These two concepts are the basis for 3D reconstruction based on imagery. In the books by Trucco and Verri [1998] and Szeliski [2010], detailed explanations are given. These books, together with the lecture series of [Collins, 2007] and paper by [Fusiello et al., 2000], were used extensively to write this summary on these two concepts. Subsequently, an overview of image segmentation (§ 2.3), clustering (§ 2.4) and plane fitting (§ 2.5) algorithms is given. All these algorithms are relevant for the reconstruction of buildings. However, not all of them are exploited in this study. Knowing these basic algorithms is critical to understanding why certain algorithms were selected.

2.1 IMAGE PROJECTION

The projection of 3D objects on a 2D image is known as image projection. The process of converting 3D world coordinates to 2D pixel coordinates is called forward projection. The principal of image projection forms the basis for most analyses in the fields of photogrammetry and computer vision [Collins, 2007]. In the following subsections the forward projection will be discussed step by step. First, the pinhole camera model is explained (§ 2.1.1). Second, the role of the intrinsics (§ 2.1.2), extrinsics (§ 2.1.3) and perspective projection matrix (§ 2.1.4) is described.

2.1.1 Pinhole camera model

The pinhole camera model describes the geometric relationship between points in 3D camera reference frame and their projection on the 2D image plane [Trucco and Verri, 1998]. In this model the camera aperture is modelled as a point known as the optical center (C). Points in the 3D camera reference frame are projected through C on the image plane. Thus, a point in camera space, its projection on the image plane and C are always collinear (see Figure 2.1) . The optical center forms the origin of the 3D camera reference frame. The x,y and z axis of the system are pointing sidewards (to the right), upwards and ahead with respect to the camera. The image plane, on which the 3D space is projected, is parallel to the x and y axis and orthogonal to the z axis. The intersection between the z axis and the image plane is the principal point (pp). Finally, the distance between the image plane and the optical center is known as the focal length (f).

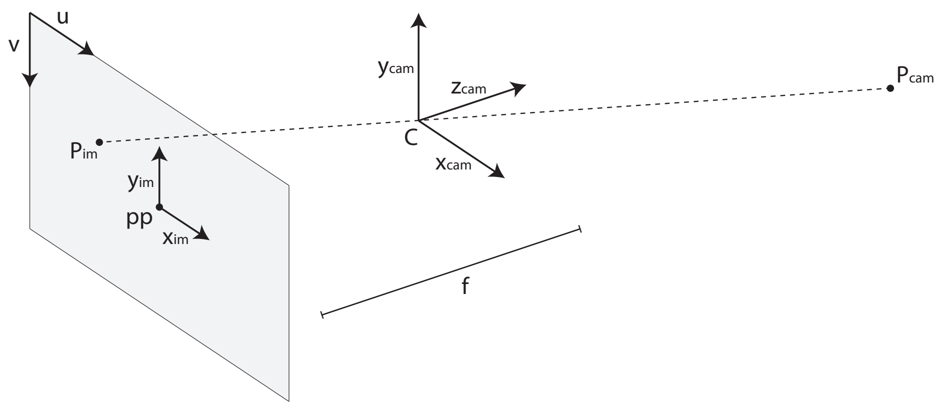


Figure 2.1: Pinhole Camera Model

Note that the ratio between the negative focal length and the image coordinates (x_{im}, y_{im}) is the same as the ratio between the depth (z_{cam}) and the x,y camera coordinates (x_{cam}, y_{cam}). The projection of camera coordinates on the image plane can thus be calculated using matrix Equation 2.1. The parameter λ is an arbitrary scale factor [Fusiello et al., 2000].

$$\lambda \begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} \tag{2.1}$$

2.1.2 Intrinsic parameters

The principal point and the focal length are both intrinsic parameters that allow the mapping from camera coordinates to pixel coordinates (u, v) [Trucco and Verri, 1998]. The resulting image coordinates from Equation 2.1 are still with respect to the camera frame. Therefore, they give the position in metric distance from the principal point and not the pixel coordinates. A new image reference frame is defined for the pixel coordinates. The origin of this frame is the left upper corner of the image. The direction of the y axis is opposite (downward), and the units are in pixels in this image reference frame. Matrix Equation 2.2 gives the transformation from camera to pixel coordinates [Collins, 2007]. Note that the transformation matrix (**A**) is similar to the matrix in Equation 2.1. The pixel coordinates of the principal point $(u0, v0)$ are added to deal with the origin shift. To determine $u0$ and $v0$ the image size in pixels and the pixel size are required. Additionally the sign in front of one f is removed to inverse the y axis. Furthermore, it should be noted that the focal length is expressed in pixels. In this simple explanation lens distortions are omitted. These distortions can also be incorporated in **A** [Collins, 2007].

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -f & 0 & u0 \\ 0 & f & v0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} \tag{2.2}$$

2.1.3 Extrinsic parameters

The extrinsic parameters give the location and orientation of the camera with respect to a 3D world reference frame. These parameters are necessary for the transformation from 3D world coordinates to 3D camera coordinates [Collins, 2007]. The location parameters (x, y, z) of the camera are taken at C, since this is the origin of the camera frame. The orientation parameters are omega, phi and kappa (ω, ϕ, κ). They give the clockwise rotation around the x,y and z axis (Figure 2.2). The location and orientation parameters are used to construct the translation vector **t** (Equation 2.3) and rotation matrix **R** (Equation 2.4) respectively. With Equation 2.5 world coordinates can be transformed to camera coordinates by rotation and translation using **R** and **t** [Trucco and Verri, 1998]. The transformation from the world reference frame to the camera reference frame is visualized in Figure 2.2.

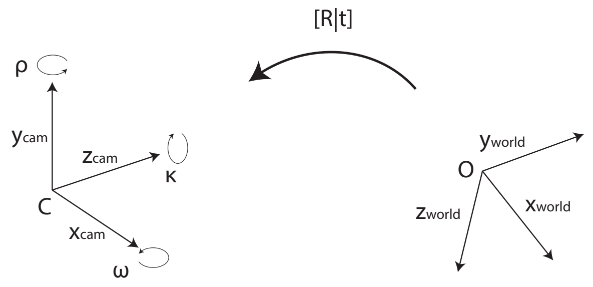


Figure 2.2: Transformation from world to camera reference frame by rotation and translation

$$\mathbf{t} = \begin{bmatrix} -x \\ -y \\ -z \end{bmatrix} \tag{2.3}$$

$$\mathbf{R} = \begin{bmatrix} cos(\kappa) & -sin(\kappa) & 0 \\ sin(\kappa) & cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\omega) & -sin(\omega) \\ 0 & sin(\omega) & cos(\omega) \end{bmatrix} \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix} \tag{2.4}$$

$$\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} = \left[\mathbf{R}|\mathbf{t}\right] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2.5}$$

2.1.4 perspective projection matrix

Equation 2.2 and 2.5 can be combined to form Equation 2.6. With this equation 3D world coordinates can be transformed directly to 2D pixel coordinates by multiplication with the perspective projection matrix (PPM). As can be derived from Equation 2.6, the PPM (**P**) is equal to the product of **A** and $\left[\mathbf{R}|\mathbf{t}\right]$ [Fusiello et al., 2000].

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \left[\mathbf{R}|\mathbf{t}\right] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2.6}$$

2.2 COMPUTER STEREO VISION

The transformation from world coordinates to pixel coordinates described in § 2.1 is called forward projection. The reverse process, back projection, is the transformation from pixel coordinates to world coordinates. Whereas forward projection always has a single solution, back projection has a fundamental ambiguity. The source of this ambiguity is the transformation from 2D image coordinates back to 3D camera coordinates [Collins, 2007]. As depicted in Figure 2.3, the 3D point (P), corresponding to a 2D projection point (p) on the left image, could be located anywhere along the projection line. P_1,P_2 and P_3 are all possible locations for point P. The ambiguity can be resolved by using a second image. If the same point is present in two images, its 3D location can be determined via triangulation [Szeliski, 2010]. These overlapping images, capturing the same scene from a different viewpoint, are known as stereo images. A pair of stereo images is called a stereo pair. The process of gathering 3D information from stereo images is known as computer stereo vision. The three main steps in the computer stereo vision process are: rectification, matching and triangulation. These three steps are discussed in more detail in § 2.2.1,§ 2.2.2 and § 2.2.3.

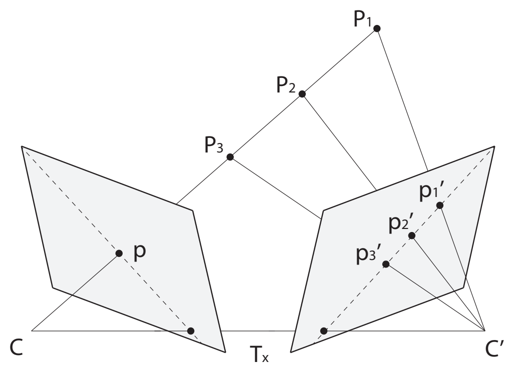


Figure 2.3: Stereo pair with epipolar lines (dashed)

2.2.1 Rectification

Image rectification is a processing step that is done prior to the matching to simplify the process of finding matching pixels. In order to understand rectification, the concept of epipolar lines is explained first. Assume the projection point p on the left image is known (Figure 2.3). This means that the projection line along which P is located is known as well. The projection of this projection line on the right image gives the epipolar line (dashed). The projection of P on the right image will be somewhere along this line. The projection points p_1',p_2' and p_3' are given as example. Similarly, the projection line of P on the right image can be projected on the left image to get the corresponding epipolar line of the left image. It does not matter which of the three depicted projection lines is used. The reason for this is that all the projection lines fall in the same plane. This is the epipolar plane defined by the two projection centers (C and C') and P. The epipolar lines are the intersections of the image planes with the epipolar plane [Fusiello et al., 2000]. So considering a point on one image, the corresponding point in the second image must be located on one particular line. Per definition the search space for a matching point is thus limited to this line. Image rectification is the process of projecting two images on a common image plane (Figure 2.4). Note that the projection centers remain at the same location and only the image planes change. The result of rectification is that the epipolar lines become exactly horizontal, meaning that matching points will always have the same y coordinate. The search space is therefore limited to 1D, since only the matching x coordinate needs to be found [Trucco and Verri, 1998]. In the case that images are retrieved with a stereo rig, the images are already rectified. The cameras in the stereo rig are mounted in such a way that the images have the same image plane.

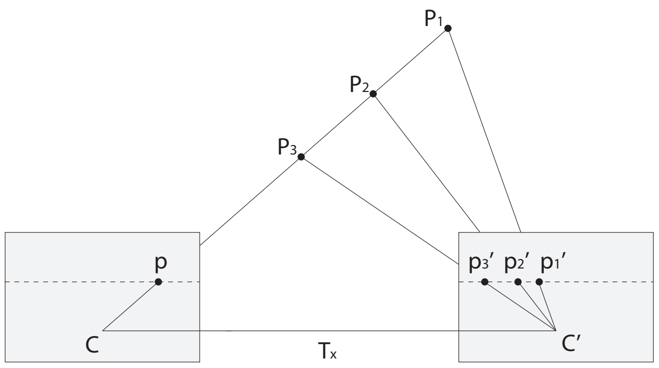
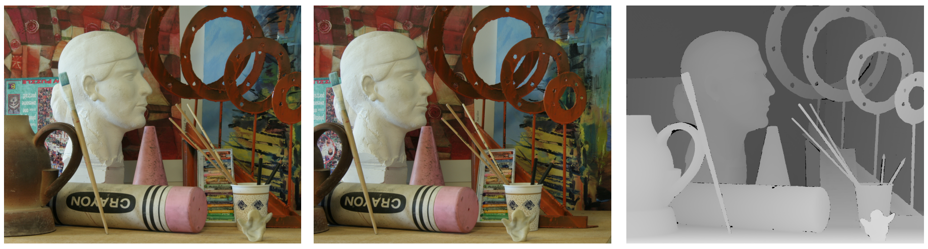


Figure 2.4: Rectified stereo pair having horizontal epipolar lines (dashed)

2.2.2 Stereo matching

Stereo matching is the process of finding corresponding pixels in a rectified stereo pair. More precisely it is determined for each pixel in the left image what the corresponding pixel is in the right image. Since the images are rectified, these corresponding pixels must be on the same row. The correspondence can therefore be expressed as the shift in x coordinate, known as the disparity. By finding the disparity for each pixel in the left image a disparity map is created [Collins, 2007]. An example of a disparity map corresponding to a rectified stereo pair is given in Figure 2.5.



(a) Left image (b) Right image (c) Left disparity map

Figure 2.5: Rectified stereo pair and disparity from Middlebury dataset [Scharstein and Pal, 2007]

Many different stereo matching algorithms exist, most of which can be either classified as local or global algorithms. In local algorithms an image patch is taken from the left image centered at the pixel in question. Subsequently, this patch is shifted along the same row in the right image. At each location a matching score is calculated. The highest score gives the location of the matching pixel and thus the disparity. Global algorithms do not work via this winner takes all principle, instead a global optimization function is defined. Other than the matching score for each pixel, this function also contains a smoothness term and possibly deals with various constraints [Hirschmüller, 2005]. Local algorithms are generally fast, but perform poorly in areas with little texture or repeated patterns. As a result, matching results are often incomplete and inaccurate. Global algorithms on the other hand can partly deal with these problems. However, this comes at the cost of much longer computation times [Geiger et al., 2010].

The semi-global matching (SGM) proposed by Hirschmüller [2005] combines the advantages of both approaches. In SGM optimization is done along several 1D paths instead of the whole image. Therefore it is much faster than global approaches, whilst still achieving a similar accuracy. A more recent algorithm for efficient and accurate stereo matching is LIBELAS, introduced by Geiger et al. [2010]. The method makes use of the triangulation of support points that can be robustly matched. A similar study by Sinha et al. [2014] present an algorithm using local plane sweeps, achieving accurate results in an efficient way as well.

Another branch of stereo matching algorithms is based on deep learning, using convolutional neural networks. Recent studies by Luo et al. [2016] and Zbontar and LeCun [2016] showed that results are very promising. However, training data for aerial images is currently nonexistent, since much of the research originates from the self-driving car industry. As a result, these algorithms can not yet be exploited for aerial stereo matching without creating training data yourself.

2.2.3 Triangulation

Using the disparity values, the pixel coordinates can be transformed back to 3D camera coordinates. The basis of this transformation is triangulation. The ratio between the focal length (f) and the depth (z_{cam}) is equal to the ratio between the negative disparity (-d) and the baseline length (T_x). The baseline is the line connecting the two projection centers (C and C'). Note that the negative disparity is used to get a positive value, since the shift in x coordinate is negative as well. Figure 2.6 visualizes the relationship by adding the two red lines which are parallel and have the same length as the right projection line and baseline respectively. The depth can be found with Equation 2.7. The disparity is thus inversely proportional to the depth, meaning that a larger disparity corresponds to a lower depth.

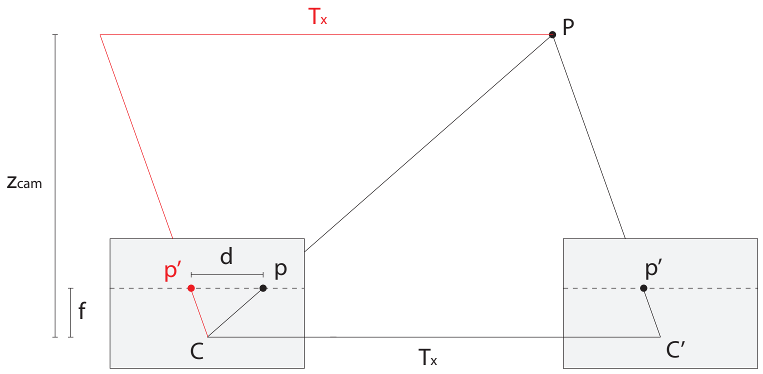


Figure 2.6: Triangulation to determine the depth (z_{cam}) if focal length (f), disparity (d) and baseline length (T_x) are known

$$z_{cam} = \frac{f * T_x}{-d} \tag{2.7}$$

Recall from § 2.1.1, that the ratio between the focal length and the depth is also the same as the ratio between the x and y image and camera coordinates. Since pixel coordinates are known x_{cam} and y_{cam} can be calculated with Equation 2.8.

$$x_{cam} = \frac{(u - u0) * T_x}{-d} \quad \text{and} \quad y_{cam} = \frac{(v - v0) * T_x}{-d} \tag{2.8}$$

Matrix Equation 2.9 captures the entire triangulation that allows the transformation from pixel to camera coordinates [Bradski and Kaehler, 2008]. The transformation matrix is denoted as **Q**. By applying the triangulation on every pixel in the image, a pointcloud is retrieved. Figure 2.7 shows the pointcloud resulting from the Middlebury dataset example in Figure 2.5.

$$\lambda \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -u0 \\ 0 & 1 & 0 & -v0 \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix} = \mathbf{Q} \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix} \tag{2.9}$$

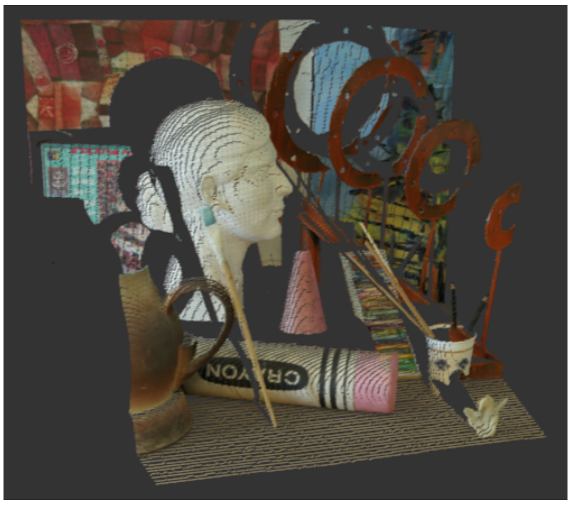


Figure 2.7: Pointcloud resulting from stereo matching of Middlebury dataset example stereo pair and disparity map

2.2.4 Aerial Stereo Imagery

Within Geomatics the concepts explained above are better known under the term stereophotogrammetry. Stereophotogrammetry is often applied to retrieve a digital surface model (DSM) from aerial stereo imagery. The imagery is gathered in an aerial survey, using any type of flying vehicle. Most commonly planes are used to cover large areas. Data is gathered in blocks, which are roughly rectangular shaped areas. Each block is captured by flying a number of parallel flight lines. These are defined in such a way that images taken in two adjacent lines have a slight overlap. In this way it is ensured that the entire block is captured [Egels and Kasser, 2003]. The overlap from one flight line to the next is known as side overlap. The imagery in a single flight line is also overlapping, this is called forward overlap. The traditional side and forward overlap is 20-30% and 60% respectively (Figure 2.8). The forward overlap ensures that each point is captured twice allowing stereophotogrammetry [Kraus, 2007].

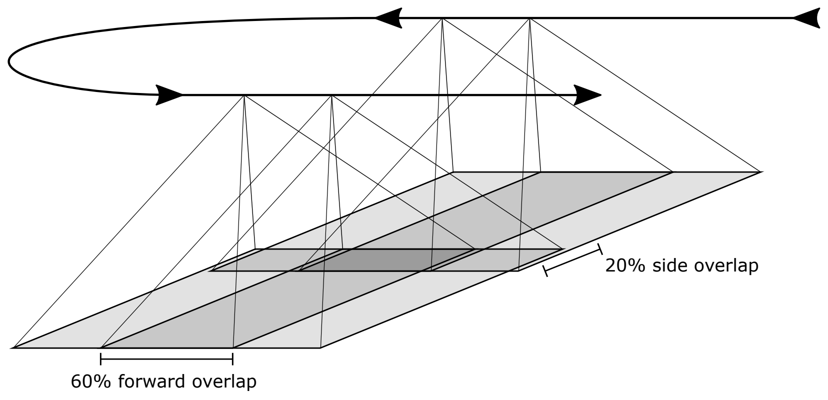


Figure 2.8: Schematic representation of four stereo images taken in two different flight lines with 60% forward and 20% side overlap

2.3 IMAGE SEGMENTATION

Image segmentation is the process of partitioning an image into segments, which are sets of pixels that have similar properties. Image segmentation is generally used to identify objects in images. A great variety of image segmentation algorithms exists [Szeliski, 2010]. Some of the most common branches of methods will be discussed shortly in the next subsections. All clustering methods discussed in § 2.4 can be used for image segmentation as well.

2.3.1 Thresholding methods

The most straightforward method to segment an image is to select a single threshold value, based on for example color or intensity, and create a binary image. This method can be extended by selecting multiple threshold values, in order to divide the image into more segments. Which threshold values should be used, can be determined based on the histogram of an image. Many different techniques exist for selecting the right threshold values [Huang and Wang, 2009] [Raju and Neelima, 2012]. Figure 2.9 gives an example of image segmentation by thresholding from Cho et al. [2013]. Thresholding methods are generally very fast, since each pixel only has to be evaluated once. The downside is that only the pixel value is taken into account and not the location or neighboring values. The assumption is thus made that different objects have different intensity values and that all pixels in a object have the same value. In reality this is not the case, resulting in segmentation errors. Furthermore the selection of good threshold values is challenging when the histogram is relatively flat [Dey et al., 2010] [Kaur and Kaur, 2014].

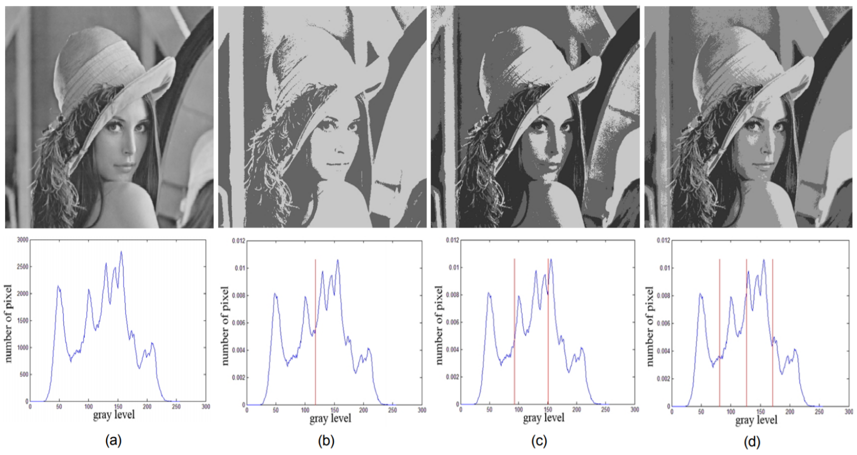


Figure 2.9: Lena image with histogram (a) and the results of histogram thresholding using 1,2 and 3 threshold values (b-d) [Cho et al., 2013]

2.3.2 Region growing methods

Region growing segmentation methods are region based, meaning that regions are retrieved directly instead of their boundaries. Region growing algorithms start with marking a number of pixels as seed points. Starting from the these seed points, regions are grown. Iteratively all neighbors are evaluated based on a similarity criterion. Neighbors that are similar are added to the region. The region is finished when no more neighbors satisfy the similarity criterion [Dey et al., 2010]. The advantage of region growing methods is that resulting segments will always be connected. The main disadvantage is that region growing algorithms are computationally expensive. Furthermore, only local evaluations are made, so the segmentation problem is not addressed globally [Kaur and Kaur, 2014].

2.3.3 Graph based methods

In graph based methods the segmentation problem is addressed globally. A graph of the image is constructed in which pixels are the nodes. Edges are formed by connecting neighboring pixels. The weight of each edge is determined based on the similarity between the connected pixels. The more similar pixels are, the higher the weight. The similarity can be based on any property of the pixels, such as the intensity [Szeliski, 2010] [Felzenszwalb and Huttenlocher, 2004]. The idea of graph based segmentation is to split up the constructed graph and in this way retrieve segments. A common way to split up a graph is by finding minimum cuts. This is a graph cut where the combined weights of the edges that are cut is minimized. The cut is thus made where the dissimilarity of pixels is highest. The problem with minimum cuts is that it is biased towards finding short cuts, creating small components [Felzenszwalb and Huttenlocher, 2004]. Shi and Malik [2000] proposed a solution for this by introducing normalized cuts. Instead of only considering the total dissimilarity of the cut, the similarity within the regions that are cut is taken into account as well. An advantage of graph based methods is that the segments resulting from graph partitioning are always connected. Additionally, global image properties are taken into account. A drawback is that graph construction and partitioning are generally computationally expensive operations [Szeliski, 2010].

2.3.4 Watershed segmentation of the gradient magnitude

The underlying idea of watershed segmentation comes from the geography. Here it is used to segment digital elevation model (DEM)s in order to identify the different catchment basins, which are separated by watersheds [Roerdink and Meijster, 2000]. In practise, watershed segmentation can be used on any image by considering the gradient image as topographic relief [Kaur and Kaur, 2014]. The idea of the algorithm is to have a water source in each regional minima and raise the water level continuously. When two water bodies meet a watershed boundary is constructed. After complete flooding the entire image is partitioned in catchment areas separated by watersheds. Figure 2.10 presents the concept of watershed segmentation in one dimension. Watershed segmentation is edge based, so instead of looking for regions directly their edges are found [Szeliski, 2010] [Dey et al., 2010]. The method is a relatively efficient since all segments are found at once and not iteratively. A common drawback of the watershed segmentation of the gradient magnitude is over-segmentation in areas with high contrast (see Figure 2.11). There are different ways to reduce over-segmentation, for example by, filling local minima, image smoothing and starting flooding at less points [Dey et al., 2010] [Kaur and Kaur, 2014].

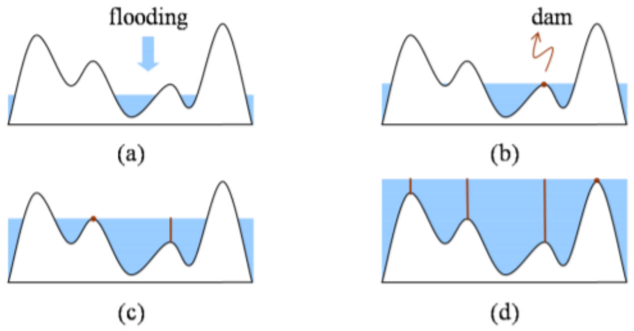


Figure 2.10: 1D presentation of watershed segmentation by raising water level [Wang, 2010]

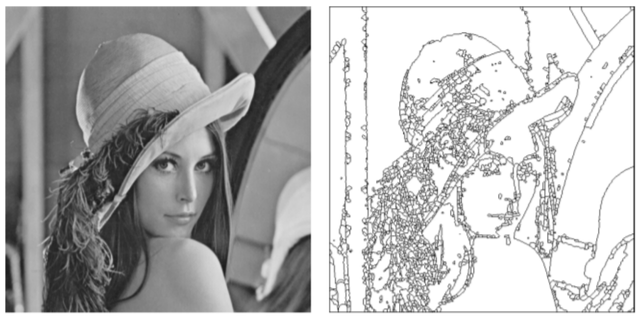


Figure 2.11: Watershed segmentation of Lena [Ruparelia, 2012]

2.4 CLUSTERING

Clustering is the unsupervised grouping of data points into groups called clusters. The goal is to create clusters in which data points have similar properties, and in this way bring structure in unorganized data. Clustering is an unsupervised machine learning task, meaning that no labeled training data is used to train the classifier [Xu and Wunsch, 2005]. Since clustering is a fundamental task in data analysis it is used in many different research fields. A great variety of clustering algorithms exists [Szeliski, 2010]. Some of the most common methods will be presented in the following subsections.

2.4.1 *K*-means clustering

K-means clustering is a centroid-based clustering algorithm, dividing the data points into K clusters. The value of K needs to be specified on beforehand. The algorithm starts by placing K centroids in random locations. Subsequently, each data point is assigned to the closest centroid. Based on the data points assigned to each centroid, the location of the centroids is updated. This process repeats itself until the location of the centroids stabilize and K clusters are found (Figure 2.12) [Jain, 2010] [Szeliski, 2010]. The strength of the k-means algorithm is the simplicity and efficiency. A potential drawback is that the amount of clusters need to be specified in advance. In the case of this study the number of roof segments that a building has is not known in advance, making it difficult to determine the right value for K. In addition it should be taken into account that K-means assumes roughly spherical clusters with equal sizes [Jain, 2010]. More precisely, the cluster shape is determined by the voronoi cells of the cluster centroids.

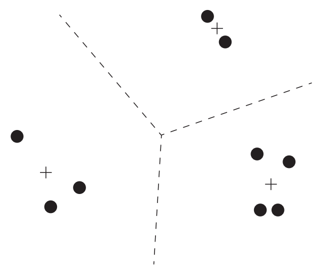


Figure 2.12: Centroids and corresponding clusters in K-means clustering ($K = 3$)

2.4.2 Hierarchical clustering

In hierarchical clustering a dendrogram is created of all the data points. This can be done in two distinct ways, divisive and agglomerative. In the divisive approach all data points start in one cluster, which is split recursively until each data point is in its own cluster. The agglomerative approach is the other way around. Each data point starts in its own cluster. Recursively, the closest clusters are merged until one cluster remains (Figure 2.13) [Xu and Wunsch, 2005]. Which distance is measured depends on the linkage criterion. Examples are: the distance between cluster centroids, the two closest points (single-linkage) or the two most distant points (complete-linkage) [Szeliski, 2010]. An advantage of hierarchical clustering is that the dendrogram can be broken at different places, giving different levels of clustering. The main disadvantage is that the construction of a dendrogram is computationally expensive, especially for larger datasets. Furthermore hierarchical clustering has a tendency towards forming spherical clusters [Xu and Wunsch, 2005].

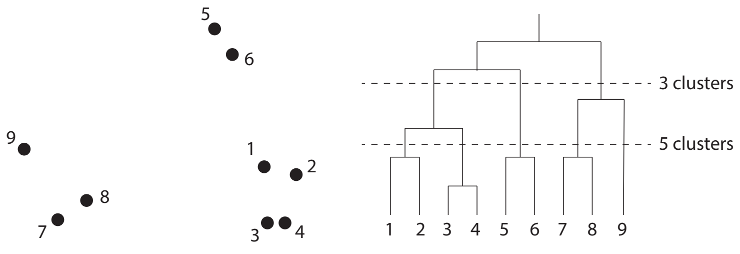
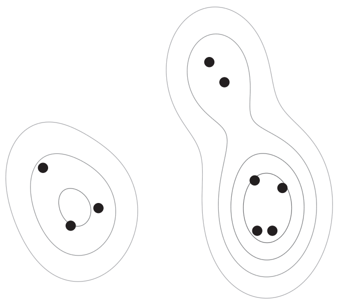


Figure 2.13: Dendrogram created in agglomerative hierarchical clustering

2.4.3 Mean shift clustering

Mean shift is a iterative procedure to find the maxima of a density function. It can be used for clustering by retrieving the underlying probability density function (PDF) of a set of data points. The PDF of a set of points can be determined using kernel density estimation (KDE) [Cheng, 1995]. The way KDE works is by placing a kernel, usually Gaussian, at each data point. The sum of these kernels form the PDF. The idea of mean shift is to iteratively shift the data points uphill, in the direction of higher probability density, until a peak is reached. The data points ending up in the same peak belong to the same cluster. These peaks are also called the modes of the PDF, making mean shift a mode-seeking algorithm [Cheng, 1995] [Szeliski, 2010]. The kernel bandwidth determines how smooth the PDF is and therefore the degree of clustering (Figure 2.14). In its simplest form the entire algorithm can be controlled only by this parameter [Comaniciu and Meer, 2002]. Advantages of mean shift clustering are the simplicity and fact that it can handle different cluster sizes and shapes. A disadvantage, especially when dealing with many data points, is that the iterative process is computationally expensive. The algorithm is however completely parallelizable since all points can be shifted at the same time [Cheng, 1995].


(a) High bandwidth

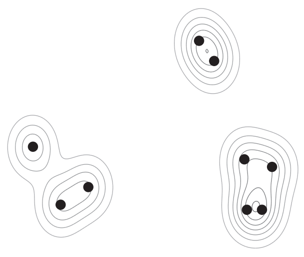

(b) Low bandwidth

Figure 2.14: Probability surfaces used in mean shift Clustering

2.5 FITTING A PLANE THROUGH A SET OF 3D POINTS

Fitting a plane through a set of 3D points is a basic mathematical operation. Especially within the field of Geomatics where pointclouds are a widely used data source. The basic formula for a plane is given in Equation 2.10. The vector (a, b, c) is the normal vector of the plane, defining the orientation. However, a plane only has three degrees of freedom. Therefore c can be arbitrarily set to 1, resulting in Equation 2.11. This simplifies things as d gives the intersection with the z axis now. A plane can thus uniquely be defined by three parameters (a, b and d). When fitting a plane through a set of points these three parameters, forming a plane model, are estimated. There are various ways to find a plane model for a set of 3D points. In the coming subsections some of the most common methods will be discussed.

$$ax + by + cz + d = 0 \tag{2.10}$$

$$\begin{aligned} ax + by + z + d &= 0 \\ ax + by + d &= -z \end{aligned} \tag{2.11}$$

2.5.1 Ordinary least squares

The most straightforward method uses ordinary least squares (OLS). With the OLS method the sum of the squared vertical distances from each point to the plane is minimized. If there are n points, a system of n linear equations can be set up. Equation 2.12 gives the matrix equation of this system. When n is larger than 4, this system is over-determined, and can be solved with OLS using Equation 2.13. The disadvantage of using OLS is that outliers have most influence on the plane model, making this approach not very robust [Meer et al., 1991] [Schnabel et al., 2007]. An option to deal with outliers is to use OLS multiple times, removing points exceeding a certain distance threshold at every iteration.

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \dots & \dots & \dots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} -z_0 \\ -z_1 \\ \dots \\ -z_n \end{bmatrix} \tag{2.12}$$
$$\mathbf{Ax} = \mathbf{b}$$

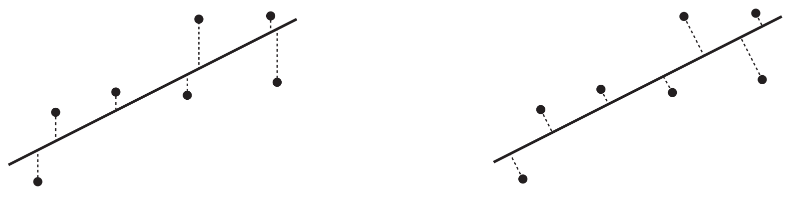
$$x = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{2.13}$$

2.5.2 Least absolute difference

An alternative is to use least absolute deviations (LAD), which minimizes the sum of the absolute vertical distances to the plane. By doing this, the outliers do not have extra influence on the model, therefore the method is more robust. However, the drawback of LAD is that it does not have a closed-form solution. The LAD must therefore be computed iteratively, making this approach less efficient [Bloomfield and Steiger, 2012].

2.5.3 Principal component analysis

Observations are often described by several correlated variables. principal component analysis (PCA) is the process of finding the set of linearly uncorrelated variables, called principal components, that describe the same data. These principal components are per definition orthogonal to each other [Abdi and Williams, 2010]. In the case of a pointcloud, three possibly correlated variables are the x, y and z coordinates of the individual points. The principal components will in this case be three orthogonal axis along which the data is uncorrelated. These principal components can be found by eigen-decomposition of the covariance matrix [Nourian, 2016]. The first principal component has the largest possible variance. The second component has the largest possible variance under the restriction that it is orthogonal to the first component. Finally the third principal component is orthogonal to the other two [Abdi and Williams, 2010]. The first two components define the plane that fits the data points. The third component is the normal vector of the plane. Furthermore, the eigenvalues give an indication of the curvature of the surface described by the points. With the PCA method the sum of the squared orthogonal, not vertical, distances of the points to the plane is minimized (see Figure 2.15) [Nourian, 2016].



(a) Vertical distances (b) Orthogonal distances

Figure 2.15: Difference between orthogonal and vertical distances to plane

2.5.4 Random sample consensus

random sample consensus (RANSAC) is a technique which is common exploited in the image processing domain to detect certain shapes (lines, circles, ellipses etc.). It can also be exploited in 3D to detect planes in a pointcloud [Tarsha-Kurdi et al., 2007a]. RANSAC is especially designed to deal with outliers, ensuring they do not influence the mathematical model that is estimated. The RANSAC algorithm iteratively takes random samples of 3 points. Such a sample defines a plane, for which the amount of inliers and outliers can be determined based on some distance threshold. The sample with the most inliers is selected as mathematical model for the plane [Tarsha-Kurdi et al., 2007a] [Sampath and Shan, 2010]. If more iterations are allowed, the solution stabilizes. The downside of RANSAC is that it is nondeterministic. Furthermore, it is computationally expensive to allow many iterations.

2.5.5 Hough transform

Hough transform is a technique designed to detect features of a particular shape in images. The concept of Hough transform can easily be transformed to 3D, to detect planes in a pointcloud [Vosselman and Dijkman, 2001]. Hough transform works by iterating over all points (x, y, z) in a pointcloud. For all possible planes going through this point it is determined how many other points are within a certain distance from this plane. Since there are theoretically infinitely many planes going through each point, the plane parameters (a, b, d) are discretized using a predefined step size. All results are stored in a 3D matrix with axis (a, b, d). By finding the maxima in this matrix, the plane parameters of the planes containing most points are retrieved. The underlying principle of 3D Hough transform is thus to find the planes containing the maximum number of points. It is important to keep in mind that these planes do not necessarily represent a single roof segment [Tarsha-Kurdi et al., 2007a]. The advantage of Hough transform is that multiple planes in a single pointcloud can be detected at once. The main drawback of 3D Hough transform is that it is computationally expensive, especially when the step size for the plane parameters is small. On the other hand a larger step size reduces the quality of the result [Tarsha-Kurdi et al., 2007a].

3 RELATED WORK

The automatic 3d reconstruction of buildings has been a widely studied topic starting from the mid 1990s. An important driver behind this research was the newly available LiDAR technology that provided more accurate and increasingly dense height measurements [Brenner, 2005]. A comprehensive overview of studies on building reconstruction is given by Brenner [2005], this overview is complemented by Haala and Kada [2010]. A more recent impression on the state of building reconstruction can be found in Rottensteiner et al. [2014]. The two major data sources that are generally exploited for building reconstruction are LiDAR and/or imagery. This data can be gathered with either airborne or ground-based sensors [Haala and Kada, 2010]. Ground-based data mostly only covers the facades of buildings and cannot be used for the reconstruction of roofs. Therefore, only research based on airborne LiDAR and aerial imagery will be discussed in this chapter. In the case of LiDAR, 3D geometric data is retrieved directly in the form of a pointcloud. Whilst with aerial imagery, radiometric data is retrieved directly. A pointcloud containing 3D geometric data is only extracted in a second step by stereo matching (see § 2.2). The pointclouds retrieved with LiDAR and imagery have inherently different characteristics (Figure 3.1). Image-based pointclouds contain more noise and blunders due to matching errors. Furthermore, low contrast areas often cannot be matched, resulting in gaps [Nex and Remondino, 2012] [Wolff et al., 2016]. Note for example the shaded side of the buildings in Figure 3.1. On the other hand, image-based pointclouds contain radiomatric (color) data and generally have a higher point density. In the Netherlands for example, the nationwide LiDAR dataset AHN2 has an approximate point density of 6-10 points/m^2 [van der Zon, 2013]. Nationwide aerial imagery, however, is available at a resolution of 10cm, giving a potential point density of 100 points/m^2.

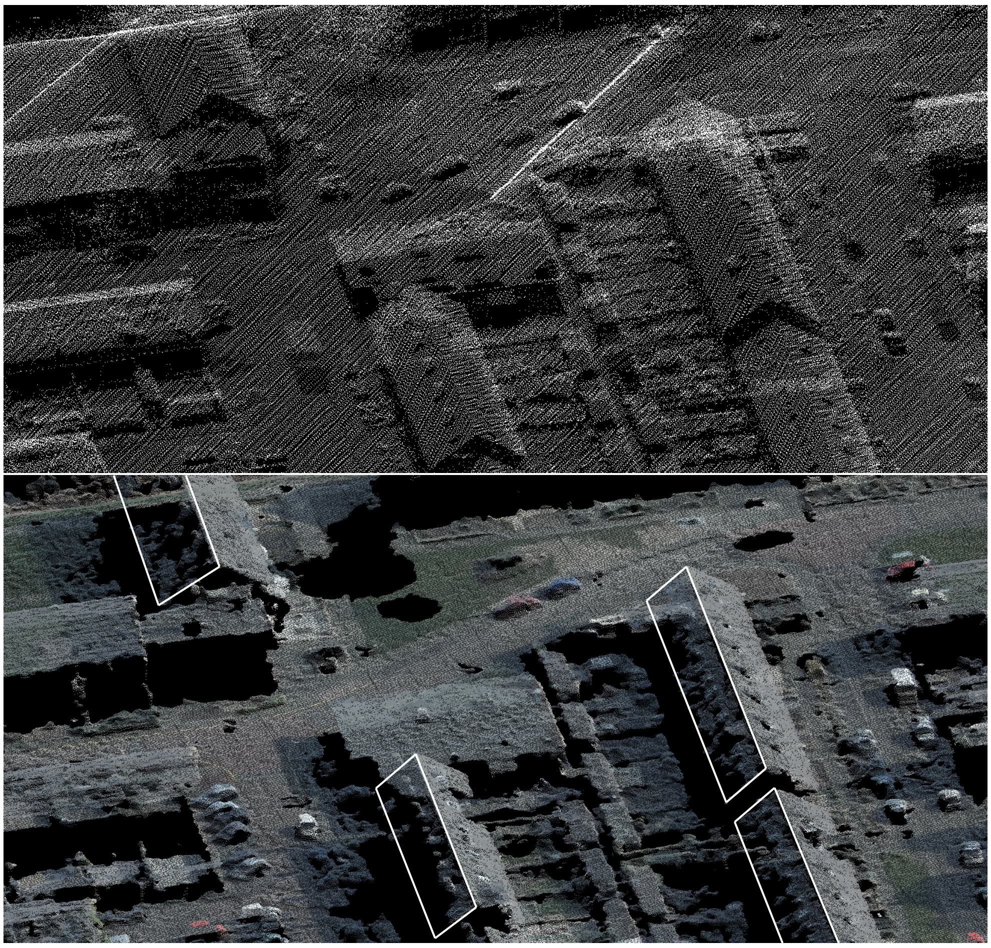


Figure 3.1: Pointcloud derived with LiDAR (upper) and stereo matching (lower). Shaded roof areas are outlined in white.

Instead of using pointclouds directly for building reconstruction, they can also be gridded or triangulated into a DSM first. The majority of studies in literature are LiDAR-based. However, due to higher resolution imagery with greater overlap and increased computing power, image-based methods provide a good alternative [Nex and Remondino, 2012]. Furthermore, many studies have a hybrid approach aiming to get the best of both worlds.

In addition to 3D geometric data gained from LiDAR or imagery, almost all studies make use of a dataset containing building footprints. Especially in Western Europe, nationwide datasets containing the building footprints are widely available [Haala and Kada, 2010]. When building footprints are not available, the possibility exists to extract these automatically from imagery or LiDAR. An example of such a study is [Vakalopoulou et al., 2015]. In literature a great variety of methods to retrieve building models from pointclouds, DSMs and imagery exists. These methods generally focus on the creation of entire building models as polyhedrons. Often, retrieving planar roof segments is an intermediate step in this process. The different reconstruction methods will be discussed on the basis of the following categorization made by Haala and Kada [2010]:

Parametric-based methods: methods based on retrieving the parameters of parameterized standard roof shapes (§ 3.1).

Segmentation-based methods: methods based on the segmentation of pointclouds and/or images (§ 3.2).

Simplification-based methods: methods based on the simplification of a DSM (§ 3.3).

Finally, § 3.4 will give a conclusion of the relevant work with respect to this research. The focus in this conclusion will thus be on the efficiency of the various methods.

3.1 RECONSTRUCTION BASED ON PARAMETRIC ROOF SHAPES

The underlying idea of reconstruction based on parametric roof shapes is that most roofs are a composition of standard roof shapes (primitives). Examples of such standard shapes are hipped, gabled and flat roofs. A list of standard roof shapes is given by Huang et al. [2011], see Figure 3.2. Reconstruction based on parametric roof shapes is model-driven. This means that certain assumptions on the shape are made beforehand, only the parameters of some predefined shapes need to be found [Tarsha-Kurdi et al., 2007b]. A short summary of different studies on parametric reconstruction follows. For a more detailed description see Commandeur [2012].

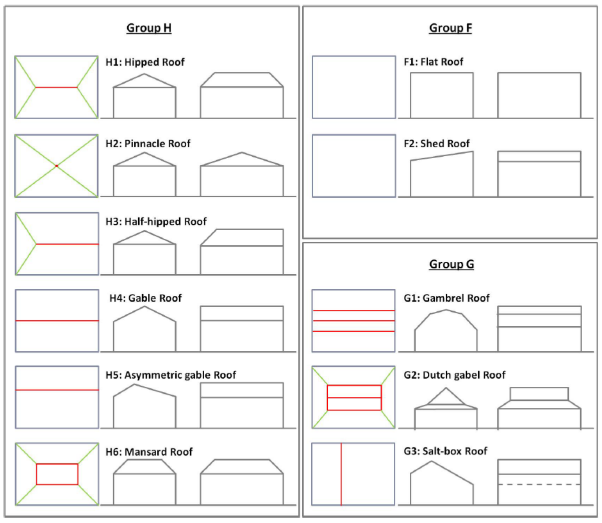


Figure 3.2: Overview of standard roof shaped from Huang et al. [2011]

An early study on automatic building reconstruction using parametric roof shapes is Haala et al. [1998]. In this method the reconstruction consists of two main steps. First, the building footprint is partitioned into rectangles by extending footprint edges. Subsequently, the best fitting primitive and its parameters, such as height and slope, are found for each rectangle. Together these parameterized primitives compose the roof. Kada and McKinley [2009] and Lafarge et al. [2010] improve the partitioning, allowing quadrangular and triangular cells. The DSM is already exploited in the process of footprint partitioning. Commandeur [2012] combines the decomposition of building footprints with pointcloud segmentation. By doing this, the different roof shapes that are not apparent from the footprint can be identified. In addition, some of these studies make use footprint generalization to suppress footprint details that have a negative impact on the reconstruction.

Taillandier [2005] has a different approach in which the footprint is not partitioned. Instead, a uniform eaves height is assumed and a slope angle for every edge is retrieved. If no tilted roof segment passes through the footprint edge, a gabled roof is assumed. Huang et al. [2013] does not use building footprints at all, but detects buildings in the LiDAR data. Primitives fitting the building are found and merged to retrieve a building model (Figure 3.3).

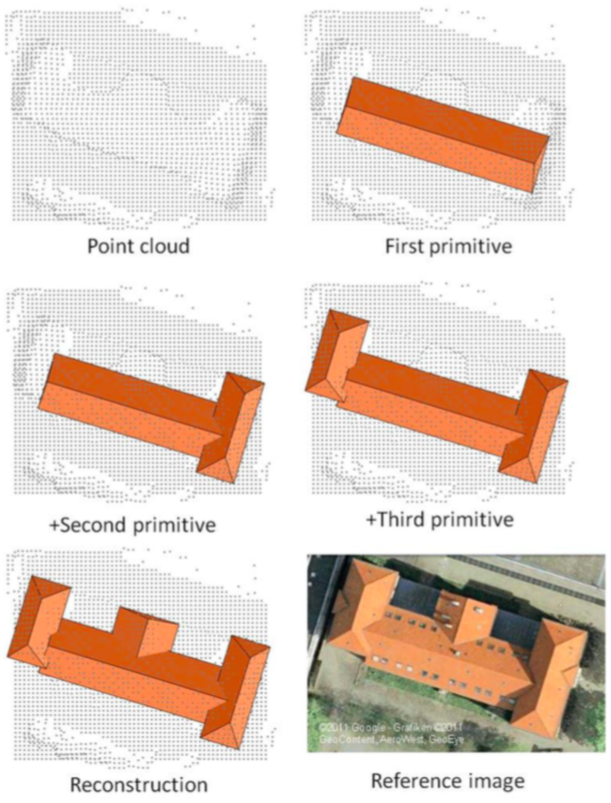


Figure 3.3: Reconstruction by merging primitive roof shapes [Huang et al., 2011]

The model-driven approach of reconstruction based on parametric roof shapes has several advantages. The resulting model will always be watertight, meaning that no gaps and overlapping segments are present. Furthermore, the created shapes will visually make sense, since they adhere to some predefined model. Finally, the estimation of parameters generally does not require much computational power [Tarsha-Kurdi et al., 2007b]. The major downside is that the shape of the building is limited to the predefined primitives. Generally, details like chimneys and dormers are therefore not incorporated. In the case only footprint partitioning is used, roof features can not be reconstructed if they are not apparent from the footprint.

3.2 RECONSTRUCTION BASED ON SEGMENTATION

The idea of reconstruction based on segmentation is that by segmenting a pointcloud or DSM into planar segments, the different roof segment can be identified (Figure 3.4). This approach is data-driven, since no particular shapes for the model are assumed in advance. The main advantage of a data-driven approach is that there are no predefined limitations on the shape of the building, other than that it should be a polyhedron [Tarsha-Kurdi et al., 2007b]. There are various ways to segment a pointcloud or DSM into planar regions [Sampath and Shan, 2010]. Hough transform (§ 3.2.1) and RANSAC (§ 3.2.2) are methods that detect planes within the pointcloud directly. Image segmentation (§ 3.2.3) and clustering (§ 3.2.4) approaches work by grouping points or segments with a similar orientation, defined by their normal vectors. Therefore, the normal vectors need to be retrieved in advance, for these two methods. Normal vectors for each point can be retrieved by fitting a plane through the points in its neighborhood (§ 2.5). Another option is to conduct a PCA on the points in the neighborhood.

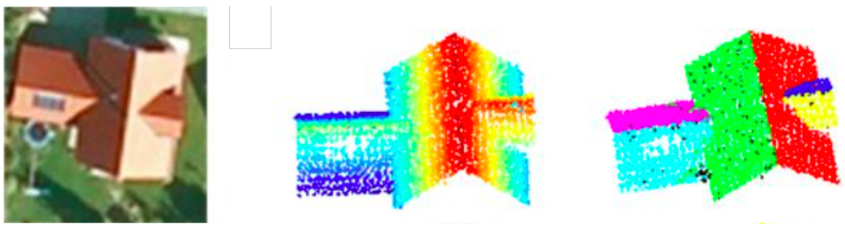


Figure 3.4: Segmented roof: Aerial photo (left), pointcloud sorted by height (middle) and pointcloud segmented (right) [Fan et al., 2014]

3.2.1 Hough transform

Hough transform can be used to detect multiple planes at once in a pointcloud (see § 2.5.5). The pointcloud of an entire building or part of it can thus be processed at once to find different planar segments. Vosselman and Dijkman [2001] combine the planes detected with 3D Hough transform with information from building footprints to detect roof segments. Tarsha-Kurdi et al. [2007a] and Novacheva [2008] detect roof segments directly with 3D Hough transform. The main drawback of 3D Hough transform is that it is computationally expensive, especially when the step size for the plane parameters is small. On the other hand a larger step size reduces the quality of the result. It also important to realize that with Hough transform and RANSAC plane models are retrieved, and not the actual segment (group of points) itself. A segment can be retrieved by taking all inliers for a particular plane. Inliers are points within a certain distance of the plane. However, commonly some of the inliers are actually part of another roof plane. These point should thus be removed in another step Tarsha-Kurdi et al. [2007a].

3.2.2 Random sample consensus

RANSAC can be used to find the best fitting plane for a pointcloud (see § 2.5.4). Considering a pointcloud of an entire building, generally more than one roof plane should be detected. Therefore the entire RANSAC procedure should be used iteratively in a substractive manner [Sampath and Shan, 2010]. After finding the best plane model using RANSAC, all points belonging to this model are removed from the pointcloud. This ensures that a different plane is found in the next iteration. [Tarsha-Kurdi et al., 2007a] extended the RANSAC algorithm to improve the detection of roof planes. It does so by considering the standard deviation of the inliers as well. Additionally, the set of points found by the RANSAC algorithm is improved. This is done by removing inliers belonging to other roof planes and adding outliers belonging to the detected roof plane. The advantage of hough transform over RANSAC is that multiple models can be found directly and no iterative approach is required. However, [Tarsha-Kurdi et al., 2007a] reports that the iterative RANSAC approach is superior to 3d Hough transform both in terms of speed and quality.

3.2.3 Image segmentation

An overview of image segmentation methods is given in § 2.3. These algorithms are developed to work on images, such as a gridded pointcloud. Most algorithms can however be adapted to work directly on a pointcloud. By segmenting points based on their normal vector, planar segments can be found. The approach developed by READAAR uses thresholding (see § 2.3.1) on the normals of the gridded LiDAR data. The main reason for using this method is the high efficiency of thresholding. Many studies such as Rottensteiner and Briese [2003], Dorninger and Pfeifer [2008], Rottensteiner [2010], Nex and Remondino [2012] and Awrangjeb et al. [2013] use region growing to detect or improve planar segments. The seed points or seed regions are often selected based on their low curvature values. Awrangjeb et al. [2013] finds seed regions by making use of edges detected in an orthophoto of the building. The advantage of region growing is that the segmentation is region-based. The points belonging to each segment will therefore always be located in the same connected area of the pointcloud. This means that it is unlikely that the segment contains points of other roof planes. The main drawback of region growing is that points are added to regions iteratively and not at once. This makes region growing a computationally expensive method.

An alternative to grouping points directly to find planar segments, is to group color segments resulting from image segmentation instead. In this case the image segmentation method is thus applied on the color values, instead of the normal vectors. For the resulting segments, consisting of multiple points, the orientation can be determined by plane fitting (§ 2.5). Both Khoshelham [2005] and Rottensteiner [2010] use watershed segmentation to retrieve initial segments. Furthermore these studies use a LiDAR derived DSM to split and merge the resulting segments. Rottensteiner [2010] additionally exploits the segmentation result of multiple images from different viewpoints.

3.2.4 Clustering

Like image segmentation algorithms, clustering algorithms can be used to group points based on their normal vectors and find planar segments. An overview of the most relevant clustering methods is given in § 2.4). Sampath and Shan [2006] uses an iterative combination of k-means and density based clustering methods. Liu and Xiong [2008] on the other hand uses cell mean shift, an adoption of the mean shift algorithm. Sampath and Shan [2010] detects roof segments with fuzzy k means clustering. The number of clusters is determined in advanced using a potential-based approach. A disadvantage of clustering based on orientation only is that different roof segments with the same orientation cannot be separated. This could be solved in a subsequent splitting step. Another option is to include the location of points (x, y, z) in the feature vector. The downside of this is that large roof segments are likely to be split up, due to large variations in the locations of its points. The general advantage of clustering methods is that it can be determined simultaneously to which cluster each data point belongs. Therefore, clustering algorithms can be parallelized, making them very efficient.

3.3 DSM SIMPLIFICATION

The DSM of a building can already be thought of as a detailed building model. A simplified polyhedral building model can thus be retrieved by simplification of the DSM. Often the DSM is converted to a mesh, which in turn is simplified. Many algorithms for mesh simplification exist [Cignoni et al., 1998]. Wahl et al. [2008] uses an edge-collapse approach for building modelling. This approach of building reconstruction gives an efficient way to reduce the amount of information that is stored. Therefore, DSM simplification is often used for real-time visualization purposes [Haala and Kada, 2010]. It is thus a good technique for acquiring a watertight model of the rough shape of the building. After DSM simplification, planar roof segments could be retrieved by grouping neighboring triangular faces with the same orientation. However, during the simplification of a building mesh information will be lost. Therefore, it is less suitable to find the exact outline of planar roof segments. For this purpose it is more logical to search for these planar segments directly in the pointcloud data.

3.4 CONCLUSIONS

The majority of methods is developed for LiDAR pointclouds. Even though these methods are applicable on image-based pointclouds, they are not designed for them. The presence of noise, blunders and gaps in image-based pointclouds is therefore likely to cause reconstruction errors [Nex and Remondino, 2012]. Reconstruction based on parametric roof shapes is particularly useful to retrieve watertight building models. However, this approach is less suitable for the accurate detection of variably shaped planar roof segments. Reconstruction by DSM simplification could be used to retrieve planar roof segments. However, information is lost during the simplification process. Therefore, the segmentation-based approach is the most promising option. This data-driven approach allows the detection of planar segments of any shape. Different segmentation methods have been exploited for building reconstruction. Hough transform, region growing and RANSAC are relatively computationally expensive, making them unsuitable for large scale applications. Clustering and image segmentation methods, like thresholding and watershed, are relatively efficient. These methods are thus of particular interest for large scale extraction of roof segments. Finally, a possible way to exploit color information is to start with an image segmentation algorithm for retrieving color segments.

4 METHODOLOGY

As described in chapter 3 a lot of research is conducted on roof reconstruction. However, the majority of these approaches are unpractical for processing large areas such as entire municipalities. Therefore, READAAR designed its own fully automatic, efficient and scalable solution to extract 3D roof segments based on LiDAR data (see § 1.1). The use of LiDAR data, however, has some fundamental limitations as discussed in the introduction of chapter 3. These limitations could potentially be solved by using aerial stereo imagery instead.

4.1 OVERVIEW OF THE PROPOSED STEREO-BASED WORKFLOW

The goal of this research is to develop a stereo-based scalable method for the fully automatic and efficient extraction of roof segments. Keeping these requirements in mind, the workflow in Figure 4.1 is designed. In addition to the stereo imagery, a building footprint dataset is required as input. These footprints are essential, since they allow to determine which areas in the image are potentially roof segments. Additionally, they are used to couple the roof segments to a particular building.

Usually the rectification, matching and pointcloud retrieval of aerial stereo imagery are pre-processing steps. All the studies cited in 3 exploit such pre-processed datasets with height information. None of them uses an integrated approach for stereo matching and roof segment extraction. Dense pointclouds or DSMs for large areas can be created using software packages such as SURE [Rothermel et al., 2012]. In case large areas are pre-processed, building footprints can be used to cut out the right part of the pointcloud or DSM, and find the roof segments. However, the pre-processing of many stereo images is computationally very expensive and requires a lot of storage space. This is a problem, especially when stereo images are updated frequently. To give a very rough indication, creating a pointcloud using images with a resolution of 10cm for an area of 1km^2 could take some minutes (using a HP ZBook Studio G4 with an i7 2.8GHz quadcore from Intel). Storing the resulting pointcloud as a las file requires about 3GB. For municipalities this will thus be in the order of terabytes and hours. Since pre-processing should be done for many municipalities that are desirably frequently updated, pre-processing is not feasible for READAAR. Especially, since dense matching software package like SURE are commercial and not free of charge. Therefore, buildings are cut directly from a stereo pair and stereo matching is done separately for each building. This improves the efficiency, because only buildings are matched and other areas are omitted. Furthermore, this ensures easy scalability since the memory footprint of the approach remains small. After every building that is processed, the matching result can be forgotten again.

The second part is to retrieve roof segments from the matched stereo images. The most straightforward approach would be to use the pointcloud resulting from the stereo matching directly, in the same way as READAARs LiDAR-based approach (see § 1.1). However, in this approach color information is not exploited, resulting in the same edge problem the LiDAR-based approach is dealing with. This problem is likely to be even larger since the matching output is generally noisy. A larger neighborhood is thus necessary to retrieve accurate normal vectors, amplifying the edge problem. Therefore, a two step approach is used, inspired by the work of [Rottensteiner, 2010]. First, color segments are retrieved from the aerial image. Second, these color segments are clustered based on their orientation to find roof segments. In this approach the edges of segments are determined based on color. Furthermore, it is theoretically more efficient for two reasons. Firstly, no neighborhoods of points are required to determine the orientation. The orientation is determined by fitting a plane through all points within each color segment. As a result every point will only be used once in the process of fitting planes. Secondly, the input for the clustering algorithm is much smaller, since segments are clustered and not points. The assumption made with this approach is that regions of pixels with a similar color share the same orientation.

In the remainder of this chapter the different processing steps are described in more detail on the basis of one example building. § 4.2 will describe how a stereo pair is clipped using a building footprint. § 4.3 and § 4.4 are about the rectification and matching of a stereo pair respectively. § 4.5 will discuss the image segmentation to retrieve color segments. Subsequently, § 4.6 concerns the fitting of planes through these color segments to determine their orientation. § 4.7 is about filtering out color segments that are not part of the roof. The clustering of color segments based on their orientation to form roof segments is discussed in § 4.8. Finally, § 4.9 describes the reconstruction of roof segments from 2D image segments to planar 3D polygons in world coordinates.

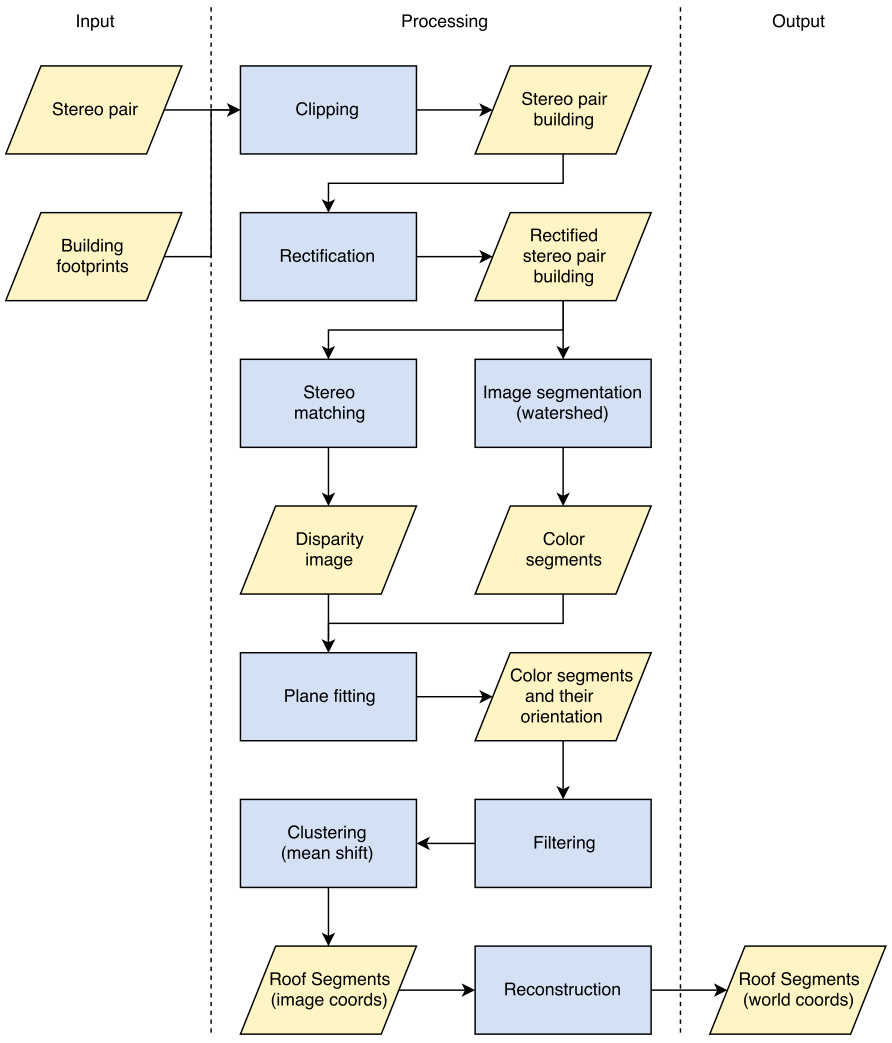


Figure 4.1: Workflow for the extraction of roof segments from stereo imagery and building footprints

4.2 CLIPPING STEREO PAIR

First of all, a stereo pair needs to be selected on which the particular building is visible. To retrieve a stereo pair of the building, the two stereo images closest to the building are selected. The distance is measured in 2D (x, y), from the centroid of the building to the position of the camera. The building centroid is calculated from the footprint polygon. The x, y position of the camera is known, since they are part of the extrinsic parameters (see § 2.1.3). The second step is to clip the building from both stereo images. In order to know which pixels in the images represent the building, the footprint its world coordinates are converted to pixel coordinates as explained in § 2.1. The z coordinates of the building footprint are taken a ground level, the roof however is located higher. Since the building on the images is probably not located exactly below the camera, a shift is present between the footprint and the roof. This shift becomes larger when the angle to the building increases and the building is higher. Therefore, a buffer is added to the image coordinates, ensuring that the entire roof is captured. Theoretically, the buffer size and shape could be optimized using the angle to the camera and height of the building. Currently, however, the buffer simply has a fixed size around the building footprint. The minimum and maximum row and column values are used to retrieve the rectangular image areas containing the building. An example of the resulting clipped stereo pair is given in Figure 4.2.

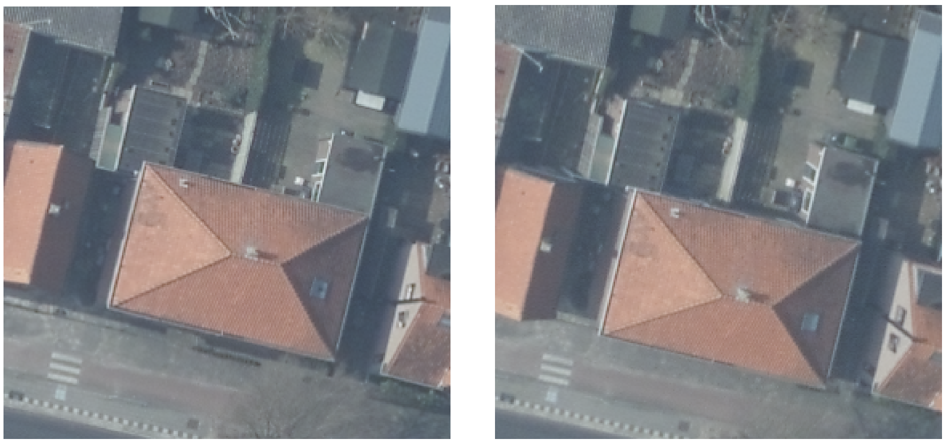
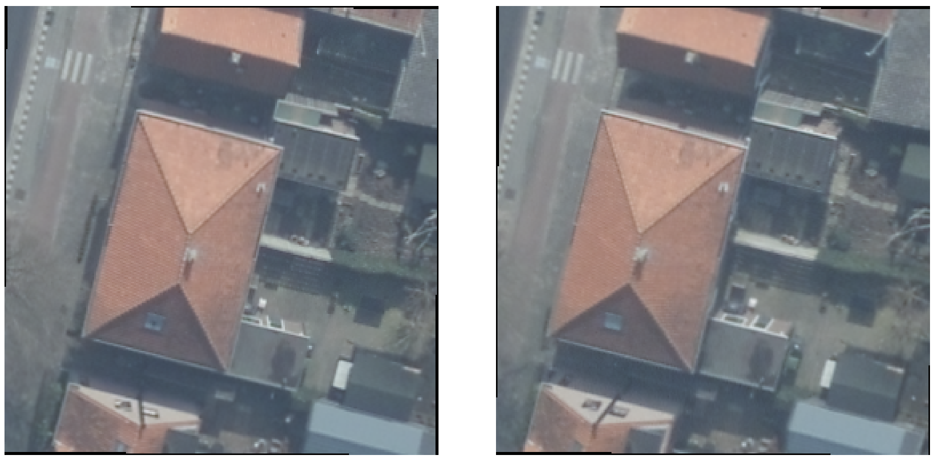

(a) First view **(b)** Second View

Figure 4.2: Clipped stereo pair

4.3 RECTIFICATION

Prior to the stereo matching, the stereo pair needs to be rectified (see § 2.2.1). Rectification is done using the algorithm developed by Fusiello et al. [2000]. The input for the algorithm are the two original PPMs (see § 2.1.4). The output are two rectified PPMs and the projective transformation matrices required to rectify the images. The rectified stereo pair for the example building is shown in Figure 4.3.



(a) Left view (b) Right View

Figure 4.3: Rectified stereo pair

4.4 STEREO MATCHING

As described in § 2.2.2 many stereo matching algorithms exist. For this particular purpose a stereo matching algorithm is required that, in the first place, is efficient. Secondly, the resulting disparity map should be as complete and as accurate as possible. Therefore, the SGM algorithm is chosen, since it is efficient whilst still achieving relatively good results.

4.4.1 Pre-processing stereo images for matching

Matching algorithms like SGM generally work on a single image band and not on images with multiple bands. Since aerial stereo imagery generally has three bands (RGB), a conversion from three bands to one band is required. Several options exist for this conversion. The most straightforward option is to convert the color image to a grayscale image by calculating the intensity value for every pixel. The downside of this method is that color information is lost in this process, because different colors can have the same intensity. To partly deal with this problem, a color to grayscale conversion based on PCA is exploited. [Dikbas et al., 2007] give an explanation of the method and use it for color edge detection. In this approach the first principal component is used to create a grayscale image (Figure 4.4).

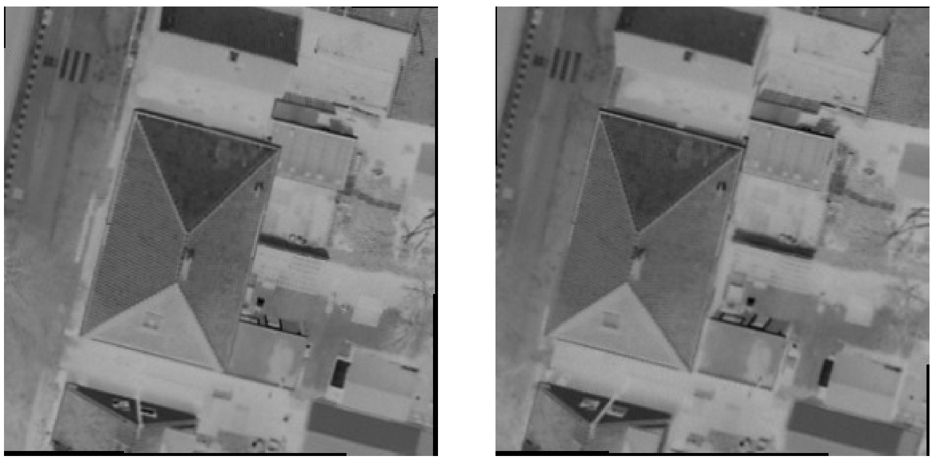

(a) Left view (b) Right View

Figure 4.4: Rectified stereo pair after conversion from a three- to one-band image using principal component analysis

4.4.2 Post-processing the disparity map

The disparity map for the left image resulting from SGM can be seen in Figure 4.5a. Since the disparity map resulting from the matching often contains noise, some post-processing steps are usually executed. Two common post-processing steps are speckle and median filtering. Speckle filtering divides the image in connected groups of pixels with the same disparity value. If such a group has a size below a certain threshold, it is removed. In this way matching errors are removed from the disparity map. Median filtering is used to smooth the data. Median filtering is done by shifting a window with a specified size over the image. The center pixel gets the median value of all pixels in the window. The result of speckle and median filtering can be seen in Figure 4.5b.

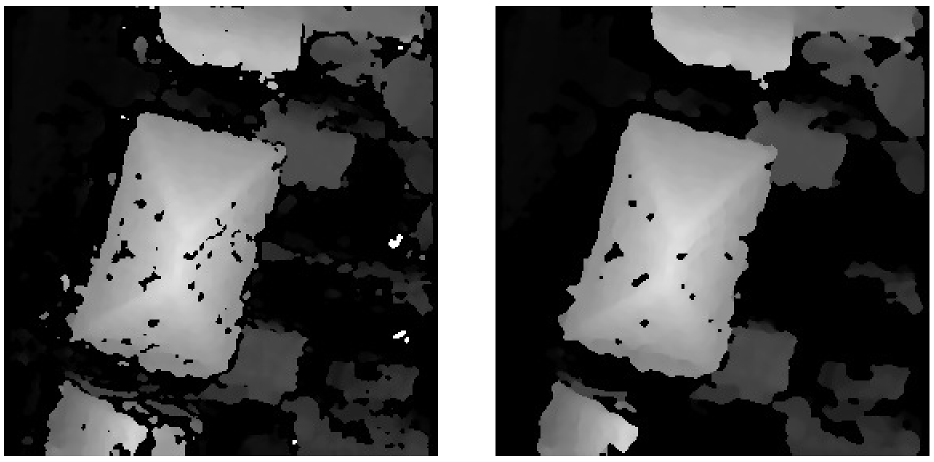

(a) Initial (b) After speckle and median filtering

Figure 4.5: Disparity maps of the left image retrieved by semi-global matching

4.5 IMAGE SEGMENTATION TO FORM COLOR SEGMENTS

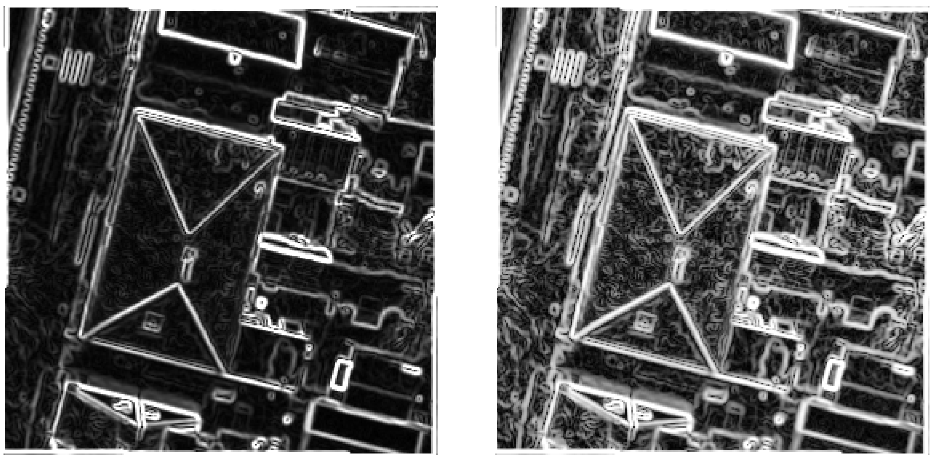
A great variety of image segmentation algorithms exists, a short overview of some of the most common approaches is given in § 2.3. Since the segmentation step is used to determine edges more accurately, it is important that the segmentation algorithm makes segments that have their boundaries on color edges. Additionally, the number of segments is not known in advance and efficiency is important. The watershed of gradient magnitude approach (§ 2.3.4) fulfills all these requirements and is therefore picked as segmentation algorithm.

4.5.1 Gradient magnitude of a color image

As explained in § 2.3.4, the watershed algorithm works on a grayscale image of the gradient magnitude and not directly on the color image. The gradient magnitude is calculated using the approach first proposed by Zenzo [1986]. In this approach the color image is treated as a vector field, meaning that the RGB values of each pixel are seen as a vector. For each pixel the Jacobian matrix **J** is constructed from the partial derivatives. These are the individual gradients of the RGB bands with respect to x and y (equation 4.1). The gradient magnitude is determined by multiplying the Jacobian with its transpose and taking the greatest eigenvalue of the resulting matrix. The approach is similar to PCA, the greatest eigenvalue and corresponding eigenvector give the direction and magnitude of the greatest color change. With this technique, edges are detected even if the intensity gradient is zero. For a more detailed explanation refer to Burger and Burge [2016]. The individual gradients at each pixel for the RGB bands with respect to x and y are determined using the Sobel operator. The gradient magnitude of the left image is shown in Figure 4.8a. In order to enhance contrast in homogeneous color areas, histogram equalization is applied on the grayscale image of the gradient magnitude (see Figure 4.8b). Furthermore, the image is smoothed with a 3x3 Gaussian filter prior to the color edge detection. The goal of Gaussian smoothing is to partly remove small local color variations that exist within a single roof segment. This should eventually reduce over-segmentation of the image.

$$\mathbf{J} = \begin{bmatrix} \delta r/\delta x & \delta r/\delta y \\ \delta g/\delta x & \delta g/\delta y \\ \delta b/\delta x & \delta b/\delta y \end{bmatrix} \tag{4.1}$$

$$\mathbf{Q} = \mathbf{J}^T\mathbf{J} \tag{4.2}$$



(a) Initial (b) After histogram equalization

Figure 4.6: Edge strength found by the color gradient magnitude before and after histogram equalization

4.5.2 Watershed segmentation of the gradient magnitude

The watershed segmentation of the gradient magnitude is given in Figure 4.7. As explained in § 2.3.4 over-segmentation is a common drawback of the algorithm. In the example building this is clearly visible. Especially for the sunlit roof segments, in which contrast is high. However, the over-segmentation is not necessarily a problem. In this particular case this is handled by the two step segmentation approach. In this approach color segments are clustered in a second step based on orientation. So the goal of the watershed segmentation is not to retrieve the roof segments directly. It is to retrieve smaller color segments that likely belong to the same planar surface. Therefore, the orientation can be determined robustly for entire segments consisting of multiple pixels. Rather than determining the orientation for each pixel individually, which can be problematic in noisy and incomplete image-based point-clouds.

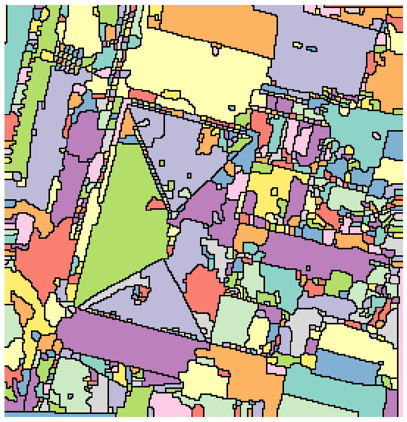
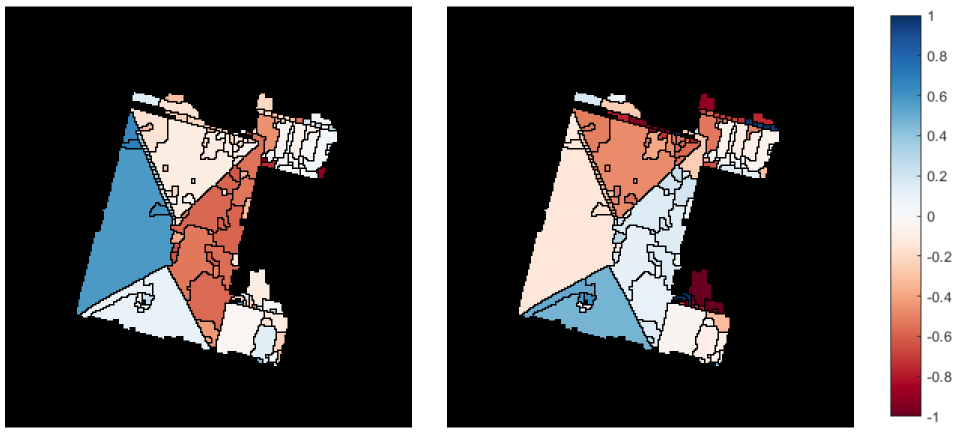


Figure 4.7: Similar color segments retrieved with watershed segmentation of the gradient magnitude

4.6 PLANE FITTING TO DETERMINE THE ORIENTATION OF THE COLOR SEGMENTS

The orientation of each color segment is determined by fitting a plane through its 3D (u, v, d) points. Figure 4.8 shows the x and y component of the normalized normal vectors for each segment. Together these components uniquely define the 3D orientation. Plane fitting is done using the basic OLS approach described in section § 2.5. The reason for this is the high efficiency of the method. The plane model is retrieved both in image and world coordinates. For the orientation to have meaning, the plane model should be in world coordinates. This is desired for the clustering, as described in § 4.8. However, for reconstruction § 4.9 it should be image coordinates. Converting a plane model from image to world coordinates and vice versa is rather straightforward. The important insight is that three non collinear points define a plane uniquely. Therefore, three random non collinear points can be selected and converted as explained in § 2.2.3.



(a) x component **(b)** y component

Figure 4.8: Normalized normal vector of the color segments

4.7 FILTERING COLOR SEGMENTS

Not all color segments are part of the roof of the building in question. Some segments might for example represent walls, the ground or the roof of another building. Therefore, some of the segments should be filtered out before clustering the color segments into roof segments. The first filtering step uses the building footprint. For each segment, the percentage of matched points falling within the building polygon is determined. Note that it cannot be checked whether pixels that are not matched fall within the building polygon. The reason for this is that the disparity is required to retrieve world coordinates, as described in § 2.2.3. If more than a specified percentage of the points falls outside the building footprint, the segment is filtered out. Segments falling for more than 50% outside the building footprint are filtered out in Figure 4.8. Based on the planes fitted through the color segments, the slope and average height of each segment can be determined. This information can be used to filter out wall and ground segments. Walls are generally vertical, whilst roof are not. All segments with a nearly vertical slope ($> 70°$) can thus be removed. Ground segments can be identified based on a height threshold. A logical threshold for filtering ground segments could be the minimum height of a single floor ($< 2m$). Figure 4.9 shows the result of filtering out these segments.

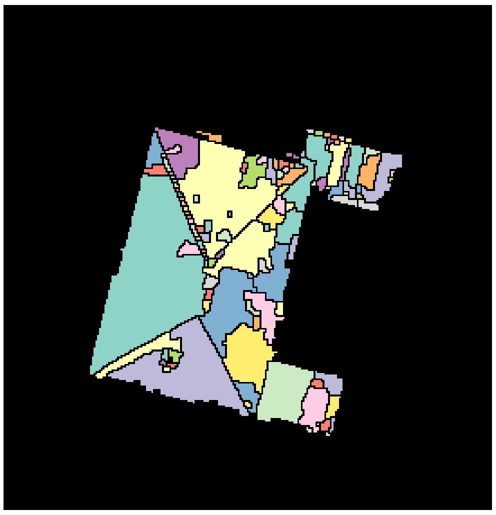


Figure 4.9: Roof segments after filtering out segments outside the building footprint, and wall and ground segments

4.8 CLUSTERING COLOR SEGMENTS BASED ON ORIENTATION

To find planar roof segments, the color segments that are part of the same planar surface should be grouped together. This can be done by clustering color segments based on their orientation using a clustering algorithm. To determine whether color segments are part of the same planar surface, the orientation of their plane models is used. The orientation of each plane is given by its normal vector (see § 2.5). Each normal vector consists of three components (nx, ny, nz), along the x,y and z axis respectively. The normal vectors of different planes cannot be compared directly, as they should be normalized first. Furthermore, it should be ensured that every normal is pointing upwards. This can simply be done by taking the negative in case the vector has a negative z component. The normalized upward pointing normal vectors are the input for the clustering algorithm. Thus, only the normal vectors of the segments are clustered. The resulting clusters consist of normal vectors, each linked to a color segment, with a similar orientation. Each of these clusters represents a roof segment. Several things should be taken into account when selecting a clustering algorithm. First of all, the number of clusters (roof segments) is not known in advance. Second, it can be expected that clusters vary in size. Cluster size in this case refers to the variation of the orientation within a cluster, not to the number of segments in a cluster. The reason for this is that some roof segments consist of many small color segments, whilst others of one or several larger segments. These differences are the result of the image segmentation described in § 4.5.2. Many small segments are likely to show a larger variation in orientation than some large segments. Theoretically, cluster sizes will thus vary for different roof segments. Based on this knowledge, the mean shift clustering algorithm discussed in § 2.4.3, is selected.

4.8.1 Adding clustering weights related to segments size

The color segments that are used as input vary in size. This potentially causes problems in retrieving accurate clusters. Each color segment, independent of size, has the same impact on the probability density function used in the mean shift algorithm. Local optima in the probability density function are therefore largely determined by the many small segments, rather than the few large segments that compromise most of the roof. This is especially problematic since the orientation value of smaller color segments is likely to be less accurate. For smaller segments, generally less points are available for plane fitting, resulting in a less accurate orientation. To retrieve accurate roof segments, clustering weights should be introduced based on segment size. Most cluster algorithms, like mean shift, can easily be adapted to deal with weights. Data points are simply put in multiple times depending on their weight. For every hundred pixels in a segment an extra copy is added. The reason not to make a copy for every pixel, is to keep the efficiency advantage of clustering segments instead of pixels. The roof segments resulting from weighted mean shift clustering are visualized in 4.10.

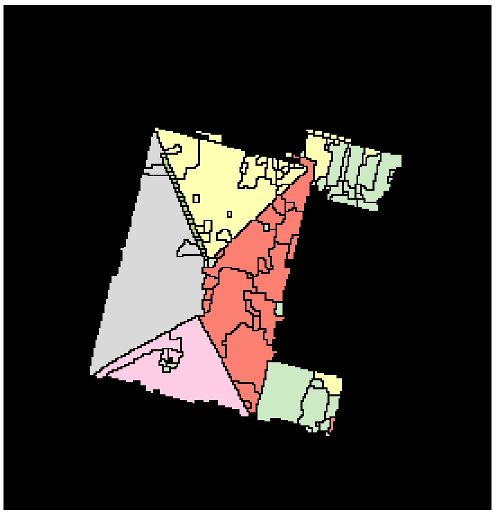


Figure 4.10: Clusters of color segments based on their normalized normal vectors

4.8.2 Filling edges in the cluster image

The different color segments that are clustered together are still separated by their old edges in the cluster image. These edges need to be filled in order to create complete roof segments. Edge filling is done by investigating the 8 neighbors for each edge pixel. If only one cluster ID is found among the neighbors, the edge pixel gets that cluster ID. If multiple cluster IDs are found, the edge pixel remains an edge pixel. The result of edge filling can be seen in Figure 4.11.

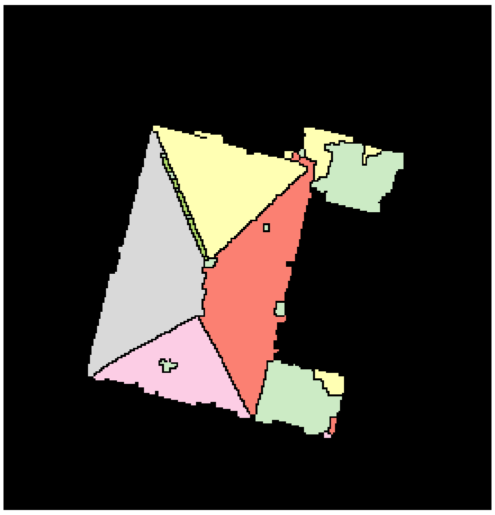
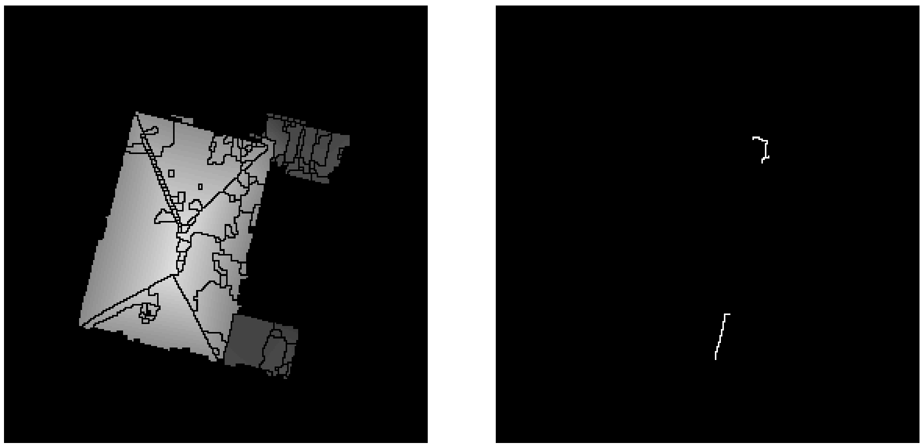


Figure 4.11: Roof segments resulting from edge filling of clustered color segments

4.8.3 Dealing with height jumps

A limitation of clustering based on orientation only is that height jumps are not identified. Consider two neighboring color segments with the same orientation but with a height jump in between. After clustering, these color segments will belong to the same cluster. This is obviously wrong, since they should form two separate clusters. In order to solve this problem, an image of the height jumps is created. This image is used to split roof segments that were incorrectly clustered. The height jump image can be viewed of as a constraint image, containing predefined boundaries between segments. The height jumps are detected using the color segments and their corresponding plane models. For each pixel belonging to a color segment, a disparity value is calculated based on the plane model. The result of this is visualized in Figure 4.12a. Subsequently, height jumps are detected by shifting a 3x3 window over the disparity image and taking the range of all values. A higher range corresponds to a larger difference in height. Using a simple threshold value, height jumps are identified (see 4.12b). In the case of this particular house, the detected height jumps are of no use. The two light green areas in Figure 4.11 were already separated from the red segment, since they have a different orientation.



(a) Disparity based on plane models (b) Height jumps roughly $> 0.5m$

Figure 4.12: Identification of height jumps based on disparity values

4.8.4 Disjoint clusters

In the resulting roof segment image, disjoint segments can still belong to the same cluster. Meaning that they still share the same ID, referring to a particular cluster. Such situations occur when two roof segments have a similar orientation. For example, the two flat roof segments on the right side in Figure4.11 are both depicted in light green, because they belong to the same cluster. This is undesirable, since they represent two different roof segments. To solve this issue, each disjoint region is given its own cluster ID using a connected component algorithm.

4.9 RECONSTRUCTION OF ROOF SEGMENTS

The reconstruction of roof segments is the conversion from 2D image segments to planar 3D polygons in world coordinates. The first step is to retrieve a plane model for every roof segment in the same way it is done for the color segments (see § 4.6). Second, the roof segments are vectorized. For each segment, a polygon is constructed from the outermost ring of pixels. This ring is not further simplified, meaning that all its vertices are kept. The coordinates of the polygons are image coordinates (u, v, d). Disparity values are determined based on the plane model related to each segment. These coordinates are converted to world coordinates (x, y, z), as described in § 2.2.3. The segments in world coordinates can be cut with the building footprint polygon. Cutting the polygons is done in 2D (x, y). For each segment, the intersection with the footprint polygon is taken (see Figure 4.13). Figure 4.14 shows the resulting 3D roof segments. Segments smaller than $1m^2$ are filtered out, since they likely do not represent real roof segments.

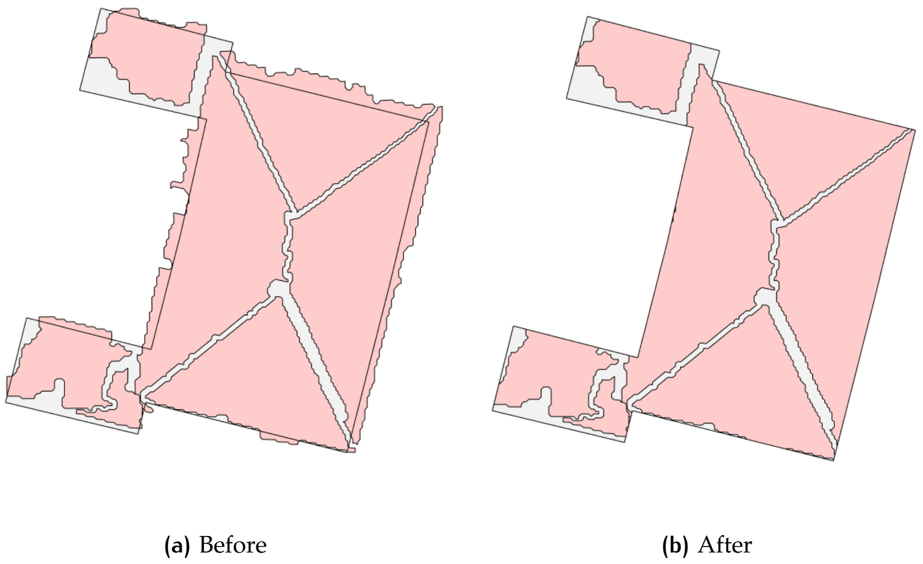

(a) Before **(b)** After

Figure 4.13: Cutting the vectorized roof segments with the building footprint

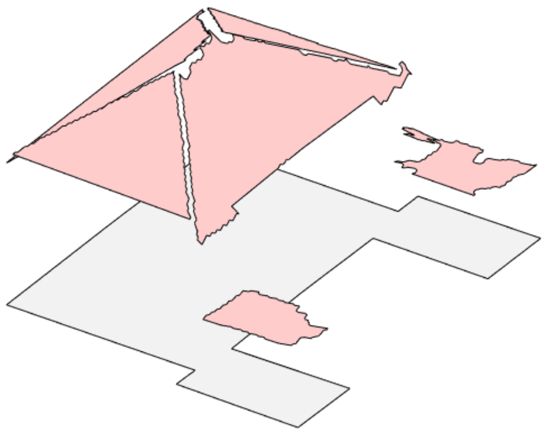


Figure 4.14: Planar 3D polygons representing the final roof segments (red) and building footprint (gray)

5 IMPLEMENTATION & RESULTS

This chapter addresses the software implementation and results of the workflow described in chapter 4. Implementation details of the developed prototype are given in § 5.1. The datasets used for the experiments are described in § 5.2. The particular area that is processed and analyzed is presented in § 5.3. Subsequently, § 5.4 describes which metrics are used to analyse the quality of the resulting roof segments. In § 5.5, the resulting roof segments are presented and analyzed. Finally, a comparison with other methods is given in § 5.6.

5.1 PROTOTYPE DEVELOPED

The prototype software for the extraction of roof segments from stereo imagery is mainly created in MATLAB. Data storage and management is done in PostgreSQL with the PostGIS extension. Additionally, Python is used for communication with the databse. In the following subsections, these three components will be discussed in detail.

5.1.1 PostgreSQL & PostGIS

The database management system used in this project is PostgreSQL with the PostGIS extension for spatial data. Figure 5.1 shows the three tables in which all input data is stored (Imagery, camera intrinsics and building footprints). Furthermore, one output table is created for storing the roof segments. The schema is a simplification of reality, only the fields that are relevant are shown. The only user input for the developed prototype is a list of BAG ids and a year for the imagery. First, the BAG table is queried to get the footprint. Using the centroid of the building footprint, the two closest images, forming a stereo pair, are selected. Since the stereo imagery table contains data about 70.000 images, the location column has a spatial index. The stereo imagery table is linked to the camera table, allowing to get the intrinsic parameters as well. Note that the file name and location are retrieved from the database and not the actual images. The actual images are stored as tif files on another server. The resulting roof segments are stored in a table with their quality metrics. The segments are also linked to the BAG (building footprint) table.

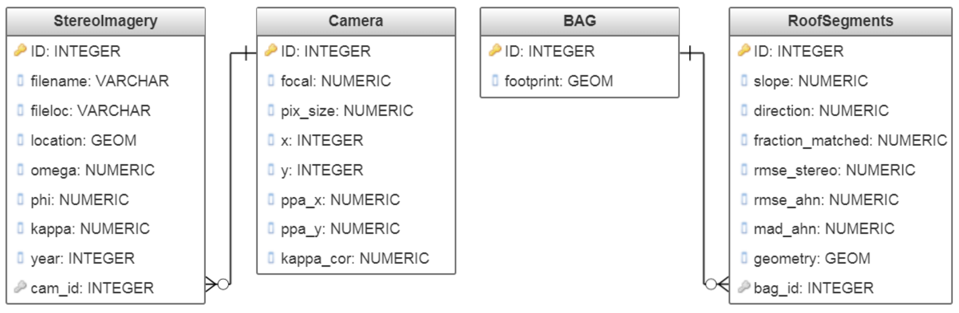


Figure 5.1: Database schema of the input data

5.1.2 Python

Python is mainly used for the communication with the postgreSQL database. The reason for this is that the *psycopg2* library provides a convenient framework for communication with the database. Furthermore, python and MATLAB integrate rather well, making it is easy to convert the input for use in matlab.

5.1.3 MATLAB

The main tool used during this project is MATLAB. The company READAAR generally develops in MATLAB enabling this project to be integrated easily. Furthermore, large matrix calculations are common in the field of photogrammetry and computer vision. Since MATLAB is particularly designed for this purpose, it is a good choice as well. Examples of advantages are easy visualization and efficient matrix computation. The major downside, however, is that it is not open source. Implementations for several algorithms were already available, either within MATLAB toolboxes or from external sources. Other algorithms were implemented by myself. The following functions from MATLAB toolboxes we used:

COMPUTER VISION SYSTEM TOOLBOX:

- *disparity* is used for the disparity map calculation with SGM (see § 4.4).

IMAGE PROCESSING TOOLBOX:

- *medfilt2*: For median filtering, which is a post-processing step for the disparity map (see § 4.4.2).
- *bwareaopen:* For speckle filtering, which is a post-processing step for the disparity map (see § 4.4.2).
- *imgaussfilt:* For guassian filtering, which is a pre-processing step in the color edge detection (see § 4.5.1).
- *histeq:* For histogram equalization, which is a post-processing step in the color edge detection (see § 4.5.1).
- *watershed:* For the watershed segmentation SGM (see § 4.5.2).
- *bwconncomp:* For the detection of clusters consisting of disjoint regions (see § 4.8).
- *bwboundaries:* For the vectorization of the raster roof segments in the reconstruction step (see § 4.9).
- *polyarea:* For determining the size of the roof segments (see § 4.9)

MAPPING TOOLBOX:

- *polybool:* For cutting the roof segments with the BAG. This operation is a intersection of two polygons (§ 4.9) .

External implementations were used and adapted for the following algorithms:

- *rectification:* For rectification of the stereopair (see § 4.3). The implementation is provided in Fusiello et al. [2000].
- *color edge detection:* For retrieving the gradient magnitude required for watershed segmentation (see § 4.5.1). The code is retrieved from MathWorks file exchange. The implementation is based on the algorithm presented in Zenzo [1986].
- *mean shift clustering:* For clustering color segments (see § 4.8). The code is retrieved from MathWorks file exchange and is a basic implementation of this well-known algorithm.

Remaining algorithms were implemented by myself:

- *forward projection:* For the transformation of world to image coordinates. This is used to clip the stereo pair § 4.2.
- *plane fitting*. For retrieving planar 3D color and roof segments (see § 4.6 & § 4.9).
- *edge filling*. For removing edges between clustered color segments (see § 4.8.2).
- *height jump detection*. For detecting height jumps between different color segments (see § 4.8.3).
- *back projection*. For the transformation of image to world coordinates. This is used to filter out color segments located outside the BAG (see § 4.7) and to reconstruct the roof segments (see § 4.9).

5.2 DATASETS

5.2.1 Basisregistraties adressen en gebouwen

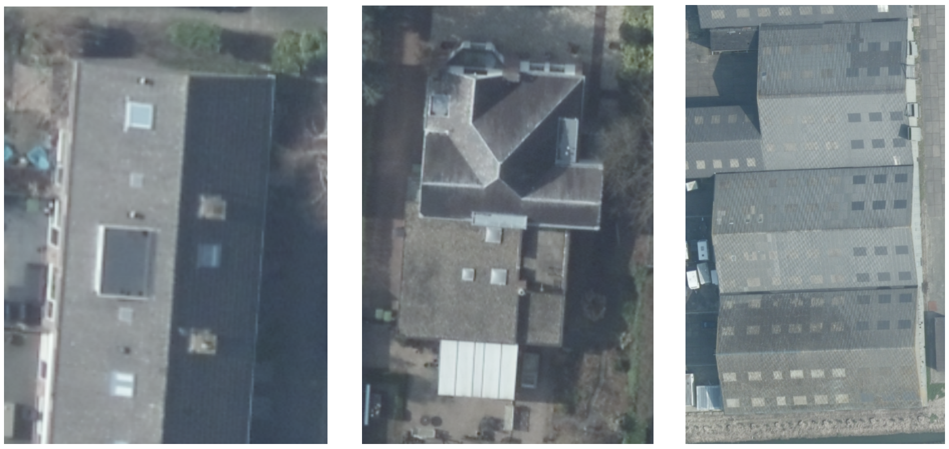
The basisregistraties adressen en gebouwen (BAG) is the registration of addresses and buildings in the Netherlands. The BAG dataset is managed by *het Kadaster*, which is a Dutch governmental organization. In this dataset every building is uniquely defined. In relation to this study, the most important feature in the BAG is the building footprint that is available for each building. The BAG contains demolished, planned and existing buildings. For existing buildings, the quality of the BAG is an absolute point precision of 20cm in urban areas and 40cm rural areas. The geometry of new building footprints is made available within 6 month after registering the building as *in use*.

5.2.2 Stereo10

Stereo10 is a dataset containing aerial stereo imagery of the entire Netherlands at a ground resolution of 10cm. The Stereo10 dataset is managed by Cyclomedia, which is a Dutch company. Cyclomedia is not collecting the imagery itself. Gathering of the images is outsourced to other companies such as Aerodata, Eurosense and Miramap. Therefore, the dataset contains imagery retrieved with different camera types. Stereo10 is yearly updated and mostly retrieved during winter and early spring, when weather conditions are good. This is the leafless season, giving maximum visibility of man-made structures. The particular area that will be processed during this research is captured by an ultracam xp camera. Imagery was gathered from an altitude of 1600m with the traditional 60% forward and 20% side overlap. These values ensure that every point is at least captured twice. Taking into account variations in roll, pitch and yaw of the camera, which is mounted to the airplane.

5.3 RESEARCH AREA

In order to asses the designed method and corresponding prototype, a research area is selected for which the roof segments are extracted. The research area for this study is the Dutch town Numansdorp. The town is located south of Rotterdam, in the Province of South-Holland. A map of the research area is attached in Appendix A. The map also shows the locations of the stereo images used for the extraction of the roof segments. A total of 11 images were used, divided over 3 different flight lines. In this way it is ensured the method can handle both side and forward overlapping stereo pairs. The size of the area for which segments are extracted is approximately 1 by $2km$ and contains 4087 buildings. The reason that this town was selected is that it is a rather average compact town with several different neighborhood types. Within the research area, three smaller areas are selected to be evaluated more thoroughly. These areas contain terraced houses, free-standing houses and industry respectively. Figure 5.2 shows some typical buildings for each of the three areas. The terraced houses are characterized by simple gabled roof, sometimes with a dormer. The free-standing houses have different types of complicated roof shapes with many dormers and roof objects. The industry is characterized by large simple segments. In Appendix B aerial imagery of the entire areas is presented, together with manually created ground truth (reference) data for the roof segments. As discussed in the next section (§ 5.4), the reference data is required to asses the quality of the resulting roof segments. This is also the main reason for selecting these three smaller areas, since manually creating reference data for the whole research area is not feasible. Furthermore, it allows to evaluate how the methods performs for these three different areas.



(a) Terraced **(b)** Free-standing **(c)** Industry

Figure 5.2: Example buildings for the 3 test areas

5.4 QUALITY METRICS

The performance of the designed method can be evaluated on different aspects. Rutzinger et al. [2009] gives a review of performance evaluation and discusses several quality metrics. These metrics are also used for the ISPRS benchmark on 3D building reconstruction. In Rottensteiner et al. [2014] the results of the ISPRS benchmark are presented. The methods of several different studies are compared. The two main things that are evaluated are the segmentation quality and geometrical accuracy. These two aspects will be discussed in § 5.4.1 and § 5.4.2. Together they give a complete assessment of the quality of the roof segments. In this study, however, the focus is not only on producing good results, but doing this in an efficient manner. Therefore, the computation time of the method will be evaluated as well.

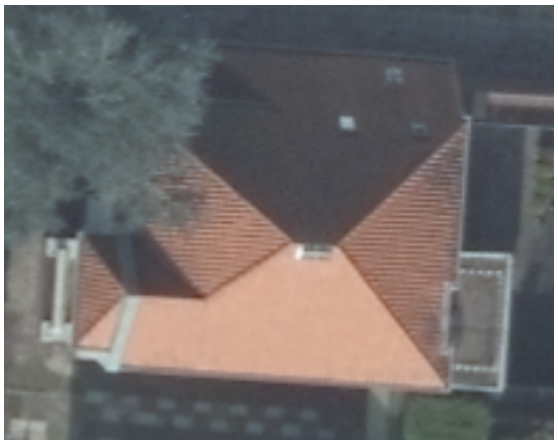
5.4.1 Segmentation quality

In object detection, the classification result is often evaluated based on the number of true positive (TP)s, true negative (TN)s, false positive (FP)s and false negative (FN)s. This principle can also be applied on the extracted roof segments. A TP is an extracted segment that has a corresponding segment in the reference data. A FP, on the other hand, is an extracted segment that does not have a corresponding segments in the reference data. A FN is a segment in the reference data for which no corresponding segment was found. TNs would be roof segments that are classified as background [Rutzinger et al., 2009]. In this case TNs do not exist since roof segments are not actually classified but extracted Rottensteiner et al. [2014]. The question remains when a detected segment and a segment in the reference data correspond with each other. [Zhan et al., 2005] discuss several ways to determine whether something is a TP. The most common way is to evaluate the overlap between the detected segment and the segment in the reference data. If the overlap exceeds a certain threshold, the segments correspond. For the ISPRS benchmark, an overlap criterion of 50% is taken Rottensteiner et al. [2014]. In this study the same criterion will be used. An example of how this works in practice is given in Figure 5.3. [Rutzinger et al., 2009] present several formulas to determine the completeness (Equation 5.1), correctness (Equation 5.2) and quality (Equation 5.3) of the segmentation. Where (|.|) denotes the number of instances in that class. The completeness is also known as the detection rate and gives the percentage of segments in the reference data that is detected. The correctness gives the percentage of extracted roof planes that have a corresponding segment in the reference data. Since a good segmentation both has a high completeness and correctness, the quality combines the two, expressing the quality in a single value [Rutzinger et al., 2009]. Similar as for the ISPRS benchmark, the $Comp$, $Corr$ and Q are also retrieved taking only segments larger than 10m^2 and 50m^2 into account. In this way the influence of roof segments size on the results can be evaluated.

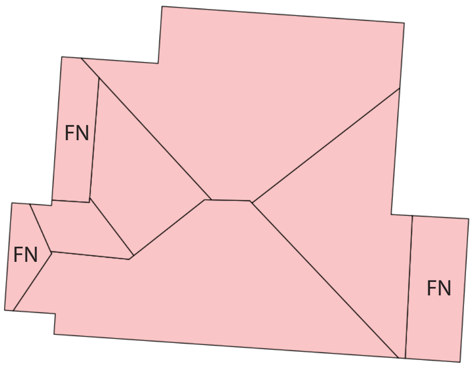
$$Comp = \frac{|TP|}{|TP| + |FN|} \tag{5.1}$$

$$Corr = \frac{|TP|}{|TP| + |FP|} \tag{5.2}$$

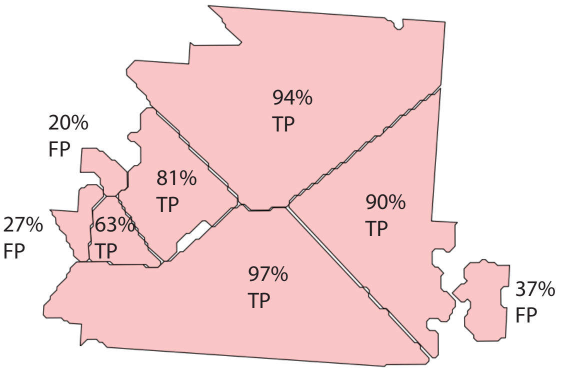
$$Q = \frac{|TP|}{|TP| + |FP| + |FN|} \tag{5.3}$$



(a) Aerial



(b) Reference data



(c) Detected segments

Figure 5.3: Example of how percentage of overlap determines whether a segment is a *TP* or *FP*, and segments that are not detected (*FN*)

5.4.2 Geometrical accuracy

For the geometrical accuracy, only the z component of the roof segments is taken into account. In order to measure the geometrical accuracy of the z component, a reference dataset with height values is required. Two of such datasets are available for the entire research area: (1) the AHN, and (2) the pointcloud resulting from stereo matching of the stereo10 images (§ 5.2.2). The height accuracy is calculated with respect to both of these datasets, since they give slightly different information. The key difference between the two is that the matched pointcloud was used to create the segments, whilst the AHN was not. A single value for the height accuracy of a segment is retrieved in two steps. First of all, pointcloud points falling within the segment are selected (based on x and y coordinates). Second, the root-mean-square deviation (RMSD) of the distances of all these points to the plane model of the segment is taken. Note that both pointclouds and the plane model are in world coordinates. The height accuracy value is denoted as $RMSD$ and is similar to the one defined in [Rottensteiner et al., 2014]. The $RMSD$ is retrieved both with respect to the matched pointcloud ($RMSD^{stereo}$) and LiDAR pointcloud ($RMSD^{LiDAR}$). The two metrics are defined in Equation 5.4 and 5.5 respectively. Since OLS is used for plane fitting, $RMSD^{stereo}$ is the value that is minimized during this process (see § 2.5.1). In the case that all points in the segments fall perfectly within a plane, the value of $RMSD^{stereo}$ will be 0. Thus, a high $RMSD^{stereo}$ indicates that the points do not fall in the same plane. This can be the case if the extracted segment represents a non planar roof segment or multiple planar roof segments. A high value can therefore be an indication of under-segmentation. In the case of $RMSD^{lidar}$, a high value can also occur when a segment is perfectly planar but has a wrong orientation, slope or height. The error compared to the AHN is thus a more general geometrical quality measure.

$$RMSD^{stereo} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i^{stereo} - z_i^{seg})^2} \tag{5.4}$$

$$RMSD^{LiDAR} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i^{LiDAR} - z_i^{seg})^2} \tag{5.5}$$

5.5 RESULTS

First, § 5.5.1 presents the results of the segmentation quality assessment for the three smaller test areas. Second, § 5.5.2 discusses the results of the geometric accuracy for the entire research area. In order to get more insight in causes of errors, § 5.5.3 will show common segmentation challenges and problems on the basis of several examples. Finally, § 5.5.4 evaluates the computation time of the developed prototype.

5.5.1 Segmentation results for Terraced, Free-standing and Industry test areas

The segmentation results of the three test areas will first be presented separately. Afterwards, a comparison is made between the three different areas.

TERRACED: In Figure 5.4 the extracted roof segments for the terraced test area are visualized. The simple gabled roof shapes are clearly visible and are reconstructed rather well. The segmentation results are quantified in Table 5.1. The *Comp* indicates that 92.9% percent of the roof segments is detected. The *Corr* shows that 86.4% of the extracted roof segments actually represent a true segment. Together this results in a overall segmentation quality of 81%. When only considering larger segments both the *Comp* and *Corr* increase.

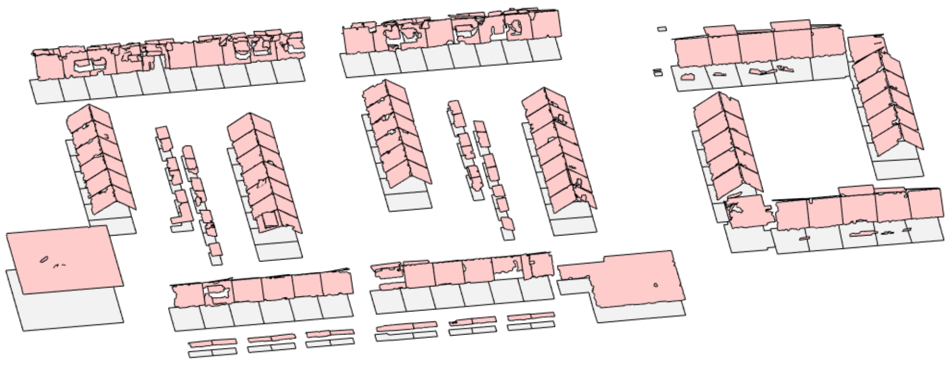


Figure 5.4: Extracted roof segment for the area with terraced buildings

FREE-STANDING: In Figure 5.5 the extracted roof segments for the free-standing test area are visualized. It is apparent from the segmentation that the roof shapes are complicated. Table 5.1 shows the results for the segmentation quality assessment. Considering segments of all sizes, the *Comp* and *Corr* are 64.5% and 76.2%. Only considering larger segments, the quality increases a lot. This indicates that the method especially has trouble detecting smaller segments for these complicated roofs. Additionally, small segments are more likely to be incorrect.

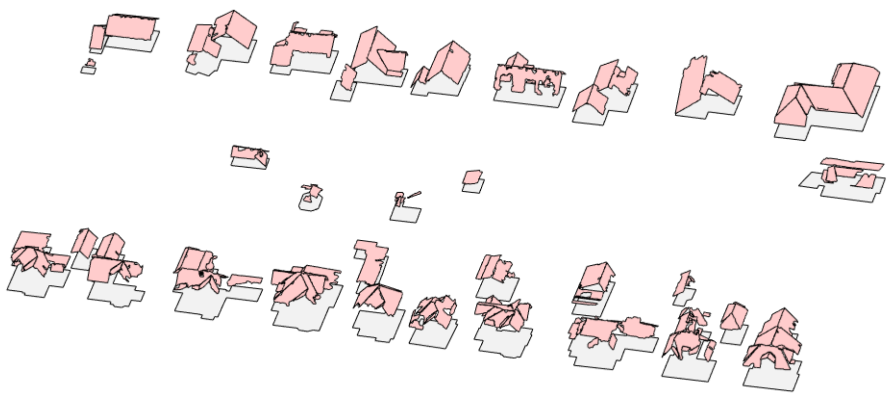


Figure 5.5: Extracted roof segment for the area with free-standing buildings

INDUSTRY: Finally, the extracted roof segments for the industry area are visualized in Figure 5.6. The large roof segments are clearly visible and seem to be reconstructed generally well. Again the results are quantified in Table 5.1. The one thing that especially stands out, is the high number of FPs resulting in a low *Corr*. This means that small segments are found that are not actually roof segments. This is not surprising, considering that long flat roof ridges, pipes and other objects are common in industrial areas. Such features are likely to be extracted as roof segment. Only looking at larger segments, this problem does not occur. All 43 segments larger than $50m^2$ had a corresponding segment in the reference data, and only 1 segment is not detected. The quality of the method is thus high for extracting large roof segments.

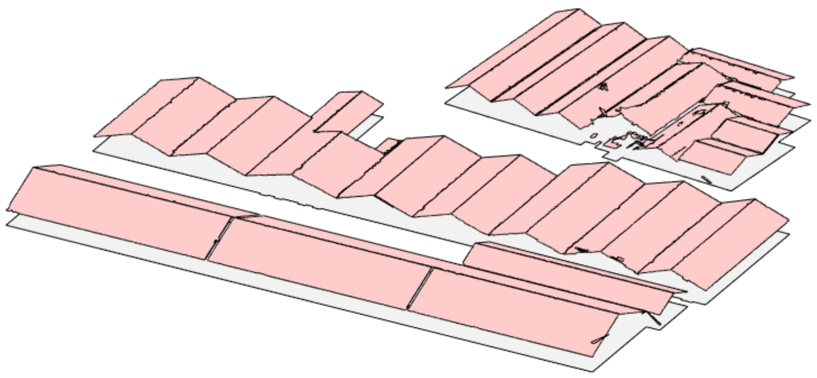


Figure 5.6: Extracted roof segment for the area with industry buildings

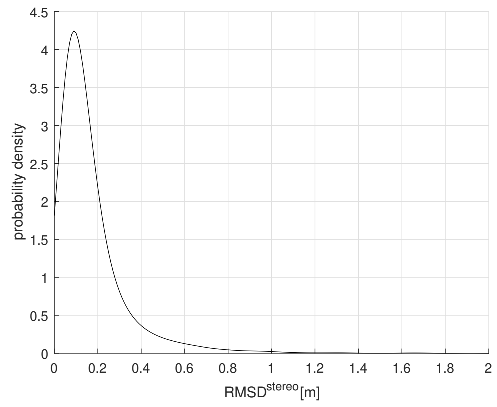
Stereo	Area (m^2)	*TP*	*FP*	*FN*	*Comp*	*Corr*	*Q*
Terraced	>1	222	35	17	92.9	86.4	81.0
	>10	206	3	7	96.7	98.6	95.4
	>50	4	0	0	100	100	100
Free-standing	>1	160	50	88	64.5	76.2	53.7
	>10	105	4	17	86.1	96.3	83.3
	>50	18	3	1	94.7	85.7	81.8
Industry	>1	45	47	6	88.2	48.9	45.9
	>10	45	5	2	95.7	90.0	86.5
	>50	43	0	1	97.7	100	97.7

Table 5.1: Quality assessment of the three test areas (terraced, free-standing and industry)

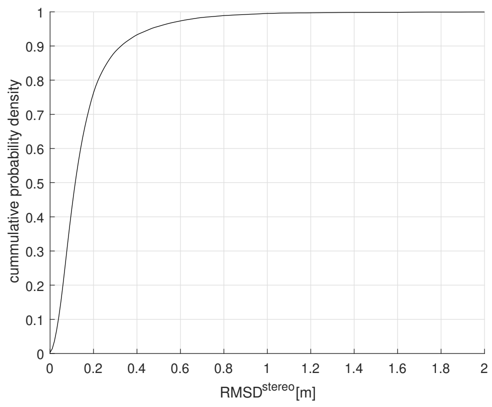
Comparing the results of the test areas, it can be seen that the quality is highest for the terraced houses. Especially for the free-standing houses, many segments are not detected. These are mostly small roof segments, that occur less in the terraced and industry test areas. For all three areas, it is clear that the size of the segments is positively correlated to the $Comp$ and $Corr$.

5.5.2 Geometrical accuracy

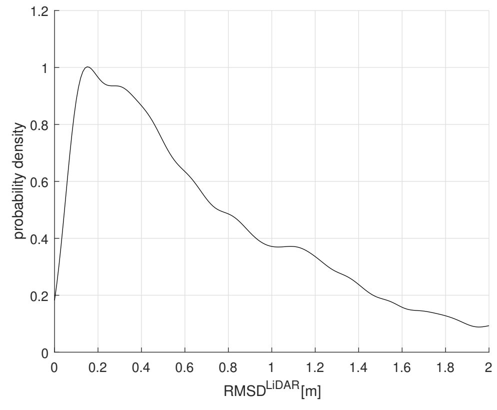
In Figure 5.7 the PDFs and cumulative probability density function (CDF)s for the geometric accuracy metrics are presented. The PDFs gives the probability that the metric in question has a certain value. So in case of $RMSD^{stereo}$ (Figure 5.7a) it can be observed that the most likely value is approximately 10cm. In the CDF the percentage of segments can be seen for which the metric is below a certain value. From the CDF in Figure 5.7b it can thus be seen that, the $RMSD^{stereo}$ is below 20cm for roughly 80% percent of the segments. These functions indicate that, the error compared to the matched pointcloud is generally very low. This is as expected since the matched pointcloud is used to produce the segments. The most likely value for $RMSD^{lidar}$ is roughly 15cm. However, a larger error is much more common in case the LiDAR is used as reference. For 30% of the segments, $RMSD^{lidar}$ is above 1 meter.



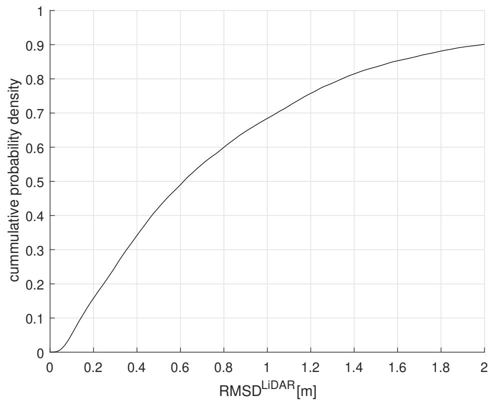
(a) pdf of $RMSD^{stereo}$



(b) cdf of $RMSD^{stereo}$



(c) pdf of $RMSD^{LiDAR}$



(d) cdf of $RMSD^{LiDAR}$

Figure 5.7: Probability density and cumulative probability density of the geometrical accuracy metrics discussed in § 5.4

5.5.3 Common segmentation problems

There are many possible reasons why the segmentation results are not perfect. Some of the most common problems are related to: height jumps, dormers, shadowing effects, overhanging roofs, roof lights and roof objects like chimneys. On the basis of several example buildings, these different things will be discussed.

1. In the ideal situation, the different roof segments are clearly distinguishable in the aerial image. This is the case when boundaries are clearly visible, segments have a rather even color and no roof objects or shades are present. Example 1 in Figure 5.8 gives such a basic situation. As can be observed, the segmentation for such a house is nearly perfect.

2. A common problem is to deal with height jumps. Example 2 in Figure 5.8 shows a large building with a flat roof at two different levels. In the reconstruction the two different roof segments are clearly separated, proving that the height jump is detected.

3. Example 3 is of a house with several dormers. Since the dormers are clearly distinguishable on the image, they are separated during watershed segmentation. They are not clustered with the roof segments they are located in, as their orientation is different. However, the dormer on the south side of the image is partly shaded. This leads to a segmentation error; the shaded part of the dormer is added to the roof segment. By carefully looking at the large roof segments on the lower side of the image, another shadowing problem can be seen. This roof segment should actually be divided. Due to the shadow, these boundaries are not clear and the entire side of the building is seen as one segment. Such under-segmentation errors are common in shaded areas.

4. Example 4 shows the impact of shadowing effects more clearly. The neighbouring building is casting a shadow over the building in question. A color segment is created for this shaded area, which actually comprises two different roof segments. The shaded area is thus under-segmented and is therefore not reconstructed correctly. A similar problem occurs on the other side, where the building itself is casting a shadow over another roof segment.

5. Another problem is to deal with overhanging roofs. These are roofs that extend beyond the building footprint. In the proposed method, the footprint is used to cut the roof segments. Example 5 gives a situation where significant parts of the roof segments are cut of, because the roof is overhanging in places. Similar problems occur when building footprints are not accurate. Furthermore, it should be noted how the rooflights are incorporated in the segments. This is as expected, since the rooflights share the same orientation as their corresponding roof segments. In the clustering step they are therefore grouped together.

6. Roof objects like chimneys are generally not added to the segments they are located in. This is because these features do not fall in the same plane as the roof segments. Example 6 gives a row of similar buildings with a chimney on each of them. The most common scenario is that the chimney forms a separate small segment. Since the prototype does not yet reconstruct holes, this is not always visible in the reconstruction. For the building on the right, the chimney is visible as a gap on the side. The shadow of a chimney or other object often gets its own color segment. Normally these color segments are grouped with the right roof segment in the clustering step. However, in some cases the orientation of the shaded segment can not be determined accurately. As a result, the segment remains separate and over-segmentation of the actual roof segment occurs. The shadow of the second chimney from the right gives such an example. This is a common error, not only for shaded areas, but for all small color segments.

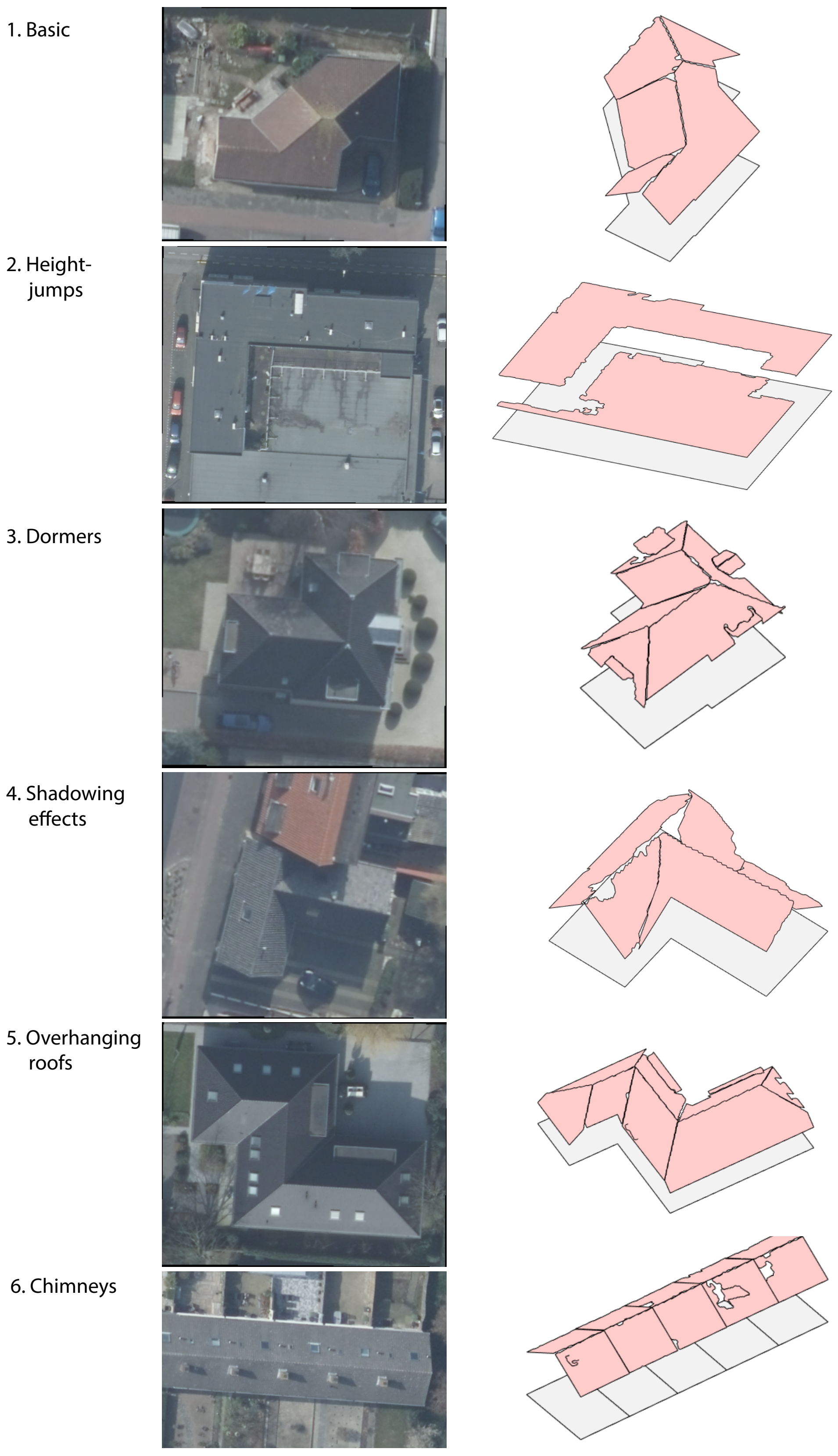


Figure 5.8: Segmentation examples

5.5.4 Computation Time

The designed method processes buildings separately. Computation times are therefore evaluated on on a per building basis. The different processing steps are timed separately in order to identify bottlenecks. In Table 5.2 the average computation times per building are listed. The loading times of the data are omitted, since they depends heavily on the database connection. The reported times are derived on a HP ZBook Studio G4 with an i7 2.8GHz quadcore from Intel. The average processing time for a single building is thus roughly a quarter of a second. This means that 14.400 buildings could be processed per hour. For reference, the average municipality in the Netherlands has approximately 25.000 buildings. The computation times of the different processes are all in the same order of magnitude, indicating that there are no obvious bottlenecks. In earlier versions of the developed prototype, when using RANSAC or LAD (see § 2.5.4 and § 2.5.2) for plane fitting, bottleneck were present. This was also the case when using mean shift and graph-based methods for image segmentation.

Process	Time (s)
Rectification	0.069
Matching	0.027
Watershed	0.018
BAG_Filter	0.033
Plane fitting	0.036
Height Jumps	0.004
Clustering	0.039
Reconstruction	0.031
Total	0.257

Table 5.2: Average computation times per building for the different processing steps

5.6 COMPARISON TO EXISTING METHODS

It is difficult to compare the designed method with other methods in a quantitative manner. The main reason for this is that other methods use different input data, both in terms of quality, and location of the test area. It is outside the scope of this research to overcome these issues and benchmark the designed method. An exception to this is READAARs LiDAR-based method, for which roof segments are available over the entire research area. Therefore, the segmentation quality of the designed method can be compared quantitatively to READAARs LiDAR based approach.

5.6.1 Comparison to READAARs LiDAR-based method

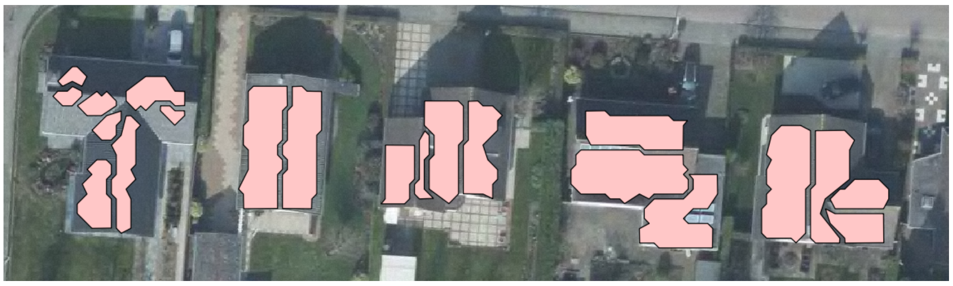
In Table 5.6 the segmentation quality of the developed stereo-based method is compared to READAARs LiDAR-based method. In terms of Completeness, the developed method outperforms the LiDAR-based method, for all three areas and for segments of all sizes. The differences in Correctness are less clear. One thing that stands out is that the developed method performs much better for small and medium sized segments in the free-standing houses area. Looking at the completeness, this is also the category where the difference between the two methods is most obvious. The LiDAR-based method especially has difficulties with the complicated roof shapes of free standing houses. Looking at the quality it becomes apparent that, the designed method especially improves the LiDAR-based method in this area. In the other areas a clear improvement is detected as well.

Stereo	Area (m^2)	*TP*	*FP*	*FN*	*Comp*	*Corr*	*Q*
Terraced	>1	222	35	17	92.9	86.4	81.0
	>10	206	3	7	96.7	98.6	95.4
	>50	4	0	0	100	100	100
Free-standing	>1	160	50	88	64.5	76.2	53.7
	>10	105	4	17	86.1	96.3	83.3
	>50	18	3	1	94.7	85.7	81.8
Industry	>1	45	47	6	88.2	48.9	45.9
	>10	45	5	2	95.7	90.0	86.5
	>50	43	0	1	97.7	100	97.7
LiDAR							
Terraced	>1	205	28	34	85.7	88.0	76.8
	>10	191	6	22	89.7	97.0	87.2
	>50	4	0	0	100	100	100
Free-standing	>1	85	86	163	34.3	49.7	25.5
	>10	71	18	51	58.2	79.8	50.7
	>50	13	1	6	68.4	92.9	65.0
Industry	>1	38	42	13	74.5	47.5	40.9
	>10	38	3	9	80.9	92.7	76.0
	>50	38	2	6	86.4	95.0	82.6

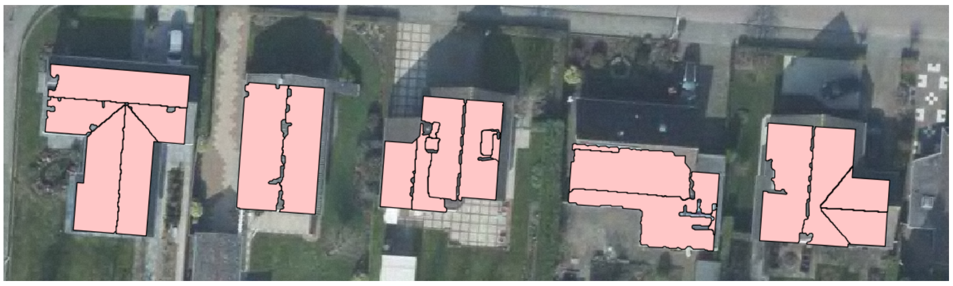
Table 5.3: Quality assessment of the three test areas (terraced, free-standing and industry)

In order to get insight in what causes these differences in quality, several examples are presented. Figure 5.9 shows the extracted roof segments for 5 buildings using the two different methods. In general, it can be observed that the designed stereo-based method extracts more complete segments. Looking carefully at the building in the middle, it can be seen that the dormers are extracted with the stereo-based method. Whilst, the LiDAR-based method fails to detect these small roof shapes. Another example of the amount of detail that is detected can be seen in the second building from the right. Note how the stereo-based method detects the entire flat roof segment. The LiDAR-based method does not detect the slim part of the segment on the left. However, the shaded roof segment of the same building is not detected at all by the stereo-based method. The LiDAR-based method does detect this segment, since LiDAR is invariant to lighting conditions. Similarly, there is a small segmentation error due to shading at the rightmost building. The diagonal boundary does not exactly meet the 90 degree corner, because it follows the boundary of the shadow. In the LiDAR segmentation this error is not present. It is thus important to realize that it can not be concluded that one method is superior to the other. Both methods have their strengths. The overall quality and level of detail is higher with the stereo-based method. However, the LiDAR-based method performs better in shaded areas.

(a) Aerial



(b) LiDAR (READAAR)



(c) Stereo (This thesis work)

Figure 5.9: Comparison of Stereo and LiDAR derived segments

6 CONCLUSIONS & FUTURE WORK

6.1 CONCLUSION

The conclusion is subdivided into three parts. First, some of the main issues related to the designed method are shortly discussed in § 6.1.1. Second, the research questions are answered in § 6.1.2. Finally, the main contributions are summarized in § 6.1.3.

6.1.1 Discussion

First of all, it can be concluded that the designed method is both scalable and efficient. This makes the method suitable for processing large areas such as municipalities and provinces. Furthermore, the method only relies on stereo imagery and building footprints. Thus, the method offers an alternative to READAARs LiDAR-based approach. Most importantly, this allows to process areas for which LiDAR data is not available, but stereo imagery is. The need for building footprints, however, is a potential problem. Footprint datasets might not always be of good quality, available or up to date. As stated in § 5.2.1, there are ways to retrieve such footprints from imagery. In case the quality of the footprint dataset is low, a buffer can be added to the footprint to ensure that the entire building is captured. Such a buffer could also be the solution for dealing with overhanging roofs. Even though the segmentation quality is improved compared to READAARs LiDAR-based approach, there is still room for improvement. Especially in shaded areas, segmentation problems are common. One of the main points of criticism on the designed method is that color segments are never split. If two different roof segments have the same color, and are not clearly separated, they could end up in the same color segment. Such segments could be split again using the height information from stereo matching. Promising ways to improvement the stereo matching and segmentation will be discussed in § 6.2.

6.1.2 Research Questions

- *Which stereo matching methods provide a good tradeoff between efficiency and quality?*
  Based on the conducted literature study, it can be concluded that SGM introduced by [Hirschmüller, 2005] is the most widely used method that is efficient, whilst still achieving a high accuracy. A possible alternative is LIBELAS presented by [Geiger et al., 2010]. Furthermore, neural networks could potentially be used for this purpose (see § 6.2.2).

- *Which methods can and cannot be used for the efficient extraction of roof segments from pointclouds?*
  The RANSAC, hough transform and region growing have all been used for the extraction of planar segments. RANSAC is especially computationally expensive when many outliers are present. The reason for this is that generally more iterations are required to get a reliable solution with many outliers. For hough transform, computation times increase exponentially with the number of data points used [Tarsha-Kurdi et al., 2007a]. Region growing is generally

slow, because it requires many iterations which can not be processed parallel. Typical efficient methods for large-scale applications are thresholding and certain clustering approaches. Furthermore, watershed segmentation has proven to be reasonable efficient.

- *How to ensure that the method is scalable to municipality level?* By integration of stereo matching and roof segmentation extraction it can be ensured that normal pre-processing steps, such as rectification and matching, are not required. In the integrated approach all processing steps are done separately for each building, ensuring a minimal memory footprint. Obviously, the designed method should be fully automatic as well.

- *Should radiometric (color) information be exploited?* The knowledge that similar color regions are more likely to have the same orientation, can and should be exploited in the building reconstruction process. In this study, this is done by using a two step approach. First, color segments are created. Second, these color segments are merged based on their orientation.

- *How to deal with data gaps and noise in the segmentation process?* Gaps and noise are to some extent inherent to stereo matching, especially when only two images are used for stereo reconstruction [Nex and Remondino, 2012]. Starting with a color segmentation partly deals with this problem. Not all point in the segment need to be matched to determine the orientation of the color segment. In this way the color segmentation allows gaps, to some degree. Especially for larger segments, noise is also largely cancelled out.

- *Which roof features can be ignored?* Roof features like chimneys, rooflights, antennas, and other roof objects are ignored. The extraction of planes for such features is not feasible. Furthermore, it is only partly relevant for this research considering the two main applications: PV potential analysis and asbestos detection. For PV potential, roof objects could be of indirect use to estimate shading effects. Roof features that are actually part of the roof structure, such as dormers, are extracted. These features are relevant, as PV panels could be placed on them.

- *What roof segment quality is achievable for an efficient stereo based method?* Especially simple terraced houses can be reconstructed rather reliable. A detection rate of 90% is achievable. The limitations of the method are especially clear for more complicated roof shapes. In addition, segmentation errors due to shading will always be presented with the proposed method.

6.1.3 Contributions

The main contributions of this research are:

- Integrating stereo matching with roof segment extraction. Therefore, preprocessing the stereo matching for the entire area is not needed. This ensures the method has a small memory footprint and makes it easily scalable. Furthermore, efficiency is improved since only buildings are matched.

- Designing a stereo-based workflow with efficient algorithms only. A literature study is conducted, in which several segmentation approaches are evaluated. Even though not all segmentation approaches are addressed, it gives an indication of which common segmentation approaches are promising for efficient extraction of roof segments.

- Dealing with gaps and noise by two step segmentation/clustering approach. Providing a approach for roof segments extraction, specifically designed for dealing with incomplete and noisy image-based pointclouds.

6.2 FUTURE WORK

Based on this study several recommendations can be made for future research. To enhance the stereo matching, multi-view matching (§ 6.2.1), and convolutional neural networks (§ 6.2.2) could be explored. To improve the segmentation part, the use of multiple images (§ 6.2.3), integration of LiDAR data (§ 6.2.4), processing of buildings blocks (§ 6.2.5), and intersection of roof segments (§ 6.2.6) could be investigated.

6.2.1 Multi-view matching

Multi-view matching is the use of multiple overlapping images for matching. These images must be gathered during the same flight, so that lighting conditions are roughly the same. As described in § 5.2.2, this dataset has 60% forward and 20% side overlap. So in most cases the building can only be found on 2 images. However, currently some datasets are gathered with more overlap [N. Haala et al., 2013]. Especially higher forward overlap is common, since it can be accomplished without any extra costs. Higher forward overlap is just a matter of taking images at a higher frequency. No extra flight lines are required, which is the case for higher side overlap. State of the art software packages for dense matching, such as SURE developed by [Rothermel et al., 2012], already exploit multiple images. If forward overlap is for example 80% and side overlap 60%, each building can be found in 10 images. This means that 45 different stereo pairs are present for a single building Haala [2011]. The challenge is thus, to exploit multiple overlapping images in an efficient manner. Multi-view matching could be incorporated in the designed workflow. Instead of only clipping the building from the two closest images, the building should be clipped from all images in which it is present. These images are rectified and matched with each other to retrieve a single pointcloud for the building. Finally, this pointcloud can be used to determine the orientation of color segments.

6.2.2 Stereo matching using neural networks

As already shortly stated in § 2.2.2, convolutional neural networks can be exploited for stereo matching. High quality results have already been reported by for example [Luo et al., 2016] and [Zbontar and LeCun, 2016]. Furthermore neural networks are very efficient once trained. The problem is however that currently no training datasets exist with aerial imagery. Well known training datasets that do exist are the KITTI and Middlebury datesets, consisting of traffic situations and random objects respectively. In order to get an impression of the possibilities, a stereo pair from this study was fed to the pre-trained network of Luo et al. [2016]. In Figure 6.1 the resulting disparity is compared to the output of the SGM algorithm. It can be observed that, the SGM algorithm clearly gives a better result. However, the output of the neural network is still very promising, considering that the network is only trained with traffic situations from the KITTI dataset. In order to investigate the full potential of neural networks for aerial stereo matching, a training dataset needs to be created. Training data could be created by using LiDAR data such as the AHN. From LIDAR data a disparity map can be constructed corresponding to an aerial stereo pair. Problematic is, however, that the AHN has a lower resolution than the imagery. Therefore, the use of very high density LiDAR data for the construction of a training set would be desirable. Possibly municipal datasets such as the one of Rotterdam can be exploited.

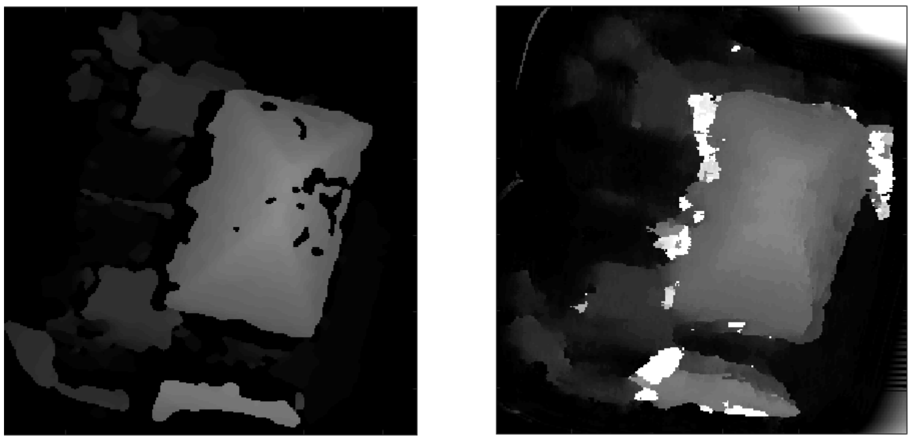


Figure 6.1: Comparison of disparity maps resulting from semi-global matching and pre-trained convulation neural network by Luo et al. [2016]

6.2.3 Segmentation using multiple images

Currently only one image is exploited for color segmentation. The segmentation can be improved by exploiting the available images from different viewpoints. Considering a black street and a black roof, the wall separating the two segments might be visible from one viewpoint, but not from the other. Even more interesting is the use of images from different times. One of the current problems is dealing with shaded areas. An area shaded in one image might be easy to segment in an image from a different time with other lighting conditions. The challenge with using multiple images is how to combine the different segmentation results.

6.2.4 Integration of LIDAR data

One of the goals of this study was to develop a stereo-based approach that is independent of LiDAR data. The reason for this is that LiDAR data is not always available. However, in case LiDAR data is available it might as well be exploited. Especially, considering that this stereo-based approach has problems in low contrast areas. Since LiDAR is an active method, it is independent of light conditions, and therefore does not have this problem. Thus, LiDAR data could complement the stereo data in shaded areas. LiDAR data could be integrated in the designed workflow relatively easily. Instead of using only the points from the matching to fit planes through the color and roof segments, the LiDAR points could be incorporated. Note that plane fitting is currently done in disparity space, so LiDAR point would need to be converted to disparity space. Another option would be to convert the matched points to world space and do the plane fitting in world space.

6.2.5 Building blocks

Often entire blocks of houses have exactly the same or a similar roof shapes (see Figure 6.2). This knowledge could be exploited by processing building blocks instead of individual buildings. In a building block with similar houses, roof segments generally span multiple buildings. Normally the slope, orientation and position of such segments is determined for each building individually. Therefore, variations will be present between these segments that are not there in reality (see Figure 6.3a). If entire building blocks are processed at once, a single position, slope and orientation can be determined for the segments spanning multiple houses. The fitted plane will also be more robust since more points are available. The result of processing an entire building block at once is given in Figure 6.3b. Note that by processing the entire block, even the shaded segments can be reconstructed accurately. Whilst in the case of processing individual buildings, the shaded segments can not be reconstructed reliably. In order to retrieve building blocks, the boundaries between buildings are dissolved. The dissolved boundaries can later be used for splitting up building segments. In this way segments for individual building can be retrieved. The main problem is, however, that many blocks do not have similar roof shapes. In such cases it is undesirable to process entire blocks. The reason for this is that, if there are many segments with varying orientations, clusters become hard to distinguish. So the challenge remains to determine automatically whether the houses in a block should be processed together or separately.

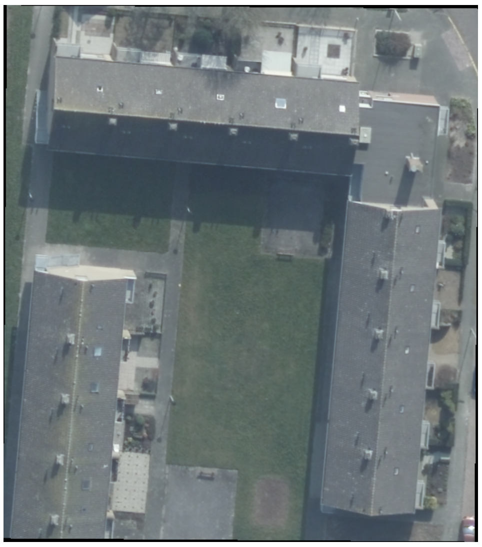
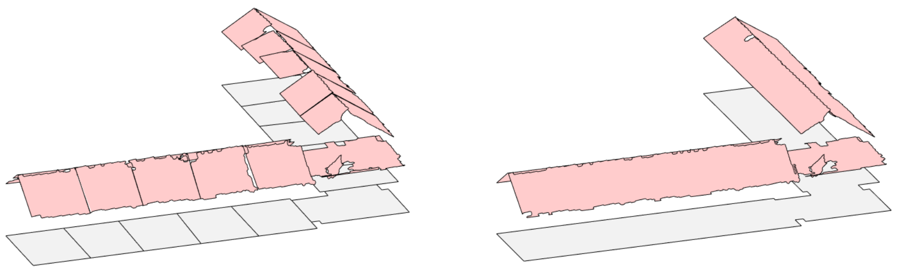


Figure 6.2: Aerial imagery of building block



(a) Individual buildings **(b)** Building block

Figure 6.3: Potential advantage of processing building blocks

6.2.6 Intersecting roof segments

The boundaries of the roof segments are currently quite rough. The reason for this is that boundaries are simply retrieved by taking the outermost ring of pixels of each segment. These boundaries could be improved, for example by intersecting neighboring segments (Figure 6.4). The intersection line between two segments can be used as new boundary for the two segments in question.

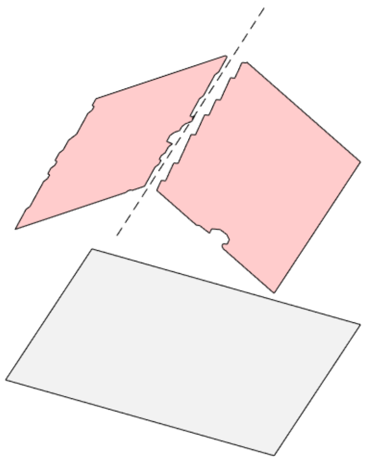


Figure 6.4: Finding boundaries (dashed) by intersecting roof segments (red)

6.3 REFLECTION

This section aims to relate this study to the main aspects of the TU Delft Geomatics program. Geomatics is the science concerning (1) data acquisition, (2) data storage/-management, (3) data analysis, (4) data visualization, and (5) data quality. These five aspects form the basic methodical line of approach for the Geomatics program at the TU Delft. All these steps can be identified is this graduation project. In the case of this research, data acquisition was done by third parties. This is the case both for the Stereo10 and BAG datasets (see § 5.2). Data storage and management is mostly done in READAARs PostGIS database (see § 5.1.1). The most substantial part of this thesis concerns data analysis. The analysis is the entire process of extracting roof segments from the two main data sources (see chapter 4). The visualization of data is present at many different steps throughout the process. Starting with visualizing of raw images and pointclouds and ending with the visualization of 3D roof segments. The quality of the data is assessed at various points throughout the process. First, the quality of input data is discussed in (§ 5.2). Matching quality is briefly discussed in chapter 3. Finally the segmentation and geometrical quality is extensively assessed in § 5.5.

BIBLIOGRAPHY

Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Awrangjeb, M., Zhang, C., and Fraser, C. S. (2013). Automatic extraction of building roofs using lidar data and multispectral imagery. *ISPRS journal of photogrammetry and remote sensing*, 83:1–18.

Bloomfield, P. and Steiger, W. (2012). *Theory, applications and algorithms*, volume 6. Springer Science & Business Media.

Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc.

Brenner, C. (2005). Building reconstruction from images and laser scanning. *International Journal of Applied Earth Observation and Geoinformation*, 6(3):187–198.

Burger, W. and Burge, M. J. (2016). *Digital image processing: an algorithmic introduction using Java*. Springer.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799.

Cho, S. H., Duy, H. T., Han, J. W., and Hwang, H. (2013). Multi-level thresholding based on non-parametric approaches for fast segmentation. *Journal of Biosystems Engineering*, 38(2):149–162.

Cignoni, P., Montani, C., and Scopigno, R. (1998). A comparison of mesh simplification algorithms. *Computers & Graphics*, 22(1):37–54.

Collins, R. (2007). Cse486/ee: Computer vision 1. Pennsylvania State University.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619.

Commandeur, T. (2012). Footprint decomposition combined with point cloud segmentation for producing valid 3d models. Master's thesis, TU Delft.

Dey, V., Zhang, Y., and Zhong, M. (2010). A review on image segmentation techniques with remote sensing perspective. *ISPRS TC VII Symposium – 100 Years ISPRS, Vienna, Austria*, 38(7A):31–42.

Dikbas, S., Arici, T., and Altunbasak, Y. (2007). Chrominance edge preserving grayscale transformation with approximate first principal component for color edge detection. *IEEE International Conference on Image Processing*, 2:261–264.

Dorninger, P. and Pfeifer, N. (2008). A comprehensive automated 3d approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. *Sensors*, 8(11):7323–7343.

Egels, Y. and Kasser, M. (2003). *Digital photogrammetry*. CRC Press.

Fan, H., Yao, W., and Fu, Q. (2014). Segmentation of sloped roofs from airborne lidar point clouds using ridge-based hierarchical decomposition. *Remote Sensing*, 6(4):3284–3301.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181.

Fusiello, A., Trucco, E., and Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22.

Geiger, A., Roser, M., and Urtasun, R. (2010). Efficient large-scale stereo matching. *Asian conference on computer vision*, pages 25–38.

Haala, N. (2011). Multiray photogrammetry and dense image matching. *Photogrammetric Week*.

Haala, N., Brenner, C., and Anders, K. H. (1998). 3d urban gis from laser altimeter and 2d map data. *International Archives of Photogrammetry and Remote Sensing*, 32:339–346.

Haala, N. and Kada, M. (2010). An update on automatic 3d building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):570–580.

Hirschmüller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hofierka, J. and Kaňuk, J. (2009). Assessment of photovoltaic potential in urban areas using open-source solar radiation tools. *Renewable Energy*, 34(10):2206–2214.

Huang, D. Y. and Wang, C. H. (2009). Optimal multi-level thresholding using a two-stage otsu optimization approach. *Pattern Recognition Letters*, 30(3):275–284.

Huang, H., Brenner, C., and Sester, M. (2011). 3d building roof reconstruction from point clouds via generative models. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 16–24.

Huang, H., Brenner, C., and Sester, M. (2013). A generative statistical approach to automatic 3d building roof reconstruction from laser scanning data. *ISPRS Journal of photogrammetry and remote sensing*, 79:29–43.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.

Kada, M. and McKinley, L. (2009). 3d building reconstruction from lidar based on a cell decomposition approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(3):47–52.

Kaur, D. and Kaur, Y. (2014). Various image segmentation techniques: A review. *International Journal of Computer Science and Mobile Computing*, 3(5):809–814.

Khoshelham, K. (2005). Region refinement and parametric reconstruction of building roofs by integration of image and height data. *Proceedings of the ISPRS Workshop CMRT 2005*, 36:3–8.

Kolbe, T. H. (2009). Representing and exchanging 3d city models with citygml. *3D geo-information sciences*, pages 15–31.

Kraus, K. (2007). *Photogrammetry: geometry from images and laser scans*. Walter de Gruyter.

Lafarge, F., Descombes, X., Zerubia, J., and Pierrot-Deseilligny, M. (2010). Structural approach for building reconstruction from a single dsm. *IEEE Transactions on pattern analysis and machine intelligence*, 32(1):135–147.

Liu, Y. and Xiong, Y. (2008). Automatic segmentation of unorganized noisy point clouds based on the gaussian map. *Computer-Aided Design*, 40(5):576–594.

Luo, W., Schwing, A. G., and Urtasun, R. (2016). Efficient deep learning for stereo matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703.

Meer, P., Rosenfeld, D. M. A., and Kim, D. Y. (1991). Robust regression methods for computer vision: A review. *International journal of computer vision*, 6(1):59–70.

Moons, T., van Gool, L., and Vergauwen, M. (2010). 3d reconstruction from multiple images part 1: principles. foundations and trends. *Computer Graphics and Vision*, 4(4):287–404.

N. Haala, N., Cramer, M., and Rothermel, M. (2013). Quality of 3d point clouds from highly overlapping uav imagery. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W2:183–188.

Nex, F. and Remondino, F. (2012). Automatic roof outlines reconstruction from photogrammetric dsm. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(3):257–262.

Nourian, P. (2016). Point cloud processing: Estimating normal vectors and curvature indicators using eigenvectors. Engineering.

Novacheva, A. (2008). Building roof reconstruction from lidar data and aerial images through plane extraction and colour edge detection. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37:53–57.

Novel, C., Keriven, R., Poux, F., and Graindorge, P. (2015). Comparing aerial photogrammetry and 3d laser scanning methods for creating 3d models of complex objects. *Capturing Reality Forum. Bentley Systems, Salzburg*, page 15.

Raju, P. D. R. and Neelima, G. (2012). Image segmentation by using histogram thresholding. *International Journal of Computer Science Engineering and Technology*, 2(1):776–779.

Roerdink, J. B. and Meijster, A. (2000). The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae*, 41(1,2):187–228.

Rothermel, M., Wenzel, K., Fritsch, D., and Haala, N. (2012). Sure: Photogrammetric surface reconstruction from imagery. *Proceedings LC3D Workshop, Berlin*, 8.

Rottensteiner, F. (2010). Roof plane segmentation by combining multiple images and point clouds. *Proceedings of the Photogrammetric Computer Vision and Image Analysis Conference*, 38:245–250.

Rottensteiner, F. and Briese, C. (2003). Automatic generation of building models from lidar data and the integration of aerial images. *ISPRS*, 19.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J. D., Breitkopf, U., and Jung, J. (2014). Results of the isprs benchmark on urban object detection and 3d building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:256–271.

Ruparelia, S. (2012). Implementation of watershed based image segmentation algorithm. Master's thesis, University of stuttgart.

Rutzinger, M., Rottensteiner, F., and Pfeifer, N. (2009). A comparison of evaluation techniques for building extraction from airborne laser scanning. *Applied Earth Observations and Remote Sensing*, 2(1):11–20.

Sampath, A. and Shan, J. (2006). Clustering based planar roof extraction from lidar data. *American Society for Photogrammetry and Remote Sensing Annual Conference, Reno, Nevada*, pages 1–6.

Sampath, A. and Shan, J. (2010). Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. *IEEE Transactions on geoscience and remote sensing*, 48(3):1554–1567.

Scharstein, D. and Pal, C. (2007). Learning conditional random fields for stereo. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, MN*.

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

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.

Sinha, S. N., Scharstein, D., and Szeliski, R. (2014). Efficient high-resolution stereo matching using local plane sweeps. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1582–1589.

Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.

SZW, I. (2015). Sectorrapportage abest 2015. Technical report, Ministerie van Sociale Zaken en Werkgelegenheid.

Taillandier, F. (2005). Automatic building reconstruction from cadastral maps and aerial images. *International Archives of Photogrammetry and Remote Sensing*, 36(3):105–110.

Tarsha-Kurdi, F., Landes, T., and Grussenmeyer, P. (2007a). Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. *Proceedings of the ISPRS Workshop on Laser Scanning*, 36:407–412.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., and Koehl, M. (2007b). Model-driven and data-driven approaches using lidar data: Analysis and comparison. *ISPRS Workshop, Photogrammetric Image Analysis (PIA07)*, pages 87–92.

Trucco, E. and Verri, A. (1998). *Introductory techniques for 3-D computer vision*, volume 201. Prentice Hall.

Vakalopoulou, M., Karantzalos, K., Komodakis, N., and Paragios, N. (2015). Building detection in very high resolution multispectral data with deep learning features. *Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1873–1876.

van der Zon, N. (2013). Kwaliteitsdocument ahn2. Delft, The Netherlands.

Vosselman, G. and Dijkman, S. (2001). 3d building model reconstruction from point clouds and ground plans. *International archives of photogrammetry remote sensing and spatial information sciences*, 34:37–44.

Wahl, R., Schnabel, R., and Klein, R. (2008). From detailed digital surface models to city models using constrained simplification. *Photogrammetrie, Fernerkundung, Geoinformation*.

Wang, Y. H. (2010). Tutorial: Image segmentation. *National Taiwan University, Taipei*, pages 1–36.

Wolff, K., Kim, C., Zimmer, H., Schroers, C., Botsch, M., Sorkine-Hornung, O., and Sorkine-Hornung, A. (2016). Point cloud noise and outlier removal for image-based 3d reconstruction. *3D Vision (3DV)*, pages 118–127.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678.

Zbontar, J. and LeCun, Y. (2016). Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(2):1–32.

Zenzo, S. D. (1986). A note on the gradient of a multi-image. *Computer vision, graphics, and image processing*, 33(1):116–125.

Zhan, Q., Molenaar, M., Tempfli, K., and Shi, W. (2005). Quality assessment for geo-spatial objects derived from remotely sensed data. *International Journal of Remote Sensing*, 26(14):2953–2974.

A | NUMANSDORP MAP

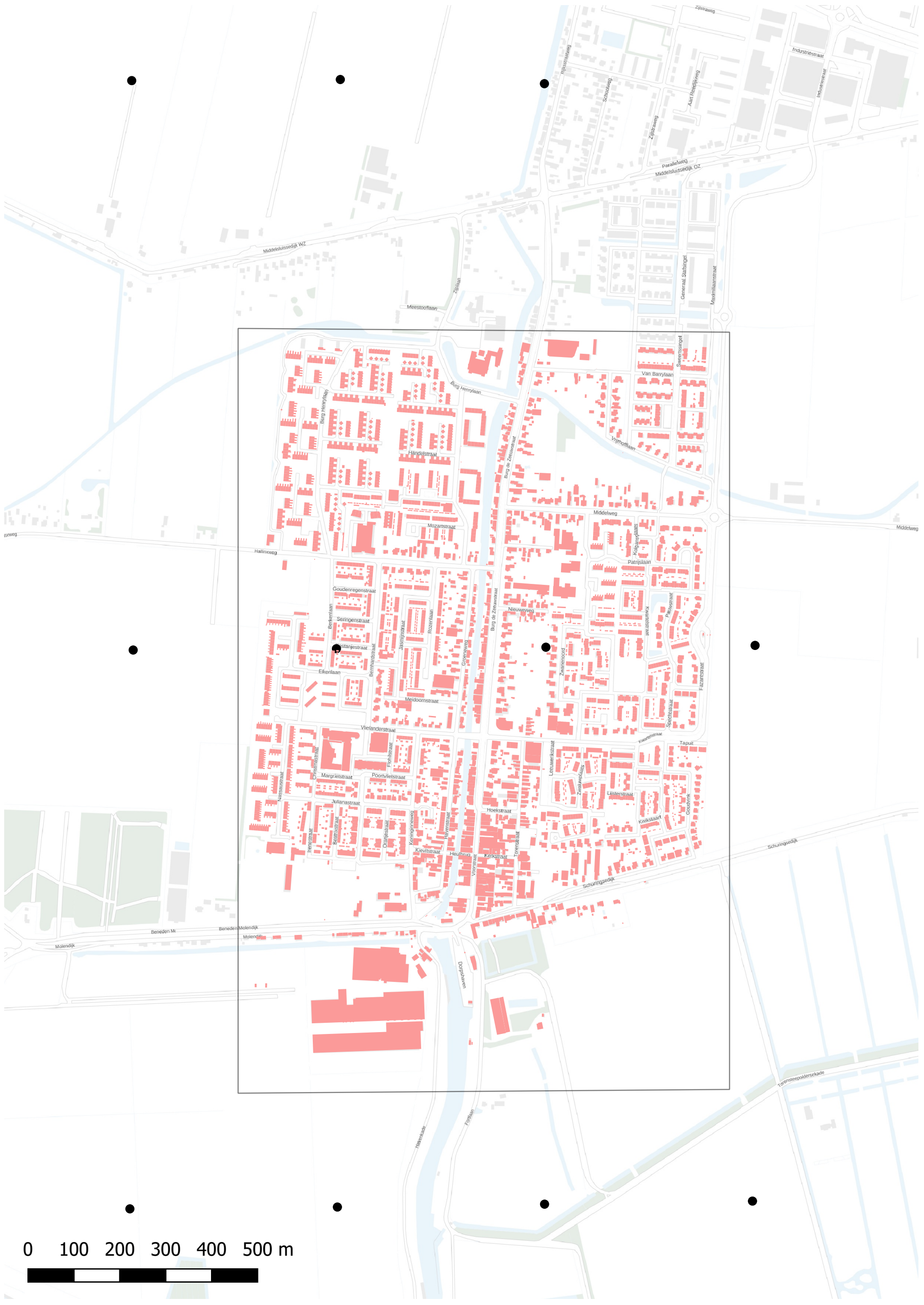
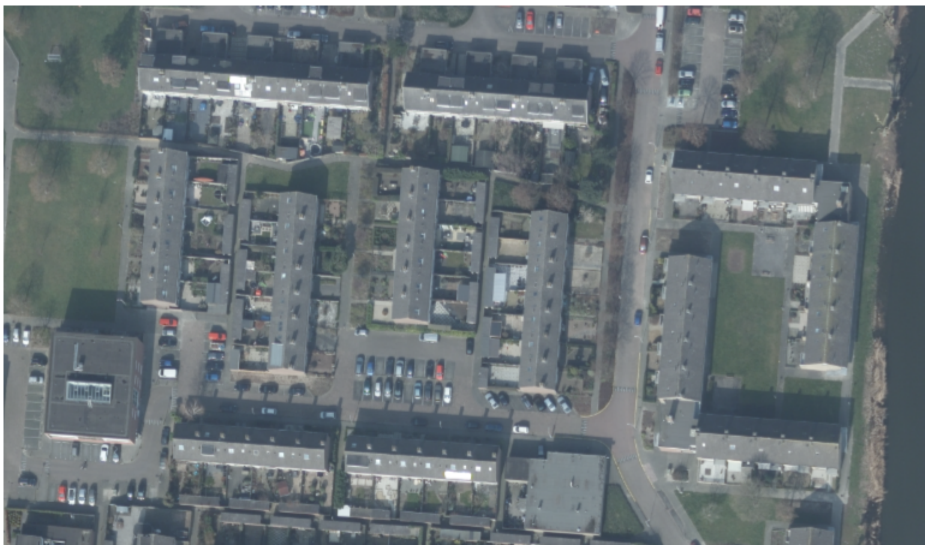


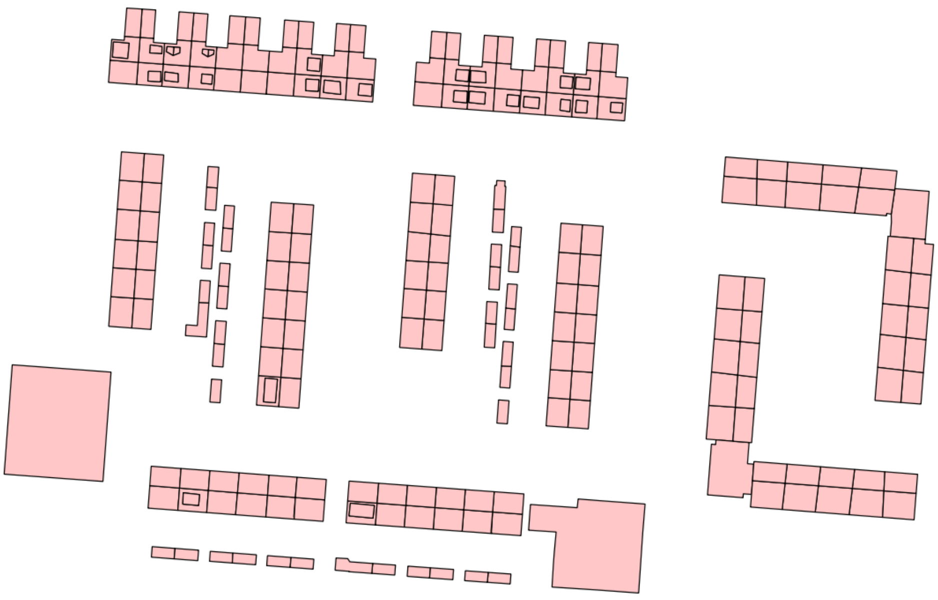
Figure A.1: Map of the research area (rectangle) in Numansdorp with houses being processed (light red) and location of images (black dots)

B REFERENCE DATA

Here the reference data for the three small test areas within the larger Numansdorp research area are presented. For each of the areas (terraced, free-standing and industry) an aerial image of the Stereo10 dataset(§ 5.2.2) is shown and the reference roof segments. The reference segments were retrieved manually using the imagery shown in this appendix. The BAG footprints (§ 5.2.1) were used as a starting point for the mapping.

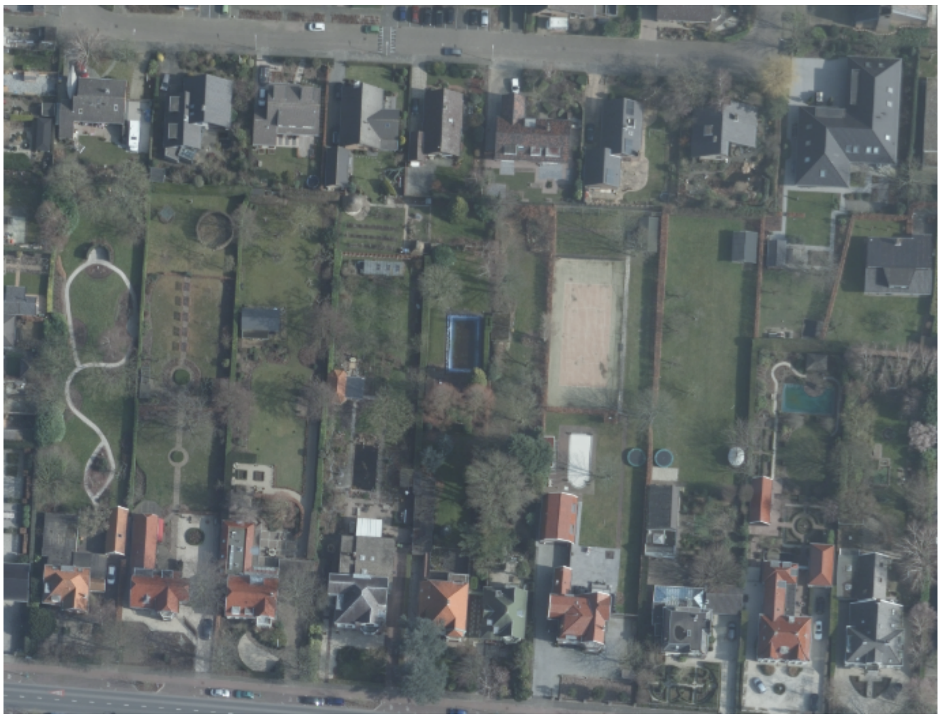


(a) Aerial

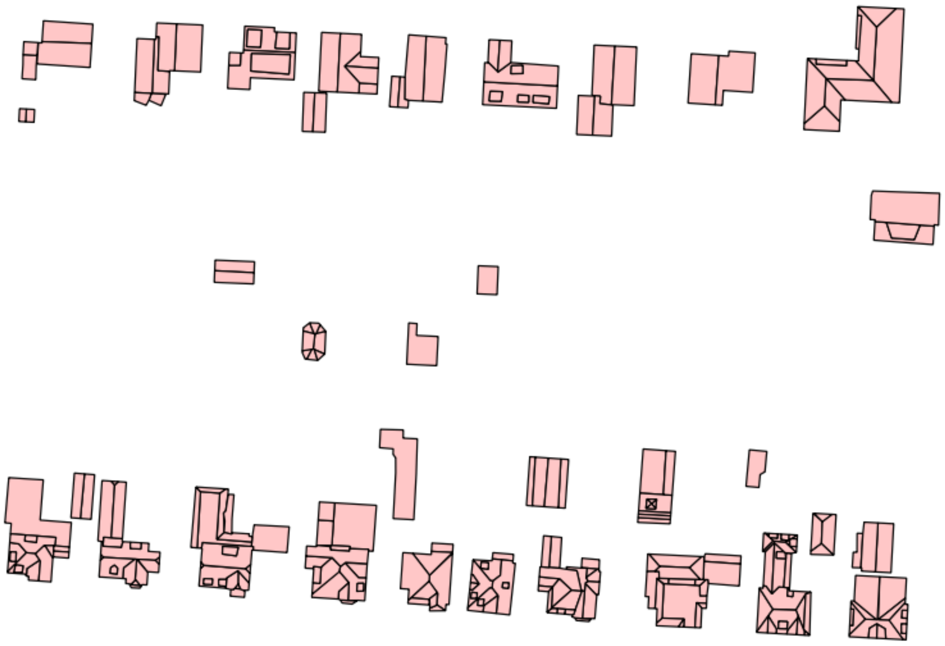


(b) Reference roof Segments

Figure B.1: Reference data for the area with terraced houses

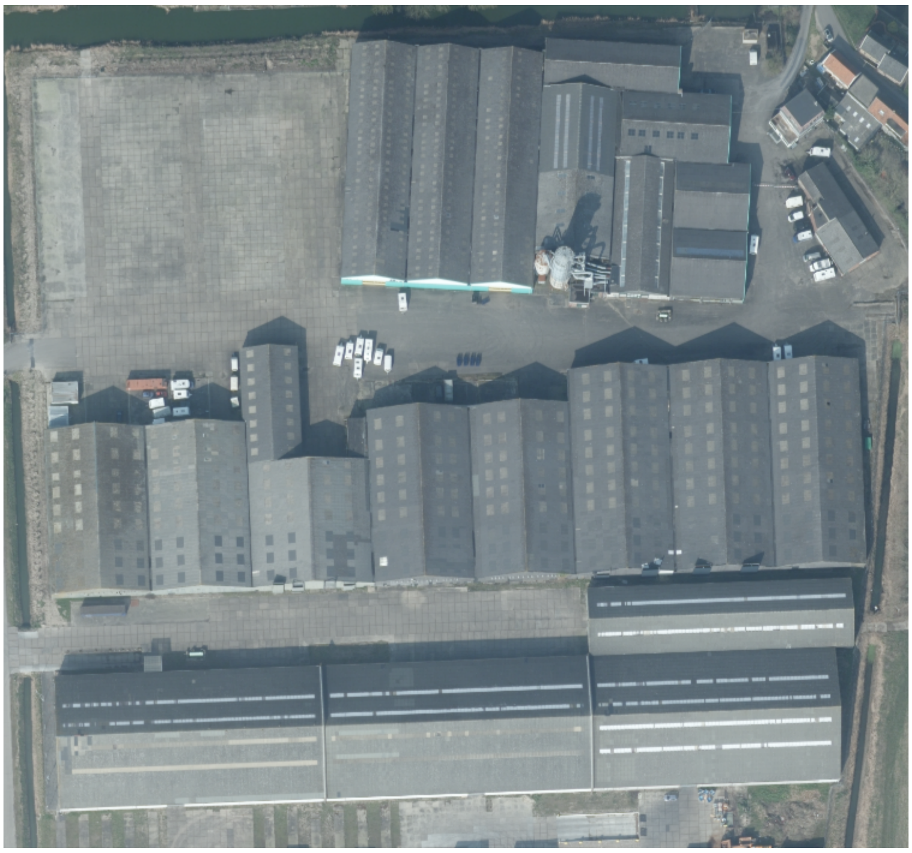


(a) Aerial



(b) Reference roof Segments

Figure B.2: Reference data for the area with free-standing houses


(a) Aerial

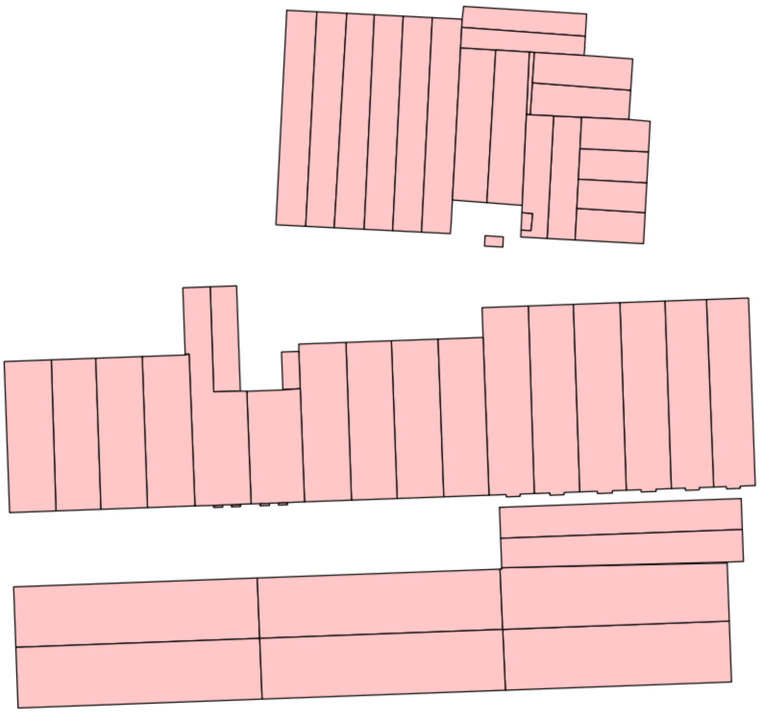

(b) Reference roof Segments

Figure B.3: Reference data for the area with industry

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.

TU Delft