# Photogrammetric point clouds: quality assessment, filtering, and change detection

Zhenchao Zhang

# Photogrammetric point clouds: quality assessment, filtering, and change detection

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof.dr.ir. A. Veldkamp
on account of the decision of the Doctorate Board,
to be publicly defended
on Friday, January 14th, 2022 at 12.45 hrs

by

Zhenchao Zhang

born on October 6, 1991

in Anhui, China

This thesis has been approved by
**Prof.dr.ir M.G. Vosselman**, supervisor
**Prof.Dr.-Ing. M. Gerke**, supervisor
**Dr. M.Y. Yang**, co-supervisor

ITC dissertation number 409
ITC, P.O. Box 217, 7500 AE Enschede, The Netherlands

ISBN 978-90-365-5265-3
DOI 10.3990/1.9789036552653

Cover designed by Zhenchao Zhang and Job Duim
Printed by CTRL-P Hengelo
Copyright © 2022 by Zhenchao Zhang

Graduation committee:

**Chairman/Secretary**

| | |
|---|---|
| Prof.dr. F.D. van der Meer | |

**Supervisors**

| | |
|---|---|
| Prof.dr.ir. M.G. Vosselman | University of Twente / ITC |
| Prof.Dr.-Ing. M. Gerke | Technische Universität Braunschweig / IGP |

**Co-supervisor**

| | |
|---|---|
| Dr. M.Y. Yang | University of Twente / ITC |

**Members**

| | |
|---|---|
| Prof.dr. A. Nelson | University of Twente / ITC |
| Dr. C. Persello | University of Twente / ITC |
| Prof.dr. B. Yang | Wuhan University / LIESMARS |
| Prof.Dr.-Ing. P. Reinartz | DLR / IMF |

# Acknowledgements

In the morning of October 28th, 2015, I arrived at Enschede and started my Ph.D. journey at ITC. I spent forty-eight months in Enschede and the remaining twenty-four months in Zhengzhou, where I work as a teacher after my PhD journey. During the four years in The Netherlands, life is rather repetitive, but knowledge and ideas are new day by day. Thanks to the support and assistance from my promoters and friends. I learned to infer logically, to analyze thoroughly, to conclude accordingly, to think critically and to write precisely. After a long journey coping with difficulties, I made progress step by step and came towards the end of this journey.

First and foremost, I greatly thank my Ph.D. promoter, Prof. George Vosselman, for giving me this opportunity. George is rigorous and careful with each detail relating to doing research. His comprehensive and profound knowledge in laser scanning and photogrammetry supports me throughout the five years. He is strict and dedicated to his students. I learn from meeting with him, and from his comments on my academic papers and progress reports. In my future career, I will try my best to be as responsible, perseverant and hard-working as him.

I am very thankful to my Ph.D. co-promoter, Dr. Michael Ying Yang. He always encourages me to be creative and hardworking during my research. Whenever I need help, he is always there and gives me much support. He supports me and helps me in both research and life. He enlightens me with new ideas when I get stuck in my research. He reminds me of stepping out my "comfort zone" and try new and innovative research ideas. Michael also provides me with much assistance during my stay in China for the last two Ph.D. years. When I got extremely busy with my daily work and life in China, he provides me with huge assistance and guides to finish the Ph.D. thesis.

I thank my Ph.D. promoter Prof. Markus Gerke for his patient and consistent support during my long Ph.D. journey. I met Markus the first day I arrived at ITC. He showed me with multiple potential Ph.D. topics, and then we selected "change detection and point cloud updating" together with George. During the remaining four years, he gave me much insightful supervision in data preparation, method design and scientific writing. During the three-month exchanging program in Technical University of Braunschweig, Markus assisted me in preparing the experimental data of Rotterdam. Last but not the least, I learn from Markus to balance life and work. That is, be rigorous and hardworking in work and research; Stay colorful and laugh loudly in life.

I am also thankful for Prof. Andy Nelson, Dr. Claudio Persello, Prof. Bisheng Yang and Prof. Peter Reinartz for being in my thesis committee. I would like to show great gratitude to Teresa, Jenny, Michael (Dr. Michael Peter), Claudio, Francesco, Prof. Alfred Stein, Valentyn, Ville, Frank, Wan and other EOS staffs for their support and help during my Ph.D. journey. I thank Prof. Devis Tuia for his insightful suggestions to my research.

I would like to show thanks to my Chinese friends in ITC. They are Fashuai Li, Mengmeng Li, Qian Lu, Biao Xiong, Man Qi, Jinfeng Mu, Yifei Xue, Sudan Xu, Peng Jia, Yaping Lin, Ye Lyu, Wufan Zhao, Bin Zhang, Yizhen Lao, Yancheng Wang, Deliang Gao, Yunmeng Zhu, Shuai Hao, Shengce Wang, Wanpeng Shao, Ling Chang, Chunjing Yao, Tonggang Zhang, Peiqi Yang, Jing Liu, Yuhang Gu, Hong Zhao, Yifang Shi and Chengliang Liu. We shared the best ITC times. Thank Tina Tian, Tiejun Wang for their support and help. Thanks also go to my Chinese friends Jinhu Wang and Kaixuan Zhou in Technical University of Delft.

I feel very lucky to have warm and supportive officemates. They are Anand, Diogo, Shayan, Yanwen Wang and Zhishuang Yang. We have shared news and knowledge together and made our office as a small family. Throughout my EOS life, I have met a group of lovely people. Without them, my Ph.D. journey will be gloomy. In the beginning of my Ph.D. journey, they helped me with my spoken English and English writing skills. In the later stage, I learned the cultural diversity from them. The gratitude goes to Caroline, Phillipp, Zill, Joep, Anand, Claudia, Andrea, Magnus, JR, Sophie, Samer, Sofia and other colleagues.

Last but not least, I would like to give sincere gratitude to my grandparents, my parents, Anqi Liu and other family members. With their support and love, I came to the Netherlands and worked on my PhD journey. Wherever I am, their care always supports me to work devotedly and go bravely.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1 – Introduction

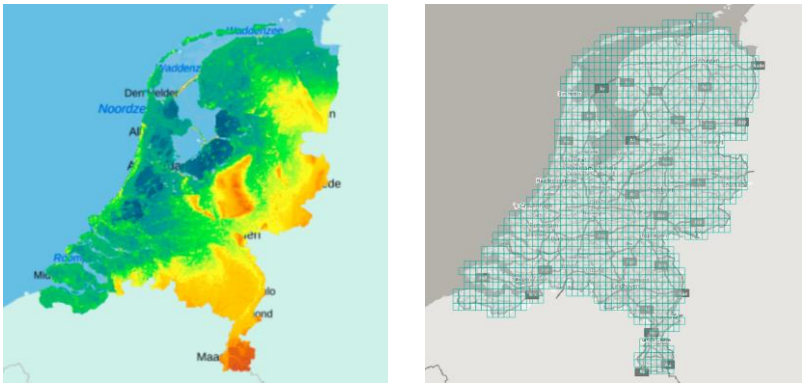
# 1.1 Background

Change detection is the process of acquiring changes in an object or phenomenon from the remote sensing data of two epochs (Lu and Weng, 2007; Qin et al., 2016). Change detection is one critical research topics in the field of photogrammetry and remote sensing. 3D change detection draws more and more attention in recent years due to the increasing availability of 3D data. The acquisition and utilization of 3D data are becoming common. For example, 3D data may present as multi-view imagery from satellite, airborne, or close-range platforms, point cloud obtained from laser scanning or photogrammetric dense matching, Digital Surface Model (DSM), topographical models, depth images from RGB-D cameras, etc.

3D change detection has been widely used in many fields and is of great significance to urban planning and administrative management, for example, land use / land cover (LULC) change detection, geographic information updating, point cloud updating, terrain deformation analysis, disaster analysis, urban construction monitoring, target tracking, vegetation growth status monitoring (Choi and Lee, 2009; Kim et al., 2013; Miller et al., 2000; Rebolj et al., 2008; Torres-Sánchez et al., 2014).

At present, aerial photogrammetry and Airborne Laser Scanning (ALS) are two leading techniques for point cloud acquisition in the city level. This study takes the case of The Netherlands as our motivation. The two techniques are both used in The Netherlands for topographic data acquisition. It is common that point clouds from two epochs are acquired by different techniques. Therefore, it is necessary to study 3D change detection between multimodal point clouds.

The Netherlands started the "Actueel Hoogtebestand Nederland (AHN)" project in 1997, which aimed to obtain elevation data covering the whole country. Every five to seven years, the AHN data were updated completely. Now the AHN project comes to its fourth generation: AHN1 (1997-2004), AHN2 (2007-2012), AHN3 (2014-2019), AHN4(2020-2022). From AHN1 to AHN4, the vertical accuracy of the point cloud increases, and the semantic labels of the provided point clouds get finer. The vertical accuracy of AHN2 reaches ±5 cm with a density of 20 points/$m^2$ (Van Der Sande, 2010). The point clouds are classified into five categories: ground, building, water, infrastructure, and other. Figure 1.1(a) shows the DSM with an interval of 5 m generated from AHN2. Figure 1.1(b) shows how the AHN data are organized and managed.



*(a) AHN3 DSM in 5 m interval* *(b) Data organization*

***Figure 1.1:*** *The Actueel Hoogtebestand Nederland project.*

Meanwhile, the Netherlands are acquiring aerial imagery for most cities as the latest topographical data every year. Compared with laser scanning, airborne photogrammetry is cheaper and more efficient towards 3D data generation. While laser scanning obtains geometric information, digital photogrammetry generates geometric, spectral and textual information (Gerke and Xiao, 2013; Nebiker et al., 2014; Nex, 2015). The highly overlapping images can fully cover the complicated urban scene from multiple views compared to traditional single-view photogrammetry, both in facades and roofs. Moreover, the dense 3D points derived from dense image matching (DIM) can meet the density required to update laser point clouds (Baltsavias, 1999; Haala and Rothermel, 2012).

As fundamental topographic data, urban point cloud data should be updated to meet the latest demand on decision making and urban planning. In large cities, 95% of the landscapes stay unchanged in five to ten years (Frontoni et al., 2006). Change detection is the most important process of the entire updating, while it's also very time-consuming and tedious. According to Champion (2007), change inspection and detection take up to 40% of the time in the workflow, compared to other 60% of charting and updating. To keep the point cloud data up-to-date, we are looking forward to automating the change detection process (Vosselman et al., 2004; Xu et al., 2015).

The main motivation of this thesis is to enable updating of outdated ALS points with dense image matching points. The question rises as whether the up-to-date photogrammetric data are qualified to update the outdated ALS data. Suppose that airborne photogrammetry and laser scanning acquire 3D point clouds with the same quality, updating the AHN data with photogrammetry would save much labor cost and time. In order to update outdated ALS points, various aspects are required for this purpose:

(1) We should know the accuracy of dense image matching;
(2) We should extract a DTM from DIM points;
(3) We should detect changes between ALS and DIM data. Changes are detected and then the point clouds are updated where changes have happened.

Change detection requires datasets from two epochs. Figure 1.2 shows the input and output for our research. The outdated datasets to be updated are ALS point clouds. The up-to-date data we utilize to detect changes are multi-view airborne images, point clouds generated from DIM and the orthoimages rectified from multi-view images. Classic change detection techniques usually obtain a binary change map, which indicates changed or non-changed for each point or object. The target of our research is not only to detect where changes happened, but also to recognize the change types, e.g. a new building, a demolished building, or a cropped tree. The fine change types can better serve the applications in topographic data updating, disaster rescue, damage management, and urban planning.

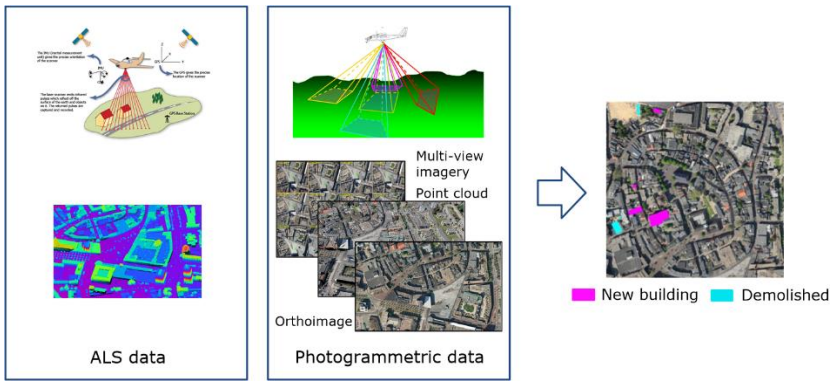



***Figure 1.2:*** *The input and output for Multimodal change detection addressed in this thesis.*

Concerning 3D change detection, current change detection techniques are usually applicable to 3D data of the same modality but not multimodal data. How to quickly obtain 3D data, how to quickly assess the quality of 3D data and how to quickly detect changes between multimodal 3D data are three major challenges that restrict the wide application of 3D change detection (Qin et al., 2016). Before we further elaborate the objectives, we give a more detailed description of the principles and data properties of airborne laser scanning and aerial photogrammetry.

## 1.2 Airborne laser scanning and photogrammetry

Obtaining high-quality 3D data is the prerequisite for performing robust change detection. 3D data are being collected from multi-baseline Synthetic Aperture Radar (SAR) interferometry (Zhu et al., 2018), satellite laser altimeter (Li et al., 2020), airborne platforms, mobile platforms, wearable devices, and ubiquitous handheld sensors. Compared to 2D data, 3D data are closer to real world and thus might be easier to interpret by human eyes or computers.

### *1.2.1 Airborne laser scanning*

ALS is the main data source for surveying and mapping, city management and urban planning, rescue response, and change detection in the city-level, which allows effective data acquisition from the top-view. Compared to other laser scanning platforms, airborne laser scanner collects object points from an altitude from 400 m to 1200 m, the problem of object occlusion can be largely alleviated through swath overlapping. It belongs to active remote sensing and can be applied at night. ALS can not only collect points from the canopy surface but also penetrate through the forest canopy. Therefore, the point cloud can be used to generate DSMs and DTMs even in forestry area.

ALS is performed on a fixed wing aircraft, a helicopter, or an Unmanned Aerial Vehicle (UAV). The technique is supported by two major systems: a laser scanner to measure the distance to object and a GPS/IMU unit to measure the position and orientation of the system. A typical airborne laser scanner contains the following components: scanner assembly, airborne GPS antenna, Inertial measurement unit (IMU), control and data recording unit, operator laptop, and flight management system (Vosselman and Maas, 2010).

Scanner assembly emits laser pulses from the aircraft's fuselage and their echoes are recorded during the flight. GPS antenna is a dual frequency antenna which records GPS signals. It is mounted at the top of the aircraft to record the system position. The IMU records acceleration data and rotation rates to determine the GPS trajectory and platform orientation. The control and data recording unit is used for synchronizing the time and managing the system. An operator laptop is taken to for flight planning and mission monitoring. The flight management system is used to display the flight lines (Vosselman and Maas, 2010).

Figure 1.3(a) shows the Leica TerrainMapper-2 as one of the latest linear-mode LiDAR airborne sensors. It includes a 2 MHz LiDAR sensor combined with two nadir 150 MP cameras in RGB and NIR. It is operated at a height from 300 m to 5000 m. The laser wavelength adopted by it is 1064 nm. The sensor can be upgraded with four additional oblique cameras turning the system into Leica CityMapper-2 for 3D city mapping. The vertical accuracy of the obtained point

clouds reaches 0.03 m, while the planimetric accuracy reaches 10 cm at a nominal flying height of 1000 m AGL.

(a) Leica TerrainMapper-2 (b) RIEGL VQ-880-GH

***Figure 1.3:*** *Examples of two airborne laser scanners.*

Figure 1.3(b) shows the RIEGL VQ-880-GH system, which can be mounted on a helicopter or aircraft with a height less than 1600 m. It applies both near-infrared channel of 1064 nm and green channel of 532 nm, which allows for combined hydrographic and topographic surveying. The pulse frequency ranges from 150 kHz to 900 kHz. The typical positioning accuracy reaches ±0.025 m at 150 m testing range.

In general, laser scanners are developing towards the direction of light-weighted and small hardware, high accuracy, and low cost. Another general trend of laser scanners is towards simultaneous acquisition of point cloud and imagery, which allows to generate topographic products with both high accuracy and spectral information.

### *1.2.2 Aerial photogrammetry*

Photogrammetry is the science or art for acquiring geometric information from the objects through photography (Thompson, 1966). With the development of digital aerial cameras, digital photogrammetry becomes an effective solution for urban topographic mapping. Aerial photogrammetry starts with aerial imagery acquisition where overlapping aerial images are obtained from cameras mounted on a fix-wing aircraft, helicopter, UAV, etc. The initial interior orientation elements are obtained through camera calibration before the flight. The initial exterior orientation elements are obtained by On-board GPS and Inertial Measurement Unit (IMU). They are refined with ground control points (GCPs) marked on the images through aerial triangulation (AT).

Development in dense matching algorithms, e.g. Patch-based Multi-View Stereo (PMVS) (Furukawa and Ponce, 2010) and Semi-global Matching (SGM) (Hirschmüller, 2008) makes it possible to obtain accurate point cloud. nFrames SURE states that the vertical accuracy of their products can be better than 1 pixel (Rothermel and Haala, 2012). The dense 3D object points are calculated

by forward intersection. Image quality and image overlap rate during acquisition both affect the quality of dense matching point cloud.

Digital photogrammetry produces DSM, DTM generation, Orthoimages and true orthoimages, 3D models and Digital Linear Graphs (DLGs). Digital Surface Models (DSMs) and Digital Terrain Models (DTMs) are generated through interpolation over dense point clouds. Concerning ortho-rectification, DSMs are used to rectify the images so that each pixel is assigned with an interpolated position and intensity value. ortho-rectification changes the imaging geometry from perspective projection to orthographic projection and eliminates the impact of camera tilt and terrain relief. The individual ortho-images can be stitched to obtain a large-format orthoimage. When the texture from images is projected to the TIN, a 3D topographic model is generated.

Figure 1.4(a) shows the Leica ADS100 camera which applies 20,000 pixel linear CCD for forward, nadir and backward imaging. Figure 1.4(b) shows the large-format digital aerial camera IGI UrbanMapper-2. It has eight lenses which obtains RGB and near-infrared images from nadir and oblique view. When operated at a height of 500 m, the nadir GSD can be as fine as 2 cm while the oblique GSD reaches 2.7 cm.

(a) Leica ADS100 (b) IGI UrbanMapper-2

***Figure 1.4:*** *Examples of eight state-of-the-art digital aerial cameras*

ADS100 and IGI UrbanMapper-2 both obtain oblique images in addition to nadir images. In recent years, multi-view oblique images are being widely acquired in aerial photogrammetry. When oblique images are acquired, achieving full coverage of the terrain gets easier and data acquisition becomes more efficient. Oblique imagery also allow better coverage on the building façades.

The density of dense matching points is influenced by GSD. If Semi-Global Mapping is utilized for dense matching, the corresponding pixels will be matches pixel-by-pixel in the Region of Interest (ROI). Haala (2015) reported that most dense matching software in the test can achieve a density of 100 pts/$m^2$ and a DSM grid spacing of 10 cm given that the image GSD is 10 cm.

**Airborne Laser Scanning vs. Airborne Photogrammetry**

The techniques of airborne laser scanning and photogrammetry differ in the following aspects:

(1) **Sensor type:** Airborne Laser Scanning belongs to active remote sensing, so the laser emitter sends out laser beams by itself; Airborne photogrammetry is passive remote sensing, which records the sunlight reflected by the objects.

(2) **Imaging geometry:** The laser scanning collects data based on the time light travels from the emitter to the object; Photogrammetry applies perspective projection.

(3) **Coverage:** Laser scanning covers the study area by point-by-point sampling, while photogrammetry covers a large area with every exposure of the camera's CCD-chip.

(4) **Point cloud generation**: Laser scanning generates point clouds from the received echoes and DGPS/IMU data. Photogrammetry generates point clouds through dense image matching and forward intersection.

(5) **Other features obtained**: Airborne laser scanning may obtain intensity, full-waveform data along with the point clouds depending on the hardware. Airborne photogrammetry obtains images with spectral and textural features.

(6) **Major topographic products:** The major topographic products of ALS include DSM, DTM, 3D model, etc. Apart from the products generated by ALS, digital photogrammetry can also be used to obtain orthoimages.

(7) **Technical Maturity:** The technique of airborne laser scanning is relatively new compared with aerial photogrammetry. The algorithms for point cloud filtering, semantic segmentation, 3D modelling, change detection and full-waveform analysis are still under rapid development. The technique of photogrammetry is relatively mature with a history of more than 150 years. Some mature software are available for photogrammetric data processing.

(8) **Weather conditions:** ALS can be used at night, but not in the rainy, cloudy or sandstorm weather since laser beam cannot penetrate these obstacles (Leberl et al., 2010; Priestnall et al., 2000). Digital images are always acquired under strict weather requirements in the daytime.

(9) **Vegetation penetration:** ALS can penetrate sparse vegetation for DTM generation. Digital photogrammetry cannot penetrate dense vegetation.

Concerning efficiency, ALS is relatively time-consuming in flight planning towards full terrain coverage, but fast in point cloud generation. Airborne photogrammetry is efficient in image acquisition but takes much effort in aerial triangulation and dense matching.

(10) **Efficiency:** ALS is relatively time-consuming in data acquisition to cover the complete terrain with scanned pulses but is fast in point cloud generation. Digital photogrammetry is relatively efficient in image acquisition, but time-consuming in aerial triangulation and dense matching.

## 1.3 Research problems

### 1.3.1 Problems with multimodal data

Change detection between ALS data and photogrammetric data is a special case in 3D change detection. 3D data of two epochs are acquired by different sensors and present different properties. Multimodal point cloud change detection requires different feature extraction methods for different epochs, so it is more challenging than single-modal change detection. In this section, we summarize the differences between multimodal data and their impact on change detection.

Figure 1.5 illustrates the differences between laser scanning points and dense matching points.

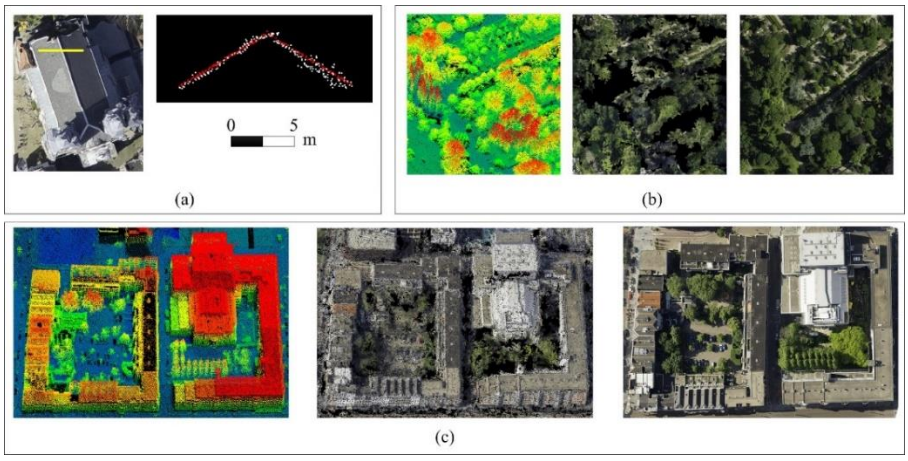



***Figure 1.5:*** *Comparison between point clouds from airborne laser scanning and dense image matching. All samples are from our study area. (a) Roof surface. The ALS (red) and DIM (white) points represent the yellow profile on the left panel; (b) Vegetated land; (c) Building with trees in courtyards. From*

*left to right in (b) and (c): ALS point cloud, DIM point cloud from bird-view and the orthoimage. The color coding from blue to red in the ALS point cloud indicates increasing height.*

Some differences are related to data accuracy:

(1) **Vertical deviation:** (Zhang et al., 2018a) reported that the vertical accuracy of ALS points is better than ±5 cm, while the vertical accuracy of dense matching points produced by state-of-the-art dense matching algorithms can be better than 1 Ground Sampling Distance (GSD), which is usually 10-20 cm for airborne platforms. The dense matching errors from Semi-Global Matching (Hirschmüller 2008; Rothermel et al. 2012) in a whole block show a normal distribution. Therefore, it is hard to set a single threshold to differentiate between the two point clouds.

(2) **Horizontal deviation:** ALS points and DIM points are acquired by different techniques and their horizontal accuracies differ. Horizontal deviations between ALS points and DIM points usually cause elongated false positives along building edges in the DSM differencing map.

Other differences include: (3) **Noise level:** On smooth terrain and roof surfaces, dense matching points usually contain much more noise than the laser scanning points. Dense matching is problematic when the image contrast is poor, e.g. in shadow areas, whereas low contrast or illumination is not a problem for laser data acquisition. The breaklines and topographic edges in the ALS data are clear and sharp but fuzzy in the DIM points. The noise and errors from dense matching cannot be simulated and are not uniform throughout the study area, thus hindering change detection.

(4) **Density difference:** The density of ALS points and DIM points usually differs. As reported in Section 1.2.1 and Section 1.2.2, the densities of ALS data and DIM data are 10 pts/$m^2$ and higher than 100 pts/$m^2$, respectively, in our data sets. On the one hand, manipulation of dense DIM point clouds usually requires high computation power; On the other hand, the imbalance of densities in the two point clouds usually requires point cloud re-sampling or data normalization so that their densities are unified before change detection.

(5) **Data gaps:** Data gaps exist in both data types. The data gaps in laser points mainly occur due to occlusion or pulse absorption by the surface material, e.g., water (Xu et al., 2015), while data gaps occur in dense matching points mainly due to poor contrast. Some small or thin objects can be recorded in the laser scanning data, e.g. wires, poles or traffic lights, while they are usually invisible in the DIM points.

(6) **Point distribution:** Point clouds are unordered, permutation invariant, and unevenly distributed (Qi et al, 2017a). They are different from raster data such as images. Therefore, the proposed algorithms for 3D change detection should be robust to point permutation and uneven distribution.

Comparing the point clouds distributed on trees, the laser points are distributed over the canopy, branches and the ground below, while for dense matching usually only points on the canopy are generated. Specifically, the point density on the trees from dense matching is largely affected by the image quality, seasonal effects and leaf density.

### *1.3.2 Uncertainty in change definition*

In order to automatically determine the type of changes, it is necessary to clarify which type of changes is relevant in the beginning of research, namely, *changes of interest*. Topographic changes are divided into three types: relevant changes, falsified changes and irrelevant changes.

(i) **Relevant changes**: These are the types of change to our interest. It usually includes terrain deformation, building changes (e.g. height changes, a new building or demolition), tree changes (e.g. a new tree or a cropped tree), etc. When a relevant change is mis-classified into other types of changes, it causes a commission error or False Positive (FP); The omission of a relevant change causes omission error or False Negative (FN). In addition, if a relevant change is detected correctly, this is True Positive (TP). If a non-relevant change is detected as "non-changed", this leads to True Negative (TN). The target of our change detection algorithms is to maximize the recall and precision for relevant changes.

(ii) **Falsified changes**: Change does not happen to the object in reality, but a *falsified change* is detected by the algorithm due to data problems or algorithmic flaws (Xu et al., 2015). For example, the height of DIM point cloud at a shaded street corner is usually higher than the true height. When implementing DSM difference between ALS points and DIM points, the surface differencing results would mistakenly indicate that there is a terrain change in the corner.
In addition, surface differencing between two DSMs usually causes linear artefacts along the building edges due to mis-registration errors. These are also *falsified changes*.

(iii) **Irrelevant changes**: These changes happen in reality and also present in the two datasets, but are not the changes we are interested in. For example, leaves grow and fall in different seasons, vehicles and pedestrians moving, water surface fluctuation, a newly-built scaffold, a new container in ports, etc.

The definition of relevant and irrelevant changes is determined on user application and data quality. For example, vegetation height change is an important relevant change in precision agriculture. In our project of topographic change detection and updating, the normal growth of vegetation is not interested. The quality of remote sensing data determines the fine level of the change types. For example, DIM points generated by a UAV allow to detect object changes finer than 5 cm; While DIM points from an airborne aircraft may only allow to detect changes larger than 10 cm.

### 1.3.3 Problems with scenes complexity

The object changes might be complicated and diversified in real scene:

(1) Concerning building changes, false positives may occur if the shape of a changed object is similar to a building. For example, moving of scaffolds or trucks in construction sites might be mis-classified into building changes because these object surfaces look similar.

(2) In addition, changes on a complex building might be mixed. For instance, one part of a building might be demolished while the other part remains unchanged. The algorithm should be advanced enough to detect these detailed changes.

(3) In urban scenes, the buildings might be adjacent to trees or high vegetation. If they are closely connected, the semantic segmentation method may misclassify their pixels into each other and not make a clear distinguishment (Gerke and Xiao, 2013). The semantic segmentation errors are propagated to the change detection results.

(4) The distribution of pixels for different topographic objects (e.g. buildings, vegetation, terrain, water) is usually uneven. For example, there are usually more terrain pixels than building pixels in the suburb region, and the numbers of pixels for different topographic objects are imbalanced. This poses a major problem in the machine learning classification algorithm (Buda et al., 2018). These errors will be propagated to change detection.

## 1.4 Research objectives and research questions

### 1.4.1 Research objectives

Motivated by the need for point cloud updating, the main objective of the thesis is to assess the quality of photogrammetric point clouds and detect changes between them and ALS data. This is achieved step-by-step through the following four sub-objectives:

(1) Evaluate the quality of dense matching point clouds to check whether the accuracy and noise level of DIM points meets the needs for change detection and potential point cloud updating.

(2) Assess the performance of LiDAR filters when the filters are applied to dense matching point clouds. We also aim to evaluate the quality of digital terrain models (DTMs) derived from dense matching point clouds and check if it can meet the requirements of change detection.

(3) Concerning change detection algorithms, the third sub-objective is to propose an algorithm to detect changes between ALS points and DIM points. The algorithm should be capable of detecting versatile change types, e.g. a new building, a demolished building and a heightened building.

(4) Since point cloud semantic segmentation and change detection are two associated tasks, the fourth sub-objective is to propose a method to fulfil the two tasks in one framework. The method not only outputs semantic label for each point, but also its change label.

Comparing sub-objective (1) and (2), sub-objective (1) directly evaluates DIM point clouds while sub-objective (2) evaluates the DIM point cloud filtering and the derived DTMs. After achieving the four sub-objectives, we can answer whether the quality of dense image matching points allows us to update the ALS points, and whether the change detection algorithms are effective to cope with the above change detection challenges. In addition, we may conclude whether the tasks of semantic segmentation and change detection can both be achieved in a single framework.

Concerning 3D change detection methods, some existing methods have demonstrated their performance in 3D change detection, such as surface-based differencing or post-classification methods (Xu et al., 2015). These methods are effective when the point clouds of two epochs are both from Airborne Laser Scanning. However, the DIM data usually contain more noise than the ALS data. When the data of one epoch are ALS data and the data of the other epoch are DIM data, the problem becomes multimodal change detection. The method should be robust towards the DIM noise. It should also cope with the difference between multimodal inputs.

Neural networks have demonstrated their superior performance in various computer vision tasks, for example in the 2D and 3D semantic segmentation (Krizhevsky et al., 2012; Qi et al., 2017a). Neural networks extract high-level features from images or point clouds by aggregating features from shallow layers to deep layers. The change detection methods proposed in our thesis are inspired by the recent progress of Neural Networks applied in the remote sensing and computer vision domain.

Since 2D Fully Convolutional Networks (FCN) have been used in image-to-image change detection (Zhan et al., 2017), we come up with transferring 2D CNNs to point cloud-based change detection. However, point clouds are 3D while images are 2D. In order to extract deep features with CNN from point clouds, point clouds should be represented and converted to 2D space. We convert the point clouds of two epochs to DSMs and concatenate them with the RGB channels, this specific "tensor" structure can be read and processed by a CNN. By this means, the change detection task is converted to a binary classification problem. This leads to our first change detection method, i.e. patch-based change detection.

The output from patch-based change detection is a not pixel-based but patch-based. The results need post-processing before accurate change boundaries can be derived. In addition, the change boundary from patch-based change detection is 2D instead of 3D. Although patch-based change detection might be efficient in localizing changes in large urban area, we still pursue point-based change labels. As PointNet and PointNet++ were proposed (Qi et al, 2017a; Qi et al., 2017b), they prove effective in learning multiscale deep features from point clouds. Therefore, we propose a Siamese PointNet++ architecture to extract point features from two point clouds. The network performs change detection implicitly and outputs a change label for each point in the ALS data.

In our project, we assume that the point clouds from ALS and dense image matching are already registered. First, the ALS point clouds are collected and provided in national mapping coordinate system. The photogrammetric products are also created in the same coordinate system through several or dozens of ground control points (GCP) (Haala and Rothermel, 2012). Accurate registration between the point clouds is the prerequisite for change detection.

### *1.4.2 Research questions*

Research questions are raised to each sub-objective accordingly:

**(1) Evaluation of the quality of dense matching point clouds and DSMs**

Q1: What entities and quality measures can be used for evaluation?

Q2: How does the GCP number affect the quality of dense matching points?

Q3: Does additional use of oblique images influence the dense matching quality?

Q4: How are the dense matching errors distributed in the study area?

Q5: Is the accuracy of DSMs and point clouds the same?

Q6: Is the dense matching error on different objects the same?

**(2) Evaluation of filtering algorithms and DTMs derived from DIM points.**

Q1: Can standard LiDAR filter be transferred to filter dense matching point clouds?

Q2: How are the errors and artefacts distributed in the DIM filtering results?

Q3: Does the filtering effect make a difference if point clouds are derived from different dense image matching algorithms?

Q4: What quality measures can be used to evaluate the DTM quality derived from DIM points?

Q5: How does the accuracy of DTMs distributed in the block?

Q6: Is DTM accuracy on the terrain the same with the that on the grassland?

Q7: Does the DTM accuracy improve if a ranking filter is applied to reduce the point cloud noise in advance?

**(3) Change detection and delineation between multimodal point clouds**

Q1: How to feed multimodal point clouds, DSMs and orthoimages into Convolutional Neural Networks (CNN)?

Q2: How to design CNN architectures for multimodal data from two epochs?

Q3: Which loss function can be used for model optimization?

Q4: Which features can be used to identify and delineate building change boundaries?

Q5: What machine learning classifiers can be used to identify the building boundaries?

Q6: How to evaluate change detection results at pixel-level and object-level, respectively?

Q7: How to visualize the feature maps in the CNN model?

Q8: How do the hyper-parameters affect the change detection results?

**(4) Synthetization of semantic segmentation and change detection for multimodal point clouds**

Q1: How to define joint categories for combined semantic segmentation and change detection and avoid information redundancy?

Q2: How to design CNN architectures to fulfil the tasks of semantic segmentation and change detection?

Q3: How to pre-process the dense matching point clouds so that the density and noise level are similar to the level of ALS data?

Q4: How to input the point cloud data from two epochs into a CNN model?

Q5: Which loss function can be used for joint model optimization?

Q6: How to handle the problem of imbalanced categories in the joint task?

## *1.5 Thesis outline*

The main motivation of the thesis is to study whether photogrammetric point clouds can be used to update ALS data and how to perform multimodal change detection. Our research did not start from developing change detection methods. Instead, we first investigate the quality of photogrammetric point clouds, DSMs and DTMs in order to answer whether the quality of dense image matching fulfills the requirement of data updating. In other words, we should answer first whether the quality of dense image matching products is equivalent to that of ALS data. After evaluating the quality of photogrammetric products, we turn to develop robust change detection methods specifically for multimodal point clouds.

The framework of this thesis is composed of two parts: data quality evaluation and change detection (Figure 1.6). We first analyze the difference between ALS data and DIM data qualitatively and quantitatively, and study whether the dense matching point clouds, DTMs and DSMs meet the requirements for ALS data updating. Then we put forward two methods for change detection between ALS data and DIM data: One change detection method is a patch-based change detection and delineation of the change boundaries. The other change detection method is combining the tasks of semantic segmentation and change detection.

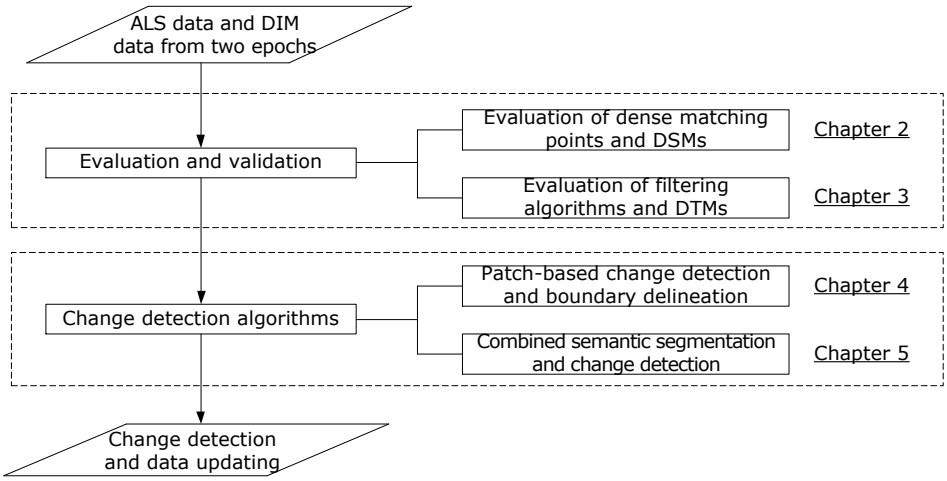



***Figure 1.6:*** *Organization of research topics in the dissertation.*

More specifically, the organization of the chapters is as follows:

**Chapter 1 –** Background, motivation and research objectives. It starts from the background and motivation of our research. Then a brief overview of the basics of airborne laser scanning and airborne photogrammetry is presented. Research problems with multimodal data, change definition and scene complexity are analyzed. Finally, research objectives and research questions are proposed.

**Chapter 2 -** Quality assessment of dense matching points and DSMs. A quantitative method for point cloud quality assessment is proposed. The proposed measures not only indicate the point cloud accuracy, but also indicate the noise level. In the experiment, the proposed method is used to evaluate the quality of Enschede data.

**Chapter 3 –** Evaluation of standard LiDAR filters and the derived DTMs. It evaluates the effect of standard LiDAR filters on photogrammetric point clouds. This chapter proposes a ranking filter to reduce the noise in DIM points, which first filters the dense matching point cloud, and then evaluates the quality of the derived DTMs.

**Chapter 4 -** Multimodal point cloud change detection based on a Convolutional Neural Network (CNN). Firstly, the method for normalizing multimodal input data is presented. Secondly, a Pseudo-Siamese Neural Network architecture is proposed. Then the boundary delineation method for changed buildings is introduced. Experiments are implemented on the Rotterdam data set. Results are evaluated at the patch-level, pixel-level, and object level, respectively.

**Chapter 5 -** A method for combined semantic segmentation and change detection. Firstly, a deep learning architecture for combined semantic segmentation and change detection is proposed. Then the workflow for DIM data denoising and training data preparation is presented. Experiments are implemented on the Rotterdam data set. The effect of the proposed method is validated by comparing its performance with three other change detection methods.

**Chapter 6 –** Conclusions and recommendations. It draws the conclusion of the whole thesis and provides the future recommendations.

It should be noted that chapters through 2 to 5 are based on published scientific articles. Although there will be some overlap in their introduction and motivation, this design ensures that each chapter can be read or referred to as an independent section, allowing a reader to focus on the areas which are of particular interest to him or her.

In addition, since the research focus from chapter 2 to chapter 5 are different, their related work also differs. Therefore, literature review will be embedded in each separate chapter. Chapter 2 reviews research on the evaluation of point clouds and DSMs generated from aerial photogrammetry. Chapter 3 reviews point cloud filtering algorithms. Chapter 4 reviews 3D change detection, multimodal change detection, and deep learning for multimodal data processing. Chapter 5 reviews semantic segmentation methods for point clouds.

# Chapter 2 – Patch-Based Evaluation of Dense Image Matching Quality[1]

[1] This chapter is based on:

Zhang, Z., Gerke, M., Vosselman, G. and Yang, M. Y., 2018. A patch-based method for the evaluation of dense image matching quality. International Journal of Applied Earth Observation and Geoinformation, 70, 25-34.

## 2.1 Introduction

Airborne laser scanning (ALS) and photogrammetry are the two main techniques to obtain 3D data representing the earth surface (Höhle and Höhle, 2009). The properties of laser scanning and photogrammetry have been widely compared before (Baltsavias, 1999; Leberl et al., 2010; Haala et al., 2010; Remondino et al., 2014; Cavegn et al., 2014; Yang and Chen, 2015; Tian et al., 2017). Compared to airborne laser scanning, image acquisition in photogrammetry is mostly cheaper and more efficient in data acquisition flights (Hobi and Ginzler, 2012; Nurminen et al., 2013; Maltezos et al. 2016). In many countries photogrammetric image blocks are captured anyway for administrative and planning purposes with decreasing time intervals, so the question is to what extent these data can be used to replace ALS data in various application domains such as Digital Elevation Model (DEM) acquisition (Ressl et al., 2016), forestry mapping (Mura et al., 2015), classification and object extraction (Tomljenovic et al., 2016; Dong et al., 2017), and 3D modeling (Xiong et al., 2015).

We want to explore the potential of using photogrammetric products as effective alternatives to laser scanning data. In order to judge this potential, it is necessary to evaluate the data quality of 3D products from dense image matching (DIM). Assessing the absolute accuracy of 3D data can be time-consuming and labor-intensive for two reasons. Firstly, the reference data must be verified as being more accurate than the compared data. Secondly, the sample size should be sufficiently large to draw sound conclusions. The contributions are as follows:

- The dense matching quality is evaluated robustly based on a large number of planar patches of the same size extracted from planar ground surfaces in both the DIM point cloud and the ALS point cloud. Quantitative quality measures are proposed to represent the accuracy and precision at both the local patch level and the whole block level. After considering possible breaklines in natural scene and excluding patches with possible changes between the DIM data and reference data, the evaluation based on these planar patches reveals the distribution of DIM errors in the whole photogrammetric block for the first time. Compared to the previous point-to-point and point-to-plane comparisons, this framework computing the plane-to-plane distance is more robust to local blunders and artefacts.

- In order to test the usability of the proposed framework, several influence factors related to the DIM quality are studied. To capture oblique airborne imagery is not yet standard, but especially in urban applications it becomes more important (Toschi et al., 2017). Hence, we also evaluate how the additional use of oblique images influences the dense matching quality.

Meanwhile, suggestions are given on the photogrammetric quality control and dense matching parameter settings.

The chapter is organized as follows: Section 2.2 presents the patch-based DIM evaluation framework. Section 2.3 gives details on the study area and experimental settings, while section 2.4 focuses on experimental results. Section 2.5 discusses those results and section 2.6 finally concludes the chapter.

## 2.2 Related work

Evaluating the quality of 3D data often comes as the preceding step before data application. As fundamental 3D products representing the object space, point clouds and regular DSMs can be derived from a standard photogrammetric workflow. A point cloud has single points which carry the full geometric information, including possible individual errors. The advantage of DSMs is that the random noise might be averaged out. Users often neglect the raw point cloud just because the regular DSM data are easier to handle (e.g. Rottensteiner et al., 2014; Qin, 2014; Gevaert et al., 2017). Meanwhile, they assume that the accuracy of DSMs is equal or at least close to the accuracy of point clouds. However, whether the object details are retained in the DSM data largely depends on the scene, raw point cloud quality and interpolation method. Therefore, both point clouds and DSMs have their pros and cons. Depending on the application, both are useful, and hence should be analyzed separately. Previous work of evaluating the absolute accuracy of 3D data can be divided into two categories based on the reference data.

In some previous evaluation studies, the reference data was collected by Real Time Kinematic (RTK) GPS. However, the sample size was relatively small in this case. Jaud et al. (2016) evaluated point clouds generated from images obtained by Unmanned Aerial Vehicles (UAVs). Twenty-four ground targets were set in the study area which served as GCPs in the triangulation and as check points in the DIM evaluation. The coordinates of these targets were obtained by post-processed differential GPS. Hobi and Ginzler (2012) evaluated the quality of Digital Surface Models (DSMs) from stereo matching of WorldView-2 satellite images and ADS80 aerial images using 36 reference points obtained by sub-decimeter differential GPS. Nurminen et al. (2013) studied the accuracy of DSMs derived from ALS and DIM in the estimation of plot-level variables. The reference variables of the forest plots were obtained by field surveys.

In addition, the reference data may be obtained by laser scanning. The basic assumption is that the point clouds obtained by laser scanning are more

accurate than point clouds from photogrammetry, at least concerning the height component. Mandlburger et al. (2017) calculated the deviation between DIM-DSM and Lidar-DSM at impervious surfaces and found a systematic deviation of 0.043 m and a dispersion of 0.041 m. Tian et al. (2017) selected 184 inventory plots as the samples for DSM evaluation in a forest area. Two datasets from ALS were taken as reference data. Similar work taking laser scanning data as reference can also be found in (Poon et al. 2005; Gehrke et al., 2010; Moussa et al., 2013; Remondino et al. 2014; Nex et al., 2015; Jaud et al., 2016; Maltezos et al., 2016; Sofia et al., 2016; Ressl et al 2016).

Some deficiencies of previous DIM evaluation work are summarized as follows: Firstly, some studies evaluated the point cloud derived from Semi-Global Matching (SGM) by making comparisons with ALS data or terrestrial laser scanning data on a planar sports field, complex castle or building façade (e.g. in Rothermel et al, 2012; Haala and Rothermel, 2012; Cavegn et al., 2014; Remondino et al., 2017). However, the small sample size or local area cannot properly represent the error distribution in the whole block. Secondly, when calculating quality measures, point-to-point distance (Kraus et al., 2006) and point-to-plane distance (Rothermel and Haala, 2011; Nex et al., 2015) were widely used as the measures to represent the accuracy. However, these measures are sensitive to blunders and random noise within the dense matching point clouds. Thirdly, the quality measures were less reliable or persuasive if calculated without consideration of the breaklines in natural scenes, such as bumpy terrain, edges of traffic islands or curbstones, and edges and ridges of roofs (e.g. in Ressl et al., 2016; Jaud et al., 2016).

In our previous work of evaluating point cloud from multi-view photogrammetry (Zhang et al., 2017), robust quality measures were calculated on each roof segment. The problem was that the roof sizes and inclination angles varied from roof to roof. In this paper, a framework for evaluating point clouds and DSMs generated from a state-of-the-art dense matching algorithm is proposed.

## *2.3 Methodology*

In our evaluation framework, an ALS point cloud is taken as the reference data. The ALS data are assumed to be accurate with regards to the external reference and precise in consideration of random noise. The "patches" used as evaluation units are regular squares selected on the planar ground from the ALS data. Every patch is a sample for quality evaluation. Therefore, the densely selected patches on the ground can indicate the error distribution in the whole photogrammetric block. The proposed framework for DIM evaluation includes four steps: Firstly, square patches are detected from the ALS data and

validated (Section 2.3.1); Secondly, corresponding DIM points are searched for each patch and the patches are further screened based on patch-based attributes (Section 2.3.2); Thirdly, quality measures are computed (Section 2.3.3); Finally, statistical analyses are performed on the valid patches.

### 2.3.1 Patch detection

The goal of patch detection is to localize candidate planar patches on the ALS point cloud. The patches taken as samples should be selected on the planar ground area from the ALS data. The selection of patches should further avoid data gaps and breaklines. Planar patches of uniform size with acceptable noise level are considered valid and thus used for evaluation purpose.

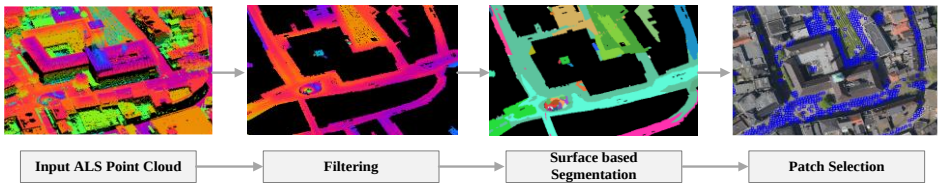



***Figure 2.1:*** *Workflow for detecting candidate patches from ALS point cloud.*

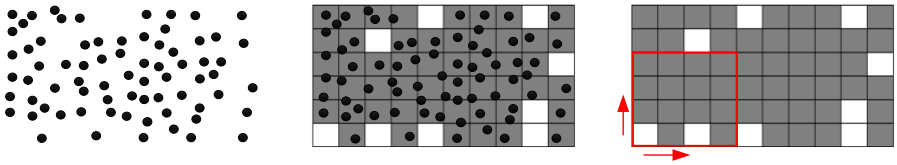
In Figure 2.1, the workflow is depicted. Firstly, ground points are identified from the ALS data using the method of (Axelsson, 2000). Then planar segments are extracted from ground points using a surface-based growing method (Vosselman, 2013). This approach employs the 3D Hough transform to detect seed surfaces. Then the nearby points are added to the surface if the distance from a certain point to the fitted plane is below a certain threshold. After new points are added to the segment, the plane parameters are recalculated before testing the next point. Slight over-segmentation is preferred over under-segmentation: over-segmentation can ensure better planarity and help avoiding breaklines in the segments.

After segmentation, the laser points with segment labels should be screened to discard small clusters or noisy segments. Features listed in Table 2.1 are used to remove these small or noisy segments. Segment size is used to eliminate small segments; linearity of segment is used to eliminate narrow segments; Plane slope is used to exclude segments on steep slopes; average angle and residual of plane fitting (RPF) are used to eliminate noisy clusters. A segment is kept only if it passes the check based on the five features.

***Table 2.1:*** *Segment-based features for extracting smooth segments.*

| Feature | Description |
|---|---|
| *Segment size* | Number of points in the segment |
| *Linearity of segment* | $(\lambda_1 - \lambda_2)/\lambda_1$ , $\lambda_1$ is the maximum eigenvalue of the covariance matrix (Weinmann et al., 2015) |
| *Plane slope* | Normal direction of the fitted plane |
| *Average angle* | Mean of the angles between local point normals and the fitted plane normal |
| *Residual of plane fitting* (*RPF*) | Standard deviation of the distances between points and the plane fitted to the segment |

After smooth segments are obtained, patch selection is implemented in the bounding box of the segments. Figure 2.2 shows that the bounding box is calculated around all the points in the segment. In Figure 2.2(b) and (c), the white cells indicate data gaps or empty cells, the grey cells indicate cells with points. (c) shows that the patch size is 4 × 4 cells represented by the red frame: the patch is valid only if there is no data gap in the 16 cells. A raster grid is built within the bounding box in the horizontal space. If there is no point within a certain grid cell, the grid cell is set to empty, i.e. white cells in Figure 2.2(b) and (c).



(a) (b) (c)

***Figure 2.2:*** *Patch selection from the data gap grid in the horizontal space. (a) points in a planar segment; (b) grid of data gaps; (c) patch selection.*

A patch is compiled out of several initial grid cells and is within the bounding box of the segment. The patch size should be scaled with the point density. It should be large enough to contain sufficient points but small enough to guarantee a large number of samples. In this paper, the cell size is set to 0.5 × 0.5 m and one patch contains 4 × 4 cells (see Figure 2.2(c)). Hence, the patch size is 2 m × 2 m. If no data gap is detected in any cell within this patch, this patch is valid. In this way, the patch selection method can automatically avoid the locations of data gaps in the segments. The cell size is determined according to the laser point density. It should be large enough to guarantee at least one point in each cell in areas without data gaps. Additionally, in order to speed up the iteration, the stride can be two or more grid cells each time. Due

to the "brute-force" search over dense grid cells, many selected patched are overlapping. The densely overlapping patches are screened automatically based on the spatial relationship to make sure that a certain location in the study area is used only once.

### 2.3.2 Patch screening

After patch detection, the candidate patch locations were obtained. The DIM points of a certain patch are selected according to the overlapping ALS patch. That is, the selection of DIM points adopts the same bounds as the ALS patch. Rule-based screening is implemented again at the patch level as previously implemented on the ALS segments in Section 2.3.1. Four rules are employed to screen the patches for different purposes:

(1) *Number of points* in the DIM patch: The DIM patches with data gaps are eliminated.

(2) *Mean deviation* between ALS patch and DIM patch: ALS data and aerial imagery could be captured at different times. This rule is to ensure that the mean deviation is caused by dense matching error but not by natural or man-made changes in between the ALS data and DIM data. The threshold for mean deviations should be for example at 0.99 quantile of the mean deviations.

(3) *Shading attribute*: The dense matching points in shadow often contain blunders and artefacts. For example, the dense matching errors along narrow alleys (often in shadow) are supposed to be much larger than the errors in the open area. Hence, patches in shadow should not be used for evaluation. The shadow mask is calculated from an orthoimage based on a grayscale histogram (Sirmacek and Unsalan, 2009). Only if all the four corners and the center location of a certain patch lie in the non-shaded area, the patch is accepted as non-shaded patch.

(4) *Green index*: After the above screening, some patches on the grassland can still be left. They should not be used for evaluation. The reason is that dense matching usually delivers points on top surface of grass, while laser scanning can penetrate the grass and represent the soil surface. In this case, the computed mean deviations will contain not only dense matching errors, but also the grass height. The Normalized Excessive Green Index (nEGI) in Eq. (2-1) is used to filter out vegetation patches on the orthoimages (Qin, 2014).

$$nEGI = (2G - R - B)/(2G + R + B) \tag{2-1}$$

Similar to the shading attribute, only if all the four corners and the center of a patch are labelled as non-vegetation, this patch will be used for evaluation. After patch screening, DIM evaluation and statistical analysis are performed based on these valid patches.

### 2.3.3 Patch-based quality measures

The quality measures are calculated at each patch. This paper evaluates two factors related to the data quality:

- Accuracy: the deviation between the compared data and the ground truth (or reference data).

- Precision: the relative closeness of many measurements (in our case, dense matching points) to each other, i.e. the level of random noise.

Accuracy and precision are independent of each other. This paper only focuses on the vertical component of the point clouds and DSMs. Assuming that the 3D data show a normal distribution and contain no blunders, Table 2.2 shows the quality measures calculated at the patch level and the photogrammetric block level to represent the data accuracy and precision. The patch-based measures are aggregated into the block-level quality measures. In Table 2.2, $i$ denotes the index of a patch in the whole block; $j$ denotes the index of a specific DIM point in a certain patch. $\Delta h_{ij}$ denotes the deviation from the $j$th DIM point to the plane which is fitted to all the ALS points within the $i$th patch. $n_i$ denotes the number of DIM points in the $i$th patch. $m$ denotes the number of patches in the whole block which is also the sample size for statistical analysis.

***Table 2.2:*** *Quantitative quality measures for dense matching evaluation*

| Level | Quality measure | Definition | Meaning |
|---|---|---|---|
| Patch-level | Mean deviation | $\mu_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \Delta h_{ij}$ | Accuracy of DIM points in a certain patch w.r.t. the reference ALS plane |
| | Standard deviation | $\sigma_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n} \left(\Delta h_{ij} - \mu_i\right)^2}$ | Precision of DIM data in a certain patch |
| Block-level | Mean of mean deviations | $\bar{\mu} = \frac{1}{m}\sum_{i=1}^{m} \mu_i$ | Overall accuracy of DIM data in the whole block |

| | | | |
|---|---|---|---|
| | Standard deviation of mean deviations | $\sigma_\mu = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(\mu_i - \bar{\mu})^2}$ | Variation of accuracy measures in the block |
| | Average standard deviation | $\mu_\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}{\sigma_i}^2}$ | Overall precision of DIM data in the block |

In Table 2.2, *Mean deviation* and *Standard deviation* are to indicate accuracy and precision at single patch level (Index $i$ indicates a patch). Both $\bar{\mu}$ and $\sigma_\mu$ are measures indicating the accuracy at the block level. Specifically, a larger $\sigma_\mu$ indicates more dispersed patch-based errors in the block. In addition, $\mu_\sigma$ is to indicate the level of precision in the whole block.

In addition to the point cloud from dense matching, a DSM is also obtained from a standard photogrammetric workflow. The DSMs from a photogrammetric workflow can be in grid data structure but saved in point cloud format. Same with cropping point cloud patches, the DSM patches are generated by cutting out the corresponding patch area from the raster DSM.

## 2.4 Study area and experimental setup

### 2.4.1 Study area

The study area is located in Enschede, The Netherlands. Figure 2.3 shows the dense matching block and the area for quality evaluation (1.6 km$^2$). This area is a densely-built urban area mainly covered by buildings, roads, squares, railways and vegetation. 510 aerial images including 102 nadir images and 408 oblique images were obtained by *Slagboom en Peeters* in 2011 together with exterior orientations. The tilt angle of oblique view is approximately 45°. The image size is 5616 × 3744 pixels. The GSD of nadir images equals 0.1 m. The overlap of nadir images is approximately 75% both along track and across track. The ALS data were acquired in 2007. The standard deviation of height differences between overlapping strips was around 2 cm (Vosselman, 2008). The absolute height accuracy has not been analysed before. In the block, 105 ground reference targets (RTs) were measured with a Leica CS15 receiver using real time kinematic GPS. When collecting RTs, the accuracy of almost all the 105 RTs was better than 0.02 m in X, Y and Z directions, respectively; For several RTs, however, the accuracy in one or two directions were in between 0.02 m and 0.03 m. All of the RTs were the corners of zebra crossings, centers of manholes or other distinctive corners in the urban scene.

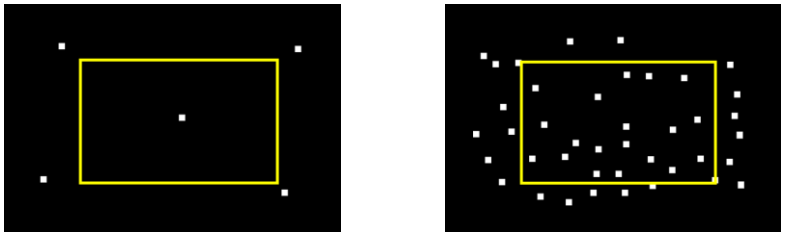
(a) (b)

***Figure 2.3:*** *Dense matching block and orthoimage for the evaluation area. (a) Dense matching points of the whole block; (b) Orthoimage of the area for quality evaluation. The area in the yellow frame in (a) is exactly the same area as shown in (b).*

The RTs are used to evaluate the ALS quality. Since all the RTs are located in the open area, planes are fitted to the neighboring ALS points. The vertical residual from a RT to the fitted ALS plane is calculated as the indicator for the ALS accuracy. Results show that the mean deviation ($\mu$) and standard deviation ($\sigma$) between the RTs and the fitted ALS plane are 0.013 m and 0.031 m. Furthermore, if the residual from RT to the ALS fitted plane is larger than three times of the standard deviations ($\sigma$), this RT will be discarded. This cross-verification ensures that both the ALS data and RTs used in the BBA, dense matching and DIM evaluation are reliable. Finally, 99 RTs passing this cross-verification are used as GCPs or check points in the BBA.

### *2.4.2 Bundle adjustment*

In the step of BBA, two configurations with 5 and 44 GCPs are set up for comparative study. The motivation to use and evaluate 2 different GCP-scenarios is to check whether block deformation, possibly caused by an insufficient GCP distribution, or by overfitting effects, will be observed by our evaluation method. The GCPs are evenly distributed in the block in both scenarios (Figure 2.4). When 5 or 44 RTs are used as GCPs, the remaining 94 and 55 RTs are taken as check points, respectively. Note that direct sensor orientation elements available in this dataset are considered unreliable, therefore an indirect senor orientation approach is implemented. The results of the two configurations with 5 GCPs and 44 GCPs are presented in Section 2.5.1 and Section 2.5.2, respectively.



(a) (b)

***Figure 2.4:*** *Two configurations with different GCP amounts used in bundle adjustment: (a) Configuration with 5 GCPs; (b) Configuration with 44 GCPs. The white dots show the GCP distributions in the block. The yellow rectangle indicates the area for DIM evaluation (1.6 km$^2$).*

The BBA was run in Pix4Dmapper Pro (version 3.2) on the original full resolution images. The standard deviation of the 3D GCPs was set to 0.02 m (default value in Pix4D) which controlled the GCPs weights in BBA. Table 2.3 shows the vertical RMSEs at GCPs and check points (CPs) when the horizontal accuracy of GCPs is 0.02 m. When the GCP amount increases from 5 to 44, the BBA network becomes more difficult to fit. Hence, the RMSE at GCPs increases. Meanwhile, the overall BBA accuracy is improved which is supported by the improved RMSE at check points.

***Table 2.3:*** *Vertical RMSEs at GCPs and CPs*

| Number of GCPs | Number of CPs | RMSE at GCPs (m) | RMSE at CPs (m) |
|---|---|---|---|
| 5 | 94 | 0.007 | 0.060 |
| 44 | 55 | 0.018 | 0.031 |

### 2.4.3 Dense image matching

For the execution of dense image matching, we select the state-of-the-art software SURE (Surface Reconstruction, version 2.1.0.33) from nFrames. A few work has reported its performance in data accuracy (Haala and Rothermel, 2012; Rothermel et al., 2012; Ressl et al., 2016). The dense matching algorithm in SURE is a tube-shaped SGM (t-SGM). The SGM method in (Hirschmüller, 2008) is improved by restricting the disparity searching space which leads to a higher efficiency. Furthermore, the redundant disparity information is exploited to eliminate blunders and increase the accuracy of depth.

The interior orientation (IOs) and exterior orientation (EOs) elements are imported from Pix4D. Several parameters are supposed to control the dense matching quality. *Minimum Model Count* (*MMC*) represents the minimum

number of models for a 3D point to be considered valid during triangulation. A larger *MMC* increases the reliability of generated points but also leads to a lower number of accepted matched points. When *MMC* is set large (e.g. ≥ 3), we find that many data gaps appear in narrow alleys. Hence, *MMC* is fixed to 2 in all our experiments. The image scale for dense matching is fixed to 1/2 so that the dense matching pipeline can be much faster compared to running at full scale. Note that different image scales used in dense matching will also affect the dense matching quality, but the impact of image scale is not the focus of our paper. The interpolation method for DSM generation is set to *Inverse Distance Weighting* (IDW). The resolution of the DSM grid is 0.1 m, i.e. equal to the size of nadir GSD.

The DIM data quality based on the configuration with 5 GCPs is evaluated to study the two issues: (1) The impact of the additional use of oblique images on the dense matching accuracy and precision; (2) Whether the accuracy of point cloud and DSM from a photogrammetric pipeline are the same. In summary, four data sets are obtained:

(1) GCP05_N+O_PC;
(2) GCP05_N+O_DSM;
(3) GCP05_N_PC;
(4) GCP05_N_DSM.

The naming scheme of the four data sets above shows different parameters of the data. GCP05 means 5 GCPs are used in BBA; N+O indicates that both nadir (N) and oblique (O) images are used in dense matching; "N" indicates that only nadir images are used in dense matching; PC or DSM refers to point cloud or DSM, respectively.

### 2.4.4 Parameter settings for DIM evaluation

In the segmentation step during patch detection, a surface growing radius of 1.0 m and maximum distance between point and fitted plane of 0.2 m are employed according to the point cloud density and noise level (Vosselman, 2013). The thresholds for the rules in Table 2.1 are set based on 200 valid segments and 200 invalid segments: *segment size* is 100, *linearity of segment* is 0.99, *plane slope* is 45°, *RPF* is 0.1 m, and *average angle* is 5°. The histograms of each feature for valid and invalid segments are depicted, respectively. Then the value that can best separate the two groups of segments is manually taken as the threshold. For example, the *segment size* is set according to the histogram of point amounts in these 200 valid segments. The smallest segment size is 100. Also segments with less than 100 points are very likely to be small noisy clusters. The segments with *linearity of segment* value

larger than 0.99 are likely to be poles or other linear structures according to the histograms of invalid segments.

In patch screening, the threshold for *number of points* is determined from the histograms of the point amounts in the valid and invalid DIM patches. The *mean deviation* threshold is set by adding the 0.99 quantile of the mean deviations with some small tolerance value (e.g. 0.02 m). The threshold for nEGI in Eq. (2-1) is set to 0.1 to recognize vegetation (Qin, 2014).

## 2.5 Experimental results

### 2.5.1 Results of the configuration with 5 GCPs

After patch screening, 7391 patches on the grassland and 2111 patches in the shadow are filtered out and then only patch samples in the bare ground areas are evaluated in this paper. Figure 2.5 shows some examples of valid patches marked in blue, which are further used for DIM evaluation. Generally, almost all the selected patches are away from shadow, grassland and breaklines. Specifically, the left figure shows the selected patches on the central bus station of Enschede. The white stripes are actually platforms higher than the grey ground by around 0.2 m. The proposed algorithm performs well in extracting planar patches away from breaklines.

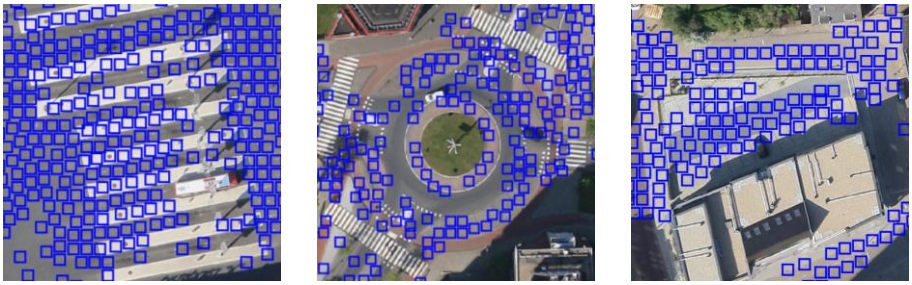


***Figure 2.5:*** *Examples for extracted patches marked by blue squares. Patch size is 2 m × 2 m.*

Finally, 24,634 non-shaded patches of 2 m × 2 m are selected in the whole block, i.e. 0.1 $km^2$ totally. In order to make the block-level quality measures comparable, the same patch samples are used to evaluate the four data sets. Table 2.4 shows the quality measures at the block level calculated for the four data sets.

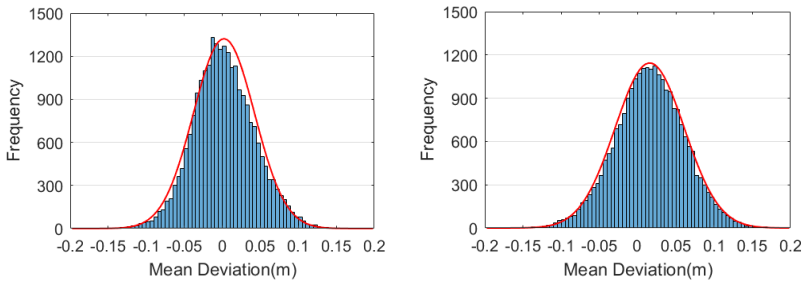
***Table 2.4:*** *Quality measures for point clouds and DSMs in configurations with 5 GCPs (m).*

| Data sets | N+O | | | N | | |
|---|---|---|---|---|---|---|
| | $\bar{\mu}$ | $\sigma_\mu$ | $\mu_\sigma$ | $\bar{\mu}$ | $\sigma_\mu$ | $\mu_\sigma$ |
| Point cloud (PC) | 0.002 | 0.040 | 0.094 | 0.016 | 0.045 | 0.106 |
| DSM | 0.034 | 0.060 | 0.048 | 0.024 | 0.066 | 0.083 |

#### *2.5.1.1 Evaluation of the impact of oblique images*

The first row of Table 2.4 shows the comparison between GCP05_N+O_PC and GCP05_N_PC. A general finding is that when both nadir and oblique images are used in dense matching, all the three quality measures are better than the measures of configuration with only nadir images. The $\bar{\mu}$ improves remarkably by 0.014 m from 0.016 m to 0.002 m when oblique images are used; The $\sigma_\mu$ improves very slightly by 0.005 m; The $\mu_\sigma$ also improves by 0.012 m.

The distribution of mean deviations in the whole block for the two configurations are shown in Figure 2.6. A normal distribution is estimated using the mean and standard deviation calculated from the same data set. The normal distribution is scaled and then superimposed on the histogram to visualize the deviation between the real measurements and a normal distribution (Höhle and Höhle, 2009). In each histogram, the horizontal axis indicates the patch-based mean deviation, the vertical axis indicates the frequency of patches in the whole block. The scale and interval of the axes for the two histograms are all the same.




[-0.06, 0.07] [-0.06, 0.09]

(a) GCP05_N+O_PC (b) GCP05_N_PC

***Figure 2.6:*** *Distribution of mean deviations for 24,634 non-shaded ground patches (Unit: m). All the histograms are overlaid with an estimated normal distribution. The interval below each histogram refers to the 0.05 and 0.95 quantile, respectively.*

The peak of Figure 2.6(a) is located at approximately 0 which corresponds with $\bar{\mu}$ = 0.002 m in Table 4. The histogram is centralized and "thin" in shape which

corresponds with $\sigma_\mu$ = 0.040 m. The mean deviations range from -0.060 m to 0.070 m which means that in most patches, the vertical error of dense matching is better than 1 GSD. Figure 2.6(b) shows a relatively dispersed histogram compared to Figure 2.6(a). In Figure 2.6(b), the peak is located at 0.016 m at the horizontal axis. The range of mean deviations from -0.060 m to 0.090 m is slightly wider than in Figure 2.6(a).

Figure 2.7 shows the patch-based mean deviations in the block for the data GCP05_N+O_PC colored according to the absolute mean deviation values. That is, each pixel indicates a patch location. The patch samples are densely distributed in the whole block, mainly on roads, squares and parking lots. According to the color bar, the absolute mean deviations range from 0 to 0.12 m. Generally, the dense matching errors are homogenous in the whole block. However, in some locations, especially along narrow alleys, the mean deviations may get worse. The point clouds in those regions get less accurate for two reasons: First, there are usually less visible image rays on the ground; Second, the image contrast is poor so dense matching will be problematic when finding correspondences among images.

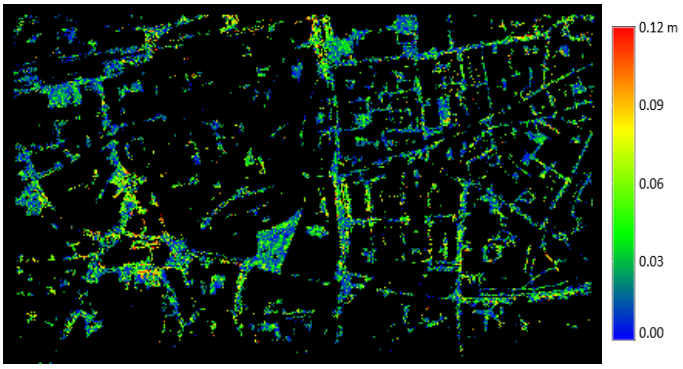



***Figure 2.7:*** *Patch-based mean deviations in the whole block colored by the absolute values for the data GCP05_N+O_PC. Color coding from blue to red indicates that the mean deviation increases. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)*

#### *2.5.1.2 Visualization of patch-based mean deviations*

The patch-based mean deviations are visualized in Figure 2.8. This square paved by concrete in our study area is relatively smooth. The square patches are colored based on positive or negative values. In Figure 2.8(b) and (d), the mean deviation values are filled in the squares to visualize the dense matching errors in each patch.

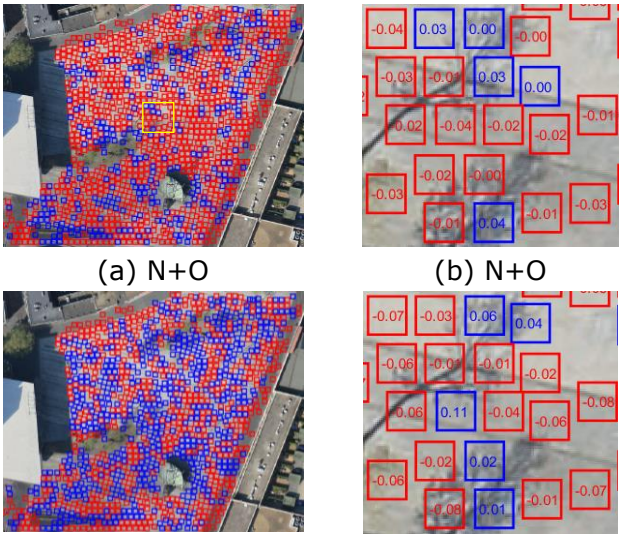
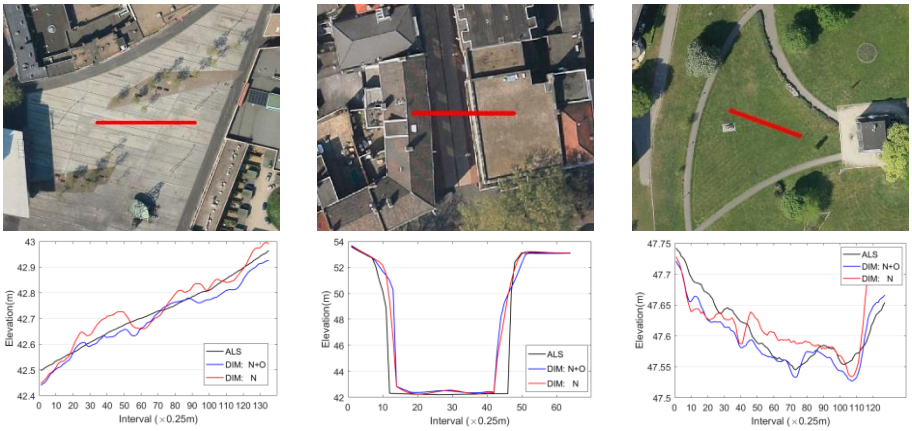



***Figure 2.8:*** *Visualization of patch-based mean deviations for the data sets GCP05_PC. The blue patches indicate positive values while the red indicates negative values, (a) and (b) show the mean deviations for the data GCP05_N+O_PC, (c) and (d) show the quality measures for GCP05_N_PC. The yellow rectangle on (a) is zoomed in and filled in the mean deviation values as shown in (b) and (d).*

Figure 2.8(a) and (c) show that the patch-based mean deviations vary between positive and negative on the square. The positive value indicates that the point cloud surface from dense matching is higher than the point cloud surface from laser scanning, and vice versa. We can infer that the point cloud surface from dense matching is fluctuating around the referred ALS surface. In addition, comparing Figure 2.8(a) and (c) in the red and blue patterns, or (b) and (d) in the values shows that whether or not oblique images are used in dense matching makes a large difference on the local accuracy.

In addition, the profiles of ALS points and DIM points are shown in Figure 2.9. All the DIM data are generated from the configurations of 5 GCPs. The profile interval in the horizontal space is 0.25 m. Figure 2.9(a) shows the profiles along a smooth downtown square. Both profile N+O and profile N are fluctuating around the ALS profile. As a comparison, Figure 2.9(b) and (c) show the profiles across a narrow shaded alley and short grass. The deviation between ALS profile and DIM profile in Figure 2.9(a) and (b) is caused by the dense matching errors on the smooth surface with poor texture while the deviation in Figure 2.9(c) is mainly caused by the rugged grassland surface itself.

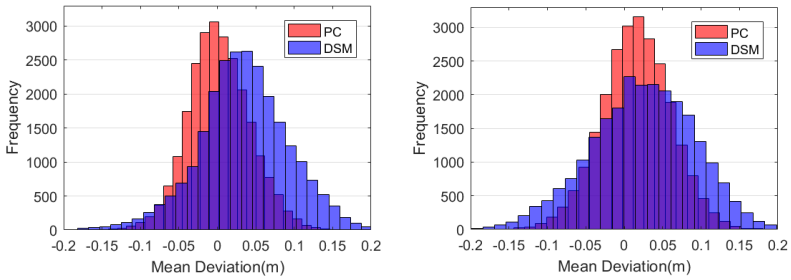



(a) Smooth square (b) Narrow alley (c) Grassland

***Figure 2.9:*** *3D data profiles for three area: (a) smooth concrete square, (b) narrow alley and (c) grassland. The top row shows the orthoimages with profiles marked in red. The bottom row depicts the relevant profiles for ALS point cloud (black), DIM_N+O (blue) and DIM_N (red). The horizontal axis indicates the interval along the profile, the vertical axis indicates the elevation.*

### *2.5.1.3 Comparison between point cloud and DSM*

As expected, the $\mu_\sigma$ of DSMs in Table 2.4 indicates that the noise level is much lower than within the point clouds since interpolation is employed. Based on our evaluation framework, we observe a bias between point clouds and DSMs from the software pipeline. The first column in Table 2.4 shows that the difference of $\bar{\mu}$ between point cloud and DSM for N+O is 0.032 m. The second column in Table 2.4 shows that the DSM surface is higher than the point cloud surface by 0.008 m when only nadir images are used in dense matching. Therefore, we conclude that the interpolation process changes the data accuracy; and the magnitude seems to depend on the point cloud density. Mandlburger et al. (2017) also reported the same deviation between point cloud surface and DSM surface so this deviation might be caused by the interpolation process in the SURE software. In order to visually analyse the deviation between point clouds and DSMs, refer to Figure 2.10.



(a) N+O (b) N

***Figure 2.10:*** *Overlaid histograms of mean deviations for point cloud and DSM. Red histograms indicate the distributions of point clouds; blue histograms indicate the distributions of DSMs. "PC" in the legend indicates point cloud. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)*

Figure 2.10(a) shows a clear deviation between the peaks of the two histograms while in Figure 2.10(b) the deviation is relatively smaller (0.032 m vs. 0.008 m) when oblique images are not used. Another finding is that the distribution of mean deviations for DSMs is more dispersed than for the point cloud, which corresponds with Table 2.4 that the $\sigma_\mu$ of DSMs is larger than point clouds. In summary, although the noise level is reduced from point clouds to DSMs, the absolute accuracy is changed during interpolation and the error distribution in DSMs is more dispersed than in point clouds.

## 2.5.2 Impact of number of GCPs and weights

We evaluate the point cloud *GCP44_N+O_PC* using the same 24,634 patches as we did for the configuration with 5 GCPs. The calculated quality measures are listed in the last row of Table 2.5. It is obvious that the dense matching accuracy indicated by $\bar{\mu}$ and $\sigma_\mu$ gets worse when more GCPs are used.

***Table 2.5:*** *Quality measures for point clouds when the GCP weights in BBA is set to 0.02 m (m).*

| Configuration | $\bar{\mu}$ | $\sigma_\mu$ | $\mu_\sigma$ |
|---|---|---|---|
| GCP05_N+O_PC | 0.002 | 0.040 | 0.094 |
| GCP44_N+O_PC | -0.026 | 0.049 | 0.098 |

However, Table 2.3 in Section 2.4.2 shows that when the GCP amounts increases from 5 to 44, the RMSE at check points in BBA reported by Pix4D decreases from 0.060 m to 0.031 m. That is, the check points indicate that the accuracy of BBA is getting better when more GCPs are used. Empirical knowledge from previous studies (e.g. Gerke et al., 2016) also indicates that the more GCPs, the better the BBA accuracy will be.

Therefore, when the GCP amount increases from 5 to 44, the accuracy of BBA gets improved, but the accuracy of the DIM point cloud deteriorates. This contradictory finding is further evaluated by visualizing the patch-based mean deviations in Figure 2.11.

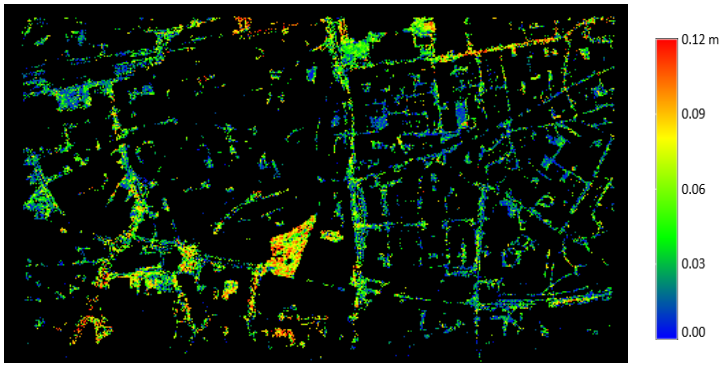



***Figure 2.11:*** *Distribution of mean deviations (absolute values) for GCP44_N+O_PC when GCP weight for BBA is set to 0.02 m.*

Figure 2.11 shows inhomogeneous error distribution in the block. Even though the absolute mean deviations still range from 0 to 0.12 m, the absolute mean deviations in the southwest of the block is generally larger than the other parts. The block is thus verified to be overfitted. The image block with large forward and sideward overlaps results in a very strong network of bundles. When many GCPs are used with relatively high weights compared to the tie points, the noise in the GCPs leads to a deformation of the network of bundles. The resulting errors in the exterior orientations propagate to locally systematic errors in the dense matching point cloud.

In order to check whether the BBA network is overfitted, the a priori standard deviation of the GCPs is set to 0.05 m in the BBA in Pix4D. In this case, the bundle adjustment network controlled by the GCPs gets "loose". The BBA result is that the RMSE at GCPs is 0.019 m and the RMSE at check points is 0.031 m. Compared with Table 2.3, the RMSEs at GCPs and check points change very slightly. Then we evaluate the new point cloud generated by SURE with new orientations. We observe a large improvement in the point cloud quality. The $\bar{\mu}$, $\sigma_\mu$ and $\mu_\sigma$ for the new point cloud (from GCP standard deviations of 0.05 m) is 0.011 m, 0.044 m and 0.094 m, respectively. Comparing these results to those obtained with the higher GCP weights (Table 2.5), in particular the $\bar{\mu}$ value of 0.011 m indicates that the systematic error is strongly reduced.

The overfitting effect is alleviated as shown in Figure 2.12. The homogeneity level of mean deviations in the block is much better than Figure 2.11 and no remarkable systematic deviations appear.

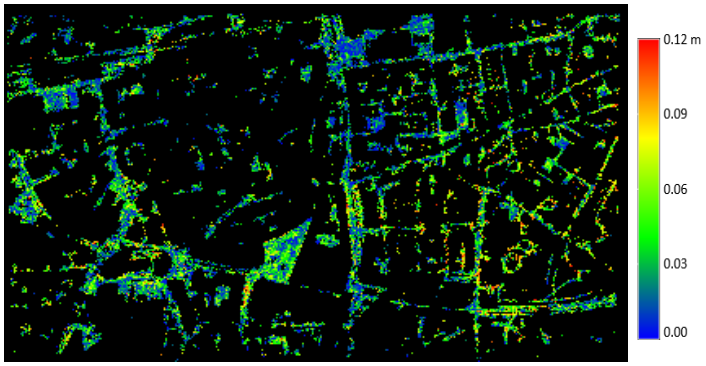



***Figure 2.12:*** *Distribution of mean deviations (absolute values) for GCP44_N+O_PC when GCP weight for BBA is set to 0.05 m.*

## 2.6 Discussion

In our evaluation framework, both the $\bar{\mu}$ and $\sigma_{\mu}$ are used to represent the dense matching accuracy in the block. The $\bar{\mu}$ indicates the general bias of the DIM points from the reference while the $\sigma_{\mu}$ indicates the dispersion level of the dense matching errors. In Table 2.4, when oblique images and nadir images are both used in dense matching, the $\bar{\mu}$ gets improved, but the $\sigma_{\mu}$ keeps relatively stable.

Even when the ALS accuracy is verified in Section 2.4.2, the noise in the ALS data still has a small impact on the computed quality measures. These should be taken into account when assessing whether the quality of a DIM point cloud meets the requirements of a project.

A good point cloud should not only be accurate but also represent the object details with little random noise. When tuning the parameters in dense matching, the key is to balance between data gap level and noise level. Dense matching quality depends largely on the image contrast and texture. In order to obtain less noisy points from the SURE software on the problematic locations (e.g. narrow streets or in shadow), the parameter *MMC* should be set as large as possible as long as the data gap level is still acceptable. The dense matching can be more challenging in densely-built urban areas. The dense matching quality on open smooth ground with better texture is usually more reliable than locations with poor texture. The DIM data profiles in Figure 2.9 show that dense matching will be problematic in representing the ground details along the narrow alleys.

Concerning the overfitting in the BBA, this effect cannot be detected by evaluating the BBA accuracy based on the RMSEs determined with a few check points as common in many previous studies but can be detected in our evaluation framework. Our finding shows that the RMSEs of check points in the BBA are not equivalent to the point cloud accuracy from dense matching. In our two comparative experiments, the BBA network becomes overfitted when 44 GCPs with high weights are employed in BBA; In contrast, the point cloud GCP05_N+O_PC with only 5 GCPs has already achieved the accuracy better than 1 GSD. It should, however, be noted that when only few GCPs are used, the BBA may become more sensitive to the selection of GCPs. That is, the BBA network is easier to become biased due to one or two inaccurate GCPs.

## 2.7 Conclusions

In order to check the potential of using point clouds derived by dense matching as effective alternatives to laser scanning data, we have presented a framework for evaluating the quality of 3D point clouds and DSMs generated by dense image matching in urban area. Square patches of uniform size are extracted from planar terrain with the guidance of ALS data. The previous evaluation work based on check points simply reveals the BBA accuracy, which is not equivalent to the accuracy of photogrammetric point clouds. In contrast, our evaluation framework based on large sample size is able to reveal the distribution of dense matching errors in a whole photogrammetric block. This framework based on "plane-to-plane" distance is robust to possible blunders and artefacts in the DIM points. Robust quality measures are proposed to represent the dense matching accuracy and precision quantitatively. Experiments show that the optimal accuracy of DIM point cloud is as follows: the overall mean offset to the reference is 0.1 GSD; the maximum mean deviation reaches 1.0 GSD.

In order to further test the usability of the proposed framework, some factors that may affect the DIM quality are studied. Based on our evaluation framework, we find that when oblique images are used in dense matching together with nadir images, the accuracy of DIM point cloud improves, and the noise level decreases on smooth ground areas. The evaluation framework also reports a deviation between the point cloud and DSM generated by a single photogrammetric workflow. The deviation is less distinct when the point cloud density drops. When many GCPs with high weights are employed in BBA, the BBA network may become overfitted, which is reflected in the inhomogeneous distribution of the patch-based DIM errors. This problem cannot be detected by check points in BBA. While this paper evaluates the impact of oblique images and compares the point clouds and DSMs, future work can still study the impact

of other factors (e.g. GCP amounts, image scale, *MMC*, land cover (bare ground or grassland), and the presence of shadow) on the DIM data quality.

# Chapter 3 – Filtering Photogrammetric Point Clouds Using Standard Lidar Filters Towards DTM Generation [2]

[2] This chapter is based on:

Zhang, Z., Gerke, M., Vosselman, G. and Yang, M. Y., 2018. Filtering photogrammetric point clouds using standard lidar filters towards DTM generation. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(2), 319-326.

## 3.1 Introduction

As basic topographical data, Digital Terrain Models (DTMs) are widely used in ortho image rectification, scene classification, 3D reconstruction, etc. Currently, DTMs can be obtained by airborne laser scanning (ALS), digital photogrammetry and interferometric synthetic aperture radar (InSAR) (Chen et al., 2016). During the last two decades, much effort has been paid to filtering the ALS points and obtaining DTMs. DTMs are derived by point cloud filtering followed by interpolation. The second method for DTM generation is aerial photogrammetry. The 3D object coordinates are obtained by matching two or more overlapping images, for instance by dense image matching (DIM). The resulting point clouds can also be used as the basis for DTM production.

While the technique of DTM generation from ALS data is relatively mature after 20 years of development, it is still valuable that we investigate the technique of DTM generation from aerial imagery. Taking the Netherlands as example, normally, a period of five years is required to update the whole national DTM using ALS data. In contrast, aerial images over the country are obtained yearly. Therefore, generating DTM from aerial imagery can significantly shorten the interval for data updating.

Advances in aerial image quality and dense matching techniques provide the feasibility of extracting high quality DTMs from aerial images. Firstly, aerial images are obtained with higher radiometric quality. On-board GPS and Inertial Measurement Unit (IMU) allow to obtain more and more accurate orientation elements for bundle adjustment. Development in dense matching algorithms, e.g. Patch-based Multi-View Stereo (PMVS) (Furukawa and Ponce, 2010) and Semi-global Matching (SGM) (Hirschmüller, 2008) makes it possible to obtain accurate point cloud. nFrames SURE states that the vertical accuracy of their products can be better than 1 pixel. Pix4Dmapper ("Pix4D" are used below) also reports 1-3 GSD vertical accuracy. The evaluation based on roof segments in (Zhang et al., 2017) also confirms that the vertical accuracy achieved by Pix4D is better than 2 GSD. These numbers give rise to the assumption that it is possible to generate accurate DTMs from dense matching points.

The aim of this paper is to study whether the standard Lidar filters can be used to filter DIM points towards DTM generation. Some previous studies have compared the characteristics of point clouds from laser scanning and dense matching. Accuracy and noise level are the two critical factors that influence the final DTM quality. In the airborne cases, the vertical accuracy of dense matching is usually worse than the accuracy from laser scanning. Compared to the ALS point cloud, the noise level of the DIM data depends on the dense matching algorithm and denoising method (Ressl et al., 2016; Zhang et al., 2017). In ALS points data gaps may appear on wet terrain surface while in DIM

points data gaps appear due to failing image matching. These data gaps will cause problems in DTM interpolation.

The chapter is structured as follows: In Section 3.2, we review some work of DTM generation from ALS data and DIM data. Section 3.3.1 introduces the data and experimental setup. Section 3.3.2 studies the robustness of standard Lidar filter to DIM noise and artefacts. Section 2.3.3 evaluates the filtering result on the DIM points in urban scenes. Based on the filtering result in Section 3.3, Section 3.4 evaluates the potential DTM accuracy derived from DIM point clouds. Section 3.5 concludes the work. The paper not only shows the deficiencies within the DIM points compared to ALS points, but also discusses the research problems related to generating accurate DTMs from DIM points.

## 3.2 Related work

Since the end of 1990s, optical sensors, radar systems and laser scanning systems have been widely used to capture topographic data (Li, 2004). 3D object coordinates are commonly obtained by photogrammetry and laser scanning. DTMs are generated through filtering point clouds and then interpolating on the ground points. It has been a hot research topic to develop robust algorithms for filtering ALS points (Meng et al., 2010; Chen et al., 2017).

Point cloud filtering is the process of discriminating between ground and non-ground points. Generally, the filtering algorithms can be divided into five categories: morphological filtering (Kim and Shan, 2011), surface-based filtering (Kraus and Pfeifer, 1998), progressive TIN (Triangulated Irregular Network) densification (Axelsson, 2000), segment-based filtering (Lin and Zhang, 2014), classification-based filtering (Hu et al., 2016). A quantitative comparison of eight filtering methods can be found in (Sithole and Vosselman, 2004). They found that filtering based on the local surface estimation was generally better than global filtering. Also no filter worked perfectly on various scene complexity. Nowadays, these standard Lidar filters are relatively mature and have already been implemented in many commercial software for laser scanning data processing, e.g. LAStools, SCOP++, Terrasolid.

Recently some studies concerning DTM generation from dense image matching data were published. Among these studies, it is quite common that the filtering operation is run on the DSM instead of on the raw point clouds. The reason is that DSM interpolated from the DIM points is less noisy than the raw points while it still retains a similar accuracy (i.e. the bias level to the ground truth). Perko et al. (2015) and Mousa et al. (2017) filtered DSMs using a Multi-directional and Slope Dependent filtering algorithm. Their DSMs were generated from satellite images and airborne images, respectively. Zhang et

al. (2016) filtered a medium resolution DSM from satellite images by using a two-step semi-global filtering method. Beumier and Idrissa (2016) tried to recognize the ground locations from the DSMs using a mean shift segmentation followed by a local regional filtering. In the DTM generation module of Pix4D, the software takes DSMs as input. The ground objects (e.g. buildings and trees) are identified and removed based on the local height gradient. Then the DSM is smoothed and interpolated into the final DTM.

In addition, there are also a few studies filtering the raw DIM points. In general, the standard Lidar filter requires a precise point cloud with little noise as input. Yilmaz and Gungor (2016) compared the effects of five standard filters on the raw DIM points derived from UAV images. Debella-Gilo (2016) filtered the DIM points based on slope-based filtering aided by an existing lower-resolution DTM. However, they did not report on the noise level of the DIM point cloud or any denoising operation.

Among the studies of generating DTM from photogrammetric point clouds, it is common to use the standard Lidar filtering algorithms or ideas to filter DIM points or DSM. Obviously, the noise level in the point clouds or DSMs has a major impact on the filtering result. However, no study has studied the impact of point cloud noise on the filtering result and thus the final DTM accuracy. In this paper, a comprehensive evaluation of the impact of noise level on the filtering result is implemented. We also evaluate the potential DTM accuracy that can be achieved in case that DIM points are filtered and then interpolated.

## 3.3 Filtering DIM points using standard LiDAR filter

In this section, we present some observations on filtering DIM points using the standard Lidar filter - LASground. The filtering algorithm used in LASground is a modification of the TIN-based approach by (Axelsson, 2000). The lowest points at the initial grid cells in the point cloud are selected as seed points; and then TIN facets are built using these seed points. The coarse TIN surface is densified with the remaining points by judging distance and angle - related criteria. LASground is widely used to filter ALS point cloud. It has been used to create DTM from photogrammetric DSM. In contrast, in this paper it is used to filter the raw photogrammetric point cloud. The research question is whether LASground can be used to filter point cloud from dense matching in which there are usually more random noise than in the Lidar data.

### 3.3.1 Pre-processing and experimental setup

The study area lies in the city center of Enschede, The Netherlands as shown in Figure 3.1. In total, 510 aerial images including 102 nadir images and 408 oblique images were obtained by Slagboom en Peeters in 2011. The Ground Sampling Distance (GSD) of nadir images is 10 cm. Bundle adjustment was run in Pix4D Pix4Dmapper (version 3.2) using the initial exterior orientations (EOs) and 15 evenly distributed GCPs. After bundle adjustment, the same EOs are used for dense matching in nFrames SURE (version 2.1.0.33) and Pix4D, respectively. Some dense matching parameters are set as below: in both software, the image scale is set to 1/2 resolution; the Minimum Model Count (MMC) in SURE is set to 2; the Minimum Number of Matches (MNM) in Pix4D is set to 3. Note that MMC and MNM in the two software are not comparable because the dense matching algorithms in them are different: SURE employs the tube-shape Semi-global matching (tSGM) (Rothermel et al., 2012) while Pix4D employs patch-based multi-view stereo. Our criterion for adjusting MMC and MNM is to balance the noise level and data gap level in the point cloud by visual inspection.

The ALS data of the same area were acquired by FLI-MAP 400 system mounted in a helicopter in 2007. The point cloud density is 10 points/$m^2$ and the maximum systematic error in height is 5 cm (van der Sande et al., 2010). The ALS data will be used as reference when evaluating the filtering result in Section 3.3.3 and when evaluating the potential DTM accuracy in Section 3.4.


***Figure 3.1:*** *Orthoimage of the study area. The two regions within the yellow rectangles are used in Section 3.3.1. The region within the red rectangle is used in Section 3.3.2. The potential DTM accuracy of the whole area is evaluated in Section 2.4. The area for the two yellow regions, red regions and the whole study region is 880 m2, 6624 m2, 0.04 km2, 1.6 km2, respectively.*

### 3.3.2 Robustness of LiDAR filter to point cloud noise

Similar to DTM extraction from ALS point cloud, we assume that DTM sample points can be obtained from two land cover types: paved (or bare) ground and grassland. In this section, we only select pieces of smooth terrain and homogeneous grassland for evaluating the impact of random noise on the filtering. The filtering effect on the bumpy terrain or other small objects is not studied here. Two homogeneous and smooth regions marked by the yellow rectangles in Figure 3.1 are used for tests: the left one is smooth ground paved by concrete; the right one is grassland.

Several parameters in LASground affect the filtering performance. Since our study area is in urban area, the scene is set to "city or warehouses" (i.e. a step size of 25 m) and the parameter for controlling the initial ground points density is set to "default". In addition, we also experimented with the parameters "spike size" and "bulge size". Since the surface of the paved ground and grassland is smooth with little spike (often outliers), these two parameters do not make a difference on the filtering. We also try to adjust the parameter "stddev" which controls the maximal standard deviation for planar patches to be retained. Interestingly, tuning "stddev" did not bring a remarkable change to the filtering result. Therefore, we adopt the "10 cm" suggested by the software.

In order to study the impact of the noise level on the filtering performance, a local evaluation method is used. Square patches of 2 m × 2 m are selected from the ALS data of the area. The patches are selected randomly as evaluating units. The Residuals of Plane Fitting (RPF) is calculated using all the points inside the patch.

$$RPF = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \Delta H_i{}^2} \tag{3-1}$$

N is the number of points in this patch. $\Delta H_i$ is the distance from the $i$ th point to the plane which is fitted to all the points within this patch. The patch will be valid only if *RPF* is smaller than 2 cm. When RPF of the ALS data in a certain patch is smaller than 2 cm, we can say that the terrain in this patch is quite smooth and planar. The patches selected on the paved ground and grassland are shown in Figure 3.2. 112 and 527 patches are selected on the paved ground and grassland, respectively. Note that on the grassland in Figure 3.2 the patches are all selected on smooth grassland. No patch lies on the bushes or trees. After patch selection, the filtering result and noise level are quantized locally within each patch:

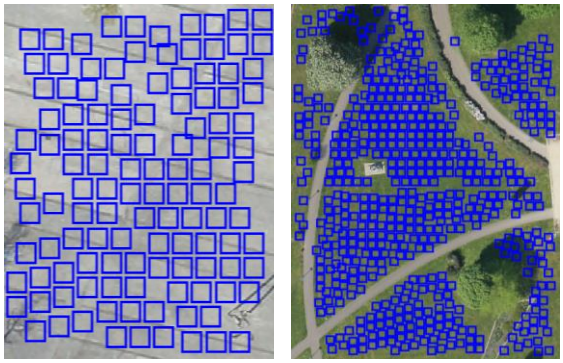
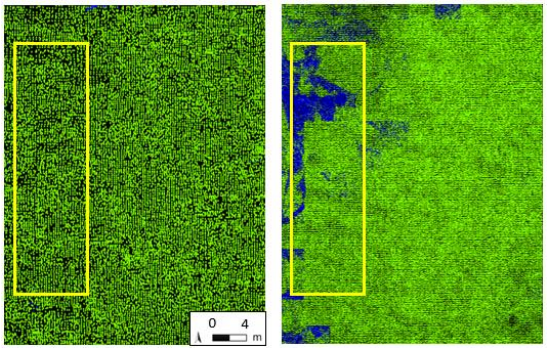


***Figure 3.2:*** *Selected patches on the paved ground (left, 112 patches) and grassland (right, 527 patches) for evaluating the filtering performance. The patch sizes are 2 m × 2 m.*

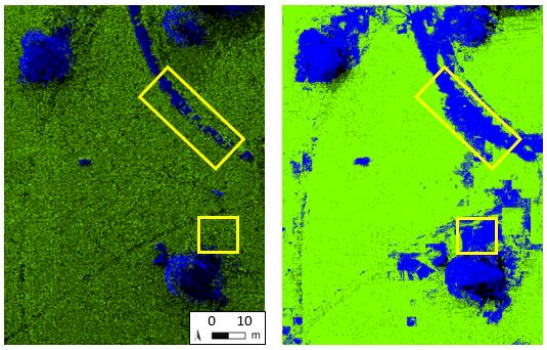
1) Filtering effect: Ideally, all points in every patch in Figure 3.2 should be classified as ground points by LASground. In consideration of scarce outliers or misclassifications, if more than 95% of the points within a patch are classified as ground points, we still take it as correct filtering; if the ratio is less than 95%, the filtering in this patch is incorrect.

2) Noise level: Height Ranking Range (*HRR*) is used to represent the noise level. It is calculated by sorting the heights of all points within a patch. The *HRR* is obtained by subtracting the *m* percentile from the *n* percentile ($m<n$). HRR represents the height range in the vertical direction. Generally, it is robust to blunders in the point cloud. In this paper, m and n are set to 5% and 95%, respectively.

The filtering results from LASground are shown in Figure 3.3. In Figure 3.3(a-d), the percentage of correctly classified patches is 100%, 80%, 100% and 89%, respectively. LASground performs very well on Pix4D point cloud because the point cloud is precise with little noise. Compared with filtering Pix4D point cloud, Figure 3.3(d) shows that filtering SURE point cloud meets more difficulty along the bush and in the shadow. The SURE point cloud is much noisier than Pix4D and this brings problems during filtering.



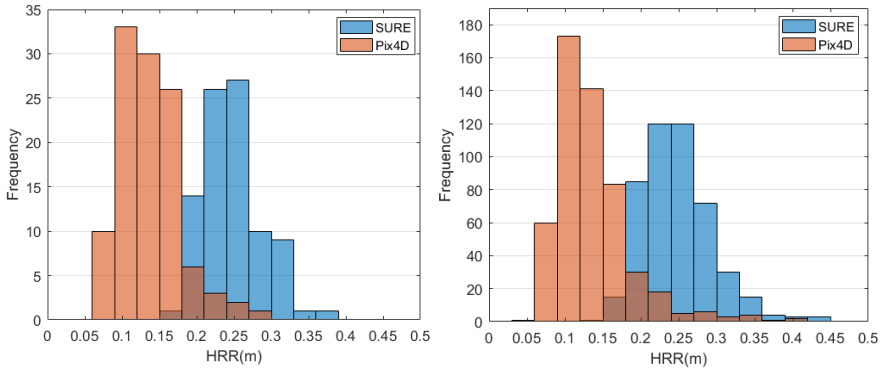

(a) paved ground, Pix4D (b) paved ground, SURE




(c) grassland, Pix4D (d) grassland, SURE

***Figure 3.3:*** *Filtering results on the paved ground and grassland. The green indicates the identified ground points; blue indicates non-ground points; black indicates data gaps. In (c) and (d), blue indicates identified non-ground points not only on the grassland, but also on the trees and bushes (cf. Figure 3.2). Generally, the Pix4D point clouds in Figure 3.3(a) and (c) are darker than SURE point clouds in Figure 3.3(b) and (d) due to a lower point density.*

In order to evaluate the robustness of LASground to point cloud noise, the distribution of the HRR values for all the patches correctly filtered are shown in Figure 3.4. The HRR values in the four histograms range from approximately 0.05 m to 0.40 m which indicates that LASground performs well in filtering a point cloud with a HRR smaller than 0.40 m.




(a) (b)

***Figure 3.4:*** *HRR Distribution for all the correctly filtered patches. Bin width is 3 cm. (a) paved ground; (b) grassland. The dark brown between the blue and light brown histograms is the overlap of the two histograms.*

In addition, the mean of HRRs for paved ground-Pix4D, paved-ground-SURE, grassland-Pix4D, grassland-SURE are 0.14 m, 0.24 m, 0.14 m, 0.24 m, respectively. This indicates that the noise level of the dense matching point clouds on paved ground and grassland are the same, for either Pix4D or SURE. To the best of our knowledge, the noise level of the point cloud from SURE is dependent on the image quality, image overlapping, orientation accuracy and dense matching algorithm. SURE does not implement any post-processing on the dense matching point cloud.

Now we study the patches which are wrongly filtered, i.e. less than 95% points within the patch are classified as ground points. Figure 3.5 visualizes the HRR values of these wrongly filtered patches. The color coding from blue to red indicates that the HRR increases. HRR in these wrongly filtered patches ranges from 0.2 m to 0.59 m. The right figure of Figure 3.5 shows that DIM point cloud from SURE is relatively noisy and contains more artefacts in the shadow than other areas. Therefore, these areas in the shadow are challenging for LASground.

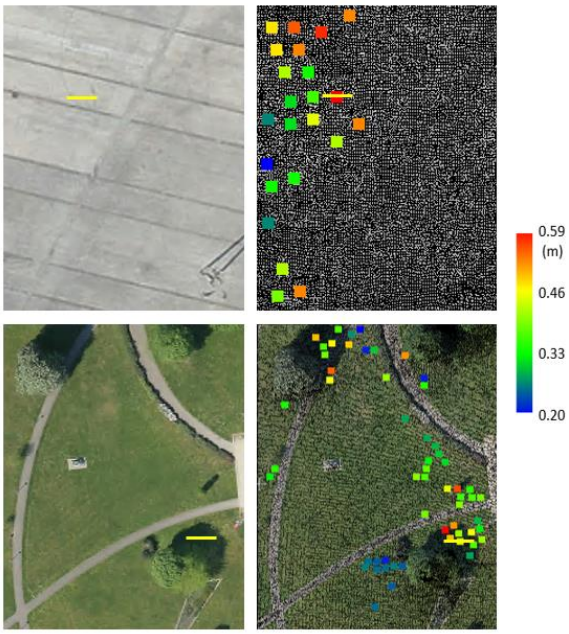



***Figure 3.5:*** *Visualization of the HRR values for the wrongly filtered patches in the SURE points. Top: 23 patches on the paved ground; Bottom: 58 patches on the grassland.*

Figure 3.6 shows the two profiles on paved ground and grassland drawn in Figure 3.5 (along the yellow lines). Checking the orthoimages and laser points shows that the profile in the left paved ground of Figure 3.5 is smooth ground with no bumps or spikes. The profile in the right grassland of Figure 3.5 is the grassland in shadow. The length of the point cloud profile is approximately 2 m and the vertical depth is 20 cm.

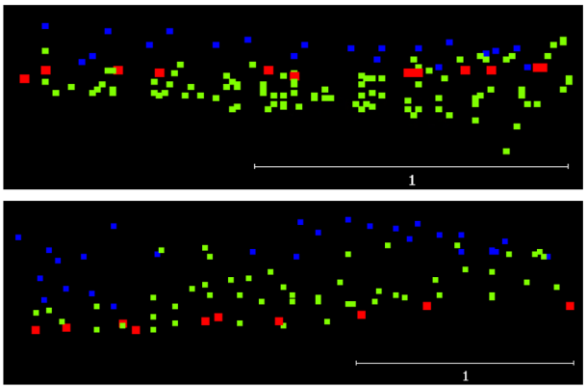


***Figure 3.6:*** *Profiles of three point clouds: ALS points (red), SURE ground points identified by LASground (green) and non-ground points (blue). Top: Profile of the line on paved ground; Bottom: Profile of the line on grassland.*

Figure 3.6 shows that some artefacts exist in the SURE point cloud. Note that the blue points and green points together form the SURE points. In the top figure of Figure 3.6, the ALS point cloud distributes between the "ground points" and "non-ground points" identified by LASground. The HRR is about 0.5 m. As the higher DIM points are classified as non-ground, the average height of the ground points shows a bias w.r.t. the average height of the ALS points.

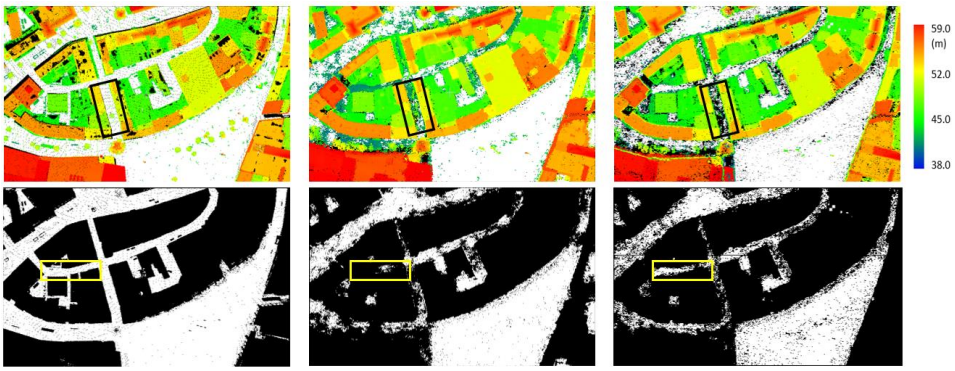
In the bottom figure of Figure 3.6, hollow space can be found inside the SURE points and the points show two layers. LASground simply takes the points in the top layer as the non-ground points. The HRR is about 0.8 m. Along this grassland profile, the ALS points are located at the bottom of the DIM points.

### *3.3.3 Filtering photogrammetric points in urban scene*

In this section, a 0.04 $km^2$ study area (red rectangle in Figure 3.1) is filtered using LASground. This area is mainly covered with buildings, streets, paved ground and individual trees. In some locations, the streets are narrow and covered with shadow. Concerning the filtering parameters in LASground, "step size" shows a large impact on the filtering result: if it is set very large, some roof points will also be taken as ground points. After some trials, we set the parameter according to the scene - "city or warehouses". That is, the step size is fixed to 25 m in this section.

Considering the possible artefacts and random noise in the DIM point cloud, a ranking filter is used to refine the raw point clouds. The rationale of ranking filter is to rank the heights of all points within a vertical raster cell. In our case, the median of the heights (i.e. 50% percentile) is taken as the final value assigned to this cell. The cell size is set to 0.5 m × 0.5 m based on heuristics. The cell size should be set small enough to contain sufficient terrain details and should be set large enough to contain points in most cells. If less than 3 points exist in a certain cell, this cell will not be assigned any value but just left empty.

Three point clouds are filtered as shown in Figure 3.7: ALS data, raw Pix4D point cloud (DIM-raw), Pix4D point cloud processed by a ranking filter (DIM-RF). We do not present the filtering results of SURE points because the filtering delivers more mistakes when the points are too noisy, especially on the narrow streets. Figure 3.7(a) shows the filtering result of ALS data. Building and individual trees are filtered out successfully. The black rectangle shows the filtering result on the narrow street. Here LASground works well.




(a) ALS data (b) DIM-raw (c) DIM-RF

***Figure 3.7:*** *Filtering results of a city block. The top row shows both the ground and non-ground points. White indicates the ground points identified by LASground; black indicates data gaps. Non-ground points are colored based on the height value. The bottom row shows only the ground points. The two figures in the first column entitled (a) is the filtering effect of ALS data; (b) shows the filtering effect of the raw point clouds generated by Pix4D; column (c) shows the filtering effect of the Pix4D point cloud processed by a ranking filter. For the meaning of black and yellow boxes, please refer to the text.*

Figure 3.7(b) shows the filtering result of the raw Pix4D point cloud. Dense matching is challenging in shadow area due to poor texture and low contrast in images. Ideally, all the ground points should be labelled as "ground", including ground points in the shadow. The black rectangle shows the filtering in the shadow. Some points are identified as ground and some are identified as non-ground. In the yellow rectangle, most of the locations are identified as non-ground. Figure 3.7(c) shows the filtering result of a Pix4D point cloud processed by ranking filtering.

Figure 3.7(b) and (c) show that LASground performs well at filtering individual trees on both the DIM-raw and DIM-RF data, especially on the southeast open square. In the black rectangles, there are more ground locations identified in DIM-raw than in the DIM-RF. This narrow street is located in shadow. Checking the data profile shows that the heights of the DIM points are higher than the real ground surface by approximately 30 cm, and the DIM points are randomly distributed because of remaining matching errors. The DIM-RF identifies fewer ground points than DIM-raw but the identified ground points are more likely to be reliable ground locations.

The yellow rectangles show the filtering effect of a road, which is not in the shadow. LASground filters classified most of the points in Pix4D-raw data as non-ground. In contrast, many locations are taken as ground points in the DIM-RF data.

In both the black and yellow rectangles, LASground tends to deliver better filtering results on the DIM-RF data than the DIM-raw data. It can be explained by the fact that median ranking filter can reduce the noise in the DIM points. The DIM point cloud after pre-processed by a ranking filter is getting more similar to the ALS data in terms of ground representation. Moreover, the noise is removed very considerably and height jumps from ground to above-ground objects are more or less better retained because of the relatively large raster. In this case, LASground can better discriminate ground and non-ground cells because outliers and noise are not affecting the TIN densification step.

Apart from the qualitative comparison above, the filtering results are also evaluated quantitatively using the measures from (Sithole and Vosselman, 2004). The filtering result of ALS data after manual check is taken as the reference. The ALS data and Pix4D-raw data are both 3D while the Pix4D-RF is 2.5D. The filtering result on Pix4D-raw is evaluated as below: Take the surface through the ALS ground points and label the DIM ground points as correct if they are within some margin of the ALS ground surface. To evaluate the 2.5D filtering result, the ALS data are also converted to 2.5D and only the label of the highest point in each bin is taken as the true label. Three quantitative measures are calculated: Type I error is the percentage of bare ground points actually labelled as non-ground points by LASground; Type II error is the percentage of non-ground points labelled as ground points; Total error is the overall statistics of points being wrongly classified. The filtering results are shown in Table 3.1.

***Table 3.1:*** *Quantitative evaluation of the filtering results*

| Dataset | Type I | Type II | Total error |
|---|---|---|---|
| DIM-raw | 22.3% | 5.2% | 8.7% |
| DIM-RF | 12.0% | 7.0% | 8.4% |

Table 3.1 shows that the total error by filtering DIM-raw (8.7%) and DIM-RF (8.4%) are similar. Type I error of DIM-raw is much larger than if DIM-RF is used. The reason is that many ground points on the narrow streets in shadow are misclassified as non-ground points. These DIM points are usually a mixture of real ground points and blunders. LASground will filter out the above points and only the lowest points will be taken as ground points. In addition, the level of Type II errors is smaller than Type I errors. Type II error of DIM-RF is slightly larger than DIM-raw. If we check the filtering effect of individual trees and objects (e.g. chairs and dustbins) on the southeast square in Figure 3.7(c), the reason for a relatively high Type II error is that some small objects are smoothed by using a median ranking filter. LASground will classify these locations into ground while the ground truth is non-ground. In contrast, the details of small objects can be better retained in the DIM-raw data. When

filtering DIM-raw data, the ground and non-ground points can be better separated.

In summary, the advantage of using a ranking filter on the point cloud is that the filtered point cloud contains less noise. When filtering the points after ranking filtering, LASground performs better in avoiding non-ground points. That is, compared to filtering the raw DIM points, filtering DIM-RF will deliver less ground locations with higher reliability. On the other hand, the disadvantage of using ranking filter is that some low objects may be smoothed. These non-ground locations are thus likely to be misclassified as ground by LASground. In contrast, the details of small objects can be better retained in the DIM-raw data. When filtering the DIM-raw data, the ground and non-ground points can be better separated by LASground.

## 3.4 Evaluation the potential accuracy of DTMs

### 3.4.1 Comparison of DTM accuracy derived from Pix4D and SURE point clouds

The observations in Section 3.3 indicated that LASground is quite tolerant to the random noise when filtering the DIM points. In particular, all the DIM points on the paved ground, bare earth and grassland are likely to be taken as terrain points by LASground. In this section, we explore the potential accuracy that can be obtained by DTM derived from dense matching. We do not interpolate on the point cloud but we directly calculate the deviation of the DIM point cloud from the reference. The ALS data are taken as reference data and only the vertical accuracy is studied. In the evaluation stage, the square patches of 2 m × 2 m are taken as the evaluation unit. Compared to the point-to-point comparison, the accuracy measures calculated based on each patch are more robust to local blunders and random noise. The study area is the whole region shown in Figure 3.1 (1.6 $km^2$).

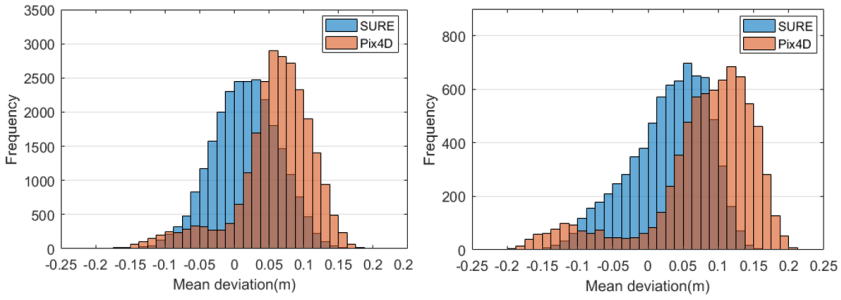
First, the ALS data are filtered using LASground. Then, square patches are detected from the ground points. A patch is valid if it meets two conditions: (1) The number of points in this patch is larger than a certain threshold; (2) The RPF (Eq. 3-1) is better than 2 cm. The patches in shadow are eliminated. The shadow mask is calculated from an orthoimage based on a grayscale histogram (Sirmacek and Unsalan, 2009). Only if all the four corners and the center location of a certain patch lie in the non-shaded locations, the patch will be taken as valid. The selected patches are divided into two categories based on the green index on the ortho image: ground and grassland. Finally, 24,634

ground patches and 7381 grassland patches are selected for accuracy evaluation.

After the patches are detected from the ALS point cloud, the DIM points within the square patch boundary in 2.5D space are cropped for evaluation. Concerning a certain patch, a plane is fitted to the ALS points, the mean deviation from the DIM points to the plane is calculated as the accuracy measure as shown in Eq. (3-2). $\mu_i$ denotes the mean deviation between the DIM points and the ALS points for the j*th* patch. i denotes the i*th* patch in the whole study area, j denotes the j*th* point in this patch. There are $n_i$ points in this patch. $\Delta h_{ij}$ is the distance from the j*th* point to the fitted ALS plane. $\mu_i$ is the mean deviation between the DIM points and the ALS points for the j*th* patch.

$$\mu_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \Delta h_{ij} \tag{3-2}$$

The distribution of mean deviations is shown in Figure 3.8. Interestingly, the distribution of the deviations for Pix4D and SURE are quite different even though the same EOs were used for dense matching. Figure 2.8 also shows that there is only one peak in the SURE histograms but there are two peaks in the Pix4D histograms. The mean deviation on the ground ranges in [-0.18 m, 0.18 m] for Pix4D data, and ranges in [-0.15 m, 0.15 m] for SURE data. The mean deviation on the grassland ranges in [-0.2 m, 0.2 m] for Pix4D data, and ranges in [-0.15 m, 0.15 m] for SURE data.




(a) (b)

***Figure 3.8:*** *Distribution of mean deviations for the DIM points generated by Pix4D and SURE. (a) 24,634 ground patches; (b) 7381 grassland patches. Note that the dark brown between the blue and light brown histograms is actually the overlapping of the two histograms.*

In order to make quantitative evaluation of the DIM accuracy in the whole study area, the following two accuracy measures are calculated considering all the patches:

- Mean of mean deviations:

$$\bar{\mu} = \frac{1}{m}\sum_{i=1}^{m}\mu_i \tag{3-3}$$

- Standard deviation of mean deviations:

$$\sigma_{\mu_i} = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(\mu_i - \bar{\mu})^2} \tag{3-4}$$

$\bar{\mu}$ is calculated by averaging the mean deviations in the whole area. $m$ is the number of patches in the whole study area. The $\sigma_{\mu_i}$ is calculated to represent the standard deviation of the mean deviations from the $\bar{\mu}$. The accuracy measures at the whole block are shown in Table 3.2.

***Table 3.2:*** *Accuracy measures of DIM point cloud in the whole block. (Unit: m)*

| Dataset | $\bar{\mu}$ | $\sigma_{\mu_i}$ |
|---|---|---|
| ground-pix4d | 0.057 | 0.056 |
| ground-sure | 0.016 | 0.048 |
| grassland-pix4d | 0.078 | 0.077 |
| grassland-sure | 0.030 | 0.056 |

Table 3.2 shows that $\bar{\mu}$ of SURE point cloud is better than for the Pix4D point cloud on both ground and grassland as could already be seen in the histograms of Figure 3.8. In addition, the $\sigma_{\mu_i}$ of SURE point cloud is better than Pix4D point cloud on both ground and grassland.

Table 3.2 also shows that the bias between the DIM data and the ALS data on the grassland is larger than the bias on the ground. That is, the accuracy on the grassland is worse than the ground. This can be explained by that dense matching usually delivers the points on the top surface of the grassland but laser scanning can penetrate the shallow grass and record the points on the real terrain. Therefore, the bias on the grassland includes not only the dense matching errors but also the grass height (Ressl et al., 2016).

When filtering the DIM point clouds in the urban scene using LASground, all the points on the ground and grassland will probably be classified as ground points without the negative impact of artefacts. However, the problem is that dense matching will deliver some points higher than the true terrain on the grassland, which will result in incorrect elevated DTMs.

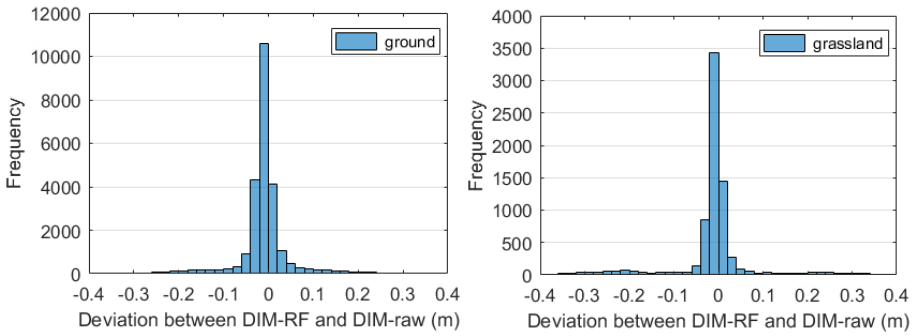
### *3.4.2 The impact of ranking filter on the potential DTM accuracy*

In Section 3.3, we found that a ranking filter leads to improvements in the ground point filtering. In this section, we check whether the ranking filter would have an impact on the potential DTM accuracy achieved by the Pix4d point cloud. Similar to Section 3.4.1, the mean deviations for 24,634 ground patches and 7381 grassland patches are calculated and incorporated into the mean of mean deviations $\bar{\mu}$ and standard deviation of mean deviations $\sigma_{\mu_i}$ as shown in Table 3.3. RF indicates that this point cloud is preprocessed by a ranking filter.

***Table 3.3:*** *Accuracy measures of DIM point cloud after pre-processed by a ranking filter. (Unit: m)*

| Dataset | $\bar{\mu}$ | $\sigma_{\mu_i}$ |
|---|---|---|
| ground-pix4d-RF | 0.048 | 0.063 |
| grassland-pix4d-RF | 0.067 | 0.085 |

Table 3.3 shows that for both the ground and grassland, when RF is used in a preprocessing step, $\bar{\mu}$ gets improved by around 1 cm. However, $\sigma_{\mu_i}$ increases slightly. That is, when the point cloud is pre-processed by a ranking filter, generally the potential DTM accuracy will improve but the ranking filter will also bring more variation to the DTM errors at the whole photogrammetric level. In addition, we can study the impact of a ranking filter onto the point cloud accuracy by calculating the deviation between DIM-RF and DIM-raw for every patch. Figure 3.9 shows the distribution of deviation values for ground patches and grassland patches, respectively. According to statistics, on 13.3% grassland patches and 8.6% patches the deviations between DIM-RF and DIM-raw are larger than 10 cm. The deviation values are relatively small compared to the large patch size (2 m × 2 m). In addition, the deviations between DIM-RF and DIM-raw on the paved ground is generally smaller than on the grassland, which can be explained by the fact that there are usually more artefacts and surface fluctuation on grassland.




(a) (b)

***Figure 3.9:*** *Distribution of deviation between DIM-RF and DIM-raw. (a) paved ground; (b) grassland.*

## 3.5 Conclusions

This work studies the question whether the standard Lidar filters can be used to filter dense matching points in order to derive accurate DTMs. Filtering results on the homogeneous ground and grassland show that the filtering performance depends on the noise level and scene complexity. LASground is verified to be relatively robust to random noise. However, filtering algorithms may only select the lower points as ground points in case of a large amount of noise. In addition, artefacts and blunders may appear in the dense matching points due to low image contrast or poor texture (e.g. in the shadow, along the narrow street, etc.). In these cases, LASground will probably classify some noisy ground points as non-ground points. Filtering results on a city block show that LASground performs well on the grassland, along bushes and around individual trees if the point cloud is sufficiently precise. In addition, a ranking filter can be used to filter the DIM point cloud before LASground filtering. LASground will identify fewer but more reliable ground locations. However, a ranking filter will also smooth some ground details so some small objects on the terrain will be filtered out. Since we aim at obtaining accurate DTMs, the ranking filtering shows its value in identifying only reliable ground points.

The accuracy of the point cloud determines the final DTM accuracy. The accuracy of the DIM point clouds is evaluated using a patch-based method. The bias from the reference is studied in the whole study area. Although the same EOs are used for dense matching, the vertical accuracy of SURE point cloud on the ground is better than the Pix4D point cloud. In addition, we also verify that the error on the grassland is larger than the error on the paved ground. We also found that the ranking filter brought only small deviation to the point cloud. Therefore, the ranking filter might be taken a useful pre-processing tool before filtering noisy photogrammetric point clouds. Future work may focus on modifying the previous Lidar filtering algorithms so that they can be used on relatively noisy DIM point clouds.

# Chapter 4 – Detecting and Delineating Building Changes Between Multimodal Data[3]

[3] This chapter is based on:

Zhang, Z., Vosselman, G., Gerke, M., Persello, C., Tuia, D. and Yang, M.Y., 2019. Detecting building changes between airborne laser scanning and photogrammetric data. Remote sensing, 11(20), p.2417.

## 4.1 Introduction

Detecting topographic changes and keeping topographic databases up-to-date in large-scale urban scenes are fundamental tasks in urban planning and environmental monitoring (Matikainen et al., 2004; Holland et al., 2006; Tran et al., 2018). Nowadays, remote sensing data over urban scenes can be acquired through satellite or airborne imaging, Airborne Laser Scanning (ALS), Synthetic Aperture Radar (SAR), etc. In practice, the remote sensing data available at different epochs over a same region are often acquired with different modalities, i.e. with different platforms and sensor characteristics. Such heterogeneity makes the detection of changes between such multimodal remote sensing data challenging.

This chapter aims to detect building changes between ALS data and airborne photogrammetric data. This is applicable to the situation of several mapping agencies, where laser scanning data are already available as archive data, while aerial images are routinely acquired every one or two years for updates. On the one hand, since acquiring the aerial images is much cheaper than acquiring the laser points (Qin et al., 2014), aerial photogrammetry is widely used for topographic data acquisition. On the other hand, since the ALS data are generally more accurate and contain less noise compared to dense image matching (DIM) data (Zhang. et al., 2018a), the fine ALS data can be used as the base data and be updated using dense matching points in the changed areas.

A traditional photogrammetric pipeline takes 2D multi-view images as input and outputs 3D dense matching point clouds with true colors, 2.5D Digital Surface Models (DSMs), and 2D orthoimages. Quantitative comparisons of the point clouds from ALS and dense matching are found in (Remondino et al. 2014; Nex et al., 2015; Ressl et al 2016; Mandlburger et al., 2017). Point clouds from laser scanning and dense matching differ in geometric accuracy, precision (i.e. noise level), density, the amount and size of data gaps, and available attributes. A detailed comparison between laser scanning points and dense matching points were made in Section 1.3.1.

Apart from the problems with multimodal point clouds, object-based change detection becomes more challenging due to the complexity of the scene. First, false positives may appear if the shape of a changed object is similar to a building, for example, changes of scaffolds, trucks, containers or even terrain height changes in construction sites. Second, changes on a connected object might be mixed: for instance, one part of a building could be heightened while another part lowered.

This chapter presents a robust method for multimodal change detection. The **contributions** are as follows:

(1) We propose a method to detect building changes and delineate change boundaries between ALS data and photogrammetric data. First, we provide an effective solution to convert and normalize multimodal point clouds to 2D image patches. The converted image patches are fed into a light-weighted pseudo-Siamese Convolutional Neural Network (PSI-CNN) to quickly detect preliminary change locations. Then, precise boundaries of the changed objects are delineated through per-pixel change detection. The coarse-to-fine framework is not only robust to data noise and scene complexity, but also leads to sharp change boundaries with *heightened* or *lowered* labels.

(2) The proposed PSI-CNN is compared to two other CNN variants with different inputs and configurations. In particular, the performance of the pseudo-Siamese architecture and a feed-forward architecture are compared quantitatively and qualitatively. Different configurations of multimodal inputs are compared.

(3) In change delineation, we propose to adopt different feature sets for per-pixel classification for ALS data and DIM data, respectively. After an initial change map is derived, an artefact removal method with morphological operations as backbone is proposed to refine the change map at minimum cost. Finally, the changed pixels are connected as individual changed buildings.

This chapter is organized as follows: Section 4.2 reviews the related works. Section 4.3 presents the method. Section 4.4 provides the results and analyses. Section 4.6 concludes the paper.

## 4.2 Related work

### 4.2.1 3D change detection

The input for change detection is remote sensing data from different epochs. According to the dimension of the input data, the change detection can be divided into 2D change detection, 3D change detection or hybrid change detection. 2D data include multi-spectral images, SAR and aerial images, 2D topographic maps, cadastral maps, and 2D GIS data. 3D data include laser point cloud, digital elevation model (DEM), digital surface model (DSM), and 3D CAD (Computer-Aided Design) model. (Strictly speaking, DSM and DTM are 2.5-dimensional data, not true 3D). If the input data of a certain epoch contains both 2D and 3D data, it is called "hybrid change detection". In recent years,

hybrid change detection has received more and more attention because the integration of multi-dimensional data may effectively reduce the ambiguity of a single data source, thereby improving the reliability of change detection.

On the other hand, according to the categories of detected changes, change detection can be divided into binary change detection, multiclass change detection, and Time Series Analysis (Lu et al., 2004). Binary change detection indicates that the label is a binary value which only distinguishes between changes and unchanged. Multiclass change detection indicates that the semantic information in both the old and new epoch is known. For example, if the semantic label is terrain in the old epoch and a building in the new epoch, the change is a new building.

Comparing 2D and 3D change detection, the quality of 2D change detection mainly depends on the image quality after strict pixel-to-pixel registration and comparison of spectral values. Therefore, the slight change of illumination or viewing angle during image acquisition may deteriorate the correspondence between image pixels, thereby reducing the change detection accuracy. In contrast, 3D change detection has the following advantages:

- Geometric information in 3D data is not affected by changes in illumination or viewing angles during data acquisition. Therefore, change detection of 3D data is more robust to exterior impact compared to 2D change detection.

- The geometric orientation of 3D data can often be accurately determined. The precise geometric model guarantees that the 3D data can be accurately registered. This provides an important premise for 3D change detection.

Qin et al. (2016) divides 3D change detection methods into two categories: geometric comparison and geometry-spectrum analysis as shown in Figure 4.1. (1) Height differencing is the most direct and fundamental method to detect change information by calculating the vertical distance between two point clouds or DSMs. (2) Euclidean distance indicates the plane-to-plane distance between two 3D data to indicate change information. (3) Projection-based differences are commonly used in multi-view images or point cloud change detection. After projecting a 3D object onto an image, the change is detected by comparing the similarities of an object on the 2D image. (4) The post-refinement method first applies geometric comparison to obtain the initial change locations, and then applies other available data sources and features to optimize the results step by step. (5) Direct feature fusion indicates that change information such as geometric change, spectral change or textural change is directly fused. Direct feature fusion methods usually include change

vector analysis (CVA), Dempster-Shafer and supervised classifiers such as SVM, random forest, etc. (6) Post-classification is a common method which performs object detection and classification first, and then detects change information by comparing the labels from two epochs. The following sub-sections review more details in height differencing, point cloud change detection and geometry-spectrum analysis.

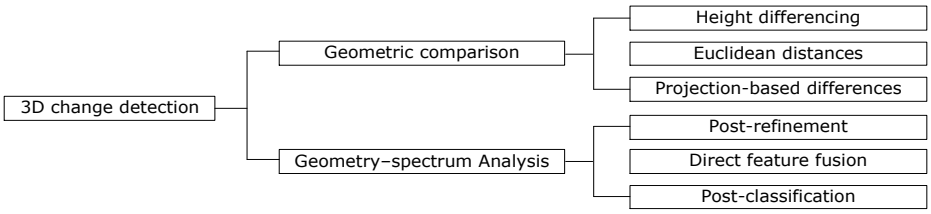



***Figure 4.1:*** *Categories of 3D change detection methods (Qin et al., 2016)*

#### *4.2.1.1 Height differencing*

Height differencing is simple but often effective in the detection of potential change locations. However, mis-registration of two DSMs usually results into linear "false alarms" along the object boundaries in the differencing map (Rutzinger et al., 2010). In addition, the differencing of DSMs generated in different seasons such as summer (leaf-on) and autumn (leaf-off) usually causes false positives in the vegetated areas. Therefore, the initial differencing map needs refinement or post-processing before delivered to the following change verification. In this section, we review some refinement methods and application of height differencing maps.

First, height differencing threshold can be set to eliminate false positives. Murakami et al. (1999) propose a DSM differencing method between two epochs. The DSMs are both derived from ALS data. A threshold of 1m is used to refine the differencing DSM. The changes are manually detected by overlaying the differencing map with an orthoimage. Vu et al. (2004) use an unsupervised method to detect changes in two time-phase ALS point clouds, first performing differential DSM, then using histogram thresholding to detect building changes as post processing.

Second, handcrafted rules can be made to refine the change maps. Jung (2004) detects building changes between two sets of stereopsis. DSMs are obtained through dense image matching, and then decision tree is used to detect changes in the DSM differencing map. Dini et al. (2012) detect building changes between stereo satellite images and GIS databases. A 1 m resolution DSM is generated by semi-global matching. The initial change map is obtained by DSM differencing and mis-registration compensation. Pang et al. (2014) detect building changes from ALS point clouds of two epochs. On the basis of

differencing DSM, connected component analysis is used to connect the changed regions into individual objects, and then RANSAC algorithm is used to distinguish between topographic change and vegetation change. Finally, rules are made to classify building changes into four categories: new, elevated, demolished, and lowered.

Third, morphological operation can be used for differencing map refinement. Choi et al. (2009) generate DSMs from ALS data from two epochs and subtract one from the other to extract initial horizontal boundaries of the changed areas. The refinement process includes binarization, morphological opening, filtering and grouping. A region growing method is used to segment the LiDAR points, following which the segments are classified into ground, vegetation or building according to roughness, size and height features. The change map is derived by fusing the refined differencing map and semantic segments. Tian et al. (2013) propose a 3D change detection method based on DSM differencing. The refinement process includes morphological filtering and shadow mask processing, which effectively eliminate the impact of occlusion in the DSMs and irregularity of building shapes.

#### *4.2.1.2 Point cloud change detection*

Transforming from point clouds to DSMs causes information loss. Some previous change detection work starts from raw point clouds instead of DSMs. It is common to calculate point-to-point distance for change detection, which is usually followed by refinement. Girardeau-Montaut (2005) develops a method of change detection by directly calculating the Hausdorff distance between two Terrestrial Laser Scanning(TLS) datasets. The Octree subdivision principle is used to reduce the computation time. For refinement, the visibility of all the 3D points is determined by depth map, sensor position and orientation. Additional operation is given to the three types of invisible points. Hebel et al. (2013) detect changes between ALS data based on grids along the scan line in real time. They fuse multiple change indicators with Dempster–Shafer.

Instead of raw point-to-point comparison, change detection can also be made in grid cells or voxels. Xu et al. (2015) detect vegetation and building changes between ALS point clouds. First, the point cloud is filtered by progressive TIN densification to obtain non-ground points. The octree structure is constructed for non-ground points, and changes are detected based on adaptive clustering. Xiao et al. (2015) uses detect changes on road surface based on mobile laser scanning data. Changes are detected along the scan line through Dempster–Shafer fusion in each 3D grid. Fuse and Yokozawa (2017) detect changes between mobile laser scanning data. They first build regular grids in the registered data, and then use Dempster–Shafer fusion to remove occlusion and detect changes.

Handcrafted features can not only be used to classify topographic objects, but also be used to distinguish change types. Xu et al. (2015) detect changes in airborne laser scanning points of two epochs. They do not subtract DSM of one epoch from the other but calculate the point-to-plane distance from one epoch to their nearest planes in the other epoch. A rule-based classifier is utilized to classify point clouds of two epochs into seven classes. A rule-based decision is utilized to detect "changed" points in surface difference maps. This surface difference map and scene classification results are used to detect changes of buildings. Based on contextual rules, the building changes are further classified as roofs, walls, roof elements and undefined objects.

Some point cloud change detection work applies post-classification method. Voegtle and Steinle (2004) present a change detection method based on ALS data of two epochs. The point cloud is segmented based on a region growing method and classified into segments of building, vegetation and terrain based on object-based features. The classification methods based on fuzzy logic and maximum likelihood are compared. The changes are detected by calculating the pixel-to-pixel overlapping rate based on height data and the classified objects.

Rutzinger et al. (2010) propose a two-step method of change detection from ALS data of two epochs. The first step is object-based building footprint extraction. A first-last-echo difference model (FLDM) is used to eliminate the vegetation region in the scene. Then building outlines are delineated by height constraint and morphological opening. The segments are further classified into buildings and non-buildings using a decision tree. In the change detection process, shape indices and height difference of the building footprints of two epochs are compared. Xi and Luo (2018) detect changes between ALS data. SVM is first used to classify point clouds into ground, building and vegetation. Morphological algorithms are then used to distinguish between topographic changes and vegetation areas. Handcrafted rules are designed to derive three types of changes: new buildings, elevated buildings and demolished buildings.

Some work aims to detect tree changes between point clouds. Xiao et al. (2012) propose a method of detecting tree changes using multi-temporal ALS data. Trees are represented by irregularly distributed points. The preprocessing includes connected components algorithm, local maxima identification and trunk point removal. Point-based and model-based method are utilized to derive tree crown parameters for comparison. Corresponding trees are compared by overlapping the bounding boxes and point-to-point distances. Similarly in Yu et al. (2006), individual tree change detection is implemented with laser scanning data using method of differentiation between DSMs and

Canopy Height Models (CHM), canopy profile comparison and analysis of height histograms.

#### *4.2.1.3 Geometry-spectrum analysis*

Geometric features and spectral features are complementary in classifying topographic objects. Therefore, apart from geometric features, available spectral features are also a valuable data source which supports change detection.

Some work directly compares images to point clouds after image-to-point cloud registration. Qin and Gruen (2014) detect changes between outdated MLS data and new imagery. First, the road imagery are registered to the point cloud, and then the point cloud is projected onto each image through a Z-buffer method based on the weighted window. Then the stereo image pairs are rectified to the point cloud space, and the similarity between them are calculated. The energy function composed of color, depth and class information is minimized by graph cut to realize change detection.

Much change detection work derives point clouds from imagery and then detect changes by comparing point cloud to point cloud. Ali-Sisto and Packalen (2017) use aerial imagery to detect forestry changes. First, the point cloud is obtained by semi-global matching, the canopy height model and volume model of the two epochs are calculated. The forestry changes are further classified by logistic regression. Pang et al. (2018) detect building changes from bi-temporal dense-matching point clouds and aerial images. Graph cut algorithm is adopted to classify the points into foreground and background, followed by region-growing algorithm to form candidate changed building objects. Structural features are constructed to classify the candidate changed buildings into buildings and non-buildings.

Concerning 3D change detection of satellite imagery, Tian et al. (2014) detect changes between satellite images of two epochs. The DSM height differencing and Kullback–Leibler similarity are fused as change indicators by Dempster–Shafer fusion. Huang et al. (2017) detect topographic changes between multi-view satellite imagery. First, textural and morphological features are extracted from images and DSMs for different epochs for semantic segmentation. Then change detection is performed at the pixel-level, grid-level and block-level. Stylianidis et al. (2019) detect changes of forestry vegetation with satellite images. First, DSMs are generated and registered from satellite images. Second, Euclidean distance is calculated as the change indicators. Finally, biometric volume change is computed with rules.

#### 4.2.1.4 Changes between maps and 3D data

3D change detection can be performed either between 3D data or by comparing 3D data of a single epoch to a bi-dimensional map (Vosselman, 2004). Vosselman et al. (2004) proposed a method to update 2D topographical maps with laser scanning data. The ALS data were first segmented and classified. The building segments were then matched against the building objects of the maps to detect the building changes. Malpica et al. (2013) detected building changes in a vector geospatial database. The building objects were extracted from satellite imagery and laser data using Support Vector Machine (SVM).

Rottensteiner et al. (2007) extract buildings from DSM and multispectral images using the Dempster-Shafer fusion. Then they compare the semantic labels with old maps to classify building changes into four categories: partially changed, new building, demolished and unchanged. Nebiker et al. (2014) first generate DSMs from old images and then extract objects using object-based image analysis (OBIA). Building changes are detected by comparing objects with cadastral maps.

Awrangjeb et al. (2015) filter the ALS points to get non-ground points. The non-ground points are classified into wall points, vegetation points and roof points. After regularization of the building edges, the buildings are compared with a topographic map for change detection. Matikainen et al. (2016) use multi-spectral LiDAR data to update topographic maps. The geometric and intensity features are extracted and classified with Random Forests. Handcrafted rules are designed to detect building changes.

### 4.2.2 Multimodal change detection

Change detection is the process of identifying differences in an object by analyzing it at different epochs (Singh, 1989). The input data of two epochs can be either raw remote sensing data or object information from an existing database (Qin et al., 2016). Zhan et al. (2017) classified the change detection methods into two categories based on the workflow: post-classification comparison (e.g. Wu et al., 2017; Mou et al., 2019) and change vector analysis (e.g. Choi et al., 2009; Volpi et al., 2015; Xu et al., 2015; Gong et al., 2017).

In post-classification comparison, independent classification maps are required for both epochs. Change detection is then performed by comparing the response at the same location between the two epochs. When the data of two epochs are of different modalities, both training and testing have to be performed at each epoch separately, thus requiring a large computational effort. Moreover, errors tend to be multiplied along object borders due to mis-classification errors in the single classification maps (Volpi et al., 2013).

In contrast, change vector analysis relies on extracting comparative change vectors between the two epochs and fuses the change indicators in the final stage (Chen et al. 2010; Tian et al., 2014; Volpi et al., 2015). Compared with post-classification comparison, change vector analysis directly makes a comparison between the data of both epochs. However, traditional change vector analysis is sensitive to data problems and usually causes many false detections, especially when the data of two epochs are in different modalities. The most widely-used change vector analysis between 3D data sets is DSM surface differencing, followed by point-to-point or point-to-mesh comparison (Remondino et al. 2014; Mandlburger et al., 2017). To reduce the number of false positives, direct comparison methods are often followed by post-processing methods or are combined with other change detection frameworks.

Considering detecting changes between multimodal 3D data, Basgall et al. (2014) compared laser points and dense matching points with the CloudCompare software. Their study area was small and only one changed building was studied. Also the method proposed was not automatic, since the changed building was detected through visual inspection. Qin and Gruen (2014) detected changes between mobile laser points and terrestrial images at street level. Image-derived point clouds were projected to each image by a weighted window based Z-buffering method. Then an over-segmentation based graph cut optimization was carried out to detect changes in the image space.

Du et al. (2016) detect changes between aerial imagery and ALS points, but the sequence of their input data is contrary to ours. Firstly, the two sets of point clouds are registered by ICP, and height differencing and gray level similarity are calculated as change indicators. Changes are detected by graph cut optimization.

Same with our inputs, Zhou et al. (2020) detect and update changes in LiDAR data using aerial imagery based on a two-step dense matching framework. First, LiDAR-guided edge-aware dense matching is used to derive accurate partial changes since accurate LiDAR data guarantees robust matching in the shadow and low texture areas. Second, hierarchical dense matching is employed to derive complete changes and update 3D information.

Politz et al. (2021) detect building changes between ALS data and photogrammetric data. They first classifies each point cloud from two epochs by semantic segmentation (Politz et al., 2020). The semantic segmentation method is to transform and normalize the geometric point cloud distribution into regular raster and feed them into an extended U-Net architecture. Then two types of change indicators are derived: height change and class change.

The two change indicators are fused based on handcrafted rules for change detection.

### 4.2.3 Deep learning for multimodal data processing

Recently, deep CNNs have demonstrated their superior performance in extracting representative features for various computer vision tasks, e.g. image classification (Krizhevsky et al., 2012; Szegedy et al., 2015), semantic segmentation (Long et al., 2015; Sherrah, 2016; Audebert et al., 2018), object detection (Ren et al., 2015). As a specific CNN architecture, Siamese networks (SI-CNN) perform well in applications which require to compute similarity or to detect changes between two inputs. Outputs from SI-CNN can be patch-based single-valued or dense pixel-by-pixel maps, depending on the specific architecture. In patch-based prediction, SI-CNN has been widely used in handwritten digit verification (Bromley et al., 1994), face verification (Chopra et al., 2005, Taigman et al., 2014), character recognition (Koch et al., 2015), patch-based matching (Zagoruyko and Komodakis, 2015) and RGB-D object recognition (Eitel et al., 2015). In the case of dense predictions, SI-CNN was used in wide-baseline stereo matching (Zbontar and LeCun, 2015; Luo et al., 2016).

In the remote sensing domain, deep learning has also been used to process two sets of input in e.g. change detection and image matching. He et al. (2018) used a SI-CNN to find corresponding satellite images with complex background variations. The coordinates of the matching points were searched using the Harris operator followed by a quadratic polynomial constraint to remove false matches. Similarly, Lefèvre et al., (2017) used a SI-CNN to detect changes between aerial images and street level panoramas re-projected on an aerial perspective. AlexNet (Krizhevsky et al., 2012) was used in the two branches for feature extraction and the Euclidean distance was used to determine the similarity between the two views. Mou et al. (2017) identified corresponding patches between SAR images and optical images using a pseudo SI-CNN ("pseudo" indicates that the weights in the two branches are unshared). The feature maps from the two Siamese branches were concatenated, which worked as a patch comparison unit. Although SAR images and aerial images involved heterogeneous properties, they achieved an overall accuracy of 97.48%.

Some recent studies obtained dense per-pixel change maps by using a Siamese Fully Convolutional Neural Network (FCN). Zhan et al. (2017) maintained the original input size in each convolutional layer in the two branches followed by a weighted contrastive loss function. The pixels with a distance measure higher than a given threshold were regarded as changed. The acquired change maps were then post-processed by a K-nearest neighbor approach. Daudt et al.

(2018a) adopted a different architecture combined with convolutional blocks and transpose convolution blocks to output full change maps between satellite images. Mou et al. (2019) proposed to learn spectral-spatial-temporal features using a Recurrent Neural Network (RNN) for change detection in multispectral images. This end-to-end architecture can adaptively learn the temporal dependence between multi-temporal images. In our work, we also propose a light-weighted SI-CNN for change detection.

Concerning deep learning for multiple 3D data sets, at present there are only scarce studies. In the field of computer vision, there are studies on the extraction of geometrical features from objects using 3D Siamese CNN. Zeng et al. (2017) propose 3DMatch model for registration between two RGB-D data sets. 3D voxels are selected from the two point clouds, and geometric features are extracted by CNN. The relationship between the two point clouds is established by minimizing the L2 loss between the feature vectors. Zhou et al. (2020) propose SiamesePointNet to extract shape descriptors from a pair of point clouds. They input raw point clouds instead of voxels. By connecting global and multi-scale local features, the extracted features show strong representation capabilities.

## 4.3 Methodology

A building is defined as *changed* in two situations: 1) it is a building in one epoch but not in the other epoch, i.e., the building is newly-built or demolished; 2) A building exists in two epochs but has changed in height or extent. In both situations, our method aims not only at delineating the boundaries of changed objects, but also at assigning a label (*heightened* or *lowered*) to each changed pixel. Our method contains two modules shown in Figure 4.2: a change detector and a change delineator. First, the input data are converted and normalized to the same range $[0,1]$ and fed into an SI-CNN for change detection. Then the boundaries of changed objects are delineated and each changed pixel is assigned a change label.

The motivation for this framework design is as follows: The change detection is performed in a light-weighted binary patch-based CNN. Although this patch-based CNN does not bring sharp changed boundaries, it localizes the changes very quickly. As shown in the center of Figure 4.2, the change map of the patch-based CNN shows only coarse change boundaries; The change delineator is then providing a fine-grained labeling for each pixel based on the coarse change map. In the following sections, we detail both the change detector (3.1) and delineator (3.2).

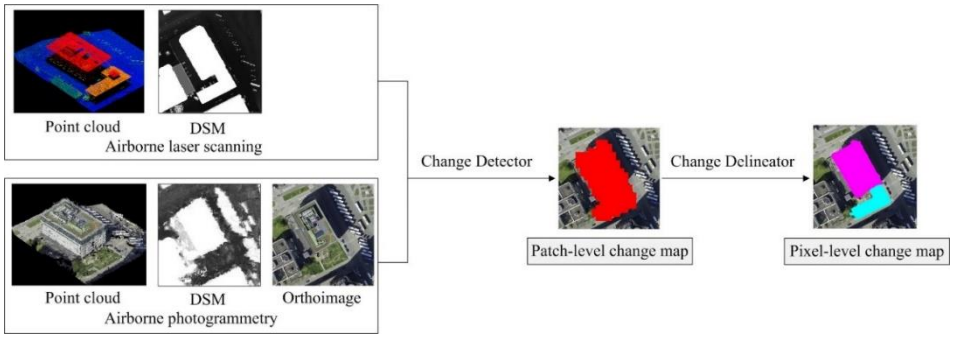



***Figure 4.2:*** *Overview of the proposed framework for change detection and delineation.*

## 4.3.1 Change detection

### 4.3.1.1 Preprocessing: registration, conversion and normalization

The goal of patch detection is to localize candidate planar patches on the ALS point cloud. The patches taken as samples should be selected on the planar ground area from the ALS data. The selection of patches should further avoid data gaps and breaklines. Planar patches of uniform size with acceptable noise level are considered valid and thus used for evaluation purpose.

First, the point clouds from two epochs and orthoimages from the second epoch are converted to the same resolution and the same coordinate system. The products from the photogrammetric workflow are geo-referenced to the world coordinate systems using Ground Control Points (GCPs) during bundle adjustment. The accuracy of the dense matching points, DSMs and orthoimages are affected by the accuracy of the interior and exterior orientation elements. The coordinates of the airborne laser points are already provided in the same world coordinate system.

Second, the laser points are converted to DSMs (ALS-DSM) using *LAStools*. The photogrammetric DSMs (DIM-DSM) and orthoimages are generated from a photogrammetric workflow. All the ALS-DSM, DIM-DSM and orthoimage are resampled to the same resolution.

Next, the heights of ALS-DSM and DIM-DSM are normalized to the same height range. The two DSMs range in $[H_{min}, H_{max}]$ where $H_{min}$ and $H_{max}$ are the minimum and maximum DSM height of the whole study area, respectively. We normalize the height values ($H_0$) of ALS-DSM and DIM-DSM using the same $H_{min}$ and $H_{max}$ as shown in Eq. (4.1). In this way, the two DSMs are converted to the range of $[0, 1]$. This representation approach maintains all the height details in DSMs.

$$H = (H_0 - H_{min})/(H_{max} - H_{min}) \quad (4\text{-}1)$$

In addition, the three channels R, G and B of the orthoimages from dense matching are also normalized to $[0,1]$ by simply dividing each pixel value by 255. Hence, all the five channels ALS-DSM, DIM-DSM, R, G and B are normalized within the $[0,1]$ range. Image patches are then cropped in the overlapping raster images for the pseudo-Siamese network.

#### *4.3.1.2 Network architecture*

The registered three patches (ALS-DSM, DIM-DSM and orthoimage) including five channels (ALS-DSM, DIM-DSM and R, G, B) are fed into the SI-CNN for change detection. The proposed SI-CNN architecture is called PSI-DC, i.e. pseudo-Siamese-DiffDSM-Color (see Figure 3). DiffDSM refers to the height difference between ALS-DSM and DIM-DSM. The input for this branch is 1 channel. For the other branch, the R, G, B channels from the orthoimage patch are provided. Our preliminary tests show that a Siamese CNN has difficulties converging when the data modalities in a given branch are heterogeneous. So we do not present the architecture with ALS-DSM as the first branch and DIM-DSM, R,G,B as the other, as a traditional Siamese network would be designed. Instead, we pass a difference DSM in the first branch and the color information from the RGB bands from the orthoimages in the other.

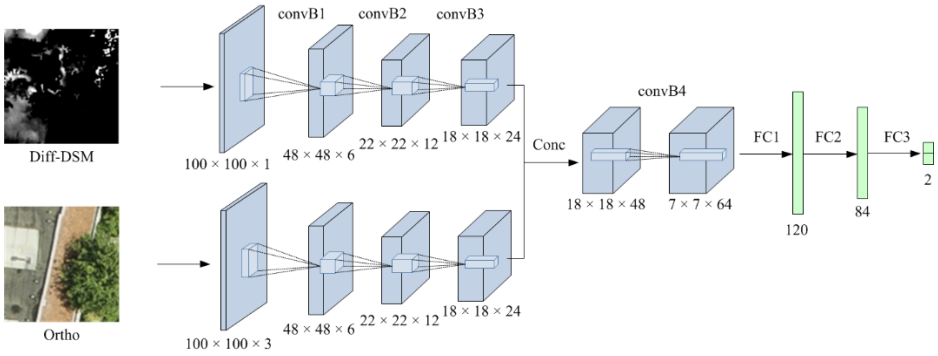



***Figure 4.3:*** *The proposed CNN architecture for multimodal change detection: PSI-DC. ConvB indicates a convolutional block; FC indicates fully connected operation. Conc indicates concatenating feature maps. The digits below each feature map are the size of (width × height × channel).*

In Figure 4.3, the inputs are processed by three convolutional blocks (convB) consecutively. The extracted feature maps are concatenated and further processed by one convB and three Fully Connected layers (FC). Each convolutional block contains a convolution operation followed by a Rectified

Linear Unit (ReLU) as activation function. convB1, convB2 and convB4 also contain a max-pooling layer which adds translation invariance to the model and decreases the number of parameters to learn (Goodfellow et al., 2016). The size of convolution kernels is 5 × 5 in our network with a padding size of 0 and slide of 1, which is verified to be an effective compromise between the feature extraction depth and contextual extent in our task.

Our network is conceptually similar to the change detection network proposed by (Mou et al., 2017), which has 8 convolutional blocks for feature extraction and 2 blocks after concatenation. In their work, the two patches to be compared are not only from different sensors (SAR and optical), but also involve translation, rotation and scale changes. Our case is simpler since our compared patches are strictly registered and normalized to the same scale. Therefore, we use fewer convolution blocks to extract features from multimodal data.

Fully connected layers are used in the final stages of the network for high-level reasoning. PSI-DC contains three FC layers. The first two FC layers are followed by a ReLU operations. The last FC layer outputs a 2×1 vector, which indicates the probability for *changed* and *unchanged*, respectively. In this way, we convert a change detection task into a binary classification task. Suppose that $(x_1, x_2)$ is the 2×1 vector predicted from the last FC layer, the loss is computed between $(x_1, x_2)$ and the ground truth (1 for *changed* and 0 for *unchanged*). First, the vector is normalized to $(0,1)$ by a Softmax function:

$$p_i = \frac{\exp(x_i)}{\exp(x_1) + \exp(x_2)}, \qquad i = 1,2 \tag{4-2}$$

where $p_1 + p_2 = 1$. Then, a weighted binary cross entropy loss is calculated:

$$Loss = -(w_1 y \log(p_1) + w_2(1-y)\log(p_2)) \tag{4-3}$$

where y is the reference label. $p_i$ is the predicted logit from the Softmax function. The ratio of $w_1$ to $w_2$ is set based on the number of negative training samples and positive samples. In urban scenes, the number of negative samples (unchanged) is usually several times the number of positive samples (changed). By assigning weights to the loss function, we provide a larger penalization to a false positive than to a false negative to suppress false positives.

### 4.3.2 Change delineation

Although the results from change detector provide preliminary change locations, they still have three limitations: First, the accuracy of change localization on the object boundary is low, i.e. determined by the patch size; Second, the zigzag change boundary cannot represent the true change boundary; Third, the attribute (*heightened* or *lowered*) we would like to obtain for each changed pixel is yet unknown. Therefore, change delineation is targeted to efficiently delineate the change boundary and assign a change label to each changed pixel. The pipeline for change delineator is shown in Figure 4.4.

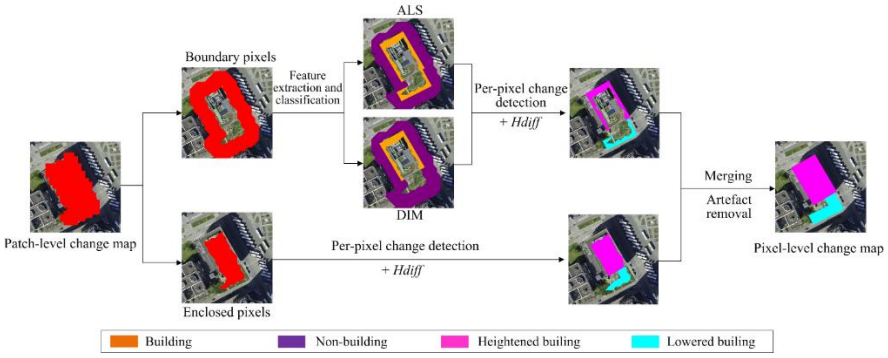



***Figure 4.4:*** *Change delineator from patch-level change map to pixel-level change map. The labeling process for boundary pixels and enclosed pixels are separated in different streams, which guarantees that the feature extraction and classification phases are only performed for a limited number of pixels.*

Each red unit in the left change map was detected by PSI-DC as a changed patch. The *boundary pixels* and *enclosed pixels* are recognized from the patch-level change map. The *enclosed pixels* are located in the center of possibly changed pixels detected from patch-based CNN, which are quite certain to be changed. Their change label is determined only based on the height difference $H_{diff}$. In contrast, the *boundary pixels* are located along the edges of possibly changed pixels from patch-based CNN, whose change label should be determined not only with $H_{diff}$, but also using class labels at both epochs. By separating the process of assigning labels to *boundary pixels* and *enclosed pixels* using different means, sophisticated feature extraction and classification are only required for the minimum number of pixels (the *boundary pixels*). This way, our proposed change delineation requires only the minimum computational effort.

#### *4.3.2.1 Identifying boundary pixels*

The patch-level change detection results are plotted on the raster grid to form a patch-level change map (see the leftmost panel of Figure 4.4). Since the patches are cropped with certain overlap, the overlapping patches are

connected into a new component. The boundary pixels are identified by subtracting an eroded component from a dilated component. The thresholds for erosion and dilation are both T_bound. T_bound should be large enough to contain the true object boundary and small enough to save computational effort. It is usually set to 1 to 2 times the patch size. The pixels enclosed by the boundary pixels are named enclosed pixels. After that, each pixel from the boundary pixels and enclosed pixels should be assigned a label heightened, lowered or non-changed. Different means of labeling are applied to the two different regions, which are described below.

#### *4.3.2.2 Feature extraction and classification for boundary pixels*

The boundary pixels are those along the edges of connected components. The classification for a boundary pixel includes two steps: First, to determine whether the pixel belongs to a changed building or not; Second, determine whether it is heightened or lowered. As mentioned before, a building change can happen in two cases: (1) It is a building in one epoch but not in the other epoch; (2) It is a building in both epochs but witnesses a height change. Therefore, the class labels in both epochs (a building class or not) and the height change between epochs are all necessary in order to make the final change decision. Note that in some special cases, a building might be re-built in the same location with almost the same height. This building is still labelled as *unchanged*.

In order to classify the boundary pixels into building or non-building in both epochs, we classify the boundary pixels in the ALS and DIM data separately. Two Random Forest (RF) (Breiman, 2001) classifiers are trained for ALS data and DIM data, respectively. We select RF for two reasons. First, our feature sets are extracted from multimodal data, RF can classify multiscale feature sets without normalization; Second, RF is less prone to overfitting through randomly selecting feature subsets and building smaller trees out of these subsets. The performance of RF is mainly affected by the hyper-parameter maximum tree depth $T_{\mathrm{td}}$ and number of trees $T_{\mathrm{tn}}$.

We train the two RF classifiers using a set of handcrafted features. Feature extraction is usually time-consuming and labor-intensive. In our change delineation framework, feature extraction is only applied to the *boundary pixels*, which accounts for only a very small portion within the complete study area. These handcrafted features are distinct for ALS data (old epoch) and DIM data (new epoch) and are listed in Table 1: 3D features from the point cloud and normalized DSM (nDSM) are extracted to distinguish building and non-building for ALS data. Both features from the point cloud and form the orthoimage are used in the classification of DIM data. The next sections detail the features used for each set.

***Table 4.1:*** *Feature sets used to classify the boundary pixels for ALS data and DIM data*

| Data | Features from point cloud (26) | Features from orthoimage (98) | nDSM (1) | Total number of features |
|---|---|---|---|---|
| ALS | √ | | √ | 27 |
| DIM | √ | √ | | 124 |

**(1) 3D features from the point cloud**

Differently from (Weinmann et al., 2015), our entities to be classified are the boundary pixels in the 2.5D space instead of the 3D point cloud. Therefore, the 3D features should be extracted for each raster pixel rather than for each 3D point. Suppose that the coordinates of a given pixel are $(X,Y)$. Spatial bins are constructed at the center of $(X,Y)$ with a size of $T_{bin}$. Gevaert et al. (2017) calculated the features for the highest point within each spatial bin. However, since the DIM point cloud is noisy, we take the 90-percentile height of all the points within the spatial bin as the *Z* value of this pixel. That is, the 3D features extracted at point $(X,Y,Z)$ are taken as the features for this pixel.

The neighborhood size affects the distinctiveness of extracted features. For example, the planarity value on a grassland might be high if calculated with a large neighborhood size and low if calculated with a small neighborhood size. We adopt the method for selecting the optimal neighborhood size proposed by (Weinmann et al., 2015). It proposes to apply an adaptive neighborhood size based on the local Shannon entropy calculated from the neighboring points. Suppose $\lambda_1, \lambda_2, \lambda_3$ $(\lambda_1 \geq \lambda_2 \geq \lambda_2)$ are the three eigenvalues of the covariance matrix calculated from the $k$ nearest neighboring (KNN) points. $e_1, e_2, e_3$ are the normalized eigenvalues $e_i = \frac{\lambda_i}{\Sigma_\lambda}$ with $i \in \{1,2,3\}$. The Shannon entropy is calculated via

$$E_\lambda = -e_1 \ln(e_1) - e_2 \ln(e_2) - e_3 \ln(e_3) \tag{4-4}$$

A series of Shannon entropies are calculated by iteratively increasing the neighborhood size from $[k_{min}, k_{max}]$. Since the Shannon entropy measures the disorder of the neighborhood, the $k$ value bringing the minimum entropy is regarded as the optimal neighborhood size. The feature sets calculated with the optimal neighborhood size are applied to represent the local shape and geometry. The following features are extracted:

**- Local 3D shape features (16)**. Given the optimal neighborhood size, local 3D shape features can be calculated by combining the normalized eigenvalues

describing linearity $L_\lambda$, planarity $P_\lambda$, scattering $S_\lambda$, omnivariance $O_\lambda$, anisotropy $A_\lambda$, eigenentropy $E_\lambda$, sum of eigenvalues $\Sigma_\lambda$ and change of curvature $C_\lambda$:

$$L_\lambda = \frac{e_1 - e_2}{e_1} \quad (4\text{-}5)$$

$$P_\lambda = \frac{e_2 - e_3}{e_1} \quad (4\text{-}6)$$

$$S_\lambda = \frac{e_3}{e_1} \quad (4\text{-}7)$$

$$O_\lambda = \sqrt[3]{e_1 e_2 e_3} \quad (4\text{-}8)$$

$$A_\lambda = \frac{e_1 - e_3}{e_1} \quad (4\text{-}9)$$

$$E_\lambda = -\sum_{i=1}^{3} e_i \ln(e_i) \quad (4\text{-}10)$$

$$\Sigma_\lambda = \lambda_1 + \lambda_2 + \lambda_3 \quad (4\text{-}11)$$

$$C_\lambda = e_3/(e_1 + e_2 + e_3) \quad (4\text{-}12)$$

In addition, the local point density $D$, radius of $k$ nearest points, height range, standard deviation of height and verticality are calculated from the neighboring 3D points. In total, 16 local shape features are extracted including $e_1$, $e_2$ and $e_3$.

**- Local 2D shape features (6)**. Most *boundary pixels* lie along building walls. Local 2D shape features are calculated after the neighboring 3D points are projected to a horizontal plane. The projected points on the 2D plane along a vertical structure (e.g. walls, poles, traffic lights) are usually denser than points projected from a horizontal plane. Therefore, 2D shape features are supposed to be distinctive features in distinguishing points close to a wall. In total, the radius of neighboring points, point density, sum of 2D eigenvalues, ratio of two eigenvalues and two normalized eigenvalues are extracted as 2D shape features.

**- Spatial binning features (4).** The spatial binning features are calculated from all the 3D points within this bin. The 90-percentile height value $Hp90$ indicating the object height is taken as one feature. Additionally, the number of points within this bin, the standard deviation of the heights and height range $\Delta Hp$ are calculated as features. $\Delta Hp$ is calculated with:

$$\Delta Hp = Hp90 - Hp10 \quad (4\text{-}13)$$

where $Hp10$ is the 10 percentile height of all the points within this bin.

**(2) 2D features from the orthoimage**

**- Radiometric features (16)**. First, 2D radiometric features are extracted from the orthoimages. The R, G, B values normalized by 255, and the R, G, B values normalized by their sum are calculated for each pixel. Two high-level radiometric features are also calculated: Normalized Excessive Green Index (*nEGI*) (Qin, 2014) and shadow index $\psi$ (Sirmacek and Unsalan, 2009). The vegetation index *nEGI* and shadow index $\psi$ are calculated so that pixels in these areas are more likely to be classified into *non-building*.

$$nEGI = (2G - R - B)/(2G + R + B) \quad (4\text{-}14)$$

$$\psi = 4/\pi \cdot \arctan\left((B - G)/(B + G)\right) \quad (4\text{-}15)$$

Gevaert et al. (2017) found that the radiometric features averaged over an image segment are also distinctive features since they are insensitive to radiometric noise. Simple Linear Iterative Clustering (SLIC) adapts a K-means clustering method to group pixels into perceptually meaningful regions in a five-dimensional feature space, which is defined by the CIELAB color space and the x, y pixel coordinates (Achanta et al., 2012). We use SLIC segmentation in this paper for two reasons: (1) It adheres well to image boundaries; (2) It generates approximately equally sized superpixels and the targeted number of superpixels can be controlled. Finally, the eight pixel-based radiometric features mentioned above are averaged over each SLIC segment to obtain eight further segment-based features.

**- Textural features (82).** Textural features are supposed to be distinctive when distinguishing buildings from other land cover types. For example, a roof covered with tiles often presents strip-like texture. In contrast, vegetation shows directionless texture; Terrain made from concrete and shaded areas both show little texture. Local Binary Pattern (LBP) features are extracted to encode local texture information (Ojala et al., 2002). LBP is robust to grayscale variations and rotations. The LBP features are calculated from a circularly symmetric neighbor set of *P* members within radius of *R* from the central pixel, denoting the operator as $LBP_{P,R}^{riu2}$. *P* determines the quantization of the angular space, whereas *R* determines the spatial resolution of the operator. *riu2* indicates that the Rotationally-Invariant Uniform features, which contain $(P + 2)$ dimensions. In order to enhance its scale-invariance, $LBP_{P,R}^{riu2}$ features are extracted at different radius, for example $R \in \{1, 2, 3\}$, from a window size of 5×5 or 11×11.

**(3) The nDSM feature**

Another distinctive feature used in classifying ALS data is the Normalized DSM (nDSM). The height from the object surface to the nearby ground is an effective indicator to the object height (Vosselman et al., 2004). nDSM is computed by subtracting the Digital Terrain Model (DTM) from the 90-percentile height $Hp90$.

Considering that DTM extracted from noisy dense matching points is less reliable (Zhang et al., 2018b), the nDSM feature is not considered for DIM data.

#### 4.3.2.3 Per-pixel change detection

Each pixel within *boundary pixels* and *enclosed pixels* is classified into *heightened*, *lowered* or *other*. This process is named *per-pixel change detection*. Before the *per-pixel change detection*, height difference $H_{diff}$ between the ALS data and DIM data is computed using the 90-percentile height ($Hp90$) so as to minimize the impact of point cloud noise. According to the definition of building changes in Section 3.2.2, a building change must involve change of its height. Therefore, $H_{diff}$ is computed as another change indicator, for both *boundary pixels* and *enclosed pixels* to distinguish *heightened* and *lowered* pixels:

$$H_{diff} = Hp90(DIM) - Hp90(ALS) \tag{4-16}$$

Then *per-pixel change detection* is performed in *boundary pixels* and *enclosed pixels* separately (see Figure 4):

**Boundary pixels:** For each *boundary pixel*, the label of the old epoch on the ALS data, the label of the new epoch on the DIM data and the $H_{diff}$ are provided to perform per-pixel change detection. First, a pixel in *boundary pixels* is classified into *building-related pixel* if the label in one or both epochs is *building*. Then the *building-related pixel* is further classified into *heightened*, *lowered* or *other* via

$$\begin{cases} if\ H_{diff} \in (T_{Hd}, +\infty), & Hightened \\ if\ H_{diff} \in (-\infty, -T_{Hd}), & Lowered \\ if\ H_{diff} \in [-T_{Hd}, T_{Hd}], & Other \end{cases} \tag{4-17}$$

$T_{Hd}$ is set according to the minimum height change of a building we aim to detect.

**Enclosed pixels:** Since the *enclosed pixels* are located in the middle of patch-based change masks, they are very likely to be changed buildings. Only $H_{diff}$ is considered to make per-pixel classification via Eq. (4.17). Then, each enclosed *pixel* is classified into *heightened*, *lowered* or *other*.

#### 4.3.2.4 Artefact removal

As shown on the right of Figure 4.4, the *per-pixel change detection* results for *boundary pixels* and *enclosed pixels* are merged. At this stage, sharp

boundaries of changed buildings are obtained. However, due to the DIM errors and data gaps, errors from the change detection and per-pixel change detection are still propagated and accumulated in our change map. The errors in the change map can show small artefacts such as isolated clusters due to change detection errors, elongated artifacts produced by the dense matching along corners of walls or holes due to low point cloud density. To cope with those artefacts, we use mathematical morphology-based post processing. Morphological operations are simple yet effective in filling holes or removing artefacts in binary images (Haralick and Shapiro, 1992). A combination of morphological operations is applied to refine the change map. First, morphological closing is applied to fill holes, i.e. to enhance the signal intensity of real changed objects (True Positives). Then morphological opening is applied to eliminate small or elongated artefacts (False Positives). The workflow for removing artefacts is as follows:

1) Separate the heightened and lowered change masks and process them separately following steps 2-3-4 below.

2) Process the one-fold change mask using morphological closing with a threshold $T_{close}$.

3) Process the change mask using morphological opening with a threshold $T_{open}$.

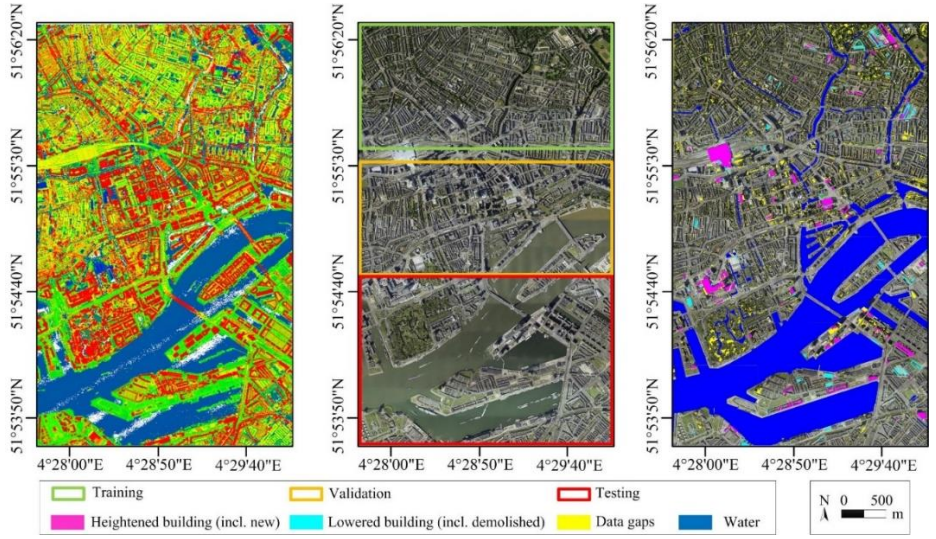
4) Connected component analysis. Connect the neighboring pixels in their 8-neighborhood to form a complete changed object. Remove those objects whose length is smaller than $T_{length}$. $T_{length}$ is determined by the minimum size of the changed buildings we aim to detect.

5) Merge the heightened and lowered change masks to form the final change map.

The effect of our artefact removal method will be presented in section 4.4.2.

## 4.4 Results and discussion

The study area is located in Rotterdam, The Netherlands, which is a densely built port city mainly covered by residential buildings, skyscrapers, vegetation, roads, and waters. The study area is 14.5 km$^2$ as shown in Figure 4.5. Figure 4.5(a) shows the ALS point cloud obtained in 2007 with a density of approximately 25 points/m$^2$. The point cloud contains approximately 226 million points.




(a) (b) (c)

***Figure 4.5:*** *Visualization of the data set for change detection. From left to right: (a) ALS points colored according to height; (b) orthoimage marked with the training area, validation area, and testing area; (c) orthoimage overlaid with reference labels.*

A total of 2160 aerial images were obtained by CycloMedia from five perspectives in 2016. The GSD of nadir images equaled 0.1 m. The bundle adjustment and dense image matching were performed in Pix4Dmapper. The vertical RMSE (Root Mean Square Error) of 48 GCPs was ±0.021 m and the vertical RMSE of 20 check points was ±0.058 m. The DIM point cloud contains approximately 281 million points. DSMs and orthoimages were also generated at the same resolution of 0.1 m. Figure 4.5(b) shows the generated orthoimage. The training, validation, and testing area make up 28%, 25%, and 42% of the study area, respectively. Note that 5% of the block (between training and validation area) is not used since this area contains the newly-built Rotterdam railway station with homogeneous building change; the samples extracted from this area will reduce the sample diversity and may lead to under-fitted CNN models. Figure 4.5(c) shows the orthoimage overlaid with four types of labels: heightened building, lowered building, water, and data gaps.

### *4.4.1 Patch-level results*

Evaluation of change detection is important for employing these results for decision-making since the evaluation can tell you the reliability of change detection results (Lu et al., 2003). Reference data for change detection evaluation usually come from manual interpretation or more accurate geographic database. The methods include patch-to-patch, pixel-to-pixel and

object-to-object evaluation to make a comprehensive evaluation. Confusion matrix is the most widely used tools for quantitative analysis in change detection.

***Table 4.2:*** *Confusion matrix for evaluating change detection*

| Detected \ GT | Changed | Unchanged |
|---|---|---|
| Changed | TP | FP |
| Unchanged | FN | TN |

In Equation (4-3) and Table 4.2, True Positive (TP) is the number of changes detected by the algorithm which is real changes. True Negative (TN) is the number of unchanged entities detected as unchanged. False Positive (FP) is the number of changes detected by the algorithm which are not changes in real scene. FP is equal to "False alarm". False Negative (FN) is the number of unchanged entities detected by the algorithm which are real changes. FN corresponds to undetected changes.

Three quality measures are computed for the evaluation: recall, precision and $F_1$-score. The recall is the percentage of the actual changes that are detected by an algorithm, and the precision is the percentage of the changes detected by an algorithm that correspond to real changes (Rottensteiner, 2007). $F_1$-score is the harmony mean of recall and precision.

$$Recall = \frac{TP}{TP + FN} \tag{4-18}$$

$$Precision = \frac{TP}{TP + FP} \tag{4-19}$$

$$F1 = \frac{2 * precision * recall}{Precision + recall} \tag{4-20}$$

During training, the CNN model is evaluated on the validation set after every three epochs to check its performance and ensure that there is no overfitting. Towards the end of training, the model with the highest $F_1$-score is selected as the final trained model. The validation results of the proposed PSI-DC are as follows: TP is 2,362; TN is 101,636; FP is 2,475; FN is 563. Recall equals 80.75%; Precision equals 48.83%; $F_1$-score equals 60.86%. That is, 80.75% positive samples are correctly inferred as positive; 97.62% negative samples are correctly inferred as negative. The patch-level change detection results on the test set from the three CNN architectures are shown in Table 4.3.

***Table 4.3:*** *Patch-level testing results (%) for three CNN architectures. The highest score in each column is shown in bold.*

| Network | Recall | Precision | $F_1$-score |
|---|---|---|---|
| PSI-DC | **86.17** | **68.16** | **76.13** |
| PSI-HHC | 84.63 | 61.03 | 70.92 |
| FF-HHC | 82.17 | 67.17 | 73.92 |

Table 4.3 shows that the proposed PSI-DC model outperforms the other two models in all three metrics. The recall of PSI-HHC is higher than FF-HHC by 2.46%, but its precision is lower than the latter by 6.14%, which results into the lowest $F_1$-score among the three. In addition, PSI-DC outperforms PSI-HHC by 5.21% in $F_1$-score. This large margin indicates that the input configuration to the CNN models has a significant impact on the change detection results. The PSI-DC and PSI-HHC networks are all the same except that one branch in PSI-DC is Diff-DSM while the same branch is replaced by raw ALS-DSM and DIM-DSM patches in PSI-HHC. This can be explained by that PSI-DC takes advanced features (height difference of two DSMs) as input, while PSI-HHC takes two raw DSMs as input. PSI-HHC has more parameters and requires the CNN model to learn deeper.

Comparing PSI-DC and FF-HHC, the Siamese architecture performs better than feed-forward architecture. This can be explained by the fact that our inputs (DSM and Orthoimage) have quite different modalities. The Siamese CNN allows processing the two inputs in different branches and then fuse their features. The feed-forward architecture takes all the inputs of different modalities as input, which might be harder for the CNN to learn.

The change maps generated from PSI-DC and FF-HHC are visualized in Figure 4.6. Most changed objects are correctly detected in the patch-level results even though some false detections occur. Although the patch-level change masks show zigzag effect, the results still reflect coarse locations of the change boundaries. The six examples in the lower part of Figure 4.6 visualize some details of the change maps. Figure 4.6(a) shows a demolished factory. The patch-based change map from PSI-DC can represent the boundary much better than FF-HHC since many FNs (i.e. omission errors) appear in the latter. Figure 4.6(b) shows that a deep pit in a construction site is misclassified into a changed building by both models.

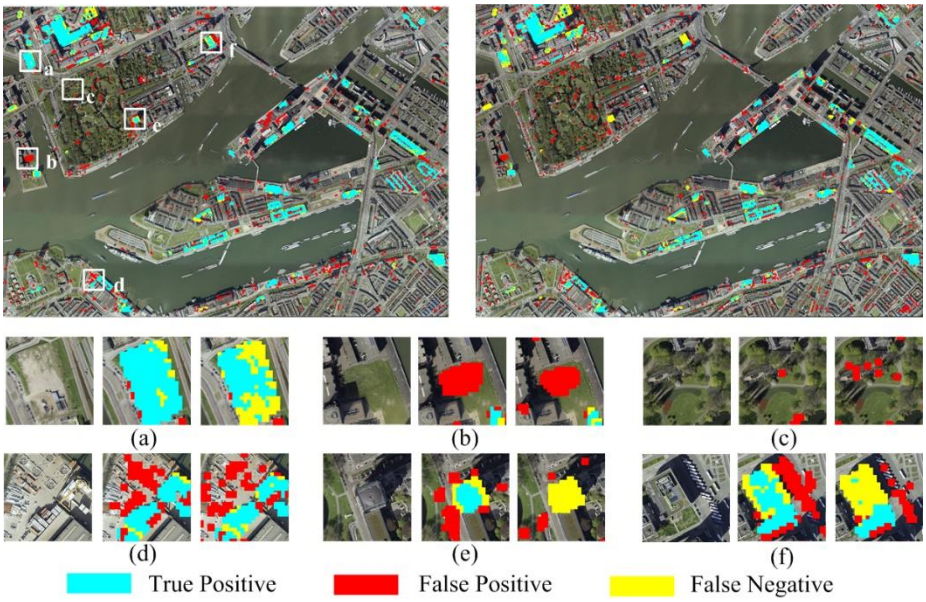



***Figure 4.6:*** *Patch-based change maps generated from the model PSI-DC (top left) and FF-HHC (top right). The two rows below show six zoom-in examples from the change maps. In each example from the left to the right: orthoimage for reference, change map from PSI-DC, and change map from FF-HHC.*

Figure 4.6(c) shows that more FPs appear in the vegetation area from the model FF-HHC than in the prediction of PSI-DC. This area is located in a park covered with grassland, trees, rivers and dirt roads. The point cloud from dense matching contains much noise in this area. PSI-DC can better classify this area into non-changes than FF-HHC. Figure 4.6(d) shows another construction site at the port. Tall tower cranes, containers and trucks are misclassified into building-related changes because their heights and surface attributes are similar to a real building. Figure 4.6(e) shows that a heightened square building is omitted by FF-HHC while it is detected by PSI-DC. Figure 4.6(f) shows that FF-HHC makes more FNs than PSI-DC, even though FF-HHC causes less FPs than PSI-DC.

### *4.4.2 Pixel-level results*

Before presenting the pixel-level results, we analyze the effect of artefact removal with morphological operations and connected component analysis (Section 4.3.2.4). Figures 4.7(a) to (e) show that some small lowered change masks are mixed inside the heightened change mask. When the heightened and lowered masks are processed separately, the small lowered artefacts are filtered out. Morphological closing fills in the holes and smoothens the object boundaries in Figures. 4.7(b-d). Note that the holes in the change masks are mainly data gaps rather than omission errors. In addition, elongated artefacts

along the building corners are removed by morphological opening in Figures 4.7(a-d). Figure 4.7(b) shows two lowered building changes. Figure 4.7(d) shows that small noisy artefacts are filtered out, while a true building change is kept because of its strong response. Figure 4.7(e) shows that some buses are misclassified as building-related changes in the initial change map. They are successfully filtered out in artefact removal. However, Figure 4.7(b) also shows some deficiencies of our method. Two separate lowered buildings are merged because they are very close to each other.

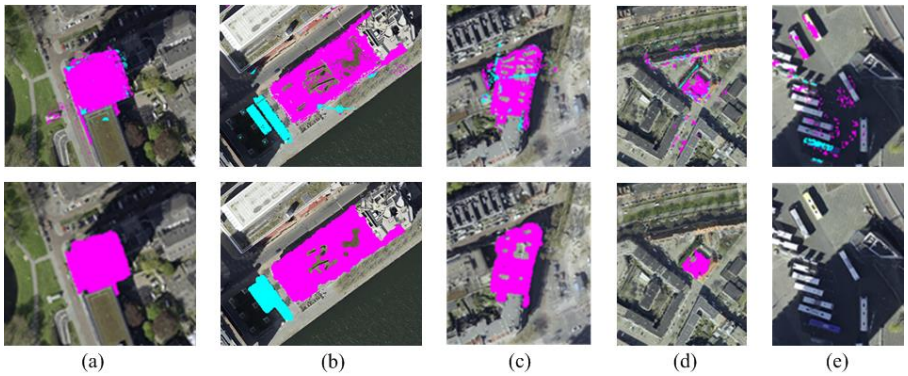
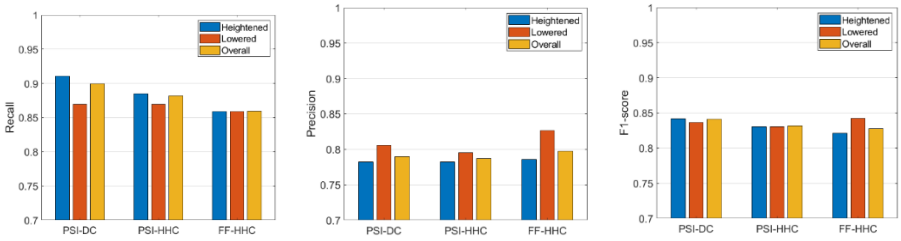



***Figure 4.7:*** *Examples for artefact removal. The top row shows the initial change maps. The bottom row shows the refined change maps. Magenta indicates heightened building; Cyan indicates lowered building.*

After artefact removal, pixel-level change maps are obtained. Some quantitative metrics are shown in Table 4.4 and depicted in Figure 4.8.

***Table 4.4:*** *Pixel-level testing results (%) for the three CNN architectures. The highest score for the overall metrics is shown in bold in each column.*

| Network | Type | Recall | Precision | $F_1$-score |
|---|---|---|---|---|
| PSI-DC | Heightened | 91.07 | 78.25 | 84.17 |
| | Lowered | 86.96 | 80.57 | 83.64 |
| | Overall | **89.97** | 78.98 | **84.12** |
| PSI-HHC | Heightened | 88.49 | 78.22 | 83.04 |
| | Lowered | 86.97 | 79.53 | 83.08 |
| | Overall | 88.16 | 78.70 | 83.16 |
| FF-HHC | Heightened | 85.91 | 78.59 | 82.09 |
| | Lowered | 85.91 | 82.64 | 84.24 |
| | Overall | 85.96 | **79.78** | 82.76 |




(a) Recall (b) Precision (c) $F_1$-score

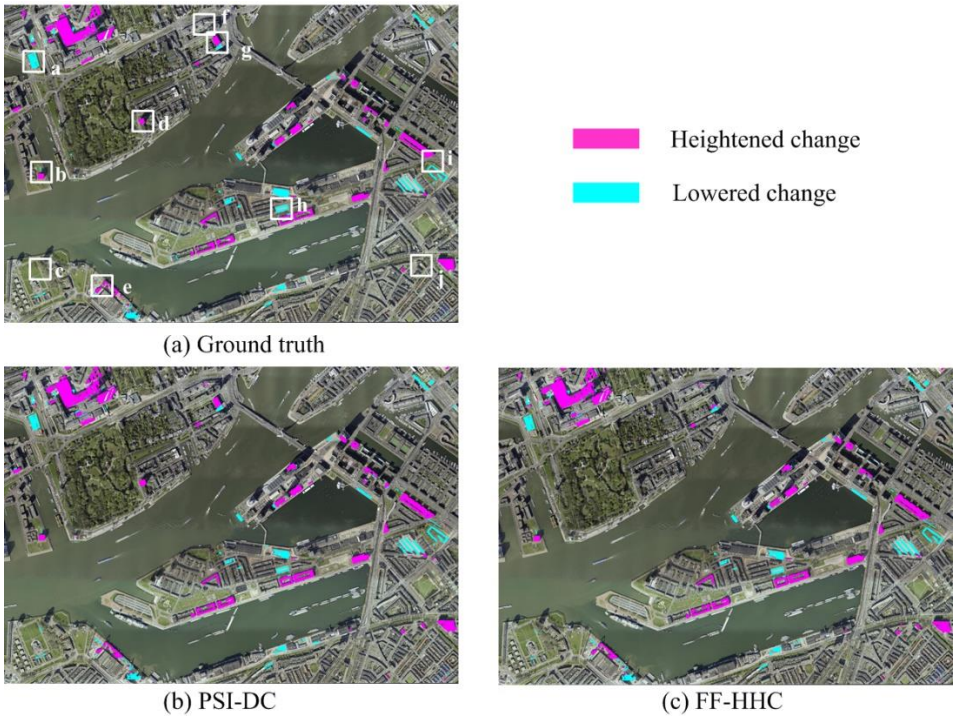
***Figure 4.8:*** *Pixel-level testing results for the three architectures. The bars in three different colors represent the metrics for heightened, lowered and overall, respectively.*

Some findings can be drawn:

1) Comparing the overall metrics for the three models, PSI-DC achieves the highest $F_1$-score and recall, while the second highest precision, only 0.8% lower than PSI-HHC. As stated above in the patch-level results, the Siamese architecture is a sound architecture for processing multimodal inputs. Using Diff-DSM in one Siamese CNN channel is better than two raw DSMs, since the latter model is more difficult to train. In the patch-level results (see Table 4.3), the highest precision was also achieved by PSI-DC. This small inconsistence between pixel-level precision and patch-level precision might be explained by the fact that PSI-DC has a higher recall in the patch-level results, which brings more suspicious pixels to the change map, which in turn slightly reduces the precision rate in the pixel-level result.

2) Comparing *heightened* and *lowered* metrics, the recalls of heightened buildings from the two Siamese models are higher than the recall of lowered buildings, while in FF-HHC model they are the same. In addition, the precisions of heightened buildings from all three models are lower than the precisions of lowered buildings. A sound explanation is that when delineating heightened changes in the pixel-level, edges are determined by the relatively noisy DIM point cloud. When delineating lowered changes, the edges are determined by the precise ALS data. For the heightened changes, our change delineator tends to classify many suspicious pixels into changed building, which leads to high recall. For the lowered changes, the change delineator can recognize the sharp edges from ALS data, which results into high precision.

The final change maps are shown in Figure 4.9. Generally, the change masks can represent the change boundaries well, even though the sizes and shapes of changed buildings vary a lot. In addition, the labels (*heightened* or *lowered*) assigned to the changed buildings are homogeneous.




(a) Ground truth

(b) PSI-DC

(c) FF-HHC

***Figure 4.9:*** *Pixel-level change maps. Ten examples in the white squares are visualized in Figure 4.12.*

Ten examples for the detected changes selected from Figure 4.9 are visualized in Figure 4.10. Figure 4.10(a) shows the change map for the demolished factory mentioned above. Two slopes can be seen from the ALS point cloud. The slopes are classified into *non-building* so they are correctly classified as *non-change* in the final change maps. Figure 4.10(b) shows a newly-built building and three small demolished buildings. The detected small buildings are merged to one big building object. Figure 4.10(c) shows a removed elongated mound of approximately 2.5 m. It is misclassified into a demolished building in both the PSI-DC and FF-HHC models. The reason might be that the height change and surface attributes are similar to a building change.

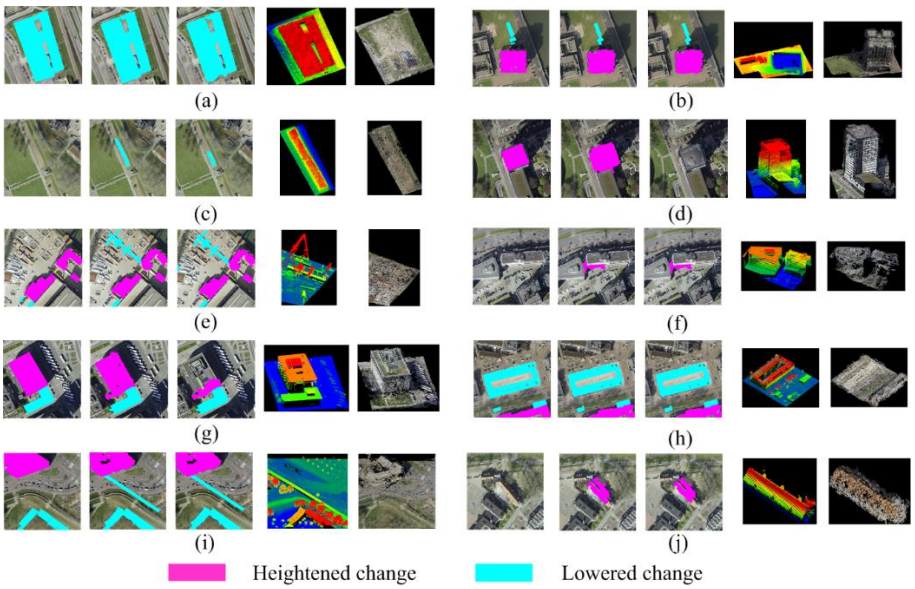



***Figure 4.10:*** *Ten examples for the detected changes and the corresponding point clouds. From the left to the right: ground truth, change map from PSI-DC, change map from FF-HHC, ALS point cloud colored by height, DIM point cloud with true color. The point clouds are visualized from the bird-view.*

Figure 4.10(d) shows that a new building change is detected in PSI-DC but is omitted in FF-HHC. Note that, when omitted in the patch-level results by FF-HHC (Figure 4.6(e)), the omission error cannot be recovered by the change delineator. Figure 4.10(e) shows the same construction site mentioned in Figure 4.6(d). Miscellaneous construction work is going on in the ALS data while the construction equipment is removed when the DIM data was captured. Also, a tower crane is misclassified into a lowered building. Figure 4.10(f) shows that a FP appears in the change maps of PSI-DC and FF-HHC. It is located at the corner of a wall, which is misclassified as a *changed patch* by both SI-CNN and FF-HHC. Then it is also misclassified into the *building* class by the RF during the per-pixel classification.

Figure 4.10(g) shows a heightened building and a lowered building. The change masks from PSI-DC are more similar to the ground truth than FF-HHC. Figure 4.10(h) shows three lowered buildings and one heightened building. The courtyard in one of the lowered buildings is correctly delineated. Figure 4.10(i) shows that a long slope, which connects the ground and a roof, is misclassified into a demolished building. The attributes of a slope change are very close to a building roof change. Figure 4.10(j) shows that an unchanged building is misclassified into a *heightened building*. The noisy DIM point cloud of this building is higher than true height possibly for two reasons: First, it is located at the border of the photogrammetric block. A low number of images and poor

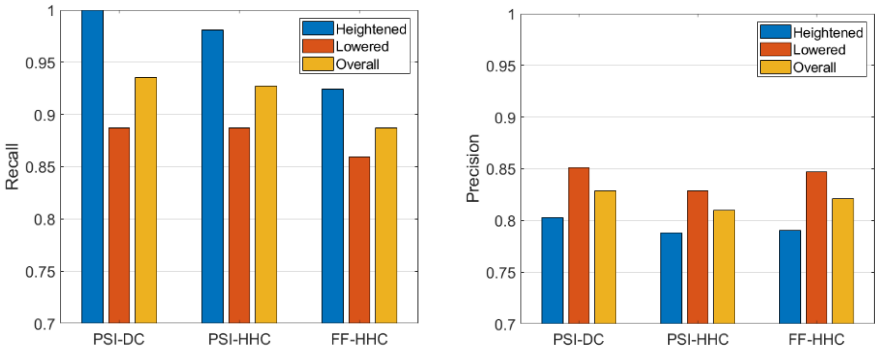
imaging geometry usually result into poor point accuracy (Zhang et al., 2018a). Second, the roof is brown with little texture. Dense matching is problematic if the correspondence among pixels is weak.

### 4.4.3 Object-level results

The object-level results are shown in Table 4.5 and depicted in Figure 4.11. Among the three models, PSI-DC achieves the highest recall and precision: 140 buildings are detected as *changed* by PSI-DC, 116 of which are true changes; 8 building changes are missed. The *overall* recall among the three models is consistent with the pixel-level recalls among the three. The recall of FF-HHC is lower than PSI-HHC by approximately 4% while its precision is higher than the latter by 1.1%.

***Table 4.5:*** *Object-level testing results (%) for the three CNN architectures. The highest score for the overall metrics is shown in bold in the last two columns.*

| Network | Type | Reference Number (RN) | Detected Number (DN) | True Detected Number (TDN) | Recall | Precision |
|---|---|---|---|---|---|---|
| PSI-DC | Heightened | 53 | 66 | 53 | 100.00 | 80.30 |
| | Lowered | 71 | 74 | 63 | 88.73 | 85.14 |
| | Overall | 124 | 140 | 116 | **93.55** | **82.86** |
| PSI-HHC | Heightened | 53 | 66 | 52 | 98.11 | 78.79 |
| | Lowered | 71 | 76 | 63 | 88.73 | 82.89 |
| | Overall | 124 | 142 | 115 | 92.74 | 80.99 |
| FF-HHC | Heightened | 53 | 62 | 49 | 92.45 | 79.03 |
| | Lowered | 71 | 72 | 61 | 85.92 | 84.72 |
| | Overall | 124 | 134 | 110 | 88.71 | 82.09 |




(a) Recall (b) Precision

***Figure 4.11:*** *Object-level testing results for the three models. The bars in three different colors represent the metrics for heightened, lowered and overall, respectively.*

Note that the recall of patch-level results from PSI-DC reaches as high as 86.17%. It shows that our three CNN models are *thorough* to keep almost every possible change and verify it later in the change delineator. Therefore, a high recall can be achieved in the object-level results. Considering the heightened and lowered changes, all the 53 heightened buildings are detected by PSI-DC, while 63 of 71 lowered buildings are detected. Meanwhile, 13 heightened FPs and 8 lowered FNs decrease the heightened precision (80.30%) and lowered precision (85.14%). Examples for FPs and FNs can be found in Figure 4.10(f)(j) and Figure 4.10(c)(e)(i), respectively.

The metrics of the final change maps depend on the synergy of both change detection and change delineation. The errors from the change detection can be propagated into the change delineation. If the CNN model achieves high patch-level precision but low recall, omission errors are more likely to occur in the pixel-level and object-level results. Our change delineator can only delineate the detected changes but cannot retrieve the omitted ones. Additionally, if the CNN model achieves a high patch-level recall but low precision, then the final results depend on the performance of the two RF classifiers. However, RF may also cause make mis-classifications. As shown in Figure 4.10(f), RF misclassifies *shaded terrain* into *building* in the DIM data, which results into FP errors. Meanwhile, the computational effort for change delineation also increases, since more pixels are taken as *boundary pixels* and *enclosed pixels*.

### *4.4.4 Visualization of feature maps*

In order to understand what the CNN learns, the feature maps from the last convB in PSI-DC are visualized in Figure 4.12. The outputs of PSI-DC are 64 7×7 feature maps. Only the strongest half feature maps are shown. Two samples are visualized including a new building and a demolished building. The activation is dispersed in multiple feature maps. In some of the feature maps, the direction and shape of the activation reflect the shape of the change, e.g. the 25th feature map in Figure 4.12(a) and the {8th, 9th, 16th, 32nd} feature maps in Figure 4.12(b), which are highlighted in red frames. The activations from the last convB are then flattened and summed up in the fully connected layers, which matches with our definition of changed patches. Namely, whether a patch is changed or not depends on the ratio of changed pixels, rather than merely the central pixel.

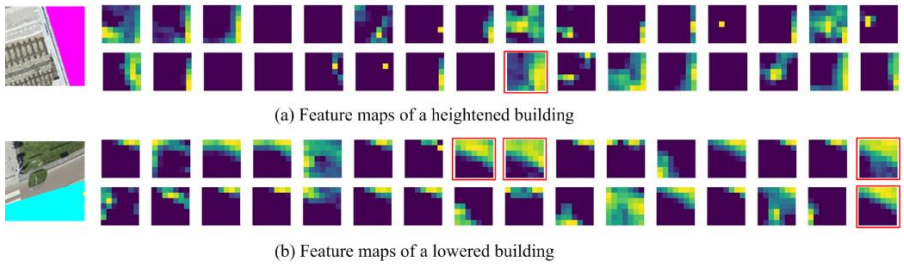



***Figure 4.12:*** *Visualization of the feature maps from the last convolutional block in PSI-DC. 32 strongest feature maps are shown.*

### 4.4.5 Sensitivity analysis

*Patch size* is a critical hyper-parameter in our framework. It should be large enough to incorporate much contextual information for patch-based change detection. It should be small enough to guarantee a relatively detailed patch-level boundary and reduce the dependency on the change delineator. We make comparative studies by selecting samples with size of 80×80 and 60×60 from the converted DSMs and orthoimages, and then run the whole workflow from scratch. To make the comparison meaningful, we maintain the original PSI-DC architecture but up-sample the three patches (ALS-DSM, DIM-DSM, and orthoimage) to 100×100 to fit the CNN inputs. In addition, the numbers of positive and negative training samples in 80×80 and 60×60 tests are all the same with those in the 100×100 test. When extracting samples from the testing area, the number of samples for 80×80 and 60×60 tests is 1.6 times and 2.9 times of the number in 100×100 test, respectively.

Figure 4.13(a) shows the impact of patch size on the patch-level change maps. Both the precision and $F_1$-score show a clear decreasing trend when the patch size decreases, even though the recall shows some fluctuation. This can be explained by the fact that the precision decreases when less contextual information is contained in the smaller patches. Figures 4.13(b) and (c) show that the pixel-level and object-level metrics are relatively robust to the variation of patch size. When the patch size decreases, the PSI-DC with different patch size always outputs a relatively high recall (Figure 4.13(a)). Then the candidate pixels are all propagated to the change delineator for the final decision. Since these changed patches always contain most true changed pixels and objects, the pixel-level and object-level results are only slightly affected if the change delineator works fine.

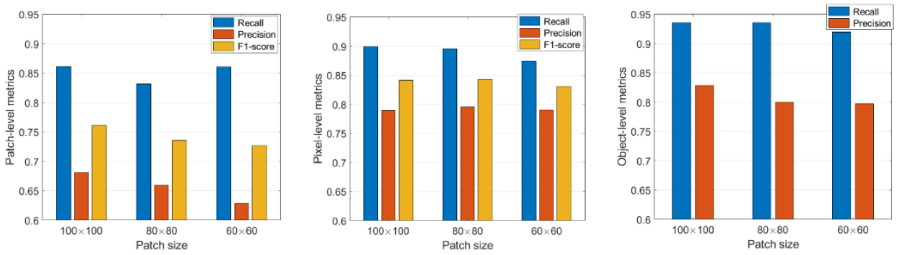



***Figure 4.13:*** *Impact of the patch size on the generated change map: (a) patch-level; (b) pixel-level; (c) object-level.*

In addition, the size of structuring elements used in Section 4.3.2.4 also affects the final change maps. We implement a pixel-level evaluation of the change maps when $T_{close}$ and $T_{open}$ are selected via a grid search in the rage of [0, 30] with an interval of 5. The impact of $(T_{close}, T_{open})$ on recall, precision and $F_1$-score is depicted as relatively smooth meshes in Figure 4.14. Figure 4.14(a) shows that recall reaches its maxima when $(T_{close}, T_{open})$ takes $(30,0)$. With a large $T_{close}$, many FNs appear so recall increases. Figure 4.14(b) shows that the precision comes to the maxima when $(T_{close}, T_{open})$ takes $(0,30)$. At this point, the morphological opening filters out all the suspicious pixels and only maintains those with the strongest response, thus leading to a high precision. Figure 4.14(c) shows that when $(T_{close}, T_{open})$ takes $(10,30)$, $F_1$-score reaches the maxima 85.06%. Therefore, we adopted the $(10,15)$ in this paper, which brought slightly lower $F_1$-score of 84.12%.

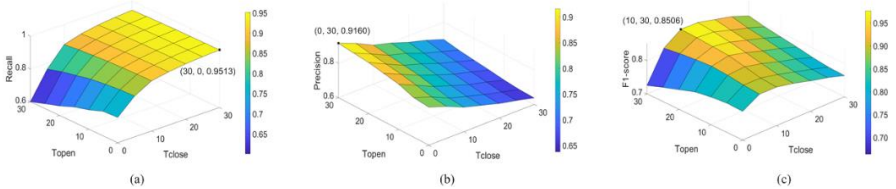



***Figure 4.14:*** *Impact of the size of morphological structuring elements on the final change map. (a) Recall; (b) Precision; (c) F1-score. The black dot on each mesh show the point* $(T_{close}, T_{open}, value)$ *where each mesh takes its maximum.*

## 4.5 Conclusions

We propose a method to detect building changes between airborne laser scanning and photogrammetric data. This task is challenging owing to the multi-modality of input data and dense matching errors. First the multimodal data are converted to the same scales and fed into a light-weighted pseudo-

Siamese CNN for change detection. Then, the changed objects are delineated through per-pixel classification and artefact removal. In the pixel-level evaluation, our change map achieves a recall rate of 89.97%, a precision rate of 78.98%, and an $F_1$-score of 84.12%. For the object-level evaluation, the recall rate reaches 93.55% and the precision rate reaches 82.86%. Although the point cloud quality from dense matching is not as good as laser scanning points, the radiometric and textural information provided by the orthoimages serves as a supplement, which leads to relatively satisfactory change delineation results.

There are two advantages with the design of our framework: First, the complicated multimodal change detection problem is disassembled into three binary classification problems. They are solved by one CNN model and two RF classifiers, which require less hyper-parameters and prior knowledge compared to (Du et al., 2016). The PSI-DC model is light-weighted but works satisfactorily for the problem at hand. Second, the change detection module and change delineator module are separated in the framework. The change detection module based on a pseudo-Siamese CNN can quickly provide some initial change maps in emergency response. The change delineator divides the candidate changed pixels into *boundary pixels* and *enclosed pixels*: Only a few *boundary pixels* require the more complex feature extraction and classification, which largely reduces the computational load. Concerning the disadvantages of the proposed method, the framework is relatively complicated including two steps instead of an end-to-end solution. The next chapter is targeted to solve the problem in a more straight-forward manner.

# Chapter 5 - Combined Semantic Segmentation and Change Detection Between Multimodal Point Clouds

## 5.1 Introduction

Our target is to detect changes between outdated ALS point clouds and new DIM point clouds. Compared with patch-based change detection in chapter 4, this chapter aims to detect changes to each 3D ALS point instead of changes to the 2D square patches.

Although semantic segmentation (SS) and change detection (CD) are always investigated separately in the previous literature, we take them as two strongly correlated tasks and propose to solve them in a single workflow. Change detection extracts change information from the old epoch to the new epoch. In order to identify the "from-to" change types, "semantic segmentation" is required to each epoch. Namely, in the "from-to" change detection task, change detection is finished only when semantic segmentation has been implemented in both epochs.

In this chapter, we propose a solution to define the categories in the outputs which combines the change labels and semantic labels. Figure 5.1 shows a schematic diagram indicating changes between two epochs. Table 5.1 shows our solution to define joint categories for Figure 5.1. Tran et al. (2018) also combined the tasks of semantic segmentation and change detection between LiDAR points from two epochs. They considered more change categories than we did in Table 5.1. Our work considers only building changes just to demonstrate the principle of joint semantic segmentation and change detection. The "building heightened" changes in Table 5.1 might be a "newly-built building" or a "heightened building"; The "building lowered" changes include "demolished buildings". Changes related to terrain or vegetation are assigned to the "Other (OT)" category.

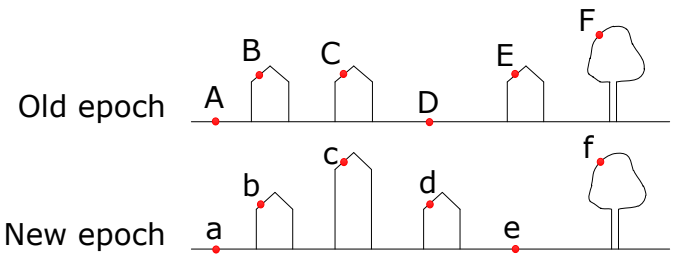



***Figure 5.1:*** *Schematic diagram indicating changes between two epochs. The top row shows the old epoch; the bottom row shows the new epoch. The red dots in the top row and bottom row are corresponding at the same location.*

***Table 5.1:*** *Our solution to define joint categories*

| Point | Change Type | Change code |
|---|---|---|
| A | Terrain to terrain, unchanged | TU |
| B | Unchanged building | BU |
| C and D | Building heightened | BH |
| E | Building lowered | BL |
| F | Vegetation to vegetation, unchanged | VU |

The rules for defining the categories in the outputs should cause the minimum information redundancy. Tran et al. (2018) extract features from both epochs, classify the point clouds in each epoch separately and then compare the labels for change detection. Their solution of defining joint categories is to assign changes labels to data of both epochs. However, this causes much redundancy. For instance, if "from D to d" shows a new building; "from d to D" must be "building lowered" change. If "from F to f" shows an unchanged tree, "from f to F" must be an unchanged tree. There is no logical need to define them separately in both epochs.

Our solution to define the joint labels is shown in Table 5.1. Specifically, only ALS points are labelled. If an ALS point is unchanged, we assign it with a semantic label; If an ALS point is changed, we assign it with a change label. Specifically, C and D indicate that a new building (terrain-to-building) and a building with an additional floor (building-to-building) are in the same Building Heightened (BH) class; E indicates that a demolished building is in the Building Lowered (BL) class. The major difference is that the method by (Tran et al., 2018) detects changes to the point clouds from both epochs, while our method labels joint-change transitions only in the point cloud of one epoch. Our solution to define joint categories is concise with no information redundancy.

In order to solve the problem of semantic segmentation and change detection in a single workflow, a Siamese neural network architecture is proposed for the joint tasks. Deep neural networks have demonstrated its superior performance in many computer vision tasks, such as image classification, semantic segmentation, change detection, object detection, etc. The neural networks have also been applied to extract features from the point clouds. In our previous work (Zhang et al, 2019), a Siamese neural network is proposed for change detection between multimodal point clouds assisted by orthoimages. The network takes 2.5D DSMs and 2D orthoimage as the inputs and makes a binary inference in the output, i.e. changed or non-changed. However, the conversion from 3D point clouds to 2.5D DSMs causes information loss and the derived change map is patch-based with a coarse resolution. In this work, we aim for point-wise labels for each ALS point. In the outputs, each ALS point is labelled into one of the six classes: TU, BU, BH, BL, VU or OT.

The contributions of this chapter are as follows:

(1) We propose an end-to-end Siamese network to infer semantic labels and change labels. The network takes multimodal point clouds from two epochs as inputs. It outputs a pointwise joint label for each ALS point. The semantic segmentation and change detection information are included in the joint labels with minimum information redundancy. Intra-epoch features are extracted at multiple scales to embed the local and global information. Inter-epoch features are concatenated to make change inference. This architecture can also be extended to other change detection tasks between point clouds from other platforms or in other modalities.

(2) We propose a Conjugated Ball Sampling (CBS) method to extract inter-epoch features from two epochs at the same centroids. It ensures that the feature vectors extracted from the DIM data are at the same location with the ALS data.

(3) Experiments are implemented on the Rotterdam data set. Comparing with three other methods, this method requires far fewer hyper-parameters and much less human intervention but achieves superior performance.

## 5.2 Related work on 3D semantic segmentation

Our work relates to both 3D semantic segmentation and change detection. The readers are referred to section 4.2.1 for a review on 3D change detection methods. This chapter reviews the methods for 3D semantic segmentation.

In the field of computer vision and remote sensing, 3D semantic segmentation refers to assigning a class label to each basic unit of 3D data, such as terrain, vegetation, building, water, pedestrian, etc. The basic processing unit, also called "entity", may be a point, a segment, or a voxel. The methods for semantic segmentation can be categorized into unsupervised and supervised as shown in Figure 5.2. Unsupervised methods are usually rule-based classification relying on handcrafted features. Supervised methods can be categorized into machine learning methods, contextual features-based classification and deep learning-based classification. Deep learning based-methods can be categorized according to the representation of 3D data, e.g. multi-view representation, 2.5D representation, voxel, point cloud, and graph.

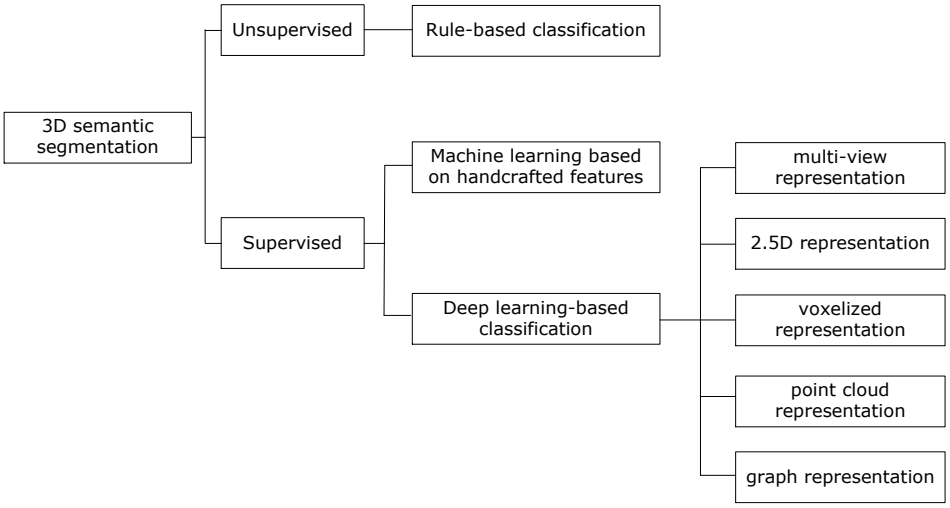



***Figure 5.2:*** *Overview of 3D semantic segmentation methods*

Figure 5.2 also indicates the general history of 3D semantic segmentation methods. For example, the features evolve from handcrafted features to deep learning-based features. At the beginning, deep learning features are extracted from multi-view images and then fused for inference. Then 3D data are transformed to 2.5D data or voxels where features are extracted. With the pioneering proposal of PointNet (Qi et al, 2017a) and PointNet++ (Qi et al., 2017b), features are extracted from each point. In graph representation, the point features are represented in graph nodes while the contextual features are represented in edges.

### *5.2.1 Rule-based classification*

The process for rule-based classification is to extract handcrafted features from the points or segments, and then to classify the features based on classification rules and prior knowledge. The method relies on representative features and discriminative rules. For example, a smoothness feature can be used to discriminate between vegetation and building roof. A point with a smoothness value below than a threshold $T$ is more likely to be a building roof rather than vegetation. Therefore, the effect of this method also relies on the set of classification thresholds. Many manual efforts are required for this type of method.

The basic entity for feature extraction and rule design can be a single point. For example, Pu et al. (2011) classify on-board MLS point clouds based on the "from coarse to fine" strategy. They first classify the points into ground surface, objects on ground, and objects off ground. The objects on ground are assigned to more detailed classes such as traffic signs, trees, building walls and barriers

based on size, shape, orientation and topological relationships. The rule-based classification are also used in (Maltezos and Ioannidis, 2015) to classify LiDAR point clouds and dense matching points with roughness and NDVI.

Instead of single point-based features, some rule-based classification takes a segment as the basic processing unit. Vosselman (2013) first extracts planar segments based on surface growing, and then extracts roof segments based on smoothness feature. Segment growing and connected component analysis are employed to classify the remaining points into terrain and vegetation. The method relies on robust surface growing method and threshold determination. Lin and Zhang (2014) proposed a segmentation-based filtering algorithm to extract ground points from laser point clouds. The method contains three steps: point cloud segmentation, multiple echoes analysis, and iterative judgment. The filtering unit is a segment rather than a single point.

In addition, some work fuses point-based features and segment-based features for robust classification. Xu et al. (2012) extract three types of features from each ALS point cloud, and classify the scene according to prior knowledge. The features contain point-based features, features based on planar segments, and features based on mean-shift segmentation. The classification tree is manually designed to divide the points into five classes. The way to determine the threshold for each feature is to plot the feature distribution as a histogram and manually determine the threshold that can distinguish different classes. Gilani et al. (2016) combine image features and point cloud features to extract buildings from ISPRS Vaihingen benchmark dataset. The candidate building region is divided into grid where vegetation and shadow are excluded. The building outlines are regularized with edge features on the images.

### *5.2.2 Machine learning based on handcrafted features*

This type of classification method employs handcrafted features which are classified by supervised machine learning algorithms. The tedious work of designing classification rules and determining their thresholds is undertaken implicitly by a classifier training. Therefore, the critical part of this method is "feature engineering". Feature engineering raises new questions: (1) How to evaluate feature contribution? (2) Is it true that the more features are better than less features? (3) How to perform feature selection? (4) Which machine classifier performs better? (5) How do the amount and distribution of training data affect the classifier performance?

Guo et al. (2011) extract echo features and full waveform features from LiDAR data, classify them with random forest classifiers, and analyze the correlation between LiDAR features and multispectral images. Xu et al. (2014) extract

point features, planar segment-based features and mean-shift segment features from point clouds, and classify laser point clouds into seven categories, using Random Trees, Adaptive Boosting, Artificial Neural Network, Support Vector Machine, and rule-based classification and compare their performance. Weinmann et al. (2015) analyze the effects of different feature combinations, neighborhood sizes of feature extraction, classifiers, and feature selection methods on classification accuracy. They propose an adaptive neighborhood selection method based on Shannon entropy.

This type of classification is relatively mature and has been widely used in other literature such as (Chehata et al., 2009), (Gerke and Xiao, 2014), (Blomley et al., 2014), (Martinovic et al., 2015), (Ni et al., 2016), (Roynard et al., 2016), (Hackel et al., 2016), (Ramiya et al., 2016), (Gevaert et al., 2016), (Thomas et al., 2018) et al. There is a trend to fuse multimodal features such as point cloud-based features, image-based spectral and textural features and DSM features on the basis that multiple features might be complementary.

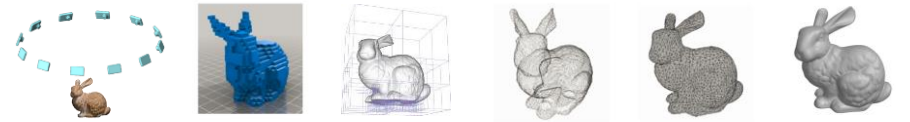
The above handcrafted features only take local neighborhood into consideration, classification with contextual features allows to take larger context into the model explicitly. Niemeyer et al (2014) classifies airborne laser points by integrating Random Forests into a Conditional Random Field (CRF) framework. The unary term of CRF is calculated from point-based features with Random Forests; The binary term is the interaction features of neighboring points. Comparing with their method, Vosselman et al. (2017) take segmented airborne laser data as the processing unit in a CRF framework. Different segmentation methods are integrated to minimize under- and over-segmentation. The edges in the CRF model are taken from the segment adjacencies and point-based features along the segment borders.

Similar contextual classification method is also used in (Lu and Rasmussen, 2012), (Wolf et al., 2015), (Guinard and Landrieu, 2017), (Zhu et al, 2017) et al. Through taking long interactions among points or segments, contextual classification results are expected to be relatively smooth and the details between object edges can be preserved.

### *5.2.3 Deep learning-based classification*

Deep learning algorithms are being used in point cloud semantic segmentation. It provides a direct solution to merge feature extraction and classification into Multilayer Perceptrons (MLPs). Convolutions Neural Networks show superior performance in 2D image segmentation but cannot directly not be applied to 3D point cloud classification. The challenges are two-folds: (1) Point cloud is a set of unordered 3D coordinates $(X,Y,Z)$. The point cloud is permutation-invariant. Namely, if the order of point cloud coordinates is changed, the object

is still the same one (Qi et al., 2017a). (2) In addition to 3D coordinates, the laser points may contain intensity, number of echoes, or other features. The DIM points may contain color information $(R, G, B)$. How to feed multimodal features into the neural networks is a problem. 3D data can be represented in different forms as shown in Figure 5.3. Deep learning-based classification methods can be divided based on different representations fed into the neural networks.



(a) Multiview images (b) Voxels (c) Octree (d) Point cloud (e) Meshes (f) CAD model

***Figure 5.3:*** *Representation of 3D data.*

Concerning multi-view representation, 2D CNNs are used on 2D images converted from point clouds. Boulch et al. (2017) convert 3D point clouds into RGB-D images from different perspectives and the depth corresponds to the geometric information. The images are then segmented by Fully Convolutional Networks (FCNs) and the labels are projected back to the original point clouds. Su et al (2015) and Zhang et al (2018) both propose multi-view CNN for 3D shape recognition. Firstly, images of an object are obtained from multiple perspectives, and then each image is classified by CNN. Finally, results of multiple views are integrated to make the inference decision.

Hu and Yuan (2016) are the first to apply CNN models for point cloud filtering. Firstly, the features extracted from point cloud is converted into the three channels of a "virtual color image". Then each image patch is classified into ground and non-ground with CNN. Similarly, Rizaldy et al. (2018) use FCN for laser point filtering. Firstly, elevation, intensity, number of echoes and elevation difference are converted into a four-channel image, and then each pixel is classified into ground and non-ground by FCN. Finally, the label is transmitted to the original point cloud.

In terms of urban scene classification, Audebert et al. (2018) combine orthophotos and DSM as input and add multi-scale modules to the SegNet and ResNet. They also compare the performance of early fusion and late fusion. Considering that the classification edges between objects get blurry, Marmanis et al. (2018) add boundary detection into the SegNet model explicitly. Similar work can be found in (Yang et al., 2017), (Liu et al., 2017), (Gupta et al., 2016), (Hazirbas et al., 2016), (Wen et al., 2019).

In addition to 2D CNNs, some works convert point clouds into voxels and then feed them into a 3D CNN. Maturana and Scherer (2015) propose to convert

point cloud into 32×32×32 voxels, and then apply 3D CNN to classify the voxelized objects. Since 3D convolution on the voxels takes much computational memory, Riegler et al. (2016) builds an Octree to represent the point cloud. They propose the OctNet architecture for learning deep features from the specific data structure. Lei et al. (2019) propose an octree-based convolutional method, in which the spherical convolution kernel is designed for fast feature learning in the point cloud space. Similar voxel-based deep learning methods are used in (Huang and You, 2016), (Nagy and Benedek, 2019), (Qi et al., 2016), (Tchapmi et al., 2017), (Zhou and Tuzel, 2017), (Liu et al. 2019) and (Roynard et al., 2018) for feature learning from 3D point clouds.

The procedure of converting 3D point clouds into multi-view images, 2.5D DSMs or voxels not only causes information loss, but increases the workload. Qi et al. (2017a) propose PointNet as the pioneering research which feeds unstructured point clouds into MLPs to learn pointwise features. The network is invariant and stable to geometric transformation and permutation with the spatial transformer and max-pooling layers. To extract features from multi-scales, PointNet++ is proposed to learn hierarchical features through a series of combined sampling layer and grouping layer (Qi et al, 2017b).

On the basis of PointNet++, Yousefhussien et al. (2018) propose a 1D-fully convolutional network which takes normalized point cloud features and spectral features as input. The network achieves superior performance on the ISPRS 3D Semantic Labeling Contest data set. Su et al., (2018) first transform the point cloud space to a lattice space. They propose SPLATNet to learn features in lattice space with bidirectional convolutional layers. Lin et al. (2020) propose an active and incremental learning method to save the manual annotation work. The model knowledge increases in each iteration. Similar work of classifying unstructured point clouds with MLPs can be found in (Soilan et al., 2019), (Winiwarter et al., 2019), (Zhao et al., 2018), (Lian et al., 2019), (Hu et al., 2020), (Zhao et al., 2021) et al.

Point cloud semantic segmentation can also be performed over the graph representation: the points are taken as graph nodes while the neighboring relations are edges. Qi et al. (2017) propose a 3D Graph Convolutional Network (GCN) for RGB-D data classification. First, a k-nearest neighbor graph is built from the point clouds. The graph node is a set of points which is concatenated with the feature vector extracted from the 2D images. Each node updates dynamically based on current status and neighboring message. Landrieu and Simonovsky (2018) propose Superpoint Graphs to build graph on superpoints for semantic segmentation. The superpoints are a set of points which are geometrically homogeneous. The contextual relations in the graph are exploited by a GCN to model long-range interactions. Wang et al. (2019)

propose Edge Convolution (EdgeConv) to model the contextual relations in the point clouds and higher dimensional feature space. The graph is dynamically updated after each convolution.

In summary, the state-of-the-art methods for 3D semantic segmentation show the following trends: (1) The researches are pursuing end-to-end inference methods which takes raw point clouds as inputs while assigning a label to each point. The workload of pre- and post-processing is being minimized. (2) The state-of-the-art methods are developing towards the direction of extracting multi-scale features from a large context. From PointNet to PointNet++ and then to the Superpoint Graphs, the neighborhood for feature aggregation becomes wider and wider. (3) Active learning and semi-supervised learning methods are taken into the 3D semantic segmentation tasks so that the requirement for large training samples can be reduced.

## 5.3 Methodology

Our workflow for combined SS and CD contains three steps: First, the dense DIM point clouds are denoised to eliminate isolated points and noise; Second, the conjugated training tiles are prepared with normalization; Third, a Siamese neural network is used to classify each ALS point into one of the six joint classes.

### 5.3.1 Thickness-Adaptive Denoising for DIM point cloud

The point clouds generated by dense image matching (DIM) are denser and noisier than the ALS points. The isolated blunders and random noise in the DIM points may hinder the extraction of representative features from the point clouds. Additionally, the high density of the DIM points causes high computational burden during the preparation of training samples and feature extraction. The target of DIM data denoising is to ensure that the DIM noise and blunders are largely reduced, and the DIM data density is at the level of the ALS point density. When the input point clouds of two epochs are at the same density level, the two Siamese branches get balanced.

A DIM point cloud shows up as a surface with points scattered around the object surface. The denoising method assumes that the most accurate DIM points lie in the center of the noisy point cloud profiles if the point cloud surface contains only one layer. The target is to select the skeleton points along the noisy DIM surface. However, point clouds may contain multiple layers such as at a heightened road, a wall, etc. Therefore, our method should distinguish two cases. We use a square without height bounds for a single layer and cubes for multiple layers to divide the point clouds. Finally, only one point is selected

from each square or cube as the sampled point. The method for DIM data denoising is illustrated in Figure 5.4.

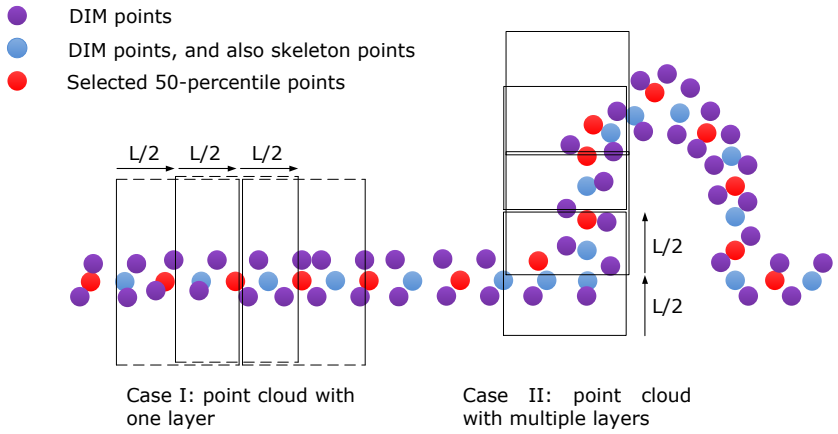



***Figure 5.4:*** *A profile for DIM point cloud denoising*

Figure 5.4 illustrates the ways of selecting skeleton points in two cases. The purple points are noise while the blue points are selected valid points. If we apply 50-percentile de-noising, all the facade points in case II would be removed. However, the facade points are assumed to be a discriminative indicator for the buildings, so it is necessary to keep the facade points during denoising. Therefore, point cloud with multiple layers should also be divided in the vertical direction.

The algorithm, named *Thickness-Adaptive Denoising*, is shown below:

(1) Given the original noisy DIM point set $(X, Y, Z)$, divide the point clouds into square grids with a length of $L$ and a stride of $L/2$ at the X-Y space.

(2) Iterate over each grid location $(X0, Y0)$ and make a judgement whether it has one layer or multiple layers. If the $H_{90\%}$-$H_{10\%}$ profile with all the points in this grid is lower than $H_d$, the point cloud surface is supposed to contain only one layer; otherwise, it is assumed to contain multiple layers. $H_d$ is the threshold for the thickness of point cloud surface. The next step goes to step (3) or step (4) depending on this judgement.

(3) If a grid has only one layer, we select the point at the 50% percentile of all the point heights within this grid as the skeleton point for this grid.

(4) If a grid has multiple layers, we divide the grid in the vertical direction into multiple cubes with a length of $L$ and a stride of $L/2$. Similarly, we select the point at the 50% percentile of all the point heights within each cube as the

skeleton point for each cube. Concerning data gaps, if there is no point in a cube, this cube is left empty.

In this manner, the skeleton points can be kept no matter whether the point cloud surface has one or multiple layers. The generated point cloud can also maintain the vertical distribution of point clouds in the raw data.

### 5.3.2 Preparation of conjugated training blocks

One training sample is a pair of tiles selected at the same centroids from two epochs. The pair of tiles are fed into the Siamese network for joint SS and CD. Given a centroid $(X, Y)$, the tile is taken in the space between $(X - \Delta X, Y - \Delta Y, -\infty)$ and $(X + \Delta X, Y + \Delta Y, +\infty)$. $\Delta X$ is usually set to be the same with $\Delta Y$ since square blocks give considerable attention to X and Y directions when embedding contextual information.

Suppose that $\left(X_i^{als}, Y_i^{als}, Z_i^{als}\right)$, $i \in (1, N)$ are the $N$ ALS points in the ALS tile. $\left(X_j^{dim}, Y_j^{dim}, Z_j^{dim}\right)$, $j \in (1, M)$ indicate the $M$ DIM points in the corresponding DIM tile. It should be noted that the number of points in the corresponding tiles are not the same; The point distribution or point sequence are different as well since the data of two epochs were acquired by different techniques. The only relation between the two corresponding tiles is that they are cropped at the same centroid within the same range. In our data preparation, the number of points sampled from each ALS tile and each DIM tile are constant so that they can be fed into the network. Concerning data normalization, the $(X, Y)$ coordinates are normalized by subtracting the centroids $(X_c, Y_c)$ of each tile; The $Z$ value remains unchanged: $\left(X_i^{als} - X_c, Y_i^{als} - Y_c, Z_i^{als}\right)$, $\left(X_i^{als} - X_c, Y_i^{als} - Y_c, Z_i^{als}\right)$.

### 5.3.3 Siamese pointwise network

Siamese Convolutional Neural Networks (S-CNN) have been used in dual-input tasks in both computer vision and remote sensing domain, e.g. for change detection, dense image matching, human re-identification, and image retrieval. The input to the two branches might be either from the same modality or from different modalities. Each of the two Siamese branches is composed of convolutional layers, batch normalization and non-linear activation for feature extraction. The extracted deep features in the end of the two branches are concatenated or subtracted for the specific tasks.

The proposed architecture aims to take point clouds as input and makes point-wise based inference. PointNet++ is a pioneering end-to-end architecture for point-wise semantic segmentation based on Multi-Layer Perception (MLP) (Qi

et al., 2017b). However, it was designed for semantic segmentation, object classification and part segmentation. There is still no research published till now on using neural networks for point-based change detection.

This paper proposes the Siamese pointwise network for joint SS and CD as shown in Figure 5.5. We take PointNet++ as the backbone of our architecture to check the feasibility of learning inter-epoch features with a Siamese architecture. The reason for taking PointNet++ as the backbone is that PointNet++ is the pioneering and fundamental model for feature extraction from point clouds. In each of the Siamese branch, multiscale features are extracted and aggregated with Multi-layer Perceptrons (MLPs). The model is named *SiamPointNet++*.

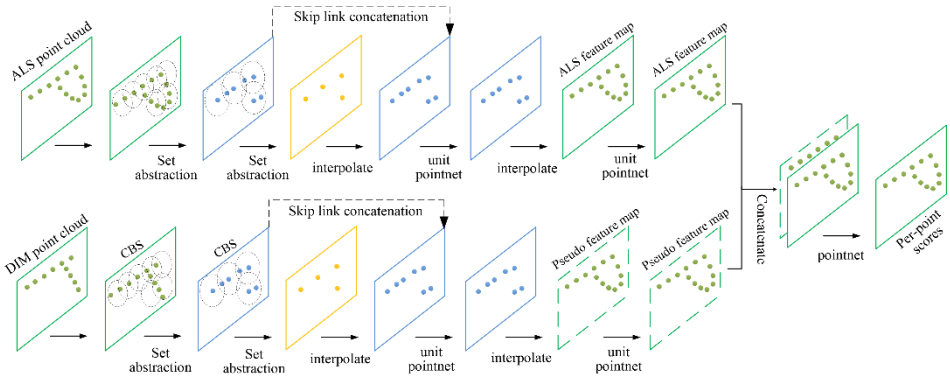



***Figure 5.5:*** *The proposed SiamPointNet++ network for joint semantic segmentation and change detection.*

**(1) Intra-epoch design:** Each Siamese branch extracts features from either the ALS points or the DIM points. Similar to Fully Convolutional Networks (FCN), our network with PointNet++ as the backbone contains encoders in the left half part and decoders in the right part. Concerning the encoder layers, local and global features are extracted from multiple scales using a series of set abstractions. Each set abstraction contains sampling, grouping and PointNet operations. Specifically, the centroids are sampled by Farthest Point Sampling (FPS) from the points in the last layer; The features of neighboring points within a fixed radius to each centroid are grouped and further processed by a unit PointNet. As the feature extraction goes deeper, more contextual information from a larger range is derived and aggregated.

In the decoder layers, the down-sampled point sets are gradually interpolated to the raw point distribution with skip link concatenations followed by a PointNet operation. The step is named "feature propagation" in the PointNet++. The skip link concatenation makes a connection between the encoders and decoders, which ensures that a feature vector is extracted for each ALS point.

Similar to (Qi et al, 2017b), we also group features from multiple neighborhoods to cope with non-uniform density in the point clouds, i.e. multi-scale grouping (MSG).

**(2) Inter-epoch design:** A similar branch is used for feature extraction from the DIM points (see the bottom row of Figure 5.5). The critical question is how to make a correspondence between the feature extraction in the two branches. If we apply a set abstraction independently to the DIM branch, the sampling centroids and the number of samples (dotted circles) would be quite different compared with the sampling in the ALS branch. If the samples are taken from different locations with no correspondence, a comparison between them is obviously meaningless.

We propose a Conjugated Ball Sampling (CBS) for the set abstraction on the DIM points as shown in Figure 5.6. When the sampling centroids are determined in the ALS points, the samples in the DIM data are taken at the same centroids as in the ALS data. This ensures that the local feature vectors taken from the ALS data and the DIM data are always corresponding to each other. Namely, we extract the features at the same ALS centroids with the neighboring DIM points – these features represent the neighboring contextual information in the DIM data.

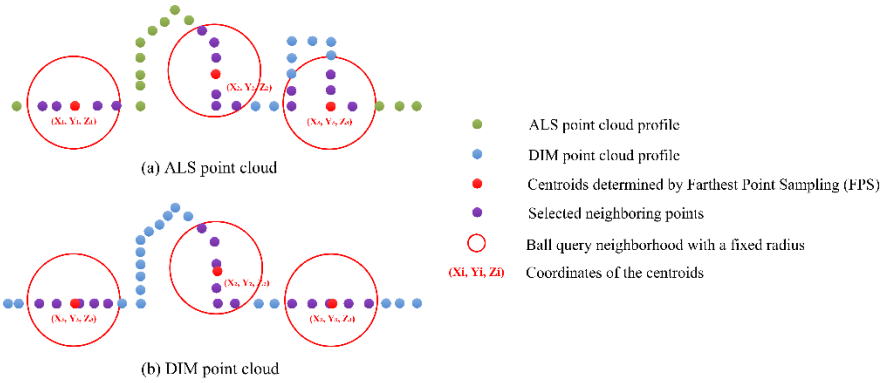



***Figure 5.6:*** *The point cloud profile for illustrating Conjugated Ball Sampling (CBS).*

After a series of set abstractions, the deep DIM features are derived from large contextual range. Concerning the decoder layers, features are interpolated and propagated to the higher densities. In the last layer, the DIM feature vectors are interpolated to the raw ALS point location, instead of the raw DIM point locations. This guarantees that the DIM features are calculated at the same centroids as the ALS data. Only those feature vectors extracted at the same centroids can be compared. Note that even if there is no DIM point in the conjugated ball of an ALS point, we should still calculate a pseudo "feature

vector" at the same centroid to "inform" the model that the ball neighborhood in the DIM data is empty. Our solution is simply to take the minimum corner of the tile as the only point in the ball neighborhood for feature vector calculation, which is [0, 0, 0] based on our normalization method. The weights in the two Siamese branches are not shared since the input point clouds have different properties.

**(3) Feature concatenation and inference**

The feature maps from the two Siamese branches in Figure 5.5 are calculated for each ALS point from multiple scales which contain local and global information. The two feature maps are concatenated and further processed by a vanilla PointNet, which fuses information from both epochs. The vanilla PointNet is composed of convolution, batch normalization, Rectified Linear Unit (ReLU) and dropout layers. The final output from the PointNet is a $6 \times 1$ feature vector for each ALS point, which indicates the probability for each category, respectively. First, the vector is normalized to $(0,1)$ by a Softmax function; Then, a weighted cross entropy loss is calculated for multiclass inference. Suppose that the class index is in the range [0, 5], which indicates one of the six categories: TU, BU, BH, BL, VU or OT. The loss for each class can be calculated by

$$Loss(x, class) = -weight[class] * log\left(\frac{\exp(x[class])}{\sum_i \exp(x[i])}\right), \quad i = 0{\sim}5 \qquad (5\text{-}1)$$

Where $x$ is the predicted vector from the vanilla PointNet. The $weight$ is set based on the ratio of number of different classes. By assigning weights to the loss function, we impose a stronger response to the model when small-sample classes are met. This gives larger penalization to a false positive than to a false negative to suppress false positives.

## 5.4 Experimental settings

### 5.4.1 Data description

The study area is located in Rotterdam, The Netherlands, which is a densely built port city mainly covered by residential buildings, skyscrapers, vegetation, roads, and waters. The study area is 14.5 km$^2$ as shown in Figure 5.7. Figure 5.7(a) shows the ALS point cloud obtained in 2007 with a density of approximately 25 points/m$^2$. The point cloud contains approximately 226 million points.

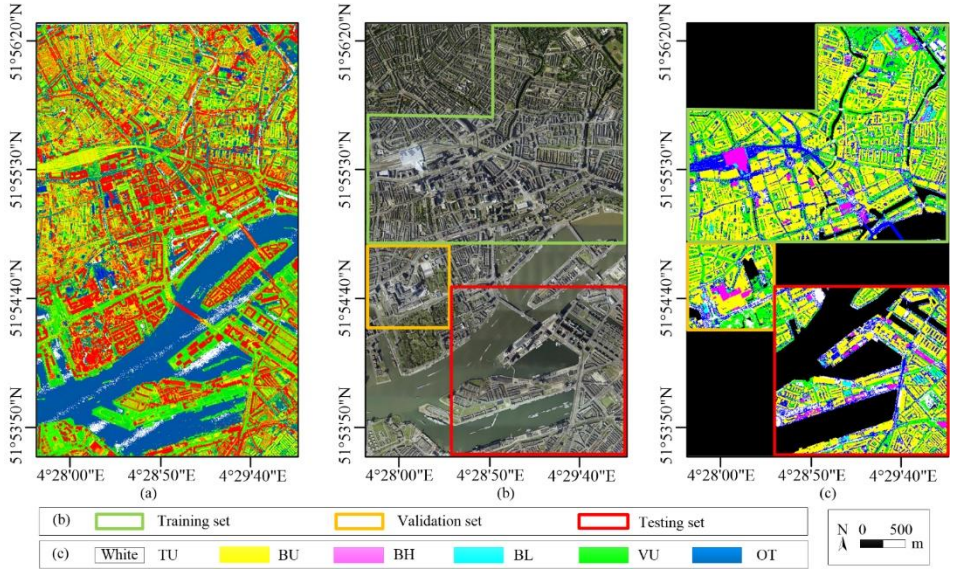



***Figure 5.7:*** *Visualization of the data set. Top row from left to right: (a) ALS points colored according to height; (b) Orthoimage marked with training, validation and testing area; (c) ALS points labelled into six categories.*

A total of 2160 aerial images were obtained by CycloMedia from five perspectives in 2016. The flying altitude was approximately 450 m. The tilt angle of the oblique view was approximately 45°. The image size was 7360 × 4912 pixels. The GSD of nadir images equaled 0.1 m. The bundle adjustment and dense image matching were performed in Pix4Dmapper. The vertical RMSE (Root Mean Square Error) of 48 GCPs was ±0.021 m and the vertical RMSE of 20 check points was ±0.058 m. The overlap of nadir images was approximately 80% along the track and 40% across the track. Even though the overlap rate from five views is high, dense matching still cannot perform well in the narrow alleys between tall buildings due to poor illumination or a lack of surface texture. The DIM point cloud contains approximately 281 million points. DSMs and orthoimages were also generated at the same resolution of 0.1 m. Figure 5.7(b) shows the generated orthoimage. The training, validation, and testing area are 6 $km^2$, 1 $km^2$, and 4 $km^2$, respectively. Note that 4 $km^2$ of the block is not used since these areas contain few changes. Figure 5.7(c) shows the point clouds labelled into six categories: *Terrain Unchanged* (TU), *Building Unchanged* (BU), *Building Heightened* (*BH*), *Building Lowered* (*BL*), *Vegetation Unchanged* (*VU*) or *Other* (*OT*).

Some statistics of the experimental data are shown in Table 5.2. As to be expected it shows that the six classes in the data set are imbalanced. The TU and BU are the majority classes; The BL class contains the least samples.

***Table 5.2:*** *Number of samples for training, validation and testing.*

| Number of Points | TU | BU | BH | BL | VU | OT | Sum (ALS) | Sum (DIM) |
|---|---|---|---|---|---|---|---|---|
| Training set | 36,064,519 | 10,657,456 | 1,327,245 | 313,964 | 3,983,353 | 2,373,646 | 54,720,183 | 72,258,941 |
| Validation set | 2,435,464 | 1,269,007 | 186,981 | 16,862 | 650,426 | 288,388 | 4,847,128 | 6,587,198 |
| Testing set | 8,134,193 | 4,452,199 | 317,917 | 181,361 | 571,972 | 796,712 | 14,454,354 | 19,703,395 |

The following should be noted during manual labeling: (1) The ALS point cloud is manually labelled in two steps. Firstly, it is manually labelled into terrain, buildings, vegetation and other; Secondly, the semantic labels are further labelled into change labels with guidance of the ALS points, DIM points, and DSM differencing map. (2) Specifically, when a building is newly-built or heightened, its boundary is delineated from the DIM points; When a building is demolished, its boundary is delineated from the outdated ALS points. (3) Water areas are not considered for change detection and therefore omitted. (4) Data gaps may appear in ALS points and DIM points. If there is no data in either epoch, we simply cannot make any inference whether it is changed or not. Therefore, if an ALS point appears where it is data gap in the DIM data, this ALS point is labelled into other (OT). (5) Finally, each ALS point is labelled into TU, BU, BH, BL, VU or OT. The BH class includes heighted buildings and new buildings; The BL class indicates demolished buildings.

## *5.4.2 Preprocessing*

Concerning the Thickness-Adaptive Denoising for DIM point cloud, the parameter $L$ for grid sampling is set to 0.2 m according to the density of ALS point cloud so that the input data to the Siamese network are at the same density level. $L$ is. $H_d$ is set to 0.5 m as the normal thickness of DIM point cloud surface.

ALS data and DIM data in the data set are cropped into tiles so that they can be fed into the network. The tile size should be large enough to contain sufficient context, and small enough concerning the limited GPU memory. In our experiments, the training area is divided into 50 m $\times$ 50 m tiles with a stride of 20 m. The overlap allows to generate more training samples. When preparing validation and testing tiles, the tile size remains 50 m $\times$ 50 m at a stride of 50 m so that each tile is inferred only once. Concerning data normalization, Qi et al. (2017b) normalize the point cloud coordinates in each tile to [0,1] in X, Y and Z directions, respectively. In contrast, we subtract the $X$ and $Y$ coordinates with the starting position in each tile; The true Z coordinates are fed into the network without normalization. The brings two

benefits: First, the $Xs$ and $Ys$ in all the tiles are normalized into [0, 50m], which avoids the impact of horizontal deviations of tiles; Second, this allows to keep the relative geometric relations among X, Y, Z. Our trial test shows that this normalization method works better than the method in (Qi et al., 2017b) for our tasks.

The hyper-parameters in the proposed network are fixed whereas the number of points varies from tile to tile. Therefore, fixed number of points are sampled from the conjugated ALS tile and DIM tile, respectively during each iteration. Specifically, we randomly select 20,000 points from the ALS tile and DIM tile, respectively at the same location. For tiles with more than 20,000 points, we select points without replacement; For those with less than 20,000 points, all points are used as input and the rest is selected by random and repetitive sampling (Qi et al., 2017b). It should be noted that the samples taken from the same tile might differ due to random sampling, which makes the training samples more diverse. In addition, we also add Gaussian white noise with a $\sigma$ of 3 cm to XYZ coordinates to augment the training data in order to make the model more robust to noise.

### 5.4.3 Network implementation and training details

The backbone of the proposed network is a Siamese PointNet++ architecture. Each Siamese branch contains three set abstraction modules for hierarchical sampling. Since the densities of ALS points and DIM points are not uniform, the proposed method adopts multi-scale grouping to aggregate local features from multiple scales (Qi et al., 2017b). Table 5.3 shows the hyper-parameter configuration in the set abstraction modules.

***Table 5.3:*** *Parameter configuration of multiple set abstraction modules in each PointNet++ (MSG) branch*

| Level | Number of points | Search radius (m) | Number of neighbors |
|---|---|---|---|
| 0 | 20,000 | | |
| 1 | 4096 | [4.0, 8.0] | [16, 32] |
| 2 | 1024 | [8.0, 16.0] | [16, 32] |
| 3 | 256 | [16.0, 32.0] | [16, 32] |

In Table 5.3, the number of points and number of neighbors are set based on empirical tests, while the search radius is set based on the point cloud density. The first grouping layer selects 4096 points from 20,000 points in the ALS tile by Farthest Point Sampling. Then, 16 neighboring points are grouped within a spherical search radius of 4 m; 32 neighboring points are grouped within a spherical search radius of 8 m; The features from two scales are concatenated. For the next set abstraction layers, 4096 points are subsampled to be 1024 by FPS. Then 16 neighboring points are gathered within a search radius of 8 m

and 32 points are gathered within 16 m. In the last set abstraction layer, 256 points are sampled from 1024 points and the neighboring points are grouped. As the set abstraction layers go deeper, fewer points are sub-sampled at higher levels which inevitably causes information loss but allows representing point features in wider range. It should be noted that FPS is not implemented in the DIM branch and the centroids for feature grouping are taken from the ALS branch.

The outputs from the ALS branch and DIM branch are concatenated and then go through a sequence of Convolution-Batch Normalization-ReLU-Dropout layers before making the output. The Cross Entropy Loss is computed with a weight of [0.04, 0.04, 0.34, 0.34, 0.12, 0.12] to cope with the sample imbalance. This weight is set according to the approximate ratio of number of different classes. In the testing stage, some points remain unlabeled in the original point clouds due to down-sampling. The labels of sub-sampled points are propagated to the original point cloud by Nearest Neighboring interpolation.

The SiamPointNet++ network is trained from scratch. The CNN architectures were implemented in PyTorch. The batch size is set to 4, which indicates that four pairs of conjugated ALS and DIM tiles are fed into the network in each iteration. The Adam optimizer (Kingma and Ba, 2014) is utilized for network optimization. The network is trained for 100 epochs on an NVIDIA RTX2080 GPU and validated every three epochs to ensure that there is no overfitting. Towards the end of training, the model with the best validation performance is taken for testing.

### 5.4.4 Contrast experiments

Apart from the proposed method, three other methods are implemented to compare their performance: SiamPointNet++ (SSG), Object-based change detection (OBCD) and Supervised change detection (SCD). The details of these methods are listed below:

**(1) SiamPointNet++(SSG):** Apart from the multi-scale grouping, single-scale grouping (SSG) is also implemented as a baseline method for comparison. SSG extracts features from single scale in each set abstraction layer. The parameter configuration is shown in Table 5.4. In order to make a distinction, we name the SiamPointNet++ with multi-scale grouping "SiamPointNet++ (MSG)" and name the SiamPointNet++ with single-scale grouping "SiamPointNet++ (SSG)".

***Table 5.4:*** *Parameter configuration of each SiamPointNet++ (SSG) branch*

| Level | Number of points | Search radius (m) | Number of neighbors |
|---|---|---|---|
| 0 | 20,000 | | |
| 1 | 4096 | [4.0] | [16] |
| 2 | 1024 | [8.0] | [16] |
| 3 | 256 | [16.0] | [16] |

**(2) Object-based change detection (OBCD):** This method is based on robust object extraction from the ALS data. It can be divided into two stages as shown in Figure 5.8: Firstly, ALS points are segmented and classified into terrain (T), buildings (B), vegetation (V) and other (O); Secondly, changes are detected by comparing the objects or segments to the new DIM data. Specifically, the ALS points are filtered with progressive TIN densification (Axelsson, 2000). Then planar segments are extracted from the non-terrain points by surface-based growing (Vosselman, 2013). The planar segments are merged to guarantee spatial coherence of the points belonging to one segment. Then six handcrafted features are calculated to recognize roof segments and wall segment: segment size, linearity of segment, planar slope, average angle, residual of plane fitting, and planarity (see the feature definitions in Table 2.1 of Chapter 2). The roof segments and wall segments are selected based on handcrafted features and then merged into complete buildings.

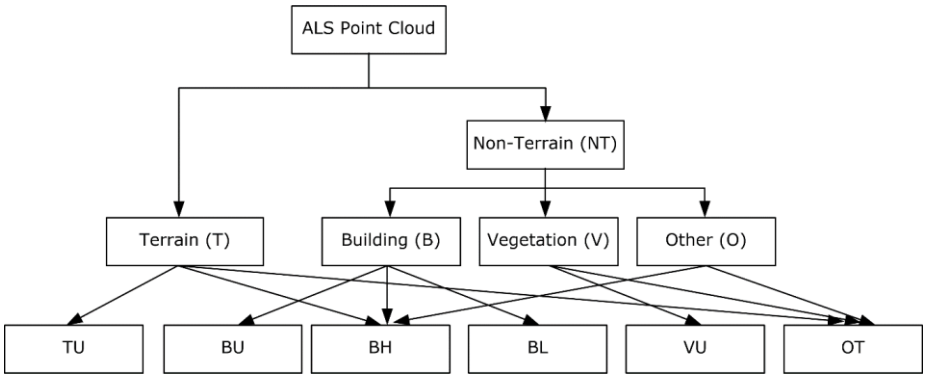



***Figure 5.8:*** *The workflow for object-based change detection (OBCD)*

Vegetation shows up in clusters in ALS points. For the remaining points, major vegetation points are identified by a *planarity* feature (Vosselman, 2013). The neighboring vegetation points are added to the major clusters by connected component analysis. By this time, unsegmented points are further classified based on the neighboring points within some distance by majority filtering.

In the change detection stage, the well-segmented ALS points are compared to the DIM data. Terrain is classified into TU, BH and OT based on the point to

plane distance. If the distance from an ALS point to the DIM plane is larger than a change threshold $T$, the ALS point is changed; Otherwise, it is unchanged. The heightened terrain points are grouped into BH change. $T$ is set to 2.5 m in our experiment.

The building points are classified into BU, BH and BL. Building changes are determined by comparing each roof segments with the corresponding DIM points. The rules for change detection are as follows: (1) For small roof segments with less than 150 points, if the segment-based height change is larger than the threshold $T$, the building is heightened or lowered; Otherwise, it is unchanged. (2) For larger segments, a part of a building might be changed, but the other part may remain unchanged. Therefore, we make a judgement to each roof point separately based on the point-to-plane distance. Then the BU, BH or BL roof points are grouped, respectively. Finally, the roof points are also grouped with the below wall points to form complete changed buildings.

Unchanged vegetation (VU) is identified by validating the vegetation class in the corresponding DIM data and calculating the height change. If the nDSM and normalized vegetation index (nEGI) are both larger than their thresholds, the objects are identified as vegetation in the DIM data.

It should be noted that OBCD requires to calculate certain handcrafted features and determine the thresholds for both object extraction and change detection. These parameters are usually determined by empirical tests based on data properties. The OBCD method for multimodal change detection requires careful refinement and post-processing before the results are obtained.

(3) Supervised change detection (SCD): This method is taken from (Tran et al., 2018) for integrated semantic segmentation and change detection. They first extract handcrafted features for each ALS point and then use Random Forests (RF) (Breiman, 2001) for per-point classification. Since the inferred labels are change labels, the features contain not only features from each epoch, but also multi-epoch features indicating topographic changes. The applied features are listed in Table 5.5. Geometric features are calculated from a rectangular neighborhood or k-nearest neighbor (kNN) to represent local point distribution and physical properties (Weinmann et al., 2015; Gevaert et al., 2017); Features from orthoimages include normalized R, G, B and nEGI; nDSM is the normalized Z for each point. The readers can refer to section 4.3.2.2 for the details of these single-epoch features.

***Table 5.5:*** *Feature sets used to classify the ALS points*

| | Geometric features | Features from orthoimages | nDSM |
|---|---|---|---|
| Single-epoch ALS | 23 | 0 | 1 |
| Single-epoch DIM | 23 | 4 | 1 |
| | DiffH | Stability | |
| Multi-epoch | 1 | 2 | |

Multi-epoch features contain *DiffH* and *stability*. DiffH is the Z difference between the considered ALS point and the closest DIM point. *Stability* is proposed in (Tran et al., 2018) as a discriminative feature for combined semantic segmentation and change detection. It is calculated from the neighboring point cloud of the other epoch as the ratio of number of points within the 3D neighborhood to the 2D neighborhood.

The next problem is to concatenate the ALS features and DIM features. We simply search the closest DIM point to each ALS point. The 28 single-epoch features from the closest DIM point are assigned to each corresponding ALS point. Finally, the single-epoch features and multi-epoch features are concatenated into a 55-dimensional vector (see Table 5.5). In the experiment, we select 1000 samples randomly from the training set for each class and train a RF model. Then the model is tested on the testing set.

### *5.4.5 Evaluation metrics*

Intersection over Union (IoU) is computed on the testing set to evaluate the performance of each method. IoU per class is computed from true positives (TP), false negatives (FN) and false positives (FP). Overall accuracy (OA) and mean IoU (mIoU) are computed to evaluate the overall performance:

$$OA = (TP + TN)/(TP + FN + FP + TN) \tag{5-2}$$

$$IoU_i = TP_i/(TP_i + FN_i + FP_i) \tag{5-3}$$

$$mIoU = \frac{1}{N} \cdot \sum_1^N IoU_i \tag{5-4}$$

Where $N$ indicates the number of classes; I indicates a certain class.

## *5.5 Results and analysis*

During validation, the highest mIoU achieved by SiamPointNet++ (MSG) model reaches 69.18%. The validation OA reaches 91.07%. The validation IoU for TU,

BU, BH, BL, VU and OT is 91.53%, 84.60%, 63.52%, 59.55%, 83.24%, 32.67%, respectively. This model is taken as the final model for testing. The testing results from the four methods are evaluated quantitatively as shown in Table 5.6.

***Table 5.6:*** *Testing results of the four methods (%)*

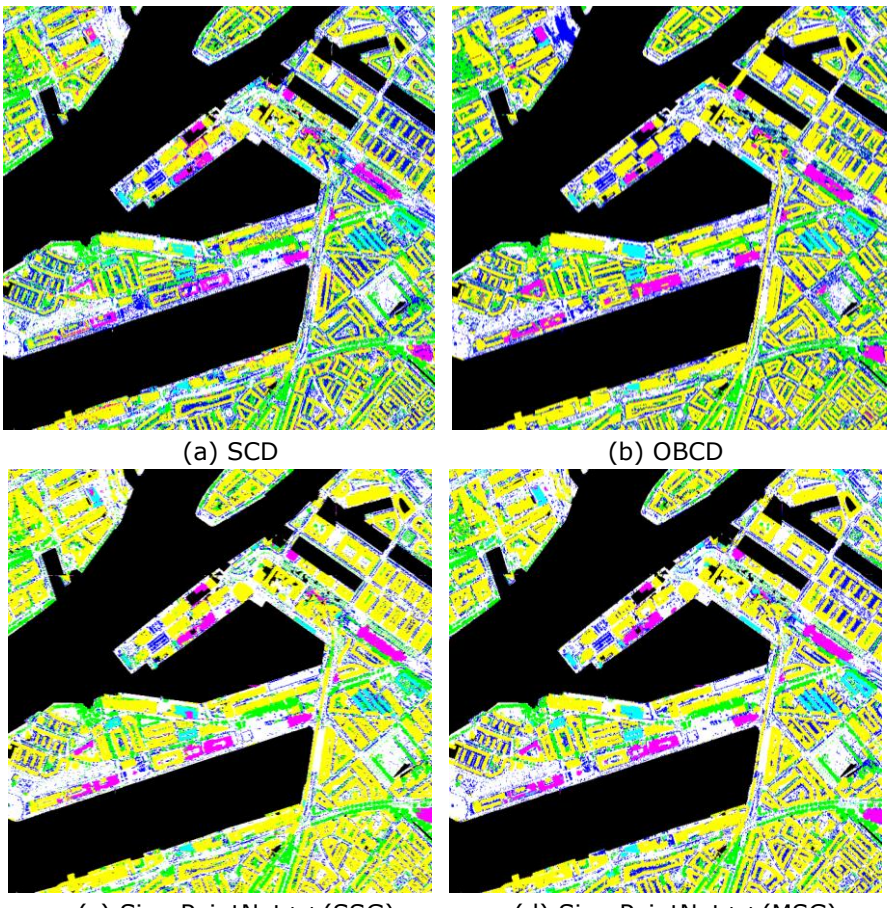
| | OA | mIoU | TU | BU | BH | BL | VU | OT |
|---|---|---|---|---|---|---|---|---|
| SCD | 86.86 | 56.51 | 90.70 | 78.16 | 26.04 | 36.40 | 67.90 | 39.84 |
| OBCD | 80.16 | 67.01 | 70.18 | 93.18 | 58.98 | 86.73 | 76.92 | 16.06 |
| SiamPointNet++(SSG) | 91.08 | 65.03 | 92.39 | 84.73 | 61.57 | 60.69 | 63.88 | 26.92 |
| SiamPointNet++(MSG) | 91.06 | 68.07 | 91.74 | 84.46 | 58.61 | 65.44 | 73.92 | 34.27 |

Table 5.6 shows that the proposed SiamPointNet++ (MSG) model outperforms the other three methods in mIoU. The mIoU of MSG is higher than that of SSG by a margin of 3.04%, while its OA is very close to the highest OA achieved by SSG with a small gap of 0.02%. The mIoU of OBCD ranks the second while its OA ranks last. SCD ranks last in mIoU although its OA ranks between the two MLP methods and the object-based method. Concerning the IoU for each class, OBCD achieves the highest IoU among the four methods for the classes of BU, BL and VU; SCD achieves the highest IoU in TU and BH. Although none of the six classes in MSG achieves the highest IoU, its averaged mIoU ranks the first.

Some initial analyses can be made based on Table 5.6. The proposed SiamPointNet++(MSG) model performs the best among the four methods. Its mIoU outperforms that from the SSG model, which indicates that multi-scale grouping allows to group more representative features from larger context in a hierarchical manner. The MSG model is more robust to the non-uniform point density in the point clouds compared with SSG. The OBCD achieves the highest IoU for BU, BL and VU. The reason is that sophisticated object extraction workflow leads to reliable segments for building and vegetation classes. The object-based change detection is implemented by segment-to-segment comparison, which is more tolerant to data noise compared with point-to-point change detection. Therefore, OBCD outputs fine results in BU, BL and VU classes. However, concerning the terrain change, OBCD performs worst among the four methods. The reason might be that point-to-point height differencing is sensitive to data noise. The DIM points are usually noisy in the area with low image contrast, e.g. in the shadow or along the narrow alley. Since TU takes the biggest share in terms of sample size, the OA decreases to the last, but its mIoU still ranks the second. SCD performs last in terms of mIoU despite the additional use of RGB features. This can be explained by that SCD extracts features from a small neighborhood and aggregates limited contextual

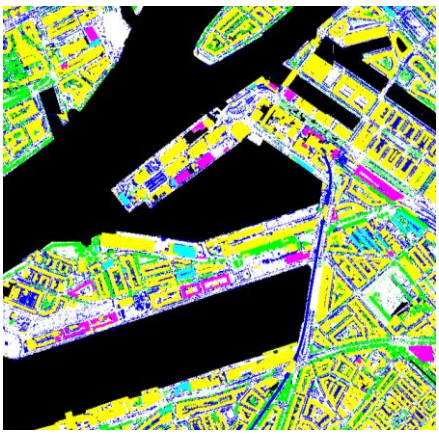
information. In contrast, SiamPointNet++ groups features in a hierarchical manner; OBCD makes use of wide contextual features embedded in each segment or cluster after surface-based growing and connected component analysis.

The predicted labels and ground truth are visualized in Figure 5.9. A comparison with ground truth shows that all the four methods can generate strong responses at changed locations even though some confusion occurs in the change maps. Generally, the two SiamPointNet++ models produce smoother change detection results compared with SCD or OBCD. Namely, less confusion appears among different classes in the change map from SiamPointNet++. This verifies that the proposed SiamPointNet++ model can learn both intra-epoch features and inter-epoch features for combined semantic segmentation and change detection tasks.



(a) SCD (b) OBCD

(c) SiamPointNet++(SSG) (d) SiamPointNet++(MSG)



(e) Ground truth

White TU BU BH BL VU OT

***Figure 5.9:*** *Visualization of the predicted labels and ground truth on the testing set.*

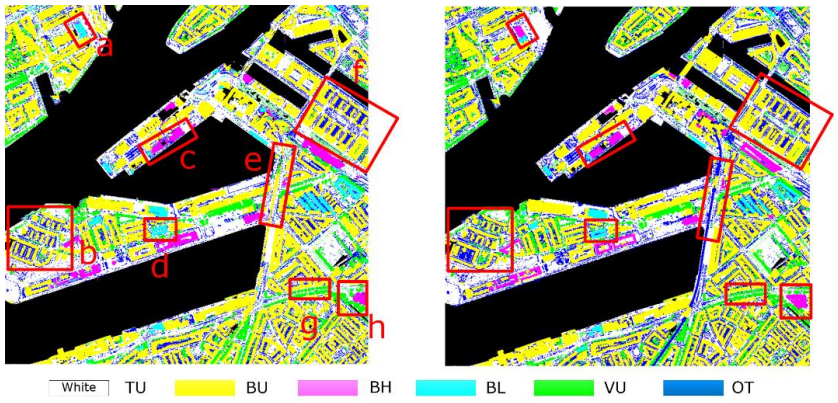
Figure 5.9 also shows that all the four methods can extract demolished buildings with sharp boundaries but the boundaries for heightened buildings are hard to determine. The reason is that the boundaries for BL are delineated from the precise ALS points while the boundaries for BH are determined from the noisy DIM points. This corresponds with the quantitative results in Table 5.6 in which the IoU for BL is typically higher than the IoU for BH, especially in the method of OBCD.

Figure 5.9(a) shows that many scattered TU and BU points are mis-classified into BH by SCD. This situation is more common along building boundaries. The reason might be that point clouds generated by dense matching in these locations are usually noisier due to low image contrast than those points in the open area. During change detection, the heightened DIM noise is mis-classified into heightened building. In contrast, the errors of BU misclassified into BH are less common since object-based comparison is more robust to point cloud noise. Figure 5.9(b) shows that most changes to buildings and vegetation are correctly detected, which is in accordance with the results in Table 5.6. However, Figure 5.9(b) also shows that many TUs are mis-classified into OTs due to sensitive point-to-surface comparison.

The visual difference between Figure 5.9(c) and (d) is not distinctive. A comparison with GT shows that many OTs are omitted in the SSG but correctly detected by MSG. The multi-scale grouping seems to learn more representative features so that small and difficult OT class are better identified by this model.

Eight examples selected from the change maps in Figure 5.10 are visualized in Figure 5.11. Figure 5.11(a) shows that a new building and a demolished building are adjacent to each other. The OBCD performs the best in this case. Although SCD succeeds in distinguishing heightened and lowered building, the walls are misclassified into BU. The two SiamPointNet++ models classify most part of the roofs into BL with a small portion of BH. The wall points are mis-classified into BU by MSG and BU and VU by SSG. Figure 5.11(b) shows the scene with TU, BU, VU and OT. Generally, major points are correctly classified by the four methods, except that many TU points close to buildings are mis-classified into BH by SCD.

Figure 5.11(c) shows that all the four methods can detect the new building. Although the new building has regular boundary, the detected change boundaries are not regular because they are determined on the DIM points and delineated on the ALS terrain. A scaffold appears in the ALS data but disappears from the construction site in the DIM data. It is mis-classified into VU by SCD and OBCD but mis-classified into BL by SSG and MSG. The property of the scaffold is similar to vegetation in that it is elevated objects with scattered structures. In some locations, the surface of the scaffold may appear as planes which are misclassified into buildings. Figure 5.11(d) shows that the demolished building is correctly detected by SCD, OBCD and MSG. The facade is mis-classified into VU by SSG. The change map from SSG shows that some VUs are close to this building. The VU and BL are connected in the results of SSG. In contrast, MSG correctly distinguishes these two changes.




(a) SiamPointNet++(MSG) (b) Ground truth

***Figure 5.10:*** *Eight examples selected from the testing results for visual analysis.*

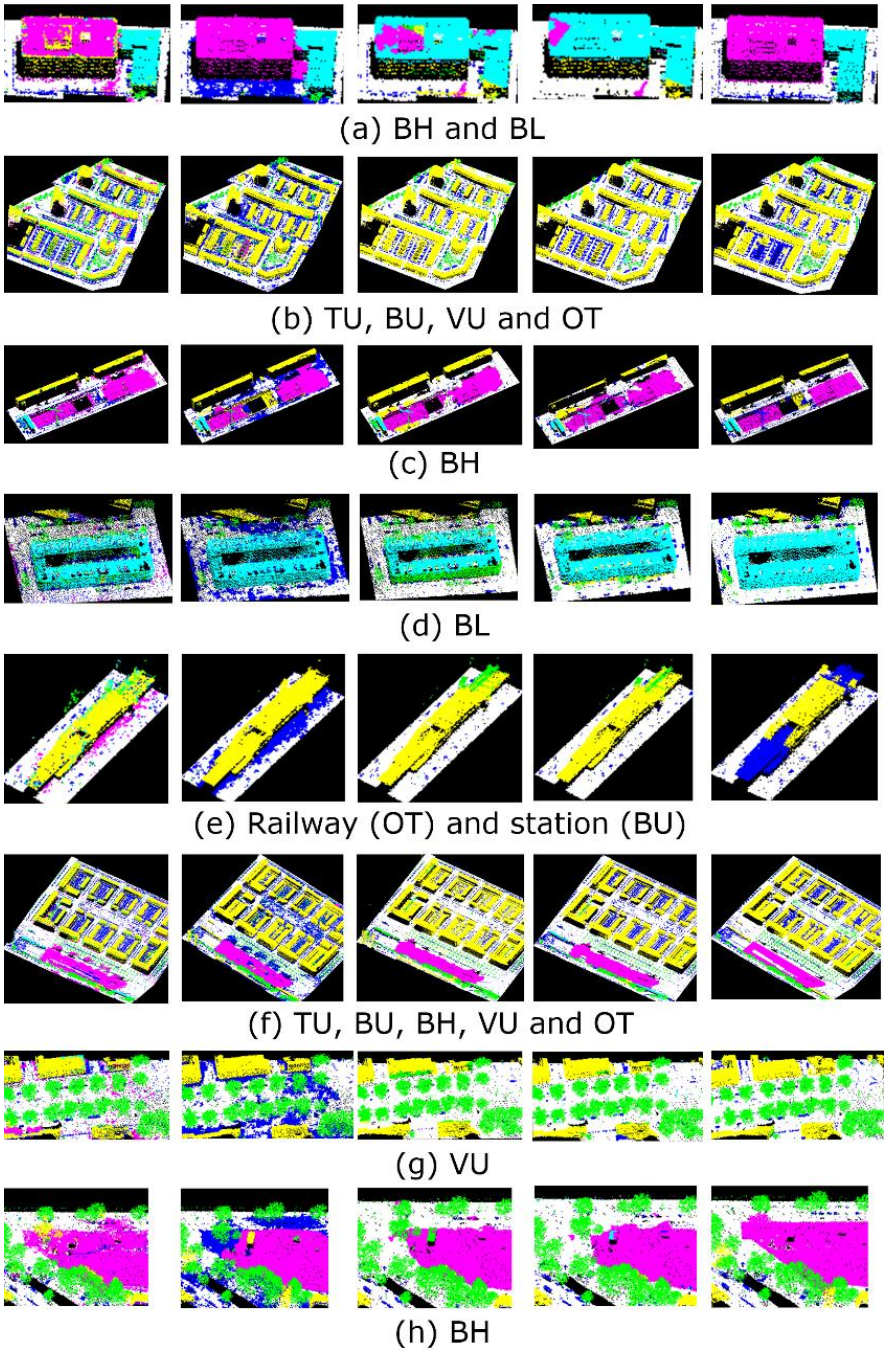



***Figure 5.11:*** *Eight examples selected from the testing results. In each example from left to right: SCD, OBCD, SiamPointNet++(SSG), SiamPointNet++(MSG), and ground truth.*

Figure 5.11(e) shows elevated railway track and station. The ground truth takes the station as BU and the track as OT. Results show that all the four methods mis-classify them into BU. This can be explained by that both unchanged buildings and elevated track are high-standing infrastructure with planar surfaces. Figure 5.11(f) shows that most points of TU, BU and BH are correctly detected in the mixed scene, while many OT are omitted by SSG. Figure 5.11(g) and (h) show that unchanged vegetation can be well detected by the four methods. In Figure 5.11(h), the boundary of heightened building is irregular from the four methods. The reason might be that the heightened building is close to vegetation and their boundaries are adjacent in some locations, which causes confusion in the change map.

Lastly, the change maps by SCD in Figure 5.11(e) and (g) show that many TU points are mis-classified into BH; The change maps by OBCD from Figure 5.11(d) to (h) all show that some TUs are mis-classified into OTs. These may be caused by the dense matching errors. Namely, the DIM point heights are higher than the true object heights, so the methods output false changes.

## 5.6 Conclusions

We proposed a method to combine the tasks of semantic segmentation and change detection for multimodal point clouds. The proposed SiamPointNet++(MSG) network learns both intra-epoch and inter-epoch features in a hierarchical manner. The contextual information from multiple scales is aggregated for robust inference. The method is compared with supervised change detection (SCD), object-based change detection (OBCD), and SiamPointNet++(SSG) to validate its performance. The proposed method achieves the highest mIoU of 68.07%, while its overall accuracy of 91.06% is very close to the highest score achieved by SSG.

There are two advantages with our method: Firstly, this supervised method requires only a little human intervention. The point clouds from two epochs are fed into the network after some initial denoising. In contrast, SCD requires to set many parameters based on empirical tests and prior knowledge for feature extraction. OBCD also requires to make sophisticated rules and set parameters for object extraction. In object-based change detection, the errors from object detection are propagated to the following change detection. Secondly, SiamPointNet++ learns intra-epoch and inter-epoch features together by MLPs. Intra-epoch features are extracted by multi-scale grouping so that information from a large context can be embedded; Inter-epoch features are extracted by Conjugated Ball Sampling. Conjugated Ball Sampling guarantees that the compared features from the two epochs are extracted from the same location.

Then features from inter-epochs are concatenated and further processed by vanilla PointNet++ for change detection.

However, our method still presents some limitations. The results are largely dependent on the quality and resolution of the input data. The method cannot handle with small objects or complicated situations where BH and BL are adjacent to each other. When building changes and vegetation are adjacent, the resulting boundary between different objects may get confused. In addition, sharp boundaries for the BH class are hard to obtain from our method, since the boundaries of new buildings are determined in the noisy DIM data and delineated on the ALS terrain.

Future work can be performed in the following aspects: (1) More training samples of BH and BL may be added to improve the diversity of training set. (2) The proposed SiamPointNet++ model takes only point coordinates as input. The spectral features within each DIM point may be added to the Siamese model to improve its performance. (3) This work takes the naive PointNet++ as the backbone of our architecture to demonstrate. However, many new models have been proposed for feature extraction from point clouds, which demonstrate better performance than PointNet++, such as PointCNN (Li et al., 2018), PointConv (Wu et al., 2019), KPConv (Thomas, et al., 2019) or RandLA-Net (Hu et al., 2020). Future work may take these new models as backbone for feature extraction and evaluate their performance. (4) This work only detects changes in buildings. If we also take changes in terrain and vegetation into consideration, the number of change labels would be quadratic with the number of land cover classes. The possible change types between different land cover classes would become more complicated. Certain new confusions may appear, such as the confusion between a new tree and a new building. The change types with small training samples might be hard to recognize by our method, e.g. a tree changed to a building.

# Chapter 6 – Synthesis

## 6.1 Conclusions per objective

The main goal of this thesis was to assess the quality of photogrammetric point clouds and detect changes between them and ALS data. The goal was achieved by evaluating the quality of DTMs, point clouds and DSMs generated from aerial imagery, and developing two change detection methods.

Overall, this thesis integrates knowledge from remote sensing, ALS, photogrammetry, computer vision and machine learning during the complete research process. We illustrate how the techniques from other domains can be combined to solve our technique questions. The work of the thesis can be divided into two parts: In part one (Chapter 2 and chapter 3), we evaluate the quality of photogrammetric products and investigate their potential for change detection. In part two (Chapter 4 and chapter 5), we develop two methods for change detection that meet different application requirements.

Our contributions are made on four aspects. The conclusions related to each contribution and the connections among chapters are given below.

**(1) Evaluation of the quality of dense matching point clouds and DSMs**

This contribution is addressed in chapter 2. Firstly, to investigate the potential of using point clouds derived by dense matching for change detection, we propose a framework for evaluating the quality of 3D point clouds and DSMs generated by dense image matching. Our evaluation framework based on a large number of square patches reveals the distribution of dense matching errors in a whole photogrammetric block. Robust quality measures are proposed to represent the dense matching accuracy and precision quantitatively. The overall mean offset to the reference is 0.1 GSD; the maximum mean deviation reaches 1.0 GSD. Given that the GSD of our data set for change detection is 10 cm, this indicates that our change detection methods can never handle with object sizes or object changes that are smaller 10 cm. Considering dense matching noise and possible blunders, the allowed change detection scale might be rougher.

Generally, the distribution of dense matching errors is homogenous in the whole block and close to a normal distribution based on many patch-based samples. However, in some locations, especially along narrow alleys, the mean deviations may get worse. The point clouds in those regions get less accurate because there are usually less visible image rays on the ground or the image contrast is poor. In addition, the profiles of ALS points and DIM points reveal that the DIM profile fluctuates around the ALS profile.

We also find that when oblique images are used in dense matching together with nadir images, the accuracy of DIM point cloud improves, and the noise level decreases on smooth ground areas. Therefore, we use the point clouds generated from nadir and oblique imagery for change detection. When many GCPs with high weights are employed in bundle adjustment, the BBA network may become overfitted, which is reflected in the inhomogeneous distribution of the patch-based DIM errors.

The allowed scale of change detection depends on the accuracy and noise level of input data. Evaluation of point clouds indicates that the mean deviation between ALS data and DIM data are better than 1 GSD. This finding helps to set the threshold for surface differencing. Point cloud differencing with a threshold of 1~1.5 GSD may allow us to localize some initial changed objects. The change maps may contain false alarms because mean deviations may get worse in some locations, for example along narrow alleys or over tree canopy. The coarse change differencing map can be refined to get fine change map.

**(2) Evaluation of filtering algorithms and DTMs derived from DIM points.**

This contribution is addressed in chapter 3. Firstly, we propose a method to evaluate whether the standard LiDAR filters can be used to filter dense matching points in order to derive accurate DTMs. To conclude, filtering results on the homogeneous ground and grassland show that the filtering performance depends on the noise level and scene complexity. LASground is verified to be relatively robust to random noise. However, filtering algorithms may only select the lower points as ground points in case of a large amount of noise. In addition, artefacts and blunders may appear in the dense matching points due to low image contrast or poor texture (e.g. in the shadow, along the narrow street, etc.). Filtering results on a city block show that LASground performs well on the grassland, along bushes and around individual trees if the point cloud is sufficiently precise.

Secondly, we use a ranking filter to process the DIM point cloud before LASground filtering. After processing with the ranking filter, LASground will identify fewer but more reliable ground locations. However, a ranking filter also eliminates ground details so some small objects on the terrain will be filtered out. Therefore, pre-processing DIM data with a ranking filter before change detection might be a necessary step to filter out the noise. In Chapter 5, we propose a Thickness-Adaptive Denoising method for DIM data pre-preprocessing based on this finding. This basic idea of Thickness-Adaptive Denoising is to maintain the skeleton points and filter our noisy points, which is similar to the ranking filter.

Finally, the DTMs derived from DIM data are evaluated quantitatively based on patch-based measures. We evaluate the quality from dense matching software SURE and Pix4d. The vertical accuracy of SURE point cloud on the ground is better than that of the Pix4D point cloud. We select Pix4D point clouds for change detection since their vertical accuracy is within 1.5 GSD which is acceptable, and its noise level is low. Although the vertical accuracy of SURE point clouds is comparatively high, more data gaps are found in the point clouds, which is less suitable for change detection. In addition, the errors on the grassland are more severe than the errors on the paved ground. This indicates that change detection between ALS data and DIM data on the smooth surface like terrain or building roof should be easier than on the vegetation since the height representation by DIM data is more accurate and precise on smooth surfaces.

**(3) Change detection and delineation between multimodal point clouds**

This contribution is addressed in chapter 4. Based on the previous findings in DIM data quality, we propose a method to detect building changes between ALS points and DIM points. Firstly, the DSM difference map generated from the ALS points and DIM points is concatenated with the orthoimages. The multimodal data are normalized to feed into a pseudo-Siamese Neural network for change detection. Then, the changed objects are delineated through per-pixel classification and artefact removal.

Results show that the proposed pseudo-Siamese Neural network can cope with the DIM errors and output plausible change detection results. Although the point cloud from dense matching is not as good as ALS points, the spectral and textural information provided by the orthoimages serve as a supplement, which leads to relatively satisfactory change delineation results.

This method disassembles the complicated multimodal change detection problem into three binary classification problems. They are solved by a light-weighted CNN model and two Random Forest classifiers, which require less hyper-parameters and prior knowledge compared to the change detection method used by (Du et al., 2016). Even though some training samples are required for the classifiers, this supervised method does not need to design classification rules manually.

To conclude, this is a "coarse-to-fine" method to detect building changes, which contains a change detection module and a change delineation module. The change detection module based on a pseudo-Siamese CNN can quickly localize the changes and generate coarse change maps, which might be used in the application of emergency responses such as aerial reconnaissance and

supervision of illegal construction. In contrast, change delineation can be used in precise mapping of change boundaries. It is applied only to the boundary pixels, which largely reduces the computational load.

**(4) Combination of semantic segmentation and change detection**

This contribution is addressed in chapter 5. Taking a step back, chapter 4 proposes a change detection method between multimodal point clouds. This derived change map is 2D instead of 3D, and the method is divided into two separate modules. In this chapter, we aim to design a more direct method for "end-to-end" change detection. Considering that the tasks of semantic segmentation and change detection are correlated, this method combines the two tasks in one framework.

The method outputs a pointwise joint label for each ALS point. If an ALS point is unchanged, assign it with a *semantic label*; If an ALS point is changed, assign it with a *change label*. The SS and CD information are included in the joint labels with minimum information redundancy. This chapter brings our work one step forward towards the application of point cloud updating. Our method derives semantic labels and change labels for each ALS point. If they are changed, we replace these points with the neighboring DIM points so that up-to-date points are obtained.

The proposed SiamPointNet++ model can learn both intra-epoch and inter-epoch features. The previous Siamese network architecture usually takes two images as inputs; In contrast, our Siamese network takes unstructured point clouds from two epochs as inputs. Intra-epoch features are extracted at multiple scales to embed the local and global information. Inter-epoch features are extracted by Conjugated Ball Sampling (CBS) and concatenated to make change inference. Concerning other potential applications, this architecture may also be extended to other change detection tasks between point clouds from other platforms or in other modalities.

Experiments on the Rotterdam data set indicate that the method is effective in the combined tasks. The findings in chapter 2 and chapter 3 reflect the noise level and inaccuracy of the DIM data. In this chapter, a Thickness-Adaptive Denoising method is first proposed to unify the density of two types of unstructured point clouds before they are fed into a Siamese network. To conclude, the Siamese network is invariant to the permutation and noise of inputs and robust to the data difference between two epochs. Compared with sophisticated object-based methods, this method requires much less hyper-parameters and human intervention but achieves superior performance.

Concerning the requirements for model training, chapter 4 trains one light-weighted pseudo-Siamese network and two Random Forests, while chapter 5 trains one Siamese PointNet++ network which takes unstructured point clouds as input. Based on our experience, training the light-weighted models in chapter 4 requires less training samples and workload compared to training a model in chapter 5. Preparing training samples by manual labeling in 3D usually takes more effort than in 2D. The readers can select their method based on their requirements and available conditions.

## 6.2 Reflections and outlook

**Limitations of the DIM data quality**

Motivated by the need for detecting topographic changes and updating outdated point clouds, our thesis evaluates the point cloud quality generated by state-of-the-art dense matching algorithms and investigate the different factors influencing the DIM quality. It is not our focus to develop new dense matching algorithms by ourselves. Even though chapter 2 and chapter 3 have studied many factors that affect the DIM accuracy and noise level, the DIM quality is determined by a mixture of multiple factors, such as the image quality and overlapping rate during acquisition, accuracy of exterior orientation elements, GCP distribution and precision, dense matching methods, etc. This work reveals the gap between the DIM data quality and ALS data quality and gives suggestions on the photogrammetric quality control.

Fine-level change detection requires point clouds of high quality. The thesis only studies the changes to buildings because the point clouds quality generated by the current state-of-the-art dense matching methods still does not permit fine-grained change detection, such as changes to vegetation or traffic poles.

**Limitations of the study data amount**

Apart from the limitations of dense matching quality, one major limitation is the lack of diverse experimental data and reference labels. We only have two study areas, i.e. the Enschede study area and Rotterdam study area. There are only four building changes and one terrain change within the Enschede data, which are not sufficient for experimental usage. We need more data sets including quite many diverse object changes to validate the proposed methods. To validate the generalizability of our models, it is better to test the model on some study areas from different regions of the country.

In addition, labeling the reference data is also time-consuming and labor-intensive. In chapter 4, labeling the Rotterdam point clouds of 15.4 $km^2$ took two weeks by comparing the point clouds from two epochs and marking the labels on the orthoimages. In chapter 5, labeling the combined labels for semantic segmentation and change detection in 3D space is more complicated. For future work, weakly supervised models or active learning might be used to alleviate the dependance on large training samples.

Training a deep model relies on many diverse samples. In the future work, more BH and BL samples will be added to the training data by data augmentation to improve the model generalizability. Model testing should also be implemented on a city level (such as 25 $km^2$) instead of a local region, to validate the performance. A large study area should contain more building changes, vegetation changes, and terrain changes, even though manual labeling and data processing would take much more computational effort.

**Reflections on the proposed change detection methods**

Chapter 4 proposes a patch-based method for multimodal change detection. The method requires pre-processing to convert point clouds of two epochs and orthoimages into registered patches. Although the PSI-DC model is light-weighted and works satisfactorily for the problem at hand, the pre-processing work is relatively time-consuming and labor-intensive. In addition, the change detection results are patch-based instead of point-based. Change delineation is required to make further inference along the building boundaries in order to derive sharp change boundaries. Therefore, the framework is relatively complicated which contains two sub-steps.

Chapter 5 proposes a Siamese network architecture to combine the tasks of semantic segmentation and change detection. The results are largely dependent on the quality and resolution of the input data. It is still hard to separate small objects or complicated situations where heightened buildings and lowered buildings are adjacent to each other. When building changes and vegetation are adjacent, the resulting boundary between different objects may also be fuzzy. In addition, sharp boundaries of heightened buildings are hard to obtain from our method since they are determined by the noisy DIM point clouds.

Both methods are based on MLPs which depend on large amount of training data. As claimed earlier, it is difficult to find many changed samples in different categories from our study area. It would be more interesting and useful to investigate the changes to vegetation, terrain or other land cover types. Since we cannot prepare sufficient training samples for those land cover types, the proposed models are not capable of detecting changes in those types either.

**Comparison with the state-of-the-art methods**

The proposed framework verifies that SiamPointNet++(MSG) are capable of learning deep features from two types of point clouds. The features can be applied in tasks such as change detection or potential point cloud updating. However, the proposed methods rely too much on the dense matching quality. Zhou et al. (2020) applies a different solution to detect changes between ALS data and DIM data. First, LiDAR-guided edge-aware dense matching is used to derive accurate partial changes. Then hierarchical dense matching is employed to derive complete changes and update 3D information. In contrast, we first obtain point clouds covering the complete study area. Then we detect changes by comparing DIM point cloud to the ALS point cloud. Although a complete DIM point cloud can be obtained from our workflow, their coarse-to-fine method is relatively efficient.

Comparing with object-based change detection (OBCD) and supervised change detection (SCD) (Tran et al., 2018), the SiamPointNet++(MSG) method requires only a little human intervention but achieves superior performance. The point clouds from two epochs are fed into the network after some initial denoising. In contrast, SCD requires setting many parameters based on empirical tests and prior knowledge for feature extraction. OBCD also requires making sophisticated rules and setting parameters for object extraction. In addition, the errors from object detection are inevitably propagated to the following change detection in the OBCD method.

In terms of feature aggregation, the multi-scale features are extracted in wider context by our method than those from OBCD or SCD. The SiamPointNet++(MSG) learns intra-epoch and inter-epoch features by MLPs. Intra-epoch features are extracted by multi-scale grouping so that information from large context can be embedded; Inter-epoch features are extracted by Conjugated Ball Sampling. Conjugated Ball Sampling guarantees that the compared features from the two epochs are extracted from the same location. Then features from inter-epochs are concatenated and further processed by vanilla PointNet++ as a change detection module. In contrast, the OBCD relies on surface-based growing and connected component analysis to group information from wide context in an implicit way. The SCD extracts features from local neighborhood within certain radius, which brings only limited context.

In the past three years, Siamese networks have been used to learn 3D shape descriptors from a pair of point clouds. Shen et al. (2018) propose a Siamese Network to extract feature descriptors from traffic facilities. The loss function of Euclidean distance is minimized to guarantee the similarity of the two inputs. Zhou et al. (2020) propose SiamesePointNet to extract shape descriptors from

a pair of point clouds. The Convolution-Deconvolution architecture with N-tuple loss is verified to be robust to the geometric variations of 3D shapes.

Regarding with Siamese MLP architectures for point cloud change detection, de Gélis et al. (2021) propose Siamese Networks to detect topographic changes between LiDAR points. They also compare their method with our patch-based change detection from chapter 4. Due to lack of true training data, their experiments are implemented on simulated data set. Nagy et al. (2021) propose ChangeGAN to detect changes in coarsely registered MLS point clouds in the street environment. Firstly, the point clouds are projected to 2D range images and fed into the Siamese architecture. The U-net component allows for extraction of multiscale features. The Spatial Transformation Network (STN) component allows for optimal transformation estimation.

The deep learning-based change detection algorithms can be modified in the following aspects: (1) Strictly speaking, the proposed models for change detection in chapter 4 and chapter 5 are still not end-to-end owing to pre-processing. In future work, the model may take raw point clouds from two epochs as inputs, or directly take ALS points and airborne images as inputs. (2) The inner mechanisms of Siamese Neural Networks are still difficult to track or explain. If the models are interpretable, the proposed models may get easier to be used by mapping agencies or industries. Currently, the hyper-parameters such as number of convolutional layers or fully connected layers, the locations of Batch Normalization or drop-out layers are largely determined by the feed-up from the model performance. The process of developing a robust deep learning model is still not scientific but based on experience. (3) The two deep learning-based change detection methods in chapter 4 and chapter 5 are both aimed to detect changes in buildings. If we also take terrain changes and vegetation changes into consideration, the change detection tasks would become more complicated. More training samples should be prepared for different change types. (4) Furthermore, some new architectures are proposed for feature learning from point clouds such as KPconv (Thomas, et al., 2019), RandLA-Net (Hu et al., 2020), Point Transformers (Zhao et al., 2021) or geometry-attentional network (Li et al., 2020). These architectures become more robust due to effective fusion of multimodal and multiscale features. In addition, weakly-supervised classification is being applied to point cloud semantic segmentation to lower down the dependency on the training samples (Lin et al., 2020). It would also be valuable to investigate weakly-supervised change detection methods based on fewer training data.

In terms of detecting other types of topographic changes, Hirt et al. (2021) detect tree changes in the city of Munich between MLS data sets from 2016 and 2018, respectively. The two point clouds are already registered to each other. Firstly, individual trees are extracted from the MLS points by instance

segmentation. The tree parameters of height and diameter at breast height are derived. Tree changes are classified into *unchanged*, *removed* or *newly-planted* based on the status of occupancy voxel grids. Their method belongs to post-classification analysis which is largely dependent on the instance segmentation results. Its workflow is similar to our comparative method OBCD in chapter 5. We applied Neural Networks as our classifiers in both chapter 4 and chapter 5 to develop a direct solution for change detection with less human intervention.

Concerning the limitations of our method, we have multi-view imagery, DSMs and point clouds as the new data for change detection. However, it is still difficult to make full use of all the information. In chapter 4, the orthoimages are concatenated with DSMs; In chapter 5, the spectral features are not employed. The geometric relations among multi-view images and the spectral information are far from being fully exploited. In future work, we look forward to deeper fusion of spectral and geometric information for change detection.

Our work was initially motivated by the needs for point cloud updating in the Dutch mapping agency. However, point cloud updating consists of more than data quality evaluation and 3D change detection. Point cloud updating is more related to the user requirements. It is necessary to know which topographic objects should be updated and which quality level of point cloud can be used for updating. Specifically, the new DIM point clouds used for updating should be similar to the outdated data in terms of accuracy, noise level and density. For future work, we recommend to join researchers, data users and policy makers from mapping agencies to push forward the research on change detection and point cloud updating.

# Bibliography


Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Süsstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. PAMI, 34(11), pp.2274-2282.

Actueel Hoogtebestand Nederland. https://www.ahn.nl/. (Accessed on 27/06/2021).

Ali-Sisto, D. and Packalen, P., 2016. Forest change detection by using point clouds from dense image matching together with a LiDAR-derived terrain model. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10(3), pp.1197-1206.

Audebert, N., Le Saux, B. and Lefèvre, S., 2018. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. ISPRS Journal of Photogrammetry and Remote Sensing, 140, pp.20-32.

Awrangjeb, M., Fraser, C.S. and Lu, G., 2015. BUILDING CHANGE DETECTION FROM LIDAR POINT CLOUD DATA BASED ON CONNECTED COMPONENT ANALYSIS. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 2.

Axelsson, P., 2000. DEM Generation from Laser Scanner Data Using adaptive TIN Models. Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci. 33, 110-117.

Baltsavias, E.P., 1999. A comparison between photogrammetry and laser scanning. ISPRS J. Photogram. Remote Sens. 54(2), 83-94.

Basgall, P.L., Kruse, F.A. and Olsen, R.C., 2014. Comparison of lidar and stereo photogrammetric point clouds for change detection. In Laser Radar Technology and Applications XIX; and Atmospheric Propagation XI. International Society for Optics and Photonics, Vol. 9080, pp. 90800R.

Beumier, C. and Idrissa, M., 2016. Digital terrain models derived from digital surface model uniform regions in urban areas. Int. J. of Remote Sens., 37(15), pp. 3477-3493.

Blomley, R., Weinmann, M., Leitloff, J. and Jutzi, B., 2014. Shape distribution features for point cloud analysis-a geometric histogram approach on multiple scales. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(3), p.9.

Boulch, A., Le Saux, B. and Audebert, N., 2017. Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks. 3DOR, 2, p.7.

Breiman, L., 2001. Random forests. Machine learning, 45(1), pp. 5-32.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. and Shah, R., 1994. Signature verification using a "Siamese" time delay Neural Network. In Advances in neural information processing systems. pp. 737-744.

Buda, M., Maki, A. and Mazurowski, M.A., 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 106, pp.249-259.

Cavegn, S., Haala, N., Nebiker, S., Rothermel, M. and Tutzauer, P., 2014. Benchmarking high density image matching for oblique airborne imagery. Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci. 40(3), 45-52.

Champion, N., 2007. 2D building change detection from high resolution aerial images and correlation digital surface models. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(3/W49A), pp.197-202.

Chehata, N., Guo, L. and Mallet, C., 2009, September. Airborne lidar feature selection for urban classification using random forests. In Laserscanning.

Chen, L.C. and Lin, L.J., 2010. Detection of building changes from aerial images and light detection and ranging (LIDAR) data. Journal of Applied Remote Sensing, 4(1), pp.041870.

Chen, Q., Wang, H., Zhang, H., Sun, M. and Liu, X., 2016. A point cloud filtering approach to generating DTMs for steep mountainous areas and adjacent residential areas. Remote Sens., 8(1), pp. 71.

Chen, Z., Gao, B. and Devereux, B., 2017. State-of-the-Art: DTM Generation Using Airborne LIDAR Data. Sensors, 17(1), pp.150.

Choi, K., Lee, I. and Kim, S., 2009. A feature based approach to automatic change detection from LiDAR data in urban areas. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 18, pp.259-264.

Chopra, S., Hadsell, R. and LeCun, Y., 2005, June. Learning a similarity metric discriminatively, with application to face verification. CVPR. Vol. 1, pp. 539-546.

Cyclomedia, https://www.cyclomedia.com. (Accessed on 27/06/2021)

Daudt, R.C., Le Saux, B. and Boulch, A., 2018a. Fully convolutional siamese networks for change detection. In 2018 25th IEEE International Conference on Image Processing (ICIP), IEEE, pp. 4063-4067.

Daudt, R.C., Le Saux, B., Boulch, A. and Gousseau, Y., 2018b. Urban change detection for multispectral earth observation using convolutional neural networks. In International Geoscience and Remote Sensing Symposium (IGARSS).

Debella-Gilo, M., 2016. Bare-earth extraction and DTM generation from photogrammetric point clouds including the use of an existing lower-resolution DTM. Int. J. of Remote Sens., 37(13), pp. 3104-3124.

De Gélis, I., Lefèvre, S. and Corpetti, T., 2021. 3d urban change detection with point cloud siamese networks. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 43, pp.879-886.

Dini, G. R., Jacobsen, K., Rottensteiner, F., Al Rajhi, M., & Heipke, C. (2012). 3D Building Change Detection using High Resolution Stereo Images and a GIS Database. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXIX-B7(September), 299–304. Retrieved from http://www.int-arch-

photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B7/299/2012/isprsarchives-XXXIX-B7-299-2012.pdf

Dong, W., Lan, J., Liang, S., Yao, W. and Zhan, Z., 2017. Selection of LiDAR geometric features with adaptive neighborhood size for urban land cover classification. Int. J. Appl. Earth Obs. Geoinf. 60, 99-110.

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), pp.1030.

Eitel, A., Springenberg, J.T., Spinello, L., Riedmiller, M. and Burgard, W., 2015, September. Multimodal deep learning for robust rgb-d object recognition. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference. IEEE, pp. 681-687.

Elberink, S.O. and Vosselman, G., 2012. Entities and features for classifcation of airborne laser scanning data in urban area. ISPRS Annals Photogramm Remote Sens Spat Information Sciences, pp.257-262.

Frontoni, E., Khoshelham, K., Nardinocchi, C., Nedkov, S. and Zingaretti, P., 2008. Comparative analysis of automatic approaches to building detection from multi-source aerial data. Proceedings GEOBIA 2008-Pixels, Objects, Intelligence GEOgraphic Object Based Image Analysis, Calgary, Canada, 5-8 August 2008; IAPRS, XXXVIII (4/C1), 2008.

Furukawa, Y. and Ponce, J., 2010. Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell., 32(8), pp.1362-1376.

Fuse, T. and Yokozawa, N., 2017. Development of a Change Detection Method with Low-Performance Point Cloud Data for Updating Three-Dimensional Road Maps. ISPRS International Journal of Geo-Information, 6(12), p.398.

Gehrke, S., Morin, K., Downey, M., Boehrer, N. and Fuchs, T., 2010. Semi-global matching: An alternative to LIDAR for DSM generation. Proceedings of the 2010 Canadian Geomatics Conf. and Symp. of Commission I. 1-6.

Gerke, M., & Xiao, J., 2013. Supervised and unsupervised MRF based 3D scene classification in multiple view airborne oblique images. ISPRS conference; CMRT13 - City Models, Roads and Traffic 2013,. AGU Fall Meeting Abstracts.

Gerke, M., Nex, F., Remondino, F., Jacobsen, K., Kremer, J., Karel, W., Huf, H. and Ostrowski, W., 2016. Orientation of oblique airborne image sets-experiences from the ISPRS/EUROSDR benchmark on multi-platform photogrammetry. Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci. 41, 185-191.

Gevaert, C.M., Persello, C., Sliuzas, R. and Vosselman, G., 2016. Classification of informal settlements through the integration of 2D and 3D features extracted from UAV data. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 3, p.317.

Gevaert, C.M., Persello, C., Sliuzas, R. and Vosselman, G., 2017. Informal settlement classification using point-cloud and image-based features from UAV data. ISPRS Journal of Photogrammetry and Remote Sensing, 125, pp.225-236.

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

Girardeau-Montaut, D., Roux, M., Marc, R. and Thibault, G., 2005. Change detection on points cloud data acquired with a ground laser scanner. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(part 3), p.W19.

Gong, M., Zhan, T., Zhang, P. and Miao, Q., 2017. Superpixel-based difference representation learning for change detection in multispectral remote sensing images. IEEE Transactions on Geoscience and Remote Sensing, 55(5), pp. 2658-2673.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. Deep learning (Vol. 1). Cambridge: MIT press.

Guinard, S. and Landrieu, L., 2017. Weakly supervised segmentation-aided classification of urban scenes from 3D LiDAR point clouds. In ISPRS Workshop 2017.

Guo, L., Chehata, N., Mallet, C. and Boukir, S., 2011. Relevance of airborne lidar and multispectral image data for urban scene classification using Random Forests. ISPRS Journal of Photogrammetry and Remote Sensing, 66(1), pp.56-66.

Gupta, S., Girshick, R., Arbeláez, P. and Malik, J., 2014, September. Learning rich features from RGB-D images for object detection and segmentation. In European conference on computer vision (pp. 345-360). Springer, Cham.

Haala, N. and Rothermel, M., 2012. Dense multi-stereo matching for high quality digital elevation models. Photogrammetrie-Fernerkundung-Geoinformation. 4, 331-343.

Haala, N., 2015. Dense Image Matching Final Report, EuroSDR Official Publication, pp .115-145., (JANUARY 2014).

Haala, N., Hastedt, H., Wolf, K., Ressl, C. and Baltrusch, S., 2010. Digital photogrammetric camera evaluation – Generation of digital elevation models. Photogrammetrie-Fernerkundung-Geoinformation. 2, 99-115.

Hackel, T., Wegner, J.D. and Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 3, pp.177-184.

Haralick, R.M. and Shapiro, L.G., 1992. Computer and robot vision. Reading: Addison-wesley, Vol. 1, pp. 28-48.

Hazirbas, C., Ma, L., Domokos, C. and Cremers, D., 2016, November. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn

architecture. In Asian conference on computer vision (pp. 213-228). Springer, Cham.

He, H., Chen, M., Chen, T. and Li, D., 2018. Matching of Remote Sensing Images with Complex Background Variations via Siamese Convolutional Neural Network. Remote Sensing, 10(2), pp.355.

Hebel, M., Arens, M. and Stilla, U., 2013. Change detection in urban areas by object-based analysis and on-the-fly comparison of multi-view ALS data. ISPRS Journal of Photogrammetry and Remote Sensing, 86, pp.52-64.

Hirschmüller, H., 2008. Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. 30(2), 328-341.

Hirt, P.R., Xu, Y., Hoegner, L. and Stilla, U., 2021. Change Detection of Urban Trees in MLS Point Clouds Using Occupancy Grids. PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science, pp.1-18.

Hobi, M.L. and Ginzler, C., 2012. Accuracy assessment of digital surface models based on WorldView-2 and ADS80 stereo remote sensing data. Sensors. 12(5), 6347-6368.

Höhle, J. and Höhle, M., 2009. Accuracy assessment of digital elevation models by means of robust statistical methods. ISPRS J. Photogram. Remote Sens. 64(4), 398-406.

Holland, D.A., Boyd, D.S. and Marshall, P., 2006. Updating topographic mapping in Great Britain using imagery from high-resolution satellite sensors. ISPRS Journal of Photogrammetry and Remote Sensing, 60(3), pp. 212-223.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N. and Markham, A., 2020. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11108-11117.

Hu, X. and Yuan, Y., 2016. Deep-Learning-Based Classification for DTM Extraction from ALS Point Cloud. Remote Sens., 8(9), pp.730.

Huang, J. and You, S., 2016. Point cloud labeling using 3d convolutional neural network. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 2670-2675). IEEE.

Huang, X., Wen, D., Li, J. and Qin, R., 2017. Multi-level monitoring of subtle urban changes for the megacities of China using high-resolution multi-view satellite imagery. Remote sensing of environment, 196, pp.56-75.

Jaud, M., Passot, S., Le Bivic, R., Delacourt, C., Grandjean, P. and Le Dantec, N., 2016. Assessing the accuracy of high resolution digital surface models computed by PhotoScan® and MicMac® in sub-optimal survey conditions. Remote Sens. 8(6), 465-482.

Jung, F., 2004. Detecting building changes from multitemporal aerial stereopairs. ISPRS Journal of Photogrammetry and Remote Sensing, 58(3-4), pp.187-201.

Kim, C., Kim, B. and Kim, H., 2013. 4D CAD model updating using image processing-based construction progress monitoring. Automation in Construction, 35, pp.44-52.

Kim, K. and Shan, J., 2011. Adaptive morphological filtering for DEM generation. In IEEE Geoscience and Remote Sensing Symposium (IGARSS), pp. 2539-2542.

Kingma, D.P., Ba, J.L., 2014. Adam: A method for stochastic optimization, in: ArXiv Preprint ArXiv:1412.6980.

Koch, G., Zemel, R. and Salakhutdinov, R., 2015. Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop (Vol. 2).

Kraus, K. and Pfeifer, N., 1998. Determination of terrain models in wooded areas with airborne laser scanner data. ISPRS J. Photogram. Remote Sens., 53(4), pp.193-203.

Kraus, K., Karel, W., Briese, C. and Mandlburger, G., 2006. Local accuracy measures for digital terrain models. The Photogrammetric Record. 21(116), 342-354.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097-1105.

Landrieu, L. and Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4558-4567).

Leberl, F., Irschara, A., Pock, T., Meixner, P., Gruber, M., Scholz, S. and Wiechert, A., 2010. Point clouds: Lidar versus 3D vision. Photogram. Eng. Remote Sens. 76(10), 1123-1134.

Lefèvre, S., Tuia, D., Wegner, J.D., Produit, T. and Nassaar, A.S., 2017. Toward seamless multiview scene analysis from satellite to street level. Proceedings of the IEEE, 105(10), pp.1884-1899.

Lei, H., Akhtar, N. and Mian, A., 2019. Octree guided cnn with spherical kernels for 3d point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9631-9640).

Li, G., Guo, J., Tang, X., Ye, F., Zuo, Z., Liu, Z., Chen, J. and Xue, Y., 2020. Preliminary quality analysis of GF-7 satellite laser altimeter full waveform data. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 43, pp.129-134.

Li, W., Wang, F.D. and Xia, G.S., 2020. A geometry-attentional network for ALS point cloud classification. ISPRS Journal of Photogrammetry and Remote Sensing, 164, pp.26-40.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X. and Chen, B., 2018. Pointcnn: Convolution on x-transformed points. Advances in neural information processing systems, 31, pp.820-830.

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

Lian, Y., Feng, T. and Zhou, J., 2019, July. A dense Pointnet++ architecture for 3D point cloud semantic segmentation. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium (pp. 5061-5064). IEEE.

Lin, X. and Zhang, J., 2014. Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments. Remote Sens., 6(2), pp.1294-1326.

Lin, Y., Vosselman, G., Cao, Y. and Yang, M.Y., 2020. Active and incremental learning for semantic ALS point cloud segmentation. ISPRS Journal of Photogrammetry and Remote Sensing, 169, pp.73-92.

Liu, Y., Piramanayagam, S., Monteiro, S.T. and Saber, E., 2017. Dense semantic labeling of very-high-resolution aerial imagery and LiDAR with fully-convolutional neural networks and higher-order CRFs. In Proceedings of the IEEE Conference on Compu

Liu, Z., Tang, H., Lin, Y. and Han, S., 2019. Point-voxel cnn for efficient 3d deep learning. arXiv preprint arXiv:1907.03739.

Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. CVPR, pp. 3431-3440.

Lu, D. and Weng, Q., 2007. A survey of image classification methods and techniques for improving classification performance. International Journal of Remote Sensing, 28(5), 823–870.

Lu, D., Mausel, P., Brondizio, E. and Moran, E., 2004. Change detection techniques. International journal of remote sensing, 25(12), pp.2365-2401.

Lu, Y. and Rasmussen, C., 2012, October. Simplified Markov random fields for efficient semantic labeling of 3D point clouds. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 2690-2697). IEEE.

Luo, W., Schwing, A.G. and Urtasun, R., 2016. Efficient deep learning for stereo matching. CVPR, pp. 5695-5703.

Malpica, J.A., Alonso, M.C., Papí, F., Arozarena, A. and Martínez De Agirre, A., 2013. Change detection of buildings from satellite imagery and lidar data. International Journal of Remote Sensing, 34(5), pp.1652-1675.

Maltezos, E. and Ioannidis, C., 2015. AUTOMATIC DETECTION OF BUILDING POINTS FROM LIDAR AND DENSE IMAGE MATCHING POINT CLOUDS. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 2.

Maltezos, E., Kyrkou, A. and Ioannidis, C., 2016. LiDAR vs dense image matching point clouds in complex urban scenes. Proc. SPIE 9688. Fourth Int. Conf. on Remote Sens. and Geoinform. of the Environ. 9688, 1-10.

Mandlburger, G., Wenzel, K., Spitzer, A., Haala, N., Glira, P. and Pfeifer, N., 2017. Improved topographic models via concurrent airborne lidar and dense image matching. ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci. IV-2/W4, 259-266.

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, pp.158-172.

Martinovic, A., Knopp, J., Riemenschneider, H. and Van Gool, L., 2015. 3d all the way: Semantic segmentation of urban scenes from start to end in 3d. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4456-4465).

Matikainen, L., Hyyppä, J. and Kaartinen, H., 2004. Automatic detection of changes from laser scanner and aerial image data for updating building maps. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 35, pp. 434-439.

Matikainen, L., Hyyppä, J. and Litkey, P., 2016. MULTISPECTRAL AIRBORNE LASER SCANNING FOR AUTOMATED MAP UPDATING. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 41.

Maturana, D. and Scherer, S., 2015, September. Voxnet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 922-928). IEEE.

Meng, X., Currit N., and Zhao K., 2010. Ground Filtering Algorithms for Airborne LiDAR Data: A Review of Critical Issues. Remote Sens., 2 (3), pp. 833–860.

Miller, D.R., Quine, C.P. and Hadley, W., 2000. An investigation of the potential of digital photogrammetry to provide measurements of forest characteristics and abiotic damage. Forest Ecology and Management, 135(1-3), pp.279-288.

Mou, L., Bruzzone, L. and Zhu, X.X., 2019. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. IEEE Transactions on Geoscience and Remote Sensing, 57(2), pp. 924-935.

Mou, L., Schmitt, M., Wang, Y. and Zhu, X.X., 2017, March. A CNN for the identification of corresponding patches in SAR and optical imagery of urban scenes. In Urban Remote Sensing Event (JURSE), 2017 Joint IEEE, pp. 1-4.

Mousa, A.K., Helmholz, P. and Belton, D., 2017. New DTM extraction approach from airborne images derived DSM. Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci., pp. 42.

Moussa, W., Wenzel, K., Rothermel, M., Abdel-Wahab, M. and Fritsch, D., 2013. Complementing TLS Point Clouds by Dense Image Matching. International J. of Heritage in the Digital Era. 2(3), 453-470.

Mura, M., McRoberts, R.E., Chirici, G. and Marchetti, M., 2015. Estimating and mapping forest structural diversity using airborne laser scanning data. Remote Sens. of Environment. 170, 133-142.

Murakami, H., Nakagawa, K., Hasegawa, H., Shibata, T. and Iwanami, E., 1999. Change detection of buildings using an airborne laser scanner. ISPRS Journal of Photogrammetry and Remote Sensing, 54(2-3), pp.148-152.

Nagy, B. and Benedek, C., 2019. 3D CNN-based semantic labeling approach for mobile laser scanning data. IEEE Sensors Journal, 19(21), pp.10034-10045.

Nebiker, S., Lack, N., & Deuber, M., 2014. Building Change Detection from Historical Aerial Photographs Using Dense Image Matching and Object-Based Image Analysis. Remote Sensing, 6(9), 8310–8336. http://doi.org/10.3390/rs6098310.

Nex, F., Gerke, M., Remondino, F., Przybilla, H.J., Bäumker, M. and Zurhorst, A., 2015. ISPRS benchmark for multi-platform photogrammetry. ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci. 2(3), 135-142.

nFrames. https://www.nframes.com/ (Accessed on 27/06/2021)

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, pp.152-165.

Nurminen, K., Karjalainen, M., Yu, X., Hyyppä, J. and Honkavaara, E., 2013. Performance of dense digital surface models based on image matching in the estimation of plot-level forest variables. ISPRS J. Photogram. Remote Sens. 83, 104-115.

Ojala, T., Pietikäinen, M. and Mäenpää, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI, (7), pp. 971-987.

Pang, S., Hu, X., Cai, Z., Gong, J. and Zhang, M., 2018. Building change detection from bi-temporal dense-matching point clouds and aerial images. Sensors, 18(4), p.966.

Perko, R., Raggam, H., Gutjahr, K.H. and Schardt, M., 2015. Advanced DTM generation from very high resolution satellite stereo images. ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci., 2(3), pp.165.

Pix4D. https://www.pix4d.com/ (Accessed on 27/06/2021)

Politz, F., Sester, M. and Brenner, C., 2020. GEOMETRY-BASED POINT CLOUD CLASSIFICATION USING HEIGHT DISTRIBUTIONS. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 5(2).

Politz, F., Sester, M. and Brenner, C., 2021. Building Change Detection of Airborne Laser Scanning and Dense Image Matching Point Clouds using Height and Class Information. AGILE: GIScience Series, 2, pp.1-14.

Poon, J., Fraser, C.S., Chunsun, Z., Li, Z. and Gruen, A., 2005. Quality assessment of digital surface models generated from IKONOS imagery. The Photogrammetric Record. 20(110), 162-171.

Priestnall, G., Jaafar, J. and Duncan, A., 2000. Extracting urban features from LiDAR digital surface models. Computers, Environment and Urban Systems, 24(2), pp.65-78.

Pu, S., Rutzinger, M., Vosselman, G. and Elberink, S.O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS Journal of Photogrammetry and Remote Sensing, 66(6), pp.S28-S39.

PyTorch. https://pytorch.org/ (Accessed on 27/06/2021)

Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M. and Guibas, L.J., 2016. Volumetric and multi-view cnns for object classification on 3d data. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5648-5656).

Qi, C.R., Su, H., Mo, K. and Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 652-660.

Qi, C.R., Yi, L., Su, H. and Guibas, L.J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413.

Qi, X., Liao, R., Jia, J., Fidler, S. and Urtasun, R., 2017. 3d graph neural networks for rgbd semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, pp. 5199-5208.

Qin, R. and Gruen, A., 2014. 3D change detection at street level using mobile laser scanning point clouds and terrestrial images. ISPRS Journal of Photogrammetry and Remote Sensing, 90, pp. 23-35.

Qin, R., 2014. An object-based hierarchical method for change detection using unmanned aerial vehicle images. Remote Sens. 6(9), 7911-7932.

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

Ramiya, A.M., Nidamanuri, R.R. and Krishnan, R., 2016. Object-oriented semantic labelling of spectral–spatial LiDAR point cloud for urban land cover classification and buildings detection. Geocarto International, 31(2), pp.121-139.

Rebolj, D., Babič, N.Č., Magdič, A., Podbreznik, P. and Pšunder, M., 2008. Automated construction activity monitoring system. Advanced engineering informatics, 22(4), pp.493-503.

Remondino, F., Nocerino, E., Toschi, I. and Menna, F., 2017. A critical review of automated photogrammetric processing of large datasets. ISPRS Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci. 42, 591-599.

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), pp. 144-166.

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pp. 91-99.

Ressl, C., Brockmann, H., Mandlburger, G. and Pfeifer, N., 2016. Dense image matching vs. airborne laser scanning–comparison of two methods for deriving terrain models. PFG Photogrammetrie, Fernerkundung, Geoinformation. 2, pp. 57-73.

Riegler, G., Osman Ulusoy, A. and Geiger, A., 2017. Octnet: Learning deep 3d representations at high resolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3577-3586).

Rizaldy, A., Persello, C., Gevaert, C.M. and Oude Elberink, S.J., 2018. FULLY CONVOLUTIONAL NETWORKS FOR GROUND CLASSIFICATION FROM LIDAR POINT CLOUDS. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4(2).

Rothermel, M. and Haala, N., 2011. Potential of dense matching for the generation of high quality digital elevation models. ISPRS Hannover Workshop for High-Resolution Earth Imaging for Geospatial Information. 331-343.

Rothermel, M., Wenzel, K., Fritsch, D. and Haala, N., 2012. SURE: Photogrammetric surface reconstruction from imagery. Proc. of LC3D Workshop. 1-9.

Rottensteiner, F., 2007. Building change detection from digital surface models and multi-spectral images. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS), 36(3), pp.145-150.

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 J. Photogram. Remote Sens. 93, 256-271.

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), pp.135-149.

Roynard, X., Deschaud, J.E. and Goulette, F., 2016, July. Fast and robust segmentation and classification for change detection in urban point clouds. In ISPRS 2016-XXIII ISPRS Congress.

Roynard, X., Deschaud, J.E. and Goulette, F., 2018. Classification of point cloud scenes with multiscale voxel deep network. arXiv preprint arXiv:1804.03583.

Rutzinger, M., Rüf, B., Vetter, M. and Höfle, B., 2010. Change detection of building footprints from airborne laser scanning acquired in short time intervals. ISPRS Technical Commission VII Symposium, pp. 475-480.

Shen, X., Wang, C., Wen, C., Liu, W., Sun, X. and Li, J., 2018, July. Discriminative learning of point cloud feature descriptors based on siamese network. In IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium (pp. 4519-4522). IEEE.

Sherrah, J., 2016. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv preprint arXiv:1606.02585.

Singh, A., 1989. Digital change detection techniques using remotely-sensed data. International journal of remote sensing, 10(6), pp. 989-1003.

Sirmacek, B. and Unsalan, C., 2009. Damaged building detection in aerial images using shadow information. Recent Advances in Space Technologies. pp. 249-252.

Sithole, G. and Vosselman, G., 2004. Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. ISPRS J. Photogram. Remote Sens., 59(1), pp. 85-101.

Sofia, G., Bailly, J.S., Chehata, N., Tarolli, P. and Levavasseur, F., 2016. Comparison of Pleiades and LiDAR digital elevation models for terraces detection in farmlands. IEEE J. Select. Top. Appl. Earth Observ. Remote Sens. 9(4), 1567-1576.

Soilán Rodríguez, M., Lindenbergh, R., Riveiro Rodríguez, B. and Sánchez Rodríguez, A., 2019. Pointnet for the automatic classification of aerial point clouds.

Stylianidis, E., Akca, D., Poli, D., Hofer, M., Gruen, A., Sánchez Martín, V., Smagas, K., Walli, A., Altan, O., Jimeno, E. and Garcia, A., 2020. FORSAT: a 3D forest monitoring system for cover mapping and volumetric 3D change detection. International Journal of Digital Earth, 13(8), pp.854-885.

Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.H. and Kautz, J., 2018. Splatnet: Sparse lattice networks for point cloud processing. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2530-2539).

Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE international conference on computer vision (pp. 945-953).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

Taigman, Y., Yang, M., Ranzato, M.A. and Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1701-1708.

Tchapmi, L., Choy, C., Armeni, I., Gwak, J. and Savarese, S., 2017. Segcloud: Semantic segmentation of 3d point clouds. In 2017 international conference on 3D vision (3DV) (pp. 537-547). IEEE.

Thomas, H., Goulette, F., Deschaud, J.E., Marcotegui, B. and LeGall, Y., 2018, September. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In 2018 International conference on 3D vision (3DV) (pp. 390-398).

Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F. and Guibas, L.J., 2019. KPConv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6411-6420).

Thompson, M.M., Eller, R.C., Radlinski, W.A. and Speert, J.L. eds., 1966. Manual of photogrammetry. American Society of Photogrammetry.

Tian, J., Cui, S. and Reinartz, P., 2013. Building change detection based on satellite stereo imagery and digital surface models. IEEE Transactions on Geoscience and Remote Sensing, 52(1), pp.406-417.

Tian, J., Schneider, T., Straub, C., Kugler, F. and Reinartz, P., 2017. Exploring Digital Surface Models from Nine Different Sensors for Forest Monitoring and Change Detection. Remote Sens. 9(3), 287-312.

Tomljenovic, I., Tiede, D. and Blaschke, T., 2016. A building extraction approach for airborne laser scanner data utilizing the object based image analysis paradigm. Int. J. Appl. Earth Obs. Geoinf. 52, 137-148.

Torres-Sánchez, J., Pena, J.M., de Castro, A.I. and López-Granados, F., 2014. Multi-temporal mapping of the vegetation fraction in early-season wheat fields using images from UAV. Computers and Electronics in Agriculture, 103, pp.104-113.

Toschi, I., Ramos, M.M., Nocerino, E., Menna, F., Remondino, F., Moe, K., Poli, D., Legat, K. and Fassi, F., 2017. Oblique photogrammetry supporting 3d urban reconstruction of complex scenarios. Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci. 42, 519-526.

Tran, T.H.G., Ressl, C. and Pfeifer, N., 2018. Integrated change detection and classification in urban areas based on airborne laser scanning point clouds. Sensors, 18(2), pp. 448.

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), pp. 8198-8214.

Vögtle, T. and Steinle, E., 2004. Detection and recognition of changes in building geometry derived from multitemporal laserscanning data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 35(B2), pp.428-433.

Volpi, M., Camps-Valls, G. and Tuia, D., 2015. Spectral alignment of multi-temporal cross-sensor images with automated kernel canonical correlation analysis. ISPRS Journal of Photogrammetry and Remote Sensing, 107, pp. 50-63.

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, pp. 77-85.

Vosselman G, Gorte B., Sithole G., 2004. Change detection for updating medium scale maps using laser altimetry. International Archives of

Photogrammetry, Remote Sensing and Spatial Information Sciences. 12;34(B3):207-12.

Vosselman, G. and Maas, H.G., 2010. Airborne and terrestrial laser scanning. CRC press.

Vosselman, G., 2008. Analysis of planimetric accuracy of airborne laser scanning surveys. ISPRS Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci., 37(3a), 99-104.

Vosselman, G., 2013. Point cloud segmentation for urban scene classification. ISPRS Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci. 1, 257-262.

Vosselman, G., Coenen, M. and Rottensteiner, F., 2017. Contextual segment-based classification of airborne laser scanner data. ISPRS journal of photogrammetry and remote sensing, 128, pp.354-371.

Vosselman, G., Gorte, B.G.H. and Sithole, G., 2004. Change detection for updating medium scale maps using laser altimetry. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 34(B3), pp.207-212.

Vu, T.T., Matsuoka, M. and Yamazaki, F., 2004, September. LIDAR-based change detection of buildings in dense urban areas. In IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium (Vol. 5, pp. 3413-3416). IEEE.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M. and Solomon, J.M., 2019. Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog), 38(5), pp.1-12.

Weinmann, M., Jutzi, B., Hinz, S. and Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing, 105, pp. 286-304.

Wen, C., Sun, X., Li, J., Wang, C., Guo, Y. and Habib, A., 2019. A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds. ISPRS journal of photogrammetry and remote sensing, 147, pp.178-192.

Winiwarter, L., Mandlburger, G., Schmohl, S. and Pfeifer, N., 2019. Classification of ALS point clouds using end-to-end deep learning. PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 87(3), pp.75-90.

Wolf, D., Prankl, J. and Vincze, M., 2015. Fast semantic segmentation of 3D point clouds using a dense CRF with learned parameters. In 2015 IEEE International conference on robotics and automation (ICRA) (pp. 4867-4873). IEEE.

Wu, C., Du, B., Cui, X. and Zhang, L., 2017. A post-classification change detection method based on iterative slow feature analysis and Bayesian soft fusion. Remote Sensing of Environment, 199, pp. 241-255.

Wu, W., Qi, Z. and Fuxin, L., 2019. Pointconv: Deep convolutional networks on 3d point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9621-9630).

Xiao, W., Xu, S., Elberink, S.O. and Vosselman, G., 2012. Change detection of trees in urban areas using multi-temporal airborne lidar point clouds. In Remote Sensing of the Ocean, Sea Ice, Coastal Waters, and Large Water Regions 2012. International Society for Optics and Photonics. (Vol. 8532, p. 853207).

Xu, S., Vosselman, G. and Elberink, S.O., 2014. Multiple-entity based classification of airborne laser scanning data in urban areas. ISPRS Journal of photogrammetry and remote sensing, 88, pp.1-15.

Xu, S., Vosselman, G., & Oude Elberink, S., 2015. Detection and Classification of Changes in Buildings from Airborne Laser Scanning Data. Remote Sensing, II-5/W2(November), 343–348.

Yang, B. and Chen, C., 2015. Automatic registration of UAV-borne sequent images and LiDAR data. ISPRS J. Photogram. Remote Sens. 101, 262-274.

Yang, Z., Jiang, W., Xu, B., Zhu, Q., Jiang, S. and Huang, W., 2017. A convolutional neural network-based 3D semantic labeling method for ALS point clouds. Remote Sensing, 9(9), p.936.

Yilmaz, C. S. and Gungor, O., 2016. Comparison of the performances of ground filtering algorithms and DTM generation from a UAV-based point cloud. Geocarto Int., pp. 1-16.

Yousefhussien, M., Kelbe, D.J., Ientilucci, E.J. and Salvaggio, C., 2018. A multi-scale fully convolutional network for semantic labeling of 3D point clouds. ISPRS journal of photogrammetry and remote sensing, 143, pp.191-204.

Yu, X., Hyyppä, J., Kukko, A., Maltamo, M. and Kaartinen, H., 2006. Change detection techniques for canopy height growth measurements using airborne laser scanner data. Photogrammetric Engineering & Remote Sensing, 72(12), pp.1339-1348.

Zagoruyko, S. and Komodakis, N., 2015. Learning to compare image patches via convolutional neural networks. CVPR, pp. 4353-4361.

Zbontar, J. and LeCun, Y., 2015. Computing the stereo matching cost with a convolutional neural network. CVPR, pp. 1592-1599.

Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J. and Funkhouser, T., 2017. 3dmatch: Learning local geometric descriptors from RGB-D reconstructions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1802-1811).

Zhan, Y., Fu, K., Yan, M., Sun, X., Wang, H. and Qiu, X., 2017. Change detection based on deep Siamese Convolutional Network for optical aerial images. IEEE Geoscience and Remote Sensing Letters, 14(10), pp.1845-1849.

Zhang, L., Sun, J. and Zheng, Q., 2018. 3D Point Cloud Recognition Based on a Multi-View Convolutional Neural Network. Sensors, 18(11), p.3681.

Zhang, Y., Zhang, Y., Zhang, Y. and Li, X., 2016. Automatic extraction of DTM from low resolution DSM by two-steps semi-global filtering. ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci., 3(3), pp. 249-255.

Zhang, Z., Gerke, M., Peter, M., Yang, M.Y. and Vosselman, G., 2017. Dense matching quality evaluation - an empirical study. IEEE Joint Urban Remote Sensing Event (JURSE), pp. 1-4.

Zhang, Z., Gerke, M., Vosselman, G. and Yang, M.Y., 2018a. A patch-based method for the evaluation of dense image matching quality. International Journal of Applied Earth Observation and Geoinformation, 70, pp. 25-34.

Zhang, Z., Gerke, M., Vosselman, G. and Yang, M.Y., 2018b. Filtering photogrammetric point clouds using standard lidar filters towards DTM generation. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4(2).

Zhang, Z., Vosselman, G., Gerke, M., Persello, C., and Yang, M.Y, 2019. Detecting building changes between airborne laser scanning and photogrammetric data. Remote Sensing, 11(20), 2417.

Zhao, H., Jiang, L., Jia, J., Torr, P.H. and Koltun, V., 2021. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16259-16268.

Zhao, Z., Cheng, Y., Shi, X., Qin, X. and Sun, L., 2018, November. Classification of LiDAR point cloud based on multiscale features and pointnet. In 2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA) (pp. 1-7). IEEE.

Zhou, J., Wang, M.J., Mao, W.D., Gong, M.L. and Liu, X.P., 2020. SiamesePointNet: A siamese point network architecture for learning 3d shape descriptor. In Computer Graphics Forum (Vol. 39, No. 1, pp. 309-321).

Zhou, K., Lindenbergh, R., Gorte, B. and Zlatanova, S., 2020. LiDAR-guided dense matching for detecting changes and updating of buildings in Airborne LiDAR data. ISPRS Journal of Photogrammetry and Remote Sensing, 162, pp.200-213.

Zhou, Y. and Tuzel, O., 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4490-4499).

Zhu, Q., Li, Y., Hu, H. and Wu, B., 2017. Robust point cloud classification based on multi-level semantic relationships for urban scenes. ISPRS journal of photogrammetry and remote sensing, 129, pp.86-102.

Zhu, X.X., Wang, Y., Montazeri, S. and Ge, N., 2018. A review of ten-year advances of multi-baseline SAR interferometry using TerraSAR-X data. Remote Sensing, 10(9), pp.1374.

# Summary

3D change detection draws more and more attention in recent years due to the increasing availability of 3D data. It can be used in the fields of land use / land cover (LULC) change detection, 3D geographic information updating, terrain deformation analysis, urban construction monitoring et al. Our motivation to study 3D change detection is mainly related to the practical need to update the outdated point clouds captured by Airborne Laser Scanning (ALS) with new point clouds obtained by dense image matching (DIM).

The thesis has three main parts. The first part, chapter 1, explains the motivation, providing a review of current ALS and airborne photogrammetry techniques. It also presents the research objectives and questions. The second part including chapter 2 and chapter 3 evaluates the quality of photogrammetric products and investigates their potential for change detection. The third part including chapter 4 and chapter 5 proposes two methods for change detection that meet different requirements.

To investigate the potential of using point clouds derived by dense matching for change detection, we propose a framework for evaluating the quality of 3D point clouds and DSMs generated by dense image matching. Our evaluation framework based on a large number of square patches reveals the distribution of dense matching errors in the whole photogrammetric block. Robust quality measures are proposed to indicate the DIM accuracy and precision quantitatively. The overall mean offset to the reference is 0.1 Ground Sample Distance (GSD); the maximum mean deviation reaches 1.0 GSD. We also find that the distribution of dense matching errors is homogenous in the whole block and close to a normal distribution based on many patch-based samples. However, in some locations, especially along narrow alleys, the mean deviations may get worse. In addition, the profiles of ALS points and DIM points reveal that the DIM profile fluctuates around the ALS profile. We find that the accuracy of DIM point cloud improves and that the noise level decreases on smooth ground areas when oblique images are used in dense matching together with nadir images.

Then we evaluate whether the standard LiDAR filters are effective to filter dense matching points in order to derive accurate DTMs. Filtering results on a city block show that LiDAR filters perform well on the grassland, along bushes and around individual trees if the point cloud is sufficiently precise. When a ranking filter is used on the point clouds before filtering, the filtering will identify fewer but more reliable ground points. However, some small objects on the terrain will be filtered out. Since we aim at obtaining accurate DTMs, the ranking filter shows its value in identifying reliable ground points.

Based on the previous findings in DIM quality, we propose a method to detect building changes between ALS and photogrammetric data. Firstly, the ALS points and DIM points are split out and concatenated with the orthoimages. The multimodal data are normalized to feed into a pseudo-Siamese Neural network for change detection. Then, the changed objects are delineated through per-pixel classification and artefact removal. The change detection module based on a pseudo-Siamese CNN can quickly localize the changes and generate coarse change maps. The next module can be used in precise mapping of change boundaries. Experimental results show that the proposed pseudo-Siamese Neural network can cope with the DIM errors and output plausible change detection results. Although the point cloud quality from dense matching is not as fine as laser scanning points, the spectral and textural information provided by the orthoimages serve as a supplement.

Considering that the tasks of semantic segmentation and change detection are correlated, we propose SiamPointNet++ model to combine the two tasks in one framework. The method outputs a pointwise joint label for each ALS point. If an ALS point is unchanged, it is assigned a semantic label; If an ALS point is changed, it is assigned a change label. The sematic and change information are included in the joint labels with minimum information redundancy. The combined Siamese network learns both intra-epoch and inter-epoch features. Intra-epoch features are extracted at multiple scales to embed the local and global information. Inter-epoch features are extracted by Conjugated Ball Sampling (CBS) and concatenated to make change inference. Experiments on the Rotterdam data set indicate that the network is effective in learning multi-task features. It is invariant to the permutation and noise of inputs and robust to the data difference between ALS and DIM data. Compared with a sophisticated object-based method and supervised change detection, this method requires much less hyper-parameters and human intervention but achieves superior performance.

As a conclusion, the thesis evaluates the quality of dense matching points and investigates its potential of updating outdated ALS points. The two change detection methods developed for different applications show their potential in the automation of topographic change detection and point cloud updating. Future work may focus on improving the generalizability and interpretability of the proposed models.

# Samenvatting

3D-veranderingsdetectie krijgt de laatste jaren steeds meer aandacht door de toenemende beschikbaarheid van 3D-gegevens. Het kan worden gebruikt op het gebied van detectie van veranderingen in landgebruik / landbedekking (LULC), actualiseren van 3D geografische informatie, analyse van terreinvervorming, monitoring van stedelijke bouw etc. Onze motivatie om 3D-veranderingsdetectie te bestuderen heeft voornamelijk te maken met de praktische noodzaak om de verouderde puntwolken verkregen met vliegtuiglaserscanning (VLS) punten te actualiseren met nieuwe puntwolken die met dense image matching (DIM) zijn verkregen.

Het proefschrift heeft drie hoofddelen. In het eerste deel, hoofdstuk 1, wordt de motivatie toegelicht en wordt een overzicht gegeven van de huidige VLS en luchtfotogrammetrische technieken. Ook worden de onderzoeksdoelstellingen en -vragen gepresenteerd. Het tweede deel, dat hoofdstuk 2 en hoofdstuk 3 omvat, evalueert de kwaliteit van fotogrammetrische producten en onderzoekt hun potentieel voor het detecteren van veranderingen. In het derde deel, dat hoofdstuk 4 en hoofdstuk 5 omvat, worden twee methoden voor veranderingsdetectie voorgesteld die aan verschillende eisen voldoen.

Om het potentieel te onderzoeken van het gebruik van puntenwolken afgeleid door middel van dense matching voor het detecteren van veranderingen, stellen we een raamwerk voor het evalueren van de kwaliteit van 3D puntenwolken en DSMs gegenereerd door middel van dense image matching. Ons evaluatiekader, gebaseerd op een groot aantal vierkante terreinstukken, laat de verdeling van fouten in dense matching in het hele fotogrammetrische blok zien. Robuuste kwaliteitsmaatstaven worden voorgesteld om de DIM nauwkeurigheid en precisie kwantitatief aan te geven. De totale gemiddelde afwijking ten opzichte van de referentie is 0,1 keer de grootte van een pixel in het terrein, de Ground Sample Distance (GSD); de maximale gemiddelde afwijking bereikt 1,0 GSD. Gebaseerd op steekproeven, stellen we ook vast dat de verdeling van dense matching fouten homogeen is in het hele blok en dicht bij een normale verdeling ligt. Op sommige locaties, vooral langs smalle steegjes, kunnen de gemiddelde afwijkingen echter groter worden. Bovendien blijkt uit de profielen van VLS-punten en DIM-punten dat het DIM-profiel rond het VLS-profiel fluctueert. We vinden dat de nauwkeurigheid van DIM-puntenwolk verbetert en het ruisniveau op gladde grondgebieden daalt, wanneer oblieke luchtfoto's samen met de nadirluchtfoto's voor de dense matching worden gebruikt.

Vervolgens evalueren we of de standaard LiDAR-filters effectief zijn om nauwkeurige DTM's af te leiden uit puntwolken die met dense image matching

zijn verkregen. Filterresultaten op een stadsblok laten zien dat LiDAR-filters goed presteren op het grasland, langs struiken en rond vrijstaande bomen als de puntenwolk voldoende nauwkeurig is. Wanneer een rangschikkingsfilter wordt gebruikt op de puntenwolken vóór het filteren, zal het filteren minder maar wel betrouwbaardere grondpunten identificeren. Wel worden enkele kleine objecten op het terrein er uitgefilterd. Aangezien we ernaar streven nauwkeurige DTM's te verkrijgen, toont het rangschikkingsfilter zijn waarde bij het identificeren van betrouwbare grondpunten.

Op basis van de eerdere bevindingen van de DIM-kwaliteit stellen we een methode voor om veranderingen in gebouwen tussen VLS en fotogrammetrische gegevens te detecteren. Ten eerste worden de VLS-punten en DIM-punten gesplitst en gecombineerd met de orthobeelden. De multimodale gegevens worden genormaliseerd om te worden ingevoerd in een pseudo-Siamees neuraal netwerk voor veranderingsdetectie. Vervolgens worden de gewijzigde objecten afgebakend met een classificatie per pixel en door verwijdering van artefacten. De module voor veranderingsdetectie op basis van een pseudo-Siamese CNN kan de veranderingen snel lokaliseren en grove mutatiekaarten genereren. Een aansluitende module kan worden gebruikt voor het nauwkeurig in kaart brengen van veranderingsgrenzen. Experimentele resultaten tonen aan dat het voorgestelde pseudo-Siamese neurale netwerk de DIM-fouten aankan en plausibele veranderings-detectieresultaten oplevert. Hoewel de puntenwolk van dense matching minder precies is als die van laserscanning, dienen de spectrale en textuurinformatie die door de orthobeelden wordt geleverd als een aanvulling.

Aangezien de taken van semantische segmentatie en veranderingsdetectie gerelateerd zijn, stellen we het SiamPointNet++-model voor om de twee taken in één raamwerk te combineren. De methode geeft een puntsgewijs gezamenlijk label af voor elk VLS-punt. Als een VLS-punt ongewijzigd is, krijgt het een semantisch label; als een VLS-punt is gewijzigd, krijgt het een wijzigingslabel. De semantische en wijzigingsinformatie zijn opgenomen in de gezamenlijke labels met minimale redundantie. Het gecombineerde Siamese netwerk leert zowel intra-epoche- als inter-epoche-kenmerken. Kenmerken binnen één epoche worden op meerdere schalen geëxtraheerd om de lokale en globale informatie te benutten. Functies tussen verschillende epoches worden geëxtraheerd door middel van geconjugeerde bolsampling en aaneengeschakeld om veranderingen te detecteren. Experimenten met de Rotterdamse dataset geven aan dat het netwerk effectief is in het leren van multi-tasking-functies. Het is invariant voor de permutatie en ruis van de puntwolken en robuust voor het verschil in eigenschappen tussen VLS- en DIM-gegevens. Vergeleken met een geavanceerde objectgebaseerde methode en met een supervised veranderingsdetectie, vereist deze methode veel minder hyperparameters en handmatig werk, maar levert het superieure prestaties.

Tot slot evalueert het proefschrift de kwaliteit van dense matching punten en onderzoekt het het potentieel om verouderde VLS-punten te actualiseren. De twee veranderingsdetectiemethoden die voor verschillende toepassingen zijn ontwikkeld, tonen hun potentieel voor de automatisering van topografische veranderingsdetectie en het actualiseren van puntenwolken. Toekomstig werk kan zich richten op het verbeteren van de generaliseerbaarheid en interpreteerbaarheid van de voorgestelde modellen.

# Biography

Zhenchao Zhang was born in Wuhu, Anhui, China in 1991. He received a BSc. Degree and an MSc Degree in Photogrammetry and Remote Sensing at the Zhengzhou Institute of Surveying & Mapping in 2012 and 2015, respectively. On November 1st of 2015, he started as a Ph.D. candidate at the Earth Observation Science department at the Faculty of Geo-Information Science and Earth Observation Science (ITC), University of Twente, the Netherlands. The PhD program was sponsored by China Scholarship Council (CSC). In 2017, he visited Technical University of Braunschweig as a research scientist for three months. He focused on photogrammetric quality control and multimodal change detection. He has published various papers in the remote sensing journals and ISPRS events.

# List of Publications


**Zhang, Z.**, Yang, M. and Vosselman, M.G., 2016. Dense matching quality evaluation - towards updating national point clouds. In NCG Symposium 2016: Nederlands Centrum voor Geodesie en Geo-Informatica. (Abstract & presentation)

**Zhang, Z.**, Gerke, M., Peter, M., Yang, M. Y., and Vosselman, G., 2017. Dense matching quality evaluation-an empirical study. In 2017 Joint IEEE Urban Remote Sensing Event (JURSE), 1-4.

**Zhang, Z.**, Gerke, M., Vosselman, G., and Yang, M. Y., 2018. Filtering photogrammetric point clouds using standard lidar filters towards DTM generation. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4(2).

**Zhang, Z.**, Gerke, M., Vosselman, G., and Yang, M. Y., 2018. A patch-based method for the evaluation of dense image matching quality. International Journal of Applied Earth Observation and Geoinformation, 70, 25-34.

**Zhang, Z.**, Vosselman, G., Gerke, M., Tuia, D., and Yang, M. Y., 2018. Change detection between multimodal remote sensing data using Siamese CNN. arXiv preprint arXiv:1807.09562.

**Zhang, Z.**, Gerke, M., Vosselman, G. and Yang, M. Y., 2018. Patch-based evaluation of dense image matching quality. arXiv preprint arXiv:1807.09546.

**Zhang, Z.**, Vosselman, G., Gerke, M., Persello, C., Tuia, D., and Yang, M. Y., 2019. Change detection between digital surface models from airborne laser scanning and dense image matching using convolutional neural networks. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4.

**Zhang, Z.**, Vosselman, G., Gerke, M., Persello, C., Tuia, D. and Yang, M.Y., 2019. Detecting building changes between airborne laser scanning and photogrammetric data. Remote sensing, 11(20), 2417.

Li, K., Zhang, Y., **Zhang, Z.**, and Yu, Y., 2018. An automatic recognition and positioning method for point source targets on satellite images. ISPRS International Journal of Geo-Information, 7(11), 434.

Li, K., Zhang, Y., **Zhang, Z.**, and Xu, L., 2018. High-precision centroid extraction and PSF calculation on remote sensing image of point source array. In 2018 10th IEEE IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS), 1-8.

Li, K., Zhang, Y., **Zhang, Z.**, and Lai, G., 2019. A coarse-to-fine registration strategy for multi-sensor images with large resolution differences. Remote Sensing, 11(4), 470.

# ITC Dissertation List

https://www.itc.nl/Pub/research_programme/Research-review-and-output/PhD-Graduates