TU Delft

Delft University of Technology

# Combining LiDAR and Photogrammetry to Generate Up-to-date 3D City Models

Zhou, Kaixuan

**DOI**
10.4233/uuid:89466e36-b579-4943-b3da-b251dd52209f

**Publication date**
2020

**Document Version**
Final published version

**Citation (APA)**
Zhou, K. (2020). *Combining LiDAR and Photogrammetry to Generate Up-to-date 3D City Models*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:89466e36-b579-4943-b3da-b251dd52209f

**Important note**
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

**Copyright**
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

*This work is downloaded from Delft University of Technology.*
*For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.*

# Combining LiDAR and Photogrammetry to Generate Up-to-date 3D City Models

# Combining LiDAR and Photogrammetry to Generate Up-to-date 3D City Models

## Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 9 juni om 10:00 uur

door

**Kaixuan ZHOU**

Master of Science in Geomatics,
Delft University of Technology, the Netherlands

geboren te Dongtai, China

This dissertation has been approved by the promotors:

Prof. Dr. Ir. R.F. Hanssen
Dr. R.C. Lindenbergh

Composition of the doctoral committee:

| | |
|---|---|
| Rector Magnificus, | Chairman |
| Prof. Dr. Ir. R.F. Hanssen, | Delft University of Technology, promotor |
| Dr. R.C. Lindenbergh, | Delft University of Technology, promotor |
| *Independent members:* | |
| Prof. Dr. D.M. Gavrila | Delft University of Technology |
| apl. Prof. Dr. -Ing. N. Haala | University of Stuttgart, Germany |
| Dr. M. Pierrot-Deseilligny | Université Gustave Eiffel, France & LASTIG |
| Dr. Ir. B.G.H. Gorte | University of New South Wales, Australia |
| Dr. Ir. S.J. Oude Elberink | University of Twente |
| Prof. Dr. Ir. H.W.J. Russchenberg | Delft University of Technology, reserve member |

*Keywords:* 3D city model, large scale mapping, change detection and updating, building extraction, airborne laser scanning, airborne camera imagery, shadow, machine learning, dense matching

ISBN 978-94-6366-278-9
Copyright © 2020 by K. Zhou
All rights reserved. No part of the material protected by this copyright notice may be re-produced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval systems, without the prior permission of the author.

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

# Contents

**Summary** **9**

**Samenvatting** **11**

**1 Introduction** **13**
1.1 Motivation . . . 13
1.2 Background . . . 13
1.2.1 Airborne laser scanning . . . 14
1.2.2 Airborne camera imagery . . . 15
1.3 Problem statement . . . 16
1.3.1 3D change detection and updating. . . . 17
1.3.2 Building extraction from LiDAR data . . . 20
1.4 Research objectives . . . 22
1.5 Scope and limitations. . . . 23
1.6 Organization of the thesis . . . 23

**2 3D geometry in LiDAR and photogrammetry** **25**
2.1 Airborne laser scanning . . . 25
2.1.1 Airborne laser scanning system . . . 25
2.1.2 3D geometry from ALS point clouds . . . 26
2.2 Airborne camera imagery . . . 27
2.2.1 Airborne camera imagery system . . . 27
2.2.2 3D geometry from ACIM stereo images . . . 28
2.3 Photogrammetric fundamentals for combining ALS and ACIM . . . 29
2.3.1 Perspective projection . . . 30
2.3.2 Ray tracing . . . 35
2.3.3 Stereo rectification . . . 35
2.4 Conclusions . . . 40

**3 Shadow detection in a single image** **41**
3.1 Introduction. . . . 41
3.2 Study area and data preparation . . . 44
3.2.1 3D model generation . . . 44
3.2.2 VHR image description. . . . 46
3.3 Shadow detection . . . 46
3.3.1 Shadow reconstruction using an existing 3D Model. . . . 46
3.3.2 Adaptive erosion filtering. . . . 49
3.3.3 Classification using automatically selected samples . . . 51
3.3.4 Evaluation of classification . . . 53

3.4 Experiments and comparisons . . . 55
3.4.1 Feasibility of shadow reconstruction for training samples . . 55
3.4.2 Method comparison. . . . 57
3.5 Conclusions . . . 64

**4 3D building change detection and updating using a stereo pair 65**
4.1 Introduction. . . . 66
4.2 Related work . . . 67
4.2.1 Single image change detection . . . 67
4.2.2 Direct geometric change detection . . . 68
4.2.3 Projection-based geometric change detection . . . 68
4.3 Methodology . . . 69
4.3.1 Partial change detection—LiDAR-guided dense matching . . 69
4.3.2 Change propagation and update . . . 77
4.3.3 Post-processing . . . 79
4.3.4 Evaluation. . . . 80
4.4 Experiment and discussion . . . 80
4.4.1 Data specification and pre-processing . . . 80
4.4.2 Results . . . 82
4.4.3 Discussion . . . 87
4.5 Conclusions . . . 90

**5 Improving building extraction using multi-view images 91**
5.1 Introduction. . . . 91
5.2 Study area and materials. . . . 94
5.3 Methodology . . . 95
5.3.1 LiDAR-guided edge-aware dense matching . . . 95
5.3.2 Generation of iDSM and true ortho-image . . . 97
5.3.3 Integration with multi-view images . . . 98
5.3.4 Building extraction . . . 101
5.3.5 Evaluation. . . . 102
5.4 Experiment and results. . . . 102
5.4.1 Vaihingen results . . . 102
5.4.2 Amersfoort results . . . 106
5.5 Discussion . . . 108
5.5.1 Building extraction . . . 108
5.5.2 Parameter Setting . . . 109
5.6 Conclusions . . . 110

**6 Conclusions and recommendations 111**
6.1 Conclusions . . . 111
6.1.1 Shadow detection from a single image . . . 112
6.1.2 3D building detection and updating using a stereo pair . . . 114
6.1.3 Improving building extraction using multi-view images . . . 116
6.2 Recommendations for future work . . . 117

**Bibliography** **121**

**Curriculum Vitæ** **131**

**List of Publications** **133**

# Summary

3D city models are increasingly used to maintain and improve urban infrastructure. Keeping 3D city models accurate and up-to-date is essential for municipalities to make decisions in a time of strongly increasing urbanization. 3D information provided by airborne laser scanning (ALS) is widely used for generating 3D city models. However, ALS data is sparse and irregularly spaced, and not frequently acquired due to its high costs. Airborne camera imagery (ACIM) is an alternative to extract denser but less accurate 3D information. Given these limitations in acquisition frequency and quality, using either ALS or ACIM to generate up-to-date large-scale 3D city models is sub-optimal.

Therefore, we combine the complementary characteristics of both data sources to achieve two objectives: (i) 3D change detection and updating of buildings in ALS data using ACIM data, and (ii) improving the planimetric accuracy of building extraction from ALS data using ACIM data. ALS data is integrated with a single image or a single stereo pair for the first objective, and with multiple stereo pairs for the second objective. Our methods are validated over three areas: Vaihingen, Germany, and Amersfoort and Assen, the Netherlands.

Shadow in a single image is indicative for a 3D object and is represented in the image by RGB color values. However, these color values are not unique, as they depend on the local conditions, such as material and environment. We propose a supervised machine learning approach, random forest, to effectively characterize the color properties. To generate training samples, accelerated ray tracing is used to efficiently reconstruct shadow locations in the image using 3D ALS data.

Using shadow alone is not sufficient to detect accurate building changes, as shadows only partially represent 3D information. 3D information can be extracted from corresponding pixels in a stereo pair, but this information is not accurate in shadow and low texture areas. To address this, we propose LEAD-Matching (LiDAR-guided edge-aware dense matching). It starts from using accurate plane information extracted from ALS data to densify sparse ALS points. Three candidate heights are then obtained for each densified point to guide the dense matching in these problematic areas. Subsequently, detailed building information in the stereo pair is integrated to choose the final optimal height. If the optimal height obtained by LEAD-Matching points to corresponding pixels of different color, a likely building change is found. Test results on the Amersfoort and Assen data show a successful verification of unchanged buildings while changes are detected starting from $2 \times 2 \times 2\ m^3$, as conventionally required for large-scale 3D mapping, with an F1 score of 0.8 and 0.9 respectively.

To achieve the second objective, we extend LEAD-Matching to multiple stereo pairs, to improve the planimetric accuracy of building extraction in ALS data. E-LEAD-Matching integrates building boundaries of high planimetric accuracy from multiple stereo pairs to the ALS data. Using multiple stereo pairs, occlusions in

single stereo pairs are compensated, while the accuracy of building boundaries is improved. Compared to using ALS alone, the planimetric accuracy of extracted buildings improves from 0.40 m to 0.22 m in Vaihingen, and from 0.48 m to 0.21 m in Amersfoort. This improved planimetric accuracy actually meets conventional requirements of large-scale mapping.

Our methods enable us to integrate the beneficial aspects from ALS and ACIM to generate accurate and up-to-date large-scale 3D city models. We anticipate that our research will save both money and time in generating future up-to-date large-scale 3D city models.

# Samenvatting

3D-stadsmodellen worden steeds vaker gebruikt om de stedelijke infrastructuur te onderhouden en te verbeteren. In een tijd van sterk toenemende verstedelijking is het correct en actueel houden van 3D-stadsmodellen van essentieel belang voor gemeenten bij het nemen van beslissingen. Bij het genereren van 3D-stadsmodellen wordt veel gebruik gemaakt van via laserscannen vanuit de lucht (ALS = Airborne Laser Scanning) verkregen 3D-informatie. Echter, ALS data punten zijn relatief dun gezaaid, onregelmatig verdeeld over het gebied, en, met het oog op de hoge kosten, worden zij niet vaak ingewonnen. Camerabeelden vanuit de lucht (ACIM = Airborne Camera IMagery) vormen een alternatief voor het verkrijgen van compactere maar minder nauwkeurige 3D-informatie. Gezien deze beperkingen betreffende inwinningfrequentie en kwaliteit, is gebruik van enkel ALS of ACIM gegevens voor het genereren van actuele, grootschalige 3D-stadsmodellen niet optimaal.

Daarom combineren we de complementaire eigenschappen van beide gegevensbronnen om twee doelen te bereiken: (i) 3D-detectie van veranderingen en actualisatie van de gebouwde omgeving in ALS-gegevens met gebruikmaking van ACIM-gegevens, en (ii) verhoging van de nauwkeurigheid van gebouw extractie in ALS-gegevens met gebruikmaking van ACIM-gegevens. Voor het eerste doel worden ALS-gegevens met een enkel beeld of een enkel stereopaar geïntegreerd, terwijl voor het tweede doel meerdere stereoparen worden gebruikt. Onze methoden worden in drie gebieden gevalideerd: Vaihingen, Duitsland, en Amersfoort en Assen, Nederland.

In een enkel beeld duidt schaduw een 3D-object aan en dit wordt weergegeven door RGB-kleurenwaarden in het beeld. Deze kleurenwaarden zijn echter niet uniek, omdat zij afhankelijk zijn van de plaatselijke omstandigheden, zoals materiaal en omgeving. Ons voorstel betreft een aanpak met gecontroleerd automatisch leren, 'random forest', voor het effectief karakteriseren van de kleureneigenschappen. Voor het creëren van trainingsmonsters wordt gebruik gemaakt van 'accelerated ray tracing' voor efficiënte reconstructie van schaduwlocaties in het beeld met behulp van 3D ALS-gegevens. Alleen gebruik van schaduw is niet voldoende voor het nauwkeurig detecteren van veranderingen in bebouwing, omdat schaduwen slechts gedeeltelijke 3D-informatie weergeven. 3D-informatie kan uit overeenkomstige pixels in een stereopaar worden verkregen, maar deze informatie is niet nauwkeurig in gebieden met schaduw en lage textuur. Om dit probleem op te lossen, stellen wij LEAD-Matching (LiDAR-guided Edge-Aware Dense matching) voor. Hierbij wordt uitgegaan van gebruik van uit ALS-gegevens verkregen nauwkeurige informatie over vlakken om zo de schaarse ALS-punten te verdichten. Vervolgens worden voor ieder verdicht punt drie mogelijke hoogtes verkregen om de 'dense matching' in deze probleemgebieden te begeleiden. Daarna wordt gedetailleerde informatie over bebouwing in het stereopaar geïntegreerd om uiteindelijk de optimale hoogte te kiezen. Als de door LEAD-Matching verkregen optimale hoogte

naar overeenkomstige pixels van een andere kleur verwijst, is een verandering in de bebouwing opgespoord. Testresultaten op basis van de gegevens van Amersfoort en Assen tonen een succesvolle verificatie van onveranderde bebouwing, terwijl veranderingen vanaf $2 \times 2 \times 2\ m^3$ worden gedetecteerd, zoals conventioneel vereist voor grootschalige 3D-kartering, met een F1-score van respectievelijk 0,8 en 0,9.

Om het tweede doel te behalen, breiden we LEAD-Matching uit naar meerdere stereoparen, om zo de planimetrische nauwkeurigheid van gebouw extractie uit ALS-gegevens te verhogen. E-LEAD-Matching integreert planimetrisch zeer nauwkeurige grenzen van bebouwing afkomstig van meerdere stereoparen in de ALS-gegevens. Met gebruikmaking van meerdere stereoparen worden occlusies in enkele stereoparen vermeden, terwijl de nauwkeurigheid van de grenzen van bebouwing wordt verhoogd. Vergeleken met de toepassing van alleen ALS wordt de planimetrische nauwkeurigheid van de uitgelichte bebouwing in Vaihingen van 0,40 m tot 0,22 m en in Amersfoort van 0,48 m tot 0,21 m verbeterd. Deze verhoogde planimetrische nauwkeurigheid voldoet aan de conventionele eisen voor grootschalige kartering.

Onze methoden stellen ons in staat de goede eigenschappen van ALS en ACIM gegevens te integreren voor het genereren van nauwkeurige en actuele grootschalige 3D-stadsmodellen. Wij verwachten dat ons onderzoek zowel geld als tijd zal besparen bij het genereren van actuele, grootschalige 3D-stadsmodellen.

# 1
# Introduction

## 1.1. Motivation

3D geometric information extracted from airborne laser scanning (ALS) and airborne camera imagery (ACIM) has complementary characteristics, but is currently not routinely used in concert. We propose methods to combine and integrate 3D information from ALS and ACIM to generate accurate and up-to-date 3D city models.

## 1.2. Background

According to the United Nations, in 2007, our world population was more urban than rural for the first time (United Nations, 2007). To date, 55% of the world population is living in cities and this proportion is expected to increase to 68% by 2050 (United Nations, 2018). There is a need to constantly maintain and improve the urban infrastructure to serve the increasing population. 3D city models are digital models of an urban area that represent, e.g. terrain surface, buildings, vegetation, infrastructures, belonging to city areas. Municipalities increasingly require 3D city models to support various applications, such as water management and urban planning (Zhu et al., 2009). Keeping 3D models up-to-date is essential for decision making. For example, accurate and up-to-date 3D information on building roofs is required for solar irradiation estimation to assess the efficiency of solar panels (Biljecki et al., 2015). In addition, the urban heat island is mainly caused by the high density of buildings in the city. This hampers air circulation, while the presence of roof result in an increased solar radiation absorption (Mirzaei, 2015). A heat flow simulation requires accurate 3D buildings and materials to help to design an efficient strategy to cool down the city. As different stakeholders have different requirements for 3D city models, a nation wide general model is necessary to be created to serve as a basis, which can be further refined and developed for different applications (Oude Elberink et al., 2013).

The automatic reconstruction of 3D city models became an important research theme in photogrammetric and computer vision communities in recent decades (e.g. Gruen et al., 1995; Haala and Kada, 2010; Wang, 2013). A high level of 3D geometric information, such as complex roof structures, is key for 3D city modelling. Airborne laser scanning systems use Light Detection And Ranging (LiDAR) technology to acquire 3D information using a light in the form of a laser pulse to measure the range to 3D objects, while airborne camera imagery systems use photogrammetry techniques to acquire 3D information using cameras to collect

overlapping very high resolution (VHR) images in different positions. ALS and ACIM with their capability of collecting detailed 3D information are commonly used for 3D city modelling (Haala and Kada, 2010), as further elaborated below.

### 1.2.1. Airborne laser scanning

Airborne laser scanning equips a LiDAR system, combined with Global Navigation Satellite Systems (GNSS) and Inertial Measurement Unit (IMU) to determine the position and orientation of the LiDAR system respectively. The LiDAR system measures the range distance between the LiDAR system and the terrain surface, while the position and orientation of the LiDAR system provided by GNSS and IMU transforms the ranges to 3D coordinates (Vosselman and Maas, 2010). Repeating this procedure from different locations, a point cloud with 3D coordinates is obtained as shown in Figure 1.1a. The explicit geometric information of LiDAR point clouds facilitates the automatic reconstruction of 3D city models.

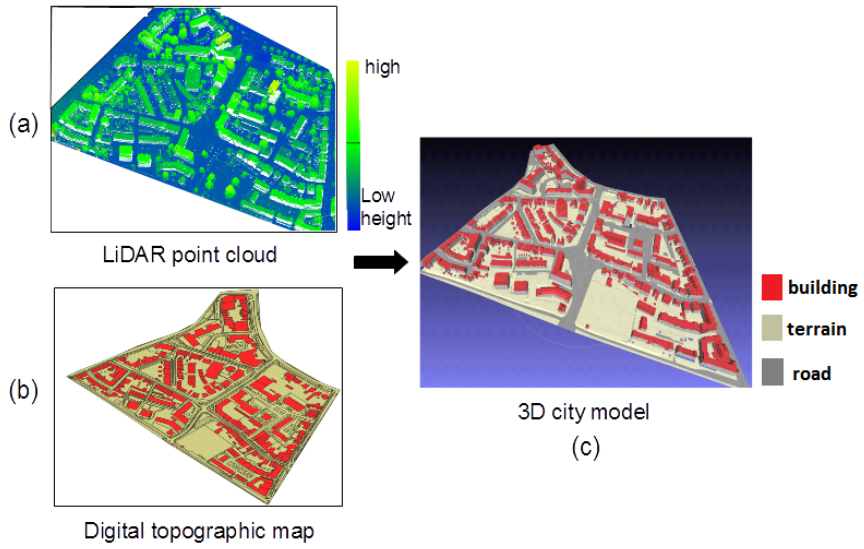



Figure 1.1: (a) Dutch national LiDAR point clouds from AHN2 colored by height. (b) Dutch digital topographic map from BGT with polygons colored by different classes. (c) A 3D city model created by combining the LiDAR point cloud and the digital topographic map using the Software 3DIMGeo from University of Twente. (Oude Elberink et al., 2013). Three classes of objects, building, terrain and road, are included and colored in the model.

The first prototype of a commercial ALS system aiming for topographic mapping became available in 1993 (Flood, 1999). Since then, LiDAR point clouds with continuously increasing availability, density and accuracy facilitate studies on a standard and robust approach on creating 3D city models for large areas (Vosselman et al., 2001; Zhou and Neumann, 2009; Lafarge and Mallet, 2012). The Netherlands was the first to launch a national mission, called Actueel Hoogtebestand Nederland (AHN), to acquire airborne laser scanning data from

1997 (PDOK, 2019) and publish the data as an open source. Oude Elberink et al. (2013) proposed to create a 3D city model at a national level combining AHN with another open source digital topographic map, Basisregistratie Grootschalige Topografie (BGT) as shown in Figure 1.1b. The topographic map provides not only object polygons, but also class information. Five basic object classes are included in the 3D model defined by the classes from the topographic map. These classes are water, road, terrain, buildings and forest. Vegetation can be an additional class or be included in the terrain class. These objects together form a seamless terrain which is important for many applications, such as simulating water flow for flood prediction. The 3D model with the class information is shown in Figure 1.1c. The data sources, a topographic map and a 3D point cloud, for creating the 3D city models are generic and easy to obtain. So this procedure is applicable worldwide. However, updating ALS data at nation scale often takes years. In the Netherlands, AHN updates the ALS point clouds for the whole nation (Van Der Sande et al., 2010) at 5-10 years intervals.

### 1.2.2. Airborne camera imagery

Meanwhile, airborne cameras imagery acquire very high resolution (VHR) stereo images with overlaps using GNSS and IMU to determine camera's position and orientation respectively. As camera systems are less expensive than ALS systems, ACIM images are often yearly updated at national scale. 3D geometric information can be also extracted from a single image or stereo images. Within a single image, shadow not only causes radiometric distortions which should be primary considered for image interpretation, but also indicates height differences of 3D information. In addition, with development of dense image matching (Furukawa and Ponce, 2010; Hirschmuller, 2008), 3D point clouds can be reconstructed from stereo images, which consists of at least two overlapping images, also can be multi-view images, acquired by the camera from different positions (Förstner and Wrobel, 2016). The dense image matching aims to find corresponding pixels that indicate the same 3D point. This procedure is shown in Figure 1.2a, while two images out of multi-view images acquired by a camera from different positions are shown in Figure 1.2b left. Due to the high ground sampling distance (GSD) of ACIM VHR images, a denser point cloud than the ALS point cloud can be obtained. In addition, the color information obtained by the airborne camera is assigned to the geometric information in the photogrammetric point clouds as shown in Figure 1.2b, which can be used to improve the classification of different objects (Vosselman, 2002; Zebedin et al., 2006). Therefore, ACIM images provide an alternative to generate a 3D city model using a single data source. Baillard and Maıtre (1999) already used both color and geometric information from stereo images to construct a 3D city model with buildings and trees two decades ago. However, the quality of 3D photogrammetric point clouds is largely affected by shadows, low texture and repetitive patterns, as finding corresponding pixels to reconstruct 3D information relies on texture or color similarity of pixels. This is shown in the red boxes in Figure 1.2b. These problems prevent a robust and accurate processing workflow to generate accurate 3D city models automatically at national scale.

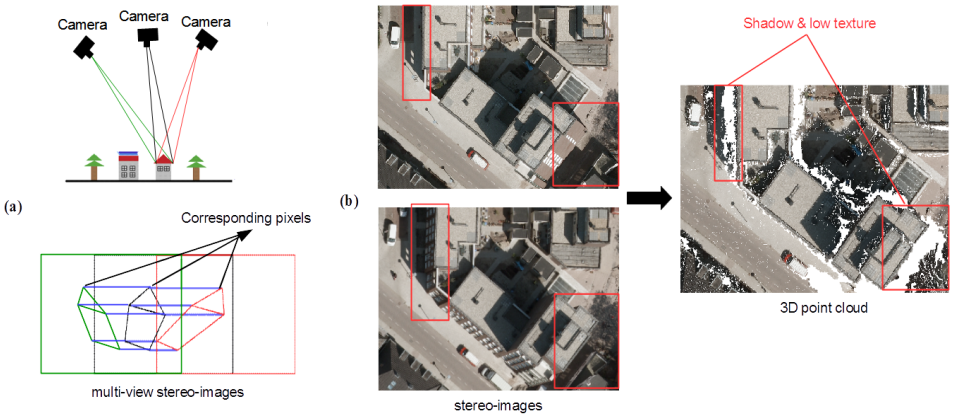



Figure 1.2: (a) Airborne cameras acquire stereo images showing the same 3D object in different positions. 3D point clouds can be extracted by finding corresponding pixels from stereo images. (b) 3D point cloud with color information extracted from VHR multi-view images. Two images from multi-view images are presented. In the red boxes, the quality of 3D point clouds is affected by shadow and low textures in the images.

According to the description above, when relying on 3D information from a single data source, it is difficult to generate 3D city models with accurate and up-to-date 3D information at national scale. In addition, objects on the ground, such as roads and grass, are difficult to be updated using 3D information. The most prominent 3D objects above the ground in the cities are buildings, which often change due to urbanization. Therefore, this research focuses on combining the two data sources from LiDAR and photogrammetry to generate up-to-date and accurate 3D city models, with a focus on buildings, aiming at nation scale.

## 1.3. Problem statement

As urban areas typically do not change dramatically each year, a large portion of buildings is still valid in the infrequent ALS data. Even though the quality problems of 3D information extracted from airborne images prevent the generation of an accurate 3D city model from scratch, airborne images are actually a good resource to update and improve 3D information in ALS data for large-scale maps. However, the different characteristics of the two data sources still prevent an accurate combining of their complementary information. Until recently, the integration of two data sources has not been commonly applied. As up-to-date 3D city models, focusing on buildings, can be generated by combining the 3D buildings updated from the ACIM images and unchanged 3D buildings in the ALS data, in this section, we focus on discussing the problems in combining the two data sources for two objectives: (i) accurate 3D change detection and updating of buildings in ALS data using ACIM images and (ii) improving accuracy of extracted buildings from ALS data using ACIM images. To achieve objective (i), newly acquired images are

required. To achieve objective (ii), the acquisition time of images can either be similar to or more recent than the acquisition time of the ALS data. As we already restrict our data to be acquired on the airborne platform, LiDAR and image data are used to represent ALS and ACIM data respectively for the simplicity.

### 1.3.1. 3D change detection and updating

There are two types of information in VHR images which can be used for change detection in LiDAR data: color and 3D geometry. Color information has been used for several decades for building extraction (e.g. Shufelt, 1999; Wei et al., 2004; Yuan, 2018). The extracted buildings can be compared with the buildings extracted from LiDAR data for change detection. The high spatial resolution of images gives an opportunity to see objects much more clear, however, the spatial resolution refinement causes also color variations between pixels from the same class. (Cushnie, 1987; Thomas et al., 2003). For example, as shown in Figure 1.2b, the color and texture of different building roofs vary, while many building roofs look similar to streets. Object-based detection and classification in VHR images have been well studied in the last decades (Blaschke, 2010). Instead of using features in an individual pixel, shape, texture and even context features can be extracted from objects obtained from segmentation. However, this approach strongly relies on the segmentation results. Convolutional neural networks (CNN) draw increasing attention in interpreting VHR images (Yuan, 2018; Marmanis et al., 2018) due to their ability to automatically learn the most useful features for classification, instead of the hand-crafted features which are manually designed. The context for extracting effective features is no longer restricted to the segments, but learned by the networks. However, CNNs often require a large amount of labeled training data and are domain specific (Wang et al., 2017b). The state-of-the-art performance of deep neural networks is mostly due to extensive training on large-scale benchmark datasets. However, benchmarks for VHR aerial images are limited and CNNs trained on one dataset do not generalize well to the aerial images from another area (Zhou et al., 2019a). Besides, terrain relief causes relief displacement in VHR images, especially for high buildings (Habib et al., 2007). As shown in Figure 1.2b, the buildings in the stereo images are displaced differently, especially the high buildings. Only if accurate building heights, and camera position and orientations are known, true ortho-rectification (Zhou et al., 2005) can be applied to correct the displacement.

However, if accurate 3D geometric information from images is obtained, this information can be used directly and primarily for change detection in LiDAR data. Therefore, the key to building change detection when combining LiDAR and image data is 3D geometric information. 3D information is explicitly present as point clouds in LiDAR data, while 3D information can be extracted from single images and stereo images. As an increasing number of images contains more 3D information but also increases the difficulties and computation efforts, change detection are discussed using 3D information in a single image and stereo images as follows.

**Change detection on LiDAR data using a single image**
Shadow cast by height differences on the terrain is an indicator of 3D information for buildings from a single image. Several studies (Lin and Nevatia, 1998; Sirmacek and Unsalan, 2008; Ok, 2013) have used shadows for building detection. The advantage of using shadows to indicate 3D information is that the processing workflow is simpler than extracting 3D information from stereo images. In addition, shadow has relatively simple spectral characteristics and is easier to detect than buildings in images. Deriving shadow changes relies on two aspects. First, reconstructing shadows from LiDAR data is required. Shadow reconstruction means to estimate a shadow image using 3D information in LiDAR data with the same sun position and camera parameters as the airborne image as shown in Figure 1.3a. The fundamentals of projecting 3D information, including shadows, to a camera image plane is explained in chapter 2. Second, detecting shadow from VHR images, representing up-to-date 3D information, is required. With respect to the accurate change detection using shadows, there are three problems as follows.

1) Accurate shadow reconstruction is time consuming and requires an accurate and watertight 3D model. Shadow reconstruction often requires to reconstruct an image with more than millions of pixels using a 3D model with millions of triangles. The quality of watertight 3D model automatically reconstructed from LiDAR point clouds often suffers from modelling errors.

2) Accurate shadow detection is also difficult as spectral properties of shadows are environmental and material dependent in urban VHR images (Adeline et al., 2013). Traditional methods (Tsai, 2006; Adeline et al., 2013) studying the spectral property of shadows fail to take environmental factors and object materials into consideration.

3) Shadows can not indicate all 3D information as they are cast from a single direct only. The low building adjacent to buildings may not cast shadows as shown in Figure 1.3a.

**Change detection on LiDAR data using stereo images**
As 3D point clouds can be extracted from stereo images, a direct approach to detect changes between LiDAR and photogrammetric point clouds is to compare the two point clouds. Therefore, the accurate change detection relies on the quality of 3D information in both datasets. Several quality problems need to be solved:

1) LiDAR points are sparse and irregularly spaced.

2) Photogrammetric points are missing or outlying in problematic areas with low texture and shadow. Occlusions often happen when only two-view stereo images are used.

3) The varying and different point density of LiDAR and photogrammetric points introduce difficulties to find corresponding points to compare.

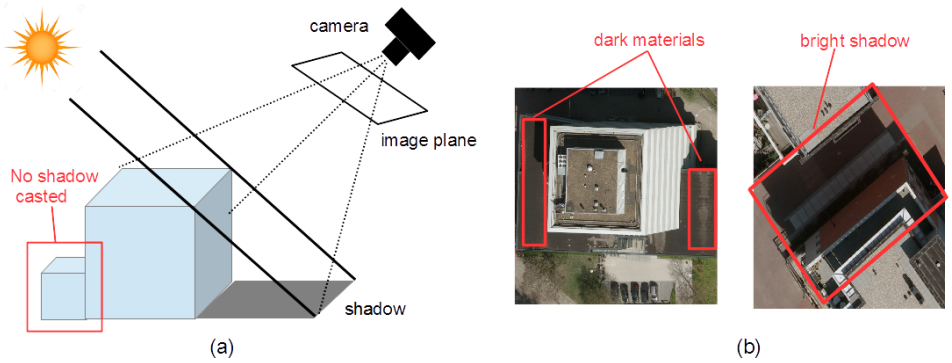



Figure 1.3: (a) Shadow reconstruction by projecting the 3D information to the 2D image plane in the camera. The object indicated in the red box does not cast shadows. (b) The roof is dark due to the roof material, while the shadow is bright due to environmental reflections.

First, due to the unstable aircraft velocity and survey height, and reflectivity of object materials, the point density is varying and irregularly spaced. The average point density of LiDAR data, e.g. the AHN part 2 (AHN2), acquired between 2007 and 2012, is 10 points per square meters (ppm) (Van Der Sande et al., 2010), corresponding to an average ground sampling distance (GSD) of around 35 cm. In some areas, the point density of LiDAR can be even less in practice as laser pulses may be reflected off or penetrate through the objects. As shown in the red boxes in Figure 1.4a left, laser pulse are reflected and less than 5 points fall into $1\,m^2$. The point cloud is sparse when compared to VHR images with an average GSD of less than 10 cm.

Second, as point clouds are extracted from stereo images using dense image matching by searching for corresponding pixels based on color and texture in different images, in shadow or low texture areas as shown in Figure 1.4b left, it is difficult to find corresponding pixels correctly. Therefore, 3D points extracted in these areas are incomplete or outlying as shown in Figure 1.4b right. The size of these effects can be a few meters, resulting in a large amount of uncertain areas or false alarms in change detection. If only a two-view of stereo images is used, some areas may be occluded in the the images, as shown in Figure 1.4b left. No 3D points can be reconstructed from the occluded areas. If a stereo pair with small viewing angle on the research area with less densely located high buildings, the occlusions are limited.

Third, the LiDAR and photogrammetric point clouds have varying and different point densities as shown in Figure 1.4. It is difficult to find corresponding points for each point in the LiDAR or image data to detect changes. One approach to obtain the same point density is to interpolate two point clouds to a digital surface model (DSM) with the same grid size. However, the quality of interpolation on photogrammetric point clouds is affected by the quality problems in shadow and low texture areas, while quality of interpolation from the LiDAR point cloud

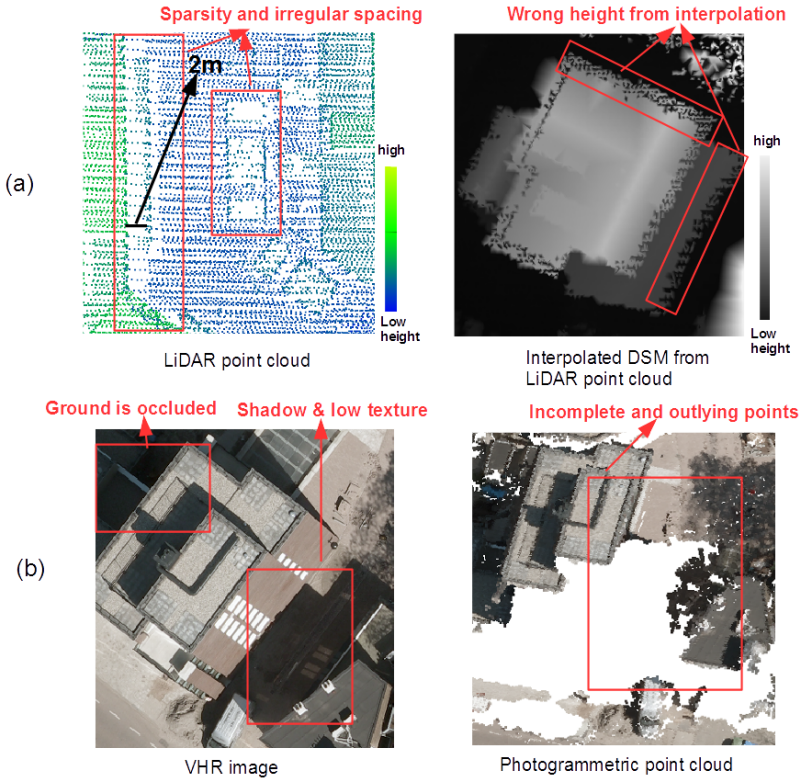



Figure 1.4: (a) Left: the LiDAR point cloud is sparse and irregularly spaced. Right: the interpolated DSM has wrong heights near roof boundaries due to mix return points. (b) Left: the shadow, low texture and occlusion areas are indicated in the VHR image. Right: in the indicated area, points extracted are incomplete or outlying.

is affected near building edges due to mixed points from roof, wall and ground in top view, especially near the edges of overhanging roofs. Wrong heights are interpolated as shown in Figure 1.4a right. This width of the area with wrongly interpolated heights can be more than 1 meter.

### 1.3.2. Building extraction from LiDAR data

The buildings from infrequent LiDAR data are still valid when they do not change between the acquisition time between LiDAR and image data. The building extraction accuracy relies on extraction successful rate and planimetric accuracy (Gilani et al., 2016). The extraction successful rate defines how well the buildings are differentiate from other objects, while the planimetric accuracy defines how well the building boundaries are extracted. In literature, main focus has been on improving the successful rate by better segmenting the buildings from the other objects in LiDAR data, image data or both (Awrangjeb et al., 2013). However,

the planimetric accuracy of building extraction is also important in order to meet the requirements of large scale mapping. For example, a map scale of 1:1000 and 1:2000, requires planimetric accuracy of building boundaries are 0.25 m and 0.50 m respectively (Merchant, 1987).

Using LiDAR data alone, the planimetric accuracy of building extraction depends on the actual point spacing of LiDAR data (Gerke and Xiao, 2014). The buildings extracted from LiDAR data are often smaller than their actual size due to the sparse and irregular spaced LiDAR points. For certain buildings with even lower point density, the complete building area is hardly to be extracted as shown in the red boxes in Figure 1.2a left. As airborne images are more frequent, and have smaller ground sampling distance (GSD) and more detailed building boundaries, they can also be used to extract buildings. However, the relief displacement of buildings should be addressed by extracting accurate 3D information. As we aim at obtaining planimetric accuracy to 0.25 cm or 0.5 cm, multi-view images should be used as in this way building boundaries in the multiple camera position are captured and occlusions are reduced. Still 3D information is largely affected in shadows and low texture areas remaining in the multi-view images. As shown in Figure 1.5, if the DSM from a provided photogrammetric point cloud is used to correct relief displacement, the building boundaries in the ortho-rectified image are distorted.

Therefore, images with detailed building boundaries can be integrated to LiDAR data to improve the planimetric accuracy of building extraction. If quality problems of 3D information in both data sources is addressed in Section 1.3.1 for change detection, the same method can be applied, but multi-view images should be used, for improving planimetric accuracy of building extraction.

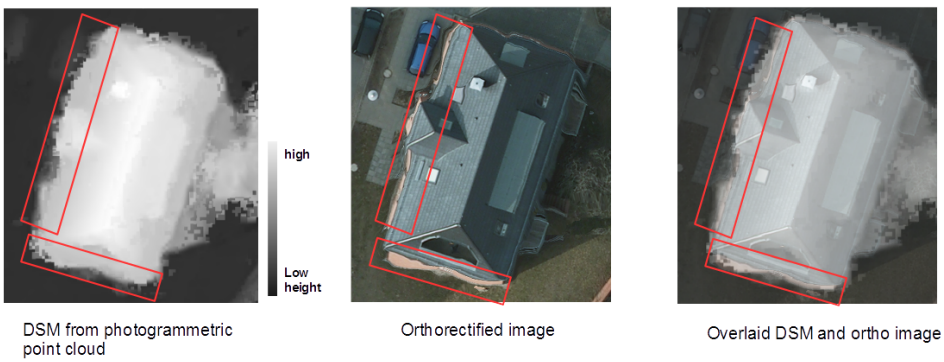



Figure 1.5: Left: a DSM interpolated from a photogrammetric point cloud. Middle: the ortho-rectified image derived using the DSM. Right: Overlaid DSM and ortho-rectified image. The accuracy of the DSM determines the accuracy of colors near building edges. The data from Potsdam is from an ISPRS benchmark test (ISPRS WG III/4, 2019).

## 1.4. Research objectives

Generating an accurate up-to-date 3D city model, focusing on buildings, is largely affected by quality problems in the 3D information extracted from both LiDAR and image data. 3D information extracted from a single data source is problematic. However, the complementary information in both LiDAR and image data can be combined. The main research question of this thesis reads:

**How to integrate 3D information from airborne laser scanning and airborne camera imagery to create an accurate and up-to-date 3D city model, focusing on buildings?**

As mentioned in Section 1.3, we focus on two objectives for generating accurate and up-to-date 3D city models: (1) accurate 3D change detection and updating of buildings in LiDAR data using VHR images, and (2) improving the planimetric accuracy of building extraction from LiDAR data using VHR images. Based on the problems stated in Section 1.3, the following questions are derived.

1) How to detect shadows from a single image by integrating information from LiDAR point clouds?

   The accurate change detection depends on the accurate shadow reconstruction from infrequent LiDAR data and shadow detection from up-to-date image data. However, accuracy of shadow reconstructed is often affected by error of 3D modelling from LiDAR data, which is out of scope of our research. In addition, shadow only represents a partial of 3D information as shown in Figure 1.3a, which largely affects the accuracy of change detection. Therefore, only shadow detection from single images is studied by integrating shadows reconstructed from LiDAR point clouds. A machine learning approach is proposed to use reconstructed shadow from LiDAR data as training samples for shadow detection. This approach addresses the problem that shadows are environmental and material dependent in urban VHR images as stated in Section 1.3.1. The following sub-questions will be addressed:

   1.1. How to efficiently reconstruct shadows from LiDAR data to generate training samples for a subsequent machine learning approach?

   1.2. How can we choose a machine learning approach using training samples provided by LiDAR data to detect shadows in a single image?

2) How to detect accurate 3D changes and perform 3D updating by integrating 3D information from LiDAR point clouds and a stereo pair?

   The quality problems of 3D information in LiDAR point clouds and VHR stereo images, as described in section 1.3.1, prevents accurate change detection between these two data sources. Complementary 3D information from the two data sources can be used to mitigate quality problems. As the requirement of minimum detectable size is $2 \times 2 \times 2$ m$^3$, we select a stereo pair which has relatively small occlusions in the research area for change detection on LiDAR data. Here we address the following sub-questions:

2.1. How can LiDAR point clouds be integrated to a stereo pair to improve the quality of 3D information extracted from a stereo images, especially in regions affected by shadow and low texture ?

2.2. How can a stereo pair be integrated to LiDAR data to address the problem of sparsity and irregular spacing in LiDAR point clouds, especially near edge areas?

2.3. How to design a framework for accurate 3D change detection and updating in LiDAR data using a stereo pair?

3) How to integrate 3D information from LiDAR data and multi-view images to improve the planimetric accuracy of building extraction?

The high ground sampling distance (GSD) and detailed building boundaries in VHR images provide an opportunity to extract building with higher planimetric accuracy than the buildings extracted from LiDAR point clouds with sparse and irregularly spaced points. However, the quality problem of 3D information extracted from stereo images directly affects the planimetric accuracy. The same idea of combining complementary 3D information from LiDAR data and now multi-view images is applied to reduce occlusions and ensure the detailed boundary information of different sides of building well represented. In this context, the sub-questions are as follows:

3.1 How to integrate 3D information from both LiDAR data and multi-view images to reduce effect of occlusions in building extraction?

3.2 How to integrate LiDAR data with detailed boundary information from multi-view images to improve planimetric accuracy of building extraction?

## 1.5. Scope and limitations

To narrow down the research scope, in this thesis, only buildings are considered within 3D city models. The minimum size of 3D building change and updating is $2 \times 2 \times 2$ m$^3$, which is considered as the smallest area people can live in. Changes of dormers on building roofs are not considered. In addition, accurate geo-referencing of image and LiDAR data is assumed to be performed before the research. First, the camera intrinsic and extrinsic parameters are considered to be estimated accurately and provided by bundle adjustment. Second, the strip adjustment of LiDAR data is considered to be performed accurately. The misalignment between LiDAR and image data is considered to be small, less than 10 cm, due to their accurate geo-referencing. Finally, improving 3D information for individual data sources separately is not considered in this study.

## 1.6. Organization of the thesis

This chapter briefly introduced the background of the research, and the problems and objectives of this study. The main research question is formulated and subdivided into three questions. These questions will be addressed in the following chapters.

Chapter 2 provides an overview of the mechanism and problems of acquiring 3D information of high quality from ALS and airborne cameras. The fundamental photogrammetric theory of combining 3D information from LiDAR point clouds and stereo images is explained.

As mentioned above, Chapter 3 only focuses on shadow detection from a single image which is linked to question 1. A machine learning approach is developed to detect shadows adaptive to complex scenes in VHR images by integrating shadows reconstructed from LiDAR data as training samples. This part is based on a *Remote Sensing* article published in January 2019 (Zhou et al., 2019b).

Chapter 4 is relevant to question 2 and proposes an integrated approach of using complementary 3D information from LiDAR point clouds and a stereo pair to explicitly address quality problems for 3D change detection and updating. An up-to-date 3D city model is generated from integrated point clouds. In unchanged areas, point cloud information is from LiDAR data and in changed areas, point clouds are from new stereo images. This chapter is based on a *ISPRS Journal of Photogrammetry and Remote Sensing* article in February 2020 (Zhou et al., 2020b).

Chapter 5 is related to question 3 and applies the integrated approach in Chapter 4 on multi-view images to improve building extraction in LiDAR data. This chapter is based on an article which is currently on review (Zhou et al., 2020a).

Conclusions and recommendations for future work are given in Chapter 6.

# 2
# 3D geometry in LiDAR and photogrammetry

*In this chapter, a brief overview of the characteristics of the 3D geometric information provided by airborne LiDAR scanning (ALS) and airborne camera imagery (ACIM) is given in section 2.1-2.2, including the reason of the quality problems described in Chapter 1. As a starting point for combining the two data sources, fundamental photogrammetric notions for transforming 3D information from ALS point clouds to ACIM stereo images are described in section 2.3.*

## 2.1. Airborne laser scanning

In this section, the mechanism of how airborne laser scanning (ALS) systems to acquire 3D information is described. Furthermore, the characteristics of the 3D information are discussed with a focus on the introduced quality problems in Chapter 1.

### 2.1.1. Airborne laser scanning system

An ALS system combines a laser ranging system, a Global Navigation Satellite System (GNSS) and an Inertial Measurement Unit (IMU) in an aircraft as shown in Figure 2.1. The laser ranging system, using Light Detection and Ranging (LiDAR) technique, measures the distance to a spot on the terrain surface by illuminating the spot with a laser light while the GNSS and IMU measures the position and the orientation of ALS system respectively (Vosselman and Maas, 2010). Combined with the GNSS and IMU, the ranges measured by the laser are converted to 3D data and geo-referenced to a geodetic coordinate system, such as WGS 84 (World Geodetic system), ETRS89 (European Terrestrial Reference System 1989) or Amersfoort/RD New (Dutch geodetic coordinate system). The final output of an ALS system is a 3D point cloud with geo-referenced coordinates.

A laser system emits and receives millions of laser pulses to and from the terrain surface as shown in Figure 2.1a. The range is generally determined in two different ways, either time-of-flight (ToF) or continuous-wave (CW) (Vosselman and Maas, 2010). The ToF records the precise time between emitting the laser pulse and receiving the return pulse. Combined with the speed of light, the range is calculated. CW systems record the precise phase difference between the emitted and returned laser pulse to calculate the range. The range relies on the GNSS

and IMU system to provide accurate position and orientation of the laser system to be transformed to accurate 3D coordinates in a geodetic coordinate system. Therefore, a differential GNSS is often required with a network of fixed ground-based reference stations as shown in Figure 2.1 to broadcast the corrections for the positions indicated by GNSS. Applying with differential GNSS, the accuracy of positioning can be improved to centimeter level. The IMU consists of inertial sensors providing the accurate orientation of the aircraft in terms of roll, pitch and yaw. A laser system is often mounted with its **X**-axis pointing towards the right wing, the **Y**-axis pointing towards the tail and the **Z**-axis pointing down. The yaw, pitch and roll angles further determine the laser beam orientation.

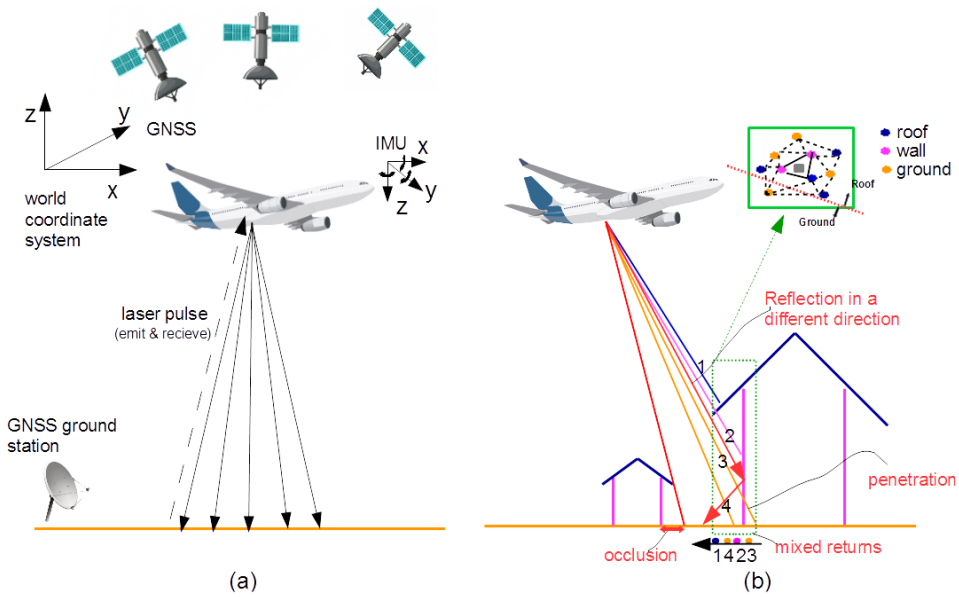



Figure 2.1: (a) An airborne laser scanning system equipped with a laser ranging system, a GNSS and an IMU. The laser system emits and receives laser pulses to measure the range to the terrain surface. (b) Several problems of 3D geometry from ALS point clouds. Some object surfaces can reflect laser pulses in a different direct, while some object surfaces can be penetrated by the laser pulses. In addition, the laser pulse may also be occluded in some areas. These effects all contribute to the problem that points are missing at certain object surfaces. Besides, the order of points may get shuffled when projecting them onto the ground due to the effect of overhanging roofs and material penetration. As a consequence, roof, wall and ground points get mixed as shown in the green box.

### 2.1.2. 3D geometry from ALS point clouds

The standard vertical and horizontal accuracy of ALS point clouds is 0.05–0.2 m and 0.2–1.0 m respectively (Vosselman and Maas, 2010). Van Der Sande et al. (2010) performed an assessment on the relative accuracy of the Dutch national ALS data, AHN2, using overlapping strips and found that the vertical accuracy

is better than 0.04 m, while the planimetric accuracy is 0.02–0.34 m. The point density can vary between 0.2 and 50 points per square meter due to the aircraft velocity and survey height (Vosselman and Maas, 2010). This is the main reason why the points are irregularly spaced. An average point density is 10 points per square meter Van Der Sande et al. (2010). Compared to the point clouds extracted from image data, as will be describe below, ALS point clouds are sparse. This sparsity and irregular spacing can be increased when an object surface reflects the laser pulse in a different direction or can be penetrated by the laser pulse. Both effects are shown in Figure 2.1b. These points will be missing at the considered object surfaces. In addition, the laser pulses may be occluded at some part of a terrain surface and points are also missed as also shown in Figure 2.1b.

Due to the relatively large ground sampling distance, the laser pulses from ALS will not exactly sample the building edges. Therefore, buildings extracted from ALS point clouds are often smaller than the ground truth. In addition, if the roofs are overhanging or walls are made of glass which are penetrable to laser pulses, the order of ALS points get shuffled when projecting them to the ground. In Figure 2.1, when the points from the laser pulses [1 2 3 4] are projected to the ground surface, the order becomes shuffled to [3 2 4 1]. When displaying the area into a 2D space, the points from the roof, wall and ground get mixed as shown in the green box in Figure 2.1b. If interpolation is performed to densify the sparse and irregular ALS points to a digital surface model (DSM) with a higher ground sampling distance and uniform point spacing, the wrong heights will be interpolated as indicated in Section 1.3.1 in Figure 1.4(a).

## 2.2. Airborne camera imagery

In this section, the mechanism of airborne camera imagery (ACIM) systems to acquire stereo images is described. Further, the characteristics of the 3D information extracted from stereo images are discussed with a focus on the quality problems introduced in Chapter 1.

### 2.2.1. Airborne camera imagery system

An ACIM system combines a very high resolution (VHR) camera to acquire very high resolution (VHR) images, with GNSS and IMU to determine camera position and orientation. The images are acquired by a pinhole camera model which will be described in section 2.3.1. Extracting 3D information from images requires stereo images, which consist of at least two overlapping images acquired by a camera from different positions (Förstner and Wrobel, 2016). The three or more stereo images can be called multi-view images. The camera system captures at least 60% overlap in the forward direction, the flight direction, to ensure the full scene covered by at least two stereo images. Also around 60% of the objects appear in the three images (Sandau, 2009). In side direction which is perpendicular to the flight direction, 30% overlap is often taken. Ideally, around 50% and 40% objects appear in 3-view and 4-view images respectively. There is also a tendency to increase the overlaps in the urban scene to compensate occlusions from city canyons. The

ground sampling distance (GSD) of VHR images is often less than 10 cm. Geo-referencing is often performed by combining GNSS and IMU with ground control points through bundle block adjustment (Sandau, 2009). During geo-referencing, the camera intrinsic and extrinsic parameters are accurately estimated, such that the back-projection error of 3D points in the image can be smaller than GSD (Sandau, 2009). The details of camera intrinsic and extrinsic parameters will be elaborated in section 2.3.1.

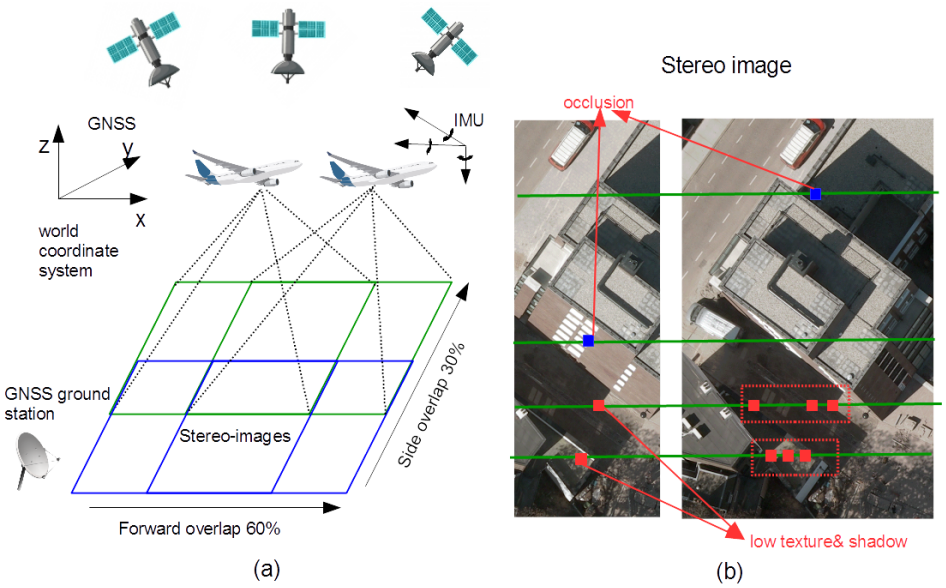



Figure 2.2: (a) An airborne camera system equipped with a camera, a GNSS and an IMU. The camera system captures overlapping images often with a requirement of 60% and 30% overlap in forward and side direction. (b) The stereo images are stereo rectified to align corresponding pixels in the same row. Some low texture or shadow pixels are marked red in the left image. Several pixels of similar color in the right image,as indicated, could be the corresponding pixel for the pixels in the left image. In addition, some pixels marked blue are occluded from the other image.

### 2.2.2. 3D geometry from ACIM stereo images

Different with ALS systems using range measurements directly to obtain 3D information, ACIM reconstructs 3D information using the positions of the corresponding pixels in stereo images, indicating the same 3D objects, combined with the camera intrinsic and extrinsic parameters to estimate 3D points from the correspondence. Dense image matching is key of 3D reconstruction to search for the corresponding pixels. However, searching for the accurate corresponding pixels through full images is difficult and computational intensive. The geometry of camera models of each stereo pair defines a so-called epipolar geometry, which constrains the search space for the corresponding pixels to a line. This is an ef-

fective step for reducing the search space from 2D to 1D. The stereo rectification further determines a transformation of each camera image plane such that the lines become aligned in the same columns or rows of the stereo pair. As shown in Figure 2.2, the corresponding pixels in two images are aligned in the same row or column after stereo rectification. The difference between column(or row) values of corresponding pixels is called disparity. Then dense matching can be defined to estimate the disparities for finding accurate corresponding pixels in the stereo images. Therefore, extracting 3D point clouds from stereo images often requires two steps: stereo rectification and dense matching. As our research does not focus on improving dense matching from stereo image alone, our interest is to combine 3D information from ALS and ACIM. Therefore, epipolar geometry and stereo rectification are elaborated in section 2.3.3 which is an important step to transform the 3D point clouds to disparities, while the fundamentals of dense matching can be found in paper of Scharstein and Szeliski (2002) and Remondino et al. (2014).

Due to the high ground sampling distance (GSD) of airborne VHR images, the pixel-wise matching would create a photogrammetric point cloud with hundreds of points per square meter which is in general much more than the points in ALS point clouds. The accuracy of photogrammetric point clouds is varying mainly according to varying performance in finding corresponding pixels. In area with rich textures, the accuracy is comparable to ALS point clouds. However, in shadow, low texture and repetitive pattern areas, the point accuracy is lower (Stal et al., 2013). As shown in Figure 2.2b, in the left image, one pixel is in the shadow while another pixel has low texture. In the right image, several similar pixels can be selected as potential corresponding pixels as indicated. Therefore, it is difficult to choose the correct corresponding pixel in these problematic areas. The quality of points extracted from these problematic areas will be largely affected. In addition, some pixels can be seen from one image but occluded in the other images as shown in Figure 2.2b. These pixels cannot find accurate corresponding pixels and often cannot reconstruct accurate 3D points.

## 2.3. Photogrammetric fundamentals for combining ALS and ACIM

Combining ALS and ACIM data requires to transform 3D information from one approach to the other. If disparities in ACIM stereo images are transformed to 3D point clouds, the quality problems of the photogrammetric point clouds will directly affect the result in combining the 3D information from the two data sources. As ALS points have high accuracy, transforming 3D ALS point clouds to disparities in stereo images is used for our research. In this section, the photogrammetric fundamentals to transform 3D point clouds to disparities are elaborated. It starts from transforming 3D point clouds to pixels in a single image using two approaches, perspective projection and ray tracing. Stereo rectification is to transform positions of corresponding pixels to a single value, disparity, when the corresponding pixels are found by transforming a 3D point to a stereo image pair.

### 2.3.1. Perspective projection

This section explains the fundamental theory of perspective projection, which is used by the so-called pinhole camera model to project 3D points onto the image plane in the camera. In this model, the light rays all go through the infinitely small pinhole, or aperture in the camera, while the intersections of the light rays to the image plane create 2D objects in the image as shown in Figure 2.3a. This theory has been intensively studied in two different ways from computer vision and photogrammetry. In our point of view, there are two main differences: (1) in computer vision, the perspective projection is formed as a matrix multiplication using homogeneous coordinates which will be explained later, while in photogrammetry, a simplified form, the collinearity equation, is derived. The collinearity equation has a more concise form but is less computational efficient than matrix multiplication which has been highly paralleled in programming. (2) The definitions of object, camera and image coordinate systems are different between computer vision and photogrammetry. In practice, for processing airborne images, the intrinsic and extrinsic parameters of airborne camera are accurately calibrated and defined in the coordinate systems in the photogrammetric point of view. Therefore, in computer vision, the matrices constructed using these parameters should be adapted.

In this section, we explain perspective projection in the form of the matrix multiplications from computer vision point of view, while the camera intrinsic and extrinsic parameters from airborne camera system are used to form the matrices respectively. The definition of different coordinate systems used in computer vision and photogrammetry are explained to adapt the matrices defined in computer vision using camera parameters from an airborne camera system. Finally, the perspective projection matrix is derived which has been used as a core in this research. The collinearity equation is also derived from the perspective projection matrix to show the consistency of perspective projection between computer vision and photogrammetry. This section is mainly based on the books of Kaehler and Bradski (2016) and Schenk (2005), which explain the theory from computer vision and photogrammetry point of view respectively.

**Matrix of intrinsic parameters**

Figure 2.3b illustrates the detailed geometry of the camera model. The camera model consist of a *projection center* (pinhole) $C$ and an image plane. The perspective projection is to project a 3D point $W$ on a 2D image point $M$ by intersecting the line containing $C$ and $W$ with the image plane. The line through $C$ perpendicular to the image plane is called the *optical axis* and its intersection is called the *principal point*. The distance, $f$, between $C$ and image plane is the $focal\ length$. Three coordinate systems are defined: a 2D image coordinate system, a 3D camera and object coordinate system. The origin of the image coordinate system is the top left corner of the image plane, while the axis $\mathbf{u}$ and $\mathbf{v}$ are along image row and column respectively. The origin of the camera coordinate system is located at $C$. $\mathbf{Z_c}$ axis coincides with the optical axis, while the $\mathbf{X_c}$ and $\mathbf{Y_c}$ are parallel with $\mathbf{u}$ and $\mathbf{v}$ axis respectively. In this part, the object coordinate system is assumed to be the same as camera coordinate system. Both coordinate system are defined as right handed.

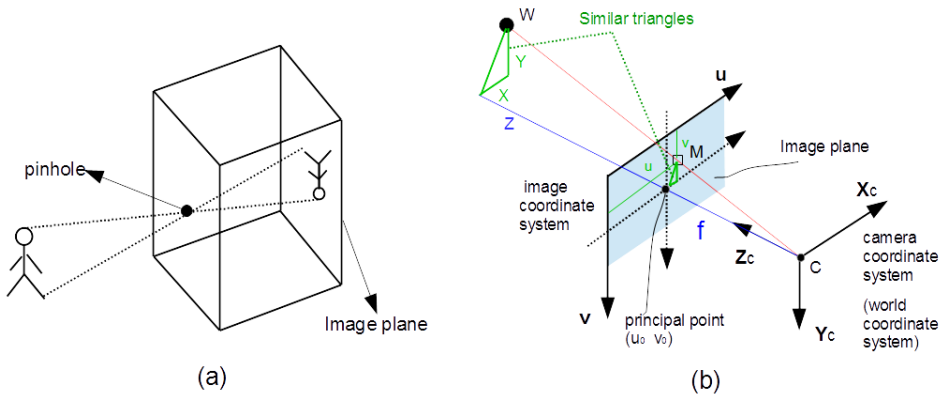



Figure 2.3: (a) The demonstration of pinhole camera model. (b) The geometry of perspective projection to project 3D point in the camera coordinate system to 2D point in image coordinate system. The two similar triangles defines the relations between coordinate of the 3D point and the corresponding 2D image point.

Let $\mathbf{w} = [X\ Y\ Z]^T$ be the coordinates of $W$ in the object coordinate system and $\mathbf{m} = [u\ v]^T$ be the coordinate of $M$ in the image coordinate system (pixels). Let $[u_0\ v_0]^T$ be the coordinates of the principal point in the image coordinate system. The geometry of perspective projection gives out two right triangles, i.e. in the two green triangles in Figure 2.3b. The two right triangles are similar triangles that is corresponding sides have the same length ratio. The ratio equals the ratio between focal length $f$ and $Z$ coordinate of $\mathbf{w}$. Therefore, the projection of $W$ to $M$ is presented by two equations: $\frac{u-u_0}{X} = \frac{f}{Z}$ and $\frac{v-v_0}{Y} = \frac{f}{Z}$. The perspective projection is given by:

$$\begin{cases} u = u_0 + \frac{Xf}{Z} \\ v = v_0 + \frac{Yf}{Z}. \end{cases} \tag{2.1}$$

Homogeneous coordinates add an other dimension, called projective dimension, to the original coordinates. The homogeneous coordinates of $[u\ v]$ becomes $[u\ v\ w]$. The projecitve dimension is the distance from the projection center to the image plane and the homogeneous coordinates can represent that the points whose value are proportional are equivalent. As shown in Figure 2.4, by moving the original image plane away from the projection center, the coordinates of image pixel $M^{'}$ in the new plane is proportional to coordinates of image pixel in the original plane with a ratio between the distance from the projection center to the two image planes. The homogeneous coordinates of pixel on the original image is often defined as $[u\ v\ 1]$ which represents the homogeneous coordinates of image point M. The homogeneous coordinates of the equivalent pixel in another image plane,$M^{'}$, can be represented as $[wu\ wv\ w]$. Equation 2.1 then can be used as a matrix multiplication using the homogeneous coordinates, which makes the equation a linear transformation.

The equation becomes:

$$\widetilde{\mathbf{m}} = \mathbf{A}\mathbf{w}, \tag{2.2}$$

where matrix $\mathbf{A}$ depends on $f$ and $[u_0\ v_0]$ which are often regarded as the camera intrinsic parameters. Matrix $\mathbf{A}$ has the following form:

$$\mathbf{A} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

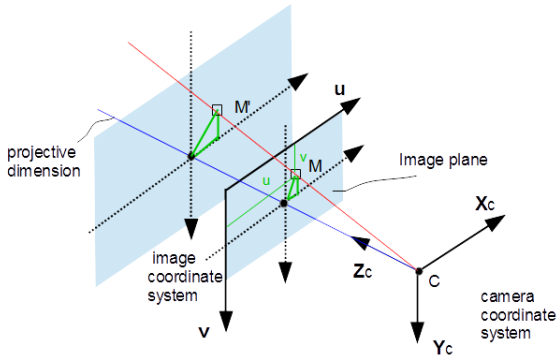



Figure 2.4: The projective dimension defines the distance of the projection center to the image plane. The coordinates of image pixel $M$ and $M^{'}$ are proportional and can be represented equivalently in homogeneous coordinates.

Matrix $\mathbf{A}$ can also include more intrinsic parameters, such as camera lens radial and tangential distortion, or a skew coefficient between two axes. All these intrinsic parameters, including $f$, $[u_0\ v_0]$ and distortion and skew parameters, can be estimated accurately by calibration (Zhang, 2000). In airborne camera lenses, the distortion and skew effects are often very small. Therefore, we do not add these parameters in the $\mathbf{A}$ matrix.

However, if the camera is amounted on an airplane, the camera coordinate system is different. A new camera coordinate system is defined with its $\mathbf{Z}_c$ axis pointing away from the image plane as shown in 2.5. In order to keep the coordinate system to be right handed, the $\mathbf{X}_c$ axis points to the same direction as $\mathbf{u}$ axis, but the $\mathbf{Y}_c$ axis points to the opposite direction as $\mathbf{v}$. Therefore, *boldA* matrix is defined differently. The projection of $W$ to $M$ becomes: $\frac{u-u_0}{X} = \frac{-f}{Z}$ and $\frac{v-v_0}{Y} = -(\frac{-f}{Z})$. Therefore, the $\mathbf{A}$ matrix becomes:

$$\mathbf{A} = \begin{bmatrix} -f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.3}$$

**Matrix of extrinsic parameters**

The object coordinate system is often not identical to the camera coordinate system. The transformation between the object and camera coordinate systems should be considered. As shown in Figure 2.5, both rotation and translation are required to perform a rigid transform from the object coordinate system to the camera coordinate system. Here, $\mathbf{R}$ is the $3 \times 3$ rotation matrix and $\mathbf{T}$ is the $3 \times 1$ translation vector. Let $\widetilde{\mathbf{w}} = [X\ Y\ Z\ 1]^T$ be the homogeneous coordinates of W in a object coordinate system. Combining with Equation 2.2, the perspective projection equation is formed as follows:

$$\underset{3\times1}{\widetilde{\mathbf{m}}} = \underset{3\times3}{\mathbf{A}}\underset{3\times4}{[\mathbf{R} \mid \mathbf{T}]}\underset{4\times1}{\widetilde{\mathbf{w}}}. \tag{2.4}$$

Recall that both homogeneous coordinates of 3D point $W$ and 2D image pixel $M$, represented as $\widetilde{\mathbf{w}}$ and $\widetilde{\mathbf{m}}$ are used in the equation. The $[\mathbf{R}|\mathbf{T}]$ is a matrix of concatenation between the rotation matrix and the translation vector. This is the widely used perspective projection in form of matrix multiplication in computer vision.

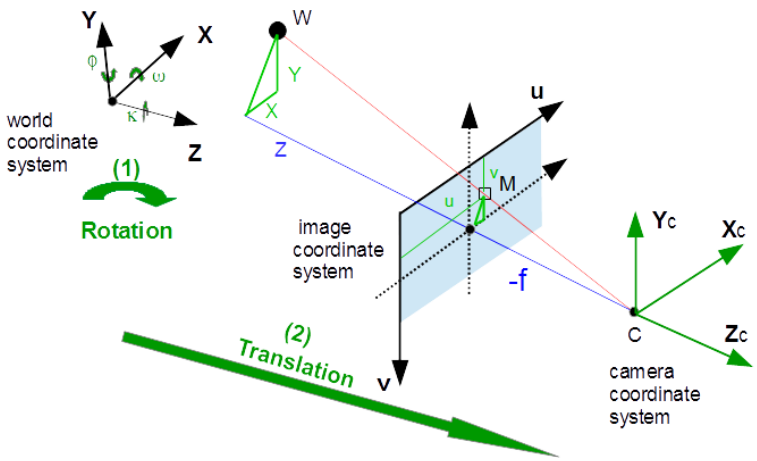



Figure 2.5: A new camera coordinate system is defined for the airborne camera system as shown in green axes. The coordinate system is defined as a right-handed coordinate system with its **Zc** axis pointing away from the image plane. The transformation of object coordinate system to camera coordinate system by rotation and followed by translation. However, the object coordinate system is not consistent with the camera coordinate system.

The extrinsic parameters of airborne cameras notably consists of camera orientations and position in a object coordinate system. Let the angle of camera orientation in X, Y and Z axes be $\omega$, $\phi$ and $\kappa$ respectively. Coordinate system rotations of the X, Y and Z axes give three matrices: $\mathbf{R_x}$, $\mathbf{R_y}$ and $\mathbf{R_z}$. The rotation matrix $\mathbf{R}$ is given as a composition of rotations about three axes, which is presented by matrices multiplication as $\mathbf{R_z R_y R_x}$. Then the rotation matrix

$\mathbf{R}$ derived from extrinsic orientations can rotate the object coordinate system to align the axes with the axes in the camera coordinate system.

Let the object coordinates of the camera position be $\mathbf{c} = [X_0\ Y_0\ Z_0]^T$. As $[\mathbf{R} \mid \mathbf{T}]$ in Equation 2.4 presents a transformation by a rotation followed by a translation, the object coordinate system is rotated first. The translation vector $\mathbf{T}$ is not simple $-\mathbf{c}$. Instead, the translation vector is formed as $\mathbf{T} = -\mathbf{Rc}$. The matrix of camera extrinsic parameters is formed as $[\mathbf{R} \mid -\mathbf{Rc}]$.

**Perspective projection matrix & collinearity equation**

Finally, the perspective projection with a matrix multiplication form, using intrinsic and extrinsic parameters from airborne cameras is defined as:

$$\widetilde{\mathbf{m}} = \mathbf{A}[\mathbf{R} \mid -\mathbf{Rc}]\widetilde{\mathbf{w}}; \qquad \widetilde{\mathbf{P}} = \mathbf{A}[\mathbf{R} \mid -\mathbf{Rc}], \tag{2.5}$$

where $\widetilde{\mathbf{P}}$ represents the *perspective projection matrix*. The equation can be simplified as follows:

$$\widetilde{\mathbf{m}} = \mathbf{A}[\mathbf{R} \mid -\mathbf{Rc}]\widetilde{\mathbf{w}} = \mathbf{AR}\ (\mathbf{w} - \mathbf{c}). \tag{2.6}$$

The collinearity equation is equivalent form in photogrammetry which also uses airborne camera parameters. The collinearity can be obtained by expanding Equation 2.6. However, in photogrammetry, the image coordinate system is also defined differently as shown in Figure 2.6b. The original of coordinate system is located in the right bottom corner of the image plane. Therefore, the $\mathbf{A}$ is redefined as:

$$\mathbf{A} = \begin{bmatrix} -f & 0 & u_0 \\ 0 & -f & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The equation is further expanded as :

$$[u\ v\ 1]^T = \begin{bmatrix} -f & 0 & u_0 \\ 0 & -f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{21} & R_{31} \\ R_{12} & R_{22} & R_{32} \\ R_{13} & R_{23} & R_{33} \end{bmatrix} [X - X_0\ Y - Y_0\ Z - Z_0]^T. \tag{2.7}$$

$$\begin{cases} u - u_0 = -f\frac{R_{11}(X-X_0)+R_{21}(Y-Y_0)+R_{31}(Z-Z_0)}{R_{13}(X-X_0)+R_{23}(Y-Y_0)+R_{33}(Z-Z_0)} \\ \\ v - v_0 = -f\frac{R_{12}(X-X_0)+R_{22}(Y-Y_0)+R_{32}(Z-Z_0)}{R_{13}(X-X_0)+R_{23}(Y-Y_0)+R_{33}(Z-Z_0)}, \end{cases} \tag{2.8}$$

where the rotation matrix $\mathbf{R}$ is presented as:

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{21} & R_{31} \\ R_{12} & R_{22} & R_{32} \\ R_{13} & R_{23} & R_{33} \end{bmatrix}.$$

Equation 2.8 is the collinearity equation. However, in practice, the image coordinate system is often defined as shown in Figure 2.5. In addition, using matrix multiplication form of perspective projection is more computationally efficient. Therefore, Equation 2.5 and 2.6 with $\mathbf{A}$ matrix described in Equation 2.3 are the core of the research. The coordinate systems defined in Figure 2.5a are used for the rest of paper.

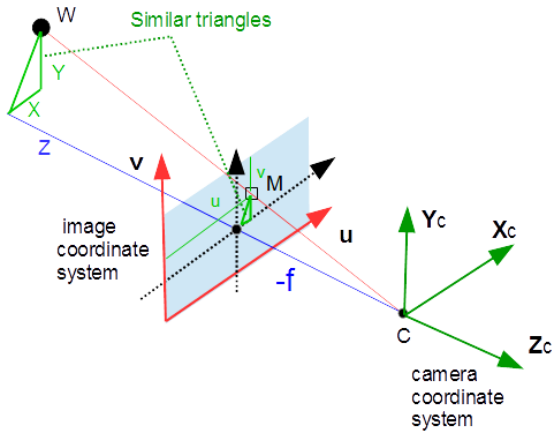



Figure 2.6: A new image coordinate system used by collinearity equation is shown in red axes. However, in practice, the image coordinate system in Figure 2.5 is used.

### 2.3.2. Ray tracing

Another way to transform 3D points to pixels in a image is ray tracing. Instead of using perspective projection to project 3D points to an image, ray tracing is to shot a optical ray from each image pixel to intersect a 3D surface. The intersection point is the 3D point the image pixel looks at. Still optical ray equation relies on perspective projection.

In perspective projection, many 3D points, such as $W$ and $W^{'}$, with different distance to the projection center $C$, are projected to the same image point as shown in Figure 2.7. All these points form a optical ray. The geometry of perspective projection results in two similar triangles, i.e. in the two green triangles in Figure 2.7. Therefore, the coordinates of $W^{'}$ is a scale of $W$ in the camera coordinate system. A scale factor,$s$, is added to the Equation 2.5 to present all the possible 3D points projected to the same image point as follows:

$$s\ \widetilde{\mathbf{m}} = \mathbf{A}[\mathbf{R} \mid -\mathbf{Rc}]\widetilde{\mathbf{w}}. \tag{2.9}$$

According to Equation 2.6, the equation can be simplified to: $s\ \widetilde{\mathbf{m}} = \mathbf{AR}\ (\mathbf{w} - \mathbf{c})$. Finally, the optical ray shot from $C$ intersecting with image point $M$ is presented as:

$$\mathbf{w} = \mathbf{c} + s\ (\mathbf{AR})^{-1}\widetilde{\mathbf{m}}. \tag{2.10}$$

### 2.3.3. Stereo rectification

After performing transformation of a 3D point to a stereo pair, the position of the corresponding pixels in a stereo pair are found. This section explains the fundamentals of transforming the positions of corresponding pixels to a single value

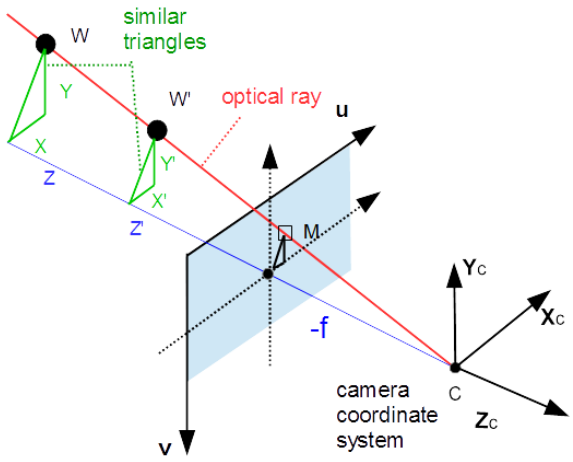



Figure 2.7: The optical ray, shown as the red line, containing projection center $C$ and image point $M$, can intersect many 3D points, such as $W$ and $W^{'}$. In the camera coordinate system, the coordinates of the point $W^{'}$ is a scale of $W$.

called disparity. The section will start from the epipolar geometry between two images, which constrains the search for the corresponding pixels in to an epipolar line. Then stereo rectification is described to align epipolar lines to image rows or columns. A stereo rectification algorithm using camera intrinsic and extrinsic parameters is elaborated to perform robust and accurate stereo rectification. This section is mainly based on the book of Kaehler and Bradski (2016) and the paper of Fusiello et al. (2000).

**Epipolar geometry and stereo rectification**

Epipolar geometry between stereo images is to set constraints for finding corresponding pixels, e.g. the image pixels $M_1$ and $M_2$ in the left and right images respectively as shown in Figure 2.8a. Instead of searching for corresponding pixels in the whole image, incorporating the epipolar constraint reduces the search space to 1D. As shown in Figure 2.8a, $M_1$ may correspond to many 3D points at different depths, while the projection of these 3D points to the right image forms a line, which is called the epipolar line. Therefore, the problem of finding the pixel corresponding to $M_1$ can be restricted to a search among the pixels on the epipolar line. Epipolar geometry has some more characteristics. The projection center of two cameras $C_1$ and $C_2$ and a 3D point $W$ defines a epipolar plane. All epipolar lines go through the corresponding epipoles as shown in Figure 2.8a. The epipoles $e_1$ and $e_2$ are defined by the intersections between line, containing $C_1$ and $C_2$, and the left and right image planes respectively.

Epipolar lines are in general not aligned along row (or column) in given images. As a consequence, pixels through the epipolar line should be interpolated every time for each image pixel when searching for corresponding pixel. As shown

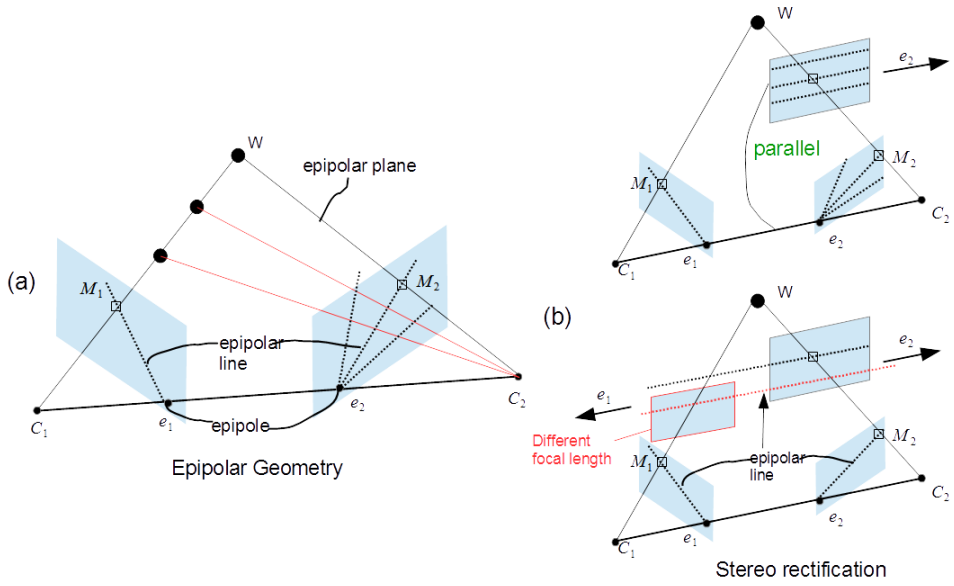



Figure 2.8: (a) Epipolar geometry of stereo images. The epipolar line containing $M_2$ in the right image defines a 1D domain that search for $M_1$ in the left image. All epipolar lines in the left and right image pass through the epipoles $e_1$ and $e_2$ respectively. The epipolar plane is formed by connecting the two camera projection center of two cameras $C_1$ and $C_2$ and a 3D point $W$. (b) Demonstration of the stereo rectification. Top: when the right image plane is parallel to the baseline $C_1C_2$, its epipole is at infinity such that all the epipolar lines become parallel. Bottom: when the left image plane is also parallel with baseline, but with a different focal length, the epipolar lines between two images are parallel, but not aligned. The final stereo rectification with the epipolar lines aligned in the image row or column is shown in Figure 2.9a.

in Figure 2.8b top, if the right image plane is parallel to the base line baseline $C_1C_2$, its epipole is at infinity such that all the epipolar lines are parallel. The same applied for the left image. However, if the left image plane has a different focal length from the right image plane as shown in Figure 2.8b bottom, their epipolar lines between two images are parallel, but they are not aligned. As long as they have the same focal length such that the left and right image planes are coplanar, the epipolar lines are aligned to the same row (or column) as shown in Figure 2.9a. This process is called stereo rectification. The advantage of stereo rectification is that only one time image interpolation on left and right images is needed respectively. Finding the corresponding pixel of $M_{n1}$ in the left rectified image is simplified to search among the pixels in the same row(or column) in the right image. The difference between row or column values of corresponding pixels is called disparity as shown in Figure 2.9a, which needs to be estimated in dense image matching algorithms.

In this research, estimating epipolar lines for each pixel is not required, so the details of deriving the epipolar lines for searching corresponding pixels will not

discussed but can be found in (Kaehler and Bradski, 2016). In order to perform stereo rectification, the two original cameras should rotate around their projection center until the two image planes parallel to the baseline $C_1C_2$ and coplanar. This ensures the epipoles are at infinity as shown in Figure 2.9. The camera matrices with the new extrinsic and intrinsic parameters should be estimated. With the new camera matrices, the transformation matrices to map the original images to rectified images can be also obtained. Both the rectification of camera matrices and images are elaborated below.

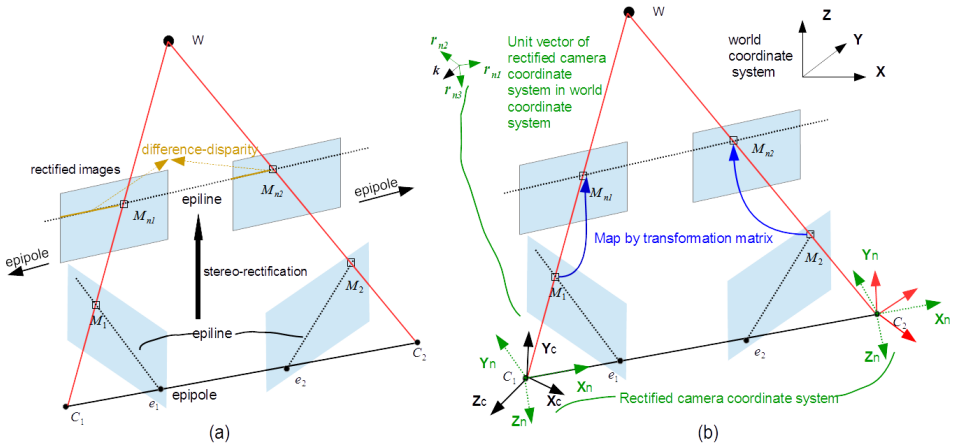



Figure 2.9: (a) The process of stereo rectification. Stereo rectification rotates each camera around its projection center $C_1$ and $C_2$ respectively to transform two image planes to a co-plane and align the epipolar lines to the same row or column. The coplane is parallel to the base line $C_1C_2$ and epipoles are at infinity. The disparity of corresponding pixels is shown in brown. (b) Rectification of camera matrices and images. The two new camera coordinate systems only differ from the projection centers. The intrinsic parameters and projection center of the two new cameras is kept unchanged. Only a new rotation matrix indicating the transformation of object coordinate system to the new camera coordinate system needs to be defined as shown in the three unit vectors. After the rectification of camera matrices, the transformation matrix maps $M_1$ to $M_{n1}$ and $M_2$ to $M_{n2}$ respectively, as shown in blue.

### Rectification of camera matrices

If the intrinsic and extrinsic parameters of two cameras are unknown, the stereo rectification is done using corresponding image points (Fusiello and Irsara, 2011), which are often extracted by SIFT. However, accurate rectification is not guaranteed and relied on the accuracy of the correspondences. If the cameras are calibrated with ground control points and accurate intrinsic and extrinsic parameters are provided, stereo rectification can be performed robustly and accurately. Let $\widetilde{\mathbf{P}}_1$ and $\widetilde{\mathbf{P}}_2$ be the perspective projection matrices of the two original cameras which has been defined in Equation 2.5:

$$\widetilde{\mathbf{P}}_1 = \mathbf{A}_1[\mathbf{R}_1| \ -\mathbf{R}_1\mathbf{c}_1]; \qquad \widetilde{\mathbf{P}}_2 = \mathbf{A}_2[\mathbf{R}_2| \ -\mathbf{R}_2\mathbf{c}_2]. \tag{2.11}$$

The matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ contains the intrinsic parameters of the two cameras. $\mathbf{R}_1$ and $\mathbf{R}_2$ are formed by extrinsic orientations of the two cameras. Vectors $\mathbf{c}_1$ and $\mathbf{c}_2$ present the coordinates of the original camera positions in a object coordinate system.

Stereo rectification is to find two new perspective projection matrices, $\widetilde{\mathbf{P}}_{n1}$ and $\widetilde{\mathbf{P}}_{n2}$ to align the corresponding pixels in the same rows or columns. The new extrinsic and intrinsic parameters are defined as follows: (1) the new matrix of intrinsic parameters $A_n$ can be defined arbitrarily but should be the same to ensure the epipolar line of two images aligned in the row or columns as described above; when all the airborne images are acquired from the same project, all the images have the same intrinsic parameters which are kept same for the new camera matrices; (2) the $\mathbf{c}_1$ and $\mathbf{c}_2$ is also kept unchanged, as the two original cameras rotates around their projection center; (3) only the new camera orientations or rotation matrices $\mathbf{R}_n$ should be derived such that the new camera coordinate system is parallel to the baseline. As shown in Figure 2.9b, the two new cameras have the same rotation matrix and the two coordinate systems only differ from the projection center.

Therefore, the two new perspective projection matrices are written according to the Equation 2.5:

$$\widetilde{\mathbf{P}}_{n1} = \mathbf{A_n}[\mathbf{R}_n| \ -\mathbf{R}_n\mathbf{c}_1]; \qquad \widetilde{\mathbf{P}}_{n2} = \mathbf{A_n}[\mathbf{R}_n| \ -\mathbf{R}_n\mathbf{c}_2]. \tag{2.12}$$

The row vectors $\mathbf{r}_{n1}$, $\mathbf{r}_{n2}$ and $\mathbf{r}_{n3}$ of $\mathbf{R}_n$ represent the unit vector of the camera coordinate system axes in the object coordinate system. The object coordinates of the unit vectors of X, Y and Z axes as shown in Figure 2.9b, are formed in three steps:

1) the X axis is parallel to the baseline: $\mathbf{r}_{n1} = (\mathbf{c}_1 - \mathbf{c}_2)/||\mathbf{c}_1 - \mathbf{c}_2||$. This also ensures that the epipolar lines of rectified images aligned horizontally.

2) the Y axis is orthogonal to X and to $\mathbf{k}$: $\mathbf{r}_{n2} = \mathbf{k} \wedge \mathbf{r}_{n1}$. Here $\mathbf{k}$ is an arbitrary unit vector. The unit vector of the axis $\mathbf{Z}_c$ of the original left camera coordinate system is often chosen as $\mathbf{k}$.

3) the Z axis is orthogonal to XY: $\mathbf{r}_{n3} = \mathbf{r}_{n1} \wedge \mathbf{r}_{n2}$.

Overall, new perspective projection matrices ensures that the epipolar lines between two new images aligned along the image rows. Combining the old and new perspective projection matrices, the final rectified images can be derived below.

**Rectification of image**

When the original and new camera matrices ear obtained, transformation matrices $\mathbf{t}_1$ and $\mathbf{t}_2$ should be computed to map pixels from the orginal images to the rectified images respectively as shown in blue in Figure 2.9b. When the perspective projection matrices of original and rectified camera are obtained as described

above, projecting the same 3D points $\mathbf{w}$ to the original and new left image planes using Equation 2.6 is derived as follows:

$$\begin{cases} \widetilde{\mathbf{m}}_1 = \mathbf{A}_1[\mathbf{R}_1 \mid -\mathbf{R}_1\mathbf{c}_1]\widetilde{\mathbf{w}} = \mathbf{A}_1\mathbf{R}\ (\mathbf{w} - \mathbf{c}_1) \\ \widetilde{\mathbf{m}}_n = \mathbf{A}_n[\mathbf{R}_{n1} \mid -\mathbf{R}_{n1}\mathbf{c}_1]\widetilde{\mathbf{w}} = \mathbf{A}_n\mathbf{R}_{n1}\ (\mathbf{w} - \mathbf{c}_1). \end{cases} \tag{2.13}$$

The equation is further expanded as:

$$\mathbf{c}_1 + (\mathbf{A}_n\mathbf{R}_n)^{-1}\widetilde{\mathbf{m}}_{n1} = \mathbf{c}_1 + (\mathbf{A}_1\mathbf{R}_1)^{-1}\widetilde{\mathbf{m}}_1; \qquad \widetilde{\mathbf{m}}_{n1} = \mathbf{A}_n\mathbf{R}_n\mathbf{R}_1^{-1}\mathbf{A}_1^{-1}\widetilde{\mathbf{m}}_1. \tag{2.14}$$

The transformation matrix, $\mathbf{t}_1 = \mathbf{A}_n\mathbf{R}_n\mathbf{R}_1^{-1}\mathbf{A}_1^{-1}$, is applied to the left image to create rectified image, similarly, $\mathbf{t}_2 = \mathbf{A}_n\mathbf{R}_n\mathbf{R}_2^{-1}\mathbf{A}_2^{-1}$, is applied to the right image to create rectified image. As the map of the original to rectified image is not one-to-one, a billinear interpolation is applied to create the rectified images (Kaehler and Bradski, 2016).

## 2.4. Conclusions

In this chapter, according to the characteristics of 3D information provided by airborne laser scanning and camera imagery, combining complementary information between two data source can help to address their quality problems. Using fundamental photogrammetric theory brings LiDAR data to image domain for combining their information is better than bringing image data to LiDAR domain.

# 3

# Shadow detection in a single image

*Shadow is the only geometric indicator in a single image and relatively easy to detect compared to other objects, such as building, road and car. However, shadows are material and environment dependent in different urban VHR images. Supervised learning can learn characteristics of shadow adaptively in different images when training samples selected from these images are available. Unfortunately, manual labeling of images is expensive. 3D models generated from existing LiDAR data can be integrated by reconstructing shadows to provide training samples automatically. However, accurate shadow reconstruction for large 3D models is time-consuming. In addition, due to modelling errors and different acquisition time between LiDAR and image data, many training samples selected from the reconstructed shadow image are mislabeled. We propose a ray-tracing approach with an effective KD tree construction to feasibly reconstruct accurate shadows for a large 3D model. The experiments using different size of models for ray tracing show that the ray tracing with the KD tree is efficient and applicable to an even larger model than our experiments. Next, a comparative study considering four classification methods, quadratic discriminant analysis (QDA) fusion, support vector machine (SVM), K nearest neighbors (KNN) and Random forest (RF), is performed to select the best classification method with respect to capturing the complicated properties of shadows and robustness to mislabeling. Both results of experiments in Amersfoort and Toronto data shows that RF performs the best in terms of the two aspects above. Therefore, RF with training samples provided by LiDAR data is an automatic, effective, and robust approach for shadow detection from VHR images.*

## 3.1. Introduction

Automatic 3D city model reconstruction has been intensively investigated in recent decades. Airborne LiDAR data with high vertical accuracy are widely used for 3D city modelling, especially for buildings. However, LiDAR data are often infrequent, as the airborne LiDAR system and acquisition cost are often expensive. As urban area often change dramatically, airborne VHR images often updated annually with a cheap expense are often used for change detection to keep 3D models up-to-date. To detect changes using images, shadows should be considered. Shadow is the only geometric indicator in a single image, which often requires simpler processing chain

This chapter has been published in the Remote Sensing, 2019 (Zhou et al., 2019b)

than extracting 3D information from stereo images. In addition, the color property of shadow is relatively less complicated than other objects, such as building, car and road, which are represented by various textures in the images. Therefore, this chapter concentrates on detecting shadows from VHR images in urban areas.

Literature presents three main categories of shadow detection (Adeline et al., 2013; Lorenzi et al., 2012): property-based, supervised learning and model-based.

Property-based methods do not need any prior knowledge and are often combined with automatic thresholding. They focus on exploring spectral properties to identify shadows. Shadow properties have been studied intensively (Tsai, 2006; Chung et al., 2009; Adeline et al., 2013) and are summarized into four properties: (1) shadows have low radiance due to obstruction of direct sunlight;(2) radiance received from shadow areas decreases from short (blue-violet) to long (red) wavelength due to scattering effects; (3) in urban canyons, reflection effects of surroundings cannot be ignored in the shadow area; (4) radiation received from shadow areas is material-dependent.

The first two properties explain why property-based methods work reasonably well in many studies. According to property 1, shadows tend to have low RGB or intensity values. Referring to property 2, in RGB images, shadows should have higher blue than their counterpart in the sun, while non-shadows receive direct light with more red and green radiation. Tsai (2006) discusses the ratio of hue over intensity to combine these two properties to enlarge the difference between shadow and non-shadow. With high hue (more blue) and low intensity values, shadows are expected to have much higher ratio values than non-shadows. The $HSI$ color space is tested with best performance in Tsai's work and the ratio is defined as $\frac{H+1}{I+1}$, where H and I present hue and intensity value in HSI color space. Finally, a thresholding method (Otsu, 1975) is used to separate shadows from non-shadows. However, the other two properties which affect the efficiency of the first two properties are not considered. According to property 3, if the material of a surrounding object is highly reflective, shadows may receive high reflection which can be confused with dark objects. Referring to property 4, the red value of reddish objects under shadow is largely reduced, but they may have very different hue values depending on the red value or blue value is a little higher. If red is a bit higher than blue, the hue is much smaller than the opposite situation. Due to the high sensitivity of hue, the effectiveness of property 2 is adversely affected.

Supervised learning is expected to have better performance in shadow detection in images by learning the characteristics of shadows in complex scenes. Many researches have applied supervised learning on airborne VHR images and natural images in computer vision. The shadows in these images share the same characteristics of all four properties described. A support vector machine (SVM) performs well for natural RGB images (Arbel and Hel-Or, 2011). Apart from RGB features, the texture features from texton histograms are used with SVM for classifying natural images (Guo et al., 2013). In VHR images, Lorenzi et al. (Lorenzi et al., 2012) extract texture features from a wavelet transform for SVM to detect shadow. As textures in shadows are not very informative, RGB features are more often used for shadow detection. However, supervised approaches require lots of man-

ual work to generate good training examples with large varieties. In Xiao et al. (2013), a limited amount of training samples are selected by manually drawing strokes. Two Gaussian mixture models (GMM) were built for shadow and non-shadow regions respectively selected from strokes. Mean-shift clustering is applied to segment the whole image. Finally, the clusters are classified based on which GMM they are similar to. The goal of clustering is to remove small dissimilarities from pixels which is difficult to be captured by the limited training samples. However, the result is strongly dependent on cluster results and the number of Gaussian components chosen. An advanced closed-form solution is provided in Levin et al. (2008) to reduce the large amount of user input to detect shadows from regular images by image matting. The idea is to use a local smoothness constraint to propagate the characteristics of shadows (foreground) and non-shadow (background) from user inputs to other unknown regions. This approach works well for different environments with limited manual samples, but manual works are still required. As shadows are strongly dependent on the environment, training samples extracted from one image may have a very bad generalization capability to apply on the images acquired from different areas or different time. This implies a large amount of work to select training samples for the images from which we want to detect shadows.

Model-based methods use an existing watertight 3D model or DSM to reconstruct shadows with camera parameters and sun position (Tolt et al., 2011; Gorte and van der Sande, 2014). Misalignment often occurs between LiDAR and image data as they are obtained from different resources and geo-referenced independently. Shadows reconstructed are often wrong near shadow boundaries. In Tolt et al. (2011), the interior of large shadows or non-shadows in the reconstructed images is automatically selected as training examples by an erosion morphological filter. A SVM classifier is then applied to improve shadow detection. More importantly, the method did not consider the modelling error of 3D models if they are automatically reconstructed from LiDAR or image data, and different acquisition time between models and images. Many samples selected from reconstructed shadow images are mislabeled. Wang et al. (2017a) adopted image matting (Levin et al., 2008) using skeletons of (non-)shadow regions as input to detect shadows from mislabels caused by moving cars, trees, and water. These method does not assess the number of mislabels and robustness of the methods to mislabeling.

A property-based approach that only considers property 1 and 2 cannot capture these complicated shadow properties, but supervised learning approaches which are data driven are expected to find reasonably good boundaries to separate different classes. However, the generalization ability of applying the classifier trained in one place to classify shadows in another place is affected by illumination conditions, environment, and object material. Therefore, training samples should be selected from each image, which is labor intensive. Model-based methods can provide large and various training samples from the image automatically which needs to be classified. However, the training samples indicated from the reconstructed shadow images are not all labeled correctly. If the samples are used for training classifiers, the effect of these mislabeled training samples is called mislabeling.

In this chapter, we provide an automatic, effective, and robust approach of shadow detection in VHR images by integrating LiDAR data for shadow reconstruction to provide training samples automatically. The contributions of this work are as follows. (1) We propose a ray tracing approach with an efficient KD tree construction for accurate shadow reconstruction from 3D models for a large scene. (2) We conduct a comparative study on how machine learning methods are affected by mislabeling effects introduced by automatically generated training samples and their ability to detect complicated shadows.

## 3.2. Study area and data preparation

The study areas are in Amersfoort, the Netherlands and Toronto, Canada. The Toronto dataset is from an ISPRS benchmark test. The data required for the two study areas are the same: an existing 3D city model and a VHR image.

### 3.2.1. 3D model generation

A 3D triangular model of Amersfoort is created automatically by combining a topographic map and LiDAR point clouds acquired around the same time using software provided by the University of Twente (Oude Elberink et al., 2013). Four object classes are aggregated from classes defined in BGT, the Dutch large-scale topographic maps and included in the 3D model (Oude Elberink et al., 2013): water, road, terrain and building. These objects together form a seamless terrain which is important for reconstructing shadows on the ground. The models are triangulated based on simplification rules specific for different classes. For example, roads are regarded as planes and can be constructed by triangulating the polygonal vertices from the map with heights determined by a local plane fitting all LiDAR points inside the polygon. Building polygons from the map are used for a constraint triangulation for buildings, and their roofs are simplified from a triangulated mesh by considering the planarity of adjacent triangles. Details can be found in Oude Elberink et al. (2013) and the 3D model is displayed in the left image of Figure 3.1a. Another 3D model of Toronto is created by extruding manually created reference 3D roof planes. In this chapter, they are used for constructing 3D buildings. on the other hand, ground points are first segmented from LiDAR point clouds using a progressive morphological filter (Zhang et al., 2003). As no topographic map is available for simplifying ground triangles, a voxel-grid filter which replaces points in a voxel with the voxel centroid is used to simplify ground points for triangulation. The 3D buildings are obtained by extruding roof planes to the ground using FME software (Soetaert et al., 2010), Safe Software Inc. As shown in the right image of Figure 3.1a, this 3D city model has only two classes: terrain and building.

There are several reasons for not considering trees in two models: (1) trees are difficult to accurately model from airborne LiDAR. Tree points are often sparse, and many points penetrate into leaves and even reach to trunks and ground. (2) if the image has an acquisition time different from LiDAR, the status of trees is different. The numbers of triangles in the models of Amersfoort and Toronto are

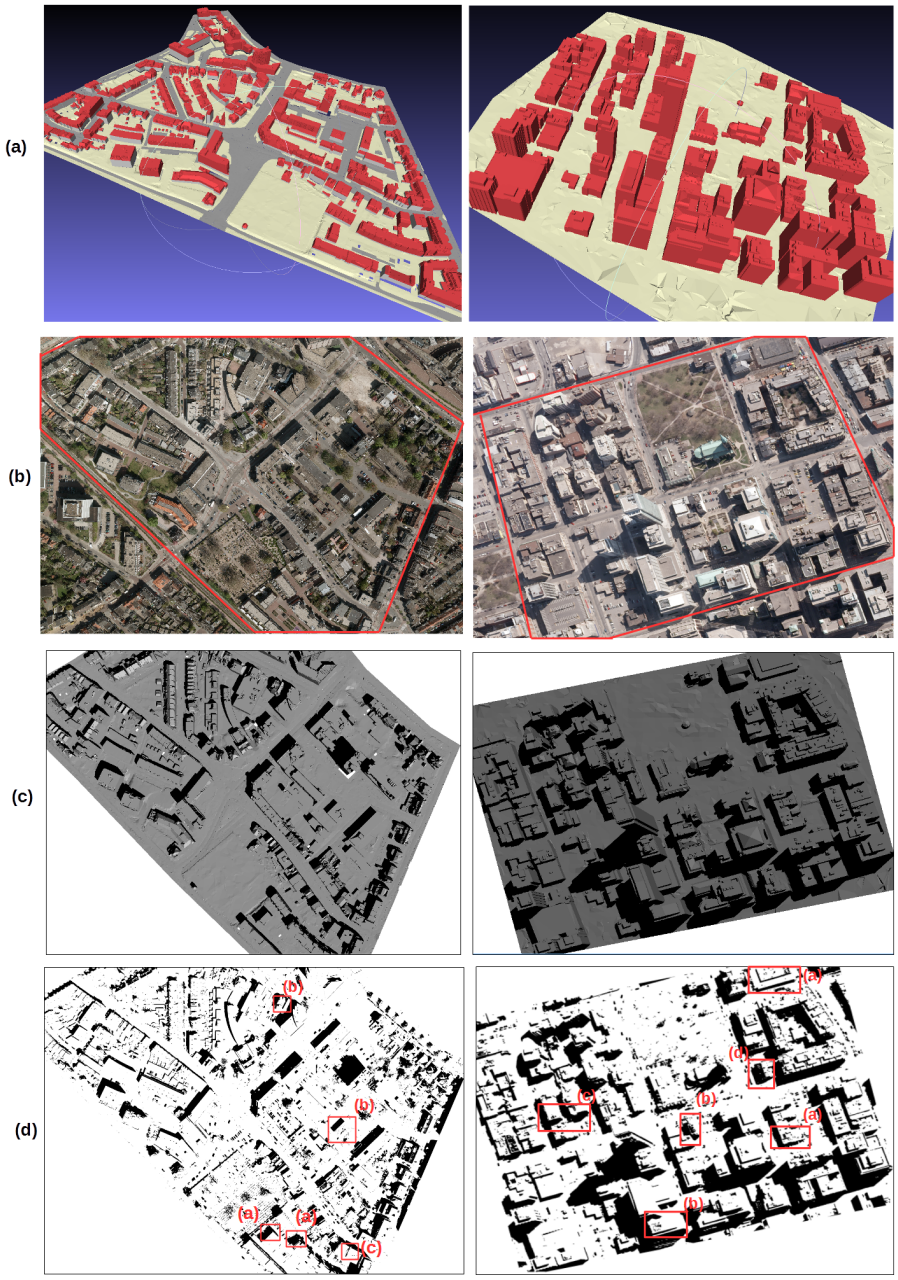



Figure 3.1: (**a**) The 3D city model for Amersfoort and Toronto. (**b**) The VHR images corresponding to the 3D city models. The area of interest is in the red box of two images. The green boxes indicate the area for creating test dataset manually. (**c**) The reconstructed shadow images from 3D models by ray tracing. (**d**) The shadow detected from VHR images by RF with automatically generated training samples. The red boxes indicate the examples used in Section 3.4.2 to compare the results from four classification methods.

429,904 and 30,770, respectively. The areas covered by each model are around 0.2 $km^2$. As the automatically generated buildings in Amersfoort are not represented by optimal triangles such as those manually created for Toronto, the Amersfoort model has more triangles than the Toronto model. The reason of using Toronto dataset is that shadow reconstructed from the 3D city model with accurate buildings and sparse and limited trees are accurate. Mislabels can be simulated in order to evaluate the robustness of machine learning methods.

### 3.2.2. VHR image description

The VHR aerial images from Amersfoort and Toronto are taken by a Microsoft Vexcel UltraCam-Xp with 3.5 cm spatial resolution and an UltraCam-D camera with 15 cm spatial resolution, respectively. The data channels of Amersfoort and Toronto are RGB. The image size per frame is $11310 \times 17310$ and $7500 \times 11500$ for Amersfoort and Toronto, respectively. On all the images, a bundle adjustment is performed by ground control points to provide a file with camera parameters. The back-projection error of using these parameters is less than 1 pixel. The 3D model for Amersfoort is created from LiDAR data and maps acquired in 2008, while the images were acquired in April 2010. There were many building constructions during these two years. No explicit acquisition time of the images from Toronto is known. However, by visually checking, very limited changes of buildings are found between model and images. The trees are thin and sparse and with the accurate reference roofs, the reconstructed shadows match the shadows in the image very well. The VHR images corresponding to the 3D models are shown in red polygons of Figure 3.1b.

The two datasets are chosen for two reasons. (1) The shadows properties in both areas are complicated. In Amersfoort, the reflectance of shadows casted on many reddish roads are material dependent. Apart from material problems, the Toronto area has many high-rise buildings. Shadows receive light reflected from glass walls in the buildings, resulting in shadows with a high reflectance. This requires a method that is capable of robustly capturing these complicated properties to best separate shadows and non-shadows in different environments. (2) In Amersfoort, due to time differences and because tree models are not incorporate, many mislabeled samples are included. The Toronto model matches the image very well, so the training samples are in general correctly labeled. The comparison of shadow detection on these two datasets can evaluate the robustness of methods on mislabeling effects.

## 3.3. Shadow detection

In this chapter, three steps are elaborated: (1) shadow reconstruction for creating training samples, (2) adaptive erosion filtering to reduce mislabels, (3) a comparative study of four different supervised learning method.

### 3.3.1. Shadow reconstruction using an existing 3D Model

### Ray tracing

We define shadow as that area in VHR images where direct sun lights are blocked. However, also areas that do not follow this definition suffer from lower reflectance. Indeed, if the incidence angle between a ray from the sun and the illuminated surface is small, reflectance is reduced. This effect is in principle quantified by means of the so-called Bidirectional Reflectance Distribution Function (BRDF), (Nicodemus, 1965). However, the notion of BRDF is difficult to apply in practice as object reflectance is both material and environment-dependent. Therefore, only shadows are reconstructed in the areas where direct sun lights are blocked.

Given a 3D model, it is possible to estimate the locations of shadows in a VHR image, according to the definition above, if the sun position and camera interior and exterior orientations are known. The sun position is defined in spherical coordinates by one azimuth and one elevation angle, which can be accurately estimated given the time of acquisition of the VHR image. Camera orientations are accurately estimated by means of bundle adjustment. Given this information, a computational efficient procedure is to use so-called z-buffering. However, z-buffering is known to cause unwanted aliasing and acne effects. Several approaches, e.g., (Dimitrov, 2007; Wimmer et al., 2004), have been designed to mitigate these effects, but implementation is not as straightforward and results are not as accurate as ray tracing (Whitted, 1979).

Ray tracing simply generates a viewing ray from each pixel in the image plane through the project center the camera to intersect with the 3D model. If the first point of intersection is found, another ray is generated from this point to the sun position to intersect with the 3D model again. If an intersect point is found, the image pixel is in the shadow, Otherwise, the pixel is in the sun. The disadvantage of this approach is the high computational load. The computational costs of ray tracing are $O(N * W * H)$, where $N$ represents the number of triangles of a 3D model, often represented by a triangular mesh. The size of the image plane is given as $W \times H$, where $W$ is the number of image columns, and $H$ the number of image rows. Typically, a VHR image consists of millions of pixels, while a 3D model may also contain millions of triangles. Parallelization in CPU and GPU is often applied to reduce computation time. In addition, by incorporating the field of view of the camera in the implementation, the number of triangles considered in the ray tracing can be reduced. However, these two adaptations will only reduce computations linearly, which is not enough to make the overall method computationally feasible.

### KD tree

A KD tree is a space-partitioning data structure, which can ideally reduce the cost of intersection exponentially. Using a KD tree, search operations are split in two phases. In the first phase, objects, which are in our case the triangles of the 3D mesh, are stored in a search tree of depth $k$. In the second phase, individual triangle intersection points are determined much faster, given the availability of the search tree. If the triangular space is split into two ($k = 2$), a ray first intersects one of two subspaces and next only consider the intersections with triangles in that subspace. If an intersection is found, all triangles in the other subspace can be discarded.

To find optimal planes to split the triangles, a local greedy surface area heuristic (SAH) is used under the assumption that rays are equally distributed in space and triangles cover the entire space (Wald and Havran, 2006). This heuristic minimizes the expected heuristic intersection cost, $C_V(p)$, of splitting plane $p$, defined by

$$C_V(p) = \kappa_T + \kappa_I\left(\frac{A(V_L)}{A(V_L)+A(V_R)}|T_L| + \frac{A(V_R)}{A(V_L)+A(V_R))}|T_R|\right). \tag{3.1}$$

Here $V$ indicates the space to be split. In Figure 3.2 this is the black bounding box containing all triangles. In the figure, splitting plane $p$ is indicated by a red bold line. $\kappa_T$ denotes the costs for traversing the two subspaces separated by by splitting plane $V$, i.e., the rectangles left and right of the red bold line in Figure 3.2. $\kappa_I$ denote the costs for intersecting one triangle, while $A(V_L)$ and $A(V_R)$ denote the area of the left and right subspace, respectively. The ratio $\frac{A(V_L)}{(A(V_L)+A(V_R))}$ approximates the percentage of rays going through subspace $V_L$, while $|T_L|$ and $|T_R|$ are the number of triangles in the left and right subspace, respectively.

The equation favors subspaces that are as large as possible but at the same time include as few triangles as possible. The configuration in Figure 3.2b, shows a case where the split, shown in bold red line, is far from optimal: the left subspace is a little bigger than in Figure 3.2a, but it contains many more triangles. Every subdivision step minimizes local costs while assuming no subdivision is further performed. The accumulated minimum local costs tend to overestimate the correct total costs as subspaces are in most cases further subdivided. However, in practice, this approximation works well (Wald and Havran, 2006). The space is subdivided iteratively until $C_V(p) > \kappa_I \cdot |T|$.

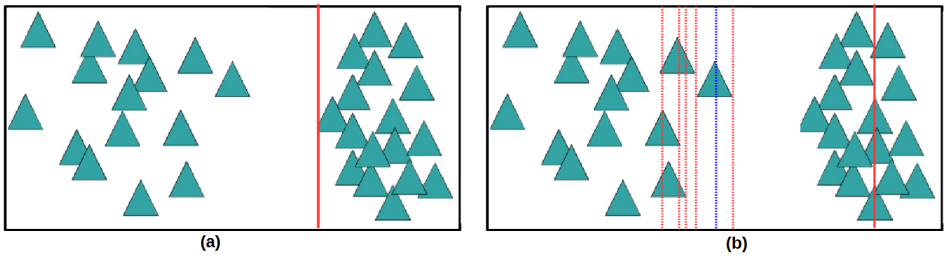



Figure 3.2: Two ways, (**a**,**b**), of splitting a space containing triangles by a subspace, here indicated by a red bold line. Split (**a**) is more efficient. The red dashed lines in (**b**) indicate examples of candidate splitting planes. The blue dashes line in (**b**) indicates a wrong candidate plane as the efficiency of this split is worse than that of each of the adjacent splitting planes.

If a splitting plane slides through the interior of a triangle, compare the blue dashed line in Figure 3.2b, $|T_L|$ and $|T_R|$ do not change while $C(p)$ changes in a monotonic way. So optimal planes always occur when the $|T_L|$ and $|T_R|$ are changing, locations indicated by the red dashed lines in the figure. Therefore, the six bounding box planes of each triangle are chosen as candidate planes.

A naive way of plane selection is that for each candidate plane, all left and right triangles are traversed to calculate the SAH cost of that plane, which has a

computational load of $O(N)$. This should be done for all $6 \cdot N$ candidate planes so the total computational costs to identify the optimal plane is $O(N^2)$. Adding the costs for recursive subdivision gives a total cost of approximately 2 times of $O(N^2)$ for establishing a tree structure. This construction time is not acceptable as it may take hours to apply it in case of 1 million triangles. The naive way is notably expensive due to the costs of classifying triangles for each candidate plane.

In this research, an $O(Nlog^2N)$ approach is adopted instead. A candidate plane is either on the left of, on the right of or passing through a triangle. The number of triangles of each of these types is denoted as $p^+$, $p^-$ and $p^|$ respectively. Once candidate planes are sorted, these numbers can be updated incrementally while sweeping through the sorted list of candidate planes. The full details of the update rules can be found in Wald and Havran (2006). As sorting the candidate planes costs $O(NlogN)$, while sweeping costs $O(N)$, the total costs for finding an optimal plane are $O(NlogN)$. With recursive division, the overall costs are $O(Nlog^2N)$. Consequently, construction time for 1 million triangles can be reduced to minutes in practice, (Wald and Havran, 2006). An even faster algorithm with a cost of $O(NlogN)$ is obtained by avoiding sorting at each subdivision step, which can be achieved by memorizing the sequence of planes throughout the subdivision (Please note that the triangles and therefore the candidate planes stay the same). The rest is the same as the $O(Nlog^2N)$ approach. However, at the cost of a more complicated implementation, reduction of computation time is only a factor 2 to 3 in practice. In this chapter, the $O(Nlog^2N)$ approach is implemented due to its simplicity of implementation. Parallelization of the KD tree construction is not considered for the same reason.

### 3.3.2. Adaptive erosion filtering

Reconstructing shadows by ray tracing using an existing 3D city model provides sufficient variation in training samples for VHR image classification. However, this procedure also introduces many mislabeled samples. Shadow accuracy near boundaries is affected by misalignments between 3D models and images, and lack of accuracy of the 3D model, especially considering roof details. As the absolute accuracy of both data sources is high, the misalignment is only one or two pixels. These effects are not explicitly considered, as the effects of an inaccurate 3D model are more serious. To mitigate these affects, a disk-shaped erosion is applied to remove samples near the boundary of the reconstructed shadow image from both shadow and non-shadow training samples. To decide on a good size for the filtering element, we define a simple adaptive rule. The left image in Figure 3.3a shows that small roof details are able to cast large shadows. The size of such casted shadow depends on the height of the roof detail and the position of the sun. We assume that small details of at most 1 meter height on the roof are artifacts from modeling. The position of the sun is parametrized by two angles: azimuth and elevation. To keep the criterion simple, we only consider elevation. The size, $s$, of the filter is estimated as

$$s = \frac{h}{tan(e) \times r}, \tag{3.2}$$

where $r$ denotes the the average ground sampling distance (GSD) of the image, $h$ the maximum height of roof details, while $e$ represents the angle of the sun elevation. As similarly roof details may be missing in the 3D model, the same erosion filter is also performed on non-shadow areas. In Figure 3.3a, the middle two images show an RGB image and reconstructed shadow image. In the square, a small object is poorly reconstructed and some false shadows are constructed. In the circle, a small wall or fence is missing in the 3D model, resulting in some lack of shadows. By applying filters for both shadows and non-shadows, these mislabels are removed. In the right image, the pixels in the white regions are filtered, black pixels are samples for shadows and gray pixels are samples for non-shadows. The selected samples have become more reliable.

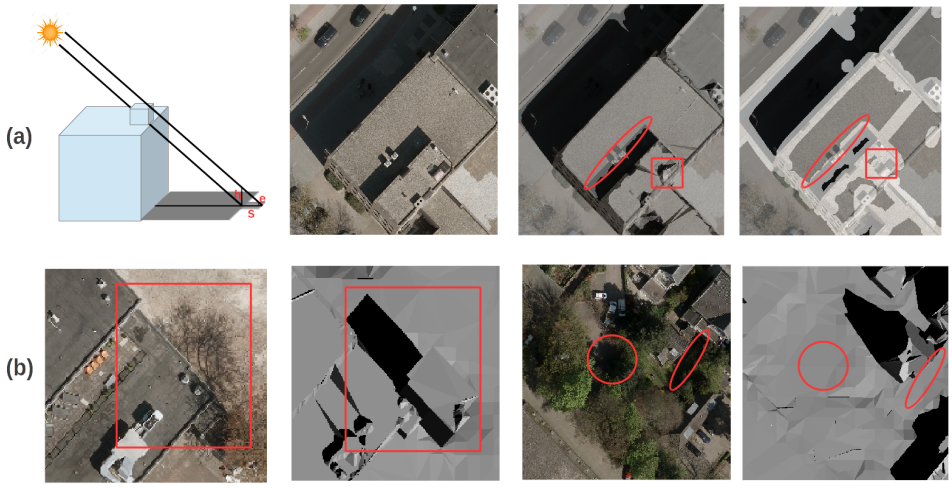



Figure 3.3: (**a**) The left image shows how the size of the adaptive erosion filtering element is calculated from the height of detail and the elevation angle of the sun. The middle two images show the VHR image and reconstructed shadows from a 3D model. The square and circle indicate shadow and non-shadow mislabels. The right image shows that mislabels are reduced by filtering. Shadow, non-shadow, and filtered samples are shown in black, gray, and white, respectively. (**b**) The left two images show a case where a building is removed after the 3D model was acquired. This introduces mislabeled shadow samples. The right two images show that a 3D model with omitted trees introduces mislabels in non-shadow samples.

However, many training samples are still mislabeled as differences in acquisition time between VHR imagery and 3D model and tree modeling are not incorporated. As shown in the left two images in Figure 3.3b, the 3D model contains a building which is removed before acquisition of the image. In the right two images, many shadows casted by trees are missing. These mislabels are not trivial and need to be considered when a classification method is selected.

### 3.3.3. Classification using automatically selected samples

A proper classifier should be capable of generalizing the characteristics of complex shadows and at the same time be robust to mislabeling effects. Four common classification approaches, quadratic discriminant analysis (QDA), support vector machine (SVM), K nearest neighbors (KNN) and random forest (RF), are discussed by considering these two requirements. The first method is based on a Gaussian distribution of features. To capture complicated properties of shadows, we use two feature domains: RGB and ratio. The other three classifiers are discriminative classifiers and we confirmed that the ratio feature does not improve classification results.

**QDA classification with decision fusion**

QDA (Theodoridis et al., 2008) assumes a Gaussian distribution for each class and fits a quadratic surface to train samples to best separate classes. QDA is robust to mislabeling effects as the Gaussian parameters are estimated properly when a large portion of samples is labeled correctly. However, shadows in urban areas are often complicated and certainly not Gaussian distributed. For example, assuming RGB features have a multi-variable Gaussian distribution, property 1, low radiance of shadows, described in Section 3.1, can be generalized. However, property 2, higher hue of shadows, is ignored. A decision fusion approach (Fauvel et al., 2006) is adopted to fuse results from QDA on different feature domains, RGB and ratio, to generalize different properties using QDAs (Zhou and Gorte, 2017). This approach uses fuzzy memberships and assesses the reliability of two classifiers to fuse the result properly. Fuzzy membership describes the confidence of a pixel belonging to a certain class. If two classifiers have different decisions on the class of a pixel, the decision with a higher membership degree is chosen. However, when two classes are not clearly separate in feature space and further confusion is added by mislabeling, a pixel can have high membership degrees for two classes, which means the classifier is not confident in classifying the pixel. Shannon entropy (Bromiley et al., 2004) is introduced to measure the reliability of membership degrees as assigned by a classifier. When membership degrees to two classes are high, the reliability is low, and the membership degrees are weighted by the reliability for decision fusion. This approach improves the QDA classifier to have more power to capture complex shadow properties. However, the properties 3 and 4 from Section 3.1 are still not considered in this method.

**Support vector machine**

SVMs have been successfully used for shadow classification (Lorenzi et al., 2012; Guo et al., 2013). SVMs aim to fit an optimal hyperplane to the training samples to maximize the margins between the separating plane and the closest training samples (support vectors) (Theodoridis et al., 2008). If the classes are not separable or the separation surface is nonlinear, the input data are mapped by kernel functions to a higher dimensional feature space where it is possible to find an optimal hyperplane. This approach avoids assumptions on the data distribution and is effective in deriving complicated nonlinear hypersurfaces using kernels to separate classes. However, mislabeled samples near the separation boundary will increase

the complexity of SVMs' boundaries and result in an overfitting problem (Frénay and Verleysen, 2014). A soft-margin using a regularization term to penalize samples located at the wrong side of the separation hyperplane. When the penalty value of the regularization term is set small, SVMs are more robust to mislabeling, but SVMs loss the ability to find accurate separation boundaries.

**K nearest neighbors**
KNN is also a non-parametric classifier. They do not use training samples to estimate a model, but they store training samples to decide which class should be assigned to an unseen sample. An instance is classified by majority voting from k nearest neighbors, which means the class for the instance depends on the most common class among its k nearest training samples. This approach is capable of capturing complicated data distributions if an optimal k is selected. A larger k can suppress noise and smoothen separation boundaries, but it may affect the ability to capture detailed properties from the training samples. Okamoto and Nobuhiro (1997) (Okamoto and Yugami, 1997) showed that the robustness of KNN to mislabeled samples is dependent on an optimal selection of k. K should increase as the ratio of mislabeled samples increases. So, the optimal value of k is case-dependent and difficult to set adaptively. Cross-validation is often used for choosing the optimal k, but an accurate validation dataset should be manually selected.

**Random forest**
Ensemble learning consists of a collection of classifiers and has been applied successfully in remote sensing. The most well-known classifiers are AdaBoost (Miao et al., 2012) and RF (Gislason et al., 2006). AdaBoost is an ensemble of classifiers, often decision trees, that reweighs samples by giving incorrect classified samples more weight for the next classifier. Every classifier gets a weight based on its classification accuracy and a weighted majority voting determines the final class assigned to test samples. AdaBoost tends to avoid overfitting when mislabels are trivial (Briem et al., 2002) and generally has good classification performance. However, if mislabels are introduced, the reweighting mechanism would emphasize to classify the mislabeled sample in the next decision tree, which results in a serious overfitting problem. RF, instead, adopts a bagging mechanism by randomly selecting a subset of samples and features by replacement for training each classifier. Breiman (2001) (Breiman, 2001) demonstrates that RF has a classification accuracy that is comparable to AdaBoost on a test consisting of 20 datasets. Moreover, the assembly of decision trees by bagging guarantees that RF is robust to noises (Breiman, 2001). It is tested to outperform AdaBoost significantly when 5% of mislabeled training samples are introduced (Breiman, 2001).

When using only one decision tree for classification, the choice of attribute selection and a pruning method are important (Pal, 2005). The Gini index (Breiman, 2017) is chosen to measure impurity of separated classes when an attribute value is chosen to split a tree. An attribute with lowest impurity of separation is chosen as the optimal value to split the tree. Choosing a pruning method affects the performance of decision trees due to overfitting effects. However, decision trees in RF do not need to be pruned. Breiman (2001) (Breiman, 2001) demonstrate that

as the number of trees grows, the generalization error of classifying unseen data always converges and overfitting is not a problem. This simplifies the setup of the hyper-parameters in RFs.

### 3.3.4. Evaluation of classification

Pixel-based classification in VHR images often has a salt and pepper noise problem. First, an opening morphological filter is applied and next a dilation morphological filter is applied to correct unwanted erosion effects. The size of the dilation morphological filter is larger than erosion filter due to penumbra effects. The equation for calculating penumbra areas is provided in Dare (2005) with respect to the height of object, $h$ and elevation angle of the sun, $e$, as follows:

$$w = h(\frac{1}{\tan{(e - \frac{\epsilon}{2})}}) - \frac{1}{\tan{(e + \frac{\epsilon}{2})}})), \tag{3.3}$$

where $w$ is the penumbra width and $\epsilon$ is the angular width of the sun which is $0.5°$. As we filter the boundary pixels from reconstructed shadows to avoid issues caused by misalignment between model and image, and low quality of model near rooftops, pixels in the penumbra areas are not included in the training samples. These pixels are more likely classified into non-shadows as they are brighter. If shadows are used to estimate building height or for change detection, the estimated building height will be too low and false alarms near shadow boundaries will result in false change detection. Therefore, the detected shadows should be dilated with size of $w$. In this study, the border between shadow and non-shadow pixels in a mixed penumbra area was visually chosen in the middle of the areas for creating test dataset. Therefore, we decided to dilate with a half size.

The test dataset is drawn manually by drawing polygons in QGIS. The resulting vector file is then converted to reference images, shown in Figure 3.4. Several criteria have been designed to create labels for the test dataset: (1) Shadow pixels are labeled based on whether it was blocked from sunlight even they have high reflectance. In penumbra areas, the boundary between shadow and non-shadow is visually chosen in the middle of the areas. (2) Shading effects are not considered in manual labeling. If the incidence angle is small, roof pixels may look similar to shadow pixels. Still, we label them as non-shadows as they are not blocked from sunlight. (3) Only shadows caused by buildings are selected, while shadows from vegetation, cars and objects in the garden are excluded. The building shadows for Amersfoort are manually selected. As the 3D model for Toronto is manually created, the building shadows are selected from the reconstructed shadows. Shadow boundary parts are excluded from the image to reduce the effects of misalignment between images and 3D model, and inaccurate details in rooftops. Wrong shadows due to a few model errors are also excluded as shown in Figure 3.4b (3). As shown in Figure 3.4a, labeling these cases manually is time-consuming. If shadows are used to help the detection of different objects, shadows are notably useful for building detection. Vegetation and cars can be detected accurately from color information and objects in the gardens are not interesting. These shadows are excluded from the test dataset by roughly draw bounding polygons. (4) Shad-

ows from small details, less than 1 m by 1 m, on the roof are often not necessary to detect. As they are too many to be excluded, these shadows are included in non-shadows as shown in Figure 3.4a (4). Accordingly, shadows from these details which are detected from images should be converted to non-shadows.

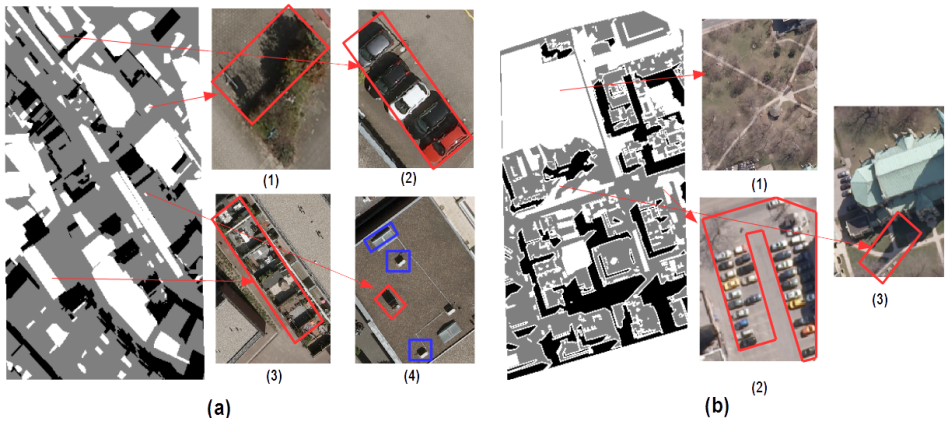



Figure 3.4: Reference images for Amersfoort (**a**) and Toronto (**b**) in the areas indicated in Figure 3.1b. The black, gray, and white pixels indicate shadow, non-shadow and excluded pixels. In Figure (**a**), image (1), (2) and (3) show shadow from vegetation, cars and objects in the gardens which are excluded from the reference images, while image (4) shows that if shadows from details on roofs ,in the red box, larger than 1m *1m are excluded, and the smaller details in the blue boxes are included in non-shadows. In Figure (**b**), image (1) and (2) show vegetation and cars are excluded in the reference images, while image (3) shows wrongly labeled areas indicated by reconstructed shadows due to model errors are excluded.

All classification methods are trained by samples labeled from the reconstructed shadow image and are consecutively applied to classify the test dataset. The results are compared with manual labels to derive the evaluation of their performance. To compare effects of different methods on shadow detection qualitatively, four metrics are selected: completeness ($Comp$), correctness($Corr$), overall accuracy(OA) and kappa coefficient (KC).

$$Comp = \frac{TP}{TP + FN}; \quad Corr = \frac{TP}{TP + FP}. \tag{3.4}$$

$$\begin{aligned} OA &= \frac{TP + TN}{N}; \qquad KC = \frac{OA - p_e}{1 - p_e}, \\ p_e &= \frac{(TP + FP) \times (TP + FN) + (FN + TN) \times (FP + TN)}{N \times N}. \end{aligned} \tag{3.5}$$

N presents the total number of pixels. TP, FP, and FN denote the true positives, false positives and false negatives respectively. The $Comp$ and $Corr$ describes the completeness and correctness of shadow detection. Overall accuracy refers to the percentage of overall shadow and non-shadows pixels correctly detected. KC is a robust measurement of how well a classifier works by taking into account how

well a classifier would work simply by chance. $p_e$ denotes the accuracy of detection by chance.

## 3.4. Experiments and comparisons

Experiments are applied to two city areas in Amersfoort and Toronto. The experiments consist of two steps: shadow reconstruction to obtain training samples and classification after training with these samples. The purpose of the first step is to show how efficient ray tracing is for shadow reconstruction in large 3D models. In the second step, the Amersfoort and Toronto dataset are tested with the four methods described in Section 3.3.3. By testing comparative methods in different environments, the robustness of each method is tested with respect to its capability of capturing complicated shadows properties and mislabeling effects.

### 3.4.1. Feasibility of shadow reconstruction for training samples

Ray tracing with KD tree acceleration for shadow reconstruction was implemented in C++ on a HP laptop with 8 GB ram and quadcore processor. The KD tree construction is not parallelized, but ray tracing is parallelized in the CPU. As described in Section 3.3.3, shadow ray tracing requires camera parameters and the sun position. The camera parameters are obtained from their bundle adjustment files. For the Amersfoort data, from the acquisition time of the image, an accurate azimuth and elevation angle of sun position are easy to obtain. They are 44.58° and 140.41° respectively. The Toronto data do not provide acquisition time, but the sun position can be estimated. One obvious building roof corner in the 3D model and its shadow pixel in the image are found. From the shadow pixel, a camera ray is generated to find its intersection with the model. The intersection shows the shadow point in the 3D model casted by the roof corner. The line between the shadow point and roof corner is actually a sun ray. The 3D line can be transformed into spherical coordinate system. Therefore, the azimuth and elevation of Toronto data are estimated at 49.93° and 133.13° respectively. The shadow images created have the same size as the aerial images: $11310 \times 17310$ and $7500 \times 11500$ for Amersfoort and Toronto, respectively.

The time for building a KD tree is 21.27 s and 1.14 s respectively as the number of triangles in the Amersfoort model is 10 times higher than in the Toronto model. The ray-tracing time is 3840 s and 281 s, respectively. The number of triangles and image size is larger in Amersfoort, but the most important factor is the structure of 3D models in the camera field of view. As shown in Figure 3.5, much fewer camera rays from Toronto image pixels intersect with the 3D model resulting in a large number of white pixels. Due to the KD tree structure, the rays from most of the white pixels only need one intersection test with the bounding box of the model indicated as the 2D red box. The reconstructed shadow images are shown in Figure 3.5.

We conducted an experiment to subsample the point cloud of Amersfoort at different size and make different triangular meshes while preserving the structure

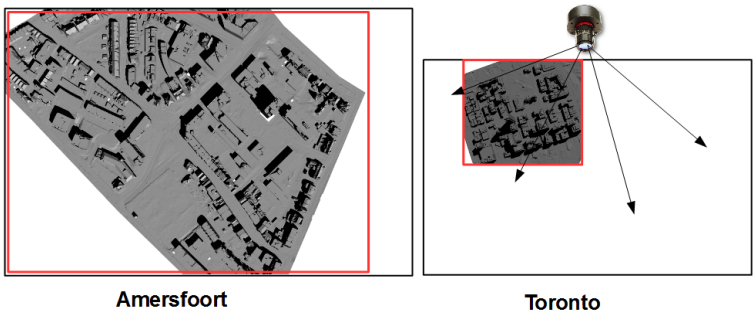



Figure 3.5: Shadow images from reconstruction by ray tracing for Amersfoort and Toronto 3D models. The pixels with rays which do not intersect with red boxes are filled in the white color with fast computations. Toronto dataset has much more rays which do not hit the model.

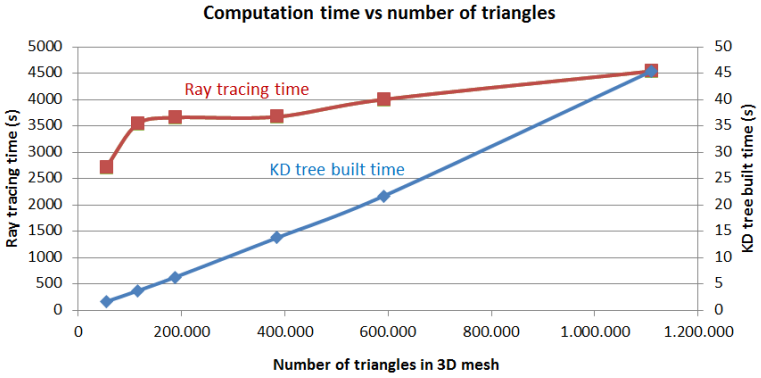



Figure 3.6: KD tree construction and ray tracing time with respect to the number of triangles in a 3D mesh model.

of the 3D model for ray tracing. These models are applied to test the efficiency of building KD trees and ray tracing relative to the number of triangles. The number of triangles of five simulated models varies from 54,000 to 1,1100,000. In each ray-tracing experiment, the number of white pixels is the same while the number of triangles is changed uniformly. The result is shown in Figure 3.6. The time for building a KD tree scales nearly linear with the number of triangles, which confirms the computation complexity of $O(Nlog^2N)$. As N becomes larger, $log^2N$ is hardly noticeable. Even from 54 k to 1 million, the $log^2N$ term only contributes by a factor of $\frac{log^2(1million)}{log^2(54\,k)} = 1.6$. Therefore, the time is almost linear in N. Due to the large number of rays from pixels, around 200 million in our experiment, ray tracing needs 45 mins with 54 k triangles. However, when the number of triangles increases a factor of 20, from 54 k to 1.11 million, the processing time only increases by a factor of 1.6. It also confirms that using a KD tree structure, the time of ray tracing is only increasing logarithmically with the number of triangles. As the

proposed approach is scalable by just using more CPUs, the experiment shows the ray tracing using the effective KD tree construction is practically feasible to reconstruct shadows for processing even larger models.

### 3.4.2. Method comparison

The classification methods used in the chapter are implemented in MATLAB using the precoded classifiers. The samples for shadow reconstruction are unbalanced as shadows occur less than non-shadows. Therefore, equal numbers of samples, 10000, are randomly selected for each class for classification. Tests also showed that adding more samples did not improve results as the randomly selected samples are representative enough for both data. Only RF randomly selects training samples from 1 million samples for each decision tree. This step increases the variety of decision trees but does not increase the computations. Parameters chosen for the different methods are the same for both datasets to test the applicability of the methods. QDA fusion is free of any input parameters. For SVMs, the weight for the regularization term is 0.2 for training samples with mislabels. The weight is set to 0 for Toronto as only a limited mislabels is present. For KNN, k is chosen to 11 to have robustness to mislabel. For RF classification, all three features are used in every tree and 100 trees are selected as the performance does not significantly improve by adding more trees.

**Amersfoort comparative results of supervised methods**

Before classification, according to Section 3.3.2 an adaptive disk-shaped erosion is applied to remove the boundary parts of shadows and non-shadows. With its 3.5 cm spatial resolution, the size of the filter is set at 30 according to Equation (2). After each classification, according to Section 3.3.4, a filtering is applied to mitigate the salt and pepper noise problem of classification results. An erosion filter with a size of 1 is applied to remove noise from detection. According to Equation (4), if we assume the height of buildings at 10 m, a half of the penumbra width is 8 cm. Therefore, a dilation filter with a size of 3 is then applied to reconstruct the erosion effects and reduce penumbra effects.

Table 3.1 shows that KNN and RF gives similar and outperform the other classifiers. The producer's accuracies (*Comp*) are 95.6% and 93.9. The user's accuracies (*Corr*) are 95.7% and 96.1% respectively. This indicates that the two methods obtain a good completeness of shadow detection while keeping a high correctness. Problematic areas as shown in Figure 3.7a,b, like dark roofs in the sun and reddish objects in the shadow are well distinguished by RF. As shown in Figure 3.7a, KNN is worse in classifying dark roofs. This explains why its *Corr* (correctness) is a bit lower. In Figure 3.7b, both perform well on detecting shadows in reddish objects. The high overall accuracy of more than 95% and KC around 0.91 for both methods show the capability of both methods to generalize the complicated properties of (non-)shadows from mislabeled samples. The complete result of shadow detection from RF is shown in the left image of Figure 3.1d.

The QDA fusion method has a relatively low completeness *Comp* (89.9%). It fails to detect reddish objects in the shadow correctly with two cases as shown

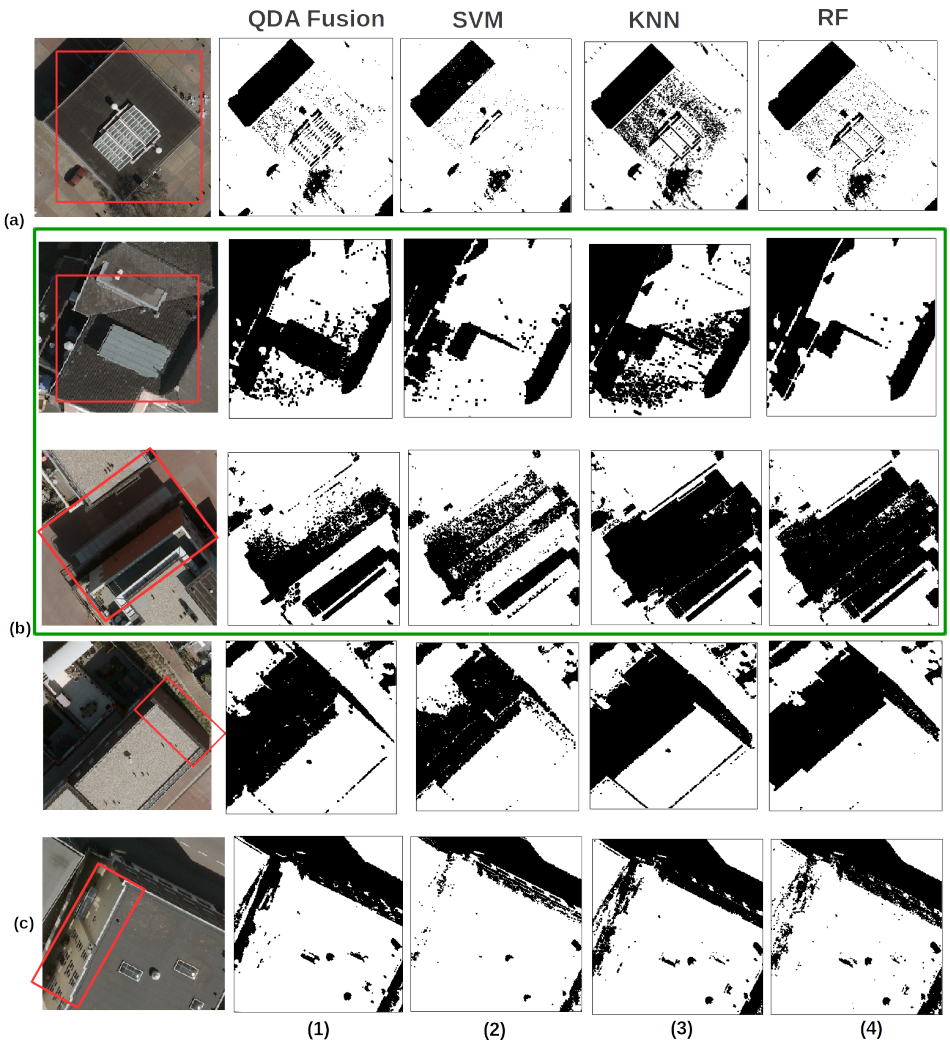



Figure 3.7: Results of four methods (1) QDA fusion; (2) SVM; (3) KNN; (4) RF for difficult cases: (**a**)two buildings with dark roofs; (**b**) two buildings with reddish bright facades and a bright reddish road. (**c**) a building with a very bright facade. The locations of these buildings are indicated in Figure 3.1. The results of property methods for the two cases in the green box are shown in Figure 3.8a.

| Method | *Comp* % | *Corr* % | OA % | KC |
|---|---|---|---|---|
| QDA Fusion | 89.9 | 95.8 | 93.5 | 0.87 |
| SVM | 84.66 | **98.9** | 92.1 | 0.84 |
| KNN | **95.6** | 95.7 | 95.7 | **0.91** |
| RF | **93.9** | **96.1** | 95.4 | **0.92** |
| Tsai's RGB | 99.1 | 70.9 | 80.5 | 0.62 |
| Tsai's ratio | 82.3 | 91.5 | 88.2 | 0.76 |
| Adeline's | 91.0 | 97.5 | 94.2 | 0.88 |

Table 3.1: Comparative results for shadow detection in Amersfoort. The upper four methods are supervised learning methods, while the lower three methods are widely used property methods.

Figure 3.7b. In the upper case, the reddish road pixels near the reddish facade are darker, so they are correctly detected as shadows. However, when reddish road pixels far away from the facade become brighter, they are wrongly classified. The same happens with reddish facades in the shadow in two buildings as pixels receive more reflected light. In the RGB domain, dark pixels could be assigned to shadows, while in the ratio domain, the pixels with reddish color could be assigned to non-shadows. The QDA fusion approach makes a decision based on two variables: confidence and reliability of each result. When the reddish pixels become lighter, the confidence that these pixels belong to shadows is less, while more reddish reflectance gives more confidence that they belong to non-shadow in the ratio domain. Therefore, the decision fusion approach is still not capable of capturing complicated shadow properties. Still QDA fusion has better results than SVM due to its robustness to mislabels. The ability of SVM to find good separation in complicated areas is largely deteriorated by mislabels. The regularization term is not really helpful to deal with the mislabels as SVMs cannot distinguish the pixels which define complicated separation boundaries from pixels which are mislabeled. This results in the lowest *Comp* (84.7%) and highest *Corr* (98.9%). It indicates that SVMs are not robust to mislabeling. In Figure 3.7c, four methods show low detection accuracy on the high reflectance facade with high RGB values.

**Amersfoort results to property methods**

Two property methods by histogram thresholding from Tsai (Tsai, 2006) and Adeline et al. (Adeline et al., 2013) are selected. The main difference of these two methods is their thresholding approach. Tsai uses Otsu's method (Otsu, 1975), while Adeline et al. set the threshold at the first valley of the histogram which gives the best shadow detection performance in their comparative study. Tsai's method is applied to two feature domains RGB (Tsai's RGB) and ratio (Tsai's ration) respectively, while Adeline's method is applied to intensity. As shadows in Amersfoort are strongly affected by reflection from environment and materials, the histogram is noisy. The value at the first valley is certainly not the best threshold and often much lower than the best threshold. We smooth the histogram by Savitzky-Golay smoothing (Press and Teukolsky, 1990) and set a constraint that

the threshold found should not be lower than 25. As the property methods perform well only for a bimodal histogram, the image is split into small patches, in which bimodal behavior is more likely to appear. As shown in Table 3.1, Tsai's methods on RGB and ratio domain tend to over- and miss-detect shadows, respectively. In Figure 3.8a (1) and (2), Tsai's methods cannot distinguish dark roofs in the sun and reddish street in the shadow properly. Both have the lowest OA and KC. Adeline's method shows a better result than Tsai's method. In Figure 3.8a (3), Adeline's method shows a good result in classifying these two cases. Still, the completeness and KC are 3 % and 0.04 lower than RF, respectively. More importantly, thresholding at the first valley of a histogram is not robust in case of complicated shadows and tends to miss-detect shadows. This effect is more obvious in the Toronto experiment.

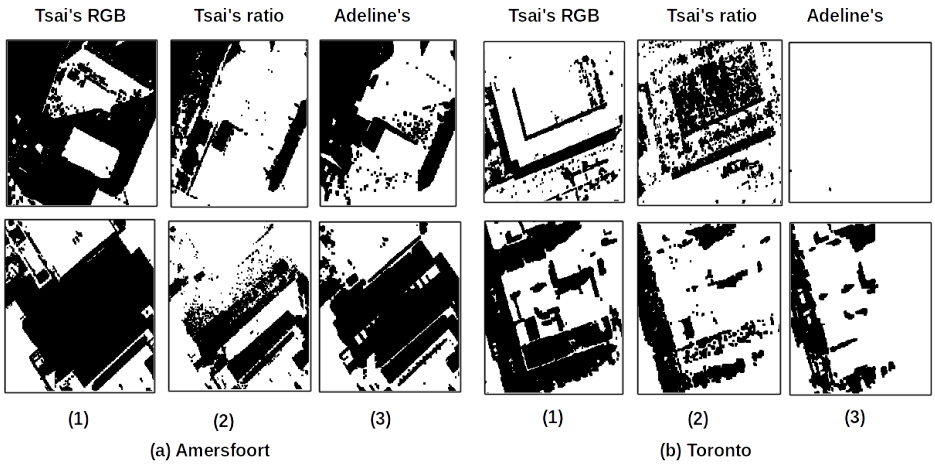



Figure 3.8: Results of property methods (1) Results for Tsai's method on RGB features; (2) Results for Tsai's method on the ratio feature; (3) Results for Adeline's method on the intensity feature; (**a**) Results of three methods for two cases in Amersfoort shown in the green box from Figure 3.7. (**b**) Results of three methods for two cases in Toronto shown in the green box from Figure 3.9.

**Toronto comparative results of supervised methods**

The same filtering steps are applied before and after classification as indicated in Amersfoort experiment. As mislabels are limited due to the accurate building models and sparse and limited trees in the Toronto training set, all methods have good completeness with completeness (*Comp*) values of more than 94 % shown in Table 3.2. However, QDA fusion has a relatively low *Corr*, 88.8%. QDA fusion makes a good detection on the bright shadows on the facade in Figure 3.9c, while the dark roofs of two buildings indicated in the red boxes are seriously misclassified as shadows in Figure 3.9a. It confirms that QDA fusion is still not good enough to capture complicate shadows. KNN has the highest completeness

of shadow detection at *Comp* value of 97.7 %. However, its correctness of shadow detection has a much lower value for Amersfoort and it has a lowest *Corr* (88.6 %) value, as KNN has difficulties in classifying very bright white walls and roofs as shown in Figure 3.9b. Although a few white pixels are labeled as shadows in the reconstructed shadow image, the percentage of white pixels in shadow training samples is higher than in the non-shadow training samples. As we select the same amount of shadow and non-shadow samples, more white pixels in the shadow training samples are selected. This problem can be solved by increasing the number of nearest neighbors, K. However, it may lose its capability to capture complicated shadow properties.

| **Methods** | ***Comp* %** | ***Corr* %** | **OA %** | **KC** |
|---|---|---|---|---|
| QDA Fusion | 95.0 | 88.8 | 95.2 | 0.88 |
| SVM | 94.2 | **94.0** | 96.7 | **0.92** |
| KNN | **97.7** | 88.6 | 95.9 | 0.90 |
| RF | **97.3** | **91.1** | 96.6 | **0.92** |
| Tsai's RGB | 98.5 | 83.4 | 92.9 | 0.85 |
| Tsai's ratio | 95.6 | 82.8 | 93.2 | 0.84 |
| Adeline's | 80.5 | 92.8 | 92.6 | 0.81 |

Table 3.2: Comparative results for shadow detection in Toronto.

SVM and RF have competitive results with balanced high *Comp* and *Corr* and a higher overall accuracy of more than 96 % and kappa coefficient around 0.92. The complete shadow detection result by RF is shown in the right image of Figure 3.1d. The good performance of SVM confirms that SVM is applied to shadow detection in literature in case of almost perfect training data. Even though RF has less accurate detection in Figure 3.9c as, the indicated facade in the shadow appears bright due to reflection and its material, but most of shadows on the facade are captured. Both methods make a good trade off in balancing dark objects in the sun and bright objects in the shadow in Figure 3.9a,c. This confirms that both SVM and RF are capable of capturing complex shadow properties. Figure 3.9d shows that the roof indicated in the red box is a bit darker due to shading effects. The incidence angle between sun ray and the roof is larger than its neighboring roof in the same building. With less reflectance and dark materials, all four methods make a wrong classification. However, this case is rare in the experiment which is also a reason why the object is not wrongly classified.

As indicated in Amersfoort experiment, SVM is expected to perform worse when the ratio of mislabeled training samples increases. This effect will be further tested in the next section by simulating mislabeled samples in the training data.

**Toronto results comparing to property methods**
Compared to Tsai's and Adeline's methods, the supervised approaches perform all better. Tsai's method tends to over-detect shadows as they have 98.5% and 95.6% completeness (*Comp*) respectively, but low correctness with 83.4% and 82.8%




Figure 3.9: Results of four methods (1)QDA with fusion; (2) SVM with rbf kernal; (3) KNN; (4) RF for difficult cases: (**a**) two building with dark roofs. (**b**) two buildings with a bright and white roof and facade, respectively. (**c**) a building with a bright facade under the shadow. (**d**) a dark roof under the sun is misclassified by all methods due to shading effects. The locations of these buildings are indicated in Figure 3.1. The results of property methods for the two cases in the green box are shown in Figure 3.8b.

(*Corr*) respectively. The low correctness of shadow detection can be seen from Figure 3.8b (1) and (2). Tsai's methods have difficulties to detect dark roofs in the

sun correctly. Adeline's methods tend to miss-detect shadows as the threshold at the location of the first valley is often lower than the best threshold. Therefore, the *Comp* is lowest at 80.5% , while the users' accuracy is high at 92.8% of *Corr*. The serious effect is shown obviously in Figure 3.8b (3). A large portion of shadows is miss-detected from two cases. In the lower case, the shadow result is combined from two patches. In the right half of the image, the threshold found is much lower than the best, so all shadows are mis-detected. The different results from adjacent patches also shows that Adeline's method is not robust due to complexed environment.

**Mislabeling simulation on Toronto dataset**

To test how robust different methods are to the effect of mislabeling in case of Toronto dataset and whether results are consistent to the Amersfoort case, different levels of mislabeling are applied to the non-shadow training samples in Toronto. The implementation is set as follows: First, shadow pixels are segmented using connected component labeling. Then randomly selected segments are inverted to non-shadows. This approach simulates a scenario where buildings did not exist in the building model but were newly built before the images were taken. Five levels of mislabeling are simulated by inverting 10% to 50% of original shadows in steps of 10%. The results are shown in Figure 3.10. The results from the clean dataset shown are included for comparison.

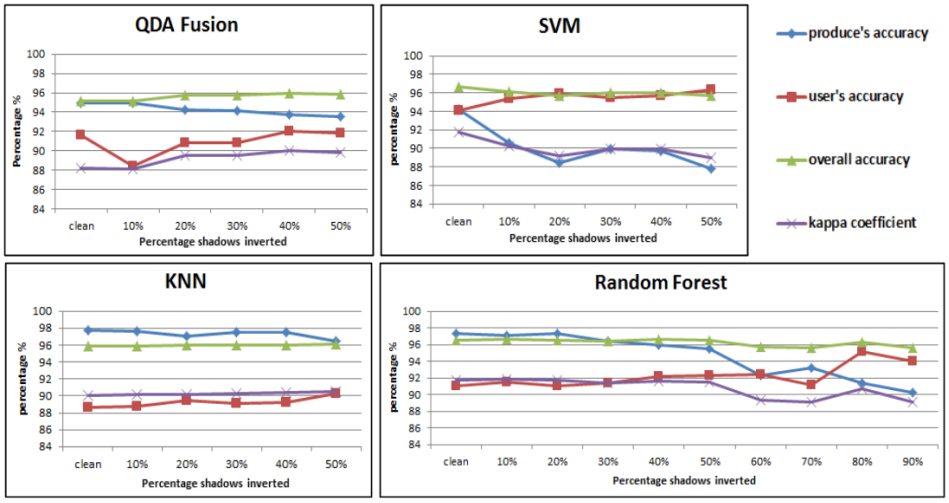



Figure 3.10: Results of four methods tested with six different level of mislabels. Kappa coefficient is also scaled to [1 100] to show in the same image with other metrics.

In general, the *Comp* decreases, and the *Corr* increases when the level of mislabeling increases in the non-shadow samples. When the non-shadow training samples are contaminated by many shadow pixels, fewer shadow pixels are detected but the detected shadows have higher correctness. As shown in Figure 3.10, QDA fusion and KNN are robust to mislabeling even with a high ratio of mislabels. The

*Comp* of the SVM drops quickly when mislabels begin to increase. It confirms the results from the Amersfoort experiment, which indicates that SVM has overfitting problems when training samples have been mislabeled. An interesting finding is that the performance stabilizes when 30% shadows are inverted. RF remains robust and has highest OA (more than 96%) and KC (around 0.92) when the percentage of inverted shadows is between 10% and 50%. To evaluate complete performance of RF to mislabeling, we simulate the percentage of inverted shadows until 90%. The *Comp* has a trend to decrease when 60% of shadows are inverted. In particular, the *Comp* decreases dramatically from 95.5% to 92% once 60% of shadows are inverted; however, the *Corr* stays the same. An interesting finding is that after 70% shadows have been inverted, the *Corr* starts to increase, which causes KC to increase. This probably means that mislabeled samples help to detect dark objects in sun which are seriously mixed by shadows. Due to its unstable performance, random forest is recommended when an urban area has not changed dramatically. It is also a reasonable assumption that if an urban area has changed dramatically, the 3D model is better to be discarded.

## 3.5. Conclusions

This chapter proposes a machine learning approach to use reconstructed shadow from LiDAR data as training samples for shadow detection. A ray-tracing approach combined with an effective KD tree construction algorithm is designed. By testing the algorithm to different sizes of 3D models, experiments show that building a KD tree is efficient as the computation time almost scales linearly with the number of triangles. However, ray tracing is more time-intensive, but the time is almost linear with the logarithm of the number of triangles. With parallel computing, the proposed shadow reconstruction approach is practically feasible and scalable by just using more CPUs for processing even larger 3D models.

A comparative study on four classification methods is performed to choose one with two criteria: capability of generalizing the complicated shadow properties and robustness to a certain level of mislabeling. The QDA fusion is robust to mislabeling but has difficulties in capturing complicated shadow properties. The performance of KNN relies on the number of nearest neighbors and is case-dependent. SVM is confirmed in Toronto with a power in classifying complicated shadows as shown in many studies (Arbel and Hel-Or, 2011; Lorenzi et al., 2012; Guo et al., 2013). However, it is strongly affected by mislabels. RF outperforms the other three methods in all the experiments when the mislabeling effect is not dominant.

However, when using shadows detected from single images to compare with shadows reconstructed from LiDAR data to detect building changes, shadows reconstructed from inaccurate 3D city models created from LiDAR data will directly affect the accuracy of change detection. More importantly, shadows only indicate partial 3D information. Therefore, they can only be used to derive partial changes.

# 4

# 3D building change detection and updating using a stereo pair

*As more complete 3D information can be extracted from a stereo pair, using stereo images can provide better changes detection from LiDAR data than using shadows in a single image. However, the quality problem of 3D information in both LiDAR data and stereo images will result in false alarms and directly affect the accurate change detection. LiDAR point clouds are sparse and irregularly spaced, and have mixed returns near building edges, while 3D information extracted from stereo images are affected by shadow and low texture. This chapter proposes LiDAR-guided edge-aware dense matching (LEAD-Matching) to address these problems explicitly for detecting accurate building changes. Data sparsity and irregular spacing is addressed by densifying LiDAR points in a form of a digital surface model (DSM). Instead of applying interpolation with associated edge problems due to mixed returns, three candidate DSMs are created by linking each DSM pixel to up to three planes as identified in segmented and triangulated LiDAR data. The candidate DSMs limit the disparity search space for dense matching, addressing low texture and shadow problems in images. Through edge-aware dense matching, the detailed building edges in stereo pairs determine the optimal heights to address LiDAR edge problem. Changes are detected where corresponding pixels from dense matching have large color differences. Due to homogeneous surroundings and shadows, only partial changes are initially detected. A second hierarchical dense matching step is employed to complete changes and update 3D information by propagating initial partial changes iteratively. The proposed method is applied on data from two cities, Amersfoort and Assen, the Netherlands, with around 1200 existing buildings. In both areas, the method successfully verifies unchanged buildings while detecting minimum changes of $2 \times 2 \times 2\ m^3$. The experiments also show that the proposed method outperforms two well-known change detection methods in terms of verifying unchanged buildings and detecting small changes simultaneously.*

This chapter has been accepted in the ISPRS Journal of Photogrammetry and Remote Sensing, 2020 (Zhou et al., 2020b)

## 4.1. Introduction

Up-to-date 3D city models are needed for many applications, e.g. water management, city climate assessment and urban planning. Airborne LiDAR data is widely used for constructing 3D city models, however, acquiring LiDAR data is expensive so the updating rate is low at national level. For example, open source nationwide point clouds are available for the whole of the Netherlands, but a complete update takes around 7-10 years (PDOK, 2019). Airborne Very High Resolution (VHR) images are often available every year due to their lower cost. Thanks to the development of dense image matching (Furukawa and Ponce, 2010; Hirschmuller, 2008), there is an opportunity to use point clouds from stereo images—photogrammetric point clouds (Remondino et al., 2014) for creating up-to-date 3D city models. However, point clouds extracted from stereo images are often outlying and incomplete in shadow and low texture areas (Huang et al., 2018). As urban areas do not change dramatically every year, stereo images are actually suitable for change detection and updating changed areas. In past decades, spectral information has been widely used for deriving changes (Dalla Mura et al., 2008; Bovolo et al., 2012; Volpi et al., 2013). However, especially for building change detection, the accuracy and effectiveness of using 3D geometric information is considered higher (Qin, 2014; Tian et al., 2014). Therefore, a method for detecting changes and updating buildings in LiDAR data using 3D information from stereo images is proposed. Our goal is to preserve LiDAR data in unchanged areas and update it with photogrammetric point clouds in changed areas. The updated point clouds can be used by any 3D reconstruction method or software to create up-to-date 3D models.

3D information extracted from both data sources have pros and cons. Airborne LiDAR data has high vertical accuracy, so accurate plane information extracted from LiDAR data has been used for decades for 3D building modelling. However, airborne LiDAR data is often sparse and irregularly spaced. If LiDAR point cloud is densified by interpolating to a DSM with uniform spatial spacing for change detection, the interpolated heights are not always reliable, especially near building edges due to mixed returns (Wu et al., 2011). For example, LiDAR points from ground, wall and roof near an overhanging roof get mixed from top view. Wrong heights directly result in false alarms in change detection. On the other hand, VHR stereo images with higher ground sampling distance (GSD) have detailed building edges and can extract dense point clouds. However, 3D points extracted from stereo images are missing or outlying in shadow and low texture areas, also resulting in false alarms in change detection.

These false alarms resulting from the quality problems so far prevented the development of an accurate 3D change detection and updating workflow that could meet requirements for a large scale 3D map, which often requires buildings with small size, for example $2 \times 2$ m$^2$, to be included. Several state-of-the-art methods (Tian et al., 2014; Qin, 2014; Du et al., 2016) only detect building changes over 100 m$^2$, 200 m$^2$ and 50 m$^2$ respectively.

In this chapter, we propose LiDAR-guided edge-aware dense matching (LEAD-Matching) on an stereo pair to address these quality problems. The method starts

from densifying LiDAR data by employing the plane information extracted from sparse LiDAR points, instead of using interpolation. As buildings often consist of planar surfaces, an accurate DSM height can be calculated, as long as a correct plane is assigned to each DSM pixel. LiDAR point clouds are first segmented and then triangulated in 2D with plane information assigned to each vertex. Each DSM pixel receives up-to three different planes from the three vertices of the 2D triangle the pixel locates in. For example, a DSM ground pixel near a building may fall in a triangle with three vertices namely ground, wall or roof, where in the most cases the correct plane is included. Three DSMs are created and then transformed to disparity images to guide the dense matching by limiting the disparity search space (DSS) to address the shadow and low texture problems in dense matching. The edge-aware dense matching method uses detailed building information from images to determine the optimal heights from three candidates to address the building edge problems in the LiDAR data. Note that, no priori classification of LiDAR data to ground, wall and roof classes is required. If corresponding pixels chosen by dense matching still have large color differences, these pixels indicate that the 3D information in the LiDAR data is outdated. However, changes are not detected completely in homogeneous areas. A second hierarchical dense matching step is applied to propagate partial changes iteratively and update 3D information simultaneously.

The scientific contributions are as follows. (1) To our knowledge, we are the first to propose LEAD-Matching method to detect accurate changes by integrating accurate plane information from LiDAR data and detailed building edges from a stereo pair to address quality problems in both data sources. (2) A novel two-step dense matching framework is proposed for accurate 3D building change detection and updating. (3) The proposed method is shown to obtain a successful building verification rate, while detecting unprecedented minimum changes of $2 \times 2 \times 2$ m$^3$ for updating large scale 3D maps.

## 4.2. Related work

Several articles (Qin, 2014; Tian et al., 2014; Qin et al., 2016) confirmed that detecting building changes from geometric information is more accurate than from spectral information. (Trinder and Salah, 2012; Tian et al., 2014; Qin, 2014) discussed the fusion of geometric and spectral information for change detection. However, tuning parameters for fusion on different datasets is difficult. Therefore, this chapter only discusses three possible approaches for deriving geometric change between LiDAR data and images.

### 4.2.1. Single image change detection

Shadow is an indicator of the 3D geometry of a scene in a single image. By using geometric information to reconstructed shadows in an image, changes are detected when the reconstructed shadows do not match the actual shadows in the acquired image. Shadow information has been used for detecting buildings in VHR images in Ok (2013). However, shadows are rarely discussed in change detection. To

reconstruct shadows, a watertight model like a DSM or a surface mesh needs to be constructed from the LiDAR data. However, an accurate model is difficult to obtain automatically due to quality problem of LiDAR data. More importantly, many lower buildings positioned next to a high building may not cast shadows, so only partial geometric change can be derived.

### 4.2.2. Direct geometric change detection

Detecting changes directly between two point clouds seems straightforward. Easiest is to subtract two DSMs interpolated from point clouds to detect changes. However, false alarms near building edges occur in both data sources described above. False alarms can be bigger than those resulting from the misalignment of the two sources. In Tian et al. (2010), a window-based approach, considering minimum height differences between two DSMs converted from photogrammetric point clouds, is applied to filter false alarms near building edges. But choosing an effective window size is critical, otherwise small building changes would be discarded. A similar approach (Teo and Shih, 2013) uses morphological filters to reduce building edge effects of DSM errors. Instead of converting point clouds to DSMs, 3D Euclidean distances are used to reduce false alarms near building edge by searching for corresponding points between two point clouds (Qin et al., 2016). The critical step in this approach is to find correct corresponding points and estimate their 3D distances. Iterative closest point (ICP) (Besl and McKay, 1992) is often used to register point clouds. Instead of using distances of nearest points as the change metric, the Hausdorff distance of local neighbors of nearest points is tested to be a better metric to deal with roughness and varying point density in point clouds. A surface based distance metric is provided by the multi-scale model to model cloud comparison (M3C2) algorithm (Lague et al., 2013) that further mitigates the effects of roughness and point density variation in point clouds. A least squares 3D surface matching algorithm (LS3D) (Gruen and Akca, 2005) registers two point clouds by minimizing distances from local surfaces instead of points. An outlier removal procedure is conducted in each minimization iteration when the distances between local surfaces are above a threshold. These outliers are considered as changes. However, none of the above methods addresses quality problems of photogrammetric point clouds explicitly. For example, in the LS3D algorithm, the outliers extracted from shadow and low texture regions will cause false alarms as they will be considered as change.

### 4.2.3. Projection-based geometric change detection

Instead of extracting point clouds from stereo images, the projection-based geometric change approach projects existing 3D information from LiDAR data to stereo images to find corresponding pixels. The color similarity of the corresponding pixels is used for validating the 3D information (Qin et al., 2016). The advantage of this approach is that no correspondences need to be identified from stereo images, especially in shadow and low texture areas. However, finding accurate corresponding pixels from stereo images for each sparse LiDAR point is not easy. As terrain relief causes relief displacement and occlusions in airborne images (Habib et al.,

2007), several LiDAR points can get projected to the same image pixel. A DSM interpolated from sparse LiDAR points is often required to determine which LiDAR point can be seen from the camera. Given a DSM and known intrinsic and extrinsic camera parameters, true ortho-rectification is applied to rectify displaced pixels to their correct place. If the given height is correct, corresponding pixels from different ortho-photos should have similar color. However, the quality of the ortho-photos relies on the quality of the DSM used in the ortho-rectification process. Instead of making true ortho-photos, dense matching is applied to compensate for misalignment of data sources in Qin (2014), to better find the correct corresponding pixels from a stereo pair. The DSM converted from an existing and accurate 3D model is transformed to a disparity image which constrains the disparity search space (DSS) by only allowing a small search range to guide semi-global matching (Hirschmuller, 2008) for compensation of misalignment between two data sources. However, the method strongly relies on the accuracy of the 3D model (Qin et al., 2016). If a DSM is obtained by interpolation from LiDAR data, the edge problem of the DSM will directly affect the quality of change detection. The method described in Sinha et al. (2014) uses planar surfaces from sparse points extracted from a feature detector to provide guidance for dense matching. Every image pixel receives not only one, but several candidate disparities, estimated from candidate planes the pixel may belong to, limit the DSS for improving the quality of dense matching in problematic areas.

## 4.3. Methodology

Our LEAD-Matching densifies height information to create candidate DSMs from planar surfaces extracted from sparse LiDAR points. Three candidate heights estimated from three adjacent planes are generated for each DSM pixel. The candidate DSMs are transformed to candidate disparities to guide dense matching. This approach avoids searching for correspondences from stereo images, especially in shadow and low texture areas, but also uses image information to determine heights for LiDAR data in uncertain areas, especially near edges. By addressing these quality problems, or data limitations, in both data sources explicitly, accurate partial changes are detected. A second dense matching step is performed to complete changes and update 3D information simultaneously. The workflow uses a two-step dense matching approach as shown in Figure 4.1, followed by a post-processing step to filter irrelevant changes caused by trees, cars and other small objects.

### 4.3.1. Partial change detection—LiDAR-guided dense matching

As shown in Figure 4.1a, the proposed LiDAR guided edge-aware dense matching step consists of four parts as follows.

**Generation of candidate DSMs**

As described in pseudo-code in Algorithm 1, the planar patches are first segmented from sparse LiDAR points from a successful plane growing algorithm proposed in

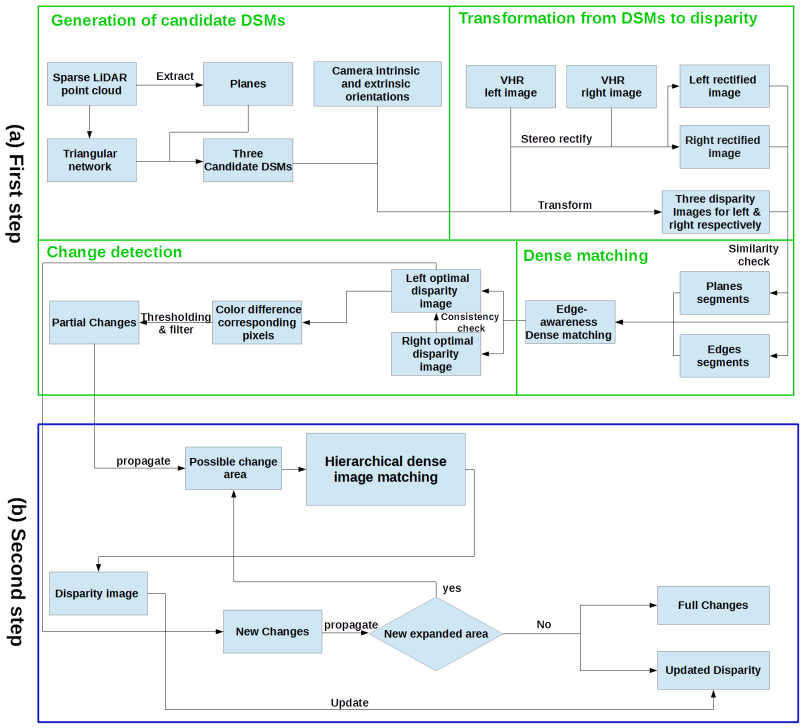



Figure 4.1: The workflow of the two-step dense matching algorithm to detect changes and update of LiDAR data using a stereo pair.

Vosselman et al. (2004). The planar information is associated to the sparse LiDAR points. In this step, we aim to find the adjacent planes for each DSM pixel by searching the neighboring LiDAR points. However, the sparsity and mixed return problems of LiDAR data make it sometimes difficult to ensure that the correct plane is present among the adjacent planes, especially in the edge areas. To enforce that the correct plane is almost always present, the two problems, sparsity and mixed return, are elaborated and addressed as follows.

Due to the sparsity problem, the adjacent points selected for the DSM pixel on the ground may all lay on the roof, when the several nearest LiDAR points in 2D are selected as shown in Figure 5.2b. A 2D Delaunay triangulation addresses this sparsity problem by defining adjacent LiDAR points for each pixel as the three vertices of the 2D Delaunay triangle in which the pixel is located. Therefore, the triangle allows a maximum of three different planar segments found for each pixel, which often provides enough candidate planes for DSM pixels near edges. Indeed, most DSM pixels near building edges are adjacent to at most three planes, corresponding to roof, wall and ground planes respectively. Note that classification of the planes is not necessary. The probability of a pixel belonging to a candidate plane is estimated according to the distance of the corresponding point of the plane

**Algorithm 1** Generation of Candidate DSMs from LiDAR Data

**INPUT:** *Pcs* - LiDAR point cloud;
**OUTPUT:** *vDs* - Vector of three DSMs;

```
1: procedure DSMGeneration(Pcs)
2:     Pls ← RegionGrow (Pcs)      ▷ Use Region Growing for plane segmentation
3:     Tris ← Delaunay (Pcs)       ▷ Obtain 2D Delaunay triangulation of LiDAR
4:     Plim ← Interppl (Tris,Pls)  ▷ Assign three planes for each DSM pixel
       using Delaunay triangles
5:     Edgesegments ← Segmentation (Plim)  ▷ Use plane id in each pixel to get
       edge segments
6:     Edgesegments ← Fillholes (Edgesegments)  ▷ Fill holes in edge segments
7:     for each pixel p in Edgesegments do
8:         Plim ← Update(Plim,window,p)  ▷ Update planes for edge pixel
9:     end for
10:    vDs ← CreateDSMs (Plim)  ▷ Create DSMs from three planes in each
       pixel
11: end procedure
```

to the pixel. These steps correspond to lines 2-4 in Algorithm 1.

Due to the mixed return problem of LiDAR points, it is still possible that a DSM pixel is not linked to its correct plane as shown in Figure 5.2c and d. Two DSM pixels on the roof shown in Figure 5.2c and d, do not have the roof plane as a candidate. In Figure 5.2c, the DSM pixel has adjacent vertices of two colors, purple and red, corresponding to wall and ground segments respectively. As this situation only happens near edges, edge areas are identified before addressing this problem. If a DSM pixel is located in a triangle with all three vertices assigned to the same plane, the pixel is labeled as planar, otherwise it is labeled as edge pixel. However, the DSM pixel in Figure 5.2d is located near an edge but still marked as a planar pixel with all three pixels assigned to the wall plane. This effect results in many holes near edges. The size of such holes is often less than the average GSD of the LiDAR points. Typically, LiDAR data comes with a specification of its point density as X points per square meters (ppm). The GSD is approximated by $\frac{1}{\sqrt{X}}$. A closing filter of the GSD size is applied to fill such holes. In these edge areas, a window with the same size is applied to every edge pixel to get robust candidate planes from neighboring pixels. The probabilities of the pixel belonging to each candidate plane are used to choose and order the top three plane candidates. Finally, three heights for each DSM pixel are estimated from the three candidate planes. Therefore, three candidate DSMs are derived. Planar pixels are simply assigned three times the same height while edge pixels have a maximum of three different heights. These steps correspond to lines 5-9 in Algorithm 1.

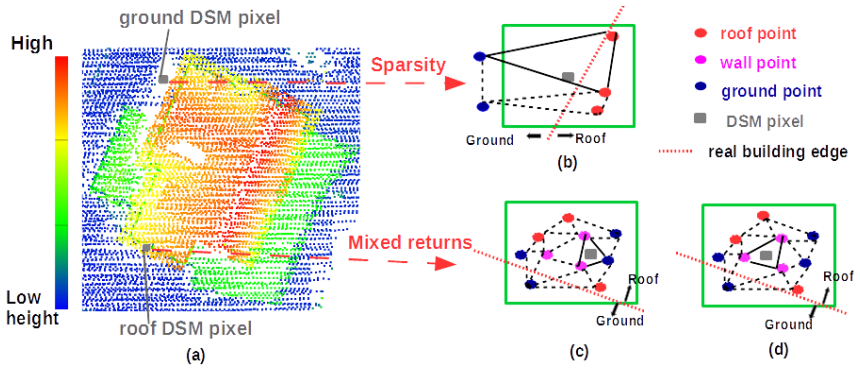



Figure 4.2: (a) Sparse LiDAR point clouds with mixed points, shown in different colors, near building edge areas. (b) Sparse LiDAR points around a DSM pixel. The black triangle defines the points adjacent to the pixel. (c) The DSM pixel, on the roof, in a triangle does not get a height from the roof plane as a candidate. The vertices of the triangle only include wall and ground points. (d) The DSM pixel, on the roof, in a triangle only get heights from wall planes as candidates.

### Transformation from DSM to disparity

In this part, the candidate DSMs created from LiDAR are transformed to candidate disparities to guide dense matching by limiting the disparity search space for a stereo pair, which consists of a left and right image, represented as $I_l$ and $I_r$ respectively. First, the perspective projection, using camera intrinsic and extrinsic parameters of a stereo pair, is applied to project each DSM pixel, in a form of a 3D point, to left and right image respectively to find the corresponding pixels. As described above, several DSM pixels can be projected to the same image pixel. A depth image of the DSM from the optical center of the left and right image respectively is created to select the correct DSM pixel, i.e. the pixel that is actually seen by the image. Second, stereo rectification using camera intrinsic and extrinsic parameters (Fusiello et al., 2000) is applied to rectify the stereo pair in order to align the corresponding pixels to the same rows. Therefore, the positions of the corresponding pixels are simplified to a single value, called disparity, which is obtained by simply subtracting the column indices of the corresponding pixels in the rectified images. Third, as adjacent DSM pixels may be projected to the same image pixel or to image pixels that are not adjacent, resulting in salt and pepper effects, a 3 by 3 median filter is applied to clean the disparity values. Many facade pixels in images don't get disparity values as DSMs do not contain accurate facade heights. They are set to a very small negative constant to indicate occlusions. Finally, all three candidate DSMs are transformed using the three steps above to candidate disparity images for $I_l$ and $I_r$ respectively. This means that both left and right images are applied with dense matching, as the consistency check is applied latter to detect occlusions.

The guidance of LiDAR data is set by using the three disparities to limit the disparity search space (DSS) for each image pixel. A small range of $[-T, +T]$ is set

to each disparities for adding flexibility to compensate for possible misalignment between data sources. The pixels in the rectified image are easily classified as edge or plane pixels by simply checking whether three disparities are the same. The image is then split into planar, edge and occlusion segments by connected component labeling. The DSS consists of $2 * T + 1$ candidate disparities for each planar pixel, and consists of $3 * (2 * T) + 1 = 6 * T + 3$ candidate disparities for each edge pixel. Each segment, except occlusion segments, is used for dense matching independently. As each segment is used for dense matching, the less computation and storage is needed.

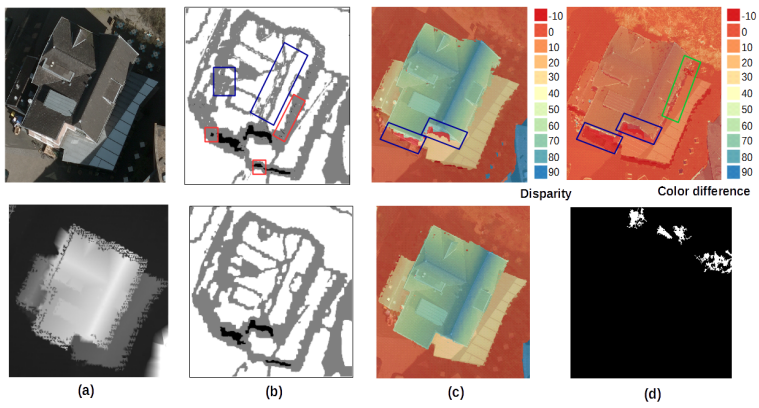



Figure 4.3: LiDAR-guided dense matching applied to a building without changes between LiDAR and image acquisition. (a) The building, shown in the image (top) and LiDAR data (bottom), did not change. (b) Top: many small isolated segments obtained after initial segmentation in blue and red boxes. Bottom: the segments after filtering small isolated segments for dense matching. (c) Optimized disparity for left image (top) and right image (bottom). (d) Top: the color difference of corresponding pixels from dense matching with occlusions indicated. Some changes, like speckle, are left and indicated in the green box. Bottom: changes detected after thresholding and filtering. Indeed, no change is detected in the building area.

Even though each disparity image has been cleaned by a median filter, still segments from candidate DSMs are not transformed exactly to disparity images. The segments in the boxes in Figure 4.3b are small and isolated and should be cleaned with the corresponding disparities updated. In the Figure, white, gray and black pixels represent planar, edge and occlusion pixels. Several steps are applied to remove small and isolated edge segments, and fill small holes as shown in the blue boxes in Figure 4.3b. A opening filter with a size of 3 is applied to remove isolated pixels. Also edge segments with area below $1m^2$ are removed. Another closing filter of a bigger size, 7, is applied to fill holes in the edge segments to better fill holes in the edges while reducing expansion effects. The same process is applied to remove small and isolated occlusion segments as shown in the red boxes in Figure 4.3b. The result is shown in Figure 4.3b bottom. The candidate disparity of pixels is updated according to the class changes introduced by the cleaning steps. For new planar pixels, a corresponding $3 \times 3$ window is applied to

select the top one disparity as the candidate. For new edge pixels, a corresponding $7 \times 7$ window is applied to select the top three disparities as candidates.

**Dense matching**

Dense matching is often defined as a maximum a posterior Markov Random Field (MAP-MRF) problem, which combines with global optimization to estimate optimal disparity for each pixel (Sinha et al., 2014). The MRF formulation aims to estimate optimal disparity by considering two aspects: (1) color similarity of corresponding image pixels, and (2) disparity similarity among neighbors. The second aspect improves the estimation especially in low texture or shadow areas. An equivalent framework is energy minimization as used in Hirschmuller (2008). A MRF is defined by an underlying graph $G = (V, E)$ where each vertex $v \in V$ denotes a pixel in the disparity image, and each undirected edge $e \in E$ connects the pixel to one of its neighbors $W \in V$. The disparity of pixels is given by $\mathbf{d} = \{d_i\}_{i \in V}$, while disparity probability depends on the color information from two rectified images, represented by $\mathbf{I} = \{I_l, I_r\}$. $c_{p_i}$ and $c_{p'_i}$ store the color information of corresponding pixels from the two images given $d_i$. A MRF assumes that $d_i$ obeys the Markov property which states that probability is also conditional on its neighbors $W_i$ (Kumar and Hebert, 2006). The posterior probability over $\mathbf{d}$, given two rectified images $\mathbf{I}$ is given as:

$$P(\mathbf{d} \mid \mathbf{I}) = \frac{1}{Z}\Big(\prod_{i \in V} \varphi(c_{p_i}, c_{p'_i}, d_i) \cdot \prod_{i \in V} \prod_{j \in W_i} \psi(d_i, d_j)\Big), \tag{4.1}$$

where Z is a normalization constant known as partition function, independent from disparity. $\varphi$ and $\psi$ are association (AP) and interaction potentials (IP), respectively. The two terms are defined as follows.

The association potential is a local term which encodes the link between a given disparity value and color similarity of the resulting corresponding pixels. The association potential , $\varphi$, is modeled as an exponential function (Niemeyer et al., 2014) of the posterior probability for $d_i$ given $c_{p_i}$ and $c_{p'_i}$ :

$$\varphi(c_{p_i}, c_{p'_i}, d_i) = \exp(P(d_i | c_{p_i}, c_{p'_i})). \tag{4.2}$$

The probability of disparity given color information of the corresponding pixels is defined as the normalized cross-correlation (NCC) score in $[-1, 1]$ between two $3 \times 3$ matrices constructed from the intensity of $3 \times 3$ image patches centered at each corresponding pixel. As negative correlation does not give more confidence on the difference between two image patches, the NCC is restricted to $[0, 1]$ using a max function, $max(0, \mathrm{NCC}(c_{p_i}, c_{p'_i}))$. Shadows are often casted near building edges. Textures in shadows are seriously affected, so the NCC probability is not reliable. Another way of LiDAR guidance is applied when pixel intensity is low. We then define:

$$P(d_i | c_{p_i}, c_{p'_i}) = \begin{cases} \max(0, \mathrm{NCC}(c_{p_i}, c_{p'_i})) + \epsilon \cdot \omega(\bar{c}_{p_i}), & \text{if } \mathrm{Ord}(d_i) = 1 \\ \max(0, \mathrm{NCC}(c_{p_i}, c_{p'_i})), & \text{otherwise.} \end{cases} \tag{4.3}$$

In case the first ranked disparity is considered, as indicated by the condition $\mathrm{Ord}(d_i) = 1$, the term $\epsilon \cdot \omega(\overline{c}_{p_i})$ adds guidance from the LiDAR data. As described above, the probabilities of the candidate heights given to each DSM pixel are estimated. When the heights are transformed to disparities, disparity values are ranked in descending order based on these probabilities. The candidate disparity ranked first is the one with highest probability according to the LiDAR data. $\epsilon$ is a constant to add guidance from the LiDAR data when a pixel is affected by shadow. The logistic function $\omega(\overline{c}_{p_i})$ defines the probability of a pixel affected by shadow and depends on the average intensity of the $3 \times 3$ image patches $\overline{c}_{p_i}$ from which the disparity is extracted:

$$\omega(\overline{c}_{p_i}) = \frac{1}{1 + a(\exp^{\frac{\overline{c}_{p_i} - b}{\theta}})}, \tag{4.4}$$

where $a, b$ and $\theta$ are parameters defining the shifting and decay speed of the possibility function. The logistic function is defined to let the LiDAR guidance decrease when the intensity increases. In this way, the LiDAR information is incorporated in designing the association potential, especially for improving dense matching in shadow areas..

The interaction potential is a term preferring similar disparity in neighborhoods. The interaction potential, $\psi$, is modeled as an exponential function (Niemeyer et al., 2014) of the joint probability of $(d_i, d_j)$:

$$\psi_{i,j}(d_i, d_j) = \exp(P(d_i, d_j | e_{ij})). \tag{4.5}$$

The interaction potential prefers disparities of neighboring pixels to be similar. However, in edge areas, disparities are more likely to be different. Therefore, an edge-aware IP is designed conditionally on edge features $e_{ij}$ from colors differences of neighboring pixels. The probability $P(d_i, d_j | e_{ij})$ conditional on $e_{ij}$ is defined as:

$$P(d_i, d_j | e_{ij}) = \begin{cases} 1 & \text{if } d_i = d_j \\ \max(0.7, \omega(\Delta g_{ij})) & \text{if } |d_i - d_j| = 1 \\ 1 - \omega(\Delta g_{ij}) & \text{otherwise.} \end{cases} \tag{4.6}$$

where $\Delta g_{ij}$ is an edge feature representing the absolute difference of the gradient of pixel i and its neighbor j. $\omega(\Delta g_{ij})$ is a logistic function similar to Eg.(5.3). Therefore, $1 - \omega(\Delta g_{ij})$ defines that if $\Delta g_{ij}$ is high when an edge probably exists between pixel i and j, the probability will be high when the difference of disparity between pixel i and j is larger than 1. The AP and IP are set to the equal weights. The optimization algorithm, alpha expansion (Boykov et al., 2001), is selected due to its ability of global optimization with a good efficiency.

**Partial change detection**

After dense matching, an optimal disparity is selected for each pixel as shown in Figure 4.3c. The example indicates that LiDAR-guided dense matching provides effective results, especially near building edges. However, many pixels in one image,

especially facade pixels, are occluded in the other image as shown in the blue box in Figure 4.3c. These disparity images are applied for consistency checking (Hirschmuller, 2008). Each pixel $p$ in the left image uses its disparity to find its corresponding pixel $p^{'}$ in the right image. If the disparity difference between $p$ and $p^{'}$ is larger than 2 pixels, $p$ is labeled as occlusion pixel, as shown in blue boxes in Figure 4.3d. Changes are detected by subtracting the color difference between $p$ and $p'$, which is defined as the root mean square of the differences in RGB value of corresponding pixels. A threshold is set to get a binary change image.

Then, an important step is to remove irrelevant changes or speckle as shown in the green boxes in Figure 4.3d and Figure 4.5b and c. Our goal is to find all height changes of at least 2 m. The relief displacement in pixel unit corresponding to 2 m of height above the ground defines the size of the filter. If camera is nearly nadir, a quick estimation of the displacement can be estimated from Figure 4.4a assuming the stereo pairs are acquired in stereo-rectified position ('normal case') without considering the stereo-rectification. An accurate estimation of the displacement of 2 m height above the ground is performed later to confirm this assumption. If the average ground height is known, the linkage between ground disparity $d_g$ and average ground height $h_g$ is a function of the baseline $B$ between two images, focal length $f$ in the unit of pixels and flight height $H$ (Rothermel et al., 2012): $H - h_g = \frac{B \cdot f}{d_g}$. The relation between relief displacement $\Delta d$ and height with respect to ground $\Delta h$ is:

$$\Delta d = B \cdot f(\frac{1}{H - h_g - \Delta h} - \frac{1}{H - h_g}). \tag{4.7}$$

An example from our research data provides the graph of this relation in Figure 4.4b which shows that 2 m of height change from the ground corresponds to a displacement of around 14.2 pixels approximately. Accurate displacements using the same data are estimated by considering the stereo-rectifications as shown in Figure 4.4c. The figure shows the displacement of all pixels with a height of 2 m above the average ground level in an area of 512 m by 240 m. All displacements are around 14.5 pixels which only differs 0.3 pixel from the quick estimation. When the size of displacement is estimated, connected component labelling is applied to group components and an erosion filter with a size slightly smaller than displacement size, e.g. 10 pixels, is then applied to remove small components. After this filtering, the unchanged building is verified effectively as no changes are detected in the building area as shown in Figure 4.3d.

However, in homogeneous areas, corresponding pixels found using LiDAR data would still have similar colors. In Figure 4.5a and b, when many small buildings with homogeneous roof colors are newly built,the detected changes only occur near building boundaries. The same problem happens for shadow areas. In Figure 4.5c and d, when a new building stands in the shadow. only partial change is detected. More difficultly, if a building would have been completely removed and replaced by homogeneous ground, even no partial change would be found by our method. This problem is solved by using the ground disparity in buildings surrounding to find corresponding pixels for the pixels in building areas indicated by LiDAR data.

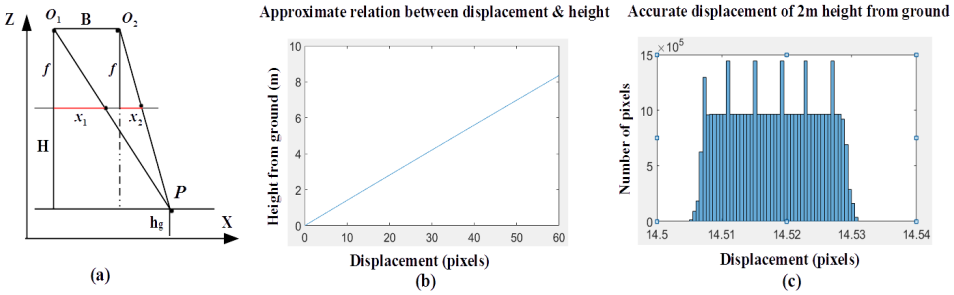



Figure 4.4: (a) Geometric explanation of the relation between disparity and height in 'normal case'. (b) Relation between the approximate displacement from the function and height above the ground. (c) The accurate displacement calculated for 2 m heights above the ground.

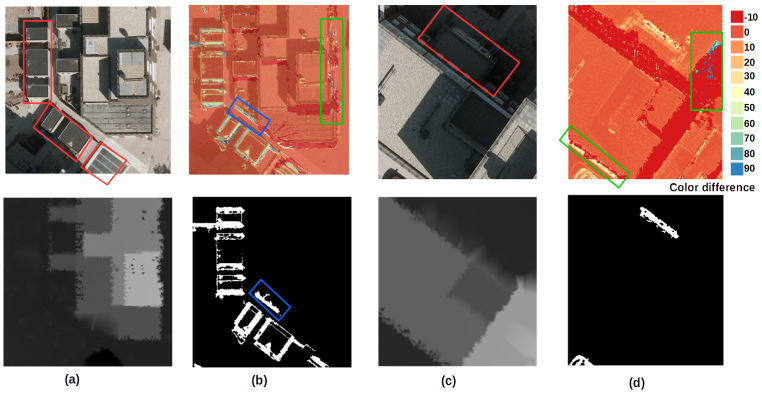



Figure 4.5: (a) LiDAR-guided dense matching applied to two examples where buildings were built between LiDAR and image acquisition. (b) The color difference (top) and binary partial changes (bottom) shows the effect of homogeneous roofs. Examples of speckle and irrelevant changes from a fence are shown in the green and blue box respectively. (c) One building is newly built between the old LiDAR data (bottom) and new image (top) acquisition. (d) The color difference (top) and binary partial changes (bottom) show the effects of shadow.

If all these corresponding pixels are similar, all pixels in building areas are marked as change.

### 4.3.2. Change propagation and update

#### Hierarchical dense matching

The second step of dense matching is to iteratively propagate the partial changes. The disparity image estimated in the first LiDAR-guided dense matching step represents the state of the geometric information in the LiDAR data. The second step aims to estimate disparities, from images only, in areas propagated from where partial changes were detected. If these disparities estimated from images are different

from the disparities from the first step, changes and updated 3D heights are derived simultaneously. However, the disparity search space (DSS) is large without guidance from LiDAR data, requiring a large computation and storage. A hierarchical dense matching method borrowed from the SURE algorithm (Rothermel et al., 2012) is implemented to estimate disparities to overcome the computation and storage issues. The same MAP-MRF framework as above is applied with the difference that in this case no information from LiDAR data is used. The same alpha expansion algorithm is used for optimization.

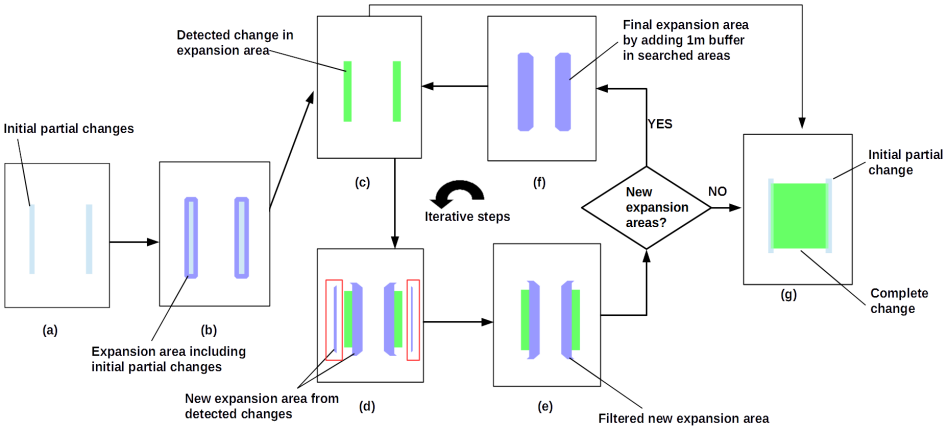



Figure 4.6: Iterative change propagation work flow. (a) Partial changes identified in the first dense matching step. (b) Areas propagated from partial changes. (c) Detected new changes in propagation areas shown in green. (d) Areas propagated from new changes. (e) Removed propagation areas not adjacent to the new changes. (f) Final propagation area for further disparity estimation including the 1 m search areas. (g) Full change propagated from partial changes.

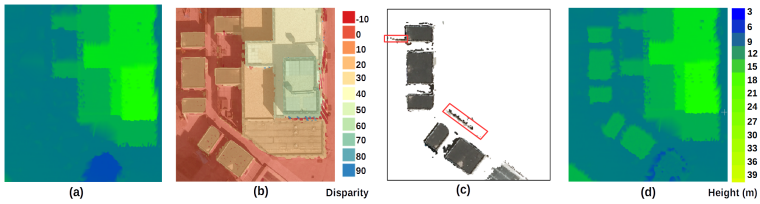



Figure 4.7: Transformation of updated disparity and changes in images to point cloud data. (a) 3D mesh of old LiDAR data. (b) Updated disparities for the new buildings overlaid on the VHR image. (c) 3D points extracted from the updated disparities. A wall connected to a newly built building and a separate fence are detected as new objects shown in a red and blue box respectively and are filtered as being too small. (d) 3D mesh of the updated LiDAR data using 3D points extracted from images after filtering.

**Iterative change propagation**

First, hierarchical dense matching is applied in areas surrounding partial changes. A dilation filter of 1 m is applied on the partial change areas. The reason of choosing this small size of expansion is that many changes detected in first step of dense matching are walls and fences as shown in the blue box in Figure 4.5b. Propagation at a larger size would introduce unnecessary areas for processing. A connected component labeling is applied to the dilated changes to create segments, which are applied with dense matching independently. In Figures 4.6a,b, the propagation is shown for two segments identified in the partial change step, similar to the changes shown in Figure 4.5b (bottom).

When disparities for the left image are estimated, corresponding pixels in the right image are queried. The disparities from the right image are used to verify the disparities extracted from the left image through a similar consistency check as described in Section 4.3.1–Partial change detection. After the check, the disparities of pixels in expansion areas are compared to the disparities resulted from the first step of dense matching to detect 2 m height changes. If the difference is larger than the filter size, e.g. 10, as described in Section 4.3.1–Partial change detection, pixels are marked as changed as shown in Figure 4.6c. The irrelevant change and outliers of with a size below $1 \times 1$ m$^2$ is removed.

A larger propagation area of $2 \times 2$ m$^2$ is applied to create new expansion areas from the detected changes in order to increase the speed of propagation in the iterative steps as shown in Figure 4.6. The new expansion areas are subtracted from the searched areas, which are the expansion areas shown in Figure 4.6b. The remaining new expansion area is shown in Figure 4.6d. If the expansion areas is not adjacent to the detected changes as shown in the red boxes in Figure 4.6d, these areas are removed as shown in Figure 4.6e. Dense matching relies on a smoothness constraint to estimate disparities in shadows and low texture areas. In order to get disparities with smooth transition following estimated disparities in the previous step, the final expansion areas as shown in Figure 4.6f also include 1 m areas in the searched areas. The same hierarchical dense matching is applied to derive new changes iteratively until no more propagated areas are added as shown in Figure 4.6e. Using this approach, complete changed areas are detected as shown in Figure 4.6g , while simultaneously, corresponding disparities are extracted that are used for updating height information.

### 4.3.3. Post-processing

As changes and updates are only obtained in the image domain, these changes and updates should be transformed to the changes and 3D updates on the LiDAR point cloud.. Changes are separated into new objects and (partially) removed objects by evaluating the differences between updated disparity and disparity guided by the LiDAR data. If updated disparity is higher, a newly built object is identified. Otherwise, a removed object is identified. The updated disparity, as shown in Figure 4.7b, is transformed to a 3D point cloud as shown in Figure 4.7c. Then irrelevant changes, such as trees, cars and small walls, should be filtered to obtain final building changes. Therefore, the point cloud is rasterized to a DSM and cleaned

by the following three filtering steps: (1) filter vegetation using NDVI (normalized difference vegetation index). NDVI is calculated from additional near-infrared images with coarse spatial resolution. If the NDVI of a pixel is above 0.1, the pixel is marked as vegetation in a vegetation mask. This mask is used for filtering new objects including many trees and bushes detected as newly built objects. (2) Filter changed objects on roads and rivers. The focus of this chapter is to detect building changes, therefore, changes on roads and in rivers are disregarded. Many cars detected as new objects are parked on the side of the road. An existing vector topographic map with information of streets and rivers is used for this purpose. (3) Walls, fences, as shown in the blue and red boxes in Figure 4.7c, and remaining cars, are filtered based on their size. A filter component step is applied to filter change components with a size below $2 \times 2$ m$^2$. Finally, the LiDAR points in the remaining change areas are replaced by newly extracted 3D points from images. The original and updated 3D mesh are shown in Figures 4.7a and d, respectively.

### 4.3.4. Evaluation

Both pixel-wise and object-wise evaluation is performed. Three metrics are selected: completeness ($Comp$), correctness ($Corr$) and F1 score.

$$\begin{aligned} Comp &= \frac{TP}{TP+FN}, \quad Corr = \frac{TP}{TP+FP}, \\ F1 &= 2 \times \frac{Comp \times Corr}{Comp + Corr}. \end{aligned} \tag{4.8}$$

TP, FP and FN denote the true positives, false positives and false negatives respectively. The completeness and correctness of change detection is denoted by $Comp$ and $Corr$ respectively (Rottensteiner et al., 2014). The F1 score is a quality measurement combining these two metrics. In pixel-based evaluation, $TP$ and $FP$ are calculated based on the intersection of building pixels between detection result and ground truth. In object-based evaluation, the intersection between segmented building objects is calculated. An object is defined to be correctly detected if the detected object has at least 50% area overlap with the ground truth as suggested by (Rottensteiner et al., 2014). TP and FP also denote the number of buildings correctly detected and falsely detected respectively. The ground truth of building changes is represented by GT.

## 4.4. Experiment and discussion

### 4.4.1. Data specification and pre-processing

#### LiDAR data and BAG building map

The study area consists of two urban areas located in Amersfoort and Assen, the Netherlands, covering 512 m by 240 m and 854 m by 386 m respectively. LiDAR data is from AHN2, an open source dataset for the Netherlands which was collected between 2007 and 2012 (PDOK, 2019). The point clouds acquired for Amersfoort and Assen in 2010 and 2012 are shown in Figure 4.8a and Figure 4.11a respectively. The average horiontal and vertical accuracy of the LiDAR data are around 5 cm

and an average of 10 points per square meter (ppm) is available (Van Der Sande et al., 2010). The approximate spatial resolution is around 30 cm. BAG is an open-source Dutch building map. Trees always change between LiDAR and image data acquisition. Tree effects in the LiDAR data are reduced using BAG building polygons, while trees effects on the images are reduced using NDVI. Note that, effectiveness of NDVI depends on the image acquisition season. In general, NDVI is also affected by shadows. In Assen experiment, still, three trees were falsely detected as new buildings .

**Aerial VHR image**
VHR images were taken by the same Microsoft Vexcel's UltraCam-Xp in 2010 and 2018 for Amersfoort and Assen respectively. The bundle adjustment is performed using ground control points to provide a file with camera intrinsic and extrinsic parameters. The back-projection error of using these parameters is less than 1 pixel. The data channels are RGB and the image size per frame is $11310 \times 17310$ for Amerfoort and $8720 \times 13340$ for Assen respectively. The Assen data is pan-sharpened, resulting in a image of $13080 \times 20010$ pixels. The images are taken from near-nadir position and have 60% forward and 40% lateral overlap. For Amersfoort, flight and ground height are around 620m and 5m respectively, and baselines are around 160m and 281m in forward and lateral direction respectively. Combined with the focal length of the camera, 16750 (in unit of pixels), the GSD on the ground is approximately 3.5 cm. For Assen, flight and ground height is around 970 m and 12 m, and the baselines are around 260 m and 700 m in forward and lateral direction respectively. The focal length of the camera is 19326.923 pixels, so the GSD on the ground is approximately 5 cm. Objects of about 2 meter height on the ground have 14 and 11 pixel displacements in two data sources respectively according to Equation 4.7, which is suitable for our proposed method. The stereo pair with 60% overlap is chosen for both areas for change detection and updating. As most of the buildings in the data are not very high, occlusions are limited in the stereo pair. One VHR image in the overlapping area with its stereo pair is shown for Amersfoort and Assen in Figure 4.8b and Figure 4.11b respectively. Another open-source Dutch near-infrared ground ortho-image with a coarse GSD of 25 cm in 2010 and 2018 is obtained for Amerfoort and Assen respectively. The acquisition time is not exactly the same as the VHR images described above, but they perform reasonably good to remove trees.

**Existing and ground truth of updated building point clouds**
In our research, the existing LiDAR data and BAG maps are used to present the status of buildings in the past, while the new VHR images are used to update building point clouds. In the Amersfoort dataset, the LiDAR and image data are obtained in the same year, 2010. There are no significant building changes. Therefore, we simulate changes by using BAG map from 2008. Comparison to the BAG map shows that many buildings are newly built between 2008 and 2010. BAG building polygons are used to clip LiDAR data to create a building point cloud for 2008. In order to create ground truth for validating our method, the changed building polygons are drawn on the LiDAR data in 2010. The minimum

size of changed buildings drawn is $2 \times 2 \times 2$ m$^3$. After the building polygons in 2010 are manually delineated, the original LiDAR data from 2010 are clipped for use as ground truth to evaluate the updated point clouds from our method.

In Assen, the LiDAR and image data are obtained in different years, 2012 and 2018 respectively. An up-to-date BAG building map is available for 2018. Manual editing of the 2018 BAG map is performed to create a 2012 BAG map using the LiDAR data of 2012. Building point clouds for 2012 are created in the same way as for Amersfoort. For Assen, an updated LiDAR point cloud, AHN3, collected between 2014 and 2019 (PDOK, 2019), is available. The ground truth is created using 2018 building polygons to clip AHN3 LiDAR data. The ground truth of building changes for Amersfoort and Assen is shown in Figure 4.8c and Figure 4.11c.

The LiDAR point clouds are rasterized to DSMs for evaluation by interpolating heights from LiDAR points in each polygon. Due to the limited resolution of LiDAR data, the boundaries are not exactly correct. We removed an area of 20 cm buffer around the boundaries from the ground truth. This step is suggested by the ISPRS benchmark evaluation which removed a buffer area of 24 cm to reduce the impact of uncertain borders on the evaluation (ISPRS, 2019).. The programs are written in C++ using opencv, gdal and liblas libraries except for two sections: (1) the surface growing algorithm to find planes from LiDAR data from Twente University (Vosselman et al., 2004); (2) the optimization algorithm for the MRF model, alpha expansion, from the UGM (Schmidt, 2007).

### 4.4.2. Results

**Amersfoort results**

The change detection and update results are shown in Figure 4.8d. As shown in Table 4.1, all 253 unchanged buildings are verified. One removed building has been correctly detected. There are 16 new buildings detected correctly, while 4 new buildings are missed, and 3 objects different from buildings were falsely-detected. The completeness and correctness of the new building detection are 0.80 and 0.84 respectively. The missed buildings are mainly due to shadow and tree effects. For example, in Figure 4.9a, a small new building is correctly detected. However, half of the building is covered by trees. After filtering trees, the size of remaining changes is less than $2 \times 2$ m$^2$. Another new building is barely visible in the image due to shadows and occlusions casted by trees. The reconstructed building points are not complete and the area is also less than $2 \times 2$ m$^2$. Two false-detections were caused by cars in the image. One example is shown in Figure 4.9b. A truck with a size large than $2 \times 2$ m$^2$ is identified as a new building. One false-detection is caused by problems in the LiDAR data. As shown in Figure 4.9c, LiDAR points penetrate through the window on the ground, so the heights from the LiDAR data are lower than the terrain height. The 3D mesh from the LiDAR data shows that the height of the window is lower than the ground. The ground heights are captured by images and an object change is detected as shown in the lower right image. One removed building is correctly detected as shown in Figure 4.9d. The building is shown in the 3D mesh of the LiDAR data in Figure 4.9d left, while the

building is removed in the updated 3D mesh in Figure 4.9d right.

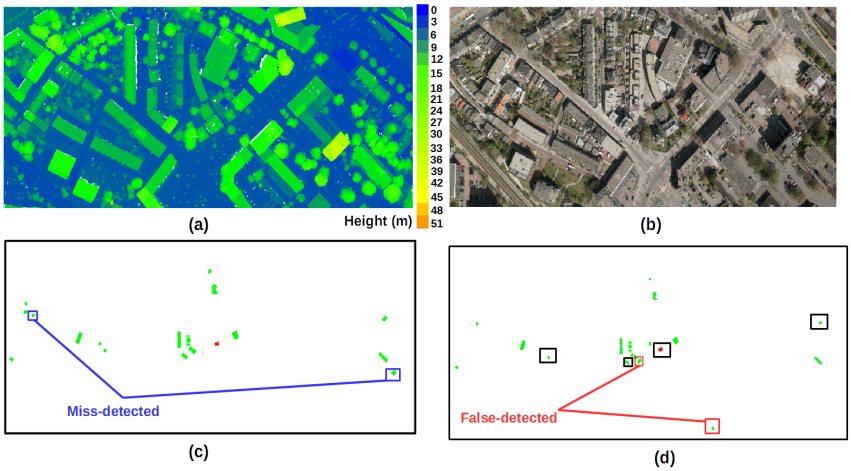



Figure 4.8: Building change detection results in Amersfoort. (a) AHN2 LiDAR data for Amersfoort. (b) VHR image of the same area. (c) Ground truth for building changes detection. Removed buildings are shown in red and new buildings are shown in green. Missed buildings are indicated by Blue boxes. (d) Changes detected from the proposed method. False-detected buildings are indicated by red boxes. Black boxes indicate examples of correctly detected buildings. The details in the boxes are shown in Figure 4.9.

| Pixel-based | *Comp* | *Corr* | F1 | Object-based | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | GT | TP | FP | *Comp* | *Corr* | F1 |
| Building verification | | | | 253 | 253 | | | | |
| New building | 0.85 | 0.86 | 0.86 | 20 | 16 | 3 | 0.80 | 0.84 | 0.82 |
| Removed building | 0.99 | 0.99 | 0.99 | 1 | 1 | 0 | 1 | 1 | 1 |

Table 4.1: Change detection result for Amersfoort.

The proposed method also manages to detect many small buildings. Three newly built buildings are shown in Figure 4.9e. The smallest building found is around $2 \times 2 \times 2$ m$^3$. The examples are given by the upper two buildings in Figure 4.9e. The bottom image shows that the method manages to detect a new building even in the shadow. In this case, partial changes were found in the first step of image matching, while change propagation in the second image matching step is effective by estimating disparities only for the area of interest. Pixel-based evaluation in Table 4.1 shows that the completeness of new building detection is 0.86, while the correctness is similar with a value of 0.86. It indicates that the proposed method manages to keep both completeness and correctness high, resulting in a high F1 score with a value of 0.86. As only one removed building exists in the experiment, this high detection rate with a F1 score of 0.986 is not meaningful. But the effectiveness of the proposed method to detect removed buildings is shown be-

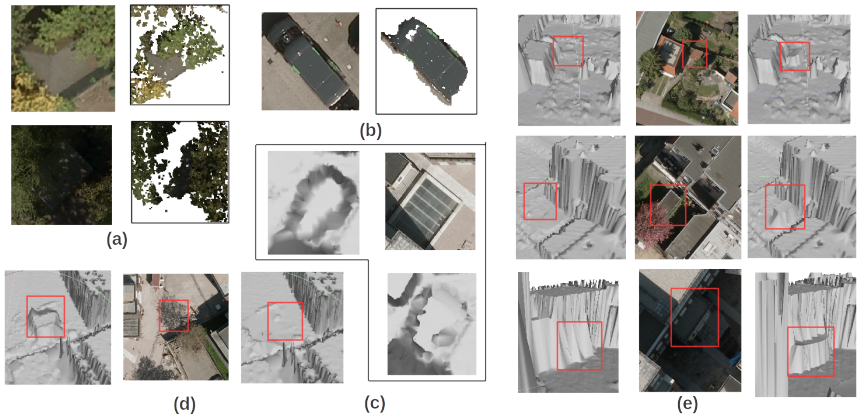



Figure 4.9: Examples of (a) missed, (b, c) falsely detected, and (d, e) correctly detected building changes. (a) Missed buildings due to trees or shadows casted by trees. (b) a truck wrongly detected as a building. (c) An over-detected object due to a ground window. (d) Correctly detected removed buildings. The original 3D mesh, a new VHR image and a updated 3D mesh are shown from left to right respectively. (e) Three examples of correctly detected buildings that are small or in the shadow. The original 3D mesh, new VHR image and updated 3D mesh for three cases are shown from left to right.

low in the Assen experiment. The height difference between updated point clouds from the proposed method and ground truth is calculated from the overlapping area. The average height difference in the changed areas is 0.20 m.

**Amersfoort comparative results**

The detection results are compared qualitatively to two types of methods, a projection-based geometric approach (projection–geometry) (Qin, 2014) and robust DSM subtraction (robust–dDSM) (Tian et al., 2014) as described in Section 4.2.3 and 4.2.2. The results of the three methods on the three cases introduced in Figures 4.3 and 4.5 are compared. In Figure 4.10a, as projection–geometry uses the interpolated LiDAR DSM for guiding dense matching to detect changes, the error in edge areas propagates to the final change detection result as shown in the red boxes. Another problem shown in the blue boxes is on the building facades. As the DSM does not have heights on facades, the corresponding pixels extracted by DSM guidance are wrong. These false alarms with a size similar to the real building changes, as shown in the green boxes, are difficult remove. The proposed method rely on the guidance from three candidate DSMs instead of one interpolated DSM. The false alarms from building edges and facade are largely reduced. In addition, the changes obtained by projection–geometry are not complete due to homogeneous areas.

In order to apply robust–dDSM, a photogrammetric point cloud is constructed from four-view images using the software Pix4D. For each pixel in the DSM of the photogrammetric point cloud, a $7 \times 7$ window is applied to select neighborhood pixels in the LiDAR DSM to calculate the minimum difference. If the height

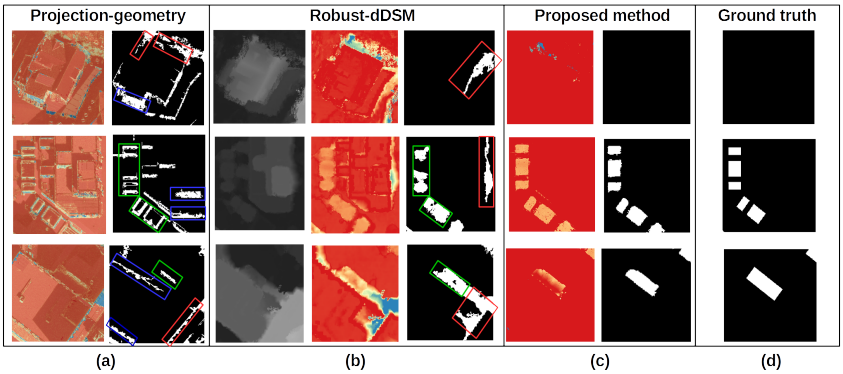



Figure 4.10: Comparison of results from projection–geometry (a), robust–dDSM (b) and proposed method (c) for three cases introduced in Figures 4.3 and 4.5. The ground truth is shown in (d). Top: no change. Middle: multiple small new buildings. Bottom: a new building in the shadow.

difference is more than 2 m, the pixel is labeled as change. Changes are filtered following the filtering steps in Section 3.4. In Figure 4.10b, the poor quality of the edges from DSMs converted from LiDAR and photogrammetric point clouds is mitigated by robust–dDSM. However, the quality problem of photogrammetric point clouds in low texture and shadow regions results in large areas of false changes as shown in the red boxes in Figure 4.10b. These false changes are difficult to separate from real changes in the green boxes. The results of the proposed method in Figure 4.10c shows the proposed LiDAR guidance in dense matching improves 3D information extracted in low quality regions.

**Assen results**

The change detection and updating results are shown in Figure 4.11d. As shown in Table 4.2, all 952 unchanged buildings are verified. There are 163 buildings correctly detected. while 12 buildings are missed, and 9 objects are falsely-detected. The *Comp* and *Corr* of new building detection is 0.93 and 0.95 respectively. As trees adjacent to buildings in Assen data are less, no buildings were removed due to tree as described for the Amersfoort experiment. In the 12 missed buildings, two were missed in the first step of dense matching due to homogeneous areas and shadows, while the other 10 missed buildings were caused by the failure to extract point clouds in change areas due to low textures and shadows in the second step of dense matching. As shown in Figure 4.12a, the quality of the 3D points reconstructed for the new building in the image is poor, due to low texture and repetitive rooftop pattern. From the 9 falsely-detected objects, three objects are high trees and vegetation, which are not removed using NDVI. Another two objects are balcony parasol and pavilion in the park. The pavilion is shown in the upper image of Figure 4.12b. Two big trucks were also false detected. One of them is shown in the lower image of Figure 4.12b. Two objects are piles of sands and stones on a construction site as shown in the Figure 4.13b.

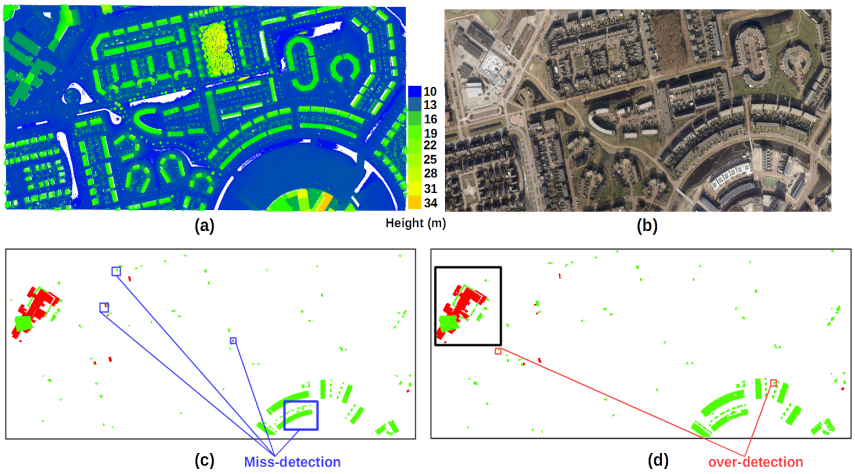



Figure 4.11: Building change detection results in Assen. (a) AHN2 LiDAR data for Assen. (b) VHR image of the same area of the AHN2 LiDAR data. (c) Ground truth of building changes. Removed buildings are shown in red and new buildings are shown in green. Missed new or removed building are indicated in blue boxes. (d) Changes detected from the proposed method. False-detected buildings are indicated in the red boxes. Black boxes indicate examples of correctly detected buildings. The details in the boxes are shown in Figure 4.13.

| Pixel-based | *Comp* | *Corr* | F1 | Object-based | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | GT | TP | FP | *Comp* | *Corr* | F1 |
| Building verification | | | | 952 | 952 | | | | |
| New building | 0.88 | 0.94 | 0.91 | 175 | 163 | 9 | 0.93 | 0.95 | 0.94 |
| Removed building | 0.84 | 0.98 | 0.91 | 11 | 9 | 0 | 0.82 | 1 | 0.90 |

Table 4.2: Quantitative change detection result for Assen.

There are 9 (partially) removed buildings correctly detected, while 2 removed buildings are missed. As shown in the upper image of Figure 4.12c, one small building was removed, however, a car was parked at the location in the VHR image. As the black car has similar height as the small building, the displacement is too small to be detected in the first step of dense matching. Another removed building is correctly detected after two dense matching steps, However, there is a wall near the removed building in the VHR image as shown in the red boxes of Figure 4.12c bottom images. The wall occludes parts of grounds resulting in incomplete change detected as shown in the red boxes in the bottom right image of Figure 4.12c. The updated change is split into two. One, shown in the red box in the bottom right image of Figure 4.12c, is removed when filtering small changes with areas less than $2 \times 2$ m$^2$. Finally, the remaining change detected covers less than 50% of ground truth. Therefore, the removed building is miss-detected.

The proposed method manages to detect many small buildings and performs effectively in complicated change scenarios. In Figure 4.13, a large building was

subject to complicated change. Some parts were rebuilt while other parts were demolished, regarded as removed buildings. Even making the ground truth takes lots of effort for a human operator. In general, the detections matches well with the ground truth. In addition, three correctly detected small changes are shown in the red boxes in Figure 4.13c. The detection results shown in Figure 4.13 proves the good performance of the proposed method. There are still missed detections in several small areas as shown in the blue boxes of Figure 4.13b. The reason is that no points are reconstructed from images in occlusion areas and low texture sand areas.

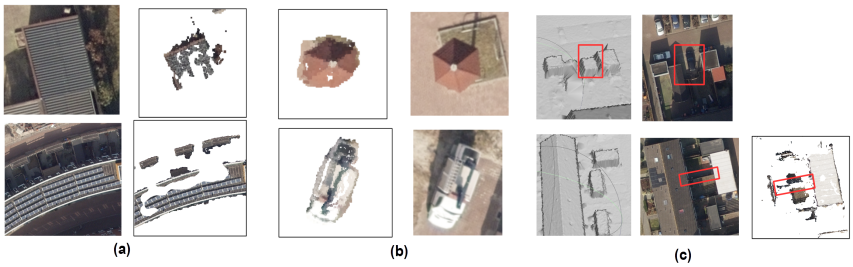



Figure 4.12: Examples of missed and falsely detected building changes. (a) Missed new buildings due to a failure of the second dense matching step for a building with low and repetitive textures. (b) The false-detected objects of a pavilion in a park and a truck are not considered as buildings. (c) Missed removed buildings due to a car and a wall.

Table 4.2 summarizes the pixel-based evaluation. The correctness of new and removed building detection are 0.94 and 0.98 respectively. The completeness of new and removed building detection are a bit lower at 0.88 and 0.84 respectively. The completeness of new building detection is affected by the failure of the second step of dense matching. Both new and removed building detection have a high F1 score with values of 0.91. The average height difference between updated point clouds and ground truth is 0.32 m.

### 4.4.3. Discussion

In this discussion, the main error sources of change detection and updating are first discussed and summarized. Then, parameter settings are discussed. Finally, the contributions are revisited to provide an overall evaluation.

**Error sources**

The reasons for false detection of new buildings are various. The main reason is building-like small objects, such as cars, trees, sand piles and pavilions. One false detection in Amersfoort is due to wrong height measurements of a ground window caused by LiDAR penetration. There are three reasons of miss-detection: (1) buildings are covered by trees; (2) missed detection in the first step of dense matching due to homogeneous areas and shadows; (3) missed detection in the second step of dense matching due to the failure of extracting point clouds in the

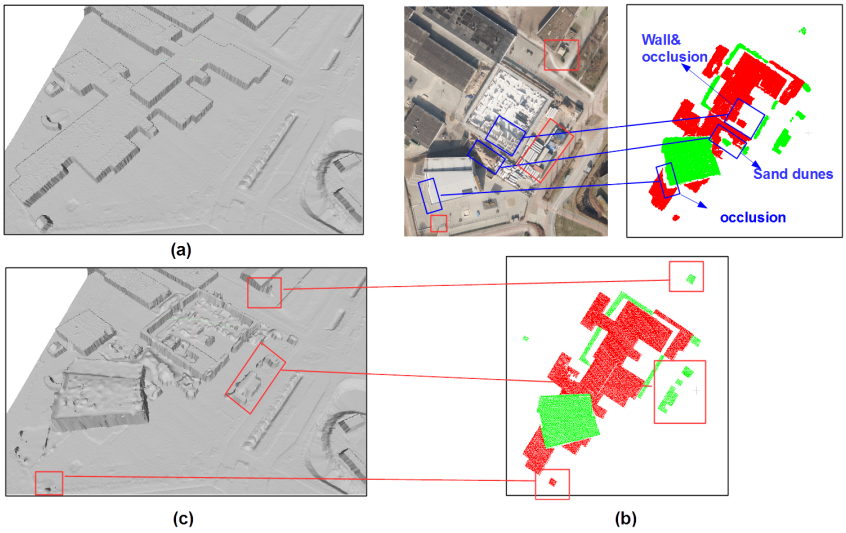



Figure 4.13: Building change detection and updating for a complicated scenario. (a) 3D mesh of 2012 LiDAR data. (b) Top-left: 2018 VHR image used to update LiDAR data. Top-right: changes detected by proposed method. Removed areas are shown in red, while rebuilt areas are shown in green. Blue boxes show missed detection by the proposed method. Bottom: Ground truth for change detection. (c) 3D mesh of updated LiDAR data. The correctly detected new and removed small buildings are shown in red boxes.

areas affected by shadow, low textures and occlusion. Table 4.3 shows the number of false detection and miss detection due to the reasons listed above.

In Rottensteiner et al. (2005), the percentage of overlap, above 50% or 80%, between detected objects and ground truth defines if objects are partially or strongly detected respectively. Table 4.1 and 4.2 show that the proposed method provides effective results on detecting change partially. However, when we test for 80% overlap in the Assen experiment, the number of correctly detected (TP) new buildings and removed buildings drops from 163 to 127 and from 9 to 1 respectively. We expect that the main reason is that only one stereo pair is used for detecting changes. The effects of occlusion, shadow and low texture in one stereo pair can be reduced by adding more images.

| False detection | building-like objects | LiDAR penetration | Missed detection | | |
|---|---|---|---|---|---|
| | | | Trees or occluded | F-DM | S-DM |
| Amersfoort | 2 | 1 | 2 | 0 | 2 |
| Assen | 9 | 0 | 0 | 2 | 10 |

Table 4.3: The table of the false detections and miss detections of new buildings in Amersfoort and Assen. (F-DM: first dense matching step. S-DM: second dense matching step)

**Parameters setting**

Most parameters are involved in the LEAD-Matching step to detect partial changes,which consists of four parts. The parameters in the first two parts, are self-explanatory. In the first part, in order to address the mixed return problem in LiDAR points, a window size is set based on the ground sampling distance (GSD) of the LiDAR points to get better candidates planes for each DSM pixel from its neighborhood. In the second part, in order to create clean segments in the image for dense matching, the filter size for (i) removing small and isolated edge segments, and (ii) filling hole in edge segments are set to 3 and 7 pixels respectively. As the effects considered caused by the transformation between height and disparity, the parameter values can be set the same for different datasets. After candidate disparities are obtained, the disparity range is set to $[-2, 2]$ pixels for reducing the effect of misalignment between two data sources. In Qin (2014), the length of the disparity range is suggested to be 5–10 pixels. As long as the two data sources are georeferenced independently within an accuracy of a few centimeters as described in our data specification, a range length between 5 and 7 pixels is sufficient.

Several parameters in part 3 and 4 are critical. In the association potential of the Markov random field in Equation 5.1, the parameters of the logistic function $\omega(\overline{c}_{p_i})$ and the constant $\epsilon$ to tune LiDAR guidance in shadow areas are important. The parameters of $\omega(\overline{c}_{p_i})$, defining the probability that a pixel is in the shadow, are set according to the shadow intensity in the VHR images. The parameters $a, b$ and $\theta$ are set to 0.1, 40 and 8 respectively based on the observation that the intensity of shadows is often less than 40 while non-shadows often have values of more than 70. With these parameter values, the function defines that if $\overline{c}_{p_i}$ is below 40, $\omega(\overline{c}_{p_i}) > 0.9$, while if it is above 70, $\omega(\overline{c}_{p_i}) < 0.2$. In general, the defined parameter values would work effectively on normal images. If shadow intensity is largely affected by illumination and environment reflections, a shadow detection algorithm (Zhou et al., 2019b), which is able to adapt to different environments, can be applied. We do not require to set the constant $\epsilon$ to solve problem in shadow completely. As long as the LiDAR guidance term reduces the size of wrong disparity estimation in shadows, false alarms will be reduced. The filter designed to remove speckle and irrelevant changes in the fourth part will help to further reduce false alarms. Empirically, $\epsilon$ is set to 0.1–0.2. In addition, in the interaction potential in Equation 4.6, the parameters of the logistic function $\omega(\Delta g_{ij})$, defining the probability that an edge occurs between two pixels, is set based on the gradient difference of the two adjacent pixels. The logistic function is defined with the same $a$ and $\theta$, but with a different shifting parameter $b$, now set to 10. If $\Delta g_{ij}$ is smaller than 10, the probability is above 0.9, while if it is above 40, the probability drops below 0.2. These parameter values are expected to be applicable for other image scenarios as the gradient is less affected by illumination.

In the partial change detection part, the color difference of a same object from a stereo pair, acquired consecutively in a forward path with similar illumination conditions, is small. The threshold for detecting changes allows a small color variation, e.g. 20–30. This value works well for different datasets. Even though such small thresholding value results in more partial changes, these changes will

be verified in the second step of dense matching. Finally, the value for the filter size to remove speckle and irrelevant changes from actual building changes is set adaptively to different datasets according to Equation 4.7.

**Overall evaluation**
Three contributions are revisited and confirmed by experiments. (1) The comparison of our proposed approach with two other change detection methods shows that LiDAR-guided dense matching successfully addresses the quality problems in both data sources. The related false alarms as shown in Figure 4.10a and b are largely reduced as shown in Figure 4.10c. (2) The comparison with the projection–geometry approach also shows that the second step of dense matching is necessary and effective. Otherwise, only partial changes can be derived as shown in Figure 4.10a. (3) Both experiments show that the proposed method successfully verifies unchanged buildings, while detecting minimum building changes of $2 \times 2 \times 2$ m$^3$ with a high rate of success as indicated in Tables 4.1 and 4.2. Examples of accurate change detection were also given in Figure 4.13, while change detection of small objects was shown in Figure 4.9e. These results demonstrate that our method could meet the requirements for updating a large scale 3D map.

## 4.5. Conclusions

This chapter proposes a two-step dense matching framework to detect and update building changes in LiDAR data using new VHR images. The first step proposes LEAD-Matching to derive accurate partial changes by addressing the quality problems from two data sources. Accurate plane information from LiDAR data is exploited to limit the disparity search space for dense matching, especially in the shadow and low texture areas, while dense matching exploited detailed building boundary information in a stereo pair to select optimal disparity. The second dense matching step employs hierarchical dense matching to derive complete changes and update 3D information simultaneously. The proposed method successfully verifies unchanged buildings, while detecting minimum building changes of $2 \times 2 \times 2$ m$^3$ with a F1 score of more than 0.9 in the Assen experiment. This is the first time that it could be shown that the airborne stereo images for 3D change detection and updating can be used to meet the requirements of a large scale 3D map.

# 5

# Improving building extraction using multi-view images

*Accurate building extraction is important for creating large scale 3D city models. The accuracy of building extraction from just LiDAR data or multi-view images is affected by irregular and sparse point spacing, or shadows and low texture areas respectively. This chapter proposes an extended LiDAR-guided edge-aware dense matching (E-LEAD-Matching) to improve the planimetric accuracy of building extraction by integrating accurate plane information of LiDAR data and detailed building boundary information of multi-view images. E-LEAD-Matching starts by applying LiDAR-guided edge-aware dense matching (LEAD-Matching) to integrate LiDAR data with each stereo pair selected from multi-view images. However, the integration result of one stereo pair has typical facade and occlusion problems. In addition, building boundaries may not be fully clear in one stereo pair. Therefore, a probabilistic integration approach is proposed to integrate multiple stereo pairs by quantifying the quality of each stereo pair result to reduce facade and occlusion effects and further improve planimetric accuracy using multi-view building boundary information. The method is evaluated in Vaihingen, Germany and Amersfoort, the Netherlands. In Vaihingen, the planimetric accuracy improves from 0.393 m using LiDAR data alone to 0.215 m by integrating LiDAR data with 6-view images. In Amersfoort, the planimetric accuracy improves from 0.477 m to 0.205 m by integrating 4-view images. A comparative study on integrating increasing numbers of images also shows that the effects of problems as described above are reduced.*

## 5.1. Introduction

Building extraction has been an active topic in remote sensing and computer vision communities for decades. The building extraction accuracy relies on the extraction rate, which defines how well the buildings are differentiated from other objects, and the planimetric accuracy, which defines how well the building boundaries are extracted (Gilani et al., 2016). Many Airborne images with very high resolution provide detailed color information and high planimetric accuracy, which gives an opportunity to extract accurate buildings in a single image. However, the high spatial resolution comes together with color variations between pixels from the

This chapter has been submitted, 2020 (Zhou et al., 2020a)

same class (Cushnie, 1987). Convolutional neural networks (CNNs) are increasingly applied to improve the extraction rate of buildings from VHR images (Yuan, 2018; Zhou et al., 2019a) due to their ability to automatically learn effective complex (con-)textural features from color information. However, training benchmarks for airborne VHR images are limited and have domain transfer problems (Chen et al., 2018). Airborne images are usually acquired through perspective projection. This projection results in relief displacement of high objects (Habib et al., 2007). As a consequence, extracted buildings are in the wrong geographic location with low planimetric accuracy. Dense image matching (Furukawa and Ponce, 2010; **?**) aiming at extracting 3D point clouds from stereo images, also called photogrammetric point clouds, addresses relief displacement by making true ortho images using the 3D information. This 3D information is often incomplete when only one stereo pair is used due to occlusions. Multi-view images are often used to reduce the occlusions. In addition, the 3D information still has low quality in shadow and low texture areas where accurate corresponding pixels are difficult to find (Remondino et al., 2014). In a building related ISPRS benchmark test (Cramer, 2010), many building boundaries has obvious distortions in the provided true ortho images, which directly affects the planimetric accuracy of building extraction. Only if accurate buildings heights are extracted, the high planimetric accuracy of VHR images can be fully explored.

On the other hand, airborne LiDAR point clouds provide direct 3D information with high vertical accuracy which is not affected by shadows and low texture. Geometric features are more suitable to distinguish buildings from other objects than spectral features in VHR images (Haala and Kada, 2010). However, sometimes the appearance of some trees and buildings may be similar in LiDAR data (Rottensteiner et al., 2005). More importantly, the sparse and irregular point spacing of LiDAR data affects the planimetric accuracy of the extracted buildings. Many studies (Sampath and Shan, 2007; Zhou and Neumann, 2012; Awrangjeb, 2016) tried to use regularization techniques to improve the accuracy of building boundaries, but planimetric accuracy still depends on the actual point spacing (Gerke and Xiao, 2014).

Many studies (Rottensteiner et al., 2007; Awrangjeb et al., 2010; Chen et al., 2012) combine geometric features from LiDAR data and spectral features from image data to improve the building extraction rate by eliminating tree effects in LiDAR data, and shadow and occlusion effects in images. However, planimetric accuracy still fails to meet requirements for large scale topographic maps. For example, at a map scale of 1:1000 and 1:2000, the required root mean square error of planimetric accuracy is 0.25 m and 0.50 m respectively (Merchant, 1987). Fortunately, the complementary information on buildings, i.e. high vertical accuracy of LiDAR data and planimetric accuracy of images, can be integrated to improve the planmetric accuracy of building extraction. However, relatively few approaches have been published to optimally integrate two data sources (Awrangjeb et al., 2013). This is mainly due to the sparsity and irregularity of LiDAR data, and occlusions, shadows and low textures problems of VHR stereo images.

Some studies (Gilani et al., 2016; Chen et al., 2005) extract buildings from

sparse LiDAR data and use the initial building boundaries to help to select lines from a single true ortho-photo created by DSMs from photogrammetric point clouds. As mentioned above, the ortho images will be affected by in shadow and low texture areas. Instead of using true othophotos, stereo images are used. In Habib et al. (2010), LiDAR data is first segmented to 3D planes which are projected to a stereo pair. Each image line from the left and right raw images is linked to a plane to create 3D lines. These 3D lines along the LiDAR boundaries from left and right images are matched to reduce falsely detected boundary lines. The extracted boundary lines have high planimetric accuracy, but many boundary lines are missing. In addition, it is difficult to assign the correct plane to each line and lines may be occluded in either left or right images. In general, using LiDAR data to select image lines to present accurate building boundaries suffers from the falsely detected and missed lines in the images. In Habib et al. (2011), image lines from multi-view images are used implicitly for generating the final building boundaries. Each building rectangle extracted from LiDAR data is projected to multi-view raw images, while the building lines in multi-view images are used simultaneously to adjust rectangle parameters. This approach is less affected by falsely detected and missed image lines. However, the buildings are restricted to be composed by rectangles. Another approach is to densify sparse LiDAR points to extend buildings to match accurate boundaries in image segments extracted from a raw image (Gerke and Xiao, 2014). However, only a single image is used and the method relies on the quality of image segmentation, which is often problematic due to spectral variability in VHR images.

In chapter 3, we proposed a novel LEAD-Matching method to integrate LiDAR data and a single stereo pair for improving change detection. First, LiDAR sparse points are densified from plane information in the form of a DSM. Every DSM pixel finds up-to three adjacent planes, where it is assumed that at least one plane is correct. The three candidate DSMs (C-DSMs) are generated to limit the disparity search space to improve dense matching especially in shadow and low texture areas, while detailed building boundaries in images are used to identify optimal heights. By addressing the sparsity problem of LiDAR data and shadow and low texture problems in a stereo pair, false alarms are reduced such that accurate changes could be detected. However, when only one stereo pair was used, integrated building boundaries still suffer from facade, occlusion and unclear building boundary problems as will be elaborated in the methodology section. These problems should be addressed to meet the requirements for large scale topographic maps.

This chapter proposes an extended LiDAR-guided edge-aware dense matching (E-LEAD-Matching) method to integrate LiDAR data with multi-view images, but now with the goal to improve the planimetric accuracy of building extraction. The idea is to densify sparse LiDAR points using plane information to extend building boundaries to match the detailed multi-view building boundaries. Figure 5.1 illustrates the three main steps for E-LEAD-Matching. Firstly, LEAD-Matching is applied to integrate LiDAR data with each single stereo pair selected from multi-view images. Then, an integrated DSM (iDSM) and a true ortho image is

extracted from the dense matching result on each stereo pair. Finally, a probabilistic approach to integrate these iDSMs from multiple stereo pairs by considering their relative quality is provided to reduce occlusion and facade problems in stereo pairs, and to improve planimetric accuracy using detailed boundary information in multi-view images. Note that, boundary simplification and regularization is out of scope of our research. The contribution of this work consist of three main points:

1) To our knowledge, we are the first to propose the E-LEAD-Matching method to improve the planimetric accuracy of building extraction by integrating detailed building boundaries of high planimetric accuracy from multi-view images and plane information of high vertical accuracy from LiDAR data.

2) We propose a new probabilistic approach to integrate multiple stereo pairs which solves problems on facades, occlusions and unclear building boundaries that occur when using a single stereo pair.

3) We show the potential of using LiDAR data and multi-view images for extracting buildings automatically to meet requirements of large scale topographic maps.

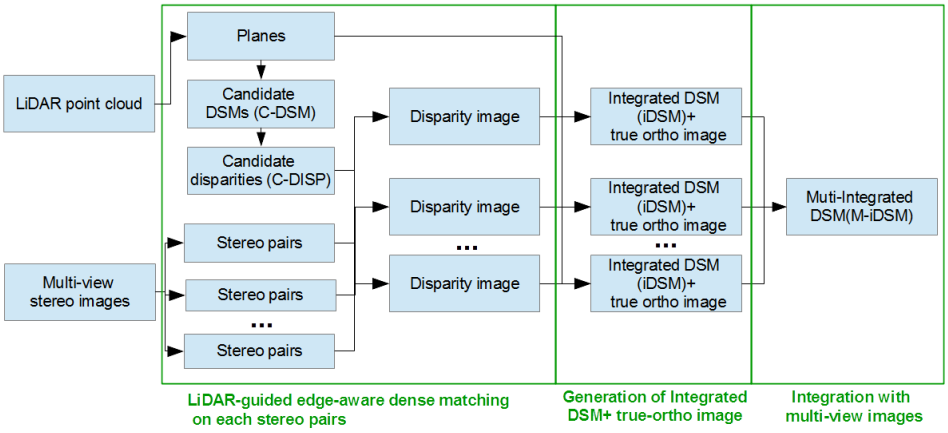



Figure 5.1: The three-step integration framework of LiDAR point clouds and multi-view stereo images.

## 5.2. Study area and materials

The method is demonstrated on two urban areas located in Vaihingen (Germany) and Amersfoort (the Netherlands). Both LiDAR and image data from Vaihingen is provided by the ISPRS benchmark, which was originally produced for a DGPF camera test (Cramer, 2010). The airborne LiDAR and image data in Vaihingen were acquired with one month difference in July and August 2010, respectively. The LiDAR data was acquired by Leica Geosystems using a Leica ALS60 system

with a median point density of 6.7 points/$m^2$. The images were acquired using an Intergraph / ZI DMC camera system by the company RWE Power. The ground sampling distance (GSD) of the images is 8cm. The images are geo-referenced by bundle block adjustment with accurate camera extrinsic and intrinsic parameters provided. The camera is fixed nearly in nadir position with less than 1 degree deviation. The vertical and horizontal viewing angles are 69.30° and 49.01°. Both forward and side overlap are 60%, such that most areas are covered by 4 image and some areas are covered by 6 images. Area 1 in the benchmark, located in the city centre of Vaihingen, is used in our experiment. It consists of buildings of various size and complex shape, while the whole area is covered by 6 images. The three experiments on 2, 4 and 6 images are performed using the proposed method. This allows us to show the relation between planimetric accuracy of buildings, the number of images and their relative viewing geometry. The area size is 150m×200m.

LiDAR data from Amersfoort is obtained from AHN2, an open source dataset for the Netherlands (PDOK, 2019), while the image data is provided by Neo, Netherlands Geomatics & Earth Observation B.V. The images were acquired in April, 2010. The LiDAR data were acquired in spring, 2010. Although the exact date of LiDAR acquisition is unknown, comparison shows that differences are very limited. The vertical accuracy of the LiDAR data is 5cm and the average density is 10 points per square meter (ppm) (Van Der Sande et al., 2010). The images were taken by the Microsoft Vexcel's UltraCam-Xp camera system. The image GSD is 3.5cm. The images are geo-referenced by bundle block adjustment with accurate camera extrinsic and intrinsic parameters provided. The camera is fixed nearly in nadir position with less than 1 degree deviation. The vertical and horizontal viewing angles are 37.31° and 54.65°. The forward and side overlap are 60% and 30% respectively. This means that almost 40% of the area of interest is covered by 4 images, but areas covered by 6 images are rare. Therefore, two experiments on 2 and 4 images are performed using the proposed method. The area size is 320m×190m.

## 5.3. Methodology

The E-LEAD-Matching consists of three steps as shown in Figure 5.1. In addition, an effective approach is proposed to combining reference building polygons with E-LEAD-Matching to extract buildings to evaluate their planimetric accuracy.

### 5.3.1. LiDAR-guided edge-aware dense matching

LEAD-Matching, in Chapter 4, starts from densifying sparse LiDAR data to generate three candidate DSMs (C-DSMs). As shown in Figure 5.2a, if the sparse LiDAR points can be densified accurately in a top view in the form of a DSM, the LiDAR building boundaries can be successfully extended to match the actual building boundaries. As plane information can be accurately estimated in LiDAR data, accurate densification relies on assigning the correct plane to each DSM pixel. Finding the exact correct plane for each DSM pixel is difficult, especially

near building boundaries as shown in Figure 5.2. It is , however, feasible to select several candidate planes from adjacent points. Therefore, LiDAR points are first segmented into planes, and next triangulated with plane membership assigned to each vertex. Then, three adjacent planes are identified from the three vertices of the 2D Delaunay triangle the pixel locates in. For example, in Figure 5.2b, the grey roof DSM pixel is located in a triangle whose three vertices belong to ground, wall and roof planes respectively. Finally, three heights for each DSM pixel are estimated from the three candidate planes such that three candidate DSMs are created. The three heights are ordered in a descending order based on the distance between the pixel and the corresponding point of the plane.

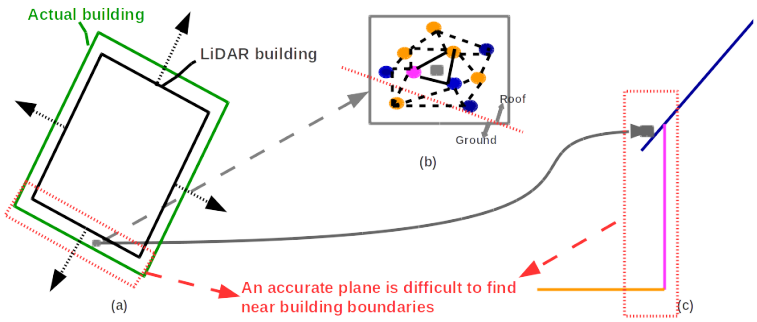



Figure 5.2: (a) The LiDAR derived building boundaries (top-view) can be extended to the actual building boundaries by densifying the LiDAR points in top view, which is often a DSM. (b) Roof (gray) pixel in a triangle with LiDAR points from different planes. The pixel is surrounded by LiDAR points from different planes. The black Delaunay triangle identifies three points adjacent to the pixel, belonging to three different planes. (c) Plane skeleton of the roof shown in (a).

Each C-DSM is then transformed to a disparity image (C-DISP) to guide dense matching, especially in shadow and low texture areas, in two steps. First, the disparity search space (DSS) for each image pixel is limited to three candidate disparity values with a small range. Second, the probability of each candidate given by LiDAR data is included in the Markov random field (MRF) model, a well-known framework for dense matching, to further improve dense matching in shadow areas near building boundaries. The second step is elaborated below.

The MRF model consists of two terms: an association potential (data term) and a pairwise potential (smoothing term). The LiDAR guidance is included in the data term. The original data term defines a high probability to a disparity when the result has similar colors, while the smoothing term prefers that neighboring pixels have similar disparity. Buildings often cast shadows on the ground near building boundaries. If an image pixel near a building boundary is in the shadow, the color information is not informative to select the right disparity from different candidates. As mentioned above, the order of the C-DSMs indicates which height is more likely given the LiDAR data. The corresponding order is also transformed to C-DISPs. This chapter designs a new local term relying more on LiDAR guidance

in shadow regions as follows:

$$P(d_i|c_{p_i}, c_{p'_i}) = (1 - \omega(\overline{c}_{p_i})) \cdot \max(0, \mathrm{NCC}(c_{p_i}, c_{p'_i})) + \omega(\overline{c}_{p_i}) \cdot \epsilon(\mathrm{Ord}(d_i)), \quad (5.1)$$

where $P(d_i|c_{p_i}, c_{p'_i})$ defines the probability on disparity $d_i$ given the color information of corresponding pixels $p_i$ and $p'_i$ found from left and right image. $c_{p_i}$ and $c_{p'_i}$ stores $3 \times 3$ image patches centered at the corresponding pixels in the left and right image respectively. Probability $P(d_i|c_{p_i}, c_{p'_i})$ relies on two terms: color similarity measurements of the corresponding pixels and a LiDAR guidance term. First, the color similarity of the corresponding pixels is defined as the normalized cross correlation (NCC) over the intensities of the image patches. A max function is applied to restrict the NCC to $[0, 1]$. Second, $\epsilon(\mathrm{Ord}(d_i))$ guides the information from the LiDAR data. $\epsilon(\mathrm{Ord}(d_i))$ is defined as 0.9, 0.7 and 0.5 when the order of disparity $\mathrm{Ord}(d_i)$ is 1,2 and 3 respectively. These two terms are weighted based on whether the left image pixel is in the shadow according to the logistic function $\omega(\overline{c}_{p_i})$, as defined in Chapter 4.

The LiDAR guidance addresses problematic areas in the stereo pair considered, while in return, dense matching uses detailed building boundary information to select the optimal height among candidate heights from LiDAR data near building boundaries. The proposed edge-aware smoothing term in a MRF model, defined in LEAD-Matching, further uses color differences across building boundaries from the stereo pair to enable height discontinuity across the same building boundaries.

### 5.3.2. Generation of iDSM and true ortho-image

The integrated 3D information can be reconstructed from disparities. However, the 3D results often are affected by a layering or fronto-parallel effect in slanted surfaces (Bleyer et al., 2011), as the estimated optimal disparities from dense matching are integer values. Instead, we transform optimal disparities to integrated DSM (iDSM) heights by linking the corresponding C-DSMs and C-DISPs. As shown in Figure 5.3, when every C-DSM is transformed to a C-DISP, a linkage image, called C-DSM-DISP, is created to store the disparity positions (column and row) where the C-DSM pixels are projected to. If an optimal disparity value is selected, as shown in yellow in Figure 5.3 right, from the C-DISP after dense matching, the corresponding DSM values are selected through the C-DSM-DISP image as shown in yellow in Figure 5.3 left. By iterating all the optimal disparities, an integrated DSM (iDSM) using LiDAR data and a stereo pair is obtained. As shown in Figure 5.3, every DSM pixel is linked to a plane. The corresponding plane information for the iDSM is also derived. In addition, a true ortho-image is used to obtain colors, as shown in yellow in Figure 5.3 right, from the raw VHR image back to each DSM pixel. As the transformation from C-DSM to C-DISP is not pixel to pixel as shown in Figure 5.3, in order to assign smooth colors from raw images to true ortho-image, bilinear interpolation is applied. The true ortho-image will be used in the section below.

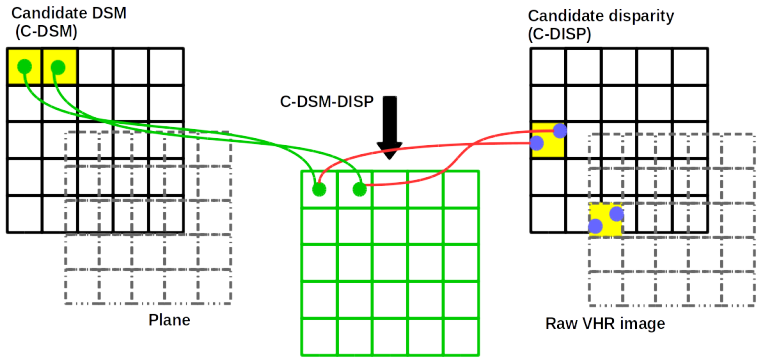



Figure 5.3: The C-DSM-DISP stores the link between a C-DSM and C-DISP. The highlighted yellow pixel presents the selected optimal disparity and the corresponding heights in iDSM. The link is not one pixel to one pixel.

### 5.3.3. Integration with multi-view images

In this section, the problem of using one single stereo pair and the reason for addressing these problems using multi-view images are explained first. Then, a probabilistic integration approach is elaborated to integrate the result from each stereo pair, the iDSM, while incorporating its quality.

**Problem of integration with only one stereo pair**

The integration with only one stereo pair has typical facade and occlusion problems. However, in airborne image acquisition, many buildings can be seen by four images, as shown in Figure 5.4a. Due to the viewing geometry as displayed, relief displacement of the building occurs in four different directions, while four stereo pairs can be selected for dense matching. The stereo pairs in the cross direction are not selected, as shown in Figure 5.4b, as the large difference of relief displacements in the two images would affect the quality of dense matching. In each image, two facade are better visible, while two other facades are occluded, and boundaries are often clearly separated from ground. Four different images where these effects occur are shown in Figure 5.5a.

If a facade is visible in two images, the accuracy of the side of the building boundary in the integration result, the iDSM, is affected. As the candidate DSM created from LiDAR data presents 2.5D height information and does not contain accurate heights on facades, the facade pixels in the image will not be guided to find the correct disparity from candidates in the dense matching step. As shown in Figure 5.4c, one building side in each iDSM will suffer from this facade problem, which is also shown in the example in Figure 5.5b(5,7). Note that on this building side, occlusions are limited. On the other hand, if a facade is occluded in at least one image from a stereo pair, the iDSM would have better accuracy on this side of the building as shown in Figure 5.5b(8). As the candidate DSM created from LiDAR data more likely includes correct candidates for roof and ground image pixels, dense matching using the clear image building boundaries is effective to

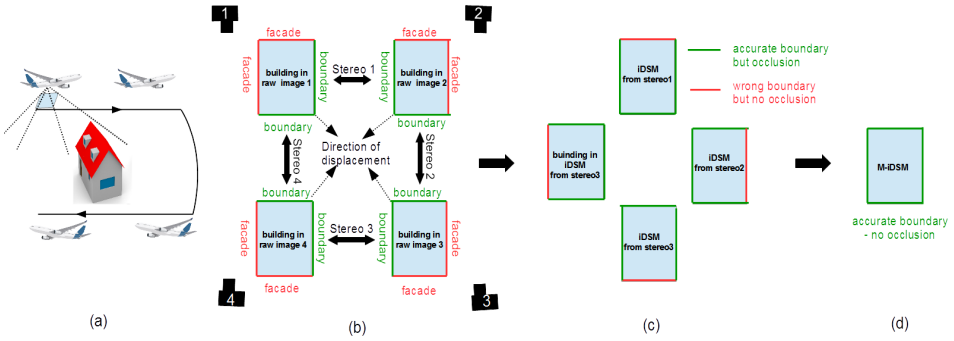



Figure 5.4: (a) A building can often be seen in four-view of stereo aerial images acquire by an airborne camera in four different positions. (b) Different displacements of buildings are in the images acquired from different viewing positions. Usually, facade and clear boundaries are shown in different sides of buildings. (c) The characteristics of iDSM of four stereo pairs are displayed. If facades are both seen in the stereo pair, the building in the iDSM will have wrong heights near building boundaries. In the other sides, building boundaries are more accurate, but occlusions are resulted. (d) By integrating the iDSM from multiple stereo images, the resulted M-iDSM would have accurate building boundaries without occlusions. (b), (c) and (d) are corresponding to a real example displayed in Figure 5.5 (a), (b) and (c) respectively.

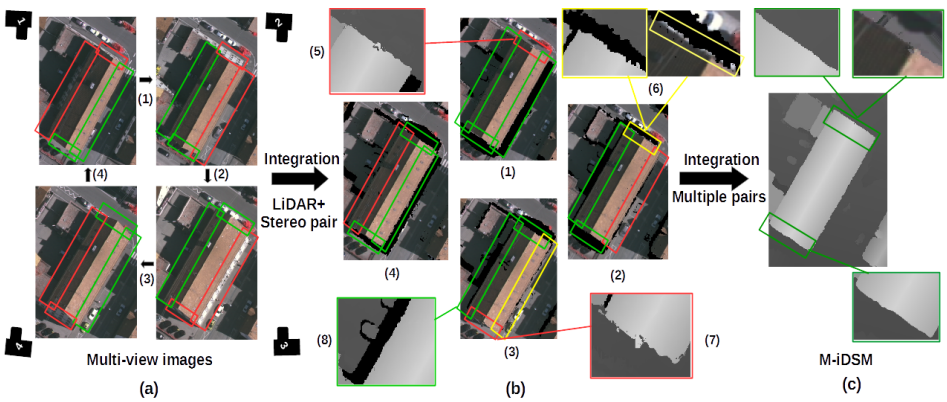



Figure 5.5: (a) Multi-view images from cameras in four different positions. The red boxes indicate that facade areas are more likely to be visible in a building boundary region, while green boxes indicate that facades are more occluded. (b) True ortho images from LEAD-Matching for four stereo pairs. (c) M-iDSM from integrating multi-view images.

choose the accurate height around building boundaries. However, as ground is often occluded in at least one image from a stereo pair, no height value in iDSM is reconstructed in the occluded areas as shown in black in Figure 5.5b.

As shown in Figure 5.4c, in four stereo pairs, each building side would have relatively accurate building boundaries in three stereo pairs and limited occlusions in one stereo pair. Ideally, accurate building boundaries without occlusion can be

derived by integrating the iDSMs from multiple stereo pairs as shown in Figure 5.4d. However, if building boundaries in areas with no occlusions, as shown in Figure 5.5(5,7), are used to fill up occluded areas, the accuracy of these side of building boundaries will be affected. In addition, wrong heights would be assigned in the building boundary areas due to unclear building boundaries as shown in yellow boxes in Figure 5.5b(2, 3). Therefore, a proper integration strategy should be designed.

**Probabilistic integration with multiple stereo pairs**

A probabilistic integration approach is proposed to assign probability to pixels in each iDSM to encode their quality when integrating multiple iDSMs. Two steps are taken to reduce the effect of wrong height assignments and occlusions respectively.

First, if a pixel is assigned a wrong height, the color is often different from other pixels which share the same plane. For example, as a wrong height from the roof plane is assigned to a ground pixel as shown in the yellow box in Figure 5.5b (6), the ground pixel gets color different from the roof in the true ortho-image. In this way, the probability $P(\cdot)$ of the candidate height for a pixel is defined according to color information in the true ortho-images:

$$P(h_{ii}) = \omega(\Delta S) \qquad \text{if } h_{ii} = h_{fDSM}^{jj}, \tag{5.2}$$

where $h_{ii}$ presents the $ii-$th candidate height extracted from LiDAR for guidance and $ii \in \{1, 2, 3\}$. $h_{fDSM}^{jj}$ denotes the height of the pixel in the $jj-$th iDSM and $jj \in \{1, 2, ..., N\}$. $N$ is the total number of stereo pairs. $\Delta S$ is the difference between the pixel color and median color of neighboring pixels in the same plane within 1m radius in the $jj-$th true ortho-image. $\omega(\Delta S)$ denotes the logistic function:

$$\omega(\Delta S) = \frac{1}{1 + a(e^{\frac{\Delta S - b}{\theta}})}, \tag{5.3}$$

where $a, b$ and $\theta$ are parameters defining the shifting and decay speed of the possibility curve.

Second, as described above, in occlusion areas, the heights filled from other iDSMs may not accurate. As the occlusion happens due to higher objects blocking the camera, the height for a pixel in occlusions is less likely to be the highest height among its three height candidates provided by LiDAR data. Therefore, LiDAR guidance is added in the integration. First, the height candidates are sorted in descending order. The probability of the candidate height $P(h_{ii})$ for a pixel in the occlusion area is defined as:

$$P(h_{ii}) = \epsilon^{'}(\text{Ord}(h_{ii})) \quad \text{if } h_{fDSM}^{jj} = None, \tag{5.4}$$

where $\epsilon^{'}$ is a constant to assign the probability to height candidates ordered second or third. $\epsilon^{'}$ is set to 0.6 when $\text{Ord}(h_{ii})$ is 2 or 3 empirically. $h_{fDSM}^{jj} = None$ indicates occlusions in $jj - th$ iDSM.

Finally, the multi-integrated DSM (M-iDSM) is obtained by choosing a height from three candidates for each DSM pixel with the highest probability added from

all iDSMs:

$$\arg\max_{ii} \sum_{jj} P^{jj}(h_{ii})' \tag{5.5}$$

The resulting M-iDSM is shown in Figure 5.5c. The upper example shows that the improved accuracy of the boundary previously assigned with wrong height and color as shown in Figure 5.5b(6). In addition, the lower example shows that including LiDAR guidance (i) helps to fill in accurate heights in the occlusion area previously shown in Figure 5.5(1,2,4), and (ii) reduces the effects of facades previously shown in Figure 5.5(7).

### 5.3.4. Building extraction

In order to compare planimetric accuracy of buildings, building detection is unnecessary. There are tens of extraction methods tested on benchmark data (ISPRS WG III/4, 2019). Methods rely on many steps and assumptions to distinguish buildings from other objects, especially from trees and bushes. In order to reduce the effects of building extraction rate and to better evaluate the planimetric accuracy, reference building polygons are used to extract buildings to largely remove the effects from trees and bushes. The building are extracted and delineated from LiDAR data alone and our integration approach. The steps are shown in Figure 5.6.

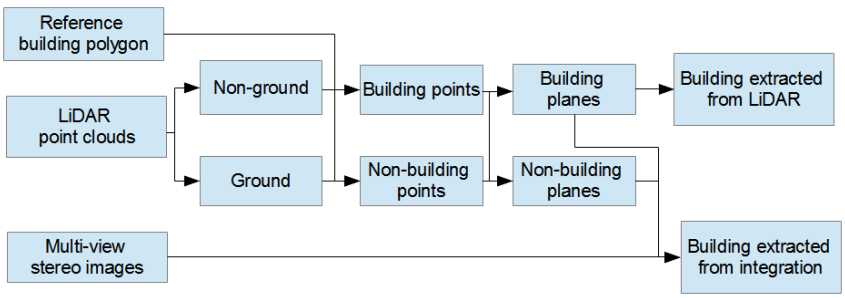



Figure 5.6: The framework of building extraction using referecen building polygons to best extract buildings.

The LiDAR data is first separated into ground and non-ground by LASTools (Isenburg, 2019). Reference building polygons are used to clip the building points from non-ground points and the remaining points are merged with the ground points. Therefore, the resulting points are classified into building and non-building points. Among building points, still a few trees and bushes near buildings are included. Buildings often consist of many planar surfaces, while points of trees or bushes are more scattered. The plane segmentation algorithm in the Point Cloud Mapper software from Twente university (Vosselman et al., 2004) is used to find planes with more than 6 and 10 points for Vaihingen and Amersfoort respectively, which corresponds to an area of 1 $m^2$. The points in the extracted planes

are the final building points extracted. The building polygons from LiDAR data are delineated from the remaining building points by alpha shape (Edelsbrunner et al., 1983), while building polygons from our integrated method are extracted and delineated as follows. The plane segmentation algorithm is also applied to non-building points. Building and non-building points are combined again with the planar information to provide candidates for LEAD-Matching. After integration with multi-view images, the height of M-iDSM is obtained along with the corresponding plane information. Buildings are extracted in the pixels assigned to the building planes. The polygon of each building is easily obtained by finding the corresponding boundary pixels.

### 5.3.5. Evaluation

The visual results are first provided for evaluation. When comparing the proposed integration method to multi-view images alone, the two DSMs are compared in the visual results. When comparing the proposed integration method to LiDAR data, buildings are extracted. The visual results compare detected building areas to the reference and show true positives, false negatives and false positives in different colors (Rottensteiner et al., 2014).

The quantitative result in terms of the planimetric accuracy of building extraction is represented by the RMS (root mean square) of the planimetric distances of the extracted boundary to their nearest point in the reference boundaries(Rottensteiner et al., 2014). Only distances shorter than 2m are considered. However, this metric cannot represent the effects of occlusions well. For example, when many wrong boundaries are extracted from a building in occluded areas, the reference boundary points can still correspond to the nearest points from extracted boundaries, while the wrong boundaries in the extracted building cannot. Therefore, the RMS of the planimetric distances of the extracted boundaries to the reference boundary is calculated. We denote these two RMS as, RMS1 and RMS2. Finally, an overall metric, denoted aRMS, is the average of RMS1 and RMS2.

## 5.4. Experiment and results

### 5.4.1. Vaihingen results

In this section, the planimetric accuracy of building extraction from either LiDAR or image data alone is compared to our proposed integration approach with different number of images. In order to better understand the effects of using multiple stereo pairs, the viewing geometry and comparative analysis of integrating increasing number of stereo pairs are discussed.

#### Comparison of planimetric accuracy

The visual results of planimetric accuracy of building extraction are compared in Figure 5.7. For quantitative evaluation, the results of integrating different number of images are added and compared in Table 5.1 and the effects of increasing the number of images that are integrated are shown in Figure 5.8.

In Figures 5.7a and b, buildings extracted from LiDAR data alone and from LiDAR data with 6-view images are compared. As indicated by the many false negatives around building boundaries, the buildings extracted from LiDAR data alone are in general smaller than the reference due to the sparsity and irregular spacing problem of LiDAR data. However, when the LiDAR data is integrated with 6-view images, the false negatives are largely reduced. The planimetric accuracy indicated in Table 5.1 also shows that the aRMS decreases from 0.40 m to 0.22 m.

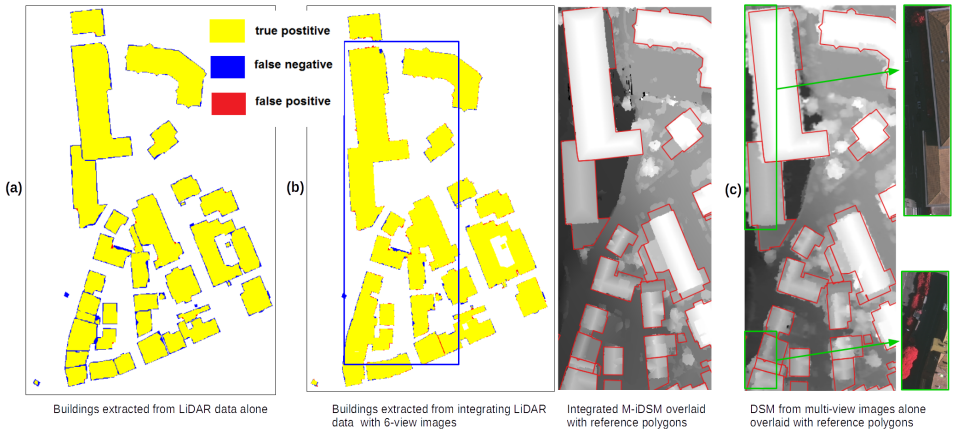



Figure 5.7: Visual comparison of building extraction results from (a) LiDAR alone, (b) integration with 6-view images and (c) multi-view images alone. The ground sampling distance of the DSM converted in (b) and (c) is corresponding to 8 cm and 9 cm respectively. The area of the DSMs is indicated in the blue box in (b).

The M-iDSM obtained by integration is compared to the DSM obtained from multi-view image in Figures 5.7b and c. As indicated in the green boxes, the extracted 3D information is largely affected by shadows casted by buildings when using multi-view image alone. If the buildings are extracted, the planimetric accuracy will be even worse than 1 meter. On the other hand, the M-iDSM resulting from using LiDAR guidance largely reduces the negative effects of dense matching while giving a good match with the reference polygons.

| Methods | RMS1(m) | RMS2(m) | aRMS(m) |
|---|---|---|---|
| LiDAR alone | 0.39 | 0.40 | 0.40 |
| LiDAR+2 images | 0.25 | 0.45 | 0.35 |
| LiDAR+4 upper images | 0.25 | 0.25 | 0.25 |
| LiDAR+4 lower images | 0.24 | 0.25 | 0.25 |
| LiDAR+6 images | **0.22** | **0.21** | **0.22** |

Table 5.1: Quantitative evaluation in Vaihingen. The boundary average RMS (Root Mean Square) is used to measure the planimetric accuracy of building extraction.

In Figure 5.9a, the viewing geometry of 6-view images is displayed. Several

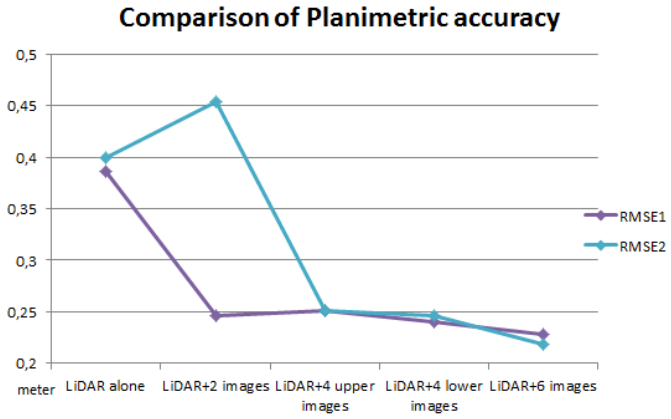



Figure 5.8: The trend of planimetric accuracy by integrating increasing number of stereo pairs.

integration experiments using different number of images are compared. First, LiDAR data is integrated with two images with small viewing angles as shown in the middle of Figure 5.9a, such that the effects of relief displacement is reduced. Second, the upper and lower 2 images with larger view angles are added to form 4-view images for integration. Finally, all 6-view images are integrated. The details of designing these experiments will be elaborated in the following sections. As shown in Figure 5.8, RMS1 drops quickly when the first two images are integrated, however, the RMS2 increases. The reason is that integration employs detailed building boundaries in the single stereo pair to improve accuracy in boundary areas, however, still the occlusions in the single stereo pair create wrong boundaries which increases the RMS2. When the 4 upper or lower images are integrated, the RMS2 improves strongly by around 0.2 m as the occlusions are largely reduced. However, the RMS1 remains around 0.25 m. The reason is that when adding two images with large view angles, some buildings are completely occluded which is clearly shown in the 2 images used at the beginning. Finally, by integrating all 6 images, both RSM1 and RMS2 decreases to 0.22 m and 0.21 m, which corresponds to 2.9 and 2.7 pixels. These effects of adding different number of images are discussed in the following sections.

**Viewing Geometry and experiment design**

The viewing geometry of multi-view images is an important factor influencing relief displacement, which consecutively results in occlusions and facade problems discussed before. Therefore, we display the position and viewing angle of all 6 images with respect to the research area in Figure 5.9a. Part of the area is selected from the upper four images to analyse the effects of the viewing angles as shown in Figure 5.9b. Due to the displacement, occlusion and facade problems happen at different sides of the buildings as shown in Figure 5.9b below. Image 1, with small vertical and horizontal viewing angle, has limited displacements in any direction,

while image 2 has a bit larger displacement in west direction only. However, in images 3 and 4, the large relief displacement in south direction is a result of the large vertical viewing angle while the closeby high buildings occlude some lower buildings. Two building examples are indicated in the boxes in Figure 5.9b. As images 5 and 6, acquired from opposite sides, have large vertical angles similar to image 3 and 4, large relief displacement occurs in the north direction.

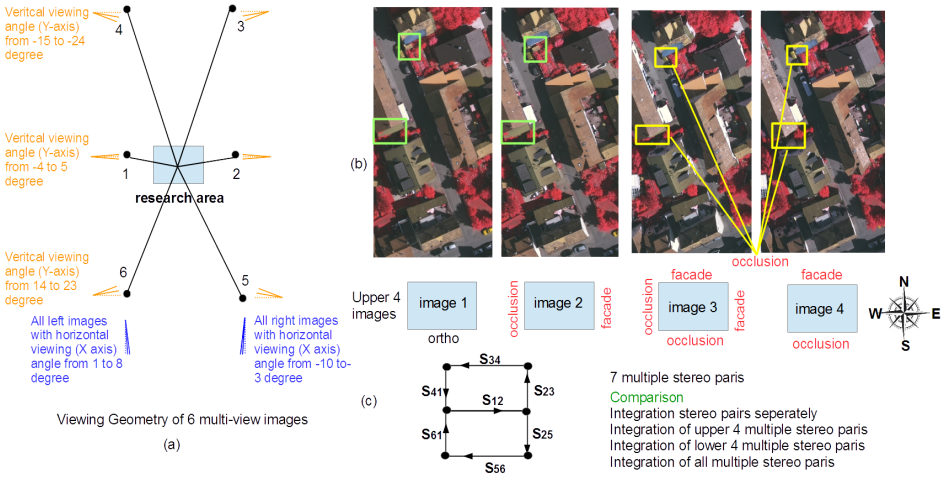



Figure 5.9: (a) The viewing geometry of 6-view images as available for the research area. (b) Part of the area is selected and the four upper images are displayed. The direction of occlusions and facades are indicated in each images. The yellow boxes show that some buildings are completely occluded in the images, while the green boxes show buildings that are visible in the images. (c) 7 stereo pairs are selected from 6-view images. E.g. stereo pair $S_{12}$ consists of images 1 and 2.

From the available 6-view images, 7 stereo pairs are selected as shown in Figure 5.9c. If a stereo pair consists of images 1 and 2, we denote this stereo pair by $S_{12}$. Similar terminology is applied to all stereo pairs as shown in Figure 5.9c. An experiment is performed to evaluate effects on the planimetric accuracy of buildings of integrating different number of stereo pairs. First, LEAD-Matching is applied to integrate LiDAR data with each stereo pair. Then, the integration of multiple stereo pairs is performed on the lower and upper four stereo pairs respectively. Finally, integration of all stereo pairs is applied. The results are elaborated below.

**Comparison of integrating multiple stereo pairs**

In the comparison, we consider whether effects of occlusions, facades and unclear building boundaries when integrating a single stereo pair can be reduced by integrating multiple stereo pairs. First, two different buildings get occluded in the integration results with three individual stereo pairs, indicated as Occlusion 1 and 2, in the red boxes in Figure 5.10a. The reason is that these three stereo pairs all involve either image 3 or 4 where the two buildings are occluded as shown in the yellow boxes in Figure 5.9b. When the 4 upper stereo pairs are integrated,

Occlusion 1 is reduced, but Occlusion 2 remains, as shown in Figure 5.10b left. The difference between Occlusion 1 and 2 is that the building in Occlusion 2 is small and separated from the high building occluding it. LiDAR guidance introduced in probabilistic integration approach, gives more probability to the two candidates with lower heights, which come from the ground planes. The occlusion area will have a higher probability on ground height (plane) than on roof height (plane), such that the building is still missing after integration of the upper 4 stereo pairs. This also explains why RMS1 of the buildings, when integrating 2 more images, does not decrease as indicated in Table 5.1. However, this building is not occluded, when integrating the lower 4 stereo pairs as indicated in Figure 5.10b middle. Therefore, by integrating all stereo pairs, the building is finally detected as shown in Figure 5.10b right. A similar occlusion problem happens in the integration with the lower 4 stereo pairs as shown in 5.10b middle. Due to the vertical large viewing angle of images 5 and 6 as shown in Figure 5.9a, the lower buildings in the north direction are occluded. This occluded building is finally compensated by involving the upper stereo pairs as shown in Figure 5.10b right.

Second, facade problems are indicated in Figure 5.10 when integrating $S_{34}$. The extracted building involves false positives on the building boundaries. The reason is that $S_{34}$ consists of images 3 and 4 that see the north facing facades of the buildings as shown in Figure 5.9b. On the other hand, the boundaries are relatively good compared to the other three integration results, as shown in Figure 5.10a. Therefore, when the 4 upper stereo pairs are integrated, the false positives are reduced, as shown in Figure 5.10b left.

Third, in the result of integrating $S_{12}$, the building boundaries are less affected by occlusions and facade problems, as the two images have small viewing angles. Still, some building boundaries are affected, as indicated in Figure 5.10a, due to unclear boundaries in the stereo pair. By integrating the other three stereo pairs, the boundaries are improved as shown in Figure 5.10b left.

The RMS1 of building extraction finally decreases to 0.22 m with the highest planimetric accuracy when 6 images are integrated, which also confirms the effectiveness of integrating multi-view images to address the problems discussed above.

### 5.4.2. Amersfoort results

The buildings considered in Amersfoort are less high and less densely located. Only 4 view images are available. The viewing geometry is similar to the geometry of the upper 4 images described in Vaihingen. In general, the results confirms those for Vaihingen area.

The false positives around building boundaries as shown in Figure 5.11a confirms that the accuracy of building extraction from LiDAR alone is affected by sparsity and irregularly spaced points. The DSM from point clouds extracted from multi-view images only as shown in Figure 5.11b confirms that the quality problems due to shadows and low textures near building boundaries largely affects the planimetric accuracy of buildings. When 4-view images are integrated with LiDAR data, the boundary errors are reduced as shown in Figure 5.11c and the




Figure 5.10: Visualization of extracted building areas, compared to reference data, for different scenarios. (a) the comparative building extraction results of integration with 4 individual stereo pairs from upper four images as shown in Figure 5.9b. The comparative building extraction results of integration with 4 upper stereo pairs, 4 lower stereo pairs and all 7 stereo pairs respectively as shown in Figure 5.9c. Red boxes show the problems which affects the accuracy of buildings, while the green boxes show the problems have been resolved by integrating multiple stereo pairs.

M-iDSM matches well the reference polygons. The quantitative results shown in Table 5.2 also confirms that both RMS1 and RMS2 decrease from 0.48 m to 0.20 m (5.8 pixels) and 0.47 m to 0.21 m (5.8 pixels) respectively compared to the buildings extracted from LiDAR alone. Two examples of integrating images to solve the problem of sparse LiDAR points are shown in the green boxes in Figures 5.11a and c. In addition, by integrating only one stereo pair, the accuracy of building boundaries is confirmed to increase, while the occlusions will create wrong boundaries, such that the RMS2 is still high as shown in Table 5.2.

| Methods | RMS1 (m) | RMS2 (m) | aRMS (m) |
|---|---|---|---|
| LiDAR alone | 0.48 | 0.47 | 0.48 |
| LiDAR+2 images | 0.26 | 0.47 | 0.37 |
| LiDAR+4 images | **0.20** | **0.21** | **0.21** |

Table 5.2: Quantitative evaluation in Amersfoort. The boundary average RMS (Root Mean Square) is used to measure the planimetric accuracy of building extraction.

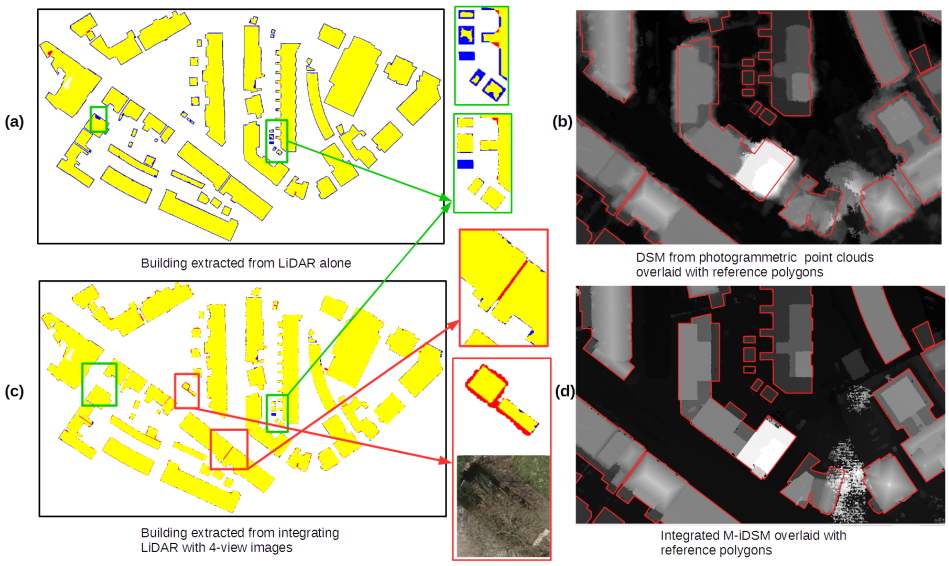



Figure 5.11: Visual comparison between building extraction from LiDAR alone (a) and integration with 4-view images (b), and visual comparison between the DSM obtained from multi-view images alone (b) and the M-iDSM obtained by integration (d). The ground sampling distance of both DSM is the same at, 3.5 cm. In the green box, the missing point problem in LiDAR data is reduced by integrating images. In the red box, two drawbacks from the integration results are shown.

The planimetric accuracy of buildings in Amersfoort is around 3 pixels higher than those in Vaihingen. By adding more images, the planimetric accuracy is expected to increase a bit more. However, there are three main problems which will remain, even when integrating more images. First, in the green box in Figure 5.11b(2), the blue building is still missing as the correct planes can not be correctly extracted from the very few points available in the LiDAR data. Second, in Figure 5.11b, two separated buildings are connected as the gap between the buildings is small and in the shadow. Such problems occur at several locations in the research area. Third, in the lower example in Figure 5.11b, the result is affected by trees and walls adjacent to the building.

## 5.5. Discussion

In this discussion, planimetric accuracy of our building extraction results is discussed in the context of the ISPRS benchmark contest. In addition, parameter settings are discussed.

### 5.5.1. Building extraction

The planimetric accuracy of building extraction for Vaihingen as shown in Table 5.1 is in general better than all results delivered in the benchmark contest (ISPRS

WG III/4, 2019). The best RMS reported in the benchmark contest is 0.67m. The reason is that the building extraction rate also affects the final RMS value. This chapter uses reference data to reduce the effect of building extraction rate. Therefore, the results in Table 5.1 are suitable to compare the planimetric accuracy using LiDAR alone to our proposed integration approach, while other irrelevant effects are largely reduced.

### 5.5.2. Parameter Setting

The parameter setting for LEAD-Matching is discussed in Chapter 4. In general, for E-LEAD-Matching, the parameters values can be set the same. Even though the datasets from Vaihingen and Amersfoort are acquired in different environments and by different instruments, the same set of parameter values has been applied to all experiments. Only the parameters involved in integrating multiple stereo images are discussed here.

First, the parameters involved in Equation 5.2 define the color similarity between a pixel and the neighboring pixels in the same plane. As color similarity is often less affected by illumination and is less environment dependent in the neighborhood, the value $a, b$ and $\theta$ are set to 0.1, 10 and 8 respectively. If the color difference, $\Delta S$ is smaller than 10, the $P(h_{ii})$ is above 0.9. If $\Delta S$ is bigger than 50, $P(h_{ii})$ is below 0.1. Second, the parameter $\epsilon^{'}$ is involved in tuning LiDAR guidance in the occlusion areas in Equation 5.4. According to Figure 5.12 from the sensitivity test, the value for $\epsilon^{'}$ should be set to 0.4-0.6, which suggests a moderate LiDAR guidance in the integration.

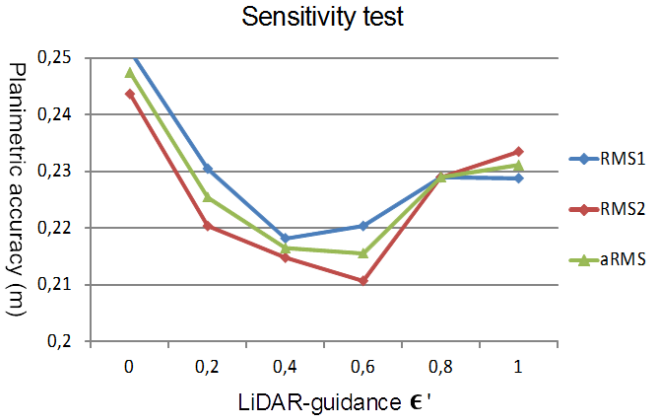



Figure 5.12: Sensitivity test of $\epsilon^{'}$ on integrating 6-view images in Amersfoort with metrics of RMS1, RMS2 and their average, aRMS.

## 5.6. Conclusions

This chapter applies LEAD-Matching to integrate plane information in LiDAR data of high vertical accuracy with detailed building boundaries in multiple images with high planimetric accuracy. A probabilistic integration approach is proposed to integrate multiple stereo pairs to reduce facade and occlusion problems when integrating only one stereo pair, while the building boundaries from different images are also integrated to further improve the planimetric accuracy of building extraction.

The proposed method is applied in two urban areas: Vaihingen, Germany and Amersfoort, the Netherlands. Both results show that by integrating multi-view images with LiDAR data, the planimetric accuracy, aRMS, is improved from 0.40 m to 0.22 m and from 0.48 m to 0.21 m respectively. These improved planimetric accuracy shows the potential of using integration for generating building for a large scale map. Also the effect of integrating increasing numbers of images is compared. The results shows that by integrating more images, occlusion and facade problems are truly reduced. The building boundaries in different images also improve the boundary accuracy. In general, 4-view images are enough to compensate for occlusion problems. However, when the 4-view images include images with large viewing angle in an area with dense high buildings, some small buildings may be still occluded. In these cases, more images, which can see the buildings well, should be added to reduce these effects. As our airborne camera is in nadir position, the occlusions in the images are less compared to oblique images. In our experiment, we show that acquiring 6-view images works well in areas with dense high buildings. However, adding even more images is often difficult in practice and may not give significant improvement. The reason is that the three main problems described in Section IV (B) which affect planimetric accuracy are not relevant to the number of images integrated.

# 6

# Conclusions and recommendations

*The goal of this research is to integrate 3D information from airborne laser scanning and airborne camera imagery to generate accurate and up-to-date 3D city models, focusing on buildings. In this chapter, we present the main conclusions and recommendations for future research.*

## 6.1. Conclusions

3D information provided by airborne laser scanning is widely used for generating 3D city models, especially buildings. However, LiDAR data is less frequently acquired than image data, as the price of an airborne laser scanning system is about 10 times higher than that of an airborne camera imagery system (Wingtra, 2019). Since image acquisition is cheaper and therefore often more frequent, this study focuses on integrating 3D information from image and LiDAR data with two objectives: (i) 3D change detection and updating of buildings in LiDAR data using images and (ii) improving the planimetric accuracy of building extraction from LiDAR data using images. To fulfill objective (ii), the acquisition time of images is preferable similar to the ALS data. To achieve objective (i), newly acquired images are required. To achieve objective (ii), the acquisition time of images can either be similar to or more recent than the acquisition time of the LiDAR data.

As an increasing number of images contains more 3D information but also increases the difficulties and computation efforts, this study presents three methods to integrate 3D information from images to LiDAR data: (1) using a single image, (2) using a single stereo pair and (3) using multiple stereo pairs. The first two methods are designed for objective (i), while the third method is designed for objective (ii). As shadow represents 3D information in a single image, we propose **Umbra Method** using ray tracing with a KD tree to efficiently reconstruct shadows from LiDAR data to provide training samples for a supervised learning to detect shadows from single images. However, shadows only indicate partial 3D information, therefore they can only be used to derive partial changes. Using a single stereo pair, more complete 3D changes can be detected. We propose **LEAD-Matching Method** (LiDAR-guided edge-aware dense matching) to use a single stereo pair to detect accurate changes of buildings in LiDAR data. LEAD-Matching is combined with a second dense matching step to extract 3D information from the stereo pair to update buildings in changed areas. Finally, We propose

**E-LEAD-Matching Method** (extended LEAD-Matching) to integrate LiDAR data with multiple stereo pairs to improve the planimetric accuracy of building extraction from LiDAR data. The main conclusion is that **(E-)LEAD-Matching** enables us to integrate the beneficial aspects from two data sources to create an accurate and up-to-date 3D city model.

As stated in Chapter 1, the research question of this research is "**How to integrate 3D information from airborne laser scanning and airborne camera imagery to create an accurate and up-to-date 3D city model, focusing on buildings?**". In order to answer this question, we defined three main questions and several sub-questions. Section 6.1.1 and 6.1.2 discuss change detection and 3D updating by integrating LiDAR data with a single image or a single stereo pair respectively. Section 6.1.3 discusses how to improve the planimetric accuracy of building extraction by integrating LiDAR data with multi-view images giving multiple stereo pairs.

### 6.1.1. Shadow detection from a single image

The advantage of extracting shadows to indicate building information from a single image is that the processing workflow is simpler than extracting 3D information from stereo images. However, the color properties of shadows in images are material and environment dependent. As urban areas often do not change dramatically, many objects are still geometrically correctly represented in LiDAR data acquired one or several years before. Therefore, the first research question states "**How to detect shadows from a single image by integrating information from LiDAR point clouds?**".

A supervised machine learning approach can learn color properties of shadows if training samples are provided. On the other hand, given a perfect and up-to-date 3D model, the sun's position at image acquisition time and the camera's intrinsic and extrinsic parameters, shadows in an image can be reconstructed accurately using ray tracing. However, when a 3D model is automatically created from LiDAR data acquired in the past, a portion about 20% of the reconstructed shadows will be wrong due to modelling errors and outdated 3D information. Still, the reconstructed shadows can be used to automatically generate training samples for a supervised learning approach to learn the color properties of shadows. As shadow reconstruction is often time consuming and a portion of training samples are mislabeled, two sub-questions are considered and addressed as follows.

***Sub-question 1. How to efficiently reconstruct shadows from LiDAR data to generate training samples for a subsequent machine learning approach?***

Ray tracing can reconstruct accurate shadows from 3D models, but is computationally intensive. The computation costs are mostly due to the intersection costs between the optical ray generated from each image pixel and all the triangles in the mesh representing the 3D model. The efficient shadow reconstruction in Umbra method developed in Chapter 3 focuses on two aspects: simplifying the 3D triangular model and constructing a data structure for fast intersection. First, a

topographic map which is commonly used in construction with LiDAR data for 3D reconstruction, is employed to create a simplified 3D triangular model. The model is further simplified by selecting only triangles within the field of view of the image considered. As a KD tree data structure for 3D triangular models could reduce the intersection time of ray tracing exponentially, we employ an efficient KD tree construction method to reduce the construction cost from $O(N^2)$ to $O(Nlog^2N)$, where $N$ represents the number of triangles. Finally, the optical rays are parallelized using CPUs.

The efficiency of shadow reconstruction for an image of around 20 million pixels is tested with 3D triangular meshes of different sizes of up to 1 million triangles. The experiments show that both the costs of KD tree construction and ray tracing match the theoretical estimation. Even though ray tracing takes more time than KD tree construction, time only scales almost linearly with the logarithm of the number of triangles, which is practically efficient. As ray tracing is easy to be parallelized, combining ray tracing with an efficient KD tree construction to reconstruct shadows is practically feasible and scalable by just using more CPUs for processing even larger 3D models.

***Sub-question 2. How can we choose a machine learning approach using training samples provided by LiDAR data to detect shadows in a single image?***

As mentioned above, we notice that often about 20% of reconstructed shadows from LiDAR data is wrong for the image where shadow needs to be detected from. When using the reconstructed shadow image to select training samples, a small portion of samples is, therefore, mislabeled. In addition, material and environmental effects complicate the color properties of shadows. Therefore, an effective supervised learning approach for classifying image pixels is required that is able to capture complex color properties of shadows and is robust to mislabeling.

In Chapter 3, we compare four common machine learning methods, quadratic discriminant analysis (QDA) fusion, support vector machine (SVM), K nearest neighbors (KNN) and random forest (RF). Both theoretical and experimental comparison are performed. We found that QDA fusion relying on the Gaussian distribution of features has difficulties to characterize complicated property of shadows, while it is robust to mislabeling. SVM is capable to capture the non-Gaussian distributed spectral properties of shadows. However, its performance drops quickly when mislabeling effects are introduced. KNN relies on the number of nearest neighbors chosen and its robustness to mislabeling and effectiveness in capturing shadow property is case dependent. Random forest uses a collection of classifiers to effectively capture complicated shadow properties, while it uses a bagging mechanism to be robust to mislabeling. All the experiments show that RF outperforms or at least perform equally well compared to the other three machine learning methods.

### 6.1.2. 3D building detection and updating using a stereo pair

As more complete 3D information can be extracted from a stereo pair, using a stereo pair can provide better change detection from LiDAR data than using shadows in a single image. However, the quality problems of 3D information in both LiDAR and image data will affect accurate change detection. The complementary information in both data sources can be integrated to address these quality problems. Therefore, the second research question states "**How to detect accurate 3D changes and perform 3D updating by integrating 3D information from LiDAR point clouds and a stereo pair?**".

To be more specific, the quality of 3D information extracted from a stereo pair is affected by shadows and low texture in images, while LiDAR point clouds are sparse and irregularly spaced. In addition, a framework of incorporating the integration of the two data sources, to derive accurate 3D change detection and updating is required. Therefore, three sub-questions are considered and addressed as follows.

***Sub-question 1. How can LiDAR point clouds be integrated with a stereo pair to improve the quality of 3D information extracted from a stereo pair, especially in regions affected by shadow and low texture?***

3D information is often extracted by searching for corresponding pixels in a full disparity search space (DSS) along epipolar lines in stereo images. In problematic regions where pixels are affected by shadow and low texture, many pixels along an epipolar line would have similar color or texture as the queried pixel.

In Chapter 4, we proposed LEAD-Matching method that integrates 3D information in LiDAR data as a guidance for better identifying corresponding pixels in two ways. First, LiDAR data is densified into three candidate digital surface models (DSMs) to provide effective candidates for each image pixel to limit its DSS. Effective candidate heights are estimated by assigning each DSM pixel to up-to three different neighboring planes extracted from the LiDAR data. In most cases, the correct plane is included. Therefore, dense matching is more likely to find the correct corresponding pixels from the limited number of but effective candidates, especially in problematic regions. Second, LiDAR guidance is added in designing a probabilistic model for dense matching in shadow areas. Such model is originally defined to assign a high probability to a disparity if the corresponding pixels have similar colors. In our approach, the probability of candidates obtained from LiDAR data based on the adjacency of the candidate planes to the DSM pixel is added to reduce the dependency on color similarity in shadow areas to choose corresponding pixels.

We tested the proposed method by comparing it to the robust-dDSM method (Tian et al., 2014) that uses 3D information extracted from stereo images to detect changes in LiDAR data. The comparison demonstrates that the robust-dDSM method generates many false alarms for change detection in case of missing or outlying points extracted from problematic regions in images, while the proposed approach is able to largely avoid these false alarms.

***Sub-question 2. How can a stereo pair be integrated with LiDAR***

***data to address the problem of sparsity and irregular spacing in LiDAR point clouds, especially near edge areas?***

Sparse and irregularly spaced LiDAR data may only have few points to sample a small building. The changes detected for small buildings can be difficult to be separated from other irrelevant small objects. In addition, linking such LiDAR points to the denser 3D information in stereo images to perform change detection is difficult.

As mentioned above in Chapter 4, LEAD-Matching densifies sparse LiDAR data according to the ground sampling distance (GSD) of images to create three candidate DSMs by assigning three adjacent planes for each DSM pixel to estimate the three candidate heights. Especially for pixels near building edges, these three adjacent planes may be all different, representing planes from roof, wall and ground respectively. Then we employ the detailed color information in a stereo pair to determine the height near the building edges from the three candidates. The proposed edge-awareness term in LEAD-Matching also exploits the color difference across image edges to better determine height jumps at building edges. Finally, LEAD-Matching enables to reduce the change detection procedure to a simple check on the color difference of corresponding pixels extracted from dense matching.

We tested the proposed method by comparing it with the existing projection–geometry (Qin, 2014) method that densifies LiDAR data to one DSM using interpolation for finding corresponding pixels in a stereo pair to detect changes from color differences. The result demonstrates that indeed the proposed method is able to largely reduce false alarms near unchanged building edge areas compared to the projection–geometry method.

***Sub-question 3. How to design a framework for accurate 3D change detection and updating in airborne LiDAR data using a stereo pair?***

By addressing the two problems above, LEAD-Matching is able to obtain high success rate of both building verification and change detection on small buildings. However, only partial changes are detected due to homogeneous surroundings and shadows, where wrong corresponding pixels can still have similar colors.

Therefore, we propose a second dense matching step to estimate disparities from a stereo pair, but only in the areas propagated from the partial changes identified previously. The resulting disparities are compared to the disparities estimated in the first step of dense matching for detecting new 3D changes. Accordingly, 3D information is further reconstructed for updating. The same procedure is applied to areas propagated from the new changes iteratively until no more new changes are found. This gives a two-step dense matching framework. The first step aims at verifying buildings while detecting accurate partial changes, while the second step focuses on the areas around the partial changes detected in the first step to complete changes and perform 3D updating.

The proposed two-step dense matching method is applied to two cities, Amersfoort and Assen, in the Netherlands. The results shows that in both cities, all the unchanged buildings are successfully verified, while the size of minimum detectable

building changes is $2 \times 2 \times 2\ m^3$ with a F1 score over 0.9. Therefore, we conclude that our proposed framework is able to combine infrequent LiDAR data with newly acquired images to perform accurate 3D change detection and updating for large scale 3D mapping.

### 6.1.3. Improving building extraction using multi-view images

Building extraction accuracy relies on both success rate and planimetric accuracy. The extraction rate defines how well buildings are differentiated from other objects, while the planimetric accuracy defines how well the building boundaries are extracted. We focus on improving the planimetric accuracy which is important to meet requirements of large scale maps. The required planimetric accuracy is in the order of 0.25 m–0.50 m. Due to the sparsity of LiDAR data and the quality problem of 3D information extracted from image data, the extracted building boundaries from either LiDAR or image data alone can hardly meet this high accuracy. In addition, in a stereo pair, some building parts are occluded and some building boundary information may not be useful to improve boundary accuracy. With multiple stereo images, occlusions can be reduced and building boundaries information from different images can improve the boundary accuracy. Therefore, the third research question states "**How to integrate 3D information from LiDAR point clouds and multi-view images to improve the planimetric accuracy of building extraction?**" and is addressed by the following two sub-questions.

***Sub-question 1. How to integrate 3D information from both LiDAR data and multi-view images to reduce the effect of occlusions in building extraction?***

The planimetric accuracy of extracted building boundaries using LiDAR data alone is often affected by the sparsity and irregular spacing of LiDAR data. As a consequence, building areas are often estimated too smaller. By applying the proposed integration method as described in Chapter 4, the planimetric accuracy can be improved by integrating the plane information from LiDAR data to densify points for extending the building boundaries, while high quality building boundaries in a stereo pair are used to improve 3D information near building boundaries. However, many building parts are missing due to occlusions when only one stereo pair is used.

E-LEAD-Matching we propose in Chapter 5 first applies LEAD-Matching to integrate LiDAR data with each stereo pair selected from multi-view images, and then designed a probabilistic integration approach to integrate the result from all the stereo pairs. In each single stereo pair result obtained by dense matching, also called an integrated DSM (iDSM), the probability of each estimated pixel height is estimated by considering its relative quality, which will be explained in the second question below. However, no height value is obtained in occlusion areas. As occlusions are often caused by high objects nearby, the height for a pixel in occlusion areas is more probably to be the two lower heights among its three height candidates provided by LiDAR data. Therefore, we assign a constant probability

to the two lower heights for the pixel. The integration of multiple stereo pairs is to choose a height with highest probability for each pixel.

We apply the proposed method on the Amerfoort data that has 4-view images, which is commonly available. The results confirm that using multiple stereo pairs from 4-view images reduces occlusions and improves planimetric accuracy compared to the results of single stereo pairs. But when applying the method on the Vaihingen data in a neighborhood where many tall buildings are located closely together, 6-view images are recommended to further reduce more serious occlusion effects. In addition, we perform a sensitivity test on the value of the constant probability, and the test shows that using LiDAR information improves the planimetric accuracy and that a moderate constant probability value works best.

***Sub-question 2. How to integrate LiDAR data with detailed boundary information from multi-view images to improve planimetric accuracy of building extraction?***

The next challenge is that the facade problems and unclear building boundaries in single stereo pairs affect the planimetric accuracy of building boundaries. First, the LiDAR data cannot provide correct candidate heights for facade areas as the candidate DSMs provided by LiDAR data do not present accurate facade heights. As a result, wrong heights are estimated on the facades. Second, some building boundaries are not clear in a given stereo pair. As a result, wrong heights may be estimated near building boundaries as the color information around building boundaries is not informative.

In Chapter 5, we provide an analysis on why introducing multi-view images can address the facade problem using an example of 4-view images. In addition, in the probabilistic integration approach mentioned above in E-LEAD-Matching, the probability of the height of each pixel in each iDSM, a dense matching result from each stereo pair, is assigned according to the color difference of the pixel to its neighbors in the same plane. The idea is that when a wrong height is assigned, the color of the pixel is often different to a pixel which shares the same roof or ground plane.

Visual comparison of planimetric accuracy of buildings extracted from integrating an increasing number of stereo pairs demonstrate that adding more image pairs improves building boundary. The result also shows that the proposed E-LEAD-Matching could extract buildings to meet the requirements of large scale topographic maps, while using LiDAR or multi-view image data alone is not possible.

## 6.2. Recommendations for future work

The work presented for integrating airborne laser scanning and camera imagery data for generating up-to-date 3D city models also provides insights for new developments and research directions. Recommendations are presented as follows.

(i) We have demonstrated that using shadows from a single image alone is difficult to detect changes in LiDAR data. However, it is still worth to study the use of

shadows, as shadow information can be used as a supplementary change detection indicator if some buildings are occluded in a given image while the shadows casted by these buildings are not.

(ii) Identifying shadows is important when integrating LiDAR data and stereo images. If shadow intensity is largely affected by illumination and environmental reflections in images, our proposed shadow detection approach, Umbra method, in Chapter 3 can be combined with LEAD-Matching and E-LEAD-Matching of Chapter 4 and 5 to make the latter algorithms more robust to variations in different images obtained under different conditions.

(iii) 3D change detection and updating is performed in Chapter 4 using LiDAR data and one stereo pair. However, the main reason affecting the completeness of building change detection is the failure of extracting accurate 3D information from low (repetitive) texture, shadow and occlusion areas in the single stereo pair considered. This problem can be mitigated by additional stereo pairs from multi-view images. In addition, an advanced deep learning approach to learn effective textural and contextural features can be applied to further increase the accuracy of the 3D information extracted from multi-view images.

(iv) The main reason affecting the correctness of building change detection is the problem of distinguishing changes caused by small building-like objects, such as cars, trees and bushes from real building changes. A classification approach can be designed to distinguish building changes from other irrelevant changes. Multi-scale features should be extracted as objects in urban areas often have various sizes.

(v) In Chapter 5, we confirm that building boundaries integrated from LiDAR and multi-view image data improve the planimetric accuracy. Integrated geometric and color information can be further used for designing a building extraction method. Buildings extracted from the integrated data can be further combined with building boundary simplification and regularization to form a complete workflow for creating building polygons for maps. The results should be assessed to decide whether building polygons obtained in this way can meet the requirements of large scale topographic maps.

(vi) To determine accurate edges inside building roofs with high planimetric accuracy is important for creating an accurate 3D building model with detailed roofs. Ideally, our approach of integrating detailed image boundaries and accurate planes from LiDAR data is applicable for improving not only the building outer boundaries, but also edges inside of building polygons. However, inside building roofs, the occlusions and shadows are even more complicated, and textures of different roofs parts could be similar. An assessment of integrating 3D information from stereo images to improve the planimetric accuracy of edges inside buildings should be performed.

(vii) Our E-LEAD-Matching method depends on integrating multiple dense matching results on different stereo pairs. Another dense matching approach is to find corresponding pixels in multi-view images directly, such that the integration of LiDAR data with multi-view images becomes one single step. Therefore, a different integration approach can be designed. The famous patch-based multi-view stereo

dense matching method discussed in Furukawa and Ponce (2010) could be adapted to integrate LiDAR data with multi-view images in a single step.

(viii) Deep learning can be used in our framework to improve integration. First, deep learning with its capability to learn complicated effective contextural features can be applied to improve dense image matching in problematic areas. On the other hand, deep learning is also effective to learn not only plane information but more complicated geometric features from LiDAR data. In addition, our LEAD-Matching approach strongly relies on whether the accurate plane is included in candidate planes extracted using a plane segmentation method from LiDAR data. If an accurate plane fails to be extracted due to the sparsity of LiDAR points, integration will also fail to derive accurate 3D information, even though the 3D information extracted from stereo images alone is actually accurate. Deep learning can be applied to achieve intelligent and adaptive integration in different areas. Within such deep learning framework, integration can be applied to any laser scanning and imagery data from airborne, mobile or indoor platforms for 3D city or indoor modelling.

# Bibliography


Adeline, K., Chen, M., Briottet, X., Pang, S., and Paparoditis, N. (2013). Shadow detection in very high spatial resolution aerial images: A comparative study. *ISPRS Journal of Photogrammetry and Remote Sensing*, 80:21–38.

Arbel, E. and Hel-Or, H. (2011). Shadow removal using intensity surfaces and texture anchor points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6):1202–1216.

Awrangjeb, M. (2016). Using point cloud data to identify, trace, and regularize the outlines of buildings. *International Journal of Remote Sensing*, 37(3):551–579.

Awrangjeb, M., Ravanbakhsh, M., and Fraser, C. S. (2010). Automatic detection of residential buildings using LIDAR data and multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(5):457–467.

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.

Baillard, C. and Maıtre, H. (1999). 3-D reconstruction of urban scenes from aerial stereo imagery: a focusing strategy. *Computer Vision and Image Understanding*, 76(3):244–258.

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-[d] shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics.

Biljecki, F., Heuvelink, G. B., Ledoux, H., and Stoter, J. (2015). Propagation of positional error in 3D GIS: estimation of the solar irradiation of building roofs. *International Journal of Geographical Information Science*, 29(12):2269–2294.

Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1):2–16.

Bleyer, M., Rhemann, C., and Rother, C. (2011). Patchmatch stereo-stereo matching with slanted support windows. In *British Machine Vision Conference*, volume 11, pages 1–11.

Bovolo, F., Marchesi, S., and Bruzzone, L. (2012). A framework for automatic and unsupervised detection of multiple changes in multitemporal images. *IEEE Transactions on Geoscience and Remote Sensing*, 50(6):2196–2212.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Briem, G. J., Benediktsson, J. A., and Sveinsson, J. R. (2002). Multiple classifiers applied to multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 40(10):2291–2299.

Bromiley, P., Thacker, N., and Bouhova-Thacker, E. (2004). Shannon entropy, Renyi entropy, and information. *Statistics and Inf. Series (2004-004)*.

Chen, L., Teo, T., Rau, J., Liu, J., and Hsu, W. (2005). Building reconstruction from LIDAR data and aerial imagery. In *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS'05.*, volume 4, pages 2846–2849. IEEE.

Chen, L., Zhao, S., Han, W., and Li, Y. (2012). Building detection in an urban area using lidar data and QuickBird imagery. *International Journal of Remote Sensing*, 33(16):5135–5148.

Chen, Y., Gao, W., Widyaningrum, E., Zheng, M., and Zhou, K. (2018). Building classification of VHR airborne stereo images using fully convolutional networks and free training samples. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(4).

Chung, K.-L., Lin, Y.-R., and Huang, Y.-H. (2009). Efficient shadow detection of color aerial images based on successive thresholding scheme. *IEEE Transactions on Geoscience and Remote Sensing*, 47(2):671–682.

Cramer, M. (2010). The DGPF-test on digital airborne camera evaluation–overview and test design. *Photogrammetrie-Fernerkundung-Geoinformation*, 2010(2):73–82.

Cushnie, J. L. (1987). The interactive effect of spatial resolution and degree of internal variability within land-cover types on classification accuracies. *International Journal of Remote Sensing*, 8(1):15–29.

Dalla Mura, M., Benediktsson, J. A., Bovolo, F., and Bruzzone, L. (2008). An unsupervised technique based on morphological filters for change detection in very high resolution images. *IEEE Geoscience and Remote Sensing Letters*, 5(3):433–437.

Dare, P. M. (2005). Shadow analysis in high-resolution satellite imagery of urban areas. *Photogrammetric Engineering and Remote Sensing*, 71(2):169–177.

Dimitrov, R. (2007). Cascaded shadow maps. *Developer Documentation, NVIDIA Corp.*

Du, S., Zhang, Y., Qin, R., Yang, Z., Zou, Z., Tang, Y., and Fan, C. (2016). Building change detection using old aerial images and new LiDAR data. *Remote Sensing*, 8(12):1030.

Edelsbrunner, H., Kirkpatrick, D., and Seidel, R. (1983). On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559.

Fauvel, M., Chanussot, J., and Benediktsson, J. A. (2006). Decision fusion for the classification of urban remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(10):2828–2838.

Flood, M. (1999). Commercial development of airborne laser altimetry. *International Archives of Photogrammetry and Remote Sensing*, 32(Part 3):W14.

Förstner, W. and Wrobel, B. P. (2016). *Photogrammetric computer vision*. Springer.

Frénay, B. and Verleysen, M. (2014). Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376.

Fusiello, A. and Irsara, L. (2011). Quasi-euclidean epipolar rectification of uncalibrated images. *Machine Vision and Applications*, 22(4):663–670.

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

Gerke, M. and Xiao, J. (2014). Fusion of airborne laserscanning point clouds and images for supervised and unsupervised scene classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87:78–92.

Gilani, S., Awrangjeb, M., and Lu, G. (2016). An automatic building extraction and regularisation technique using lidar point cloud data and orthoimage. *Remote Sensing*, 8(3):258.

Gislason, P. O., Benediktsson, J. A., and Sveinsson, J. R. (2006). Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294–300.

Gorte, B. and van der Sande, C. (2014). Reducing false alarm rates during change detection by modeling relief, shade and shadow of multi-temporal imagery. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(4):65.

Gruen, A. and Akca, D. (2005). Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):151–174.

Gruen, A., Baltsavias, E., and Henricsson, O. (1995). *Automatic extraction of man-made objects from aerial and space images*. Springer Science & Business Media.

Guo, R., Dai, Q., and Hoiem, D. (2013). Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2956–2967.

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

Habib, A., Kwak, E., and Al Durgham, M. (2011). Model-based automatic 3D building model generation by integrating lidar and aerial images. *Archives of Photogrammetry, Cartography and Remote Sensing*, 22.

Habib, A., Zhai, R., and Kim, C. (2010). Generation of complex polyhedral building models by integrating stereo-aerial imagery and lidar data. *Photogrammetric Engineering and Remote Sensing*, 76(5):609–623.

Habib, A. F., Kim, E. M., and Kim, C. J. (2007). New methodologies for true orthophoto generation. *Photogrammetric Engineering and Remote Sensing*, 73(1):25–36.

Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341.

Huang, X., Qin, R., Xiao, C., and Lu, X. (2018). Super resolution of laser range data based on image-guided fusion and dense matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 144:105–118.

Isenburg, M. (2019). LAStools - efficient tools for LIDAR processing. `https://rapidlasso.com/lastools/`.

ISPRS (2019). Evaluation of 2D semantic labeling contest. `http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html`.

ISPRS WG III/4 (2019). ISPRS 2D semantic labeling contest. `http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html`.

Kaehler, A. and Bradski, G. (2016). *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc.".

Kumar, S. and Hebert, M. (2006). Discriminative random fields. *International Journal of Computer Vision*, 68(2):179–201.

Lafarge, F. and Mallet, C. (2012). Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. *International Journal of Computer Vision*, 99(1):69–85.

Lague, D., Brodu, N., and Leroux, J. (2013). Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (NZ). *ISPRS Journal of Photogrammetry and Remote Sensing*, 82:10–26.

Levin, A., Lischinski, D., and Weiss, Y. (2008). A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242.

Lin, C. and Nevatia, R. (1998). Building detection and description from a single intensity image. *Computer Vision and Image Understanding*, 72(2):101–121.

Lorenzi, L., Melgani, F., and Mercier, G. (2012). A complete processing chain for shadow detection and reconstruction in VHR images. *IEEE Transactions on Geoscience and Remote Sensing*, 50(9):3440–3452.

Marmanis, D., Schindler, K., Wegner, J. D., Galliani, S., Datcu, M., and Stilla, U. (2018). Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135:158–172.

Merchant, D. (1987). Spatial accuracy specification for large scale topographic maps. *Photogrammetric Engineering and Remote Sensing*, 53(7):958–961.

Miao, X., Heaton, J. S., Zheng, S., Charlet, D. A., and Liu, H. (2012). Applying tree-based ensemble algorithms to the classification of ecological zones using multi-temporal multi-source remote-sensing data. *International Journal of Remote Sensing*, 33(6):1823–1849.

Mirzaei, P. A. (2015). Recent challenges in modeling of urban heat island. *Sustainable cities and society*, 19:200–206.

Nicodemus, F. E. (1965). Directional reflectance and emissivity of an opaque surface. *Applied optics*, 4(7):767–775.

Niemeyer, J., Rottensteiner, F., and Soergel, U. (2014). Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87:152–165.

Ok, A. O. (2013). Automated detection of buildings from single vhr multispectral images using shadow information and graph cuts. *ISPRS Journal of Photogrammetry and Remote Sensing*, 86:21–40.

Okamoto, S. and Yugami, N. (1997). An average-case analysis of the k-nearest neighbor classifier for noisy domains. In *IJCAI (1)*, pages 238–245.

Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27.

Oude Elberink, S., Stoter, J., Ledoux, H., and Commandeur, T. (2013). Generation and dissemination of a national virtual 3D city and landscape model for the Netherlands. *Photogrammetric Engineering and Remote Sensing*, 79(2):147–158.

Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222.

PDOK (2019). Dutch national open source LiDAR point clouds: AHN. `https://www.ahn.nl/`.

Press, W. H. and Teukolsky, S. A. (1990). Savitzky-Golay smoothing filters. *Computers in Physics*, 4(6):669–672.

Qin, R. (2014). Change detection on LOD 2 building models with very high resolution spaceborne stereo imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 96:179–192.

Qin, R., Tian, J., and Reinartz, P. (2016). 3D change detection–approaches and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 122:41–56.

Remondino, F., Spera, M. G., Nocerino, E., Menna, F., and Nex, F. (2014). State of the art in high density image matching. *The Photogrammetric Record*, 29(146):144–166.

Rothermel, M., Wenzel, K., Fritsch, D., and Haala, N. (2012). SURE: Photogrammetric surface reconstruction from imagery. In *Proceedings LC3D Workshop, Berlin*, volume 8, page 2.

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.

Rottensteiner, F., Trinder, J., Clode, S., and Kubik, K. (2005). Using the Dempster–Shafer method for the fusion of LIDAR data and multi-spectral images for building detection. *Information Fusion*, 6(4):283–300.

Rottensteiner, F., Trinder, J., Clode, S., and Kubik, K. (2007). Building detection by fusion of airborne laser scanner data and multi-spectral images: Performance evaluation and sensitivity analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(2):135–149.

Sampath, A. and Shan, J. (2007). Building boundary tracing and regularization from airborne LiDAR point clouds. *Photogrammetric Engineering and Remote Sensing*, 73(7):805–812.

Sandau, R. (2009). *Digital airborne camera: introduction and technology*. Springer Science & Business Media.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42.

Schenk, T. (2005). Introduction to photogrammetry. *The Ohio State University, Columbus*, 106.

Schmidt, M. (2007). UGM: A Matlab toolbox for probabilistic undirected graphical models.

Shufelt, J. A. (1999). Performance evaluation and analysis of monocular building extraction from aerial imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):311–326.

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

Sirmacek, B. and Unsalan, C. (2008). Building detection from aerial images using invariant color features and shadow information. In *2008 23rd International Symposium on Computer and Information Sciences*, pages 1–5. IEEE.

Soetaert, K., Petzoldt, T., et al. (2010). Inverse modelling, sensitivity and monte carlo analysis in R using package FME. *Journal of Statistical Software*, 33(3):1–28.

Stal, C., Tack, F., De Maeyer, P., De Wulf, A., and Goossens, R. (2013). Airborne photogrammetry and lidar for DSM extraction and 3D change detection over an urban area–a comparative study. *International Journal of Remote Sensing*, 34(4):1087–1110.

Teo, T. and Shih, T. (2013). LiDAR-based change detection and change-type determination in urban areas. *International Journal of Remote Sensing*, 34(3):968–981.

Theodoridis, S., Koutroumbas, K., et al. (2008). Pattern recognition. *IEEE Transactions on Neural Networks*, 19(2):376.

Thomas, N., Hendrix, C., and Congalton, R. G. (2003). A comparison of urban mapping methods using high-resolution digital imagery. *Photogrammetric Engineering and Remote Sensing*, 69(9):963–972.

Tian, J., Chaabouni-Chouayakh, H., Reinartz, P., Krauß, T., and d'Angelo, P. (2010). Automatic 3D change detection based on optical satellite stereo imagery. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 586–591.

Tian, J., Cui, S., and Reinartz, P. (2014). Building change detection based on satellite stereo imagery and digital surface models. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):406–417.

Tolt, G., Shimoni, M., and Ahlberg, J. (2011). A shadow detection method for remote sensing images using VHR hyperspectral and LIDAR data. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, pages 4423–4426. IEEE.

Trinder, J. and Salah, M. (2012). Aerial images and LiDAR data fusion for disaster change detection. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:227–232.

Tsai, V. J. (2006). A comparative study on shadow compensation of color aerial images in invariant color models. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1661–1671.

United Nations (2007). World urbanization prospects the 2007 revision.

United Nations (2018). World urbanization prospects the 2018 revision.

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

Volpi, M., Tuia, D., Bovolo, F., Kanevski, M., and Bruzzone, L. (2013). Supervised change detection in VHR images using contextual information and support vector machines. *International Journal of Applied Earth Observation and Geoinformation*, 20:77–85.

Vosselman, G. (2002). Fusion of laser scanning data, maps, and aerial photographs for building reconstruction. In *IEEE International Geoscience and Remote Sensing Symposium*, volume 1, pages 85–88. IEEE.

Vosselman, G., Dijkman, S., et al. (2001). 3D building model reconstruction from point clouds and ground plans. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(3/W4):37–44.

Vosselman, G., Gorte, B. G., Sithole, G., and Rabbani, T. (2004). Recognising structure in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46(8):33–38.

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

Wald, I. and Havran, V. (2006). On building fast kd-trees for ray tracing, and on doing that in O (N log N). In *2006 IEEE Symposium on Interactive Ray Tracing*, pages 61–69. IEEE.

Wang, Q., Yan, L., Yuan, Q., and Ma, Z. (2017a). An automatic shadow detection method for VHR remote sensing orthoimagery. *Remote Sensing*, 9(5):469.

Wang, R. (2013). 3D building modeling using images and LiDAR: A review. *International Journal of Image and Data Fusion*, 4(4):273–292.

Wang, S., Bai, M., Mattyus, G., Chu, H., Luo, W., Yang, B., Liang, J., Cheverie, J., Fidler, S., and Urtasun, R. (2017b). Torontocity: Seeing the world with a million eyes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3028–3036. IEEE.

Wei, Y., Zhao, Z., and Song, J. (2004). Urban building extraction from high-resolution satellite panchromatic image using clustering and edge detection. In *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, volume 3, pages 2008–2010. IEEE.

Whitted, T. (1979). An improved illumination model for shaded display. In *ACM SIGGRAPH Computer Graphics*, volume 13, page 14. ACM.

Wimmer, M., Scherzer, D., and Purgathofer, W. (2004). Light space perspective shadow maps. *Rendering Techniques*, 2004:15th.

Wingtra (2019). Drone photogrammetry vs. LIDAR: what sensor to choose for a given application. `https://wingtra.com/drone-photogrammetry-vs-lidar/`.

Wu, J., Jie, S., Yao, W., and Stilla, U. (2011). Building boundary improvement for true orthophoto generation by fusing airborne LiDAR data. In *Proceedings of Joint Urban Remote Sensing Event*, pages 125–128. IEEE.

Xiao, C., She, R., Xiao, D., and Ma, K.-L. (2013). Fast shadow removal using adaptive multi-scale illumination transfer. In *Computer Graphics Forum*, volume 32, pages 207–218. Wiley Online Library.

Yuan, J. (2018). Learning building extraction in aerial scenes with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11):2793–2798.

Zebedin, L., Klaus, A., Gruber-Geymayer, B., and Karner, K. (2006). Towards 3D map generation from digital aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(6):413–427.

Zhang, K., Chen, S.-C., Whitman, D., Shyu, M.-L., Yan, J., and Zhang, C. (2003). A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4):872–882.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22.

Zhou, G., Chen, W., Kelmelis, J. A., and Zhang, D. (2005). A comprehensive study on urban true orthorectification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(9):2138–2147.

Zhou, K., Chen, Y., Smal, I., and Lindenbergh, R. (2019a). Building segmentation from airborne VHR images using mask R-CNN. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(2/W13).

Zhou, K. and Gorte, B. (2017). Shadow detection from VHR aerial images in urban area by using 3D city models and a decision fusion approach. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42.

Zhou, K., Gorte, B., Smal, I., and Lindenbergh, R. (2020a). E-LEADMatching—Improving the planimetric accuracy of ALS-derived building boundaries using airborne multi-view images. on review.

Zhou, K., Lindenbergh, R., and Gorte, B. (2019b). Automatic shadow detection in urban very-high-resolution images using existing 3D models for free training. *Remote Sensing*, 11(1):72.

Zhou, K., Lindenbergh, R., Gorte, B., and Zlatanova, S. (2020b). LiDAR-guided dense matching for detecting changes and updating of 3D buildings in LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:200–213.

Zhou, Q. and Neumann, U. (2009). A streaming framework for seamless building reconstruction from large-scale aerial lidar data. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2759–2766. IEEE.

Zhou, Q. and Neumann, U. (2012). 2.5 D building modeling by discovering global regularities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–333. IEEE.

Zhu, Q., Hu, M., Zhang, Y., and Du, Z. (2009). Research and practice in three-dimensional city modeling. *Geo-spatial Information Science*, 12(1):18–24.

# Curriculum Vitæ

## Kaixuan Zhou

| | |
|---|---|
| 22-09-1991 | Born in Jiangsu, China. |

## Education

| | |
|---|---|
| 2009–2013 | Bachelor in Geomatics Engineering<br>Central South University, China |
| 2013–2015 | Master in Geomatics<br>Delft University of Technology, the Netherlands |
| 2016– | PhD. Geoscience and Remote Sensing<br>Delft University of Technology, the Netherlands<br>*website:* http://3dkaixuanzhou.com<br>*Thesis:* Combining LiDAR and Photogrammetry to Generate Up-to-date 3D City Models<br>*Promotor:* Prof. Dr. Ir. R.F. Hanssen<br>Dr. R.C. Lindenbergh |
| Feb.–Mar. 2015 | Exchange Program of WHU-TU Delft Joint Research Center<br>Wuhan University, China |
| Jul.–Sept. 2019 | Study Abroad Research Practicum program<br>University of New South Wales, Australia |

## Working Experience

| | |
|---|---|
| 2014 | Intern<br>CycloMedia Technology B.V., the Netherlands |

## Awards

| | |
|---|---|
| 2017 | Best Youth Oral Paper Award, ISPRS Geospatial Week |
| 2018 | Best Poster Award, ISPRS TC-IV Symposium |

# List of Publications

### Peer-viewed journals

- **Zhou, K.**, Gorte, B., Smal. I., & Lindenbergh, R. Improving the Planimetric Accuracy of ALS-Derived Building Boundaries using Airborne Multi-view Images. under review.(Chapter 5)
- **Zhou, K.**, Lindenbergh, R., Gorte, B., & Zlatanova, S. (2020). LiDAR-guided dense matching for detecting changes and updating of 3D buildings in LiDAR data. ISPRS Journal of Photogrammetry and Remote Sensing, 162, 200-213.(Chapter 4)
- **Zhou, K.**, Lindenbergh, R., & Gorte, B. (2019). Automatic Shadow Detection in Urban Very High Resolution Images Using Existing 3D Models for Free Training. Remote Sensing, 11(1), 72. (Chapter 3)

### Peer-viewed conference proceedings

- **Zhou, K.**, Chen, Y., Smal, I., & Lindenbergh, R. (2019). Building segmentation from airborne VHR images using mask R-CNN. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42.
- Chen, Y., Gao, W., Widyaningrum, E., Zheng, M., & **Zhou, K.** (2018). Building classification of VHR airborne stereo images using fully convolutional networks and free training samples. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42. (Corresponding author, **Best Poster Award**, ISPRS TC-IV Symposium 2018)
- Gorte, B., **Zhou, K.**, Van Der Sande, C., & Valk, C. (2018). A computational cheap trick to determine shadow in a voxel model. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4.
- **Zhou, K.**, Gorte, B., Lindenbergh, R., & Widyaningrum, E. (2018). 3D Building change detection between current VHR images and past LiDAR data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42.
- **Zhou, K.**, & Gorte, B. (2017). Shadow detection from VHR aerial images in urban area by using 3D city models and a decision fusion approach. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42. (**Best Youth Oral Paper Award**, ISPRS Geospatial Week 2017)
- **Zhou, K.**, Gorte, B., & Zlatanova, S. (2016). Exploring Regularities for Improving Façade Reconstruction from Point Clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 5.