Automatic Reconstruction of Three-Dimensional Building Models from Dense Image Matching Datasets

Andrew Philip McClune

BSc (Hons) Surveying and Mapping Science

Thesis submitted for the Degree of Doctorate of Philosophy

School of Civil Engineering and Geosciences

Newcastle University

May 2017

Abstract

The generation of three-dimensional (3D) building models without roof geometry is currently easily automated using a building footprint and single height value. The automatic reconstruction of roof structures, however, remains challenging because of the complexity and variability in building geometry. Attempts from imagery have utilised high spatial resolution but have only reconstructed simple geometry. This research addresses the complexity of roof geometry reconstruction by developing an approach, which focuses on the extraction of corners to reconstruct 3D buildings as boundary representation models, to try overcome the limitations of planar fitting procedures, which are currently favoured.

Roof geometry information was extracted from surface models, true orthophotos and photogrammetric point clouds; reconstructed at the same spatial resolution of the captured aerial imagery, with developments in pixel-to-pixel matching. Edges of roof planes were extracted by the Canny edge detector, and then refined with a workflow based on the principles of scan-line segmentation to remove false positive detection. Line tracing procedures defined the corner positions of the extracted edges. A connectivity ruleset was developed, which searches around the endpoints of unconnected lines, testing for potential connecting corners. All unconnected lines were then removed reconstruct 3D models as a closed network of connecting roof corners. Building models have been reconstructed both as block models and also with roof structures.

The methodology was tested on data of Newcastle upon Tyne, United Kingdom, with results showing corner extraction success at 75% and to within a planimetric accuracy of ±0.5 m. The methodology was then tested on data of Vaihingen, Germany, which forms part of the ISPRS 3D reconstruction benchmark. This allowed direct comparisons to be made with other methods. The results from both study areas showed similar planimetric accuracy of extracted corners. However, both sites were not as successful in the reconstruction of roof planes.

Dedicated to my grandfather, Philip Partridge. A great engineer in his own right, who always pushed me to achieve more.

Acknowledgements

Pierre de Coubertin once said “the important thing in life is not triumph, but the struggle; the essential thing is not to have conquered, but to have fought well.” This research has proved beyond challenging at times, and there are many people I owe a huge amount of gratitude towards to their assistance along this journey.

Firstly, I would like to thank Professor Jon Mills for his supervision and guidance throughout the duration of this research. There have been many changes to the supervision team, but thankfully Jon has always been there. He has taught me how to be a successful researcher and grow within the academic environment.

I would like to thank all the other supervisors who have helped with this project. Dr Pauline Miller, who originally approached me regarding the PhD, who believed I had the qualities to undertake the research and has continuously believed in me; even when I did not believe in myself. Although she moved away, she has always made time for me to talk things through and championed my research. Thanks also go to Dr Darion Grant who stepped into the supervisory role upon Pauline’s departure from Newcastle. His encouragement and knowledge were gratefully appreciated during a difficult period in the research.

My thanks go to Ordnance Survey for their continued support throughout the project with funding, data supply and assistance in progressing the research. Particular thanks go to Dr David Holland, who supervised the project from Ordnance Survey, has provided guidance when called upon, and has always been very hospitable during study visits to Southampton. Alongside Dr Isabel Sargent, they have both been extremely kind and supportive of my research.

Whilst I would not recommend a long-distance relationship whilst undertaking a PhD I would like to thank my girlfriend, who even despite the great distance and tough times she has had, she has always provided necessary comfort. She was a constant rock throughout my research and I don’t think I would be where I am today without her love and strength.

I would like to thank my friends and family, both close and extended, who have been constant advocates throughout my research. My parents have always provided sound council to prevent me making rash choices, particularly in the many hours of frustration. I have encountered many stumbling blocks throughout this research, but they have always given strong advice, helped keep me focussed and shown that there was always light at the end of the tunnel.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

1D – One-dimensional

2D – Two-dimensional

2.5D – Two and a Half Dimensional

3D – Three-dimensional

α-shape – Alpha-shape

B-Rep – Boundary Representation

CAD – Computer Aided Design

CCD – Charge-Coupled Device

CityGML – City Geographic Markup Language

CMOS - Complementary Metal-Oxide Semiconductor

CSG – Constructive Solid Geometry

csv – Comma-Separated Value

DGPF – The German Association for Photogrammetry and Remote Sensing

dm – decimetres

DSM – Digital Surface Model

DTM – Digital Terrain Model

DXF – Drawing Exchange File

EuroSDR – European Spatial Data Research

FME – Feature Manipulation Engine

FoV – Field of View

GIS – Geographical Information Systems

GNSS – Global Navigation Satellite Systems

GRNIR – Green, Red, Near Infrared

GSD – Ground Sampling Distance

IAAF – International Association of Athletics Federations

ISO – International Organisation for Standardization

ITN – Innovative Training Networks

ISPRS – International Society for Photogrammetry and Remote Sensing

Lidar – Light Detection and Ranging

LOD – Level of Detail

MBR – Minimum Bounding Rectangle

nDSM – normalised Digital Surface Model

NDVI – Normalised Difference Vegetation Index

NIR – Near Infrared

NMCAs – National Mapping and Cadastral Agencies

NSEW – North, South, East, West

OGC – Open Geospatial Consortium

OS – Ordnance Survey

OS MM – Ordnance Survey MasterMap

pts/m^2 - Points per Square Metre

RAG – Roof Adjacency Graph

RANSAC – Random Sampling Consensus

RGB – Red, Green, Blue

RMSE – Root Mean Square Error

RTG – Roof Topology Graphs

SGM – Semi Global Matching

SIFT - Scale Invariant Feature Transform

SIG 3D - Special Interest Group 3D

tfw – TIFF World File

TIFF – Tagged Image File Format

TIN – Triangular Irregular Network

UAS – Unmanned Aerial System

UN-GGIM – United Nations Global Geospatial Information Management

USGS – United States Geological Survey

VRML – Virtual Reality Modelling Language

WG – Working Group

WP – Work Packages

X3D – Extensible 3D

Chapter 1. Introduction

1.1 The Need for Three-Dimensional Data

Three-dimensional (3D) data is becoming more prominent in various applications as computer hardware and software is continually developed (Gröger and Plümer, 2012). The United States Geological Survey (USGS) indicated that 3D data has the potential to generate $13 billion worth of annual benefits to several sectors including floor risk management, security and aviation safety (Synder, 2012). However, many processes are still undertaken using two-dimensional (2D) data, because 3D data is not readily available. Sampling 3D data onto a 2D map also results in the loss of information (Stoter *et al.*, 2016a).

A study by the United Nations Initiative of Global Geospatial Information Management (UN-GGIM) predicted an acceleration of users moving from 2D mapping to 3D, with users expecting more complex and realistic 3D models as a core principle of geospatial analysis (Carpenter and Snell, 2013). The increased use of 3D is partly due to the freely available Virtual Globe software packages such as Google Earth and NASA Web World Wind (Brovelli *et al.*, 2016). Virtual Globe software has made 3D maps readily accessible through the online visualisation and combination of digital surface models (DSM), satellite imagery and building footprints (Sheppard and Cizek, 2009). Whilst these platforms have opened the possibility of 3D analysis, outstanding issues of Virtual Globes relate to the resolution of the imagery, and mismatches between 3D objects and the imagery. 3D data is also becoming an integral component as applications move towards 4D real-time monitoring.

With the developments in data capture and visualisation, 3D analysis has been highlighted as having the potential to become a key technology for geospatial data; yet the data to perform this is usually not readily available or is out-dated (Walter, 2014; Stoter *et al.*, 2016a). This has led National Mapping and Cadastral Agencies (NMCAs) to investigate the creation, maintenance and update of 3D datasets to a similar specification as current 2D datasets (Stoter *et al.*, 2016b). For example, Ordnance Survey (OS), the national mapping agency of Great Britain, have been investigating 3D datasets for over 10 years, and in particular 3D building reconstruction (Capstick and Heathcote, 2006).

There are various applications that are now adopting 3D building models, including real estate, pollution mapping and natural disaster modelling. Analysis of these and other applications can be completed using 3D city models. Half of the world's population currently live in cities and it is expected that by 2050 this figure will grow to two-thirds (United Nations, 2015). Goal 11 of the United Nations 2030 Agenda for Sustainable Development Goals aims to manage urban spaces better and develop sustainable cities to alleviate the pressure on fresh water supplies, sewage and the living environment caused by rapid urbanization (United Nations, 2015). One of the ways to achieve this aim is through the development of Smart Cities and City Information Management, where 3D data is an underpinning application (Prandi *et al.*, 2014; Chaturvedi and Kolbe, 2016; Thompson *et al.*, 2016).

One of the main issues currently faced by NMCAs is the reconstruction of 3D models. The initial cost of creating a 3D dataset is expensive; especially as it is currently not known to what specification and scale 3D models should be reconstructed (Stoter *et al.*, 2015). There are many different levels of detail for 3D building modelling. Different applications require varying amounts of detail and the amount of detail in the models can influence the results of the analysis (Walter, 2014; Biljecki *et al.*, 2016d). Many current workflows are completed in 2D and it has been deemed that customers were unaware of the potential of 3D data, particularly for analysis, and therefore unaware of how much detail is required (Walter, 2014). NMCAs require the ability to create datasets; with minimal costs, as automated as possible and at a national scale. If these can be applied to 3D datasets, it could allow NMCAs to show clients the advantages and applications of 3D, and potentially increase uptake in the use.

1.2 Data Availability for 3D Reconstruction

With the demand for 3D data becoming more prominent due to increased technology and applications, one of the main market potentials for 3D data, outlined in a survey on the state of 3D GIS, was 3D city models (Walter, 2014). 3D building models can be reconstructed from pre-existing 2D building footprints, by extrusion to a reference height value. However, this does not model the roof shape and complex building footprints can render reconstruction errors in automated approaches (Stoter *et al.*, 2015).

3D building models, with correct roof geometry, can be reconstructed manually using stereo photogrammetry (Kraus, 2007). Once the inner and outer orientations of the imagery have been established, the imagery can be viewed in stereo to give a 3D representation of the scene. By manually adjusting the height of the floating mark, the corner positions on the ground and roof of a building can be extracted to recreate a building in the third dimension (Kraus, 2007). This is manually intensive and would require significant processing time to model an entire city in this way. Therefore, more automated procedures have been researched and developed.

Of the many different automated approaches for reconstruction that have been proposed and researched, a wide range of sensors and platforms have been investigated. Satellite sensors offer large swath width (approximately 10 km) and a current best spatial resolution of 0.31 m for WorldView 3, with spatial accuracy between 1 – 3 m (Chen *et al.*, 2016; Toth and Jóźków, 2016). Satellite imagery offers a high temporal resolution, so is applicable for change detection, but can be affected by cloud cover making some images unsuitable for reconstruction. Aerial sensors offer a considerably lower swath width when compared to satellite sensors (approximately 1 km) but the spatial resolution is much higher; being able to capture detail at centimetre level with centimetre level spatial accuracy (Toth and Jóźków, 2016). On the other hand, terrestrial mobile mapping systems offer the smallest swath width (approximately 50 m) but the highest spatial resolution and accuracy, although the accuracy can range from centimetre level up to half a metre (Toth and Jóźków, 2016). Terrestrial mobile-mapping sensors tend to only capture façade data, limiting the ability to reconstruct the correct 3D roof geometry. Not all sides of a building can be seen from terrestrial mobile-mapping sensors, especially if there is no road for the vehicle to travel along during data collection. Terrestrial survey data has been used in the past as an ancillary dataset to fill in the areas that could not be captured from airborne capture (Capstick and Heathcote, 2006). The good trade-off between ground coverage and spatial resolution offered by aerial sensors has meant that most proposed methods have tended to utilise these datasets for 3D reconstruction. Due to the reasons described above, this research focuses on methods to reconstruct buildings from airborne sensors.

1.3 Attempts to Automate 3D Building Reconstruction

A large number of alternate 3D building reconstruction procedures have been proposed by researchers from airborne sensors; utilising data captured at different times of year, with varying spatial resolution, and of scenes with varying degrees of complexity. This, therefore, makes it difficult to compare the success of one method against another. To overcome this limitation, studies have been conducted which provide benchmark datasets for researchers to test their developed methodologies and analyse the results against other procedures.

The European Spatial Data Research (EuroSDR) community conducted an investigation to compare building extraction approaches in 2006 (Kaartinen and Hyyppa, 2006). Methods analysed included manual, semi-automatic and automatic approaches using aerial imagery and light detection and ranging (lidar) point clouds. Of 13 different approaches, three methods reconstructed solely from imagery, seven methods solely from lidar and three methods used varying levels of combination between the two datasets (Kaartinen and Hyyppa, 2006). Some of the methods tested used industry-standard software such as BAE System’s SOCET SET and Bentley’s TerraScan, which were compared to procedures developed by researchers in the academic community. The results showed greater accuracy in delineation of buildings from imagery, but greater height accuracy of reconstructed models from lidar data (Kaartinen and Hyyppa, 2006). In general, it was also found that lidar allowed higher automation of the developed methods; but found planimetric accuracy was influenced by the level of automation, with more manual intensive methods producing greater point accuracy (Kaartinen and Hyyppa, 2006). The concluding remarks of this study suggested that by combining aerial imagery and lidar together, the level of automation and accuracy of the reconstructed building models could be increased; but the improvements in digital cameras and laser scanning would see algorithmic developments, which would further improve automation.

The International Society for Photogrammetry and Remote Sensing (ISPRS) conducted a more recent comparison of 3D building reconstruction. In this study, the results of 14 methods for 3D reconstruction were analysed (Rottensteiner *et al.*, 2014). 10 out of the 14 methods reconstructed buildings from lidar point clouds, one from a lidar raster-based DSM and two from imagery. Whilst the EuroSDR investigation had concluded that lidar and imagery should be combined to harmonise the advantages of both datasets, only one approach combined the data together in the ISPRS benchmark test. The results were fairly successful in reconstruction, achieving on average 90% of roof plane reconstruction and corner positions to within 70 cm in planimetry and 30 cm in height (Rottensteiner *et al.*, 2014). As Kaartinen and Hyyppa (2006) had predicted, the level of automation in reconstruction from imagery and laser scanning had increased in the ISPRS benchmark, but no advantages in accuracy or quality of the results were seen when combining the two datasets together, with one potential reason being due to registration errors. Although methods achieved a high success rate, a number of shortcomings were found. For planar roof extraction, under-segmentation was a major issue, with reconstruction adversely affected by small substructures such as chimneys and dormer windows on the roof, and methods struggled to reconstruct complex roof geometry correctly (Rottensteiner *et al.*, 2014).

As can be seen from both of the aforementioned international studies, reconstruction from lidar data was preferred. This was also supported by Haala and Kada (2010) in a report on reconstruction methods, who found preferences to lidar data were due to difficulties in aerial image interpretation. Automatic reconstruction approaches have tended to favour simple structures; with roof shapes such as flat, gable and hipped, shown in Figure 1.1. It has been concluded that approximately 40-50% of buildings in Western European urban and suburban areas can be categorised into these simple geometry shapes (Haala and Kada, 2010). The main problem to automating the procedure is, therefore, caused by the large architectural variety of buildings that exist in rural, suburban and inner city regions (Haala and Kada, 2010). Whilst a wide range of research has already been conducted, the automation of 3D reconstruction is still a widely-investigated research topic.

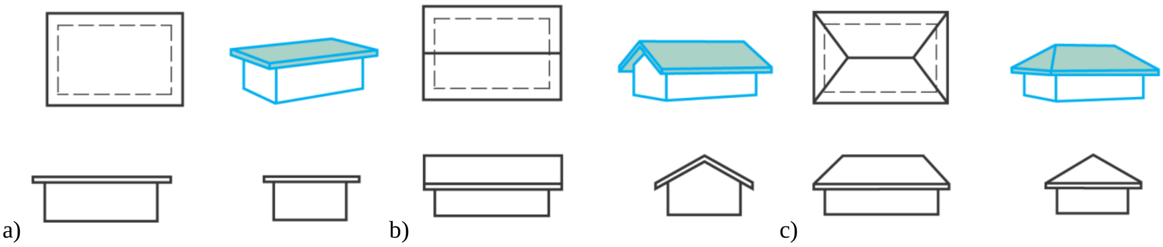


Figure 1.1 - Simple roof structures: a) flat roof; b) gable roof; and c) hipped roof (Goodheart-Willcox, 2009).

1.4 Digital Aerial Cameras

One of the concluding remarks of the ISPRS 3D reconstruction benchmark was that recent advances in image capture were not being fully exploited, and with only two approaches using imagery being analysed, more research was required in order to fully compare reconstruction from different sensors (Rottensteiner *et al.*, 2014). Digital aerial cameras have been developed over the last 20 years, which are capable of capturing imagery at a higher spatial, spectral and radiometric resolution compared to analogue film-based cameras. These developments have coincided with the progression of pixel-to-pixel matching algorithms, which produce dense image matching datasets. These datasets include raster gridded DSMs at the same spatial resolution as the captured imagery and led to the production of true-orthophotos, where all pixels in the image are ortho-rectified. The matching of each individual pixel between stereo-pairs of images has also resulted in the construction of 3D photogrammetric point clouds, at the same spatial resolution as the imagery. With regards to building reconstruction, these datasets produced from image-matching offer a high level of detail, with well-defined edges, which can be particularly useful for building reconstruction (Haala *et al.*, 2010). By exploiting developments in aerial image capture and processing, high radiometric information can also be obtained along with the geometric properties of the roof (Haala and Kada, 2010).

1.5 Aims and Objectives

The aim of this research is to produce a novel method for the automatic reconstruction of building models from dense image-matching datasets at the Level of Detail (LOD) 1 and 2 specifications, as defined by the Open Geospatial Consortium (OGC) City Geographic Mark-up Language (CityGML) standard (presented in Chapter 2).

In order to achieve this aim, the following objectives were defined:

i. Investigate the current state-of-the-art methodology in building modelling and rooftop extraction from aerial photography and lidar data, examining levels of automation as well as the relative strengths and weaknesses of different approaches;

ii. Based on objective i, propose and develop an automated procedure for the extraction of 3D rooftop vertices from dense image matching datasets;

iii. Integrate the information extracted from the procedure developed in objective ii in order to produce 3D building models;

iv. Validate and refine the methodology proposed in objectives ii and iii, delivering appropriate metrics on the accuracy and reliability of the developed approach.

1.6 Contribution to Knowledge

As shown in the aforementioned EuroSDR and ISPRS benchmark studies, recent approaches to automate 3D reconstruction have tended to favour lidar data, but have still not managed to achieve 100% automation and correct model geometry. It has also been noted that more research needs to be conducted into the reconstruction of 3D buildings from dense image matching datasets. This research will, therefore, develop a workflow for 3D reconstruction that exploits the strengths of dense image matching datasets, in an attempt to overcome the current limitations of lidar approaches, which will be discussed in Chapter 2. The method will move away from planar segmentation procedures to reconstruct building models by connecting roof-lines, as shown in Figure 1.2, to form a closed-network of roof edges.

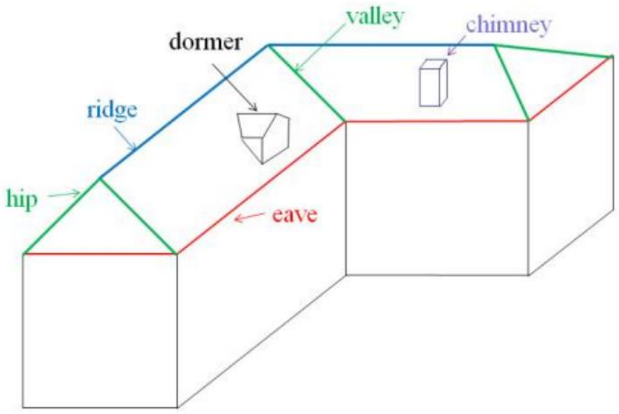


Figure 1.2 – Roof-line labels (Nex and Remondino, 2012).

Feature-based extraction methods will be shown in subsequent chapters to be highly sensitive to noise (Maini and Aggarwal, 2009), with the outputs requiring post processing. An area-based approach, which has been proposed for planar segmentation, but found to struggle with boundary extraction, will then be developed to refine the extracted edge features. This will show the complementary nature of the two approaches; how the area-based method can refine the results of feature extraction, and how these refined results can provide the boundary of areas, which have previously been difficult to extract using area-based approaches.

The use of constraints has been shown to be necessary for reconstruction (Oude Elberink and Vosselman, 2011; Nex and Remondino, 2012; Meidow *et al.*, 2016); to ensure the regular geometry of reconstructed models. However, to date, these constraints can limit the types of buildings that are reconstructed. For example, approaches have limited reconstruction to only rectangular footprints and planes, by enforcing orthogonal angles at corners (Woo *et al.*, 2008; Melnikova and Prandi, 2011). This research will investigate whether by enlarging the constraints, and not just limiting corners to orthogonal angles, a wider range of buildings can be correctly reconstructed.

The proposed methodology will also be tested on the ISPRS benchmark for 3D reconstruction. This will provide an additional set of reconstruction results, which can be compared against current state-of-the-art approaches. This will also increase the pool of research for 3D building reconstruction from aerial imagery using the benchmark dataset.

1.7 Thesis Structure

The thesis has been structured as follows. Chapter 2 will examine the current 3D building reconstruction literature. Examples of existing 3D building models, at varying scales, as well as some applications will be presented. The CityGML specification for 3D reconstruction will also be explained, with examples shown for each defined LOD. The review will then critique the methods used for the extraction of roof geometry and the reconstruction of building models.

Chapter 3 will outline recent developments in aerial image capture. The evolution of the airborne camera system from film to digital will be discussed, before examining different digital camera systems technologies and structures. A brief historical overview of image matching procedures will be presented, before outlining the development of pixel-to-pixel matching algorithms, which has led to the creation of dense image matching datasets. Dense image matching surface models, created from aerial imagery, will be evaluated in more detail in Section 3.4, where they will also be compared against the results produced from lidar systems. This chapter will conclude by presenting the study areas utilised in this research; discussing the cameras used to capture the imagery and how the data was processed.

Chapter 4 will propose the developed methodology. The workflow was split into three main work packages: the extraction of roof geometry, the refinement of the extracted roof geometry, and the reconstruction of 3D building models. The labels of the extracted roof geometry used during reconstruction, which will be described in more detail in Chapter 5 and 6, can be seen in Figure 1.2. Each of these work packages will be discussed in detail, along with explanations of the pre-processing procedures implemented. Results of requisite sensitivity testing will be presented throughout Chapter 4 for each section.

The reconstructed 3D building models will be delivered in Chapter 5 and Chapter 6. The results were analysed on a visual basis, as well as by measuring the accuracy of the corners detected for both block models and models with roof geometry, and analysing the roof planes extracted. The results will be evaluated in detail to highlight the success and current limitations of the proposed method. Chapter 5 will discuss the results of reconstructed models for the Newcastle upon Tyne, UK, dataset; and Chapter 6 will present results of transferring the methodology to a separate dataset, the ISPRS 3D reconstruction benchmark of Vaihingen, Germany.

Chapter 7 will conclude the thesis. The developed methodology will be summarised and critiqued before discussing the suitability of dense image matching datasets for 3D reconstruction and recommendations for future research.

Chapter 2. Current Uses and Methods for 3D Building Reconstruction

2.1 Introduction

This chapter will examine the current implementation of 3D building models: why they are important, what standards are used to govern their reconstruction and examples of existing cities and countries that have reconstructed 3D models. Moreover, the current literature regarding building geometry extraction and 3D building reconstruction will be critiqued to determine the level of automation in the various approaches and the methods applied. Whilst there are many platforms and sensors that have been researched for 3D reconstruction, this review will focus solely on data extracted from aerial acquisition: either using an aerial camera or lidar systems, due to reasons explained in Section 1.2. This chapter will conclude by reviewing validation techniques currently used to determine the accuracy of 3D building models.

2.2 Importance of 3D Building Models

Many processing and visualisation applications currently carried out in 2D can be better achieved in 3D, which allows more sophisticated analysis to be conducted (Walter, 2014). As the software and hardware for 3D data capture, processing and analysis matures, there has been a growing need for 3D datasets for various applications (Stoter *et al.*, 2011). Predominately, 3D models have been used to aid visualisation such as flight simulation, navigation and change detection (Biljecki *et al.*, 2015). However, visualisation is only one application of 3D building models, with the extra dimension providing additional topological and volumetric analysis (de la Losa and Cervelle, 1999; Zlatanova *et al.*, 2002). Models have also been used for non-visualisation purposes, such as the determination of floor space within a building and the measure of solar radiation of multi-storey buildings for hedonic price modelling (Helbich *et al.*, 2013; Biljecki *et al.*, 2015; Boeters *et al.*, 2015). Other applications of buildings models include disaster management and prevention. For example, depending on the amount of detail in the model, various flood simulations can be constructed. These can range from simply how high floodwater may rise against a building to more detailed analysis: including which rooms are flooded, and the damage caused to electrical substations, due to contact with flood water (Varduhn *et al.*, 2015). The distribution of particles in the air, which influence human health, can be modelled more successfully using 3D building models to model wind fields in urban canyons, in comparison to 2D (Ghassoun *et al.*, 2015). Herbert and Chen (2015) explored the use of 2D and 3D building models for urban planning and shadow analysis, by interviewing different user groups. Participants found 3D data improved visualisation of a proposed building and its interaction within the surrounding area, and to be more useful for complex tasks such as shadow analysis when compared to 2D (Herbert and Chen, 2015).

2.2.1 Types of City Models

A city model consists of many 3D building models, which can be reconstructed with varying amounts of detail; from simple block models to photo-realistic reconstructions including interior geometry. The amount of detail reconstructed is dependent on the end application and the input data used. In order to create 3D building models to a specific quality, standards have been formed to define the geometry and semantic accuracy reproduced. Several company-led standards, such as AutoCAD Drawing Exchange File (DXF), Virtual Reality Modelling Language (VRML) and Extensible 3D (X3D), have been defined to support the geometry of the models but offer little support for semantic information (van den Brink *et al.*, 2013). The International Organisation for Standardization (ISO) outlines the requirements for data quality of geospatial data through the ISO 19100 group of standards (ISO, 2016). Within this group are standards on quality evaluation procedures (EN ISO 19114), data quality measures (TS ISO 19138) and data product specifications (ISO 19131) (ISO, 2016). However, these standards do not give specific requirements for 3D city models; and the defined quality principles are outlined in a report that must be purchased for implementation. Therefore, freely available standards have been developed, such as CityGML. This was developed as an international standard by the OGC for the representation, exchange and storage of 3D geographic information (OGC, 2012). This standard specifies all aspects of a 3D city model including vegetation and city furniture. CityGML was firstly developed by the Special Interest Group 3D (SIG 3D) from Germany in 2002, which now consists of stakeholders from industry, academia and the public sector from several countries and is in the second version of the standard, with version 3.0 currently in development (Gröger and Plümer, 2012; Löwner and Gröger, 2016). Geometric, semantic and visual aspects are based upon the IS19107 Spatial schema standard for Geographic Information, which defines a d-dimensional object as a set of (d-1)-dimensional primitives (Biljecki *et al.*, 2016a). Thus, a 3D model is represented as a 2D surface which is represented as a one-dimensional (1D) line with two zero dimensional endpoints. The 1D lines of the boundaries of the surface form the shell of the building (Kolbe, 2009). All surfaces must be closed, thus connected at the endpoints of each boundary to form a solid and in order to compute the volume of the building (Biljecki *et al.*, 2016a). A surface is limited to being planar to ensure broad system support, which means curved surfaces are not reconstructed in CityGML (Kolbe, 2009; Biljecki *et al.*, 2016a). For 3D building reconstruction, five levels of detail (LODs) are defined, which determine the types of features represented, with objects becoming more detailed between the varying levels (Gröger and Plümer, 2012):

- LOD0 – a two and a half dimensional (2.5D) digital terrain model (DTM) where buildings can be represented by a footprint or roof edge polygon;
- LOD1 – block model of prismatic buildings without any roof structures;
- LOD2 – models with distinctive roof structures and larger building installations including balconies and stairs;
- LOD3 – architectural models with detailed wall and roof structures, doors, windows and bays;
- LOD4 – interior structures including rooms and furniture are added to the LOD3 model.

The LODs, shown in Figure 2.1, not only define visual appearance but also denote accuracies and minimum dimensions, shown in Table 2.1, which can be used to assess the quality of the reconstructed models (OGC, 2012).

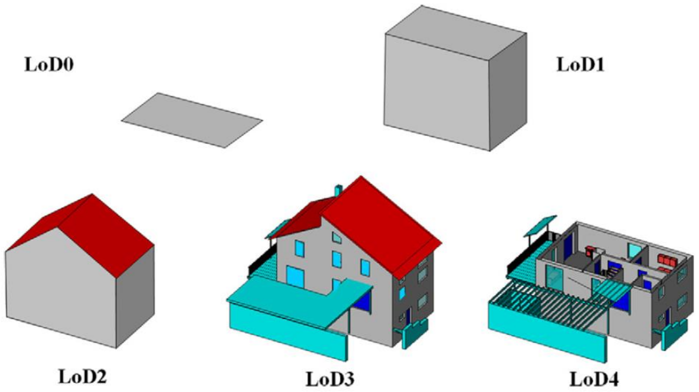


Figure 2.1 - Visual representation of the five levels of detail (LODs) outlined by CityGML (Gröger and Plümer, 2012).

	LOD0	**LOD1**	**LOD2**	**LOD3**	**LOD4**
Absolute 3D point accuracy	Lower than LOD1	5 m	2 m	0.5 m	0.2 m
Minimum footprint size	Maximal generalisation	6*6 m	4*4 m	2*2 m	Constructive elements and openings are represented
Generalisation		Object blocks as generalised features	Objects as generalised features	Objects as real features	

Table 2.1 – Summary of the required accuracy and geometry representation for each CityGML LOD (OGC, 2012).

Whilst CityGML has been widely adopted in research for the topological reconstruction of 3D building models, some problems have arisen with its implementation. For example, the lack of clear definitions for each LOD means CityGML cannot be used to quantify 3D reconstruction (Oude Elberink and Vosselman, 2011; Stoter *et al.*, 2015). The topological relationships between surfaces are often stored independently, which means the coordinates of a single point are stored three times, and it is possible that these three vertices are not exactly at the same location (Biljecki *et al.*, 2016a). This can cause invalid geometry to be reconstructed with surfaces not being watertight and closed. A lack of continuity and flexibility between different LODs has also been recognised (Biljecki *et al.*, 2014; Biljecki *et al.*, 2015; Löwner *et al.*, 2016). Models produced at the same LOD can have different geometric references, for example the assigned height of LOD1 models and variations in building footprint (Biljecki *et al.*, 2016d). These differences in the geometry of the reconstructed models have been shown to influence the results of spatial analysis (Biljecki *et al.*, 2016d). Attempts have been made to improve these limitations. Biljecki *et al.* (2014), for example, defined six quantifiable metrics, with the various combination of metrics resulting in 10 LODs for reconstruction. The metrics included the presence of objects, feature complexity, dimensionality, appearance (texture), spatio-semantic coherence and attribution (Biljecki *et al.*, 2014). The LOD concept has been further extended by Biljecki *et al.* (2016c) who proposed four models for each LOD (from LOD0 – LOD3) outlined by OGC CityGML, which, therefore, defined 16 LODs. The newly defined LODs accounted for different generalisations of building footprints, varying height extrusions, as well as modelling roof overhang, chimneys and dormer windows (Biljecki *et al.*, 2016c). These models have been proposed to support the CityGML standard and potentially assist with the specification of CityGML 3.0, which is currently under development (Biljecki *et al.*, 2016c; Löwner and Gröger, 2016). The prospects of multi-LOD data has also been recently proposed, where instead of storing a model with one type of geometry, multiple representations of a model are saved to the same model (Biljecki *et al.*, 2016b; Löwner *et al.*, 2016). This allows the user to change the visualisation and amount of detail of a model depending on the application.

The specifications from CityGML for 3D reconstruction have also been implemented within current NMCAs standards (van den Brink *et al.*, 2013). The Dutch Kadaster, for example, developed the limitations of CityGML to define a new standard for reconstruction by merging the concepts of CityGML with current 2D national standards to create a 3D national standard (Stoter *et al.*, 2011). Other specifications, such as the INSPIRE *Data Specification on Buildings*, have also been developed on the basis of CityGML (INSPIRE, 2013). Even though problems have been identified with CityGML, it is still the most widely used reference for 3D reconstruction, which has been adopted in 28 countries (Biljecki *et al.*, 2016d; Löwner *et al.*, 2016).

2.2.2 Example City Models

There are several examples of cities that have 3D models reconstructed by private vendors, as well as NMCAs, for various applications and with different details modelled. For example, the city of Berlin, Germany, offers a fully textured 3D city model consisting of approximately

474,000 buildings, with 200 buildings being reconstructed in high detail and a further five models including internal reconstruction (Berlin Business Location Centre, 2016a). This was created as part of a project involving a wide range of stakeholders and data providers: including research institutes, public, and private sector companies. The city model has many different applications and is available on different platforms. For example, a mobile and tablet version of the city is available as a freely downloadable app called smartMap Berlin (Berlin Business Location Centre, 2013). This allows the user to navigate through the city as well as search for specific addresses, with real estate metadata attached to the buildings. The city model has other visualisation applications. Berlin is a popular destination for marathon runners, being renowned for its fast course, and hosted the 12th International Association of Athletics Federation (IAAF) World Championships in 2009. A fly through video was created of the marathon route used in the championships, denoting mile markers (Berlin Business Location Centre, 2009). Historical data has also been integrated into the city model and allows users to visualise how the city previously looked. This includes a historic model of the Berlin Wall, which allows users to follow the original path of the wall in a 3D virtual aerial tour (Berlin Business Location Centre, 2016b). The 3D city model of Berlin was created and maintained as a series of individual datasets which can be changed interdependently, but are also integrated to form the city model. The datasets used for reconstruction of the city include aerial imagery, DTMs, digital architecture models and cadastral data (Kada, 2009). The building models were created from laser-scanning and photogrammetric methods. For the city centre scenes, buildings were modelled with flat roofs and imagery was projected onto the models to gives the appearance of the roof, as well as shading, as shown in Figure 2.2a (Kada, 2009). Higher detail models were converted from pre-existing architecture models into the dataset. The approach undertaken for the reconstruction of the Berlin model from lidar data, with roof geometry, will be further discussed later in this chapter.

There are many other examples of 3D city models that have been developed around the world. The city of Adelaide, Australia, provide a freely accessible 3D city model, via an online portal, with a claimed accuracy of within 0.5 m (Adelaide City Council, 2016). The list of uses include visualising transport, urban design and planning, as well as linking heritage sites with photos and textual information (Adelaide City Council, 2016). Similarly, the historical peninsula of Istanbul, Turkey, has been reconstructed at various LODs, an example of which is shown in Figure 2.2b, using aerial photography, terrestrial laser scanning and building footprints (Buhur *et al.*, 2008). The city planning office of Toronto, Canada, has a LOD1 city model called 3D Massing, which can be downloaded in various formats including Google SketchUp, AutoCAD and shapefiles for use in Geographical Information Systems (GIS) and Computer Aided Design (CAD) software packages (Toronto City Planning, 2015). A 3D city model of Vienna, Austria, has been reconstructed as part of the European Marie-Curie Innovative Training Networks (ITN) project "Ci-nergy, Smart Cities with Sustainable Energy Systems" (Agugiaro, 2016). LOD1 building models were reconstructed by extruding building footprints and an LOD2 dataset was modelled from lidar point clouds and building footprints (Agugiaro, 2016). Semantic information including building class and function, year of construction, as well as volume and area, were added to the building models in order to start modelling the energy efficiency of the buildings.

2.2.3 City Models and NMCAs

With the increasing use and demand for 3D models, NMCAs are beginning to explore the capture and production of 3D datasets. One of the first NMCAs to explore this was the Dutch Kadaster. An initial pilot study involving 65 organisations from academia, government and industry, investigated the need for 3D at a national scale (Stoter *et al.*, 2011). A national 2D polygon dataset was combined with a national height model of the Netherlands to create the 3D dataset called 3D TOP10NL (Oude Elberink *et al.*, 2013). The national height model was created from lidar data with a density of 10 points per square metre (pts/m^2) (Oude Elberink *et al.*, 2013). The reconstruction of buildings was based on a point-in-polygon procedure, which used the lidar points within each classified polygon (water, road, terrain, buildings and forest) to assign height values. Buildings were reconstructed at LOD1 by extruding the 2D building footprint to the average height of all lidar points that fell within the footprint. Constraints were applied to model the relationship between neighbouring polygons in order to ensure that the created 3D dataset was seamless (Oude Elberink *et al.*, 2013). An example of the 3D TOP10NL model can be seen in Figure 2.2c.

A similar approach has been undertaken by the OS, the national mapping agency of Great Britain. The OS MasterMap (OS MM) Topography layer, a dataset representing real world objects such as letter boxes, fences and buildings as points, lines and polygons, now attributes height to the building footprints (Ordnance Survey, 2014). Heights are defined from summary statistics of a DSM, created automatically from aerial photography (Ordnance Survey, 2014a; Sargent *et al.*, 2015). Three absolute building heights have been determined as important by user interviews: (1) absolute minimum height of the intersection of the external building walls and the underlying ground (AbsHmin); (2) the absolute height of the base of the roof (AbsH2); and (3) the absolute height of the highest point on the building (AbsHmax) (Sargent *et al.*, 2015). These three absolute heights form two relative heights, AbsH2 - AbsHmin and AbsHmax

- AbsHmin, which are attributed to the building footprints and allow the reconstruction of simple LOD1 3D building models at various roof heights (Sargent *et al.*, 2015). An example of the reconstructed OS Simple Building Heights model can be seen in Figure 2.2d.

Other NMCAs introducing 3D datasets include the Swiss Federal Office of Topography, Swisstopo, who, in 2008, changed the collection of land survey data from 2D to 3D (Stoter *et al.*, 2016b). 3D elements such as transportation networks and administrative boundaries are stored in a national database and have been incrementally released. The 3D building models, where roofs were manually created at 0.5 m accuracy from aerial photography, whilst facades were automatically reconstructed from original 2D data, are expected to be completed nationally by 2018 (Stoter *et al.*, 2016b).

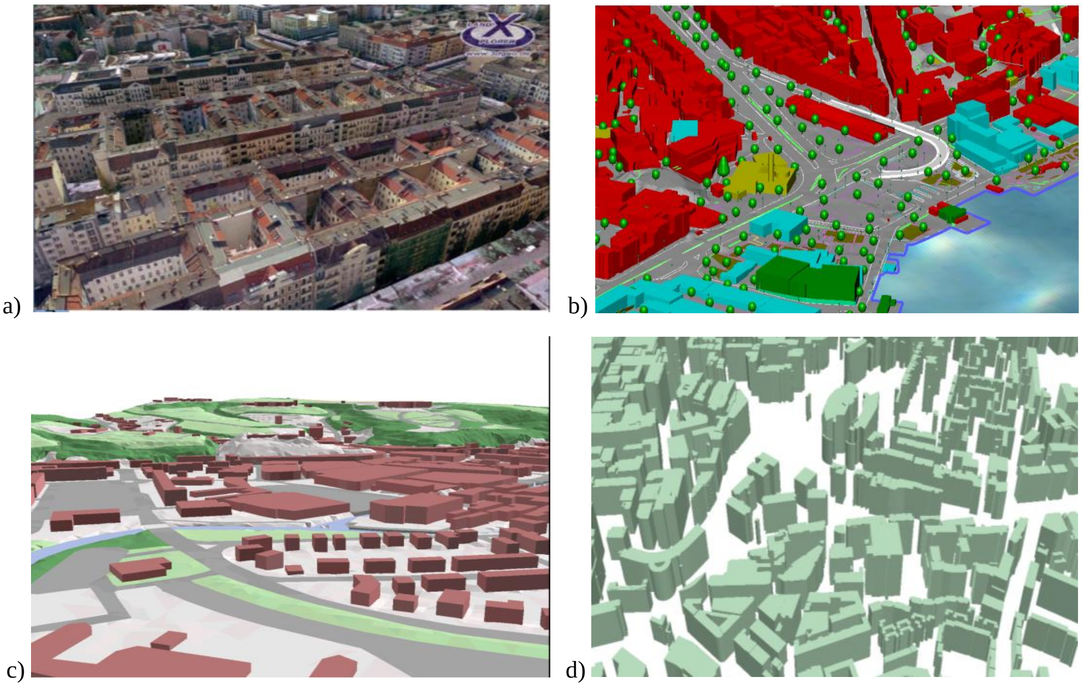


Figure 2.2 – Examples of 3D city models: a) Berlin (Kada, 2009); b) Istanbul (Emem and Batuk, 2004); c) 3D TOP10NL of the Netherlands (Oude Elberink *et al.*, 2013); and d) OS Simple Building Heights reconstructed at AbsH2 height attribute.

2.3 3D Building Model Creation

As highlighted in Section 2.2, 3D building models are becoming more prominent in various applications due to improvements in computer hardware and software for processing and analysis. The increased use of 3D models has pushed NMCAs and other organisation to firstly, create 3D models, but also maintain these datasets by incorporating any changes. The automatic reconstruction of 3D building models has therefore been a heavily researched topic in recent years. Roofs are generally defined as the combination of planar faces, following the IS19107 Spatial schema standard for Geographic Information, which can be reconstructed from modelling cues such as points, lines and planes grouped together to form the roof geometry (Sohn *et al.*, 2008). Roof planes can be extracted directly from the data or can be reconstructed using the boundary lines of the plane and points at the junction of connecting line segments (Peternell and Steiner, 2004). Once the modelling cues have been extracted, 3D models can be reconstructed. The following section investigates the extraction of roof geometry modelling cues; before examining how the extracted cues were used to reconstruct a 3D building.

2.3.1 Extraction of Roof Geometry Modelling Cues

The extraction of roof geometry for reconstruction segments the data into modelling cues. Due to the popularity of CityGML and its specifications, points, lines and planes have been the focus for the extraction of modelling cues. The extraction of these cues can be classified into two approaches: feature-based and area-based.

2.3.1.1 Feature-Based Extraction

Feature-based approaches aim to extract the corner points of the roof and the roof structure lines. These features can be manually extracted by digitising stereo-imagery. However, such an approach is very labour intensive and time-consuming, particularly for complex roof structures and large scenes. This approach could take up to weeks/months to model an entire city. Therefore, automated feature extraction algorithms, such as edge and point detectors, have been researched to extract roof lines and corners, respectively.

Edge detection algorithms delineate the boundary of separate objects to distinguish between an object and its background, as well as overlapping objects (Parker, 2010). Edge detectors are generally applied to 2D grayscale imagery, where an edge can be defined in its basic form as a change in the grey level values at a specific location. An example of this is shown in Figure 2.3 where a clear difference in the grey level values can be seen, on the left of the figure, producing a step edge in the profile of the grayscale colour gradient. However, due to artefacts such as image noise, this simple case is rarely reproduced and an edge profile of the kind seen on the right of Figure 2.3 is more frequently produced. Because of this, edge detectors have to be able to distinguish between image noise and true edges.

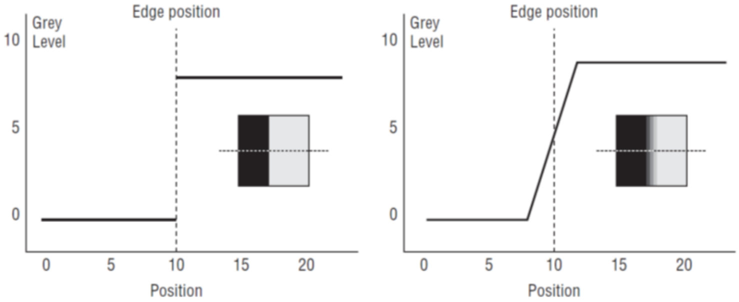


Figure 2.3 – Cross-sections of a grayscale image and the edge profile reproduced: (left) an ideal step edge; and (right) a realistic step edge due to image noise (Parker, 2010).

There are two fundamental approaches in which edge detection can be performed. Firstly, a template-based approach aims to model an edge using a small window to determine whether the central pixel of the window is at an edge between two regions. There are many different template-based approaches that have developed in image processing, including the Prewitt and Sobel edge detectors, which are shown in Figure 2.4 and Figure 2.5, respectively.

-1	0	1	-1	-1	-1
-1	0	1	0	0	0
-1	0	1	1	1	1

Figure 2.4 – Prewitt edge detector window operators.

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

Figure 2.5 – Sobel edge detector window operators.

To model the level of grayscale change from a pixel at the centre of a window to its neighbouring pixels, weights are applied to the surrounding pixels in a 3 x 3 window along each row and column of the image. As shown in Figure 2.4 and Figure 2.5, different weights can be applied, depending on the edge detector used. The Prewitt edge detector applies a uniform weight to all neighbouring pixels to give them equal influence for the determination of an edge. The Sobel operator applies more emphasis to the pixels which are the horizontal and vertical neighbours to the centre pixel, by increasing the weight applied to these grey level values. These windows are used to compute the gradient mean of each pixel, which are then used to compute the gradient magnitude and the gradient direction (Parker, 2010). A single threshold value must be defined in order to determine the strength of an edge from the gradient magnitude and direction images.

Corners can be extracted using similar template-based approaches. For example, point detectors such as the Moravec operator measure intensity changes around a pixel by shifting a window in the eight cardinal directions and determining a point as a large intensity variation in all directions (Moravec, 1980). Harris and Stephens (1988) enhanced the Moravec point detector to include an auto-correlation function of the signal in order to measure local intensity changes and thereby developed the Harris corner detector.

An alternative approach to template-based edge detection is to use a derivative edge detector. Such an approach interprets the rate of change in grayscale, in order to determine an edge using the partial derivatives along each row and column of the image. An example of a derivative based edge detector is the popular Canny edge detector (Canny, 1986). Canny mathematically defined three criteria as part of the developed edge detector: (1) good detection – the possibility of detecting a false edge or failing to detect a true edge should be low; (2) localisation – features marked as edge points should be close to a true edge; and (3) only one response to a single edge – when there are two edges extracted, one of them must be distinguished as false (Canny, 1986).

The image being analysed for edge detection was presumed to be subject to noise, so a convolution filter smoothed the pixel values (Canny, 1986). An edge was detected as the local maxima in the response of the first derivative of a Gaussian function along the X and Y axes of the image (Canny, 1986). The strength of an edge was determined by calculating the gradient magnitude of a pixel compared to the neighbouring pixels, the same as template-based detection. However, compared to the approaches of Prewitt and Sobel, Canny utilises hysteresis thresholding to determine edges with a maximum and minimum threshold. If the strength of a potential edge was above the maximum threshold, an edge pixel was defined. If a pixel was connected to an edge pixel and the magnitude value was lower than the maximum threshold, but above the minimum threshold, then the pixel was defined as an edge pixel (Canny, 1986).

The Canny edge detector has been shown in several studies to outperform other edge detectors due to the defined criteria above (Nadernejad *et al.*, 2008; Juneja and Sandhu, 2009; Maini and Aggarwal, 2009; Bhardwaj and Mittal, 2012). Because of the high performance of the Canny edge detector, this has been adopted in several approaches to extract roof structure lines from aerial imagery. Various approaches were then subsequently applied to the edge detection results in order to remove any incorrectly extracted edges, referred to as false positives; lines that were extracted due to image noise or parameter sensitivity, which are not required for reconstruction. For example, Wang (2012) used the Canny edge detector to extract lines from 2D multispectral oblique and vertical images, as well as using the Moravec operator to extract points. The extracted points and lines from different images were matched to generate 3D feature points and remove false edges. Low image contrast meant some roof vertices and corners were not extracted, resulting in under-segmentation of the roof planes. Woo *et al.* (2008) extracted Canny edges to find 2D lines, which were grouped together using the Boldt algorithm to form 3D lines. This grouping stage connected fragmented edges and grouped parallel lines to remove false positives. Edges were also refined with a height threshold to distinguish between ground and non-ground areas (Woo *et al.*, 2008). By grouping edges based on parallel and perpendicular constraints, in order to connect fragmented edges into lines, reconstruction was limited to either flat or gabled roof shapes. Melnikova and Prandi (2011) also used Canny edge detection and obtained straight line segments using the Hough Transform. However, this process was severely hindered with a manual operator having to specify the areas for reconstruction. Kim and Nevatia (2004) extracted linear features from multiple images and used a DSM to remove detected edges not located near a building edge. Perpendicular linear segments within an angular threshold of 60° were then grouped together to form corners. Analysis of the edge detection results, from the above studies, was not investigated and the final reconstructed models were only presented for a small number of buildings, mostly in ideal cases: Wang (2012) presented results for five buildings; Woo *et al.* (2008) presented the reconstruction of five buildings and Melnikova and Prandi (2011) reconstructed seven buildings, which limits the applicability of these approaches to complex datasets.

Edge detection can also be used with raster-based surface models. The Canny operator was used by Kabolizade *et al.* (2012) to detect edges along the roof boundary from a normalised DSM (nDSM) created from lidar data. A nDSM is the surface difference between a DSM and a DTM. A generalisation algorithm was used to correct the extracted shape, by checking that the point density at the edge boundary was constant to remove short edges, and measuring the main orientation of the edge (taken from the longest length of the boundary) to generalise the extracted edges into parallel lines (Kabolizade *et al.*, 2012). The polygonal boundaries were generalised to be either rectangular, an L, T, Z or U-model (Kabolizade *et al.*, 2012). Jaw and Cheng (2008) created a raster DSM from lidar point clouds and applied the Canny edge detector to extract 3D edges. However, this was used in a semi-automatic approach as manual intervention was required to remove false positives and for the refinement of the roof geometry. The extraction of edges from surface models can yield a limiting area from the sharp discontinuity of the grayscale values between the edge of the building and the ground. However, finer roof detail such as roof structure lines cannot be extracted, as the difference in grayscale across a roof is not large enough for an edge to be detected.

The edge detection image output is a binary image of pixels, where edges are represented with a pixel value of 1 and all other pixels in the image are represented as the background and assigned the value 0. The only concern of the edge detector is the image values at the individual pixels, with no concept of the image content (Parker, 2010). Therefore, in order to extract more meaningful roof information from the edge detection, line tracing procedures have been implemented to determine single edges of the roof plane, corner points and to further remove false positives. Wang (2012) traced along edges to form smooth and straight lines using a split-and-merge operation and extracted corner points not detected with the Moravec operator. Edges and points were filtered based on the spectral properties of the features, which suffered from shadow and image contrast (Wang, 2012). After extracting straight lines, Melnikova and Prandi (2011) extracted corners as the intersection of two perpendicular line segments to determine roof corners, which as previously discussed limited the roof shapes that could be reconstructed.

Whilst these approaches show that corner can be defined from extracted line segments, the constraints applied are currently too restricting.

2.3.1.2 Area-Based Extraction

The alternative to feature-based extraction is area-based extraction, which segments different regions based on a homogeneous measure (Li and Guan, 2011). This is to classify the individual roof planes from the input data, which are used for 3D reconstruction. To segment a building, there are different approaches that can be undertaken. Global methods use a top-down approach by using all the available data to segment individual planes. However, depending on the complexity of the building, such an approach can be very time-consuming. Therefore, local (bottom-up) approaches have generally been favoured for roof plane extraction, which starts with a small subsample of the data and enlarges the region of interest.

An example of a local approach is region growing, which starts with a seed point and determines the segment by clustering points to the seed location. One approach to region growing is the Random Sampling Consensus (RANSAC) algorithm. RANSAC was originally developed by Martin and Robert (1981) as a method for extracting linear models from a dataset. This is an iterative process to estimate the parameters of a certain model. A number of points are randomly sampled to determine an initial model. The initial model is grown using neighbouring points if they are within the limit of some threshold, otherwise new points are randomly sampled to determine a new model (Martin and Robert, 1981). RANSAC has been developed for area-based extraction by randomly selecting three points as seed points and testing the remaining points within a specified distance to determine the coherence to the seed points (Yan *et al.*, 2012). Martin and Robert (1981) defined three parameters required for determining the best fitting model: (1) error tolerance to determine whether a point is compatible with the model; (2) the number of subsets to test and (3) the minimum number of points to determine a correct model has been found. These principles can, generally, be applied to area-based extraction.

Both global and local segmentation require a similarity measure to distinguish between neighbouring regions. Normal vectors, determined as the vector that is perpendicular to an object, have been used for segmentation. Normal vectors can be calculated by taking a subsample of the dataset and fitting a plane to the points. The distance of the centre point within the subsample to the fitted plane yields the normal vector. Kim and Shan (2011) used level set functions to initially divide a building into four regions, with each region then tested for further subdivision. Regions were separated based on the normal vectors, distance between the points and the angular difference of the normal vectors. A lidar point cloud was sampled to a 2D grid with a cell size determined by the average point spacing to ensure uniformity and distribution of points across the roof planes. The segmentation of planes was achieved by minimising an energy function of each level set using the normal vectors of the points (Kim and Shan, 2011). The process was conducted iteratively until convergence was reached and no more planes could be segmented. Roof-lines were determined as the union of neighbouring planes. Segmentation was undertaken on two test sites and the results were significantly influenced by the point cloud density (Kim and Shan, 2011). On the first test site, two buildings were under-segmented due to a coarse point cloud. Issues with small roof structures such as dormer windows were noticed in the second test site, with reconstruction errors again due to the point density on the input data.

Sohn *et al.* (2008) used a Binary Space Partitioning (BSP) tree to segment roof planes from a lidar dataset. A building was sequentially decomposed into the individual planar segments and recorded as a graph, with nodes representing the roof planes. Each branch from the node was a positive or negative region, determined from the direction of the normal vectors and edge correspondences. The first node represented the whole roof structure. This was decomposed into further nodes as the planar regions were segmented. A node was defined as closed if it could not be subdivided into further planes (Sohn *et al.*, 2008). The decomposition of the roof was continued until no more open nodes existed and the end of each unconnected branch was closed. As well as normal vectors, Sohn *et al.* (2008) also used height similarity to segment planes, but this resulted in small flat roof facets not being reconstructed due to the height difference threshold not being reached. The reconstruction of small roof facets was also hindered by the point density of the lidar dataset.

Jochem *et al.* (2009) measured surface roughness, defined as the standard deviation of the orthogonal fitting residuals of a plane, to segment roof planes. Planar segmentation was limited to a predefined minimum size threshold, which meant holes were present in the segmentation where points from chimneys and dormer windows were not classified. Errors were also seen at the boundary of roof planes, as points along the roof ridgeline were not classified due to the difference in the normal vector of ridgeline points to the roof planes. This resulted in inaccurate planar boundary extraction, which was extracted from the segmented planes using an alpha-shape (α-shape) procedure.

Normal vectors were also used by Dorninger and Pfeifer (2008) to segment roof planes from lidar point clouds via region growing. By merging clusters together, the building outline was computed. The outline was generalised using an α-shape procedure and an angular threshold.

Verma *et al.* (2006) determined roof planes by fitting local planar patches and measuring the surface normal, which resulted in jagged building roof outlines from noisy data. Forlani *et al.* (2006) classified a rasterised DSM grid from a lidar point cloud into separate planar surfaces. Buildings were initially clustered based on the height difference of a pixel with respect to its eight-connected neighbours. The gradient orientation of each pixel was also calculated, by fitting a least-squares fitted plane to the segmented regions, and combined with the height segmentation to extract roof planes. Segmentation results of these three methods were influenced by the input lidar data. A coarse point density meant short edges and small roof slopes could not be modelled. The non-uniform sampling in some areas also meant regions were not correctly segmented, particularly around surface edges where the boundary of two planes was incorrectly modelled.

Perera and Maas (2014) used the Hough Transform to segment a lidar point cloud, with a rule-based classification step to further classify planar segments. Rules included ridgeline detection, horizontality of the detected ridgeline and azimuth constraints. Lines were then determined from the segmentation and classified as ridgelines or step edges depending on the height differences (Perera and Maas, 2014). This extraction method extracted approximately 74% of roof planes, whilst ensuring incorrect extraction was smaller than 3%.

Maltezos and Ioannidis (2016) extended the selection of seed points for the Randomised Hough Transform to include the angle between the normal vectors and the height difference between points, as well as a distance criterion which was proven to be insufficient for planar segmentation alone. Segmented planes were refined based on point intensity, density, distance and angle to plane, as well as the removal of small planes (Maltezos and Ioannidis, 2016). Although the use of these additional constraints improved the results of planar segmentation and decreased computation time, small planes were still inaccurately extracted. The planar classification results were also compared to the results using the RANSAC algorithm, which achieved a slightly higher number of correctly classified planes and lower number of under-segmented planes. However, the number of non-segmented and over-segmented planes was reduced using the Randomised Hough Transform with the additional constraints compared to RANSAC (Maltezos and Ioannidis, 2016).

Goebbels and Pohle-Fröhlich (2016) converted a sparse lidar point cloud into a dense raster surface by interpolating the height and distance between points in order to reconstruct step edges. Planar segments of a roof were then clustered by calculating the angle based on the gradient value. The local minima and maxima of a histogram of calculated angles were determined to segment the data. Planes were accepted using the RANSAC algorithm if more than 80 points were present, which again highlights how small roof planes were ignored for reconstruction and the dependency on point density.

Area-based extraction can also be achieved from colour imagery using image pixels to segment regions. Avrahami *et al.* (2007) grew regions based upon the average and standard deviation of radiometric values around a seed point to build the outline of the roof. The boundary of the roof was refined using parallel and perpendicular constraints. The orientation of line segments was measured to find the building's main orientation and edges that were not parallel, perpendicular or of a length similar to the ground sampling distance (GSD) of the imagery were removed (Avrahami *et al.*, 2007). However, the approach was semi-automatic, requiring the operator to select buildings for reconstruction, with only 75% of buildings from the test area being selected. The results of a further 5% of buildings were extracted unsatisfactorily due to large sections of the buildings being missing. The proposed approach was found to only be suitable for flat roofed extraction. This was due to the use of coarse medium-scale imagery, meaning different roof planes were not clearly defined in either the image or the DSM.

Omidalizarandi and Saadatseresgt (2013) performed region growing using photogrammetric point clouds and images from a dense-image matching dataset. The procedure to create a dense-image matching dataset will be discussed in Chapter 3. Firstly, an image-based point cloud was segmented by calculating the surface normals and comparing the values within a neighbourhood. The nearest neighbour search algorithm measured the 3D Euclidean distance of points to the seed within a window. The size of the window increased if the number of valid points was below a threshold, until the number of points within the window was equal to or above a threshold. Therefore, *a priori* knowledge of the size of the planes was required. It was also defined by Omidalizarandi and Saadatseresgt (2013) that a region should consist of pixels with similar intensity values. Therefore, the process utilised the corresponding pixel intensity value of each 3D coordinate to refine the segmented regions. The combination of pixel intensity values from imagery and the corresponding 3D coordinates of the pixel, from dense image matching, were shown to assist the interpretation and recognition of objects. Another advantage found using image-based point clouds was that no information was lost when using a sampled 2.5D raster surface, which can be the case when resampling lidar point clouds (Omidalizarandi and Saadatseresgt, 2013).

Nex and Remondino (2012) performed region growing on an image-based DSM, grouping planes based on maximum gradient, depth variations and normal vectors for an initial classification. The results of the classification were refined by constraining segmentation to the principle orientation of the roof. Due to the noise in the image-based point cloud, over-segmentation occurred and results were influenced by long shadows in the extracted DSM. This was refined using the main gradients of each area to repeat the region growing process, by limiting segmentation to the main gradients (Nex and Remondino, 2012).

Bulatov *et al.* (2012) also clustered pixels from an image-based surface model using normal vectors. The outline of a building was determined using polygonal fitting of the nDSM objects with short edges removed to generalise the polygon. For all pixels within the boundary the normal vector was computed. However, no quantitative or qualitative analysis was undertaken for the reconstruction.

The success, or otherwise, of region growing approaches is heavily dependent on the location of the seed point and the calculation of the similarity measure. Results from Yan *et al.* (2012) showed some planes on a hipped roof were not detected due to the RANSAC algorithm extracting three points not on the same planar surface. If the seed is located in a bad location, this can lead to under or over-segmentation if different regions cannot be distinguished or have similar attributes, respectively. Some approaches have manually selected the seed points, such as Avrahami *et al.* (2007). Other approaches have been developed to automatically determine the seed location. Omidalizarandi and Saadatseresgt (2013) chose the point within a fitted plane which had the lowest square sum of residuals. Jochem *et al.* (2009) computed the surface roughness to determine seed points, chosen as the point with the lowest roughness value. Dorninger and Pfeifer (2008) determined seed clusters from the histograms of the computed normal vectors. None of the methods mentioned have been proven to outperform another, with regards to the selection of seed points; but the results all showed segmentation errors for the extraction of planes, which shows seed point location is critical to the results and the ideal location of seed points has yet to be fully resolved.

The location of seed points can be aided using predefined clustering values, such as the number of subsets to test. The number of subsets to test and segment can be defined using K-means clustering. This is another region growing approach, which partitions a dataset into *k* different clusters, thus limiting the number of regions that an object can be segmented into, to try and prevent over or under-segmentation. Sampath and Shan (2010) segmented a lidar point cloud into planes using calculated eigenvectors and eigenvalues, and then clustered the individual planes using the normal vectors to determine the direction of the roof. One of the limitations of K-means clustering is that prior knowledge is required to know the number of clusters and their approximate location. To overcome this, Sampath and Shan (2010) used an iteratively increasing neighbourhood to determine the planar potential of a point. The point with the highest potential was used as a seed location for K-means clustering and to determine the number of planes. Contrary to previously mentioned studies, Sampath and Shan (2010) found segmentation results were more successful with a sparse density point cloud, due to the increase in noise with a higher point density.

The segmentation of roof planes can also be achieved using cross-sections of the data, which are grouped together to form planes. This was originally proposed by Jiang and Bunke (1994), who determined the boundaries of planar regions from close-range imagery. The approach was subsequently extended for the classification of data captured from aerial acquisition (Hebel and Stilla, 2008). Alharthy and Bethel (2004) extracted roof planes by classifying cross-sections from lidar point clouds. To classify the cross-sections, the point cloud was resampled to a regular grid and a moving window was applied to each X and Y cross-section of the data. The slope and the height change along each row and column of the grid was sampled to determine the boundary of the regions (Alharthy and Bethel, 2004). Once the data had been classified, region growing procedures were implemented to cluster all cross-sections with similar attributes. The extraction of roof break-lines was straightforward for simple buildings, such as gabled roofs, where an orientation parameter of the roof lines could be introduced to extract planar boundaries similar to the dominant roof edge for each building (Alharthy and Bethel, 2004). However, for complex roofs, this constraint did not hold, so the boundary of each planar region was traced to vectorise the outline of the planes (Alharthy and Bethel, 2004). Errors were found at the boundaries of the planar regions, which were found to be disjointed and isolated, and required refinement to connect to the boundaries of neighbouring planes. Errors were concluded to be formed from the sparse point density. However, even with increased density, errors at the boundaries were still present when using cross-sections, due to noise in the data and segmentation errors (McClune *et al.*, 2014). He *et al.* (2013) extracted straight lines from each cross-section of a lidar point cloud using the *Iterative-End-Point-Fit* algorithm, with cross-sections grouped together to form planes, based on angular differences and geodesic distance. Shortcomings of the approach included under-segmentation and small planar structures being removed, from the use of a length threshold, which meant they were not reconstructed.

Region growing techniques such as the RANSAC algorithm have been quantified for the extraction of roof planes, leading to several conclusions (Yan *et al.*, 2012). The RANSAC algorithm was found to be sensitive to the number of points on the planar surface, the parameter setting to measure the distance of a point to a fitted plane, and the complexity of the building

(Yan *et al.*, 2012). The aforementioned area-based approaches were significantly influenced by noise in the data, as well as the spatial resolution of the data. The study by Yan *et al.* (2012) mainly investigated the distance of a point to a fitted plane parameter, testing some arbitrary values, and the complexity of the roof shape. More analysis was suggested to investigate point density and topology relationships between planes (Yan *et al.*, 2012).

2.3.1.3 Feature and Area-Based Extraction Combination

The utilisation of ancillary data, such as building footprints, can benefit the segmentation of roof geometry and the data extracted from area-based and feature-based extraction can be complementary. Area-based methods, for example, can struggle to extract the boundary of regions due to irregular point spacing (Bretar, 2008; Zhang *et al.*, 2014). By contrast, feature-based approaches are sensitive to image noise and spatial resolution, and can extract many false positives from image-texture and occlusions, or miss edges due to shadow (Maini and Aggarwal, 2009). By combining the two approaches together, feature-based methods can be used to determine the boundary of area-based extraction, whilst area-based extraction can remove false edges from feature-based extraction. Zhang *et al.* (2012) extracted corners from imagery and planes from lidar point clouds to compute the 3D coordinates of corners. However, this method required manual intervention at several stages, including the segmentation of the compute tbuilding footprint to simplify the roof primitives and the extraction of primitive geometry from imagery and point cloud. This therefore rendered the method very manual intensive. Demir and Baltsavias (2012) refined the boundary of planar segmentation with extracted Canny edges by assigning straight lines to the corresponding planar segments, extracted from a lidar point cloud using normal vectors. The minimum size of a plane was determined as 4 m² and by multiplying this by the point density, it was determined that a minimum of 20 points were required for each planar feature (Demir and Baltsavias, 2012). A comparison was made between the planar extraction from just lidar and when combining edges detected from imagery with the planar segmentation. It was found that the average absolute distance of a plane to the point cloud was reduced by 0.09 m and the accuracy of the XYZ corner coordinates also improved by combining feature and area-based extraction (Demir and Baltsavias, 2012)

A similar approach was implemented by Zhang *et al.* (2014) who extracted the convex hull of a RANSAC planar segmentation and combined this with Canny edge detection to extract straight line segments through a RANSAC fit of the edge pixels and planar points. Any plane or edge not detected was extracted manually. Corners were then extracted from the intersection of lines, whilst any corner not detected was extracted manually. Thus, this procedure was only semi-automatic.

Awrangjeb *et al.* (2012) extracted edges from orthophotos using the Canny edge detector and defined straight lines by applying a least squares straight line fitting procedure to extract corners from the intersection of edges. Edges shorter than 1 m were also removed. The lines were classified as a roof edge or roof ridge using normalised difference vegetation index (NDVI) and entropy measures from imagery, as well as a ground mask obtained from the lidar point clouds (Awrangjeb *et al.*, 2012). However, registration errors between the lidar and imagery datasets caused an offset and resulted in classification errors.

Li and Guan (2011) classified roof planes using the RANSAC algorithm and determined seed points using a distance threshold. The boundary of the building was refined using edge detection from imagery and matching detected edges to the boundary of the building, extracted from lidar classification. The boundary lines were further refined using a least-squares template matching with orthogonal constraints to only extract rectangular buildings. Perpendicular constraints were also applied by Cheng *et al.* (2011). Edges were extracted from imagery, using the Edison edge detector, to calculate the two principal orientations of a roof. Edges were separated into boundary lines and non-boundary lines by measuring the height value of points either side of the edge. Roof planes were then extracted using the RANSAC algorithm, with planar intersection determined for the extraction of ridgelines. Both these methods could only reconstruct buildings with a rectangular footprint, thus limiting reconstruction.

Dal Poz and Fernandes (2016) refined the boundaries of planar extraction, from lidar data, with edges detected from imagery. Edges were extracted using the Canny edge detector and the Steger line detector within a defined bounding box of the planar edges; with the edge detection results compared against each other for false positive removal. Edges were compared based on proximity, orientation and length to find the edge that minimised an energy function (Dal Poz and Fernandes, 2016). Registration errors between the two datasets meant that the bounding box had to be increased around the planar boundaries, which increased the number of edges extracted. While the results were shown to successfully overcome this registration error, results were only presented for two simple hipped roofs, which does not show the transferability to other roof structures and more complex examples.

Fan *et al.* (2014a) segmented a roof into 20 equal interval height bands, with the highest height band determined as the top of the roof if more than 50 points were classified. A ridgeline was formed for the highest height band using the RANSAC algorithm, with points assigned to the ridgeline if their distance was less than twice the average point spacing. By using height bands to extract lines, only horizontal line segments could be formed, with sloping edges of pyramid roof, for example, not extracted. To classify the roof planes, seed points were extracted as points that were perpendicular to the classified ridgeline and within a 2 m buffer (Fan *et al.*, 2014a). A plane was then fitted through the seed points using the normal vector. The remaining points were added to the plane if the residuals to the fitted plane were below a defined threshold and the distance of the points from the plane was below 0.2 m to overcome noise in the point cloud (Fan *et al.*, 2014a). Any cluster of points not extracted from the seed points were segmented using connectivity analysis, as long as there were more than 50 points in a cluster, and adjusted to the point clusters from the ridgeline (Fan *et al.*, 2014a). This helped overcome the limitation of only horizontal lines being extracted, but was only able to segment hipped roofs because a horizontal ridgeline was required for the seed points. Flat roofs and roofs that did not have a central ridgeline, such as pyramid roofs, could not be segmented using this method. Segmentation errors were also noted for similarly sloped roofs that were at a step edge, gable roofs with a low inclination angle, as well as boundary errors between neighbouring planes.

To reduce the complexity of roof geometry extraction, additional data can also be integrated. Building footprints have been determined to aid the segmentation of roof planes by giving information on the structure of the building, the location of building walls, as well as limiting the search area for cue extraction. These footprints can also be been used to decompose the whole building into smaller segments. Vosselman and Dijkman (2001) segmented building footprints by extending edges at concave corners to reduce over-segmentation using a Hough transform. Vallet *et al.* (2011) split a building footprint along edges that followed the main direction of the building, by minimising an energy function, and then merged any incorrect splits by minimising an alternate energy function. A DSM was also used to compute height discontinuities between roof planes and further aid the split and merge approach (Vallet *et al.*, 2011). Fan *et al.* (2014b) used OpenStreetMap building footprints to extract buildings from a point cloud, but found offsets in the data due to the oblique perspective of the imagery used to digitize the footprint, to be different to the point cloud. To overcome this, a 5 m buffer was applied to the building footprints. Whilst building footprints can be useful for the segmentation of the data, they are not always available and the segmentation of the footprints does not necessarily define each of the roof regions.

To try and overcome issues of footprint partitioning not representing the roof planes, Xiong *et al.* (2016) proposed two different approaches: one using roof contours and another using step edges. The first approach calculated height layers of the roof using the 2D α-shapes algorithm.

The layers were then simplified using the Douglas-Peucker algorithm. The height bands were snapped to the corners of the building footprint to partition the building. The second approach found step edges of the roof by classifying a lidar point cloud into separate heights and combined the edges with partition lines of the building footprint. The partition lines and step edges were snapped together if they were below a distance apart and parallel (Xiong *et al.*, 2016). The results showed promising segmentation of a building footprint; the first method was able to identify small features and the second method followed the buildings dominant direction, thus producing regular alignment of segments (Xiong *et al.*, 2016). It was concluded that future work should look to utilise the two approaches together in order to try and achieve 100% success of segmentation (Xiong *et al.*, 2016).

2.3.2 Reconstruction of Roofs from Extracted Modelling Cues

Once primitives have been extracted, the 3D building model can be reconstructed. There are different approaches that can be taken for 3D reconstruction and a building can be reconstructed at varying levels of detail, as explained in Section 2.2.1. The reconstruction of simple flat roofed models, defined as LOD1 by CityGML, can be easily automated by extruding a building footprint, typically available from NMCAs, to a single height (Oude Elberink *et al.*, 2013; Sargent *et al.*, 2015). Other researchers have adopted similar approaches, for example Arefi *et al.* (2008) applied an average height of lidar points from within an extracted 2D building polygon to construct 3D buildings.

However, the automatic reconstruction of building models with the correct roof geometry is much more challenging due to the complexity of the varying roof geometry. Attempts have been made to automate the reconstruction of roof geometry using two approaches: model-driven and data driven reconstruction.

2.3.2.1 Model-Driven Reconstruction

Model-driven approaches aim to simplify reconstruction by utilising a predefined library of roof shapes, which are fitted to the extracted data in order to reconstruct 3D geometry. The simplest approach to model-driven 3D reconstruction is parametric modelling, which describes the shape of the buildings by a fixed set of parameters (Forstner, 1999). This approach is able to reconstruct simple roof structures by a minimum number of parameters such as length, width and height. Photogrammetric software packages, such as BAE Systems SOCET GXP, use parametric models to reduce the number of points required for reconstruction (BAE Systems, 2011). An operator manually extracts the required number of primitive points, shown for a flat roofed and gable roofed model in Figure 2.6, from stereo-imagery to reconstruct a 3D building

(BAE Systems, 2011). For a flat roof, four points are required for reconstruction: one on the ground and three on the roof surface to indicate the height from the ground, the length and width of the building. For a gable roof, an additional roof point must be extracted at one end of the ridgeline to indicate the height from the bottom of the roof to the top of the ridgeline and the length of the gable edges. The software then interpolates the position of the opposite ridgeline endpoint using the length and width of the building. This workflow is also applicable to hipped roofs, with the software interpreting the distance of the hipped roof point from the roof boundary.

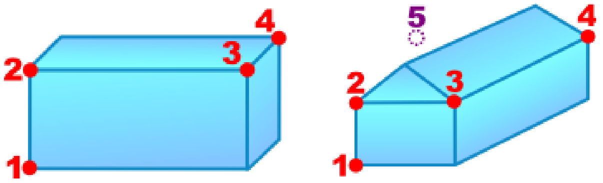


Figure 2.6 – Roof extraction points required for 3D parametric reconstruction using BAE Systems SOCET GXP photogrammetric software package: (left) a flat roof; and (right) a gable roof (BAE Systems, 2011).

Other types of parameters can be extracted for model-driven reconstruction. Zhang *et al.* (2014) derived translation, rotation and shape parameters of buildings. The number of shape primitives for reconstruction was determined based on the number of planes extracted from the RANSAC algorithm; for example, if two planes were extracted, the primitives of a gable roof were determined (Zhang *et al.*, 2014). By determining the primitives from the number of extracted planes meant some complex roof shapes could not be reconstructed. Parameters for the height from the ground and internal roof angles were determined by Kabolizade *et al.* (2012) to parameterise model reconstruction. A model was fitted to the data by iteratively improving a genetic algorithm, which minimised the associated costs of fitting a model to the data (Kabolizade *et al.*, 2012). Due to high complexity of the data, a large number of iterations were required, which increased computation time.

Roof models were also determined by minimising a cost function by Henn *et al.* (2013). Models were fitted to points within a rectangle, from decomposition of the building footprint, using the RANSAC algorithm to determine the model parameters. Several constraints were determined prior to model selection, such as all planes of a pyramid roof having the same inclination angle, and the ridgeline of a symmetrical roof being horizontal and perpendicular to the longest/shortest edge, in order to reduce the number of parameters to be extracted for reconstruction (Henn *et al.,* 2013). The building footprint also fixed the length, width and orientation parameters of the building. The first step of model selection was to determine whether a single plane or multiple planes had been extracted, which was further modelled to determine if a single plane building was a shed or flat roof, or a multi-plane roof was a gable, hipped or pyramid roof (Henn *et al.*, 2013). By fixing constraints before reconstruction, the number of parameters and points required was reduced. For example, two parameters were necessary for a symmetric gable roof reconstruction: the inclination angle (which was fixed for both planes), and the height of the roof, which could be determined by two points (Henn *et al.*, 2013). The best fitting model was then selected by supervised machine learning. The classification results of the roof showed high accuracy of flat roof reconstruction, but as the complexity of the roof increased, the level of accuracy decreased, with only 69% of hipped/pyramid roofs correctly classified (Henn *et al.,* 2013). Reconstruction results were also influenced by the building footprint, with incorrect decomposition leading to incorrect reconstruction. The reconstruction approaches of Zhang *et al.* (2014), Kabolizade *et al.* (2012) and Henn *et al.* (2013) were all restricted to the reconstruction of buildings with orthogonal footprints.

Simple roof models can be combined to reconstruct more complex buildings, known as Constructive Solid Geometry (CSG). Tseng and Wang (2003) proposed a reconstruction approach which split an object into simpler primitives for model-based reconstruction, and then reconstructed the whole object using Boolean operators, such as union and intersection, to combine primitives. The approach developed by Tseng and Wang (2003) was a semi-automated process with a user required for several steps: dividing the building into parts, specifying shape and pose constraints for the reconstruction and combining primitives.

Huang *et al.* (2013) defined the sections of a roof models as a combination of the positional (X, Y, azimuth), contour (length, width) and shape (height of ridge/eave lines, depth of hip) parameters. Roof models were combined using a number of rules including: an intersection angle of 90°±5° between models, heights of neighbouring corners were harmonized if within 0.25 m range and the merging of primitives with angles smaller than 5° (Huang *et al.*, 2013). A building was split into primitives and the roof shape of each primitive was, firstly, modelled as a gable roof. The model was then refined by making adjustments using the Reversible Jump Markov Chain Monto Carlo algorithm (Huang *et al.*, 2013). The fit of each model was assessed with the average absolute deviation in Z between the model and the data points (Huang *et al.*, 2013). However, the proposed reconstruction method was only successful on scenes with low complexity and where buildings were not close to each other. In complex scenes, building candidates could not be found. The low point density also produced errors in the reconstruction of small structures.

To automate the subdivision of a building, ancillary data has been used. Lafarge *et al.* (2010) segmented a building into smaller segments at the height discontinuities of the building footprint. A model was assigned to each segment by minimising a likelihood algorithm consisting of several parameters such as rooftop height, position and pitch, as well as prior information such as orientation compatibility to ensure the correct connectivity of neighbouring models (Lafarge *et al.*, 2010). However, models were reconstructed with a RMSE height error of 1.3 m (Lafarge *et al.*, 2010). Building footprints were also segmented by Kada (2009) for the reconstruction of the Berlin city model, as discussed in Section 2.1. Building footprints were used in two approaches for reconstruction. Firstly, flat roofs were collected using semantic attributes for the building's use, and then the building footprint was extruded to the average height of the lidar point cloud within the footprint (Kada, 2009). For all other roofs, the building footprint was partitioned into non-intersecting and parameterised primitives to give quadrilateral cells using a process termed cell decomposition. The building footprint was split to find parallel lines, which were a set distance apart and had an angle between their directions below a threshold of 15°. The normal vector was computed within each cell to compare with roof models, by measuring the percentage of points fitting the specific roof plane and determining the roof shape automatically (Kada, 2009). However, if no roof shape was found, the roof plane in the cell was manually reconstructed. Manual intervention was also required when the footprint did not decompose into regular cells.

Suveg and Vosselman (2004) constructed buildings as CSG models from aerial photography by segmenting a building footprint into smaller primitives. Corners and edges were matched based on labels, epipolar geometry, and height to form 3D features for reconstruction, which were then used as model parameters to fit predefined roof shapes. The correctness of the fitted model was determined using mutual information matching criteria between the object model and the images (Suveg and Vosselman, 2004). This metric combined information from the model contours, where large-image gradients were found, and the similarity of the image intensity. To form complex roof shapes, the models from the smaller primitives were combined by implementing geometric constraints, which included the models sharing a common edge or corner and sharing a similar parameter (e.g. same orientation) (Suveg and Vosselman, 2004). Aerial photographs were scanned to form digital representations, and errors in reconstruction were caused by the granularity of the scanned images.

Another approach to simplify the roof geometry is through the use of Region Adjacency Graphs (RAG), also referred to as Roof Topology Graphs (RTG). This was first proposed by Verma *et al.* (2006) to model the connectivity of roof planes and allow models to be determined for reconstruction. The nodes of the RAG sit in the middle of each roof plane and the edges represent the intersection between the roof planes. An example of a RAG can be seen in Figure 2.7. Surface normals were calculated to categorise three types of roof plane connectivity: planes that were orthogonal with surface normals pointing away from each other; planes that were orthogonal with surface normals pointing towards each other; or planes that were parallel with surface normals pointing away from each other (Verma *et al.*, 2006). Models were created for a predefined library of graph structures. To reconstruct the 3D geometry, the graphs of the connected planes were matched to the graphs of the models. Whilst the approach of Verma *et al.* (2006) reconstructed simple buildings successfully, some more complex buildings were not reconstructed due to the number of primitives in the library.

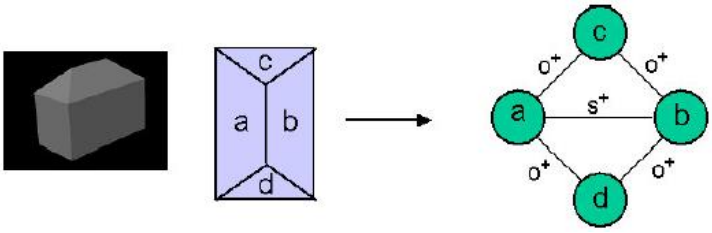


Figure 2.7 – An example of a Region Adjacency Graph (RAG) (Verma *et al.*, 2006).

Models created using RAGs can also be combined to reconstruct complex roof models, similar to the CSG approach. Oude Elberink and Vosselman (2009) determined the connectivity between different roof planes such as same/opposite normal direction, convex/concave and whether the planes were tilted or horizontal in order to create RAGs. A library of target shapes, of various interactions between roof planes, was then modelled onto the segmented data. Intersection lines were extended to form connections at the intersection of two or more planes. These connections were made based on the intersection of planes, the height difference of points and the extension of intersection edges to the building footprint (Oude Elberink and Vosselman, 2009). Incorrect models were reconstructed for several buildings due to the minimum segment size parameter and assumption of equal eave-line heights. Reasons for roof models not reconstructed included segmentation errors, the absence of topologic relations between planes and limitations of the target model (Oude Elberink and Vosselman, 2009).

Xiong *et al.* (2014) developed the reconstruction method of Oude Elberink and Vosselman (2009) by introducing a graph editing dictionary, to determine whether the model had been successfully reconstructed. A poorly reconstructed model was defined as an area larger than 3 m² and a surface that deviated by more than 0.3 m from the lidar points. The graph editing dictionary consisted of models, which could be introduced to the RAG to refine reconstruction. Errors in the roof models were found by comparing the distance between a fitted plane and the points, as well as the gradient along ridgelines, defined as the ratio of the number of points of one segment on both sides of the line (Xiong *et al.*, 2014). The RAG was improved by iteratively minimising an energy function to improve the selected model. However, by adjusting one part of the RAG meant that previously correct sections were sometimes adjusted incorrectly, with results for one test site showing a decrease in the completeness of roof plane extraction when using the graph edit dictionary, compared to the original results (Xiong *et al.*, 2014). Problems also occurred with point density, and if a roof consisted of a number of small roof planes, with segmentation errors occurring. As will be shown in Section 2.4, issues can arise when measuring the distance of fitted models to the data.

The approach of Xiong *et al.* (2014) was developed further by Meidow *et al.* (2016) to enforce constraints onto the data; in order to refine incorrect reconstruction by testing identity, verticality, orthogonality, parallelism and concurrence (where four planes meet) between reconstructed roof planes. Whilst these constraints worked successfully when tested on a small subset, it was determined that the number of constraints should be increased (Meidow *et al.*, 2016).

2.3.2.2 Data-Driven Reconstruction

Compared to model-driven approaches, data-driven approaches use the extracted cues to reconstruct the building model, whilst minimising any prior knowledge of roof geometry (Sohn *et al.*, 2008). These approaches are thus not limited to any predefined shapes. Constructed geometry can be categorised as a generic model, which allows much greater variation in the reconstructed model (Forstner, 1999).

One approach to data-driven reconstruction is to model a 3D building using Boundary Representation (B-rep). The roof is modelled by a series of surfaces, edges and vertices which form a labelled planar graph describing the relationship of planar boundary edges (Rottensteiner, 2000). An edge consists of two neighbouring vertices (a start and end point) with a closed cycle of edges forming a surface. In order to reconstruct a B-rep model, the position of the roof corner points and the connectivity of the points are required. B-rep models were produced in the CyberCity-Modeler (CC-Modeler) software, developed by CyberCity AG in Zurich (Ulm, 2003). An operator manually digitised all roof points into boundary points and interior points. This labelling of points produced a topology graph, where each point of the roof was expressed as a node and each roof plane was represented as a closed cycle of nodes. To determine roof faces, geometric constraints were enforced to ensure that: (1) a 3D object was a closed multi-plane object; (2) planes were not supposed to penetrate each other; and (3) every two adjacent boundary points were always part of a face (Gruen and Wang, 1998).

B-Rep models can also be constructed from the intersection of area-based extracted planes. Peternell and Steiner (2004) determined edges from the intersection of planar segmentation results and corners from the intersection of three planes only. The gradient between neighbouring planes that did not intersect was also calculated to determine jump edges (Peternell and Steiner, 2004). A closed boundary was then formed for each plane. However, no experimental results were presented to define the success of the reconstructed model.

Zhou and Neumann (2008) created B-rep models of various buildings by creating a uniform grid from planar segmentation and outlining the building as a watertight polygonal boundary. Edges were conformed to principal directions, determined from peaks in the histogram of calculated direction angles. However, this approach was only able to automatically reconstruct flat roof models, with user intervention required to model the roof shape.

A similar approach was undertaken by Kim and Shan (2011) to determine the structure points of a roof based on the intersection of roof planes. If only two roof planes intersected at a specific location, this was defined as a roof ridgeline. Only the intersection of three or more non-parallel planes at a specific location could be determined as a roof structure point. The intersection of two planes with the roof boundary, which was formed from the planar segmentation boundary using the α-shape algorithm, was also included to determine roof boundary points (Kim and Shan, 2011). The results of 3D reconstruction were found to suffer from over-segmentation and limited by the parameters chosen. The boundary of the building was delineated by Dorninger and Pfeifer (2008), who also used the α-shape algorithm. Results were determined to be sensitive to the value of α, so post-processing was required to regularise the building outline. The angular direction of lines was measured, with lines combined if the angular deviation was smaller than a threshold (Dorninger and Pfeifer, 2008). Lines were conformed to be orthogonal or parallel to the longest boundary line segment. Building models were then reconstructed by finding the intersection of planar segments with each other, between a planar segment and the boundary, or to define a step edge (Dorninger and Pfeifer, 2008). The boundary of each planar edge was determined by selecting the connected path of intersection edges, based on the position of the edges with respect to the points defining the roof segment, and the length of the edges (Dorninger and Pfeifer, 2008). If no edge was determined by intersection, the α-shape algorithm was used to determine a step edge. Some of the results presented required the operator to manually digitise the ridgelines, with reconstruction errors then manually rectified.

Sampath and Shan (2010) determined the adjacency of segmented roof planes by calculating the distance between points from planes. If the distance between the points was less than twice the point spacing of the point cloud, then the planes were considered to be neighbours (Sampath and Shan, 2010). The coordinates of the roof vertices were determined using boundary constraints of neighbouring planes, with interior roof vertices defined by at least three planes and the breakline by the intersection of two planes. It was determined that the boundary corners were reconstructed less accurately then the interior roof corners because there were formed without the defined planar intersection rules.

Alharthy and Bethel (2004) converted the segmented raster plane boundaries into vector format by extracting lines from the region boundaries, whilst determining segment adjacency to join edges and determine a B-rep model. As a final check, a 3D regression was applied to the lidar points within each delineated area to estimate inclination and height intercept parameters (Alharthy and Bethel, 2004). Errors occurred during the connection of ridgelines as the corners were shifted from the true corner location to enforce connectivity and the failure to reconstruct small roof planes, which were incorrectly segmented (Alharthy and Bethel, 2004).

Corner points can be determined using the shortest closed cycle from a RAG, which is the shortest path from a plane through its neighbouring planes and back to the start plane (Perera and Maas, 2014). Edges were split to determine step edges, and the boundary of the roof was determined by an outermost closed cycle, which is similar to the shortest closed cycle (Perera and Maas, 2014). This approach required several manually defined parameters based on knowledge of the scene, with some of the parameters set differently between test sites, thus potentially limiting transferability. Forlani *et al.* (2006) determined a RAG and classified interior nodes, as the intersection of adjacent roof slopes, and exterior nodes, as the intersection of two adjacent contour segments with a roof slope, from segmented planes. Once the adjacency between planes was determined, edges were labelled as external edges (connecting two external nodes), internal edges (connecting two internal nodes) or hips, which connect an external node to an internal node. The results of this reconstruction were highly unsatisfactory due to the complexity of the buildings and incorrect classification caused from low spatial resolution of the input data (Forlani *et al.*, 2006). One reason noted for the failure of the approach by Forlani *et al.* (2006) was due to no refinement being conducted of the B-rep model to ensure symmetry and orthogonal constraints. Approaches which have introduced such constraints, have shown a positive effect on the quality of the reconstructed models, with an improvement between 5 – 45% on correct reconstruction (Oude Elberink and Vosselman, 2011).

Constraints such as symmetry, parallelism and regularity, which have been used in several model-based reconstruction approaches, were also suggested in the conclusion of Nex and Remondino (2012) for data-driven reconstruction. The approach developed by Nex and Remondino (2012) smoothed extracted edges to create straight lines and applied constraints to ensure coincidence of corners. Roof edges were reconstructed as 3D polylines. The results were found to be significantly influenced by the quality of the input DSM, with errors in the segmentation caused by noisy data and reconstruction errors were found for highly complex buildings (Nex and Remondino, 2012).

Reconstruction constraints, such as those suggested above, have been implemented to varying degrees of success. Woo *et al.* (2008) determined corners as the intersection of two line segments which formed an angle between 80° and 100° to form perpendicular edges. The 3D roof was then reconstructed as a connection of corner points which formed edges, and thus formed a surface. However, the strict reconstruction of corners at 90° ± 10° meant that reconstruction was limited to rectangular buildings with either flat or gabled roofs. Melnikova and Prandi (2011) attempted to build upon this work in order to reconstruct hipped roofs where three corners formed a triangle, but failures of the Canny edge detector in not being able to extract ridgelines from low contrasting and low spatial resolution imagery meant this was not possible. Nizar *et al.* (2006) found the boundary of segmented planes, and then applied parallel and perpendicular constraints between line segments, which helped overcome limitations in planar boundary extraction from lidar point clouds, in order to reconstruct a B-rep 3D model. Engels *et al.* (2008) used topological and geometrical rulesets on three levels to limit the number of edges extracted from planar intersection. The first level defined a graph of edges where every node had to be connected to at least two other nodes and form a planar graph. Level two defined each face of the planar graph as a directed graph of edges, which enforced constraints that an edge could not appear twice for any one plane, but each connection must be in the boundary of two faces (Engels *et al.*, 2008). The third level defined the geometrical relationships of faces, which prevented faces intersecting and only being connected at a node (Engels *et al.*, 2008). However, planar segmentation errors meant some roof planes were not reconstructed and further development of the topological constraints was required. Fu and Shan (2008) limited exterior and interior angles to be either orthogonal or obtuse, as well as enforcing ridgelines to be parallel to the boundary to reconstruct a planar graph from manually digitised roof points. The use of such constraints can ensure buildings are reconstructed with regular geometry, but can also limit the types of roof that can be reconstructed. The constraints described can be applied to reconstruct simple roof shapes such as gable roofs, but are not easily as transferable to complex roof shapes.

Other reconstruction approaches from feature-based approaches include Rau (2012), who developed a TIN-Merging and Reshaping algorithm to refine a Triangulated Irregular Network (TIN) based on manually derived structure lines. The first step was to merge any triangles that shared a common border but did not cross a structure line. Reshaping then used the height at the endpoints of a structure line to classify lines as independent or shared edges. Independent edges were categorised as edges which do not share a neighbouring polygon, whilst shared edges bordered two or more planes (Rau, 2012). Results were dependent on the manual extraction of roof edges and occlusion caused several of these edges not to be extracted.

2.3.2.3 Comparison of Approaches

Research has been undertaken to compare model-driven and data-driven approaches. The conclusion by Tarsha-Kurdi *et al.* (2007) was that the building footprint was reconstructed more accurately, with respect to length, using data-driven approaches, but the height and the angle of the roof planes was reconstructed more accurately with model-driven approaches. The advantages of model-driven approaches included high-computational speed compared to data-driven approaches, producing models with low deformation errors and low probability of reconstruction failure due to the reduce number of parameters required. Although model-driven appeared to have more advantages over data-driven, no conclusion could be made as to which method was more accurate with model accuracy related more to the extraction technique used (Tarsha-Kurdi *et al.*, 2007). The reconstruction by Tarsha-Kurdi *et al.* (2007) was limited to only gable roofs, which are very simple roof structures and present in nearly all roof library primitives. Data-driven methods were found to be more flexible with the reconstruction complexity, as complex roof shapes are not always available in model-based libraries (Tarsha-Kurdi *et al.*, 2007). Dorninger and Pfeifer (2008) compared their data-driven approach to a model-driven approach and found the model-driven approach produced greater roof plane completeness (described in Section 2.4) but significant structures were neglected.

Fan *et al.* (2014a) combined model and data-driven approaches for reconstruction of gable roofs, which could combine models to form more complex roof shapes. The extracted ridgelines were used as *a priori* information to determine the position, angle and size of the gabled roof, whilst the point segmentation was used for the data-driven approach (Fan *et al.*, 2014a). Although simple roofs were successfully reconstructed, the method struggled to reconstruct planes with a small height gradient, due to the height band segmentation. These buildings were, thus, reconstructed as flat roof models (Fan *et al.*, 2014a). Dormer windows were also not reconstructed due to the low point density in these areas being below the required threshold.

Dahlke and Linkiewicz (2016) compared two data driven reconstruction approaches from imagery datasets: one from a 3D image based point cloud and the other from the 2.5D raster based surface model. The first approach split the point cloud into cells to calculate local regression planes and merged the cells together which were coplanar and with similar normal vectors (Dahlke and Linkiewicz, 2016). This approach under-segmented the reconstructed roof, due to no difference being measured between the main roof planes and large dormer windows. The second approach extracted lines from a raster based surface model using a fast line segment detector, and the orientation and slope of the lines were measured using the 3D coordinates of the line endpoints. The results were presented in a histogram and a polynomial function of degree 2 was fitted to the peaks of the histogram to enforce orthogonality of the extracted lines (Dahlke and Linkiewicz, 2016). However, this limited reconstruction to simple roof structures with edges at approximately 90° to each other, for example gable roofs. These edges were then use to aid planar segmentation of the data, with the roof topology calculated from the intersection of planes (Dahlke and Linkiewicz, 2016). When compared against one another, the second approach was found to be more successful; with the added information from edges extracting more corners and to a higher positional accuracy.

2.4 Validation

Once a building has been reconstructed, the model should be validated to determine the accuracy and quality correctness. However, often the absence of reference data means that this is not always possible and the validation of 3D building models is a topic with limited research. Most of the research undertaken in 3D reconstruction has focused on deriving a successful technique for roof modelling, compared to clearly defining its objectives and producing a consistent and complete building model. A few approaches have been suggested for quantifying 3D buildings, which have their limitations.

When reference data was not available, some methods measured the fit of the model against the input data. The planar difference and distance of a reconstructed roof plane to the input lidar point cloud has been measured to quantify reconstruction in many approaches (Dorninger and Pfeifer, 2008; Akca *et al.*, 2010; Xiong *et al.*, 2014). However, this can be misleading. Small residuals are produced between the laser points used for the extraction of the modelling cues and the reconstructed model; but this measure does not determine whether the roof model has been correctly reconstructed (Oude Elberink and Vosselman, 2011). Visual analysis has been used to manually analyse the correctness of the reconstructed building, but this is subjective and lacks quantifiable results (Durupt and Taillandier, 2006). CityGML defines geometric properties of the reconstruction, which are quoted by research to define LOD2 reconstruction if a building has been modelled with roof geometry. However, the lack of reference data can mean the defined absolute plan and height accuracies were not measured (Henn *et al.*, 2013). Problems have been outlined with this standard, as discussed in Section 2.2.1. The measures of the corner point position and height accuracies have been calculated when reference data was available (Rottensteiner *et al.*, 2014). Other measures include the length and height of a building, as well as the inclination angles of the pitched roof planes (Kaartinen and Hyyppa, 2006).

An alternate approach is to measure the segmentation of the buildings by comparing the overlap between roof planes from a reference dataset and those modelled. Rutzinger *et al.* (2009) defined measure the completeness, correctness and quality of roof planes by measuring overlap with reference data. This validation measure is subjective and requires a user-defined amount of overlap to determine whether a region has been successfully segmented. In the ISPRS benchmark on 3D reconstruction, this was defined as 50% overlap between the reference and model planes (Rottensteiner *et al.*, 2014). Similar measures were used by Dorninger and Pfeifer (2008), who evaluated the quality of the reconstruction based on individual building parts. Planar regions from the segmentation were checked to determine if they were represented by closed, polygonal boundaries. The area of each roof plane was compared to the area of the polygonal model boundary. The heights of the points defining the segment were also compared.

Wong and Ellul (2016) proposed six metrics to investigate the accuracies of 3D models, which measures the mean number of features/edges/faces per building, the mean number of vertices per face, the frequency distribution of 2D footprint area and the frequency distribution of the features length. It was found that the metrics produced an indication for the complexity of the building models but did not measure absolute geometry accuracies and could not readily compare models from different areas due to architectural variation.

Although a few methods have been proposed, there is currently no defined standard as what constitutes a successful 3D reconstruction. Current methods, such as visualisation and completeness and correctness are subjective approaches requiring an operator to determine the limits of successfulness. Measuring the RMSE of corner position and height can yield the accuracy of the points, but this is again subjective with a user required to determine how close an extracted corner must be, to be measured. This level of accuracy can be taken from standards such as CityGML, but several issues have been noted with the current implementation of the standard. There is also still research being conducted to investigate the required accuracy of models, outlined in Section 2.2.1. The validation of building models is, therefore, still a widely-researched topic area.

2.5 Summary

This chapter has presented the need for 3D building models and explained why the automatic reconstruction of 3D building models from aerial data acquisition is such a heavily researched topic. Many different types of building models can be reconstructed for various purposes. However, for current applications, which use large scale urban models of many buildings, details such as overhanging roofs, doors and chimneys, modelled in LOD3 and LOD4 are not necessary and cause an increase in computational duration (Piepereit *et al.*, 2016). This is why studies have focused on the reconstructed of building models to LOD2 and below. Methods for the automatic reconstruction of block-like buildings with no roof geometry, as described by LOD1 of CityGML, have been successfully developed from research and are now routinely being implemented into the workflows of NMCAs using pre-existing 2D building footprints. Many proposed approaches are also able to automatically reconstruct very simple roof geometry, such as a gabled roof. However, the automatic reconstruction of more complex 3D roof geometry is much more difficult, with various methods to achieve such still under investigation. Currently, only semi-automatic methods are readily available because of issues with the extraction of modelling cues and reconstruction.

Automatic 3D building reconstruction is dependent on the extraction of modelling cues, which is dependent on the quality of the input data used and, usually, several user defined parameters to compensate for noise in the data. The spatial resolution of the imagery used will influence feature-based extraction. If the spatial resolution is too low, ridgelines may not be detected, whilst imagery with higher spatial resolution can result in roof texture being extracted, which may hinder reconstruction. Gaps in the input data can lead to under-segmentation of area-based extraction, and such approaches are also dependent on the correct seed location and the measure of a similarity attribute. The presence of chimneys and antennas also influence area-based reconstruction by causing over-segmentation. Another key influence on area-based extraction is the density of the data, with several methods failing in modelling cue extraction due to sparse data. Both extraction methods are influenced by noise in the input data. The presence of noise for feature-based extraction will result in false positives being extracted, whilst over-segmentation of roof planes can result from area-based extraction of noisy data. Both of these errors require further refinement to make the data suitable for subsequent building reconstruction. Many of these errors have been rectified using semi-automatic approaches such as manual location of seed points, extraction of roof edges and refinement of the data.

Due to the aforementioned issues with the extraction of modelling cues, only the automatic reconstruction of simple roof structures is currently available. Simple buildings, including gable and hipped roofs, can be easily modelled by parametric modelling, with a limiting number of parameters required for successful reconstruction. The combination of simple models to produce more complex roof structures has also been proposed, but is again limited to the models in the predefined library and the extraction of the modelling cues. The combination of models, as well as data-driven approaches, have been developed which introduce strict geometrical and topological constraints to the reconstruction to compensate for the errors in the extraction stage. Many simple roofs can be defined within parameters, such as orthogonal corners, with ridgelines being parallel and along the line of symmetry of the roof boundary. The enforcement of these constraints is successful for reconstruction of rectangular buildings with simple roof structures, but thus limits the reconstruction of complex buildings.

With regards to validation, the short review has shown that the definable measure of 3D reconstruction is also still an open-ended research question. Whilst attempts have been made to quantify the reconstructed models, most of these are only based upon visual analysis. Many results of reconstruction are only presented for a small subset of results, which does not provide evidence of handling complexity and large datasets. Some measures have been suggested which are subjective to user defined quantities. Although current standards, such as CityGML, have been deemed unsuitable for the validation of reconstructed models, they can give a starting reference for how the reconstruction should be represented.

Chapter 3. Digital Image Capture and Data Processing

3.1 Introduction

The review in Chapter 2 shows that currently only semi-automatic approaches are available for reconstructing building models, with automatic building reconstruction methods only capable of producing simple building geometry. This is due to limitations in the planar fitting approaches using point clouds, which currently tend to be favoured (Rottensteiner *et al.*, 2014). Researchers have favoured lidar data because of its perceived higher accuracy and density of elevation data, when compared to that derived from imagery, and the difficulties of image interpretation (Habib, 2008; Haala and Kada, 2010). However, recent developments in image capture and processing have rendered the two datasets to now be comparable and have improved the spatial resolution of the imagery, which should enhance image interpretation. It was concluded in the ISPRS working group (WG) III/4 benchmark that the developments in digital photogrammetry should be investigated for suitability of 3D reconstruction (Rottensteiner *et al.*, 2014).

Digital photogrammetry has been present since the 1980s with digital image analysis undertaken first from scanned photographs (Forlani *et al.*, 2015). However, the maturity of the hardware for image capture and software for image processing has led to major developments in the digital images used today. The introduction of digital aerial cameras at the start of the 21st century saw the dawn of significant improvements in image capture, when compared to film-based cameras. By the end of 2008, an estimated 300 digital camera systems were being operated world-wide for aerial image capture (Honkavaara *et al.*, 2009). The development of software from computer vision for the processing of imagery has improved the spatial resolution and accuracy of the products. Developments in image matching procedures have resulted in pixel-to-pixel matching procedures that produce products with a high spatial resolution, which can rival those from lidar.

This chapter reviews the developments in the production of dense image matching datasets. Firstly, the development of digital aerial cameras is compared to film-based cameras: investigating the hardware required for image capture, the configurations of the camera systems and the processes used to pre-process the imagery. The results of building reconstruction from imagery in Kaartinen and Hyyppa (2006) were based on film imagery, whilst Rottensteiner *et al.* (2014) concluded the need to further investigate reconstruction from digital imagery. The developments in image capture, from film to digital, have allowed greater interpretation of imagery, making it more applicable for building reconstruction. This chapter will therefore review the change from film-based to digital aerial imagery, which has allowed improved interpretation and greater analysis of the imagery. Following this, a technique, currently popular in photogrammetry for the matching of pixels between stereo pairs of images, will be examined before the results of surface model generation are discussed. A comparison of dense image matching datasets to lidar surfaces will evaluate the accuracies of the surface and the nature of the two datasets. The chapter will conclude with an introduction and overview of the test sites used for this research project.

3.2 Film vs Digital Aerial Cameras

A photographic image is formed by a sensor capturing electromagnetic radiation, in the form of light, over a scene (Kraus, 2007). The image is captured by incoming light rays being refracted and focused onto the focal plane of a camera by the lens. The shutter and diaphragm regulate the amount of light during exposure by only allowing light through the lens when the shutter is open. For aerial photography, the amount of time that the shutter is open is dependent on the flying height and speed of the camera platform (Read and Graham, 2002).

Traditionally, a roll of film, which contained silver halide crystals, was used to capture an image, with incoming light rays directed onto the film using a single lens cone. When exposed to light, the bond of the silver halide crystals broke and produced a latent image (Mikhail *et al.*, 2001). The latent image is a grayscale image with a scale from black (low exposure to light) to white (high exposure to light). Red, Green, Blue (RGB) imagery was produced using three emulsion layers in the film, which were sensitive to different parts of the electromagnetic spectrum (Kraus, 2007). The film material used was highly sensitive and storage of the film, both before and after exposure, needed to be kept under specific conditions to control the humidity and the development of the film (Read and Graham, 2002).

Once the scene was captured with a film camera, an image was produced by placing the film in a number of chemical baths. This was to stop further development of the film and to remove any unchanged silver halide crystals (Read and Graham, 2002). To produce surface models and orthoimages, film imagery was then converted to a digital image using a photogrammetric scanner. The digital imagery was produced by placing the film image on a glass-plate flatbed scanner and using charge-coupled devices (CCD), constructed in a linear or area array, to convert each pixel into a digital signal (Kraus, 2007). For grayscale imagery, a digital scanner, which contains a number of CCDs, measures the amount of light not absorbed by the film by transmitting a constant stream of light to each pixel (Kraus, 2007). Images originally captured from film imagery can be scanned with a pixel size between 7 to 30 microns, which is dependent on the geometric resolution of the image (Kraus, 2007). However, problems have occurred with the film granularity and the loss of image information when scanning film imagery (Leberl *et al.*, 2010a; Passini *et al.*, 2012).

The problems mentioned above of the film based camera led to the development of digital metric cameras, and their introduction into workflows since the turn of the 21st century (Sandau, 2010). Digital cameras have now replaced film-based cameras in most applications (Toth and Jóźków, 2016). Digital cameras still require a lens and a shutter to control the exposure, but imagery is captured onto an electronic device, such as CCD sensors. Imagery is captured by recording the electronic charge, built up by photons of the incident light, that falls onto the image plane of the camera and converting this into an electronic signal using an analogue to digital converter (Kraus, 2007). This build up in charge is proportional to the number of photons that reach each semi-conductive detector, which is measured and converted to a digital signal using crystalline silicon and an analogue to digital converter (Sandau, 2010).

A digital image is constructed of many elements called pixels, which are usually square, and arranged in a matrix. A pixel is a discrete element with a digital number, which defines the intensity of the recorded light (Kraus, 2007). Each pixel represents part of a scene, which is determined by the spatial resolution of the camera. For example, if imagery is captured with a spatial resolution of 10 cm, then each pixel will represent a 10 x 10 cm part of the captured scene. When all pixels in an image are combined into a matrix, an image is formed of continuous digital numbers. The measure of all these pixel values is denoted by the radiometric resolution, which can be divided into level of quantization and spectral resolution (Wolf *et al.*, 2014).

The level of quantization defines the conversion of the analogue signal from the original electromagnetic energy into a discrete digital number (Wolf *et al.*, 2014). The range of digital numbers in an image is referred to as the number of bits, calculated as 2^n, where n is the number of bits per image (Kraus, 2007). The smallest possible radiometric resolution of an image is a 1-bit image, which is a binary image consisting of two colours: black and white. A grayscale image with 256 different digital numbers (0 – 255 grayscale range) is referred to as an 8-bit image and an example of this can be seen in Figure 3.1. The level of quantization influences the amount of detail that can be seen in an image. The higher the level of quantization, the more detail that can be seen in the image; whilst with a lower level of quantization, large areas appear homogenous, with tonal variation no longer detectable (Wolf *et al.*, 2014), as can be seen in Figure 3.2. Digital cameras are capable of capturing images with a radiometric resolution in the range of 12 – 16 bits per colour channel, thus reproducing up to 65,536 different tones per colour channel (Sandau, 2010). Film imagery is captured and generally scanned at 8 bits. This shows that imagery from digital cameras has a much higher dynamic range and better sensitivity to noise compared to film imagery (Sandau, 2010). This high signal-to-noise ratio and wide dynamic range means a greater amount of detail can be observed in what would previously have been homogenous areas in a film image (Paparoditis *et al.*, 2006). Although there are external factors such as scene illumination and atmospheric conditions that influence the radiometric resolution of the images, digital sensors have been shown to have a higher radiometric resolution compared to film (Sandau, 2010).

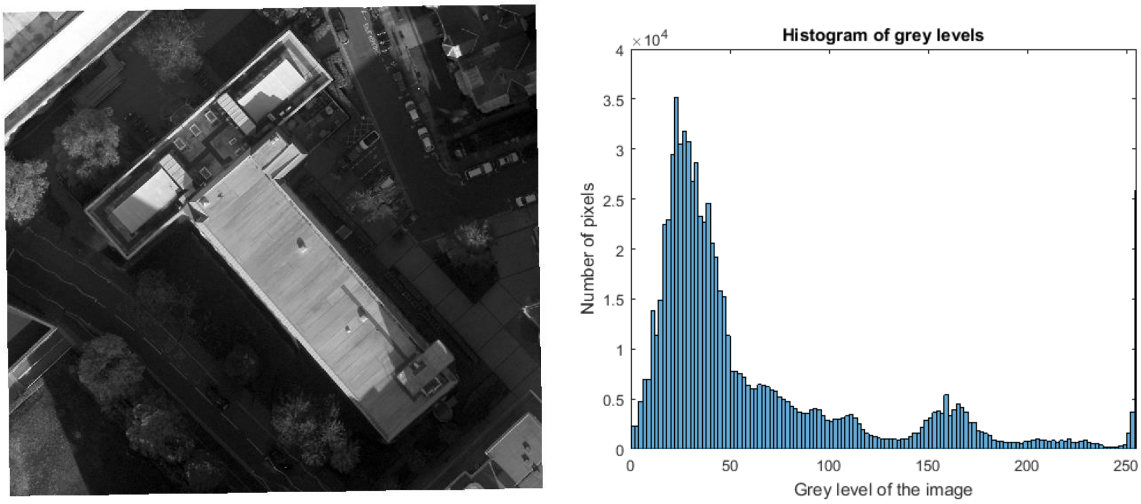


Figure 3.1 – The radiometric resolution of a grayscale image: (left) section of imagery from Newcastle upon Tyne dataset; and (right) the corresponding histogram of the pixel values.

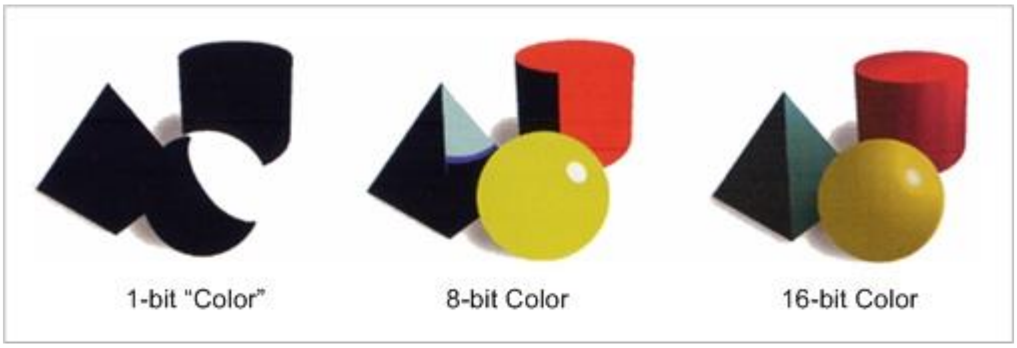


Figure 3.2 – Difference in level of quantization and amount of detail captured (Maxfield and Brown, 2006)

The spectral resolution of the data refers to the amount of electromagnetic energy that can be captured by the sensor. The relationship between exposure and illumination density of the image for film and digital sensors are expressed in Figure 3.3. The ideal relationship is a linear trend; as the exposure to the scene increases, the density of the image, calculated as a function of the logarithm of the exposure, also increases (Sandau, 2010). It can be seen in Figure 3.2 that this is the case for digital sensors. However, for film material, the relationship between exposure and density is a S-shape curve, as shown on the left of Figure 3.3. The illumination captured using film-based cameras is measured as the relative illumination as the density is influenced by the exposure settings and development of the film (Sandau, 2010). The minimum density of film images is never zero, even in unexposed films, due to the plastic base of the film and some density development when unexposed emulsion is processed (Lillesand *et al.*, 2008). The maximum density plateaus at a long exposure time, which means the range of the illumination densities is measured as the difference between the minimum and maximum densities. The greater illumination range captured with digital sensors, compared to film material, enhances scene interpretation, which is vital for image matching, described in Section 3.3.

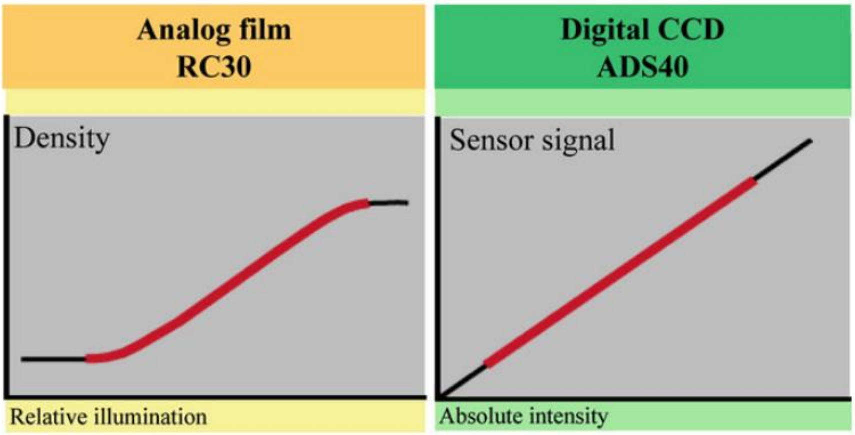


Figure 3.3 – Qualitative characteristics of a film camera and a digital camera (Sandau, 2010).

The spectrum of electromagnetic energy captured can be increased using a digital sensor compared to film material, where the amount of electromagnetic energy captured is dependent on the film used (Kraus, 2007). RGB and near infrared imagery (NIR) can be captured simultaneously using digital sensors, whereas the same area would have the be flown multiple times in order to capture all this data with a film camera, due to different film requirements and recordable wavelengths (Sandau, 2010). This increase in spectral resolution with digital sensors is possible using beam splitters to separate light rays, multiple camera cones to capture different parts of the electromagnetic spectrum or placing filters directly onto the digital sensors (Sandau, 2010). These sensors can be structured within digital aerial cameras using a matrix based area-array or a linear array.

Area-array sensors capture all image pixels simultaneously whereas line sensors use a single row of CCD sensors, which samples an image by moving the CCD sensors across the scene (Sandau, 2010). In a full-frame sensor digital camera, millions of CCD elements are arranged in a two-dimensional array. An image is acquired by exposing all CCD elements simultaneously to the scene. Filters can be applied to the optics of the camera or the surface of the CCD element to allow only certain light rays to pass through using beam splitters. Single lenses can also be combined together in a multiple cone configuration, which allows complete camera systems to be joined together in a multiple array design and increase the image footprint (Boland *et al.*, 2004). An example of a multi-camera headed configuration system is the UltraCam Eagle Mark 2, which can be seen in Figure 3.4a. This camera offers 10 cm GSD imagery to be captured from 4,000 m flying height with a pixel size of 4.6 µm (Vexcel Imaging, 2016). Multispectral imagery is stitched together from the various camera lenses to form large image footprints of 7,670 x 4,930 pixels (Vexcel Imaging, 2016). The multiple camera cones, seen in Figure 3.4a, allow panchromatic, RGB and NIR imagery to be capture simultaneously with four panchromatic lens and four multispectral lenses. The camera cones can also be interchanged to allow focal lengths of 80, 100, 120 or 210 mm to be used to increase the field of view (FoV) from 28.3° to 67° from nadir (Vexcel Imaging, 2016).

Multiple cone camera systems were developed because single lens digital frame cameras were not capable of maintaining the geometric resolution and ground coverage of film cameras (Habib, 2008). However, advances in technology have meant that single frame cameras have been able to overcome these issues (Toth and Jóźków, 2016). An example of a single frame large-format digital aerial camera is the Leica DMC III, seen in Figure 3.4b. This camera system uses a complementary metal-oxide semiconductor (CMOS) sensor instead of CCDs. Whilst CMOS sensors work in the same way as CCDs, CMOS sensors can be used in higher temperature ranges and prevent blooming and smearing effects of the images, but until recently, the image quality had been higher with a CCD sensor (Sandau, 2010; Leica Geosystems, 2016c). The Leica DMC III can capture imagery with 5 cm GSD, from a flying height of 1,179 m, and at a radiometric resolution of 14-bits (Leica Geosystems, 2016c). A focal length of 92 mm in panchromatic and 45 mm in multispectral offer a FoV of 57.2° and 61.7°, respectively (Leica Geosystems, 2016c).

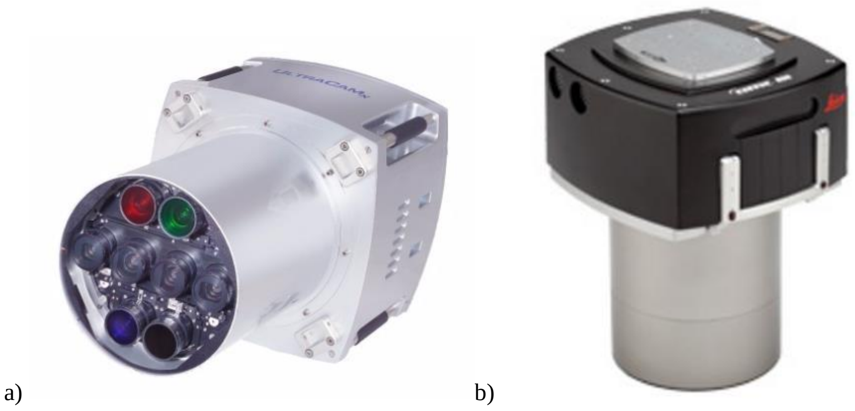


Figure 3.4 – Area-based digital cameras: a) the UltraCam Eagle Mark 2 camera system, developed by Vexcel Imaging (Vexcel Imaging, 2016); and b) the Leica DMC III (Leica Geosystems, 2016a).

The alternative to area-based array sensors are linear-based array sensors. These tend to use forward, nadir and backwards viewing angles to capture an object from three different viewing perspectives (Sandau, 2010). An example of a line-based system is the Leica ADS100, shown in Figure 3.5. This camera has three different across track FoV angles: at 65.2° for the forward view, 77.3° for the nadir view and 71.4° for the backward view, which produces footprints similar to those shown in Figure 3.6 (Leica Geosystems, 2016b). The different viewing angles causing the exterior orientation to change for each recorded line, which means at least three images are needed to determine the correct orientation via the use of connection points (Sandau, 2010). This is explained in Figure 3.6 which shows a plane flying along a flight path and the three image footprints from the three viewing angles of the line-based array camera. Figure 3.6 shows how at each exposure point (red footprints), the exterior orientation of images is different, but when images are capture of the same area at different exposure points (blue footprints) the exterior orientation is the same. In comparison, the exterior orientation is the same across the whole matrix for area arrays.

Figure 3.5 – The Leica ADS100 camera system, developed by Leica Geosystems (Leica Geosystems, 2016a).

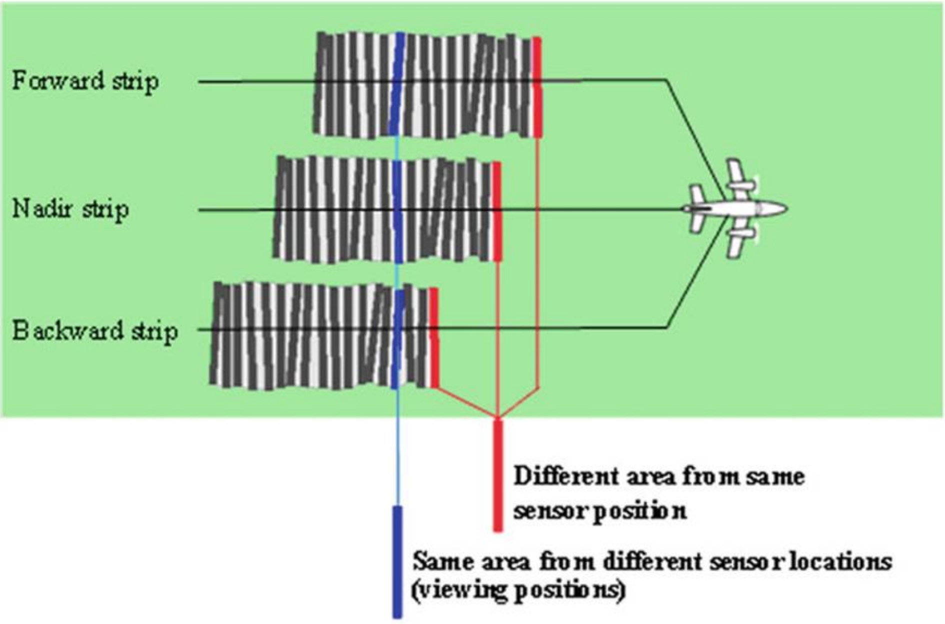

Forward strip
Nadir strip
Backward strip
Different area from same sensor position
Same area from different sensor locations (viewing positions)

Figure 3.6 – Exterior orientation change from each recorded position using linear array systems (Sandau, 2010).

For the storage of the captured digital imagery, an on-board storage device is also required. The UltraCam Eagle has an in-flight exchangeable solid state storage device capable of storing 3,900 images (Microsoft, 2014). The capture of digital imagery allows imagery to be viewed seconds after capture, allowing quality control to be conducted during flying. This quick production of images removes the process of developing the film and scanning individual images to form digital imagery (Sandau, 2010). The storage of imagery onto digital storage devices also enhances the archiving of images, which are not regulated by such strict conditions as mentioned for film imagery, and increases the volume of imagery that can be stored (Leberl *et al.*, 2012). The development of cloud computing has also improved the amount of data that can be stored digitally.

The developments of digital photogrammetry have resulted in a number of advantages compared to film-based image capture, both for the capture and image quality of a scene. Digital images have so far been shown to have a superior radiometric resolution, with a larger dynamic range and lower noise levels when compared to film imagery (Markelin *et al.*, 2008). There are also other advantages such as the reduction of mechanical parts in the camera system; CCD elements are fixed onto the focal plane whereas film is fed from the unexposed film reel, through the flattening plate and then to the take up reel. When using a film camera it is important that the film is held flat during exposure to ensure no distortion in the scene is captured (Mikhail *et al.*, 2001). The removal of these moving parts mean fiducial marks are no longer required because the interior orientation of the captured frame image is constant, when using an area-based digital camera (Boland *et al.*, 2004).

The increase in the number of camera cones and image footprint means that imagery can be captured at increased overlap, with only a small increase to the overall image acquisition cost. The improvements in the CCD technology have also meant that flying heights can be increased without reducing the spatial resolution of the imagery. The interchangeable lenses of the UltraCam Eagle means that 10 cm GSD imagery can be captured with flights flown at 1,739 m, covering a footprint of 6,956 x 6,956 pixels, or 4,565 m, covering a footprint of 18,260 x 14,790 pixels (Microsoft, 2014). The increase in image footprint and the removal of costs for film material has meant there is no extra cost for increasing fore/aft overlap along a flight track (Leberl *et al.*, 2012). By increasing the amount of overlap across a flight track, a slight increase in cost occurs due to extra flight time (Leberl *et al.*, 2010b). However, this cost is arguably negligible due to the improvement in captured image geometry. The removal of film and increased image footprints means overlaps of up to 90% along a flight-strip can routinely be captured (Haala, 2011). Overlap in adjacent flight lines has also typically been increased, up to 70%, which increases the number of flight lines required, thus increasing flying time and cost.

The increase in overlap of captured images also increases the number of stereo-pairs per ground pixel. With the nominal overlaps of 60% along a flight-strip and 20% across a flight-strip using a film camera, an individual image pixel can be seen in as many as six images. Whereas by using increased overlaps of 80% and 60% with a digital camera, an individual image pixel can be seen in up to 15 images (Haala, 2011). This is illustrated in Figure 3.7, which compares the image overlaps for film, on the left of the figure, and digital image capture, on the right of the figure, over two flight-strips. The increased overlap between images means more stereo-pairs can be matched per ground pixel, which helps reduce occlusions, especially important for building reconstruction, and improves the accuracies of the reproduced elevations (Leberl *et al.*, 2010a). An overlap of at least 50%, but typically 80% along and 60% across are required for the successful pixel-to-pixel matching between stereo pairs (Hirschmuller and Bucher, 2010).

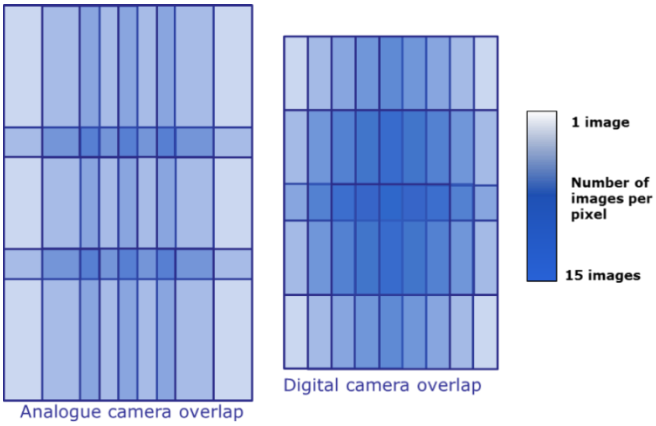


Figure 3.7 – Nominal capture overlap of imagery: (left) a film camera; and (right) a digital large format aerial camera.

Digital photogrammetric cameras were tested on geometry, radiometry, DSMs and manual stereo plotting in a study by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) (Cramer, 2010). A series of digital and analogue camera systems were flown capturing imagery of Vaihingen, Germany, at 20 cm and 8 cm GSD. The study found better vertical accuracy and point identification using digital imagery compared to scanned analogue imagery, due to improved image quality (Jacobsen *et al.*, 2010). The identification of image points was influenced by the available number of stereo models, with an increase in image overlap meaning each pixel could be seen in more stereo-pairs, thus improving identification (Spreckels *et al.*, 2010). When comparing the spatial resolution of the imagery, point and line identification was more accurate using digital imagery with higher spatial resolution, compared to scanned film images (Spreckels *et al.*, 2010). The accuracy achieved of the block adjust for digital imagery was sub-GSD for horizontal and vertical positioning, when compared to ground truth points (Jacobsen *et al.*, 2010). The comparison of surface models also took into account the developments in pixel-to-pixel matching procedures to produce dense image matching datasets.

3.3 Surface Reconstruction from Dense Image Matching

By identifying the same feature in different images, images can be matched and viewed in 3D via stereoscopy (Forlani *et al.*, 2015). As explained in Chapter 2, stereoscopy has been used for 3D building reconstruction from imagery to digitise roof points. There are currently many algorithms available to automatically match pixels between stereo-pairs of images. Historical approaches were developed to find corresponding features or patches between images in order to match stereo-pairs. With the improvements in the dynamic range and signal-to-noise ratio of digital imagery, as well as developments in stereo-matching algorithms, all pixels between stereo-pairs can be successfully matched to create dense image matching datasets (Haala, 2011).

Stereo-pair matching procedures are generally split into local and global methods, which use area-based or feature-based methods to find similar patches/objects in stereo-pairs. Feature-based approaches try to match the same object in two images using approaches such as scale invariant feature transform (SIFT), or the Harris operator (Forlani *et al.*, 2015). Area-based approaches work on the same presumption that the grayscale values at a corresponding point between pictures are similar, but match image patches instead of individual object features (Forlani *et al.*, 2015). Area-based methods try to overcome limitations of change in illumination and looking angle, which can cause feature-based methods to fail, whilst feature-based approaches are only suitable for sparse correspondences between images (Szeliski, 2010). The accuracy of matching using feature-based matching is also not as accurate as area-based matching, which is capable of sub-pixel accuracy when measured against ground-control points (GCPs) (Kraus, 2007; Gruen, 2012). However, area-based approaches fail in regions where there is little or no textural difference that can be detected, and can result in smoothing effects at height discontinuities (Haala, 2011; Gruen, 2012).

Local area-based methods use a search window by selecting the disparity with the lowest matching cost, whereas global algorithms tend to define a global energy function that includes a data term and a smoothness term (Rothermel and Haala, 2012). An example of a local method is normalised cross correlation, which has often been used in photogrammetry to match stereo-pairs of aerial images (Gruen, 2012). Subarrays statistically compare the digital numbers of pixels of the base and matched image for different image patches (Wolf *et al.*, 2014). The success of a match is determined by a correlation threshold and the size of the subarray for matching; small subarrays may produce a weak solution from low redundancy, whilst a large array may fail due to terrain variations within the image (Wolf *et al.*, 2014). Cross-correlation works well if there is sufficient signal with minimum geometric and radiometric distortion between the patches to be matched, which is often not the case (Gruen, 2012). Another area-based matching technique, which was popular in photogrammetry due to its flexibility and accuracy of matching is least-squares matching (Gruen, 2012). This approach was found to utilise the benefits of area-based and feature-based matching by using statistical estimation modelling, but the results were still not producing high-quality products at the same spatial resolution as the imagery (Gruen, 2012).

Global functions aim to match all pixels in an image instead of individual features or patches by minimizing an energy function, which determines the success of the match between a stereo pair of images (Szeliski, 2010). A smoothing term is generally applied to the energy function for measuring the differences in pixel disparity (Szeliski, 2010). The problem with calculating a global energy function is that it usually consumes a lot of memory and takes a long time to process (Hirschmüller, 2011).

To overcome the limitations of global and local methods, developments have occurred which aim to utilise both methods, with one example being Semi-Global Matching (SGM). Developed by Hirschmüller (2008), SGM has been widely adopted within the photogrammetric community for stereo-matching of aerial photographs. Stereo pairs of images are matched together by matching intensity values of all pixels along epipolar lines, which is shown in Figure 3.8.

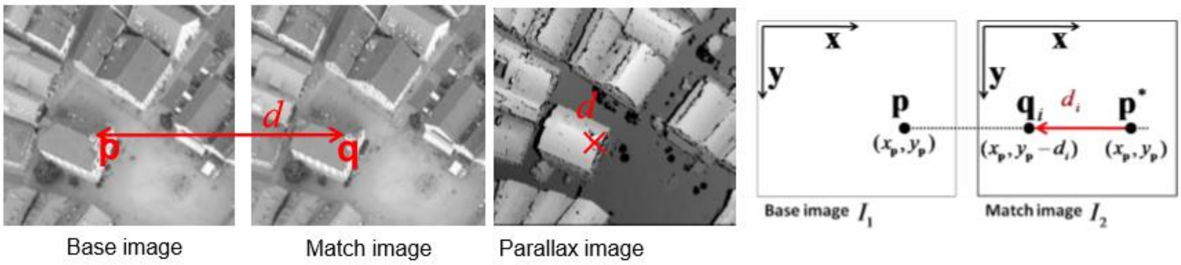


Figure 3.8 – Example of matching of pixels between base and match image (Haala, 2014).

The matching cost calculation is performed using Mutual Information, which calculates the entropy of each image and the joint entropy between the base and the matched image. Entropies are calculated from a probability distribution of intensity values, which requires one image to be warped to ensure corresponding pixels in both images are at the same location (Hirschmüller, 2008). The joint entropy is calculated as the sum of corresponding intensities at each search pixel (Hirschmüller, 2008). Gaussian smoothing is also applied to improve the match. Equation 3.1 shows the Mutual Information matching cost formula (C_{MI}).

$$C_{MI}(p,d) = -mi_{I_b f_d(I_b)}\left(I_{b_p} I_{m_q}\right) \tag{3.1}$$

$$mi_{I_b f_d(I_b)}\left(I_{b_p}, I_{m_q}\right) = h_{I_b}\left(I_{b_p}\right) + h_{I_m}\left(I_{m_q}\right) + h_{I_b I_m}\left(I_{b_p}, I_{m_q}\right) \tag{3.2}$$

Where:

d = disparity.

I_{b_p} = Intensity at the base image pixel p.

I_{m_q} = Intensity at the matched image pixel q.

h_{I_b} = Entropy of the base image.

h_{I_m} = Entropy of the matched image.

$h_{I_b I_m}$ = Entropy of the joint image (Hirschmüller, 2008).

A disparity image, *D*, is formed between the base and the match image that minimises the global energy function *E(D)* in Equation 3.3 (Hirschmüller, 2008). Two penalty terms are introduced for disparity discontinuities: (P_1) for all pixels q within the neighbourhood *Np* on the matched image where the disparity changes, and *(P_2)* for all large disparity changes (Hirschmüller, 2008). The first term in Equation 3.3 is the sum of all pixel matching costs for the disparities of *D* (Hirschmüller, 2008).

$$E(D) = \sum_P (C(p, D_p) + \sum_{q \in Np} P_1 T[|D_P - D_q| = 1] + \sum_{q \in Np} P_2 T[|D_p - D_q| > 1]) \tag{3.3}$$

The disparity image is formed by summing the costs (*S(p,d)*) along each epipolar line (*L_r(p,d)*), typically using eight paths but ideally 16 for increased redundancy, through the matched image (I_m) that end at the pixel p in the base image (I_b) (Hirschmüller, 2008).

$$L_r(p,d) = C(p,d) + \min \left\{ \begin{matrix} L_r(p-r,d) \\ L_r(p-r,d \pm 1) + P_1 \\ minL_r(p-r,i) + P_2 \end{matrix} \right\} - \min_k L_r(p-r,k) \tag{3.4}$$

$$S(p,d) = \sum\nolimits_r L_r(p,d) \tag{3.5}$$

Where:

$_r$ = traverse direction

C = pixel-wise matching cost

The disparity of the base image, D_b, is determine by selecting the disparity d that corresponds to the minimum matching cost for each pixel p (Hirschmüller, 2008). The disparity image of the matched image, D_m, is calculated using the same minimum cost function as D_b but along epipolar lines that correspond to the pixel q in the matched image (Hirschmüller, 2008). The two disparity images are compared against one another for consistency checks and to enforce uniqueness. The overall cost for each pixel p and disparity d is the summation $S(p,d)$ over all the cost paths in L_r. The disparity image is thus the minimum summed cost for each base image pixel (Hirschmüller, 2008).

The matching of pixels in stereo-pairs can be extended to match several images, for example all the overlapping images captured in a flight campaign, by matching pixels in a base image to all matched images individually (Hirschmüller, 2008). A disparity image is created for each matched image to the base image, and the individual disparity images are then fused together by calculating the weighted mean of disparity values, weighted by a factor t_k, which takes into account scale differences (Hirschmüller, 2008). Outlier matches are determined and discarded if the disparities values are not within a one pixel interval of the median of all disparity values at a pixel (Hirschmüller, 2008). The matching of all pixels within a stereo-pair of images results in a disparity image with the equivalent spatial resolution of the original image, thus produces a dense image matching dataset.

As previously stated there are a vast number of dense image matching algorithms currently available, some of which are listed and ranked by the Middlebury Stereo Test (Scharstein *et al.*, 2016). The first results of the Middlebury Stereo Test were released by Scharstein and Szeliski (2002) who created a taxonomy of stereo-matching algorithms ranked by four factors: matching cost computation, cost aggregation, disparity computation and disparity refinement. Participants of the study were given frame imagery, which consisted of indoor scenes with varying surfaces and textures, and true disparity images to compare their results against. The Middlebury Stereo Test is an ongoing study with four test sites available online for new methods to be tested against. Whilst the original study tested 20 stereo methods, there are now currently nearly 150 stereo-matching algorithms ranked online by the taxonomy (Scharstein *et al.*, 2016). Semi-Global Matching is currently ranked 87th by Middlebury Stereo and has a calculated average percent of bad pixels from the four test images of the Middlebury Stereo Evaluation of 7.5% (Scharstein *et al.*, 2016). However, this is based on four indoor scenes and does not consider runtime and scalability with respect to image sizes, where SGM has been proven to offer a good trade-off between run time and disparity image results (Hirschmuller and Bucher, 2010; Hirschmüller, 2011). SGM is only one approach to dense image matching which has been researched, developed and implemented into photogrammetric workflows. Studies, discussed in Section 3.4, have been conducted which have compared SGM to other approaches for the reconstruction of surface models from aerial imagery, and have found the results highly satisfactory, as well as comparing the reconstructed surface models to data acquired with an aerial lidar sensor, which in the past has had much higher elevation accuracy.

3.4 Surface Models from Dense Image Matching

Pixel-to-pixel matching algorithms, such as those described in Section 3.3, allow for the generation of products at the same spatial resolution as the captured imagery. A gridded raster DSM is created by fusing all matched disparity images together. The optical centre and the orientation of all cameras must be considered to fuse all disparity images onto an orthographic 2.5D projection (Hirschmüller, 2008). The use of multiple converging rays and the change of the projection can cause some cells to have multiple values due to the different viewing angles of the camera. To overcome this, the median disparity value is selected for any cell that has multiple values once all disparity images are fused together (Hirschmüller, 2008). The production of a gridded DSM at the same spatial resolution as the acquired imagery, and in an orthographic project, has led to the production of true-orthophotos. All radial distortion is removed for all image pixels, due to all pixels being perfectly registered to the DSM (Hirschmuller and Bucher, 2010). True-orthophotos are also not subject to double-mapped areas at height discontinuities, where the grey values of occluded areas are incorrectly mapped, which has previously been common in orthophoto production (Habib, 2008). Dense image matching has also introduced the production of 3D image-based point clouds, again at the same spatial resolution of the imagery (Haala, 2011). Point clouds have been previously reconstructed from imagery, by converting the centroid of each cell of a gridded surface or each node of a TIN to a XYZ point. However, this only produced a 2.5D point cloud, where each image pixel has one associated height value. By using multiple rays at different convergence angles, a full 3D point cloud, where an image pixel can have multiple height values, for example from roof overhang, can be reconstructed.

As part of the DPGF Digital Photogrammetric Camera study (Cramer, 2010), surface models created from pixel-to-pixel matching were compared with ground truth data collected from 63 Global Navigation Satellite System (GNSS) survey points, as well as airborne lidar. Raster grids of 0.2 m and 0.5 m were produced from the 8 cm and 20 cm images, respectively. The comparison showed that although lidar offered a higher height accuracy when compared against the GNSS survey points, the RMSE values from the image-matched DSMs were only slightly larger (Haala *et al.*, 2010). Point clouds were also compared against one another to evaluate the matching quality. The point clouds produced from the image-matching of the 8 cm GSD produced a much higher density with approximately 20pts/m², whereas the lidar point cloud had an average point density of 8.25pts/m² (Haala *et al.*, 2010). However, the standard deviation of the lidar point cloud was better than 2 cm, while the average of the image matching was 5.5 cm, after gross error removal (Haala *et al.*, 2010). The standard deviation of the image-matching datasets equated to an accuracy of below 1 pixel GSD for a single point measurement. When comparing the GSD of the imagery, the point density and the height accuracy were much greater using a GSD of 8 cm compared to 20 cm.

Further comparisons of surface models, created from dense image matching, were investigated in a joint EuroSDR and ISPRS benchmark (Haala, 2013). Dense image matching algorithms have been adopted by photogrammetrists and software developers. SURE, the photogrammetric suite developed at Stuttgart University, is based on SGM (Rothermal *et al.*, 2012). To test this and other matching approaches, participants were provided with aerial imagery and orientation parameters for two scenes (Haala, 2013). Independent reference data was not available, so a median surface was created from all produced surface models for comparison (Haala, 2013). The first scene was a subset of the DGPF study area over Vaihingen, Germany, with reduced image overlap of 63% forward and 62% side overlap, to determine how successful matching was with imagery at shorter image overlap. The results showed large height differences in areas cast in shadow, areas with repetitive patterns and waterbodies (Haala, 2013). Imagery with a high GSD meant small variations in the topography could also be depicted in the surface models, such as rows of vineyards, highlighting the potential of these matching algorithms to extract high resolution information. The second test site provided imagery captured at 80/80 forward and side overlap and a GSD of 10 cm with an Intergraph DMC II 230 camera over the city of Munich, Germany (Haala, 2013). Similarly to the results from the Vaihingen dataset, large height differences in the matching procedures could be seen at shadows cast across the image (Haala, 2013). The imagery contained a number of buildings, where profiles were taken from the reconstructed surface models. The sharp discontinuities at building edges were successfully reconstructed, as were finer roof details such as dormer windows, with differences between the reconstructed surfaces of approximately 1 pixel (Haala, 2013).

The study by DGPF and the EuroSDR benchmark highlight the potential of dense image matching datasets. The results have shown digital imagery to contain less noise, provide better geometric accuracy and produce highly accurate surface models. This has also been concluded in several other studies, which compared dense image matching datasets with lidar surfaces.

Gehrke *et al.* (2010) investigated the height differences in a 30 cm point spacing lidar surface model and a surface created from 5 cm spatial resolution imagery using SGM. The average height difference between the surfaces was 0.8 cm ± 5.4 cm (Gehrke *et al.*, 2010). Whilst the height difference was minimal, there was a large difference in the number of points used to construct the surface model. For a 375 m x 475 m area, there were ~1.5 million lidar points compared to ~63 million points produced from SGM (Gehrke *et al.*, 2010). Although smaller densities can allow quicker processing times, the higher density usually results in more detail being extracted. The higher density of SGM surfaces allowed the reconstruction of sharp image edges, for example at the edges of buildings, and the level of detail of structures reconstructed, including details of railway tracks, which were not extracted using a lidar system (Gehrke *et al.*, 2010).

Yastikli *et al.* (2014) created several DSMs with different software packages for the Zekeriyaköy district of Istanbul, Turkey, from 0.1 m GSD imagery captured with an UltraCam Xp camera. The DSMs were compared against lidar data with a point density of 16 pts/m^2. The test area consisted of different land use types: forestry, urban areas and large open fields. The surface models were compared based on the land use types for differences in height. The height RMSE, after elimination of height differences larger than 3 sigma, was smaller than 0.5 m for the open area of the DSM produced from SGM, and just over 1.5 m for the urban area for both the SGM DSM and that created from UltraMap v3, which uses a global matching formula for stereo matching (Yastikli *et al.*, 2014). The height RMSE of the forest areas was much higher with an approximate difference of 5 m. However, this was attributed to the multiple returns from the lidar system penetrating the tree canopy (Yastikli *et al.*, 2014). Profiles were also extracted for different land cover types, with the SGM DSM found to be coherent with the lidar surface, whilst the UltraMap v3 DSM produced heights slightly higher than the lidar surface (Yastikli *et al.*, 2014). This study implemented the lidar surface as ground truth in order to compare against dense image matching. However, the assumption that lidar is far superior is incorrect, due to the different error properties of both datasets (Gruen, 2012; Haala, 2013). The surface by pixel-matching is produced at a spatial resolution that is not achievable with standard lidar point clouds (Haala, 2013). A lidar campaign was previously determined to have to be flown 13 times longer compared to a camera, in order to achieve comparable elevation results (Leberl *et al.*, 2010b). A point density similar to lidar was reproduced with imagery at 25 cm GSD, but this spatial resolution was deemed insufficient to interpret urban objects (Leberl *et al.*, 2010b). In the EuroSDR study, pixel matching produced surface models at 25 pts/m^2 and

100 pts/m², whilst lidar was only available at 6.7 pts/m² and 4 pts/m² for the Vaihingen and Munich datasets, respectively (Haala, 2013).

The point cloud produced from a lidar sensor can vary in density, and is dependent upon factors such as flying speed, flying height, and the sensor properties (Petrie and Toth, 2008). As shown in Chapter 2, the point density can have varying influences on results. If the density is too sparse, some features may be incorrectly modelled due to missing details. Whilst more detail can usually be captured from high density datasets, there can be the result of increased noise capture, which will influence processing parameters and computation time. Point cloud densities have been quoted between 10 – 20 pts/m², but these are rarely uniform across a scene and can vary due to the reflective properties and structure of materials (Vosselman, 2012). For building reconstruction, Kim and Shan (2011) used point clouds with 1 pt/m² and 4 pts/m² density, whilst Oude Elberink and Vosselman (2009) used a point cloud with a density of 20 pts/m². These papers state the mean point density of the entire dataset, which can vary in different locations. In some locations the point density may be higher or lower than the quoted mean density, which can influence the processing parameters (Oude Elberink and Vosselman, 2011). However, by using the developed hardware of aerial cameras and the improved pixel-to-pixel matching algorithms to perform dense image matching, a much denser and uniform 3D point cloud can be reconstructed from imagery, when compared to lidar. The high density of image-based point clouds improves the distinction of height discontinuities when compared to lidar, and improves the level detail that can be captured (Leberl *et al.*, 2010b).

3.5 Study Areas

The methodology in this study was developed and tested on two dense image matching datasets for two different areas: Newcastle upon Tyne, United Kingdom, and Vaihingen, Germany. Both datasets were captured using different cameras, with different GSDs and products were created from the imagery using different photogrammetric software packages.

3.5.1 Newcastle upon Tyne, United Kingdom

The imagery over Newcastle upon Tyne was captured on 9th November 2011for a 25 km² area covering the city centre, industrial regions, along the river, to the south of the city and residential areas to the north. 95 images were captured with five strips, shown in Figure 3.9, using an UltraCam Xp large format digital aerial camera, manufactured by Vexcel Imaging. This was introduced to the commercial market in 2008 and uses a multi-headed system of four panchromatic lenses and four multispectral lenses (Gruber and Wiechert, 2009). Images captured from each lens are stitched together to create panchromatic images with 17,310 pixels across track and 11,310 pixels along track (Gruber and Wiechert, 2009). More information on the specification of the camera system can be seen in Table 3.1.

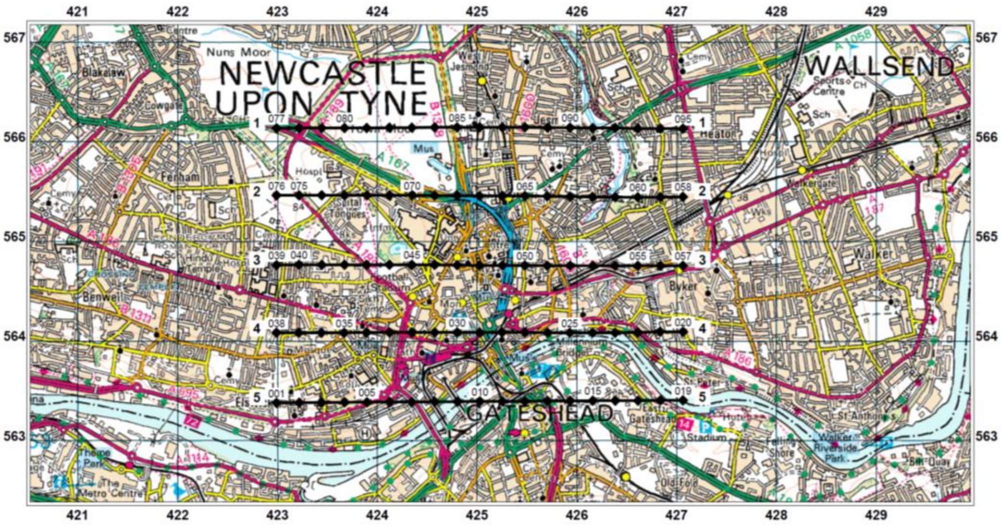


Figure 3.9 – An overview of the five flight paths for the Newcastle upon Tyne dataset. 95 images were captured in total, with each photo being represented by a black diamond.

Figure 3.9 shows each location of the 95 images captured during the Newcastle upon Tyne flight campaign. These camera positions were planned in ordered to capture stereo-pairs of images with 80% forward overlap and 60% side overlap. The stereo-pairs of images were processed in UltraMap v3 by photogrammetrists at OS, with the triangulation report shown in Appendix A. The UltraMap v3 software, developed by Microsoft for the processing of UltraCam imagery, includes a fully automated pipeline for the generation of a DSM, 3D image-based point cloud and true-orthophoto (Reitinger *et al.*, 2012). The UltraMap v3 software uses dense matching techniques to find corresponding pixels in a stereo-pair of images to create a gridded DSM. From this DSM, two types of orthophotos are created: a *DSMOrtho*, which used the DSM to compute a true-orthophoto; and a *DTMOrtho*, which generates a true-orthophoto from a temporary DTM, created with a gradient-based approach to filter buildings and other high-rising features (Reitinger *et al.*, 2012).

Building footprints were also supplied from the OS MM Topography Layer. The OS MM Topography layer is a vector dataset classifying various objects into points, lines and polygons for Great Britain (England, Scotland and Wales) to an accuracy of 1 m (Ordnance Survey, 2014b). Buildings are defined as roofed constructions, usually with walls and being permanent. This includes permanent roofed-constructions that exceed 8.0 m² in area (Ordnance Survey, 2014b).

3.5.2 Vaihingen, Germany

A subset of the previously discussed data for the DGPF study was used to test the transferability of the developed workflow. The same subset is used in the ISPRS Commission III/WG 4 benchmark, whose research focuses on automatic detection and 3D reconstruction (Rottensteiner *et al.*, 2014). Imagery was captured using an Intergraph / ZI DMC camera on 24th July and 6th August 2008 (Rottensteiner *et al.*, 2013). The specification of the DMC camera can be seen in Table 3.1, alongside the specification of the UltraCam Xp for comparison. The flight was flown at 900 m above the ground as three overlapping strips with one additional cross strips, shown on the right of Figure 3.10.

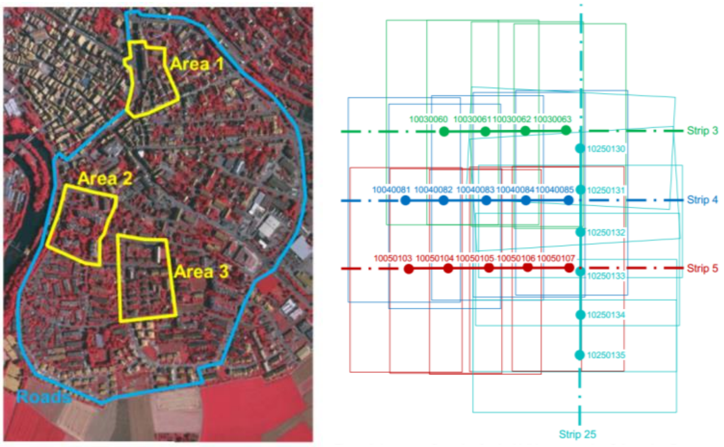


Figure 3.10 – Overview of Vaihingen dataset: (left) an example of the dataset showing the three areas of the ISPRS benchmark; and (right) the flight strips of the collected data (Rottensteiner *et al.*, 2013).

A DSM and true-orthophoto were created from the imagery at a ground resolution of 9 cm using the Trimble INPHO 5.3 software and MATCH-AT, MATCH-T DSM, SCOP++ and DTMaster modules (Rottensteiner *et al.*, 2013). No image-based point cloud was provided, so a 2.5D point cloud was produced by converting the centroid of each DSM cell into a XYZ point using the Feature Manipulation Engine (FME) software, produced by Safe Software Inc. The methodology for this will be described further in Section 4.2.1.

Vaihingen is located 25 km north-west of Stuttgart, Germany, and the imagery covers a 7.4 x 4.7 km² area. The area contains many residential and industrial buildings with various complexities in roof structures. Building outlines were provided for 17 areas, which were manually created for participants of the ISPRS benchmark to reference the results of object classification. From these areas, four test sites were selected based on roof geometry. Two of the test sites used for 3D reconstruction in this research, Area1 and Area3 shown on the left of Figure 3.10, formed part of the analysis for the ISPRS 3D reconstruction benchmark. Area1 is characterised as dense development consisting of historic buildings having rather complex shapes, whilst Area3 is a purely residential area with small detached houses (Rottensteiner *et al.*, 2013)

Specification/Camera System	UltraCam Xp	Intergraph / ZI DMC
Lens construction	Four panchromatic lens and four multispectral lens	Four panchromatic lens and four multispectral lens
Image Size	17,130 x 11,310 pixels (pan)	7,680 x 13,824 pixels
	5,770 x 3,770 pixels (colour)	
Focal length	100 mm (pan)	120 mm
	33 mm (colour)	
Field of view across track	55°	69.3°
Multispectral capability	Blue, green, red, near infrared	Blue, green, red, near infrared
Radiometric Resolution	12 bit	12 bit

Table 3.1 – Specifications of the UltraCam Xp camera (Microsoft, 2008) compared against the Intergraph / ZI DMC camera (Intergraph, 2008).

3.6 Summary

This chapter has discussed how aerial photography is captured, evolving from film-based cameras to digital sensors. The introduction of digital cameras has seen changes in the technology used, which has improved the spatial, spectral and radiometric resolution of the captured imagery. Developments in the algorithms used to process the imagery have improved the quality of the products produced from the captured imagery. Studies have been conducted to compare the surface models and point clouds created from dense image matching to lidar surfaces, which has shown good agreement between the datasets, but there is currently a trade-off between the point clouds produced from lidar and dense image matching. Lidar point clouds have a higher elevation accuracy but are irregularly spaced, whilst dense image matching point clouds offer uniform terrain sampling but at a lower elevation accuracy (Forlani *et al.*, 2015). However, as concluded in the ISPRS 3D benchmark, research is still needed to be conducted from these dense image matching datasets to determine their potential for 3D building reconstruction.

The proposed methodology of this research has therefore been developed and tested on two dense image matching datasets, discussed in the conclusion of this chapter. The two test sites show differences in building geometry and size, with imagery captured at different times of the year and with different camera systems and configurations.

Chapter 4. Reconstruction of 3D Building Geometry

4.1 Introduction

An automated workflow for the extraction of roof geometry and the 3D reconstruction of buildings from dense image matching datasets has been developed. The dense image matching datasets were created from stereo-pairs of digital imagery. An example of a stereo-pairs for the Newcastle upon Tyne datasets can be seen in Figure 4.1.

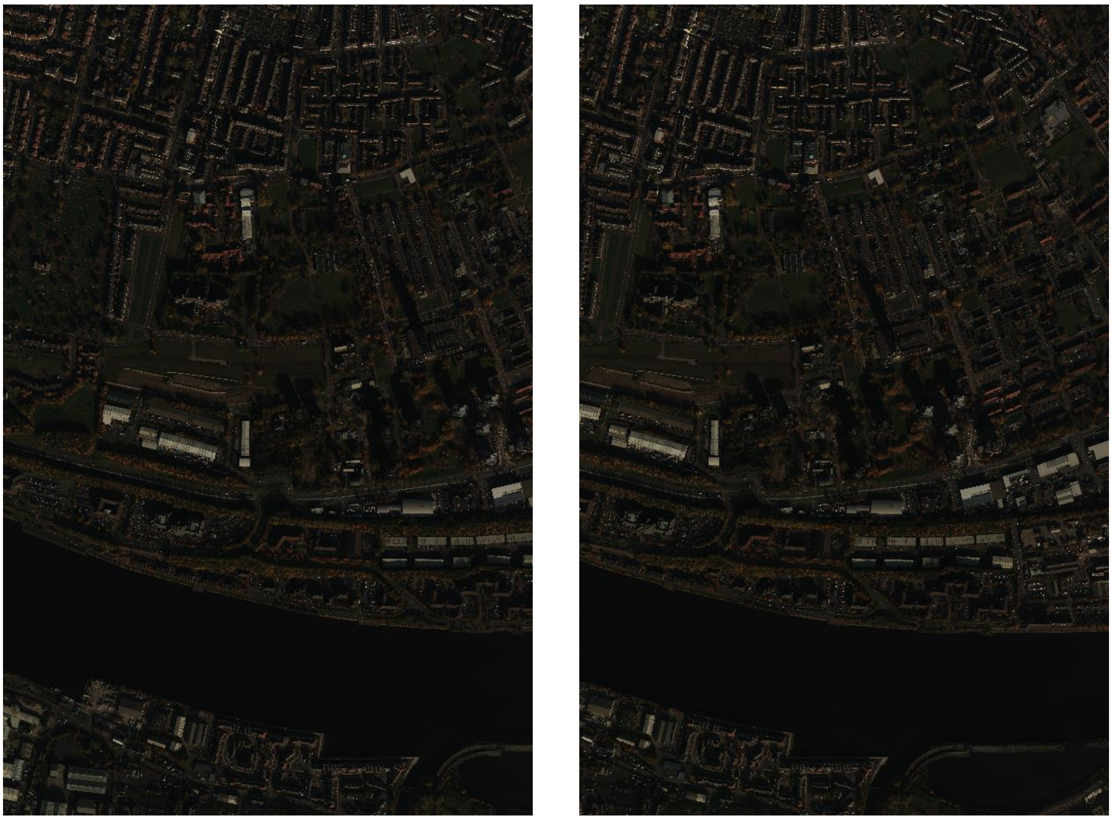


Figure 4.1 – Stereo images used to produce DSM, true-orthophoto and image-based point cloud.

These stereo-pairs of digital images were the initial input into the workflow, which is presented as a flow diagram in Figure 4.2, where rounded rectangles represent input/output, parallelograms represent intermediate datasets and rectangles represent processes. The data capture and processing to create the DSM, true-orthophoto and image-based point cloud were described in Section 3.5. The reconstruction of 3D models from the developed methodology can be defined by three main work packages (WP): (1) edge detection; (2) scan line segmentation; and (3) run graph vectorisation, which will be explored in more detail in this chapter.

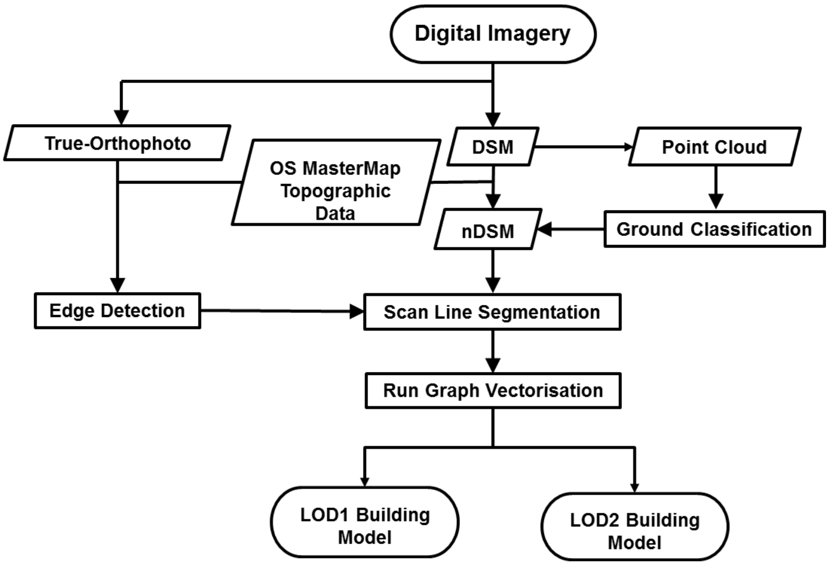


Figure 4.2 – The developed methodology for the automatic reconstruction of 3D building models at LOD1 and LOD2 specification.

The optimisation of the thresholds for WP1 and WP2, used for modelling cue extraction, will be presented in this chapter. For Canny edge detection, three thresholds must be defined: the maximum and minimum threshold to determine the strength of an edge, and the size of the Gaussian filter to apply image smoothing. Various thresholds were applied to the test sites in order to optimise these parameters. To quantify the edges extracted, reference data for the roof-lines was manually digitized from the stereo imagery. By comparing the results to the reference data, the number of correctly detected edges (true positives), incorrectly detected edges (false positives) and missed edges (false negatives) were determined (Lopez-Molina *et al.*, 2013). Results from the sensitivity testing of scan line segmentation, to remove any detected false positives, will also be presented. A least squares linear regression line was fitted through detected edges along each cross-section of the roof, using the corresponding height value from a nDSM. A false positive was determined and removed if the residual, measured as the distance from the edge pixel to the fitted line, was below a threshold. If the residual was above the threshold, then the edge pixel was determined as a breakpoint. In this research, a breakpoint is defined as the boundary of two or more planes along a cross-section, or the edge of the roof. Therefore, the size of the residual threshold was determined by sensitivity thresholding. Breakpoints were then connected together to form lines and classified into boundary lines, from the breakpoints along the edge of the roof, and roof-lines, from the breakpoints at the edge of two or more planes.

The chapter starts by discussing several pre-processing steps which had to be undertaken, including: (1) rectification of the image-based point cloud for ground classification, (2) buffering of image footprints and (3) the extraction of data within the buffered building footprints. These pre-processing steps were performed in Microstation using TerraSolid TerraScan and ESRI ArcMap 10.3. The pre-processing steps were undertaken as automated procedures: using a macro in Microstation and Model Builder in ArcMap to ensure no manual processing was needed.

After discussion of the pre-processing steps, details of the three main work packages for the reconstruction of building models, previously outlined, will be described. Building models were reconstructed at two levels of detail. Simple block models were reconstructed without roof geometry by classifying boundary lines and boundary corners. A roof model was reconstructed by classifying roof lines and roof corners, which were then connected to the boundary lines, at boundary corners, to reconstruct a building model with roof geometry. The building models were reconstructed to the accuracies defined by the LOD1 and LOD2 specification of CityGML.

The reconstruction of 3D models using the three main work packages was achieved automatically using a script within MATLAB R2015b, which will be discussed further in Section 4.3. The methodology was developed on the Newcastle upon Tyne dataset outlined in Section 3.5.1.

4.2 Data Pre-Processing

Three pre-processing steps were required before the extraction of roof geometry and 3D reconstruction could be undertaken using the developed methodology, as previously described. These were the rectification of the image-based point cloud for ground classification, buffering of image footprints and the extraction of data within the buffered building footprints

4.2.1 Rectification of the Image-Based Point Cloud and Ground Classification

As defined in Chapter 3, the introduction of dense image matching algorithms with high spatial resolution imagery, captured with high image overlap, has led to the production of image-based point clouds. Because of the high image overlap, multiple viewing angles can be reconstructed for the same image pixel, with a 3D point reconstructed for each pair of image rays. The creation of a raster gridded surface model is produced by the filtering of all points from each matched stereo pair to the median height value within each grid cell (Hirschmüller, 2008). The matching of multiple viewing angles for the same point means a 3D point cloud can be created with multiple points for each image pixel. Previously, only a 2.5D point cloud was possible from image matching due to the smaller image overlaps and matching of pixel patches and features (Vosselman *et al.*, 2004).

The image-based point cloud from the Newcastle upon Tyne dataset was produced with the UltraMap v3 software. A cross-section of a gable roof can be seen in Figure 4.3 from this image-based point cloud (red), where considerable noise can be seen when compared to a lidar point cloud (white). This was due to the image-based point cloud being unfiltered. For each image pixel, all matched rays have been used for the reconstruction of the 3D point cloud. As the imagery was captured at 10 cm GSD, the resulting point cloud density should have been 100 pts/m^2. However, because the point cloud was unfiltered, the resulting density was, on average, over 300 pts/m^2. Because of the amount of noise in the point cloud, a new 2.5D point cloud was produced from the DSM. This was achieved using FME. A workflow was formulated to convert the centroid of each DSM cell into an XYZ point. The GSD of the DSM was 10 cm, which was the same as the imagery, and resulted in a point cloud with a density of 100 pts/m^2.

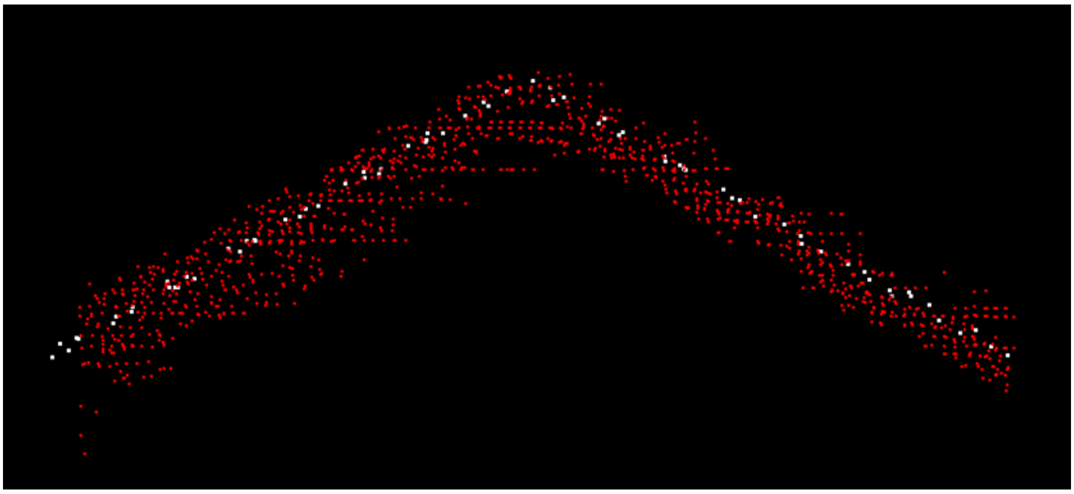


Figure 4.3 – A cross-section of a gable roof in the Newcastle upon Tyne image-based point cloud from the UltraMap v3 software (red) compared to a lidar point cloud (white).

The 2.5D point cloud was classified into "ground" and "building" classes to create a DTM, which was used to create a nDSM. The point cloud classification was undertaken in Microstation using TerraSolid TerraScan, where a macro was produced to automatically classify points. The macro used OS MM building footprints to classify any point that was within the outline of a building footprint as a "building". This was undertaken to aid the classification of ground points. The remaining unclassified points were then classified as "ground" using the ground classification procedure in TerraScan, which iteratively creates a TIN from the points. The procedure is dependent on several parameters being defined. A maximum building size parameter defines an area that has at least one point on the ground (TerraSolid Limited, 2015). The lowest point in the defined area is used as the first point of the TIN. A TIN is built from this initial ground point by triangulating other points based on iteration angle and distance parameters. The angle parameter defines the maximum angle between a point, the projection on the triangular plane and the closest triangle vertex (TerraSolid Limited, 2015). The distance ensures the triangulation does not include large vertical jumps, with suggested values between 0.5 m and 1.5 m (TerraSolid Limited, 2015). A terrain angle must also be defined, which determines the steepest allowed slope in the ground surface and a value of 88° - 90° is suggested for a scene with man-made object (TerraSolid Limited, 2015). The suggested settings from TerraSolid Limited (2015) were used for all parameters to classify the ground. The ground classification was densified by adding any point within a height of 2 m from the ground classification, in order to remove artefacts such as cars, hedgerows and noise in the dataset.

The ground classification generated from the point cloud was exported as a raster image to create a gridded DTM, at the same resolution as the imagery. A nDSM was computed in MATLAB by masking out the DTM pixels from the DSM.

4.2.2 Buffering of Building Footprints

Ancillary data, such as building footprints, has been shown in Chapter 2 to be useful for the reconstruction of 3D building models. Footprints can be used to produce the outline of the building and to limit the search area for roof geometry extraction. OS MM building footprints are captured from terrestrial surveys and aerial imagery to map the outline of a building footprint (Ordnance Survey, 2014b). These building footprints do not account for roof overhang. Therefore, the footprints were buffered to compensate for any differences in the roof outline and the building footprint. Each building footprint was buffered by 2 m, in ESRI ArcMap v10.3, to limit the search area for roof geometry extraction, whilst overcoming differences in the building footprint and roof overhang. The buffering distance was defined by the positional accuracy of LOD2 models to ensure corners were not extracted outside the reconstruction specification (Gröger and Plümer, 2012). The OS MM building footprints delineate each 2D postal address: a terrace building can be split into several polygons, as shown in Figure 4.4; whilst a multi-storey block of flats will only be constructed of one polygon. When buffering building footprints, it was decided to merge the building footprints to reconstruct the building as a whole, instead of reconstructing a 3D model within each footprint and merging the sections together.

Although some previous attempts, which are listed in Chapter 2, have used building footprints to reconstruct buildings, it was chosen not to base reconstruction on the building footprint so that the approach was not reliant upon these. The footprints were only used to limit the search area for roof modelling cue extraction, which could also have been achieved through image classification.

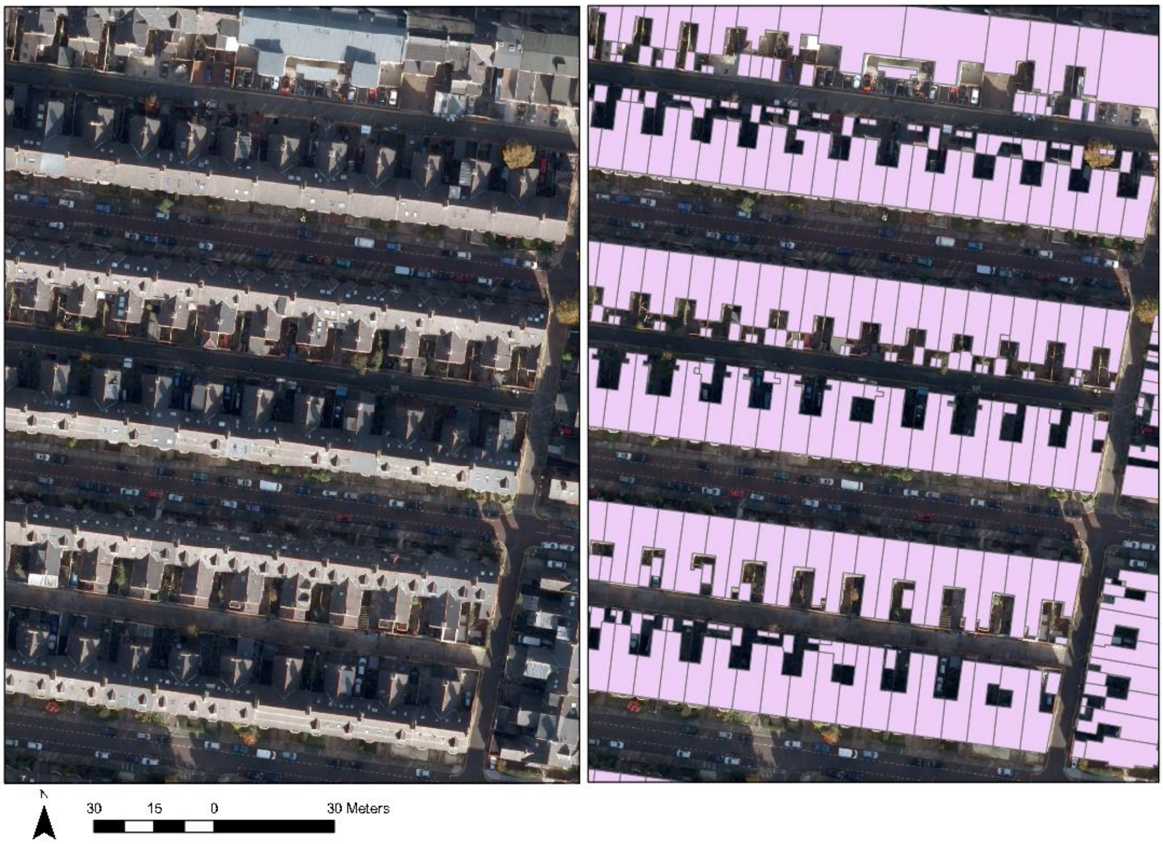


Figure 4.4 – The representation of buildings in OS MM: (left) extraction of the Newcastle upon Tyne dataset showing rows of terrace houses; and (right) the representation of the same terraced houses in OS MM, © Ordnance Survey.

4.2.3 Extracting Buildings from a Dataset

The buffered building footprints were used to limit the search area for the extraction of modelling cues, by mining areas from the true-orthophoto, DSM and DTM. The *Extract by Mask* tool in ArcMap was used to extract areas within the buffered OS MM building footprint (ESRI, 2016b). Test areas were exported as GeoTiff images with the same spatial and radiometric resolution as the original data. Examples of an extracted gable roofed building from the true-orthophoto, DSM and DTM are shown in Figure 4.5, which were used as input for the MATLAB script for 3D reconstruction. Each extracted building was given a unique name, such that the only input required for the script was the name of the folder in which extracted buildings were saved.

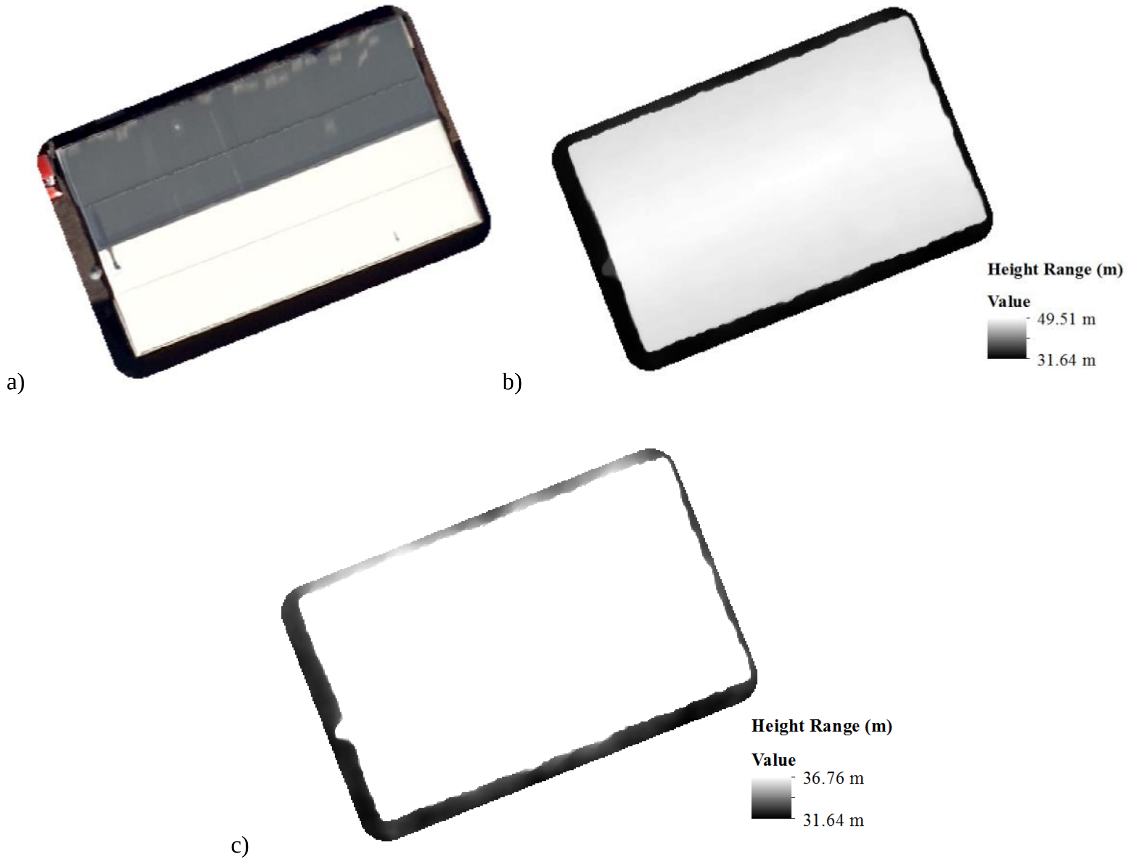


Figure 4.5 – Input to the MATLAB script for 3D reconstruction: a) true-orthophoto; b) DSM; c) DTM.

4.3 Development and Automation of Reconstruction

The previously discussed pre-processing steps were undertaken as automated workflows in two separate software packages. A MATLAB script was written to reduce the number of software packages used and to automate the remaining stages of reconstruction: outlined as edge detection, scan line segmentation, run graph vectorisation and connecting the roof model to the block model in Figure 4.2. This allowed these steps to be completed automatically using one script. An image can be defined as a matrix of pixel values, which is why MATLAB was chosen, due to the matrix and image processing capabilities. The image processing toolbox has built-in functionality for Canny edge detection, with parameter sensitivity testing undertaken. The results of this are presented in Section 4.4.1.

MATLAB is designed to work primarily on whole matrices and arrays (MATLAB, 2017). Scan line segmentation used rows and columns of an image matrix to determine roof-line geometry, whilst run-graph vectorisation uses sub-arrays of the whole matrix to firstly convert raster edges into vector line geometry, and secondly to test to connectivity of roof-lines. These two approaches were coded, expanding on the ideologies of Alharthy and Bethel (2004) Montero *et al.* (2009) to overcome the limitations found in previous research. The approach adopted in this research for run graph vectorisation uses some well-known ideals, the code for which was taken from the MATLAB exchange website.

To begin the reconstruction process, the script has been formulated to search an input folder directory for a true-orthophoto, DSM and DTM to produce a 3D building as a block model and with roof geometry. Each building to be reconstructed is required to have a unique name in the folder and saved as TIFF image files with the naming convention <name> for the orthophoto, <name>_DSM for the surface model and <name>_DTM for the terrain model.

4.4 Extraction of Roof Modelling Cues

To start the automatic reconstruction procedure, 2D roof geometry was extracted from the true-orthophoto using edge detection. As will be shown, the results of edge detection were prone to false positive extraction. Therefore, a procedure to refine the results was developed. Heights, from the nDSM, were associated to the extracted edge pixels and used to remove any extracted false positives.

4.4.1 2D Extraction of Roof Vertices

As described in Section 2.3.1.1, edge detection has been used by many researchers to extract the bounding lines of roof planes. An edge is referred to in this research as a raster pixel that has been detected at the edge of two features. One of the most popular approaches is the Canny edge detector, which is further outlined in Section 2.3.1.1. The Canny edge detector was implemented in this workflow to extract edges from the true-orthophoto of each test site. The imagery was converted to grayscale using the *RGB2GRAY* command in MATLAB (MathWorks, 2016). As discussed in Section 2.3.1.1, the edges extracted from imagery by the Canny edge detector were dependent on heuristic thresholding and Gaussian smoothing of the input imagery, to reduce the number of false positives detected. In order to optimise the results of edge detection, these thresholds must be determined before edges were extracted.

The function in MATLAB for Canny edge detection allows for the parameters to be calculated from the input image. To automatically compute the thresholds for the heuristic threshold, it is presumed that 70% of the pixels will not be an edge and a separate threshold ratio is set at 0.4 to determine the minimum threshold from the maximum threshold (MathWorks, 2006). The maximum threshold is firstly determined by calculating the histogram of the magnitude gradient image, by placing all values into 64 bins. The cumulative sum of values in the bins is computed to find the first bin with a cumulative value greater than the percentage of pixels not edges (0.7), multiplied by the number of rows in the image, multiplied by the number of columns in the image (MathWorks, 2006). This value is then divided by 64, which represents the number of histogram bins, to yield the maximum threshold value for Canny edge detection. The minimum threshold is determined by multiplying the calculated maximum threshold by the threshold ratio (0.4) (MathWorks, 2006). The size of the Gaussian filter is determined by a value of sigma, which unless defined in the input parameters for Canny edge detection, is set to √2, and thus is image independent (MathWorks, 2006).

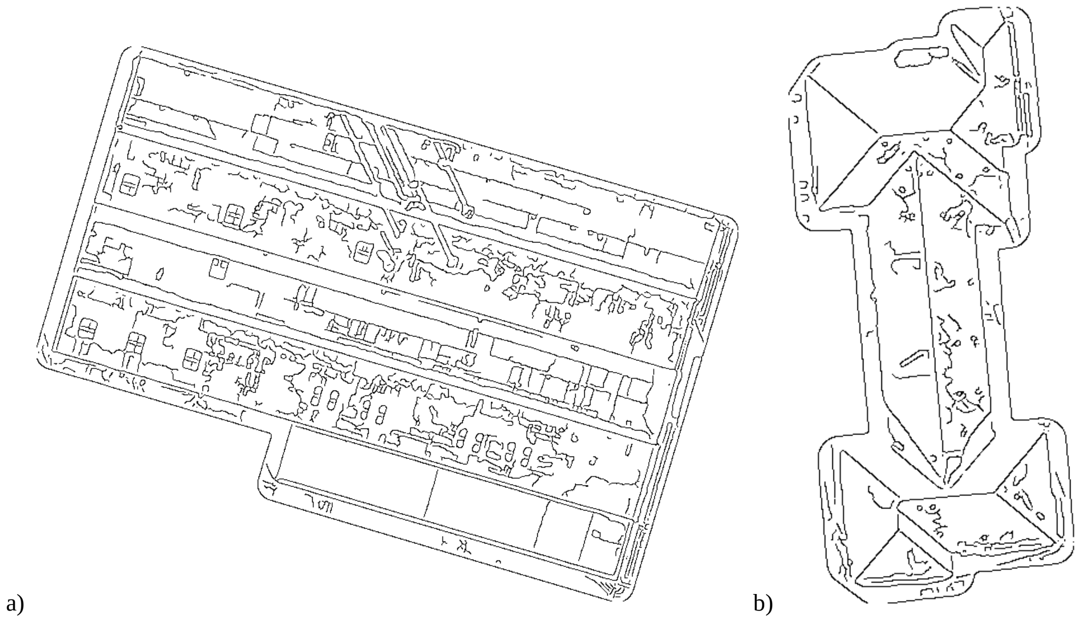


Figure 4.6 – Examples of Canny edge detection for two roof structures, using the calculated parameters from the input image by MATLAB.

These calculated thresholds for Canny edge detection were tested on 20 different buildings, with varying roof geometry. By using this fixed value of sigma and the heuristic thresholds calculated from the input image, some example results of edge detection for two buildings are shown in Figure 4.6. From the test sites used, the values calculated for the maximum threshold ranged from 0.015 to 0.031, and the minimum threshold values ranged from 0.006 to 0.012. The results in Figure 4.6 show that a vast number of false positives were extracted, particularly for the example in Figure 4.6a. These were extracted from shadow, roof texture and other unwanted artefacts on the roof, as well as extracting edges at the image extent and features on the ground. It can also be seen that some edges along the roof-lines were duplicated. This large amount of noise in the extracted modelling cues makes automatic 3D reconstruction almost impossible. The large number of false positives extracted was due to the thresholds calculated from the MATLAB script. The value of the maximum threshold, which defines the strength of an edge from the gradient magnitude image, was defined very low with a highest maximum threshold of 0.031, particularly when compared to the acceptable range of the Canny heuristic thresholds (0 – 1). Therefore, a larger maximum threshold was required to reduce the number of false positives extracted. However, as can be seen in Figure 4.6b, some edges were not extracted, particularly along the boundary-lines of the roof, using these low calculated thresholds. There is, therefore, a trade-off between the threshold values to reduce the number of false positives extracted, whilst extracting all roof structure lines.

To optimise the parameters, in order to reduce the number of false positives whilst extracting correct roof edges, sensitivity testing was undertaken. Firstly, the values for heuristic thresholding were defined. Tested values for the high and low threshold were defined based on those automatically calculated by MATLAB, as previously mentioned. The maximum threshold defines the strength of an edge, and is used in conjunction with the minimum threshold, so the sensitivity of the maximum threshold was tested first. Values of 0.05, 0.1, 0.15, 0.2, 0.25, 0.3 and 0.5 were tested. To test the maximum threshold, the value of sigma for the Gaussian filtering was kept constant at √2 and the minimum threshold was constantly kept at 0.

Examples of edge detection results using the defined maximum thresholds are shown in Table 4.1 for two buildings: one from the example in Figure 4.6a, and another for a hipped roof with dormer windows. As can be seen from the results, as the maximum threshold increased, the number of edges extracted decreased. This was due to the value of the pixels in the gradient magnitude image being below the maximum threshold, and thus not being extracted as an edge pixel. The roof-lines of the M-Shaped roof in the second column of Table 4.1 can be seen much easier using a maximum threshold of 0.1 or above, with the number of false positives clearly reduced. Nearly all the false positives were reduced with a threshold of 0.5. However, as previously described, as the maximum threshold increased, some of the true roof edges were not extracted. With a maximum threshold of 0.2, one of the roof boundary edges was not detected, and with a threshold of 0.5 several roof boundary edges were not extracted. Similar results were seen for the example in the third column of Table 4.1.

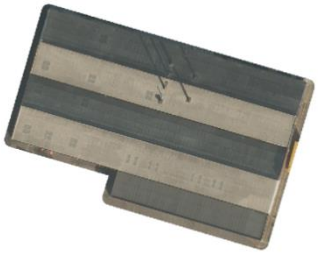
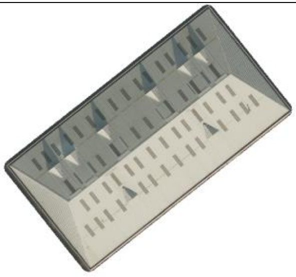
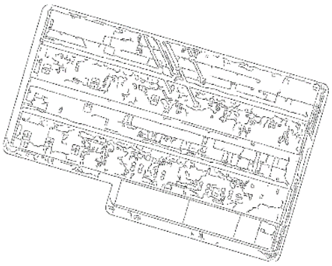
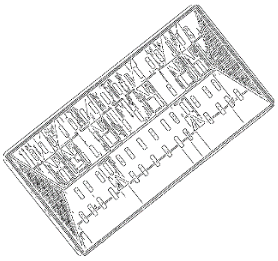
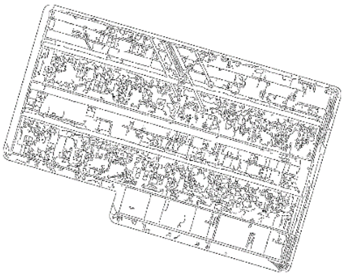
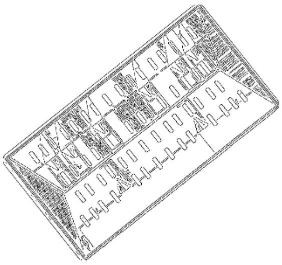
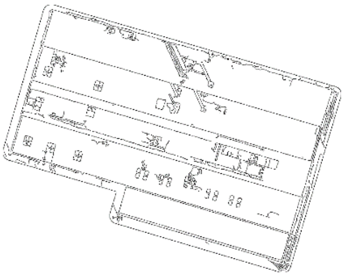
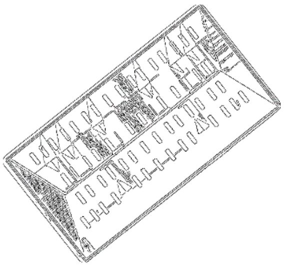
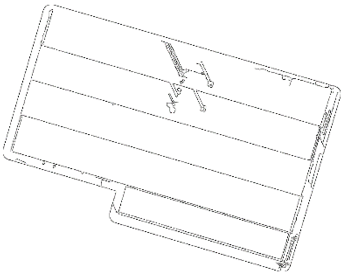
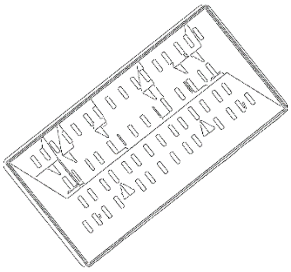
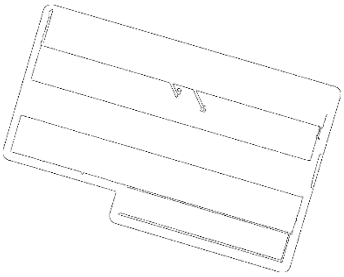
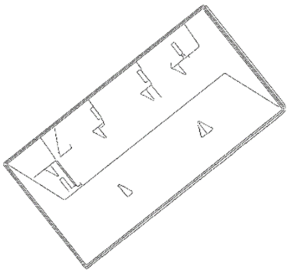
Maximum threshold values for Canny edge detection	Edge detection results for M-Shaped roof	Edge detection results for Hipped with dormer windows
True-ortho extraction		
MATLAB default		
[0, 0.05]		
[0, 0.1]		
[0, 0.2]		
[0, 0.5]		

Table 4.1 – Example results using various defined maximum thresholds for Canny edge detection of two buildings.

With the MATLAB calculated thresholds from the image, a vast number of false positives were extracted for the hipped roof example in the third column of Table 4.1, which made it almost impossible to determine the location of the roof-lines and dormer windows. As seen with the M-shaped roof, as the maximum threshold increased, the number of false positives extracted decreased. A slight reduction in the number of false positives extracted is seen by increasing the threshold to 0.05 from the MATLAB default threshold (0.015 in this example). The false positives on two roof faces were decreased, which are continually reduced as the threshold increased. However, one hipped line was not extracted using a threshold large than 0.1 and part of another hipped line was also not detected. As the threshold increased, sections of the dormer windows were not detected, but there was still a high presence of false positives extracted from roof texture in the form of small repetitive ovals. These were present in all edge detection results, apart from with a maximum threshold of 0.5. However, at this maximum threshold, two hipped lines were not extracted, and sections of the dormer windows were also not extracted. This again highlights the trade-off between extracting correct roof edges, whilst minimising the number of false positives extracted, and shows that post-processing was required. It was therefore decided to set the maximum threshold low, in order to ensure all roof edges were extracted, whilst reducing some of the false positives extracted. The maximum threshold was therefore set to 0.05.

Sensitivity testing of the minimum threshold was also conducted. However, because the value of the maximum threshold was so small, the results showed negligible differences in the edges extracted. Therefore, a value of 0 for the minimum threshold was defined.

Although the number of false positives was reduced by increasing the maximum threshold of the Canny edge detector, a number of false positives were still present. By adjusting the Gaussian filter, details on the roof surface should be smoothed to reduce the contrast between neighbouring pixels, and the strength of an edge. To test the value of sigma, for the size of the Gaussian filter, values based around the default MATLAB threshold were chosen. As √2 equals 1.41, values for sigma were chosen as 1, 2, 4, 6 and 8, with example results of the Gaussian image produced and the resulting edge detection shown in Table 4.2. Based on the results of previous tests, the maximum threshold was set to 0.05 and the minimum threshold was set to 0.

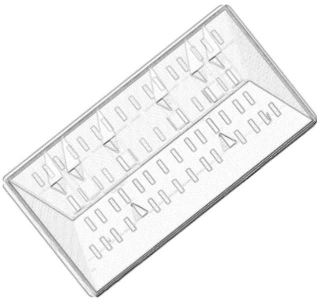
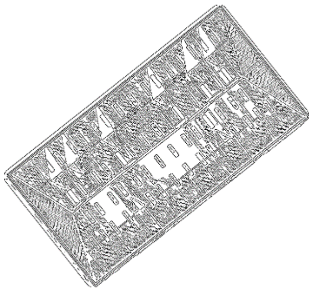
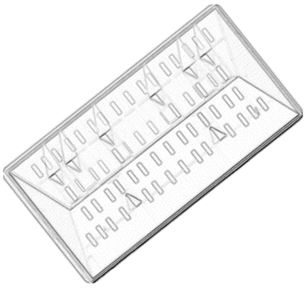
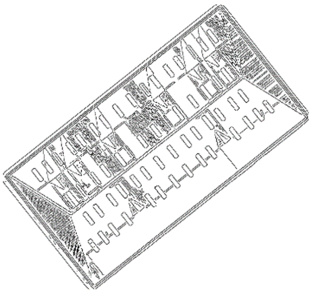
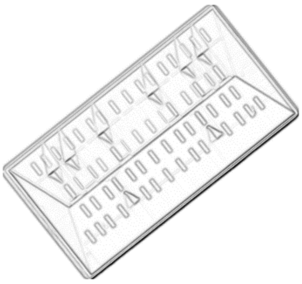
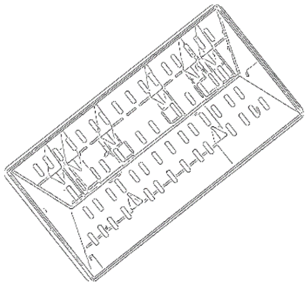
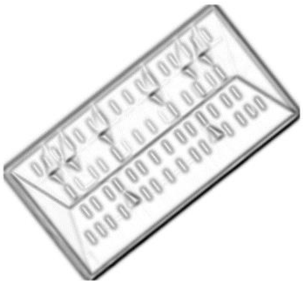
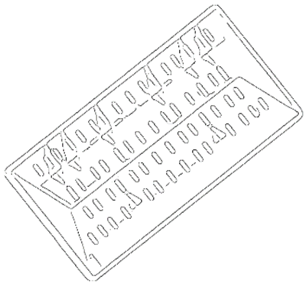
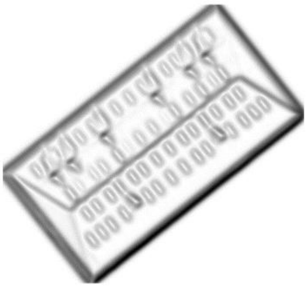
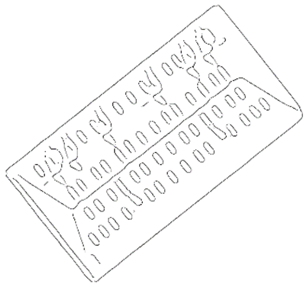
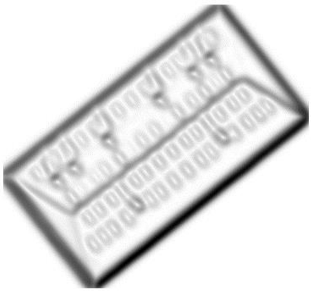
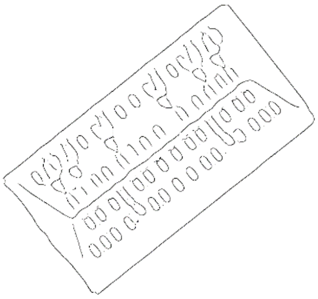
Sigma Value	Gaussian Image	Edge Detection Results
1		
Default (1.41)		
2		
4		
6		
8		

Table 4.2 – Example results of different Gaussian filter sizes used, and the resulting Canny edges detected.

The results of increasing the Gaussian filters showed that the number of extracted edges was reduced, but if the Gaussian filters became too large, separate edges were incorrectly merged together and the roof edges became curved, due to a reduction in image contrast. The optimum value of sigma was found to be 2, where nearly all of the cluttered edges were removed, whilst preserving the sharpness and straightness of the main roof edges. This is shown in Table 4.2 for the hipped roof with dormer windows. The number of edges on the triangular roof faces and between some of the dormer window has been reduced when comparing the results of the MATLAB default value of sigma to a sigma value of 2, from Table 4.2. Although the extraction of false edges was further reduced using a sigma value of 4, some edges of the dormer window were not extracted and the length of some hipped lines, the edges between the boundary corners and the central ridgeline, were reduced, as shown in Table 4.2. By using the value of 2 for sigma, a number of false positives were still present from roof texture, where the strength of the edge was too large to be filtered by smoothing or heuristic thresholding. This again highlights the fact that post-processing was required to refine false positive detection.

Example results of the edges detected from several buildings using the sigma value of 2 and the fixed heuristic thresholding values of 0 and 0.05, for the minimum and maximum thresholds, respectively, can be seen in Table 4.3, compared against the edges extracted when using the variable MATLAB calculated parameters. The number of false positives can be seen to have decreased, in most cases, whilst extracting the majority of the correct roof edges. The shed roof example in Table 4.3 is a particularly successful example of how refining the Canny edge detector parameters can enhance the extraction of roof edges. A shed roof is a flat roofed building which is sloped, with one side of the building being higher than its opposite side. Therefore, for the reconstruction of a shed roof, only the boundary of the roof is required. This is extracted with the MATLAB calculated parameters, but false positive edges running across the roof plane were also extracted from roof texture. By refining the parameters, these false positives were removed.

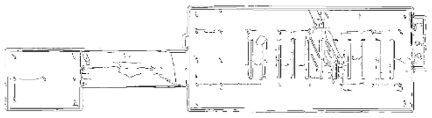
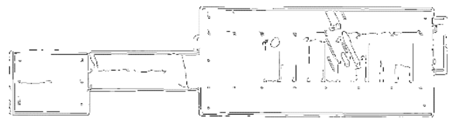
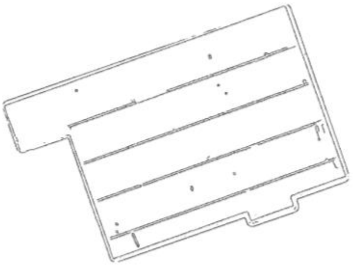
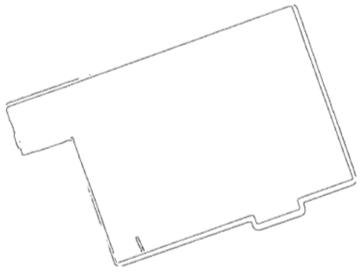
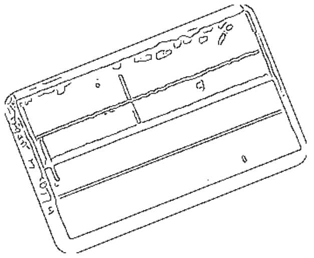
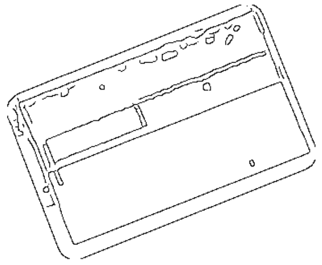
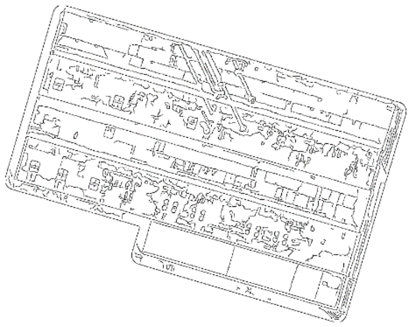
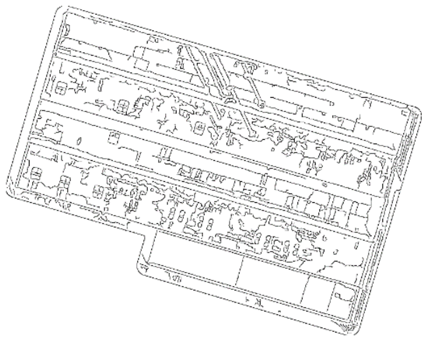
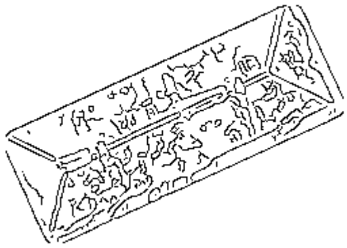
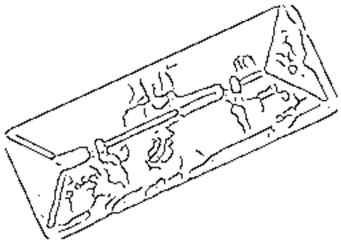
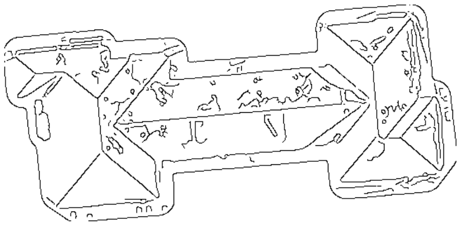
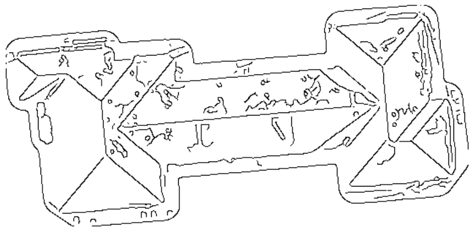
Test Sites	MATLAB defined parameters	Specified parameters
Flat		
Shed		
Gable		
M-Shaped		
Hipped		
Complex		

Table 4.3 – Examples of Canny edges detected using the MATLAB calculated and the specified thresholds.

By increasing the maximum threshold to the defined threshold, compared to the MATLAB calculated thresholds, did not cause any roof-lines to not be extracted. However, some roof edges were not extracted with either set of thresholds, as seen for the Complex test site in Table 4.3. This was due to low image contrast at the boundary of the roof, caused by shadow across the roof face and the ground, which produced a weak edge value in the Gaussian magnitude image. Although the value of the minimum threshold was defined smaller than that calculated by MATLAB, this edge was not connected to any edges above the maximum threshold, so could not be extracted. It was noted from the results that this seemed to mainly affect boundary edges, with all roof-lines extracted for all test sites. To overcome the issue of roof boundary edges not being extracted, the boundary of the nDSM was included with the edge detection for post-processing by scan line segmentation. By using the boundary of the nDSM, any edges detected from the ground surface and from the image extent were also removed. The effect of edges extracted from the ground and image extent can also be seen in Table 4.3.

The results of Canny edge detection using MATLAB calculated and the specified thresholds were measured against manually derived reference data to determine the number of roof edge pixels detected, as shown in Table 4.4. The results in Table 4.4 suggest that the detection of roof edges decreased by approximately 7% using the refined thresholds compared to those calculated from the image by MATLAB. However, as can be seen in Table 4.3, edges along all roof lines were extracted using the two thresholds with, in most cases, a reduction in the false positives. The statistical reduction in the detected edges may have been due to an offset between a detected edge and the reference pixel. The Canny edge detection was analysed by determining one-to-one correspondences between pixels in the edge detection image and a reference image. The reference image was created by converting manually delineated vector reference data into a raster image. If there was a pixel correspondence between the two images, an edge was determined to have been successfully extracted. An offset of one pixel between the reference and edge detection image resulted in the analysis defining the edge as undetected, or measuring a false positive as a correctly detected edge. Several of the roof-lines were extracted as duplicate edges using the MATLAB parameters compared to a single line using the specified parameters. This can be seen, for example, for the Complex roof in Table 4.3. Although an edge was present along the roof-lines, a slight offset between the reference image and the Canny edge detection, of just one pixel, would result in no correspondence between the images, and produce a smaller count of true positives. Also for the M-Shaped roof a lot of noise around the roof-lines was removed, which may have corresponded with the reference data. Although the statistics in Table 4.4 show a decrease in the number of true roof edges extracted, it can visually be seen in Table 4.3 that it was the number of false positives that was reduced.

Test Site	Total number of reference pixels	MATLAB parameters		Specified parameters	
		Number of reference pixels detected	Percent detected (%)	Number of reference pixels detected	Percent detected (%)
Flat	3526	2492	70.7	2100	59.6
Shed	2180	1744	80.0	1713	78.6
Gable	1534	1250	81.5	1240	80.8
M-Shaped	6204	4478	72.2	3755	60.5
Hipped	1296	1011	78.0	877	67.7
Complex	3934	3128	79.5	2914	74.1

Table 4.4 – Quantitative analysis of the number of Canny edges extracted with the MATLAB thresholds and the specified thresholds for Canny edge detection.

The nDSM was used as a ground mask to remove any detected edges that were not coincident with any nDSM pixels. To overcome any roof boundary edges missed, due for example to shadow or low contrast between the roof surface and the background, the pixels along the boundary of the nDSM were included in the edge detection image. An example of Canny edge detection, from the true-orthophoto of the gable roof in Figure 4.5a, can be seen in Figure 4.7. The results of edge detection have been overlaid onto the true-orthophoto to show the success of the edge detector. Edges detected on the ground and the image extent, which can be seen in the third row of Table 4.3 for the same gable roof, were removed using the nDSM. Some false positives were still extracted from roof texture using the refined parameters, which required post-processing.

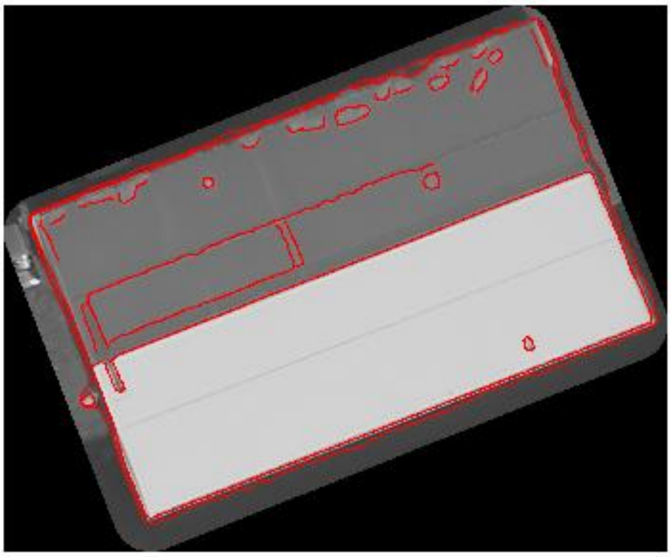


Figure 4.7 – Canny edge detection for gable roof in Figure 4.5a.

4.4.2 3D Extraction of Roof Vertices

Although parameters were set for the Gaussian smoothing of the true-orthophoto and heuristic thresholds to determine the strength of an edge, false positives were still present in the edge detection images from shadow, image noise and roof texture, as shown in Figure 4.7Figure 4.7

– Canny edge detection for gable roof in Figure 4.5a.. It was therefore necessary to filter any false positives. Previous attempts, discussed in Section 2.3.1.1, include removing edges shorter than a threshold length and combining datasets. However, the incorrect extracted edges are not always below a specific length, as can be seen in Figure 4.7.

To refine the Canny edge detection results, a methodology based on scan line segmentation was developed. As explained in Section 2.3.1.2, scan line segmentation was proposed by Jiang and Bunke (1994) for the extraction of planar boundaries using cross-sections of close range imagery. This was extended to classify roof planes from lidar point clouds and extract the boundaries of the roof planes, but the results were deemed unsuccessful due to the density of the point cloud (Alharthy and Bethel, 2004).

The principle of using cross-sections to classify the boundaries of planar regions was developed in this research to refine the Canny edge detection results. A cross-section was defined along each X and Y axes of the image coordinate system for each building, irrespective of the rotation of the building. A breakpoint was defined as a pixel at the boundary of two or more roof planes along a cross-section, or at the perimeter of the roof. A detected edge pixel along each cross-section was defined as the breakpoint of separate roof planes using least squares linear regression fit. If the residuals of a pixel to a fitted line were above a residual threshold, the pixel was classified as being on a separate roof plane to the fitted line and a breakpoint was defined. If the residual was below the residual threshold, a pixel was classified as being along a roof plane and not at the breakpoint of roof planes, so was removed as a false positive. This is explained in pseudo-code in Figure 4.8 and the process is illustrated in Figure 4.9, where Figure 4.9b linear regression of pixels on separate roof planes, where a breakpoint would be determine, whilst Figure 4.9c shows linear regression of pixels along the same roof plane.

```
For each cross-section
    Find all edge pixels (P) in a cross-section (i)
    Height of $P_i$ = nDSM value at $P_i$
    For each edge pixel ($P_i$) in a cross-section (i)
        $R_i$ = residuals of linear regression of $P_i$.
        If $R_i$ < threshold
            $P_i \neq$ Breakpoint
            Edge Image ($P_i$) = []
            Measure residuals at $P_{i+1}$
        Elseif $R_i$ > threshold
            $P_{i-1}$ == Breakpoint
            Fit new least squares linear regression from $P_{i-1}$ to $P_{i+1}$
    If any edge pixel removed
        Repeat linear regression for cross-section (i)
    Else
        Compute linear regression for the cross-section (i+1)
```

Figure 4.8 – Pseudocode formulated for scan line segmentation.

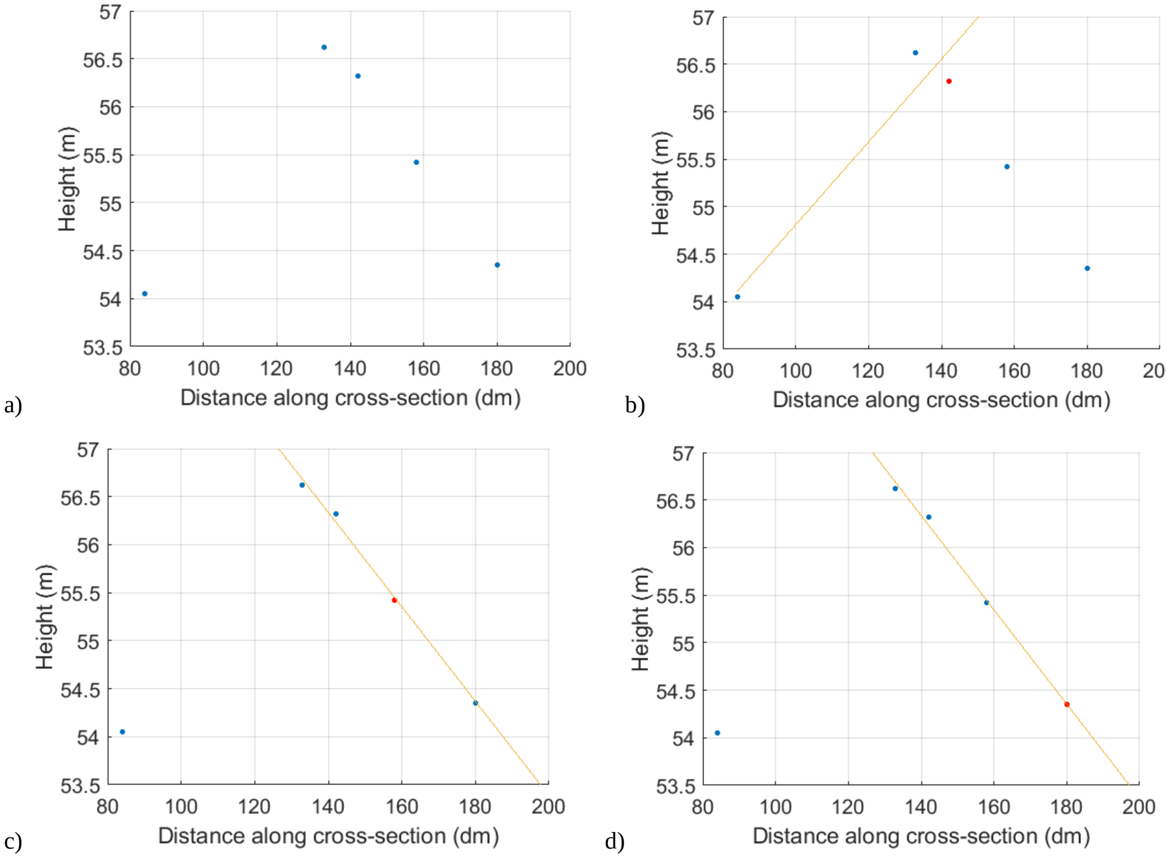


Figure 4.9 – The process of scan line segmentation: a) the cross-section of a gable roof showing the detected Canny edge pixels with associated height values; b) a fitted model along the face of a roof plane where the residuals were above the threshold (the red point represents the last point of the fitted line); c) the new fitted line after breakpoint determination from b; and d) the line fitted in c includes the next pixel along the cross section as the residuals were below the threshold.

To refine the Canny edges, the edge detection image was copied to produce two new images for scan line segmentation: one image for refining edges along the Y-axis cross-sections and another for refining edges along the X-axis cross-sections. A height value was assigned for each edge pixel from the corresponding pixel in the nDSM. This can be seen in Figure 4.9a for a gable roof, where the two roof planes can easily be seen, but with two erroneously detected edge pixels along one of the roof faces. For each cross-section along the X-Y axes of the roof, a least squares linear regression line was fitted to the detected edge pixels by iteratively adding edge pixels along the cross-section using the assigned height values. The first and last pixels found in the cross-section were determined as the roof boundary points.

The procedure started by fitting a least squares regression line to the first three detected pixels of each cross-section, as three points are needed for residual calculation. The residuals of the fit, defined as the distance from each observation to a fitted line, were measured. If any residual of an edge pixel to a fitted line was above the defined threshold, the second to last edge pixel of the fitted line was defined as the breakpoint between planar regions. If the last edge pixel of the cross-section was chosen, then the breakpoint would be in the wrong location. This is illustrated in Figure 4.9b. The red point is the last pixel of the least squares fitted line, which produced residuals above the threshold. The previous detected edge, the blue point at the top of the gable roof, was then determined as the breakpoint and was the first point to start a new least squares regression. The next detected edge pixel along the cross-section was then included with the previous two edge pixels to produce a new fitted line and the residuals were measured, as shown in Figure 4.9c.

If the residual of an edge pixel was below a defined threshold, then the pixel was assumed to be a false positive and removed from the edge detection image. This is shown in Figure 4.9c, where again the red point indicates the last pixel of the fitted line from the determined breakpoint in Figure 4.9b. The residual at this edge pixel was below the threshold, so this edge was determined to be a false positive and removed from the cross-section image. The next detected edge along the cross-section was then included in the least squares linear regression and the residuals measured, as shown in Figure 4.9d. Pixels along the cross-section were incrementally added to the regression line until the residuals were above the threshold, thus starting a new regression line, or the end of the cross-section was reached, which is the case in Figure 4.9d.

Once the residuals of all pixels in the cross-section to a least squares fitted line had been calculated, the process was either repeated for the same cross-section if pixels had been removed, or the edge pixels along the next cross-section were refined using the same procedure if no further refinement of the current cross-section was undertaken. If pixels had been removed from a cross-section, the residuals were re-measured for all refined edge pixels, along the cross-section, to determine if any further false positives were present. The refinement of edge pixels along each cross-section was deemed complete if no more pixels could be removed and the residual of all pixels was above the threshold. Once all false positive edge pixels had been removed from a cross-section, the process was started again along the adjacent cross-section to start removing false positives using least squares linear regression. All cross-sections along the Y-axis were used to remove false positives and produce the breakpoints in the direction of the Y-axis. A separate image, of the original detected edge pixels, was used to refine all cross-sections in the X-axis. The refined Y-axis edge image and the X-axis edge image were amalgamated together to produce a refined edge detection image.

To determine the residual threshold to accept and reject detected edges as breakpoints between roof planes, sensitivity testing was undertaken. The tested values for the acceptable residual measure of points from the linear regression line were defined based on the height accuracy of object points. The aerial triangulation report for the Newcastle upon Tyne dataset (Appendix

A) shows the RMSE height accuracy of object points to be 59 mm, with the poorest height accuracy to be 124 mm. Therefore, values were tested at 25 mm increments from 50 mm up to 150 mm. Research was also conducted to determine which weight function to apply to linear regression, in order to reduce the influence of outliers. Weight functions such as Andrews, Cauchy and Fair were tested against each other, as well as least squares regression with no weight function. Slight variations in the results were seen, mainly on edge pixels that should have been removed, with a difference of only a few pixels for the true roof edges. It was also deemed undesirable to use a weight function, because this reduced the residuals of the outliers, making it more unlikely to determine correct roof-lines, as well as requiring further parameter refinement of the tuning constants for the weights and taking longer to process compared to least squares regression with no weights.

Buildings with various roof shapes and inclination angles were chosen to test the refinement of edge detection, as can be seen in Table 4.5, Table 4.6 and Figure 4.10. The second row of the results in Table 4.5 shows the Canny edge detection for a gabled roof and a cross-section along the roof. Whilst the Canny edge detector has shown successful extraction of the central ridgeline and the boundary of the roof, there were also a number of false positives that were extracted from roof texture and shadow, especially on one of the roof planes. These false positives can clearly be seen to be along a roof face in the extracted cross-section in the second row of Table 4.5. By using linear regression, these false positives were removed to varying degrees of success. The results of the edges retained using different residual thresholds and the results of the same cross-section after filtering are also presented in Table 4.5. By analysing the cross-sections, it can be seen that false positives were removed using all residual thresholds. However, one false positive, close to the ridgeline, was still present in the cross-section using a threshold of 50 mm. When analysing the edge images, the effect of increasing the residual threshold can be seen to remove nearly all false positives on the roof face. Pixels along the falsely extracted rectangle, near the central ridgeline, were still present regardless of the threshold, but fewer pixels were present with a higher threshold. The edges of the falsely extracted rectangles became a fragmented edge, consisting of a singular pixel or a short length of connected pixels. Only slight differences can be seen in the results from the second column of Table 4.5 using a residual of 100 mm or bigger. With this threshold, nearly all false positives were removed at the roof boundary and only two sides of the falsely extracted rectangle were seen, which consisted of short fragmented edges.

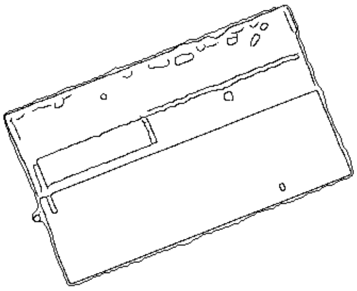
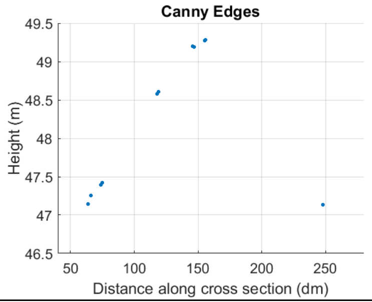
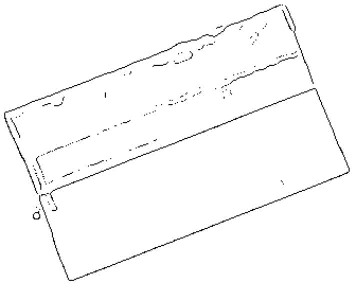
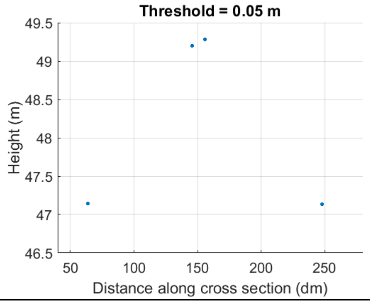
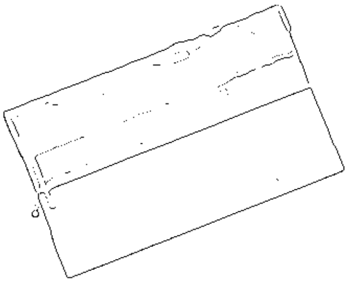
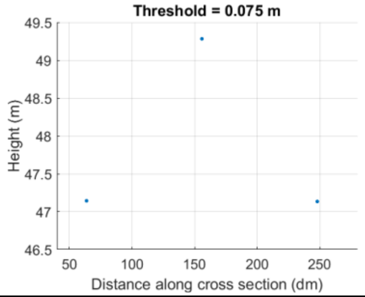
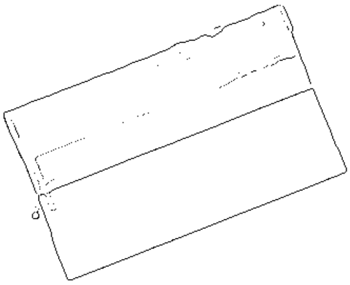
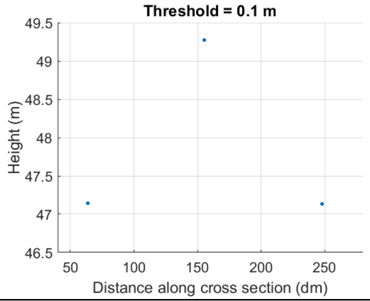
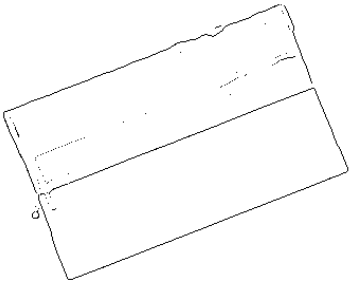
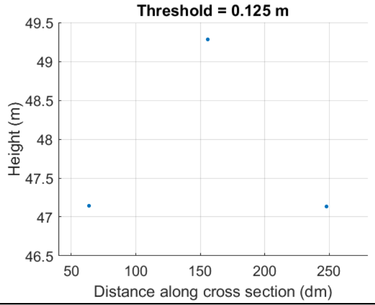
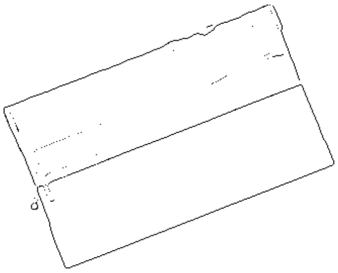
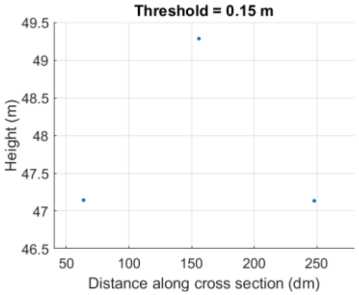
		Roof Edges	Cross-section of the roof
Edge Detection			
Residual Threshold (mm)	50		
	75		
	100		
	125		
	150		

Table 4.5 – Examples of scan line segmentation refinement with an extracted cross section of a gable roof.

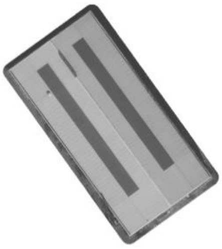
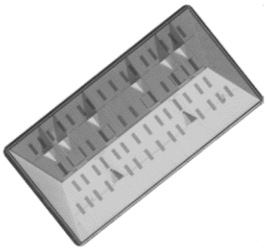
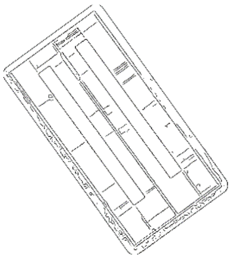
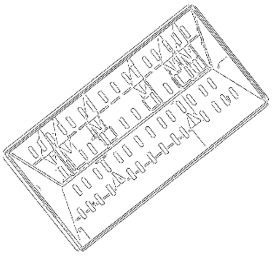
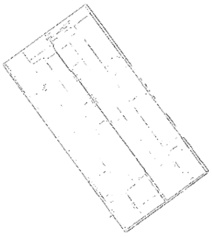
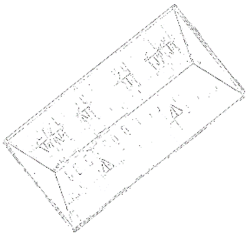
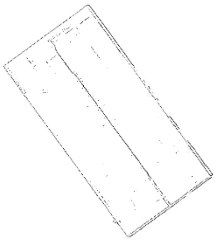
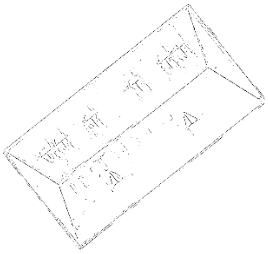
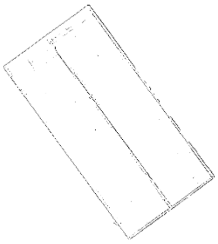
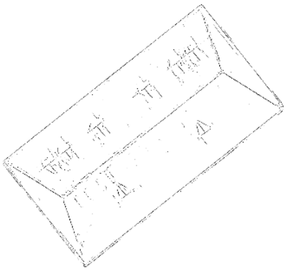
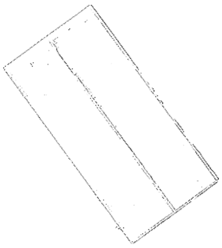
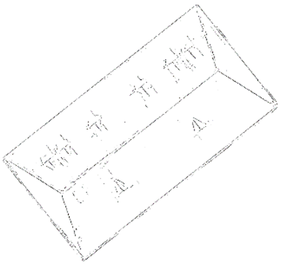
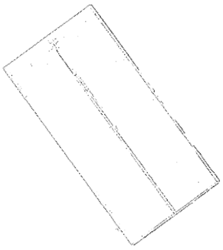
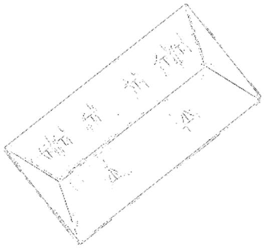
		Gable Roof	Hipped Roof
Ortho Extraction			
Canny Edge Detection			
Threshold for linear regression (m)	0.05		
	0.075		
	0.1		
	0.125		
	0.15		

Table 4.6 – Results of refining Canny edge detection using scan line segmentation and different residual measures.

Table 4.6 shows the results of scan line segmentation for two more roof structures: a different gabled roof and a hipped roof with dormer windows. Both of these roofs have distinguished roof texture that was extracted by the Canny edge detector, due to the significant change in the grayscale value at the pixel boundary, as well as extracting duplicated central ridgelines. The results of edge refinement in Table 4.6 show how false positives can be removed using the developed scan line segmentation procedure. For the gable roof in Table 4.6, fragments across the roof planes were still present with a small threshold, but were removed as the threshold increased. With a threshold of 50 mm, large sections of the false positives were still present in the edge image, as large sections of the roof texture were not removed. However, the smaller residual threshold appeared to remove false positives better close to the central ridgeline, a resulting duplicated edge was still present with the maximum tested threshold of 150 mm. The results using the 75 mm threshold show the central ridgeline as almost one single connected line, with only a few short segments near the boundaries of the roof. Similar results can be seen with the hipped roof. By using a larger residual threshold more false positives were removed. Duplicated edges of the central ridgeline were also removed. However, as the residual threshold increased, edges of the dormer windows were removed. Sections of the repetitive ovals, extracted from the roof texture were still present with thresholds of 100 mm and smaller, as well as retaining false positives around the true edges of the dormer windows. This noise was reduced with a larger threshold, but if the threshold was too large, then roof edges were incorrectly removed.

From all the results, the optimum threshold chosen was 125 mm, which showed a good trade-off between removing false positives and retaining the correct roof edges. This threshold performed successful refinement for all roof types, regardless of inclination, number of roof planes and complexity. The residual threshold almost corresponds to the poorest height residual of the object points, from the aerial triangulation file in Appendix A.

The example in Figure 4.10 shows the scan line segmentation results from the extracted edge detection of Figure 4.7. By measuring cross-sections only along the X-axis, as shown in Figure 4.10a, it can be seen that most false positives were removed, as well as most of the edges extracted along the ridgeline. Some false positives were still present, which may be due to image matching errors in the production of the DSM. Some edge pixels were also correctly extracted along the ridgeline, which were from cross-sections that ran across the two planes, due to the orientation of the building. In contrast, the edges extracted from cross-sections along the Y-axis, shown in Figure 4.10b, show that only edges at the boundary of the roof and along the ridgeline were extracted. This was due to cross-sections traversing across the roof planes, and edges not fitting to one single linear regression across both planes.

The results in Figure 4.10b successfully removed more false positives, whilst maintaining correctly extracted roof-lines, compared to the results of Figure 4.10a. Due to the orientation of the building, some of the cross-sections in the X direction did not run solely along a single roof plane and ran across the two roof planes. The residuals of the fitted line at some false positives were above the threshold for some X-axis cross-sections, which resulted in poorer refinement. By combining the results from Figure 4.10a and Figure 4.10b, the final results of scan line segmentation were produced, shown in Figure 4.10c. The results of Figure 4.10c, compared to Figure 4.7, show how nearly all false positives were removed, with only a few remaining; mainly consisting of short fragmented edges, which were removed in the next stage of reconstruction; vectorisation of the edges and connectivity of roof-line segments.

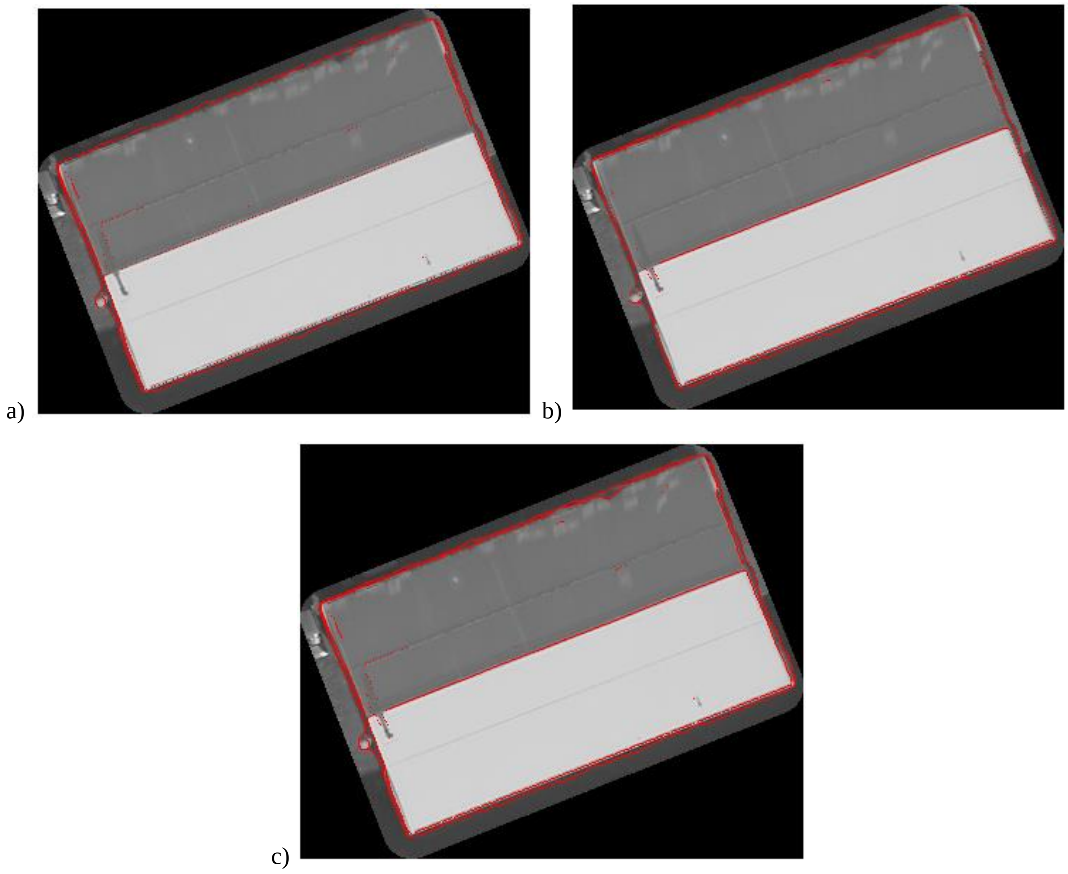


Figure 4.10 – Results from scan line segmentation from edges extracted in Figure 4.7: a) breakpoints (red pixels) extracted from cross-sections along the X-axis; b) breakpoints extraction from cross-sections along the Y-axis; c) the final result of scan line segmentation by amalgamating results from a) and b).

4.5 3D Building Reconstruction

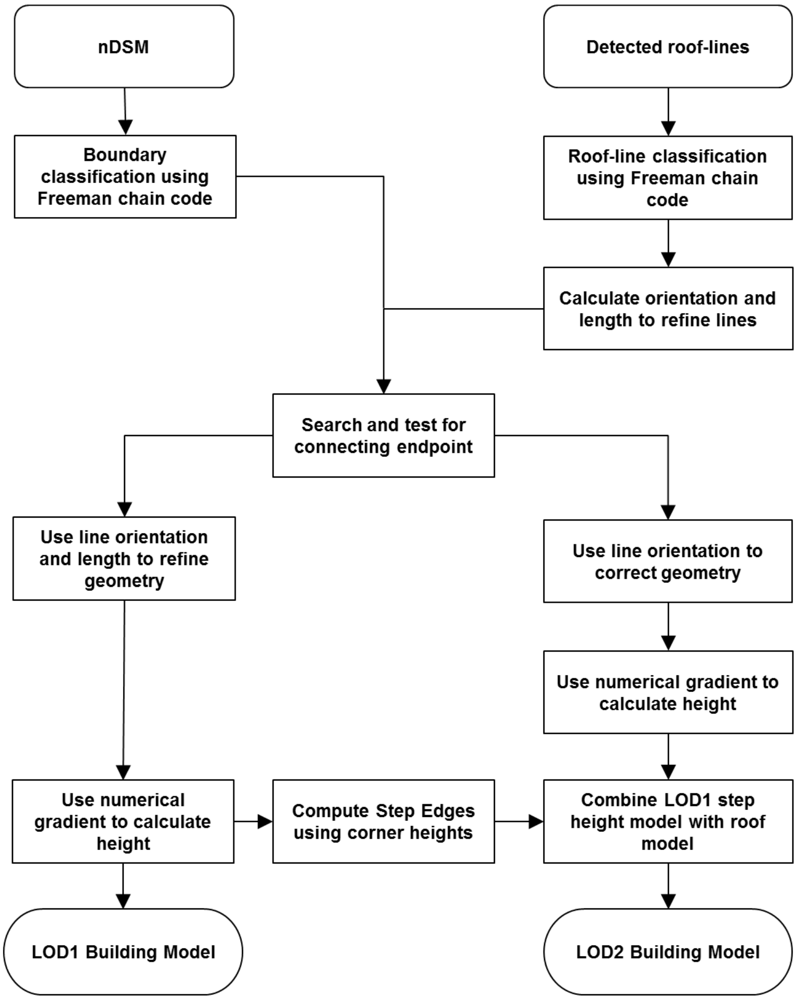

nDSM
Detected roof-lines
Boundary classification using Freeman chain code
Roof-line classification using Freeman chain code
Calculate orientation and length to refine lines
Search and test for connecting endpoint
Use line orientation and length to refine geometry
Use line orientation to correct geometry
Use numerical gradient to calculate height
Use numerical gradient to calculate height
Compute Step Edges using corner heights
Combine LOD1 step height model with roof model
LOD1 Building Model
LOD2 Building Model

Figure 4.11 – Workflow for 3D reconstruction of LOD1 and LOD2 models.

The developed workflow in Figure 4.11 shows the 3D reconstruction of LOD1 and LOD2 building models from the nDSM boundary, the extracted and refined edges, which is summarised as run graph vectorisation in Figure 4.2. The raster edges of the nDSM boundary and the extracted edges were vectorised separately using a developed run graph vectorisation approach to produce boundary-lines and roof-lines, respectively. The endpoints of unconnected roof-lines were then tested for connectivity between other roof-line endpoints and boundary corners. The extracted lines were refined separately for the roof boundary and the roof-lines using length and orientation thresholds. The roof boundary edges were then used to reconstruct LOD1 models. The predefined connection between roof-lines and boundary corners was used to combine the LOD1 model with the reconstructed roof model and create LOD2 building models. The parameters at each stage, which will be outlined in the succeeding sections, have been determined using threshold sensitivity.

4.5.1 Run Graph Vectorisation Theory

Run graph vectorisation was proposed by Boatto *et al.* (1992) for the extraction of corner points from edge imagery using line tracing. Boatto *et al.* (1992) defined horizontal and vertical lines, also referred to as runs, as a maximum sequence of pixels in either a horizontal or vertical direction. The runs were stored with a direction and coordinates for the start and the end of the run. Each run therefore formed a straight line, where a change in the run direction was the endpoint of a line and a corner point. This has been developed by researchers including Montero *et al.* (2009) to extract corners from a line tracing procedure, not limited to vertical and horizontal edges. Once edges had been detected from an image, Montero *et al.* (2009) used an 11 x 11 pixel window, split into nine directional classes (0 – 8), to determine the dominant direction of a five pixel fragment from the centre of the window to the end pixel position of the fragment, as shown in Figure 4.12

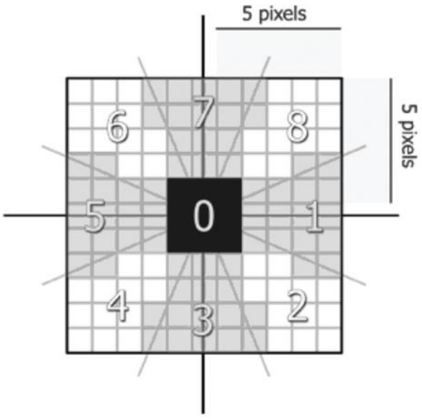


Figure 4.12 – 11x11 window used by Montero *et al.* (2009) to classify the direction of pixel fragments.

Classified fragments were then merged together to form line segments based on four criteria:

- Segments were constructed from at least three fragments with the same directional classification;
- Two segments of the same directional classification separated by less than three fragments formed one single segment;
- A directional classification of 0 formed a neutral position and was treated as the same directional class as the previous fragment;
- If the starting fragment and final fragment had the same classification, then they joined the same segment of length equal to the sum of the fragments (Montero *et al.*, 2009).

This approach to straight line detection was proven to be more successful than using the Hough transform (Montero *et al.*, 2009), which has been used by many researchers, as shown in Chapter 2. However, the results were dependent on the length of the fragments to classify direction, and the length of the fragments to merge into segments. In this research, the approach has been modified to remove the dependence of the fragment and segment length.

4.5.2 Developments to Run Graph Vectorisation

The concepts proposed by Montero *et al*. (2009) have been developed in this study to overcome the limitations of fragment length. Instead of splitting an edge into six pixel long fragments to classify an edge, each pixel along the edge was classified with a connectivity value. The connectivity of the pixels was measured with the Freeman chain code. The Freeman chain code uses a 3 x 3 cell window, as seen in Figure 4.13, to determine where the neighbouring pixel lies compared to the centre of the window (Freeman, 1961). The cell that the neighbouring pixel falls within was assigned the corresponding value from the window. The MATLAB code for Freeman chain code was based on the function written by Mannini (2010) from the MATLAB file exchange website.

3	2	1
4	■	8
5	6	7

Figure 4.13 - Freeman chain code (Freeman, 1961).

The connectivity of edge pixels was measured and a directional value was assigned to each edge pixel using the Freeman chain code. To define lines from the classified pixels, the connectivity ruleset defined by Montero *et al.* (2009) was refined in several steps.

Line segments were formed by clustering directional classification along an edge, similar to Montero *et al.* (2009), but the segment length threshold was modified. For each edge, the directional classification of the pixel closest to the image axis origin and the proceeding classified direction value along the edge were used as seed values to cluster an edge. The classified value at each pixel along the edge was compared against the seed values. An endpoint along the edge was determined when the classified direction changed from the seed values. The change in classified direction was required to be two pixels or longer to compensate for noise along the edge. Therefore, if the value of a pixel was different to the seed values, but was shorter than two pixels in length, the pixel was clustered to the same line. When the connectivity value was different to the seed values and the length of the pixels with the different connectivity value exceeded the length threshold, an endpoint was determined for that line. The next two unique direction values after the endpoint were used as seed values and the clustering of the edge continued.

This process is illustrated in Figure 4.14. The Freeman chain code window was placed on each pixel along the edge, which was then classified based on the location of the neighbouring pixel in the window. Each grid cell in Figure 4.14 represents an image pixel, with the pixels of the edge image having been classified using the above approach. The axis origin [1,1] is from the top left corner of the image extent and the closest edge pixel is in cell [1,2]. From this position, the first two uniquely classified direction values are 8 and 7, which were used as seed values. Pixels were then clustered together if they had the same value as the seed values or were shorter than two pixels in length. The pixel at location [6, 3] has the classified value 1, which is different to the seed values, but shorter than two pixels in length. This was therefore clustered to the previous pixels. The value at location [9, 3] has a value 6, which runs for two pixels. This was therefore classified as the start of a new line segment and a corner position. The next two unique values along the line segment were taken as seed values, which in this example are 6 and 5.

1	1	1	1			1			
				1	1		1	1	
								1	
								1	
							1		

8	8	8	7			7			
				8	1		8	6	
								6	
								5	
							6		

Figure 4.14 – An example of (left) a detected edge, and (right) the edge pixels classified using the Freeman chain code.

The length of the fragments with the same classification was determined using run length encoding, which measures the number of pixels between the start and end of each segmented edge. The code for this was written by Eireiner (2004) and Kendall (2010), which was also downloaded from the MATLAB file exchange website. Due to the difference in nature of the nDSM boundary and the ridgelines, slight variations on the run graph vectorisation approach were developed and will be explained further in Section 4.6.1 and 4.7.1, respectively.

4.6 LOD1 Reconstruction

LOD1 building models were reconstructed from the nDSM boundary. A closed-cycle network of roof boundary points was extracted from the rasterised nDSM boundary using the developed run graph vectorisation approach. Corners were then refined and the height of each corner was calculated from the nDSM.

4.6.1 Segmentation of nDSM Boundary

The boundary of the nDSM formed a closed loop of pixels. Once pixel connectivity had been measured using the Freeman chain code and the procedure from Section 4.5.2, all pixels in the nDSM boundary were listed in a single array with the assigned direction value. The length of each classified fragment was measured using run length encoding. The pixel closest to the image axes origin was defined as the first corner point with the seed values defined as the first two unique connectivity values. The line tracing procedure compared the classified direction values of all pixels in the nDSM boundary, in an anti-clockwise direction, until the classification value of all pixels had been compared and line segments were formed, between two endpoints.

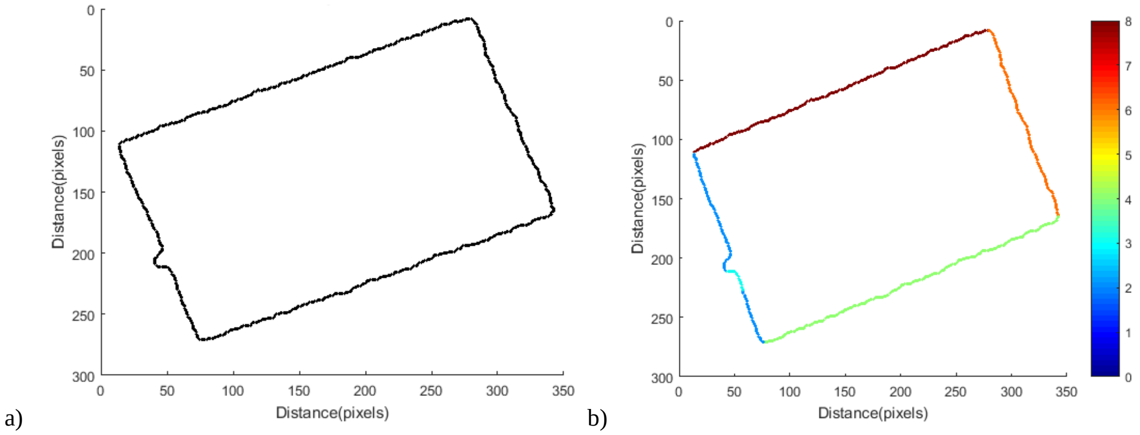


Figure 4.15 – Extraction of boundary edges: a) the unclassified nDSM boundary from Figure 4.5; b) the classified nDSM boundary using the Freeman chain code.

Examples of the nDSM boundary segmentation are shown in Figure 4.15. The image in Figure 4.15a shows the nDSM boundary, obtained from the subtraction of the ground classification (Figure 4.5c) from the DSM (Figure 4.5b). The pixel connectivity was measured and the pixels were classified based on the direction of the neighbouring pixel using the Freeman chain code, shown in Figure 4.15b. This result shows over-segmentation of an edge due to the inclusion of a chimney in the nDSM boundary, which caused a small curved extension on the nDSM boundary. This was later refined using the corner refinement procedure.

4.6.2 Corner Point Refinement

The boundary corners were firstly connected to potential roof-lines during the connectivity ruleset outlined in Section 4.7.2.2. To connect boundary corners and roof-line endpoints, two criteria were tested. Figure 4.16a and Figure 4.16b show an unconnected roof-line endpoint (yellow) and a nearby boundary corner (red). If the boundary corner was connected to other roof-lines, the angle at the unconnected edge, from its opposing connected endpoint to the boundary corner was measured, illustrated in Figure 4.16b. If the angle was within 180° ± 15° then the unconnected edge was connected to the boundary corner. The ±15° tolerance was determined based on sensitivity testing, with the results shown in a histogram of angles in Appendix C.2. The range of angles calculated for several buildings from unconnected endpoints to potential connecting endpoints are shown in Figure C.5 in the histogram split in 5° bins. The tolerance of ± 15° was chosen to be an acceptable threshold between limiting connections to correct edges without being too restrictive, and ensuring that incorrect connections were not made.

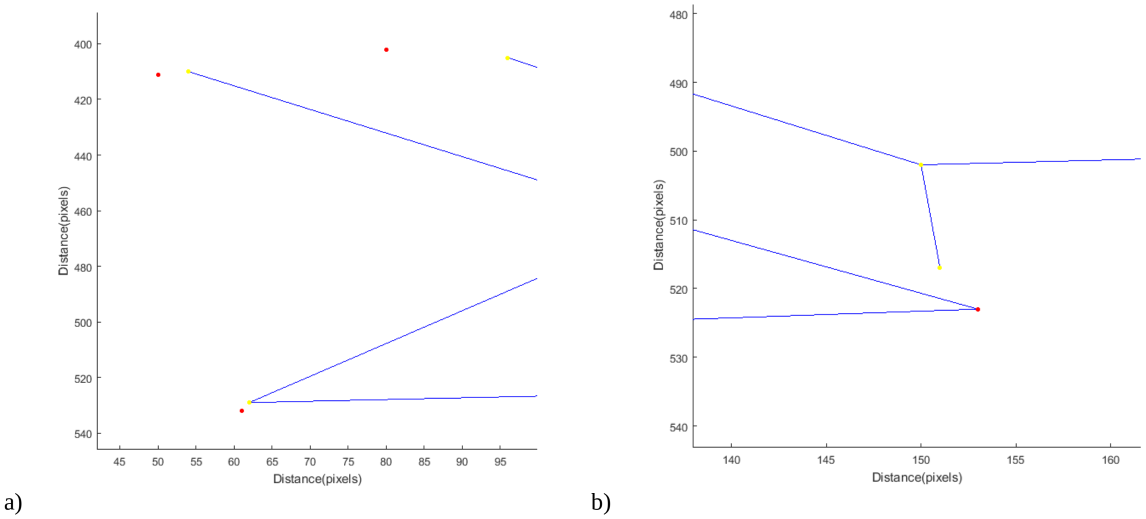


Figure 4.16 – Connectivity of boundary corners and roof-lines endpoints: a) example of measuring the angle at a boundary corner (red) and unconnected endpoint (yellow); b) example of extending an unconnected line to a connected boundary corner.

If the boundary corner was not connected to any other roof-lines, the unconnected endpoint and boundary corner were presumed to connect, and the location of the connection was determined based on orthogonal angles, as shown in Figure 4.16a. The angle at the boundary corner within the search window, from the previous boundary corner to the next boundary corner was calculated, and compared to the angle at the unconnected endpoint to the same two boundary corners. The angle closest to 90°, or 180° if both angles were above 135°, was chosen as the location of the corner and the other endpoint/corner, with the larger difference, was adjusted accordingly.

Once connectivity of roof-lines and boundary corners was measured, the orientation was then calculated for each line segment along the boundary. The orientation of each line was measured using the inverse tangent of the two endpoints of each line. The orientation of the building was determined from the longest boundary line and the orientation of other edges was compared against this. The orientation of all lines was limited to within 45° intervals of the main building orientation. This was to limit roof-lines to be within the main eight directional axes, thus not just limiting edges to being either parallel or perpendicular. If the orientation of any line compared to the longest line was not within a ±5° threshold of the limits, then this edge was removed. Previous attempts on corners constraints, presented in Chapter 2, have enforced angles at 90°, which limited the shape to rectangular buildings. By using intervals at 45° this should not limit reconstruction to buildings with rectangular footprints and orthogonal corners. The threshold tolerance was determined based on sensitivity testing with example results shown in Appendix B.1 where the blue points represent an edge and the red vertical lines represent the threshold tolerance of the main orientations. Boundary edges not connected to a roof-line and not within the orientation thresholds were removed by merging line segments. The results in Appendix B.1 show how nearly all edges fall within a ±5° tolerance of 45° intervals from the main orientation, whilst edges that were slightly outside this tolerance tended to be connected to roof-lines so were not removed. All other edges were removed as incorrect boundary lines.

The refinement of boundary corners using line orientation can be seen in Figure 4.17. The boundary edges and corners that were extracted from the segmentation in Figure 4.15 are presented in Figure 4.17a. A short edge was extracted from the classification of the nDSM boundary due to the presence of the chimney included in the segmentation. By measuring the orientation of the longest edge and comparing the orientation of all other edges to this, the short edge was removed and produced the results in Figure 4.17b. For this example, it can be seen that the correct number of corners and edges were extracted, which were all connected as a closed-cycle.

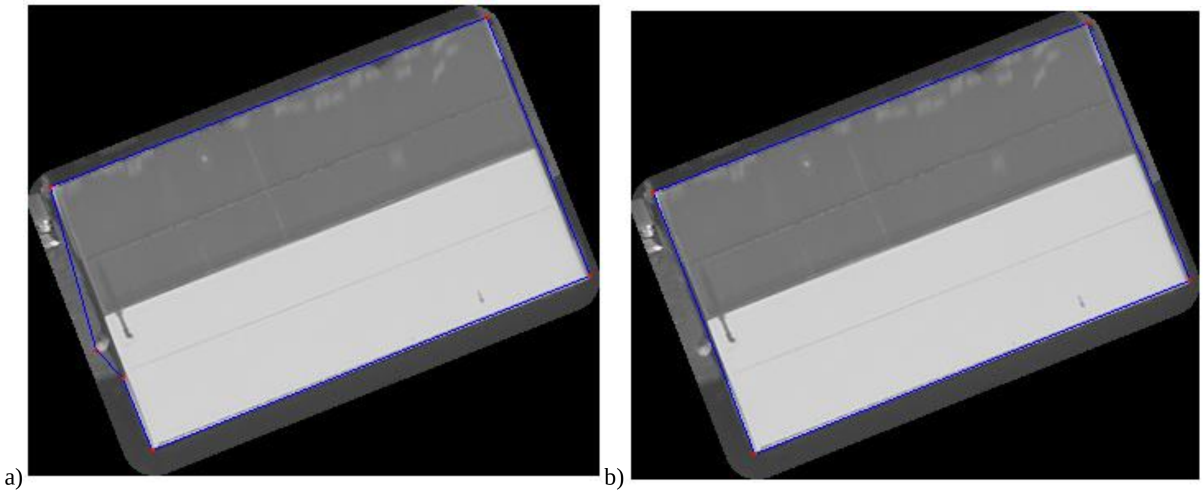


Figure 4.17 – Refinement of boundary corners and edges: a) the corners and edges extracted from the segmentation of the nDSM boundary; b) the corners and edges of the boundary after refining using the constraints.

4.6.3 Height Calculation

A workflow was formulated for the extraction of height values from the nDSM for each corner point. Firstly, a height value was assigned to each corner point using a gradient based approach. A window around each boundary corner point extracted all height values within a neighbourhood of 11 x 11 pixels, to find all height values within 0.5 m of the corner. The numerical gradient, calculated as the first derivative along the X or Y axis, was determined for the central X and Y cross-section of the window, with the extracted corner at the centre of the window. The window size meant the gradient values were calculated for a wide range of height values. The numerical gradient calculation was limited to the central cross-sections of the window to prevent height values far away from the corner point influencing the height determination. Height values with a numerical gradient above 0.2 m and below 5 m were extracted from the nDSM, for both the central X and Y cross-section of the neighbourhood. The histograms in Appendix B.2 support the selection of these thresholds. The graph in Figure B.5a show all the gradient values calculated. The maximum height value extracted within the thresholds of the numerical gradient was assigned to the corner point. The maximum threshold of 5 m was introduced so that the numerical gradient calculated along the boundary of the nDSM and from no data values of the background were not included. The graph in Figure B.5b shows part of the gradient histogram for the smallest gradient values. A large number of small gradients (from -0.05 m to 0.05 m) can be seen from where heights were taken along the roof surface. This bell shape curve tails off at approximately -0.2 m, and slightly larger on the positive side of 0.4 m. However, it was decided to use the smaller range from 0 to determine the minimum gradient threshold increase of ±0.2 m.

If no height values were extracted using the numerical gradients procedure, then the corresponding height value from the nDSM was assigned to the corner. If, however, the height value at the corresponding pixel in the nDSM was 0, the maximum height value from the nDSM, within a 5 x 5 pixel window of the corner was assigned to the endpoint.

4.6.4 Ground Point Extraction and Façade Reconstruction

Each corner point of the roof boundary was used to construct a ground point. The network of roof boundary corners was used to produce the network of ground edges at the base of the building. The mean height within a 20 x 20 pixel window was extracted for all ground corners. The minimum height value extracted from the window of all ground corners was then applied to all ground corners. A vertical edge between the ground point and the roof boundary point was created in order to produce façades.

By using the refined boundary corners in Figure 4.17b to extrude the eave lines to a ground height, a block model was reconstructed for the input data in Figure 4.5, as shown in Figure 4.18. The ground heights were calculated from the DTM in Figure 4.5c.

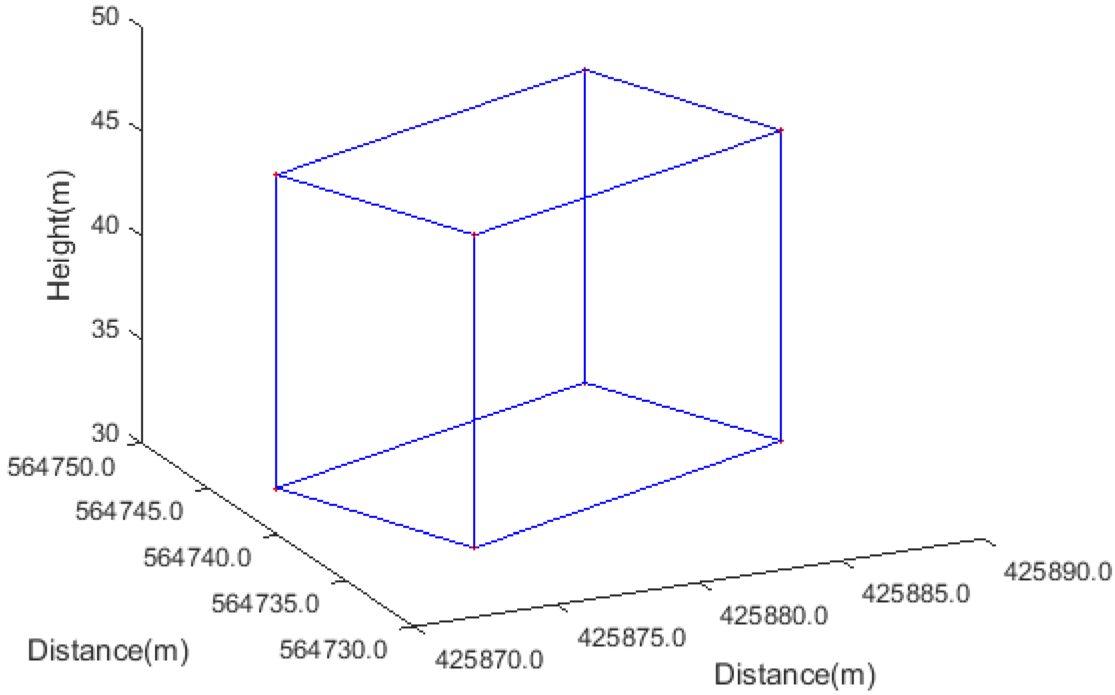


Figure 4.18 – LOD1 building model of the gable roof in Figure 4.4.

4.6.5 Exporting 3D Building Model

The real world XY coordinates of each roof corner were calculated using the TIFF world file (tfw). A tfw file contains metadata about the image and its real-world location. Information such as pixel size, datum and real world coordinates of the bounding box are stored and were used to convert corner points from image coordinates to object coordinates. The origin of the axes in the MATLAB image coordinate space start in the top left corner, shown in Figure 4.19a. The axes of the object coordinate space start from the bottom left corner, shown in Figure 4.19b. The real-world coordinates of each corner (X_w, Y_w) were calculated from the image coordinates (X_i, Y_i) and the real-world coordinates of the top-left corner of the bounding box of the image (X_{bb}, Y_{bb}) using Equation 4.1. Because of the change in coordinate reference axis, the X image coordinate was added to the X real world coordinates of the image bounding box, and the Y image coordinate was subtracted from the Y real world coordinate of the image bounding box. The image coordinate was multiplied by the pixel size to determine the real-world distance from the bounding box coordinate to the image coordinate.

$$X_w = X_{bb} + X_i * p \qquad\qquad Y_w = Y_{bb} - Y_i * p \tag{4.1}$$

Where

$X_{bb} = \ X\ coordinate\ for\ the\ top\ left\ corner\ of\ the\ image\ bounding\ box$

$X_i = image\ coordinate\ of\ X$

$p = pixel\ size$

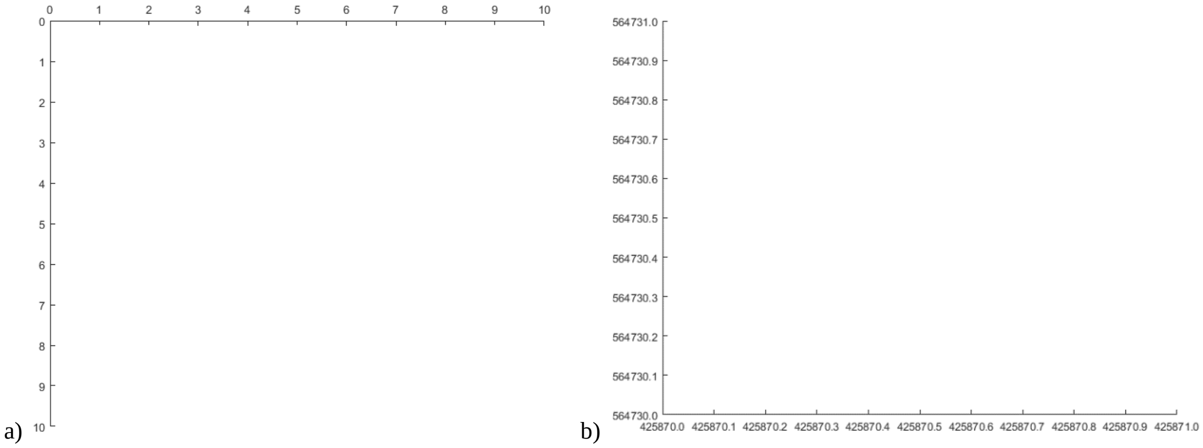


Figure 4.19 – Coordinate axis: a) MATLAB image axis and b) real world coordinate axis in British National Grid.

Each roof line was exported as a comma-separated value (csv) file to be imported into other software packages. The 3D model can be reconstructed in ArcMap using the *ASCII 3D To Feature Class* tool in the 3D Analyst toolbox (ESRI, 2016a). This tool creates shapefiles from csv files. Thus, real world coordinates information from the csv files was used to create a 3D shapefile and produce a 3D model.

4.7 LOD2 Reconstruction

LOD2 building models were created by combining the points extracted from LOD1 reconstruction to the points extracted from the segmented roof-lines. A similar procedure to LOD1 reconstruction was used for LOD2 reconstruction, with a slight adjustment for the line segmentation and an expanded connectivity ruleset to produce closed connected lines that formed the boundary of roof planes, and enforced geometric and topological constraints.

4.7.1 Segmentation of Roof-Lines

The extraction of corner points from the detected edges of the roof required a slightly different approach for run graph vectorisation, compared to the nDSM boundary, due to the complexity of the roof-lines and the edges not being connected in a single loop, as was the case for LOD1 reconstruction. Each edge pixel was classified using the Freeman chain code, as previously explained in Section 4.5.2. Due to all pixels not forming one closed loop, line segments were formed and stored in separate arrays of edge pixels for separate unconnected line segments.

To start the line tracing, the pixel with the coordinates closest to the image origin axes was selected and the image coordinates of the edge pixel were stored in an array. The Freeman chain code window (Figure 4.13) was used to measure the connectivity and direction of neighbouring pixels. All the connecting edge pixels were stored in the same array as the start pixel, along with the classified direction of travel. The array was completed when the end of the edge pixel segment was reached and a new array of edge coordinates was created with the start pixel of the new segment and its directional classification. The measurement of connectivity was straightforward when the centre pixel of the window only had one connecting pixel, thus only one direction could be classified. Special considerations were required for edges which had two or more connecting pixels. Different criteria were constructed based on the number of neighbouring pixels.

The first criterion determined if only two neighbouring pixels were found and all the pixels in the window were connected to each other. It was presumed that if all pixels were connected, they formed the same line segment, so were not split at the centre of the window. The connectivity classification and direction of travel of the window was determined as the North, South, East, West (NSEW) direction (value of 2, 6, 8 or 4 from Figure 4.13) from the neighbourhood of the two neighbouring pixels. This criterion is illustrated in Figure 4.20c for the unclassified pixels.

The second criterion determined if two neighbouring pixels were found that were not connected to each other. For this case, it was presumed that two line segments were formed, thus the centre point of the window was a corner. The pixel with the smallest NSEW connectivity value was chosen as the direction of travel for the window for a new line segment. If there were no connecting pixels in the NSEW directions, the smallest diagonal value (1, 3, 5 or 7 from Figure 4.13) was chosen to continue the measure of pixel connectivity. Once the endpoint of the current segment was reached, a new array was formed for a new segment of edge pixels starting from the split pixel, and the procedure would trace along the other classified direction. This criterion is illustrated in Figure 4.20b for the unclassified pixels.

The third criterion determined if more than three pixels were present around the centre pixel of the connectivity window. In this case, the centre point of the window was defined as an endpoint of the current segment. The smallest directional value in the NSEW direction, or diagonal direction if no pixel connectivity was detected in the four cardinal directions, was chosen as the classification and to trace a new segment. Similarly, with two non-neighbouring connecting pixels, once the endpoint of the new segment had been reached, the pixels in other directions were measured for connectivity and classification.

Another consideration brought into the ruleset was the measure of single edge pixels. These were pixels that did not connect to any other edge pixel. If this was the case, the image coordinates of the single pixel were stored in a separate array with the classification value 0 to define no connectivity.

For all criteria, greater emphasis was applied to pixels travelling in a NSEW direction compared to the diagonal directions. By emphasising the NSEW direction over the diagonal directions, this prevented the line self-intersecting and measured the full connectivity of the edge pixels in the segment. If emphasis was applied to the diagonal direction, the segment may not measure all edge pixels in a continuous segment. This is illustrated in Figure 4.20.

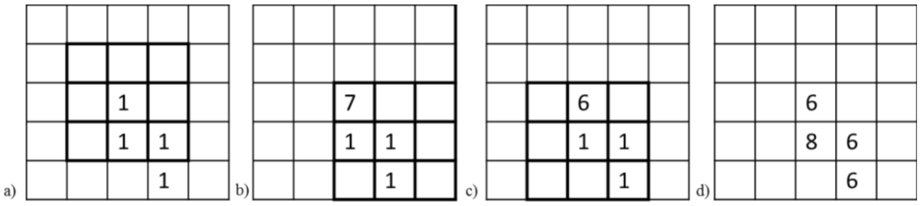


Figure 4.20 – Forming lines from the classified edge pixels: a) a section of edge pixels that form one segment with the connectivity window outlined in bold; b) if the diagonal pixel was chosen (classification value of 7) for connectivity, the window traverses in a south-east direction. This would form an endpoint at the centre of the window based on the second criterion (listed above) as the two pixels were not connected; c) by emphasising the NSEW direction and using the first criterion the pixels of the segment are traversed along cardinal directions; d) the final classification of the segment.

Once all the edge pixels had been measured for connectivity and the segments had been formed, individual lines were constructed by measuring the dominant classified directions along the segments. The clustering of edge pixels along a segment to form lines was undertaken using the same clustering method outlined in Section 4.6.1. To further refine the constructed lines, any line shorter than a fixed threshold of 0.3 m was removed as noise. Upon visual comparison of the edge detection, noise edges were found to consist of two or three pixels, which equates to a maximum distance of 0.3 m.

This classification and extraction of roof-lines can be seen in Figure 4.21 from the refined edges in Figure 4.7. The segmented roof-lines, shown in Figure 4.21a show classification of the main ridgeline and the connectivity with some gable lines and eave lines. The definition of these lines can be seen in Figure 1.2. The vectorised lines in Figure 4.21b show how a number of short edges classified in Figure 4.21a were extracted, but many of these edges were removed using the length threshold. Single pixels, classified with a directional value of 0 were not removed, as some single pixels can be seen in Figure 4.21b to be in the correct corner location of the roof.

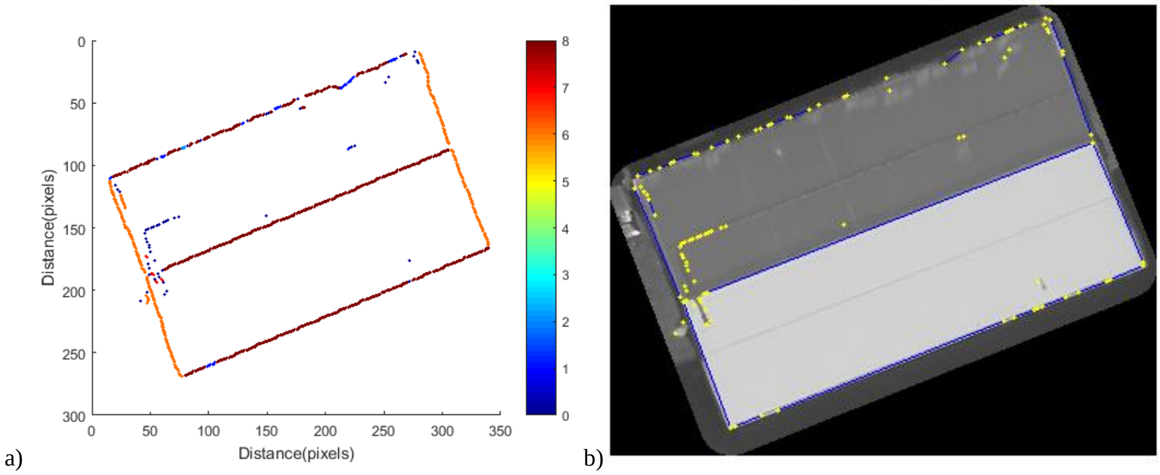


Figure 4.21 – Segmentation of extracted roof-lines: a) the classified raster edges using Freeman chain code; b) the vectorised view of the segmented roof-lines.

4.7.2 Topological and Geometric Constraints

To create connected roof edges, which formed the network boundary of planes for a 3D roof model, a number of connectivity rulesets were tested between each unconnected endpoint and its surrounding neighbouring endpoints. Firstly, the length and orientation of each edge was measured. The main orientation of the roof was calculated as the orientation of the longest roof-line. The length was calculated using Pythagoras theorem. The roof orientation and length were used to refine the number of roof edges. Orientation thresholds in intervals of 45° from the main orientation were used to remove edges which did not conform to the orientation of the longest roof-line. A ±2.5° tolerance of each orientation interval gave a 5° threshold for each orientation value. Edges that were shorter than 5% of the longest line were also removed. However, lines that were shorter than 5% of the longest line but within the orientation threshold intervals were kept. The use of these parameters from the longest roof-line, which was presumed to be a correct roof edge, meant that adaptive thresholds were applied for each building. These tolerance thresholds were determined based on sensitivity testing, with results in Appendix C.1. Each blue point on the graphs in Appendix C.1 represents an edge plotted at its corresponding length and orientation value. The red vertical lines represent the different orientation tolerances tested and the horizontal green lines represent lengths of 5% and 10% of the longest roof-line. It can be seen from the results in Appendix C.1 that most of the edges were clustered within a ±2.5° tolerance at 45 ° intervals from the largest calculated orientation, and that most of the noise edges had a length shorter than 5% of the longest roof-line. By using a threshold larger than ±2.5° meant an increased number of shorter edges would have been included for connectivity analysis and would potentially lead to incorrect connectivity between roof-line endpoints.

4.7.2.1 Roof-Line Segmentation Correction

The first phase for the formation of connected roof-lines was to correct for segmentation errors by calculating the connectivity angle between unconnected endpoints and roof-line pixels within a defined area. An example of the sort of errors to correct can be seen in Figure 4.22. The results of clustering edges based on pixel connectivity can be seen in Figure 4.22a. Two edges have been over-segmented, which has resulted in a valley line not being extracted, shown in Figure 4.22b, where a corner of the central ridgeline has been extracted in the middle of a connecting valley line. Two edges have also been over-segmented due to the classification starting from the pixel closest to the image axes origin, which was not the endpoint of the edge segment. This resulted in two corners being extracted along the same edge and in close proximity to each other.

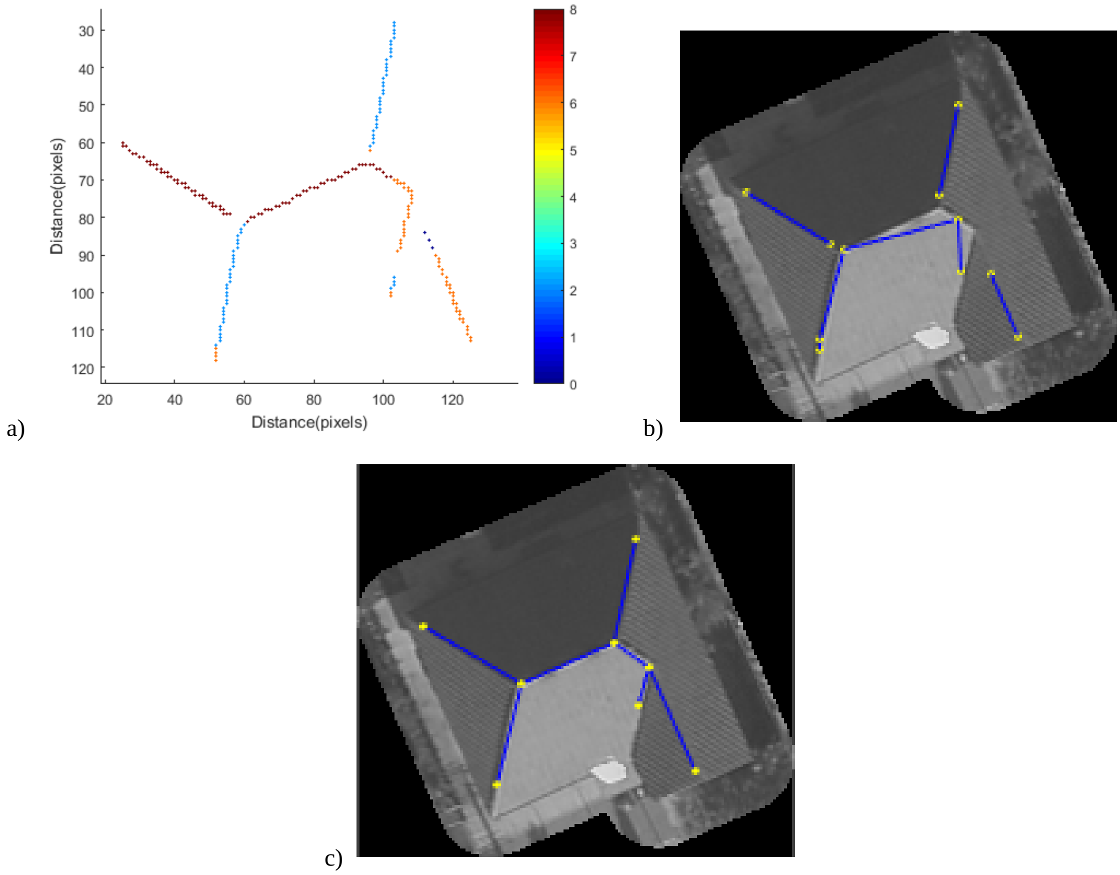


Figure 4.22 – Roof-line segmentation correction: a) Clustered edges from run graph vectorisation; b) roof-lines (blue) and corners (yellow) extracted from (a); c) refined edges using ruleset in Section 4.7.2.1.

To correct the classification of roof-lines, a neighbourhood was defined for each unconnected endpoint, to find all roof edge pixels, and the angle between the endpoint to each found pixel was calculated. The angle was measured at each unconnected endpoint, from the opposite endpoint along the line, to each pixel found within the defined neighbourhood. An unconnected endpoint was connected to a roof edge pixel if the measured angle was within an angular threshold of 180°. Potential connecting endpoints were found using an adaptive search window, with the size of the search window influencing an adaptive angular threshold. The search for connecting endpoints started with a small window and a large angular threshold. The size of the first search window was 5 x 5 pixels, denoting a 0.5 x 0.5 m window, with an angular threshold of 180° ± 15°. The angle was measured to all edge pixels within the search window and if any connectivity angle was within the threshold, the unconnected endpoint was connected to the edge pixel with the angle closest to 180°. If no edge pixels were found within the search window, or the connectivity angle of all pixels was not within the angular threshold, the size of the search window was increased to 10 x 10 pixels, denoting a 1 x 1 m window, and the angular threshold was tightened to 180° ± 10°. Again, if no edge pixels were found to match the criteria, the search window was increased to the maximum size of 15 x 15 pixels, denoting a 1.5 x 1.5 m window, and the angular threshold was restricted to 180° ± 5°. The angular threshold was restricted each time the window increased to limit the potential connectivity of edge pixels further away from the unconnected endpoint. The choice of these angular threshold tolerances can be explained in Appendix C.2, as has previously been discussed for the boundary to roof-line connectivity in Section 4.6.2.

After all unconnected endpoints were tested for potential connectivity, lines were refined to reduce the number of endpoints. Two lines were merged into a single line if one of the lines was shorter than 0.5 m or the orientation of the two lines was within a 10° tolerance. The distance threshold of 0.5m was chosen based on previous work, for example Awrangjeb *et al.* (2012), and from testing, with some example results shown in Appendix C.3. The example results in Appendix C.3 show results of combining and connecting edges based on 0.5 m and 1 m distance thresholds. Although some correct connections were made with a 1 m distance tolerance, and not with 0.5 m distance, it can be seen from the results that more incorrect connections were made with the larger distance. The common endpoint of the two lines was removed to form one line segment. Lines shorter than 1 m with an unconnected endpoint were removed as noise.

An example result of this refinement can be seen in Figure 4.22c from the line tracing results in Figure 4.22b. Corners have been refined to be in the correct location and over-segmentation of the ridge and valley lines was corrected.

4.7.2.2 Connectivity of Roof-Lines and Boundary Corners

The developed ruleset for the final connectivity of roof-lines and boundary corners used a larger search window and different connectivity rules compared to the refinement of the roof-line segmentation. This was to include and test more rules between potential connecting endpoints, which will be explained further, as well as to allow larger gaps between roof-lines to be connected. The refinement of roof-line segmentation used all edge pixels from the raster image of the roof to correct connectivity issues, whereas the following ruleset only considered the vector representation of each roof-line and its endpoints.

For each unconnected endpoint, a circular search window, with a defined radius, was used to find potential connecting endpoints. The length of the longest roof-line was calculated and used to determine the radius of the search window. The radius of the search window was either 20, 40 or 60 pixels, which denoted a 2 m, 4 m or 6 m search window radius. The radius of the search window was determined based on the length of the unconnected line, with respect to the longest roof-line length. If the length of the unconnected line was shorter than an eighth of the longest line, the search window was limited to a 20 pixel radius. If the length of the unconnected line was longer than an eighth, but shorter than a quarter of the longest line, than the search window was limited to a 40 pixel radius. Any unconnected line longer than a quarter of the longest length used a maximum search window, with a 60 pixel radius, to find potential endpoints. All the potential connecting endpoints found within the search window were ordered based on the distance from the centre of the search window, with those closest to the unconnected endpoint tested first for connectivity. A number of rules were developed, alongside the roof-line to boundary corner connection ruleset outlined in Section 4.6.2, to test connectivity:

- If two potential connecting endpoints from two different lines were found with the same distance from the centre of the search window, then the intersection of the three lines was found to form the connecting endpoint of all three lines. Figure 4.23a shows the endpoints (yellow) of two roof-line segments (blue) and a nearby boundary corner (red). The boundary corner and one of the unconnected endpoints are equidistant from the other unconnected endpoint. To connect them, the intersection of the three edges was calculated and determined as the endpoint of all three lines;
- If the distance between the two endpoints was less than or equal to 0.5 m, the endpoints were merged together. Points were merged by either moving the unconnected point to the connected point, or if both points were unconnected finding the median point. This is illustrated in Figure 4.23b, where the endpoint of the unconnected line was moved to connect to the already connecting endpoint;

- If both endpoints were unconnected and the orientation of both lines was within a 10° tolerance, the connectivity of the lines was tested. This was to compensate for endpoints whose angle was not within the angular threshold of 180°, but should have been connected, as shown in Figure 4.23c. As a further check the angle at each unconnected endpoint, from its connected endpoint along the line, to the opposing endpoint of the neighbouring unconnected line was measured. If both angles were within 180° ± 15°, then the median point was calculated to connect the two unconnected endpoints;
- If two endpoints were unconnected, it was not necessarily the endpoint at the centre of the search window that needed to be adjusted. Therefore, if the endpoint found within the search window was unconnected, then the angle at this unconnected endpoint from its connected endpoint along the roof-line, to the endpoint at the centre of the window was measured. If the calculated angle was within 180° ± 5°, then the found endpoint was connected to the endpoint at the centre of the search window (Figure 4.23d). More emphasis was placed on adjusting the position of the endpoint at the centre of the window, rather than those found within the search window, which is why a stricter angular threshold was used;
- If none of the criteria above were true, then the angle at the unconnected endpoint at the centre point of the search window, from its connected endpoint along the roof-line, to the found endpoint was calculated. If the angle was within 180° ± 15°, the endpoint at the centre of the window was connected to the found endpoint (Figure 4.23d).

The above criteria are illustrated in Figure 4.23a – d. The criteria were tested for each endpoint found within the search window, and if any of the criteria above were found to be correct, the endpoints were connected. A search was then performed to find potential connecting endpoints for the next unconnected endpoint in the list and tested. If none of the criteria above held true, further tests were undertaken to determine if by extending the line a specified distance, an intersection point with another line could be determined, to form a new corner and connection. Similar to the radius of the search window, various lengths to extend the line, along the same orientation, were defined based on the length of the unconnected line, with respect to the longest roof-line. If the length was shorter than an eighth of the longest roof-line, the unconnected endpoint was extended by 1 m. If the length was longer than an eighth, but shorter than a quarter of the longest roof-line, the unconnected endpoint was extended by 1.5 m. Else, if the length was longer than a quarter of the longest roof-line, an unconnected endpoint was extended by 2 m. These distances to extend the line and determine any intersection points were based on the CityGML positional accuracies for LOD2 models (Kolbe, 2009). If an intersection point was found by extending the line, the unconnected endpoint was extended to the intersection point and formed a closed roof-line. The roof-line upon which the unconnected endpoint intersected was split into two lines at the intersection point. This type of connection is illustrated in Figure 4.23e, where a line segment has been extended a set distance, shown by the green line, which intersections a different roof-line. This intersection point split the roof-line to form a new corner for the unconnected roof-line.

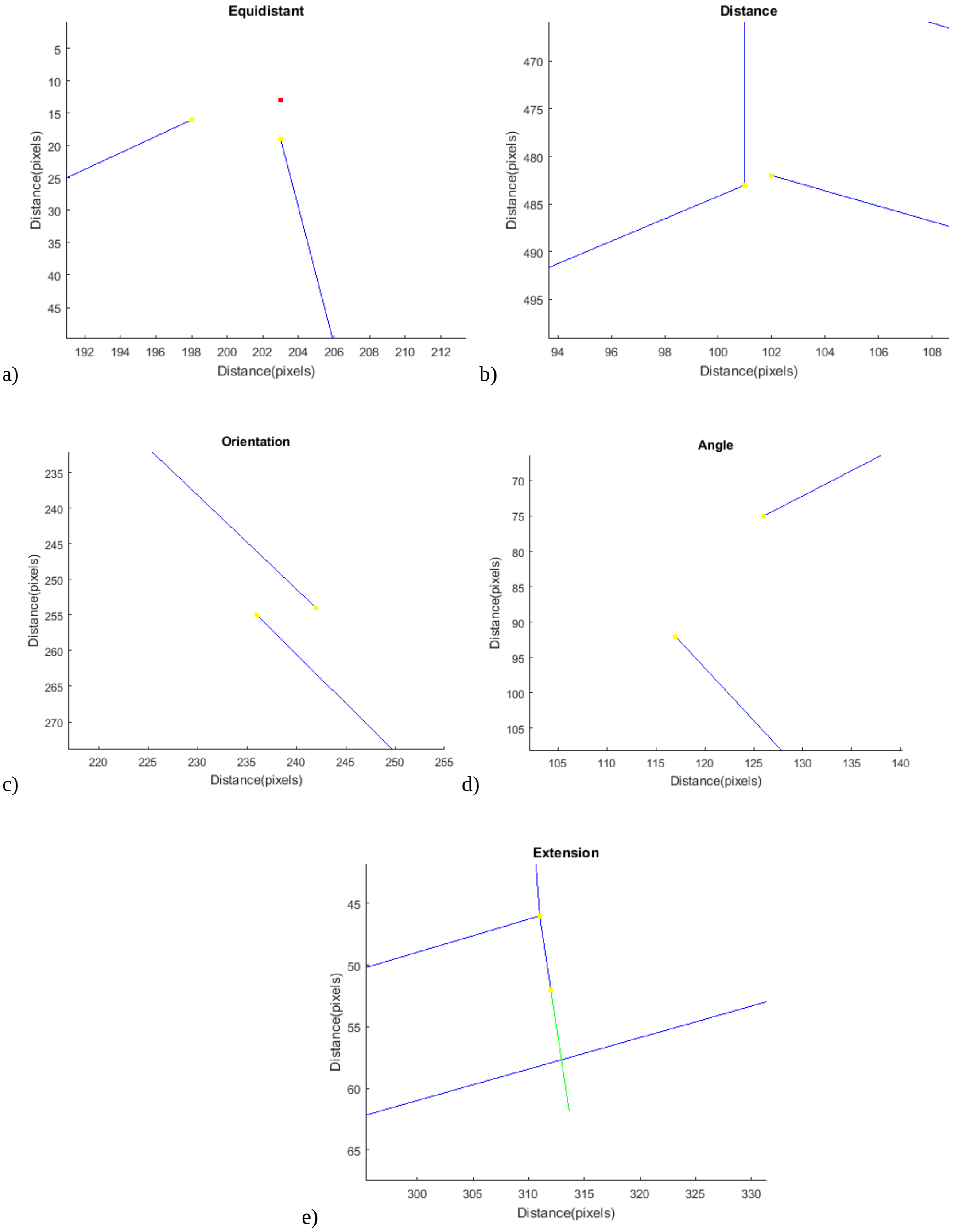


Figure 4.23 – Criteria for connecting unconnected endpoints, with roof lines shown in blue, roof line endpoints in yellow and boundary corners in red: a) if two points are found equidistant from the endpoint at the centre of the search window; b) endpoints with a distance of less than 0.5 m; c) two roof-lines which were connected by measuring line orientation; d) two unconnected points where the angle was measured between the endpoints; and e) the extension (green) of an unconnected roof-line to form a connected segment.

An example of the results of the connectivity ruleset can be seen in Figure 4.24, which has connected the segmented edges from Figure 4.21. This example shows how roof-lines were connected to other roof-lines, but also to boundary corners to form closed connected lines. This example also shows how nearly all false positives from Figure 4.21 were removed, albeit one that was incorrectly connected to an eave line. Some roof-lines still remain unconnected, due to endpoints not meeting the criteria or no endpoints being found within the search distance. These were removed and edges were refined using a final refinement ruleset in Section 4.7.2.3.

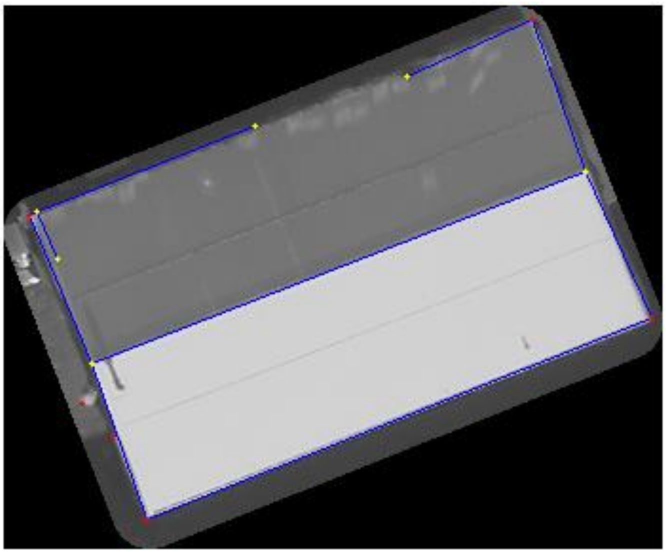


Figure 4.24 – Example of connected roof-lines and boundary corners using the connectivity ruleset outline in Section 4.7.2.2.

4.7.2.3 Refinement of Lines and Production of Closed Network

Once all unconnected lines were tested for their connectivity, a final ruleset removed all unconnected lines and refined roof-lines that consisted of multiple line segments. This was undertaken as an iterative process until all unconnected lines were removed.

To start the iterative process, all edges with at least one unconnected endpoint were found and removed. The orientation and length were then recalculated for each line segment to merge similarly orientated and connected lines. Similar to the refinement of roof edge segmentation, outlined in Section 4.7.2.1, if the two connected lines had orientation values within a 10° tolerance or one line segment was shorter than 0.5 m, the two lines were merged together. A different ruleset was introduced to correct nDSM boundary connectivity errors. If a boundary endpoint was connected to a roof-line, thus connecting three line segments, the orientation of all the lines was compared. The comparison was to find the correct location of an orthogonal corner that may have been correctly detected with the Canny edge detector but over-segmented from the Freeman chain code classification of the nDSM boundary. Over-segmentation may have occurred if the nDSM boundary over-generalised the corner of roof boundary lines, thus producing the same directional classification values and failing to extract the corner correctly. The difference in orientation was compared between the three different lines and if any two lines had an orientation difference below the 10° threshold, the angle between the line segments was measured. A 90° ± 15° threshold was included to ensure a boundary point could not be refined, if the third line was perpendicular to the two connecting lines. If the orientation difference of the third line was outside these parameters, two angles were calculated. The first angle was calculated at the boundary point, from the previous boundary corner to the next boundary corner. The second angle was calculated from the opposite endpoint of the connected roof-line at the boundary corner under investigation, between the two boundary corners used in the calculation of the first angle. If the first angle was closest to 90° then the boundary corner was not adjusted. If the second angle was closest to 90°, the opposing end of the roof-line from the boundary corner segment was determined as the boundary corner and the boundary points were adjusted accordingly.

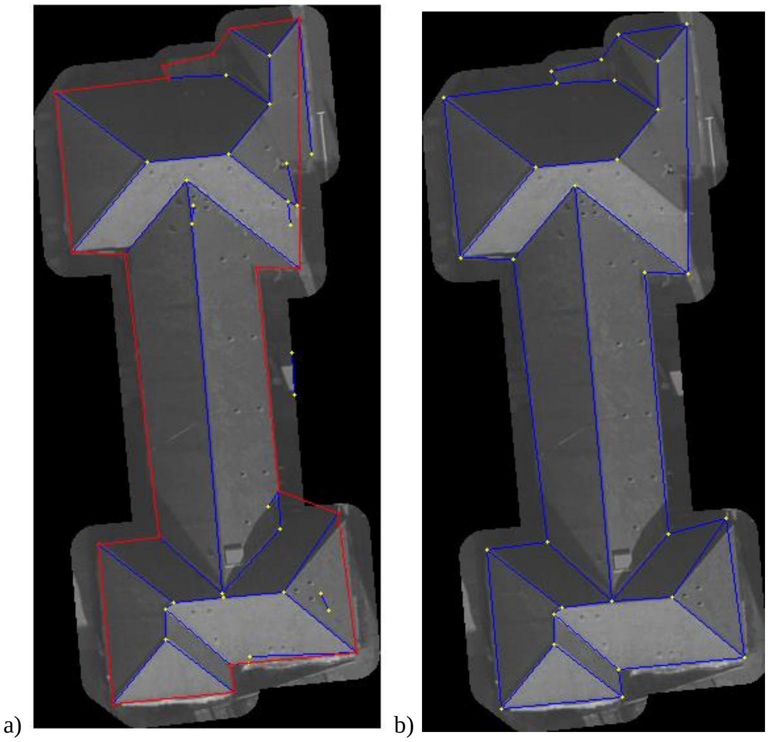


Figure 4.25 – Refinement of boundary corners with connected roof lines: a) the resulting connected roof lines (blue) and boundary lines (red) from the connectivity ruleset in Section 4.6.2 and Section 4.7.2.2; and b) the final LOD2 connected roof model.

This type of boundary corner refinement is illustrated in Figure 4.25. The connected roof-lines (blue) using the connectivity ruleset from Section 4.6.2 and Section 4.7.2.2 are shown in Figure 4.25a, with the boundary lines presented in red. It can clearly be seen that one boundary corner from Figure 4.25a has been extracted at the incorrect location, but a connecting roof-line has been extracted at the correct corner position. The boundary line and roof-line have similar orientations, and by measuring the angle between the endpoints, this boundary corner can be successfully reconstructed, as shown in Figure 4.25b.

The merging of line segments and refinement of boundary corners can produce duplicate edges by removing coincident endpoints between two lines. Therefore, for each line, any duplicated lines were removed. The connectivity ruleset discussed in Section 4.7.2.2 did not create a corner if a roof-line was extended over a pre-existing roof-line to connect to another endpoint. Therefore, the intersection of all roof-lines was checked to determine any intersection points and created corner positions by splitting any intersecting lines. The final check, for each iteration, measured the length of each line segment to remove any lines shorter than 0.5 m. These lines may have been formed from the intersection of roof-lines or during the connectivity ruleset from Section 4.7.2.2. If a roof-line shorter than 0.5 m was connected to other lines, the connecting roof-line was extended to the opposing endpoint of the short line and used as the new endpoint of all connecting roof-lines. Once all these refinements had been checked, all endpoints were assessed to find any unconnected lines. Whilst unconnected lines were removed before any refinement was completed, other edges may have become unconnected during this refinement; either from the refinement ruleset or by the connection to an unconnected line that was removed. Therefore, if any unconnected lines were still present, the refinement loop was repeated. If no unconnected lines were found, then heights were assigned to each detected corner.

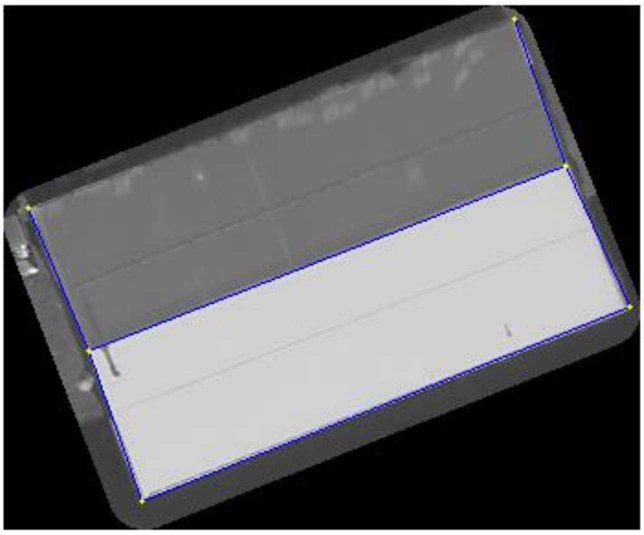


Figure 4.26 – Final connected roof-lines for a gable roof.

An example of the refinements listed above can be seen in Figure 4.26, from the results in Figure 4.24. The central ridgeline and four gable lines of the roof have been reconstructed, with the three unconnected edges in Figure 4.24, having been removed. Although it can be seen in Figure 4.26 that one of the eave lines has not been reconstructed from the segmented roof-lines, causing two gable lines to appear to be unconnected, these results were from the segmented roof-lines only. The two unconnected gable lines were connected to boundary corners from the LOD1 model, created using the workflow described in Section 4.6. This also reconstructed the boundary edge that was missing in Figure 4.26.

4.7.3 Height Extraction and Step Edge Creation

The same height extraction procedure as described in Section 4.6.3 was used to assign height values to the roof-line endpoints. If a roof-line endpoint was connected to a boundary corner, then the same height as the boundary corner was assigned to the roof-line endpoint. If the roof-line endpoint was not connected to the boundary the numerical gradient was calculated. A maximum threshold was not included during the calculation, so the maximum height value of any numerical gradient above 0.2 m, along the central row and column of an 11 x 11 pixel window, was assigned to the roof-line endpoint. If no height value was extracted using the numerical gradient of the nDSM, the height value at the corresponding pixel was assigned to the ridgeline endpoint.

Using the height values of the boundary points, step edges were determined along the eave lines. The connectivity ruleset only defines a connection of endpoints in 2D, so does not consider the height values of the endpoints. For a building with small outhouses attached to the main building, the difference in height must be determined for LOD2 reconstruction. To determine a step edge, the height difference at the two endpoints of an eave line was measured and if the difference was larger than 2 m, a step edge was determined. The 2 m height threshold was determined based on CityGML LOD2 height accuracy (Kolbe, 2009). The location of the step edge was determined with different approaches, depending on the point under investigation. For the first boundary corner in the sequence, the height was compared against the last boundary corner and the second boundary corner to test for a height difference above the threshold. The comparison of the three corners was to determine whether the first, last or second boundary corner in the closed-cycle network of boundary corners was the step edge, by calculating if the height difference between the three corners was above the threshold. The position of the step edge was determined depending on which line segment had the smallest height difference. If the height difference between the first and the last corner was the smallest, a step edge was created at the second boundary corner. A vertical edge was created using the height of the second boundary corner and the mean heights of the first and last boundary corners. Similarly, if the height difference between the last and second boundary corner was the smallest, a step edge was created at the first boundary corner using the mean height of the other two corners. If the height difference between the first corner and the second corner was deemed the smallest, then a step edge was created at the last boundary corner. In this case, an additional check was included to determine if any roof-lines separated the first and the last corner, and whether the roof-line endpoint was the position of the step edge. If no roof-lines separated the first and last boundary corner, a step edge was created as a vertical edge at the last boundary corner using the mean height of the first and second corner and the height of the last corner. If a roof-line endpoint was between the first and last boundary corner, the height difference between the last corner and the roof-line endpoint was compared. If this height difference was above the height threshold, a step edge was created at the roof-line endpoint, which was included in the array of boundary corners; otherwise the step edge was created at the last boundary point.

For all other points along the boundary, the height value of a corner was compared to the next corner in the sequence of boundary corners. If the height value between the two corners was found to be greater than the threshold, a step edge was determined at the point with the highest height value. The step edge was added to the boundary network by adding two new edges. The first edge used the XY coordinates of the two endpoints with both endpoints assigned the minimum height value of the step edge. A vertical edge with the same XY coordinates of the corner with the highest height value was created, with assigned minimum and maximum height values for the connecting endpoints. This comparison continued between all corners until the last corner was reached.

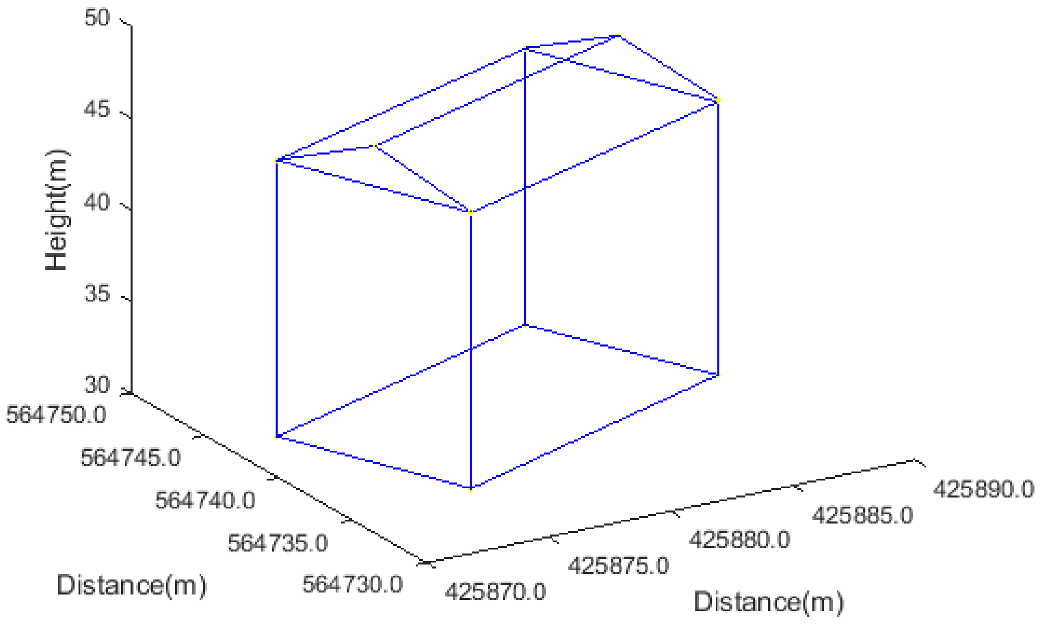


Figure 4.27 – LOD2 reconstruction of the building with a gable roof in Figure 4.5.

An example of a building model reconstructed with the correct roof geometry can be seen in Figure 4.27, from the true-orthophoto, DSM and DTM of Figure 4.5. All edges of the roof model have been correctly extracted, by the connectivity of roof corners. The two sloped roof planes have also been reconstructed from the connection of all roof edges. The roof model has been combined with the block model, shown in Figure 4.18, to model the whole building in 3D.

4.7.4 Export of 3D Building Model

By combining the block models and the roof model created in this workflow, a 3D building model with the correct roof geometry was created, as shown in Figure 4.27. The real-world coordinates of each roof corner were computed using the formulae from Equation 4.1 and each roof edge was also exported as a csv file for 3D reconstruction in other software packages, as previously described in Section 4.6.5.

4.8 Summary

An automated workflow has been presented in this chapter to reconstruct 3D building models at various levels of detail. The workflow is based on three main components: edge detection, refining the edge detection and the connectivity of roof edges.

Edge detection has been shown in Section 4.4.1 to successfully extract edges along the main structural lines of a roof, but is prone to extracting edges from unwanted artefacts such as shadow and roof texture. Even by refining the parameters of the Canny edge detector, false positives were still present, which can be removed by post-processing.

The refinement of these edges has been developed on the principles of scan line segmentation. The boundaries of the planes from surface models using scan line segmentation were deemed inaccurate by Alharthy and Bethel (2004) and McClune *et al.* (2014), with the segmentation affected by the density of the data and noise. Feature-based extraction has been shown in Section 2.3.1.1 to successfully extract boundaries of roof planes. By using only the pixels extracted by the Canny edge detector, the density of the data along each cross-section for planar boundary extraction was reduced. Not all height values along the cross-section were measured, which reduced the amount of noise in the nDSM. The density of the DSM produced from dense image matching, means that heights can be assigned to each detected edge pixel very accurately and produce a smoother interpretation of roof planes. By using the developed scan line segmentation approach nearly all false positives were removed from the edge detection images, as shown in Section 4.4.2. Those that remained generally consisted of short, fragmented edges, which were refined during the reconstruction of the building model.

The 3D building model was reconstructed as a series of connecting edges, to form a network of roof-lines and corners, which delineate the roof planes. Roof edges were vectorised using a developed line tracing procedure, shown in Section 4.5.2. The approach of Montero *et al.* (2009) was dependent on the length of the fragment and segment for determining corner points. By adapting the calculation of pixel connectivity and direction, this dependency can be removed. Roof-lines were connected based on a defined connectivity ruleset, which did not consider any planar parameters, but focussed on the connectivity of roof edges.

The methodology has been constructed to produce 3D building models at the specification defined by LOD1 and LOD2. The reconstruction of LOD1 models could be run as a separate process to the reconstruction of models with roof geometry, to produce building models without roof structures, as shown in Figure 4.2. However, the reconstruction of models with roof structures is dependent on the reconstruction of the block models, so these procedures must be undertaken together. A worked example of this can be seen throughout the chapter for a simple gable roof, which shows the success of the whole approach, to overcome any potential shortcomings. Results of further reconstruction at LOD1 and LOD2 specifications will be presented in Chapter 5, as well as testing the transferability of the workflow to a different scene, captured from a different camera, in Chapter 6.

Chapter 5. 3D Reconstruction Results of Newcastle upon Tyne, UK

5.1 Introduction

This chapter presents the reconstruction results of the building models at the LOD1 and LOD2 specification for the Newcastle upon Tyne study area, described in Section 3.5.1. A large scope of buildings and sites, including buildings from residential, industrial and city centre scenes, from the imagery were selected to give as wide a range of building geometries to reconstruct as possible. As previously explain imagery was captured over Newcastle upon Tyne at 10 cm GSD.

To quantify the accuracy of the 3D reconstructed buildings, reference data was extracted from the available stereo imagery. The procedure outlined in the ISPRS WG III/4 benchmark for 3D reconstruction was used to analyse the reconstructed results (Rottensteiner *et al.*, 2014). The Cartesian coordinates of each roof corner were extracted from the stereo imagery to compute the positional and height accuracies of each detected corner and determine how many roof corners extracted were: (1) correct, (2) a false positive or (3) a false negative: a missing edge. A roof corner was determined as being successfully extracted if the positional accuracy of a corner was within 2 m of a true corner point from the reference dataset. The 2 m threshold was determined based on the positional accuracy specification of LOD2 building models (Gröger and Plümer, 2012). Although the positional accuracy of LOD1 is required within 5 m of the truth, the tolerance was also set to 2 m for LOD1 models, as the developed methodology requires the boundary corners in order to reconstruct the roof model. Roof planes were reconstructed as a network of roof edges, and the planes extracted were also measured against reference data. The completeness, correctness and quality of each extracted roof plane were quantified by measuring the amount of overlap between an extracted roof plane and a reference plane (Rutzinger *et al.*, 2009). This was only undertaken for LOD2 reconstruction, because LOD1 building models do not model the roof geometry. As defined in the ISPRS benchmark, an overlap value of 50% between the reference plane and the extracted plane was used to determine if a roof plane had been successfully reconstructed (Rottensteiner *et al.*, 2014).

Reconstruction of 3D models from the Newcastle upon Tyne study area was tested on two different levels: by manually selecting individual buildings based on the roof geometry (Section 5.2), and by reconstructing all buildings within a defined area (Section 5.3). The red arrows on the figures throughout the chapter indicate the viewing direction for the orientation of the corresponding 3D plots.

LOD1 Reconstruction	RMSE (m)			Roof corners extracted			
	X	Y	Z	Correct corners extracted (%)	Total corners extracted (%)	Number of correct roof corners	Total number of corners extracted
Flat	0.44	0.44	0.79	79.3	104.3	69	72
Gable	0.38	0.39	0.79	91.0	104.3	93	97
Hipped	0.43	0.44	0.42	83.4	95.1	122	116
Cross-Gable	0.56	0.40	0.53	62.5	81.4	140	114
Complex	0.55	0.50	0.74	68.1	115.7	210	243

Table 5.1 – Accuracies of the extracted roof corners and the number of corners extracted for LOD1 reconstruction, grouped by roof shape.

LOD2 Reconstruction	RMSE (m)			Roof corners extracted			
	X	Y	Z	Correct corners extracted (%)	Total corners extracted (%)	Number of correct roof corners	Total number of corners extracted
Flat	0.46	0.50	0.71	85.2	181.3	80	145
Gable	0.43	0.52	0.62	70.1	117.8	163	192
Hipped	0.44	0.45	0.34	76.5	116.6	223	260
Cross-Gable	0.51	0.48	0.60	73.2	120.6	247	298
Complex	0.47	0.47	1.41	86.5	124.0	100	124

Table 5.2 – Accuracies of the extracted roof corners and the number of corners extracted for LOD2 reconstruction, grouped by roof shape.

Planar Extraction	Per-roof plane (%)			Per-roof plane (10m^2) (%)		
	Completeness	Correctness	Quality	Completeness$_{10}$	Correctness$_{10}$	Quality$_{10}$
Flat	85.5	88.6	80.1	89.7	92.5	84.5
Gable	34.3	38.7	32.5	35.8	40.8	34.2
Hipped	38.3	41.0	35.1	46.6	52.4	43.5
Cross-Gable	42.1	41.5	33.8	45.0	45.4	36.8
Complex	49.2	47.1	42.5	50.3	48.2	43.0

Table 5.3 – Planar roof extraction analysis, grouped by roof shape.

5.2 Reconstruction by Roof Geometry

Five different roof geometries were categorised for 3D reconstruction: flat, gable, hipped, cross-gable and complex. For each roof structure, 10 buildings were manually selected for reconstruction, yielding a total of 50 buildings to test the methodology. Buildings were selected from across the whole study area, outlined in Section 3.5.1, to give a wide range of building shapes and sizes. These roof shapes were chosen to determine the scale of complexity that could be reconstructed. The overview summary of results for the positional accuracy of LOD1 and LOD2 reconstructed models, based on roof shape, are shown in Table 5.1 and Table 5.2, respectively. Table 5.3 shows the results of roof plane reconstruction for LOD2 models, with planes defined as a closed network of roof edges. As followed in the ISPRS WGIII/4 benchmark, roof plane reconstruction was considered for all roof planes, and then only planes bigger than 10 m² (Rottensteiner *et al.*, 2014). A breakdown of the reconstruction by individual roof structure can be found in Appendix D.

5.2.1 LOD1 Reconstruction – Corner Extraction

The results in Table 5.1 show the number of corners reconstructed for LOD1 models and the accuracies. As can be seen in the results, the positional accuracy has good agreement for all roof structures, with an average RMSE of ±0.45 m. The associated height of each corner was not as accurate as the positional accuracy, with an average RMSE of ±0.65 m. Of the corners reconstructed, approximately 77% were in the correct location. All roof structures, apart from hipped and cross-gable, showed too many points being extracted.

Simple roof structures, flat, gable and hipped, showed more successful extraction of boundary corners compared to the more complex roof structures. This can be seen in terms of the slightly higher positional accuracy and the correct percentage of corners extracted, compared to cross-gable and complex roofs in Table 5.1. The higher percentage of correct corners reconstructed was likely to be due to the simpler nature of the building footprint. Simple roof structures generally have a rectangular footprint with long edges, and the nDSM boundary can easily be segmented to extract the four orthogonal corners. However, more complex roof structures tend to have boundaries with shorter edges, and more irregular footprints. An example of this can be seen in Figure 5.1, for the extraction of corners from the boundary of a hipped roof with small extensions. Only one edge of the small extension was reconstructed due to the segmentation of the nDSM boundary being incorrect. Errors in the nDSM boundary from ground classification and the segmentation errors will be discussed in Section 5.4.1.

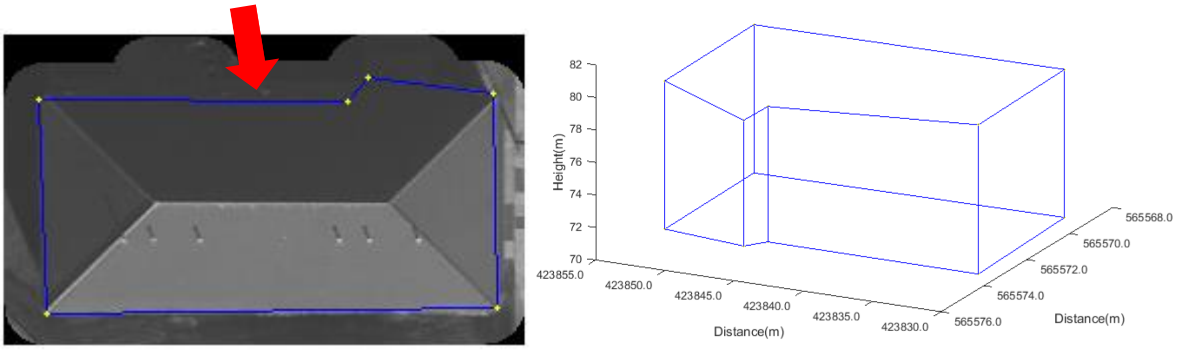


Figure 5.1 – Example of incorrect roof boundary segmentation for a hipped roof with short extensions.

Whilst all roof structures extracted incorrect corners, a difference can be seen in the total number of points extracted compared to the number of corners of the roof. The flat and gabled roof structures extracted an extra 4% of corners compared to the correct number of boundary corners. These were due to slight over-segmentation of the nDSM boundary, with an example shown in Figure 5.2 for a flat roof where five corners were extracted instead of four, represented by the five vertical edges. Hipped and cross-gable roofs extracted slightly too few corners: hipped constructed 5% too few corners and cross-gable constructed 20% too few corners. Some edges were generalised, as shown in Figure 5.1, and corners were not extracted due to segmentation errors as previously explained. The complex roof structures constructed 15% too many points as corners due to over-segmentation of the nDSM boundary.

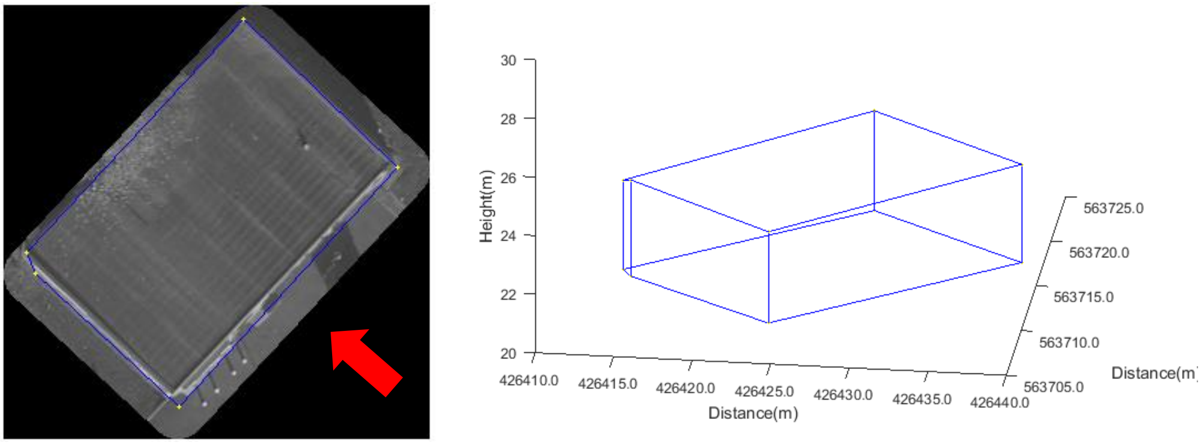


Figure 5.2 – Example of LOD1 reconstruction for a flat roof.

5.2.2 LOD2 Reconstruction – Corner Extraction

Table 5.2 shows the results of corner extraction for LOD2 reconstruction. Similar to LOD1 reconstruction, the positional accuracy of the extracted corners was higher than the accuracy of the associated height. The average positional RMSE for LOD2 models was ±0.47 m, whilst the average height RMSE was ±0.73 m.

The number of correct corners reconstructed for all roof structures was on average 78%, which is almost identical to LOD1 reconstruction. This shows that most of the roof corners were successfully reconstructed within a 2 m threshold of the correct location. When comparing Table 5.1 and Table 5.2, a difference can be seen in the correct number of corners reconstructed for flat roofs. Although there is no roof structure to reconstruct, the increase in corners reconstructed for LOD2 compared to LOD1 was due to step edges. These are not required to be reconstructed at LOD1, but are for LOD2. An example of step edge reconstructed for a multileveled flat roof can be seen in Figure 5.4a.

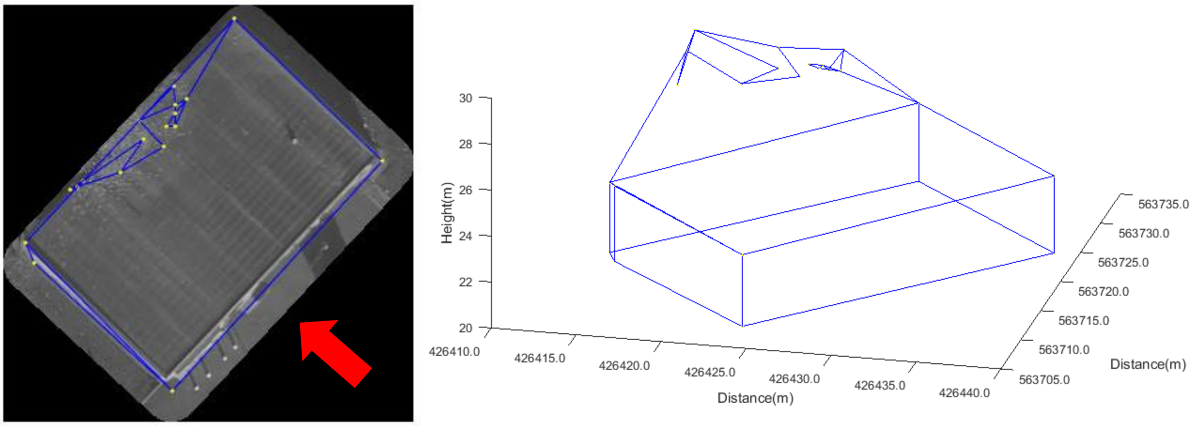


Figure 5.3 – Example of LOD2 reconstruction for a flat roof.

With regards to the number of corners extracted it can be seen that all roof types reconstructed too many corners. The flat roofs extracted nearly an extra 60% of points and constructed these as corners. An example of this can be seen in Figure 5.3, where edges were extracted and connected from overhanging vegetation. All other roof types extracted approximately an extra 18% of points and constructed these as roof corners. A number of reconstruction errors were due to edges being extracted and reconstructed along dormer windows, chimneys and other small artefacts. Examples of this can be seen in Figure 5.4 for two different roof structures. The multi-levelled roof structure in Figure 5.4a shows two faces of separate skylights that were reconstructed. The reconstructed cross-gable roof in Figure 5.4b shows how some edges from dormer windows were incorrectly connected to ridgelines, and eave lines that were correctly reconstructed. Whilst some corners and roof planes of the dormer windows appear to have been extracted in the correct location, these were not included in the reference data, as it was not the aim of this research to reconstruct small roof planes. These were therefore measured as incorrectly extracted points. However, the fact that they have been reconstructed shows the potential of this method and the datasets to extract edges from small roof geometric features.

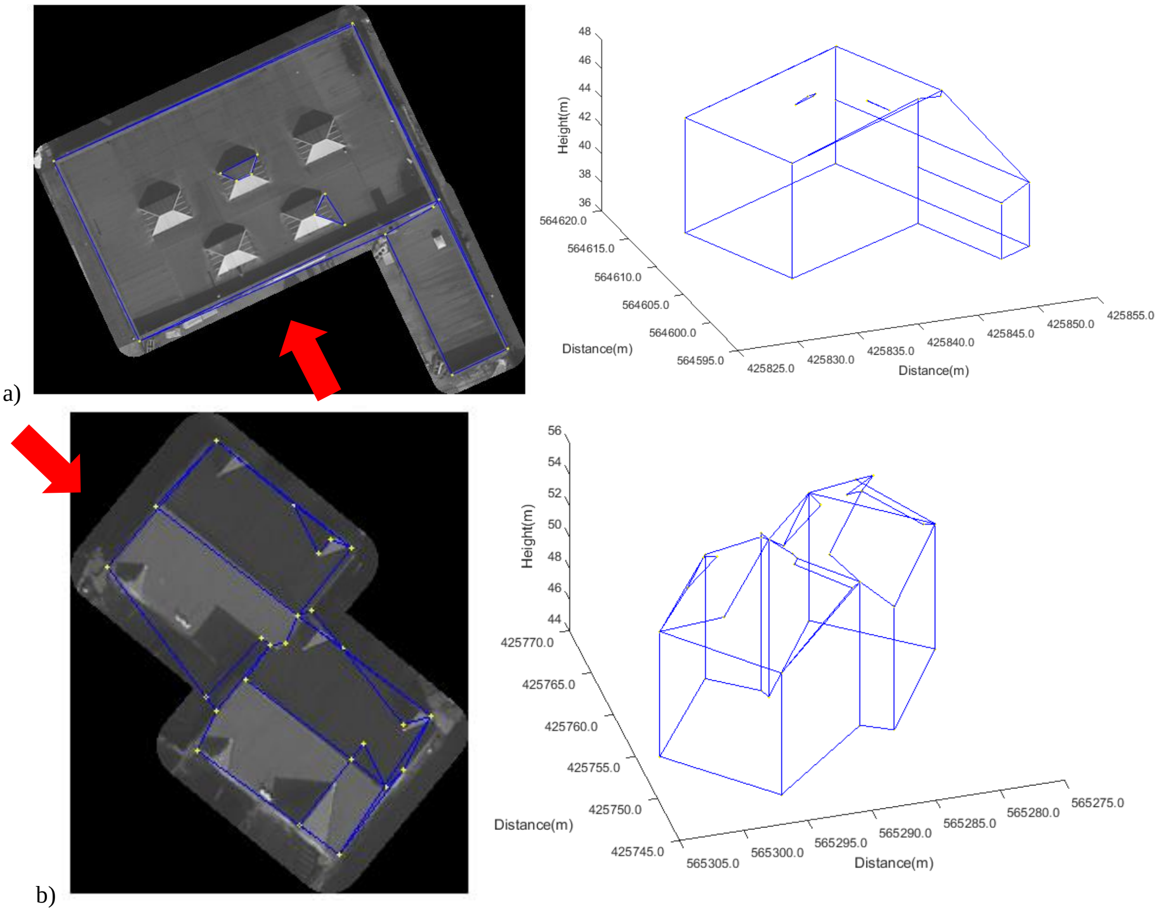


Figure 5.4 – Examples of LOD2 reconstruction where small roof geometry has affected the results: a) a multi-levelled flat roof where faces of the skylights were reconstructed and b) a cross-gable roof where incorrect edges were reconstructed from dormer windows.

5.2.3 LOD2 Reconstruction – Planar Extraction

The completeness, correctness and quality of the roof planes, reconstructed from the closed network of roof points and edges, are presented in Table 5.3. Flat roofs showed the greatest success for planar reconstruction, as most flat roofs only consisted of one roof plane, which can be reconstructed from the segmentation of the nDSM boundary. However, some flat roofs included in the roof structure were multi-levelled flat roofs, an example of which is shown in Figure 5.4a, where the two roof surfaces were not reconstructed at different heights for LOD1 models.

Overall the quality of the reconstructed roof planes for flat roofs was 80%. Although the example in Figure 5.3 shows too many corners being extracted, the roof plane was successfully reconstructed to a quality of 86%. 100% was not achieved because of the incorrectly constructed planes from edges extracted from vegetation, which overlapped the roof surface. The example of the multi-levelled flat roof in Figure 5.4a produced a quality percentage of 88%, which was hindered by the extraction of the skylights.

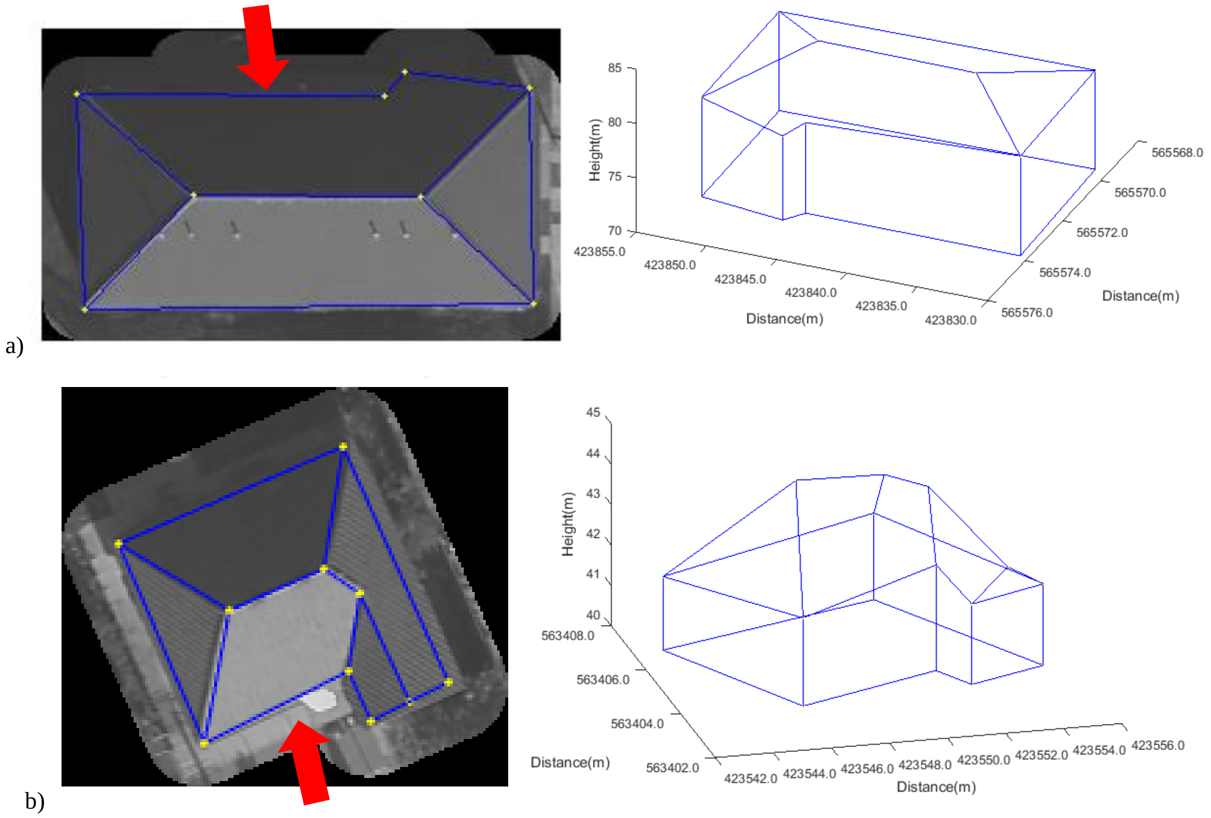


Figure 5.5 – Examples of 3D reconstruction: a) a hipped roof and b) a cross-gable roof.

Further examples of correct planar segmentation can be seen in Figure 5.5 for different roof types. The four main planes of the hipped roof in Figure 5.5a have been reconstructed, but errors in the extraction of boundary corners resulted in too few corners being extracted: only one eave line of the two small extensions were reconstructed and the flat roof surfaces of these extensions were not reconstructed. This resulted in an overall planar quality of 58%. The correct number of roof planes was reconstructed for the cross-gable roof in Figure 5.5b but, as previously discussed, errors in the extraction of the corner heights resulted in irregular geometry.

Whilst the reconstruction of flat roof planes was highly successful, the reconstruction of roof geometry with two or more planes was unsuccessful. The results in Table 5.3 show that the overall quality for four out of the five roof structures was below 50%, which appears marginally better when only considering roof planes larger than 10 m^2. Although the extraction of corners was generally quite high, with an average of 78% of correct corners extracted, a number of roof-lines were not reconstructed. Two examples can be seen in Figure 5.6 for different roof types where roof-lines were missing. Because of the missing ridgeline and step edge in Figure 5.6a, the gable roof was reconstructed as a shed roof, which is a building with one flat sloping roof plane. The hipped roof in Figure 5.6b extracted 100% of corners, with no false extraction, but because two hip-lines were not reconstructed, this resulted in a planar quality of only 25%. The low reconstruction of roof planes was also due to the incorrect connection of corners, with roof-lines being reconstructed across roof planes, an example of which is shown in Figure 5.4b.

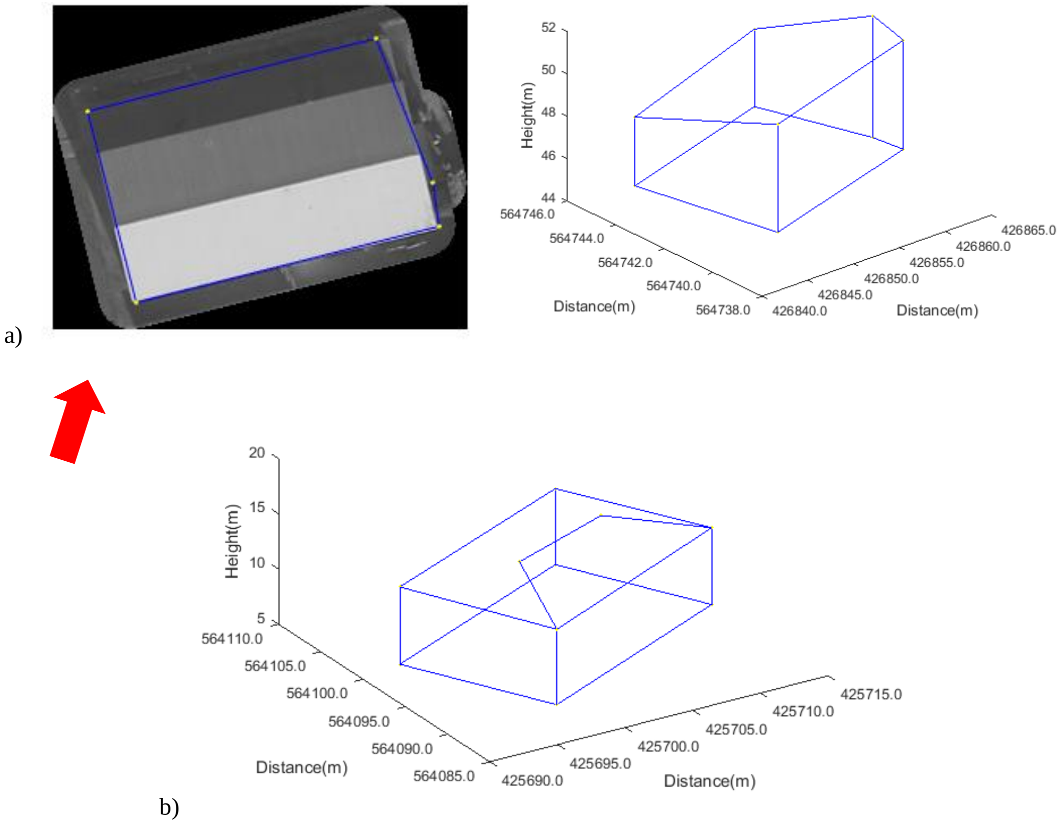


Figure 5.6 – Examples of unsuccessful planar extraction: a) a gable roof with no planes reconstructed; and b) a hipped roof with only one plane reconstructed.

The results of planar extraction in Table 5.3 suggest that as the complexity of the building increases, the quality of the roof plane extraction also increases. This was likely to be due to the size of the building and the number of roof planes to reconstruct. For a gable roof, if one roof-line was missing, this resulted in either only one plane being reconstructed or, as the example in Figure 5.6a shows, no roof planes being reconstructed. Complex roof structures tended to have more roof planes than simple roofs, which were generally larger in size. As will be discussed in Section 5.4.2, the parameters chosen for the connectivity of roof-lines may suit larger industrial buildings compared to smaller residential buildings.

5.3 Reconstruction by Selected Scene

For reconstruction based on area, three types of scenes were selected from the imagery. These were residential (three areas), industrial (two) and city centre (two) scenes. The buildings found in each scene were very different from each other. The buildings in residential areas tended to have repetitive roof structures for separate buildings but were different for each residential scene. Industrial scenes tended to be dominated by large buildings with either a flat roof or a simple roof structure. City centre scenes had large buildings, generally with flat roofs, but these were also on multiple levels and with varying degrees of complexity. This therefore gave a wide range of roof types with which to test the methodology. All the buildings for each test areas and the reconstructed models can be seen in Appendix E.

5.3.1 Residential

LOD1 Reconstruction	RMSE (m)			Roof corners extracted			
	X	Y	Z	Correct corners extracted (%)	Total corners extracted (%)	Number of corners extracted	Total number of corners extracted
Residential 1	0.63	0.51	0.77	75.5	82.1	106	87
Residential 2	0.62	0.82	1.31	64.6	83.9	168	141
Residential 3	0.55	1.00	0.51	79.1	106.4	110	117

Table 5.4 – Accuracies of the extracted corners and the number of corners extracted for LOD1 reconstruction of residential scenes.

LOD2 Reconstruction	RMSE (m)			Roof corners extracted			
	X	Y	Z	Correct corners extracted (%)	Total corners extracted (%)	Number of corners extracted	Total number of corners extracted
Residential 1	0.47	0.65	0.61	79.5	145.7	232	338
Residential 2	0.56	0.73	1.13	64.1	100.4	252	253
Residential 3	0.71	0.51	0.64	74.2	177.1	227	402

Table 5.5 – Accuracies of the extracted corners and the number of corners extracted for LOD2 reconstruction of residential scenes.

Planar Extraction	Per-roof plane (%)			Per-roof plane (10m^2) (%)		
	Completeness	Correctness	Quality	Completeness$_{10}$	Correctness$_{10}$	Quality$_{10}$
Residential 1	40.0	55.3	36.3	43.3	66.0	40.2
Residential 2	20.5	26.1	18.8	20.8	26.3	19.3
Residential 3	44.3	59.6	41.1	44.3	55.8	40.9

Table 5.6 – Planar roof extraction analysis of residential scenes.

Three residential scenes, with various building sizes and roof structures were tested for 3D reconstruction. The results of the accuracies of corners extracted for all residential sites can be seen in Table 5.4 for LOD1 reconstruction and Table 5.5 for LOD2 reconstruction. The analysis of the extracted roof planes is shown in Table 5.6.

Residential 1 covered a 9,000 m² area consisting of 15 buildings. These buildings mainly consisted of hipped and gable roof models. Residential 2 was an 8,600 m² area and consisted of 11 complex shaped buildings, with neighbouring and overhanging vegetation, which made the area unfavourable for automatic reconstruction. The buildings in the Residential 3 were similar to those in Residential 1, consisting of gabled and hipped roofs, but also with many small flat roofed outhouses included. The area of Residential 3 was 8,000 m² and consisted of 32 buildings.

5.3.1.1 LOD1 Reconstruction – Corner Extraction

Compared to the extraction of boundary corners when manually selecting buildings, the position accuracy was slightly worse for the three residential areas, with an average RMSE of ±0.57 m. The height accuracy was also worse with an average RMSE of ±0.86 m. The number of correct corners extracted was similar to the manually selected buildings with approximately 73% of corners successfully reconstructed; however, two of the sites showed that too few corners were extracted and reconstructed. This was due to a number of short edges of buildings in Residential 1, where dormer windows from the roof surface slightly extended from a rectangular footprint, and Residential 2, where the building footprints had irregular geometry consisting of short extensions and some buildings were covered with over-hanging vegetation, as shown in Figure 5.7.

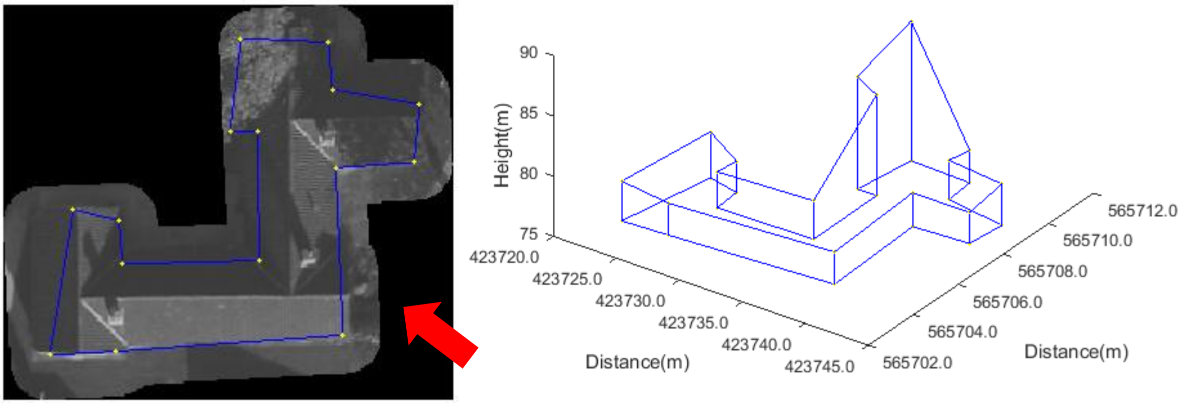


Figure 5.7 – Example of LOD1 reconstruction from Residential 2.

Due to the presence of vegetation and failures of the ground classification, the nDSM boundaries were under-segmented during run graph vectorisation, with LOD1 models reconstructed with 83% of corners extracted in total, and only 72% of them in the correct location for Residential 1 and Residential 2, which shows approximately an extra 17% of corners were extracted in the incorrect position. However, not enough corners were extracted for correct footprint reconstruction.

Residential 3 extracted an extra 6% of corners, which was caused by over-segmentation of the nDSM boundary. This was due to buildings being in close proximity to each other, with sections of a neighbouring building being extracted within the 2 m buffered footprint of a building. Several buildings were also affected by points being extracted from vegetation, as shown in Appendix E.

One small flat roofed building, with a footprint of 2 x 5 m was also in the test site and was reconstructed, with the results shown in Figure 5.8. Although the height of one boundary corner appears lower than the other three corners, the height RMSE of all corners was ±0.36 m, with a positional RMSE of ±0.34 m. This example highlights that even small buildings, which are much smaller than the search windows used for roof-line connectivity, can be reconstructed using the defined search windows and parameters.

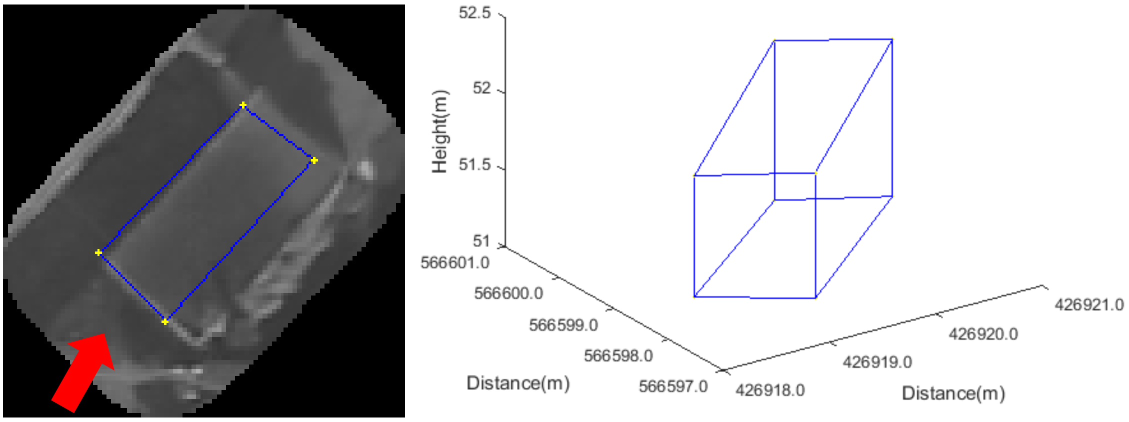


Figure 5.8 – Example of reconstructed flat roof, which has a small building footprint.

5.3.1.2 LOD2 Reconstruction – Corner Extraction

The results for LOD2 reconstruction were similar to those of LOD1 reconstruction: better positional than height accuracy and, on average, 72% of corners were correctly reconstructed. However, all residential sites extracted too many points and constructed too many roof-lines.

The results of reconstruction for Residential 1 and Residential 3 are comparable with similar positional and height accuracies, as well as a similar percentage of correct roof corners extracted, as seen in Table 5.5. Both sites extracted too many points as corners: Residential 1 extracted an extra 37% of points, and Residential 3 extracted an extra 55% of points. This was due to small substructures on the main roof structure, such as dormer windows and skylights, as well as the small outhouses having overhanging and neighbouring vegetation included in the reconstruction.

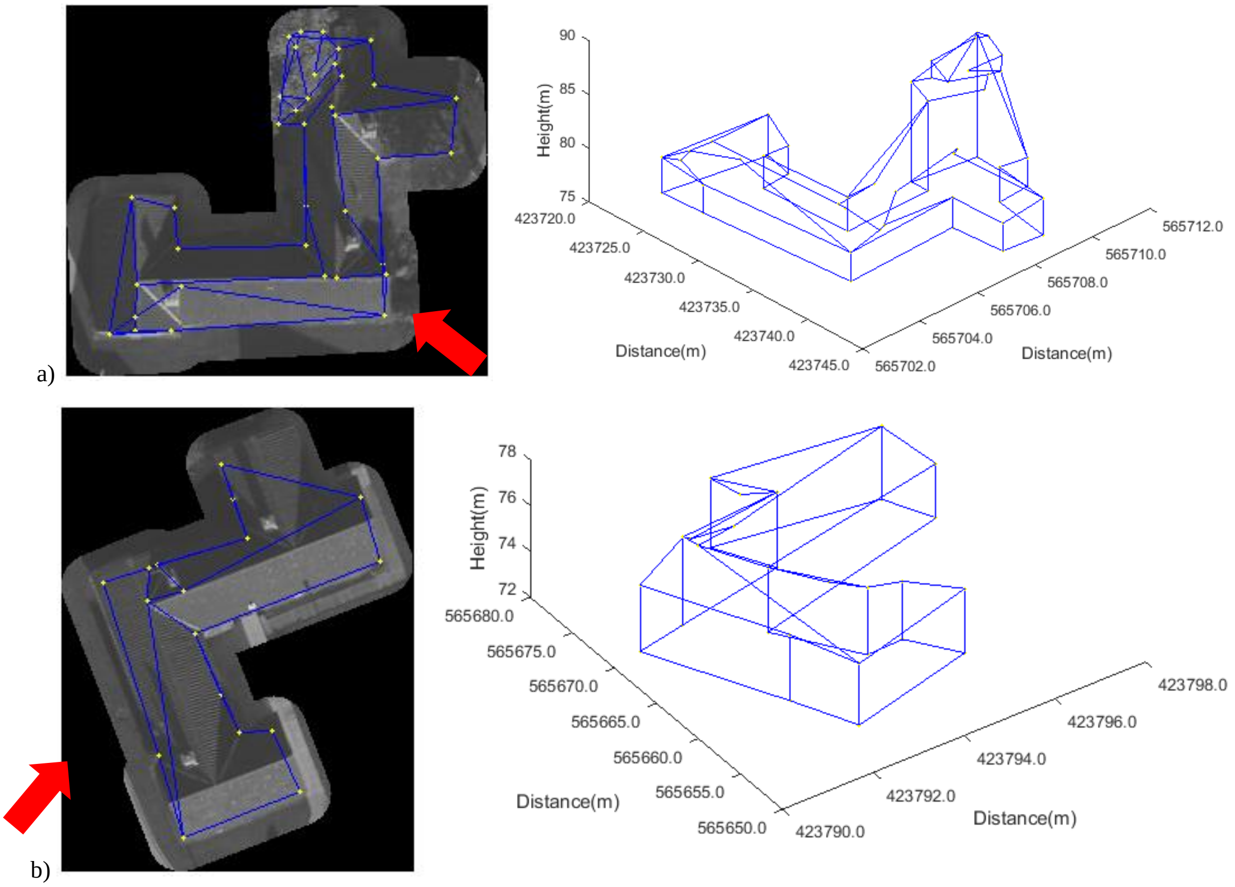


Figure 5.9 – Examples of complex roof structure reconstructed from Residential 2.

The total number of points reconstructed for Residential 2 almost represented the correct number of roof corners, with only an extra 0.4% of points constructed. However, the results in Table 5.5 show that only 64% of roof corners were correctly extracted. Some buildings in Residential 2 extracted too many points, with false edges reconstructed from vegetation, as shown in Figure 5.9a. Most roof structures extracted too few corners, due to roof lines not being extracted during edge detection and no connecting corners being reconstructed. An example of this can be seen in Figure 5.9b where most roof-lines were not reconstructed: the two reconstructed ridgelines were connected to incorrect corners; only two out of six valley lines were reconstructed and several corners of the boundary were not extracted.

5.3.1.3 LOD2 Reconstruction – Planar Extraction

Due to the issues with corner extraction explained above, the percentage of planar extraction was also very low. Residential 2 produced one of the worst results for planar extraction of all test sites, with less than 20% of planes extracted for all planes and planes larger than 10 m². This was due to previously discussed errors in the nDSM boundary segmentation, errors in the extraction of roof-lines and the connection of false edges, examples of which are shown in Figure 5.9.

The planar extraction results for Residential 1 and Residential 3 were better, with quality percentages of 36.3% and 41.1%, respectively, but still unsatisfactory. The buildings in these two sites were generally gable and hipped roofs, with two examples shown in Figure 5.10. As can be seen, the roof planes in these examples were reconstructed with quality percentages of 64% and 61% respectively, but this was hindered by a number of false edges being reconstructed. Edges were extracted from two chimneys in Figure 5.10a and boundary corners were not reconstructed in the correct location, which reduced the size of the planes. The results in Figure 5.10b show edges extracted from a chimney at the centre of the ridgeline and from shadow where hip-lines were duplicated. The example in Figure 5.10b also shows how incorrect ground classification, where two buildings were in close proximity, led to over-segmentation of the nDSM boundary for corner extraction. A point has been extracted from a neighbouring feature, which was not part of the building being reconstructed.

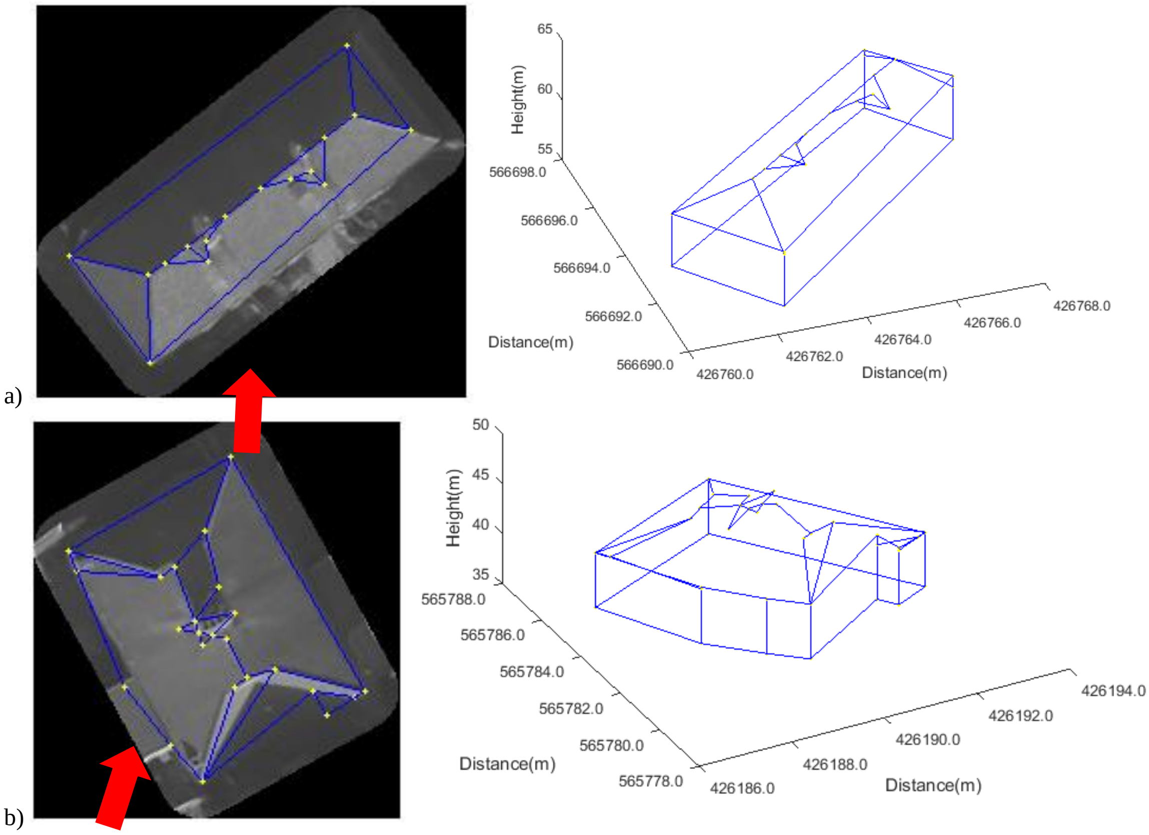


Figure 5.10 – Examples of LOD2 models: a) a hipped roof from Residential 1; and b) a hipped roof in Residential 3.

5.3.2 Industrial

LOD1 Reconstruction	**RMSE (m)**			**Roof corners extracted**			
	X	**Y**	**Z**	**Correct corners extracted (%)**	**Total corners extracted (%)**	**Number of corners extracted**	**Total number of corners extracted**
Industrial 1	0.68	0.67	2.73	88.6	124.2	157	195
Industrial 2	0.66	0.62	1.13	84.0	136	125	170

Table 5.7 – Accuracies of the extracted corners and the number of corners extracted for LOD1 reconstruction of industrial scenes.

LOD2 Reconstruction	**RMSE (m)**			**Roof corners extracted**			
	X	**Y**	**Z**	**Correct corners extracted (%)**	**Total corners extracted (%)**	**Number of corners extracted**	**Total number of corners extracted**
Industrial 1	0.77	0.67	2.67	78.6	221.6	282	625
Industrial 2	0.50	0.61	0.76	75.1	153.2	265	406

Table 5.8 – Accuracies of the extracted corners and the number of corners extracted for LOD2 reconstruction of industrial scenes.

Planar Extraction	**Per-roof plane (%)**			**Per-roof plane (10m^2) (%)**		
	Completeness	**Correctness**	**Quality**	**Completeness$_{10}$**	**Correctness$_{10}$**	**Quality$_{10}$**
Industrial	57.8	63.8	52.9	57.6	64.4	53.0
Industrial 2	53.7	58.9	50.0	56.4	61.8	52.4

Table 5.9 – Planar roof extraction analysis of industrial scenes.

The reconstruction results of the two industrial scenes can be seen in Table 5.7 to Table 5.9. Compared to the residential test sites, Industrial 1 had a much larger area, with an area of 93,500 m² and consisted of 20 buildings. This area was much larger than the residential areas, due to the increased size of the buildings. Industrial 2 had an area of 3,900 m² and consisted of 16 extracted buildings.

5.3.2.1 LOD1 Reconstruction – Corner Extraction

Compared to the residential site and the manually selected buildings, the position accuracy of LOD1 models was slightly worse with an average RMSE of ±0.65 m. An example of a LOD1 reconstructed model can be seen in Figure 5.11, which shows 100% of corners were extracted with no false positives. Although one corner extracted appears incorrect, this was still within the positional tolerance and an overall positional accuracy of ±0.48 m was achieved for this building. A sloping edge can also be seen for two of the eave lines due to the L-shaped extension being at a slightly lower height. Heights of the corners were correctly extracted with an accuracy of ±0.31 m for this example.

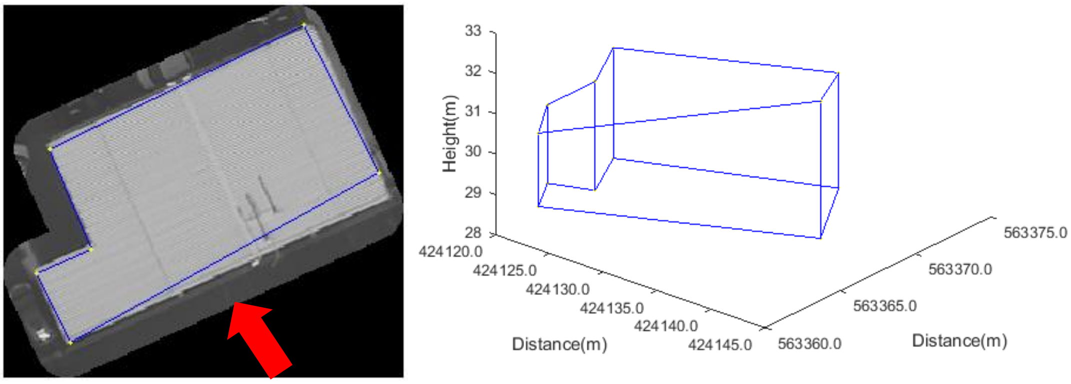


Figure 5.11 – Example of LOD1 reconstruction from Industrial 2.

Overall, LOD1 models generally suffered from poor height extraction for both test sites: with height accuracies of ±2.73 m for Industrial 1 and ±1.13 m for Industrial 2. Similar to the residential areas, this was influenced by overhanging vegetation and the close proximity of buildings. The percentage of corners correctly identified was, on average, 86%. Overhanging vegetation also caused over-segmentation of the nDSM boundary for LOD1 corner extraction, with an extra 22% of points extracted for Industrial 1 and 31% of points extracted for Industrial 2.

5.3.2.2 LOD2 Reconstruction – Corner Extraction

The two industrial scenes had a positional accuracy, on average, of ±0.64 m for LOD2 models with 75% of corners per roof extracted. Whilst there was close agreement of the positional accuracy of the points, a clear difference in the heights accuracies of corners, between the two sites, can be seen. Roof corners extracted from Industrial 1 had a height accuracy RMSE of ±2.67 m, which was outside the height accuracy specified by the LOD2 standards (Gröger and Plümer, 2012). Several corners were extracted within the positional accuracy of the reference corner, but due to overhanging vegetation and incorrect height extraction, the height of the corners was incorrectly assigned for several buildings. The height accuracy of extracted corners was much closer to the truth for the second industrial test site, with an RMSE of ±0.76 m.

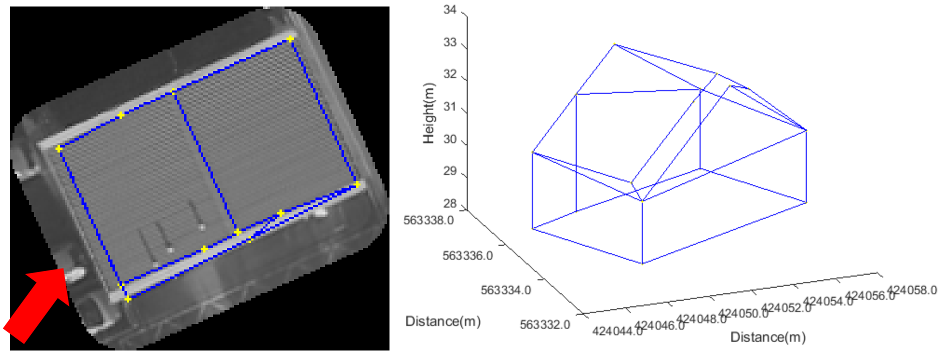


Figure 5.12 – Example of gable roof reconstruction from Industrial 2.

The example of a gable roof in Figure 5.12 shows the six corners of the roof: four corners of the roof boundary and the two ridgeline endpoints, were successfully extracted to a positional and height accuracy of ±0.44 m and ±0.19 m, respectively. However, extra points were extracted from a duplicated edge and the nDSM boundary was over-segmented during vectorisation, which resulted in an extra boundary corner being reconstructed, as seen by the extra vertical edge in Figure 5.12.

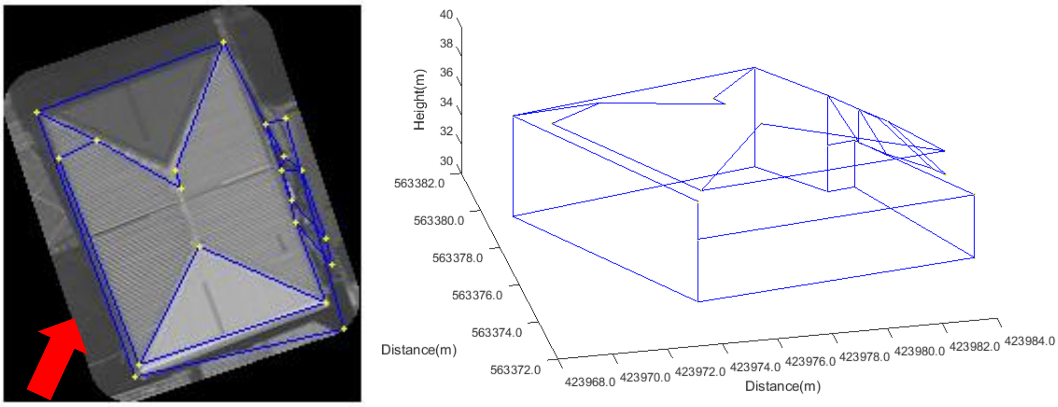


Figure 5.13 – Example of hipped roof reconstruction from Industrial 2.

Another example of LOD2 reconstruction can be seen in Figure 5.13 for a hipped roof, where 100% of roof corners were extracted to a positional and height accuracy of ±0.14 m and ±0.48 m, respectively. However, many extra points were also extracted. A point was extracted from a neighbouring building for the roof boundary, which resulted in the incorrect reconstruction of a step edge. The correct corners along the eave line of the roof were extracted from the reconstructed roof-lines and resulted in the improvement of the positional accuracy of the corners from LOD1 reconstruction to LOD2 reconstruction. This can be seen from a hip-line not connecting to a boundary corner of the LOD1 model in Figure 5.13, with the boundary corner extracted from a neighbouring building. Several other incorrect edges were also extracted from the close proximity of buildings, which in this example resulted in an extra 267% of points being constructed. Overall both industrial sites extracted too many corners: Industrial 1 extracted an extra 122% of points and Industrial 2 extracted an extra 53%. This shows on average that for Industrial 1 over twice the number of corners were extracted per building. However, this value was alleviated by one building which extracted many false edges from roof texture, with an extra 843% of corners constructed. Errors were also present in the OS MM building footprints of both sites, which extracted sections that were not buildings and reconstructed points from these areas.

5.3.2.3- LOD2 Reconstruction – Planar Extraction

Compared to the residential areas, planar extraction was achieved with a quality percentage of approximately 50% for both test sites. Buildings in both test sites were dominated by large, clearly defined roof structures. Many of these buildings also had simple roof structures such as flat, gable, a combination of gable roofs, or hipped.

The example in Figure 5.12 shows the correct extraction of the two main roof planes, but other incorrect planes were reconstructed due to extra points being reconstructed as corners. Although all the correct corners were extracted for the example in Figure 5.13, only two out of the four roof planes were reconstructed due to the central ridgeline not being extracted.

When compared to the residential scenes, planar extraction was more successful for industrial buildings. As previously discussed, this may be due to the size of the buildings being larger in the industrial scene with longer edges and larger roof planes. As suggested with the manually selected roofs, the parameters chosen may favour roofs with larger roof planes and longer edges.

5.3.3 City Centre

LOD1 Reconstruction	RMSE (m)			Roof corners extracted			
	X	Y	Z	Correct corners extracted (%)	Total corners extracted (%)	Number of corners extracted	Total number of corners extracted
City Centre	0.53	0.62	3.25	47.2	123.2	138	170
City Centre 2	0.60	0.45	3.10	38.2	91.0	89	81

Table 5.10 – Accuracies of the extracted corners and the number of corners extracted for LOD1 reconstruction of city centre scenes.

LOD2 Reconstruction	RMSE (m)			Roof corners extracted			
	X	Y	Z	Correct corners extracted (%)	Total corners extracted (%)	Number of corners extracted	Total number of corners extracted
City Centre 1	0.51	0.62	2.74	49.7	87.7	767	673
City Centre 2	0.49	0.47	3.92	34.8	111.8	397	444

Table 5.11 – Accuracies of the extracted corners and the number of corners extracted for LOD2 reconstruction of city centre scenes.

Planar Extraction	Per-roof plane (%)			Per-roof plane (10m^2) (%)		
	Completeness	Correctness	Quality	Completeness$_{10}$	Correctness$_{10}$	Quality$_{10}$
City Centre 1	37.4	45.6	33.4	43.8	50.3	38.5
City Centre 2	16.7	15.3	11.5	18.3	17.5	12.8

Table 5.12 – Planar roof extraction analysis of city centre scenes.

The results of 3D reconstruction for the two city centre scenes are presented in Table 5.10 through to Table 5.12. The size of the city centre scenes covered wide areas, 56,500 m² and 42,000 m², due to the buildings in the city centre consisting of mainly large buildings with complex roof structures and over multiple levels. City Centre 1 consisted of 10 buildings, and City Centre 2 consisted of six building.

5.3.3.1 LOD1 Reconstruction – Corner Extraction

As found with the other test sites the positional accuracy, of approximately ±0.5 m, was higher than the height accuracy, of ±3.17 m. The reason for the large height inaccuracy was due to the buildings having complex boundaries with large height discontinuities. Due to the complexity of the building shapes, a low number of boundary corners were also correctly identified, with both scenes extracting less than 50% of corners. City Centre 1 extracted approximately an extra 23% of points as corners, which may have been the result of over-segmentation of the nDSM boundary during vectorisation. The reconstructed models in this area suffered from poor extraction of boundary corners with 9% too few corners extracted on average per building. This was due to the complexity of the buildings, but may also have been caused by the refinement of the eave lines using orientation.

It is noticeable from the reconstruction results of City Centre 2 in Appendix E that only five out of the six buildings were reconstructed. The sixth building was not reconstructed due to the MALTAB script being unable to handle the complexity of the building. Reconstruction, in this case, may have favoured the building being segmented into smaller primitives, which were then combined together to form the final LOD1 and LOD2 models.

5.3.3.2 LOD2 Reconstruction – Corner Extraction

The LOD2 models reconstructed over the two scenes produced an average positional accuracy of ±0.52 m from the reference data. However, the height accuracy was much worse with an average RMSE across the two scenes of ±3.33 m.

An example of the 3D reconstruction model can be seen in Figure 5.14 for a multi-levelled flat roof. The two levels of the roof were successfully reconstructed. However, the façade between the upper and lower roof levels was not reconstructed due to step edges currently only being reconstructed for boundary corners. This is a current limitation of the method and should be addressed in future work. Because the façade was not reconstructed, due to the corners at the bottom of the roof surface not being extracted, this meant that only 67% of corners were correctly reconstructed, but to a positional accuracy of ±0.43 m and height accuracy of ±0.36 m.

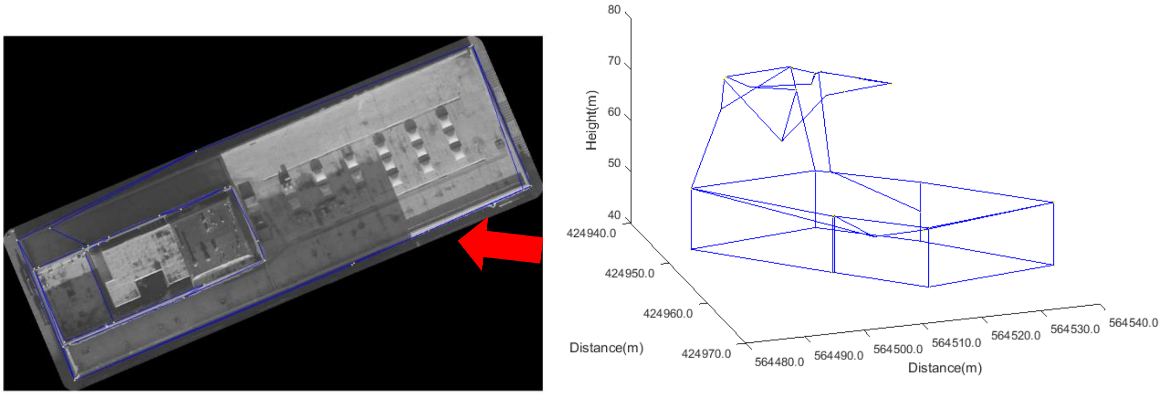


Figure 5.14 – Example of multi-levelled roof reconstruction in City Centre 1.

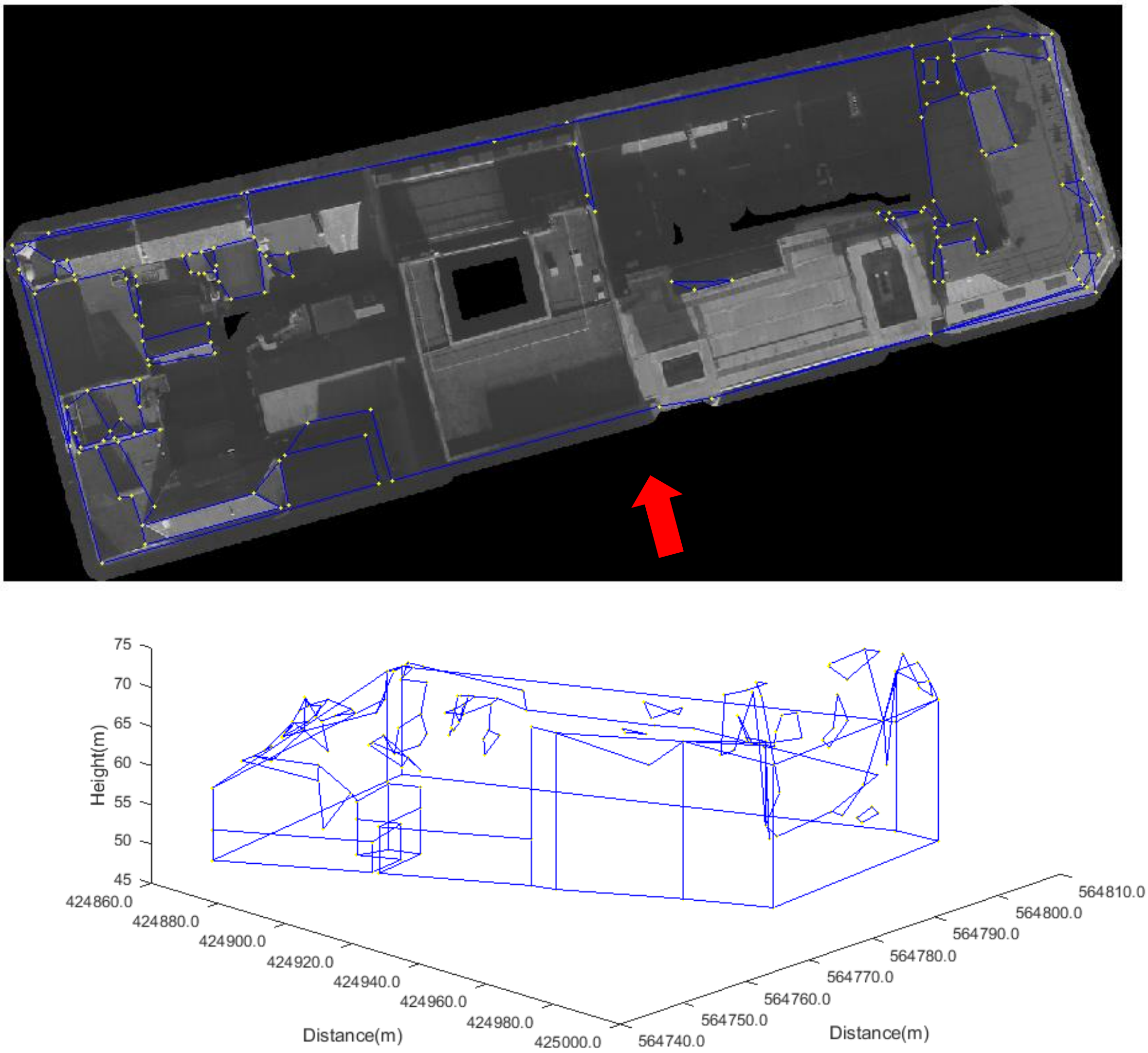


Figure 5.15 – Example of 3D reconstruction in City Centre 1.

Similar to LOD1 models, less than 50% of corners were correctly extracted for LOD2 reconstruction, which again was due to the complexity of the roof structures. An example of this complexity is shown in Figure 5.15, where several different roof structures can be seen for one building footprint and at various heights. For the example in Figure 5.15, 31% of corners were correctly reconstructed, but as can be seen in the figure, too few corners were extracted in total. The large height difference between various roof surfaces resulted in some corners being assigned an incorrect height value, when compared against the truth, which can be seen for several corners in Figure 5.15.

The total number of points extracted for LOD2 reconstruction in City Centre 2 was just over 100%, which suggests for each building the correct number of roof corners were extracted. However, only 35% were extracted in the correct location. The buildings in City Centre 2 actually suffered from too few roof corners being extracted with approximately 50 corners not being extracted for LOD2 models. For City Centre 1, too few corners were also extracted, which was again due to the complexity of the buildings and the low image contrast, as shown in Figure 5.15, which meant edges could not be extracted.

5.3.3.3- LOD2 Reconstruction – Planar Extraction

Due to the low extraction of correct corner positions, planar extraction was largely unsuccessful. The extraction of planes in City Centre 1 was slightly more successful than City Centre 2. The quality of planes extracted in City Centre 1 was 33.4%, and an example of the 3D reconstructed model can be seen in Figure 5.14 for a multi-levelled flat roof. The two levels of the roof were successfully reconstructed, with an extracted planar quality of 76%. The façade between the two roof levels was not reconstructed due to reasons previously explained.

However, for most cases, no roof geometry was reconstructed and the results produced a shell of the building boundary. This can be seen from the example in Figure 5.15, where only 9% of roof planes were reconstructed. These buildings had very complex roof structures over different height levels. As previously discussed, the low image contrast may have hindered edge detection and the complexity of the roof may have resulted in no connections being made between roof-lines. Planar extraction from City Centre 2 performed the worst of all test sites, with a quality of 11%. This was due to only 35% of corners being extracted in the correct location and the extraction of incorrect corners, which resulted in the incorrect connection of roof-lines.

5.4 Discussion

For all sites and buildings tested, the models showed good positional accuracy for the extracted corner points and that the majority of corners were extracted. The average positional accuracy of extracted roof corners for all LOD1 and LOD2 models reconstructed from the Newcastle upon Tyne study area was ±0.55 m, which, when compared to the accuracies of CityGML, is almost consistent with the positional accuracy of LOD3 models (Gröger and Plümer, 2012). However, the height of the roof corners was not so successful, with an average accuracy for all reconstructed buildings of ±1.34 m. This height accuracy was still within the specification of LOD2 (Gröger and Plümer, 2012). This shows that the developed methodology can successfully reconstruct corners of 3D building models to the specification outlined by CityGML LOD2 from extracted edges. However, as has been shown throughout the results, only 71% of corners were successfully extracted, on average, for both models, which resulted in unsuccessful extraction of roof planes, due to some corners and edges not being extracted, and will be explored further in this section.

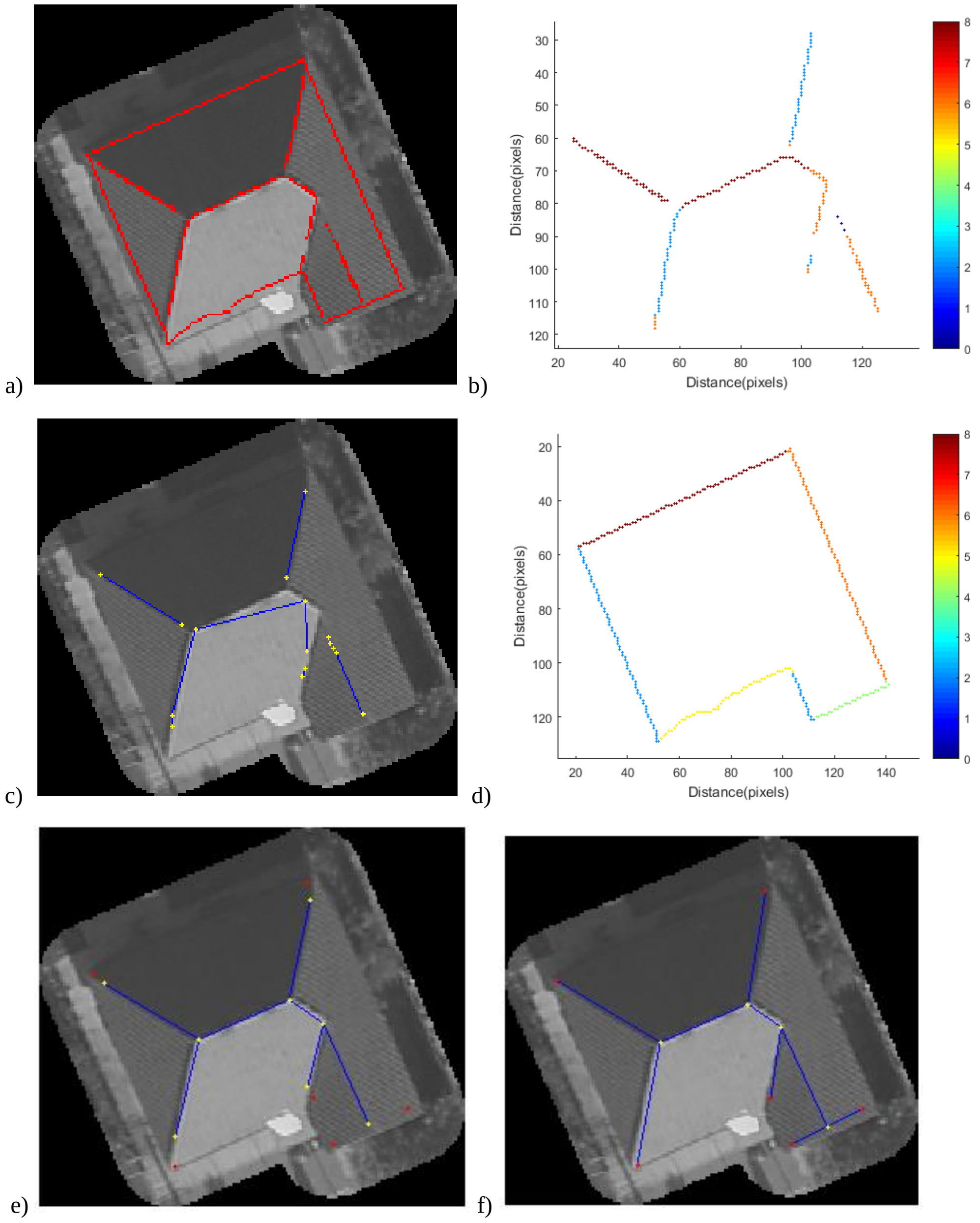


Figure 5.16 – The various stages of 3D reconstruction: a) refined Canny edges from scan line segmentation; b) roof-line segmentation using Freeman chain code; (c) extracted roof-lines from Figure 5.16b; d) nDSM segmentation; e) refined roof-lines using initial connectivity ruleset with segmented boundary corners (red); f) the connected roof-line and boundary corner network.

To reconstruct 3D building models, the concept of a closed network of roof-lines was enforced by removing any line which was not connected at both endpoints. The images in Figure 5.16

show the results at various stages of the methodology for the reconstruction of the cross-gabled roof in Figure 5.5b. The edges extracted from scan line segmentation in Figure 5.16a were classified using the Freeman chain code, to produce the segmentation results in Figure 5.16b for the ridgelines, and Figure 5.16d for the nDSM boundary. Some classification errors can be seen in Figure 5.16b, which were transferred to the vectorised lines in Figure 5.16c. Examples of over-segmentation, due to the first pixel for classification not being at the end of a roof-line, and under-segmentation, where the classification has failed to extract the correct corner position, can be seen from this example.

The first developed connectivity ruleset, described in Section 4.7.2.1, used various search windows from 0.5 m to 1.5 m, which was dependent on the length of the line with respect to the longest roof-line length, to correct segmentation errors. The angle was calculated for each roof-line endpoint to all edge pixels, found within the defined search window, to determine if any angle within a defined threshold of 180° was found. The roof-line endpoint was moved and connected to the edge pixel with the maximum angle, if the criterion was satisfied. The results of this refinement can be seen in Figure 5.16e, where valley lines have been extended to connect to the central ridgeline and have extracted the correct corner position. This resulted in the corners of the central ridgeline being correctly defined to resolve the under-segmentation of the edges and reconstruct a previously unclassified valley line. Short edges were also removed using a length threshold. The red corners in Figure 5.16e represent the boundary corners, which were the result from the line tracing classification of the nDSM boundary. The results in Figure 5.16d show the boundary of the roof was correctly segmented. Roof-lines were connected to the boundary corners using the second developed connectivity ruleset, discussed in Section 4.7.2.2 and Section 4.6.2.

Angles, lengths and line orientations were measured for unconnected endpoints to connect roof-lines to other roof-lines, as well as to boundary corners and edges. The results of the second connectivity ruleset can be seen in Figure 5.16f. Hip-lines have been extended to the boundary corners, whilst in one case a ridgeline has been extended to intersect the nDSM boundary segmentation and create the two gable lines and the gabled roof front, which can be seen in Figure 5.5b.

5.4.1 LOD1 Reconstruction

The average percent of correct roof corners extracted for LOD1 reconstruction was 77%, whilst the average percent of points extracted was 104%. This shows that not all roof corners were extracted, but overall, too many points were extracted and constructed as corners. As previously discussed the segmentation of the nDSM boundary struggled to classify corners from short extensions of the building boundary, an example of which is shown in Figure 5.1. This was due to the corners of the short extensions being modelled as a curve instead of an orthogonal corner, which was caused by errors in the ground classification. An example of the ground classification can be seen in Figure 5.17.

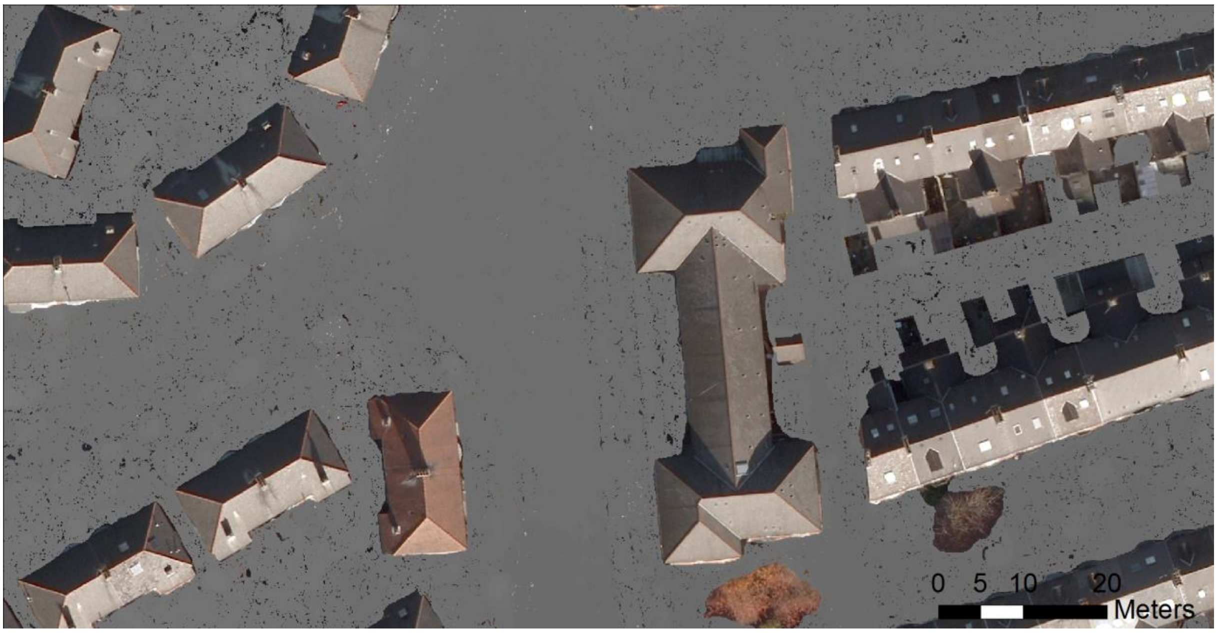


Figure 5.17 – Ground classification (grey) overlaid onto the true-orthophoto.

Several hipped roofed buildings can be seen in Figure 5.17, which have short extensions to the building. In most cases, the ground classification was not able to classify along the edges and in the corners of these short extensions. Parts of the ground can be seen from these examples in Figure 5.17. Although DSMs from developed image-matching procedures should yield sharp boundaries between objects, some roofs did not show sharp discontinuities between the roof edge and the ground. This was the case for the complex roof structure seen in Figure 5.17. A small outhouse was in close proximity to the complex roof structure, where the image matching procedure was unable to interpolate heights correctly between the two buildings. Ground pixels from the true-orthophoto can be seen between the small outhouse and complex roof structure, which meant the ground was not classified correctly in this area. This misclassification of the ground also produced smooth and curved transition at corners between the roof edges, which resulted in incorrect segmentation of the nDSM boundary, seen in Figure 5.18. Curved edges in the nDSM boundary caused under-segmentation and resulted in incorrect corner extraction.

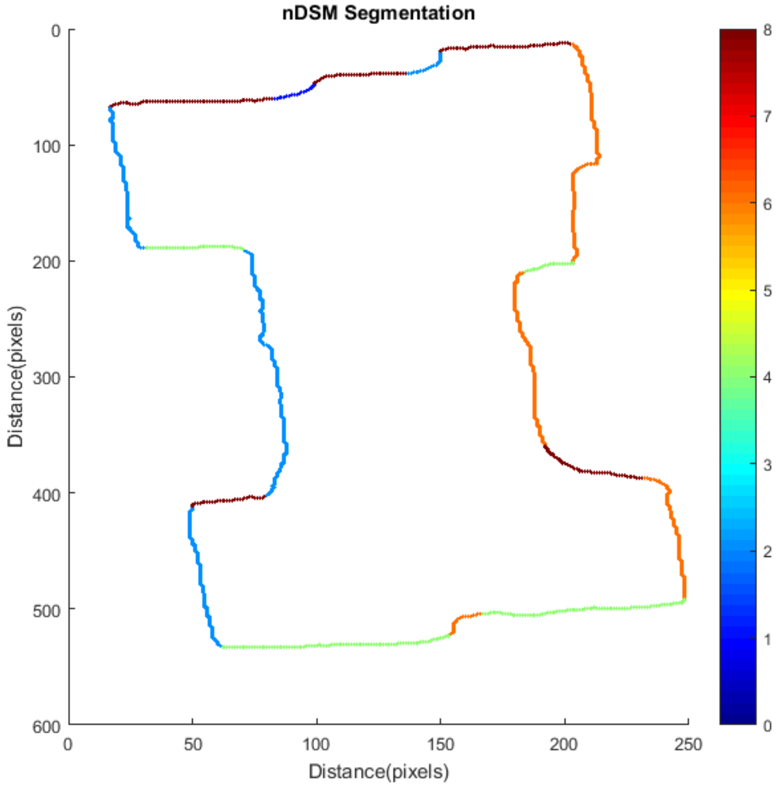


Figure 5.18 – The segmentation of the nDSM boundary for the complex roof structure in Figure 5.17.

Errors in the extraction of LOD1 corners can be seen to be due to the ground classification. The errors in the ground classification may have resulted from the point cloud being created from the conversion of the DSM to XYZ points. As previously explained, errors at the edges of some buildings in the DSM were reproduced in the conversion of the raster surface model to a point cloud. This conversion also only yielded a 2.5D point cloud. Dense image matching procedures have been shown, in Chapter 3, to produce 3D point clouds. Therefore, the procedure needs to be tested with a correct image-based point cloud from pixel to pixel matching, which may potentially correct for errors in the point cloud classification.

The errors in point cloud classification could also be due to the parameter setting of the ground classification procedure in TerraScan. The parameters used were suggested for ‘natural areas’ (Bentley, 2015). This may have made them unsuitable for some areas of the test site. Whilst the results did show highly successfully classification results, by further refining some of the parameters, better classification at the boundary of buildings might be achieved.

The extraction of extra points for LOD1 reconstruction can also be contributed to the refinement of points, using orientation, as well as neighbouring buildings being extracted in the buffered footprints. When refining the edges, the interior angle was not considered, and by using just the orientation of the longest edge to refine all other edges, meant that the angle at some corners was highly irregular. An example of this can be seen in Figure 5.9, where some boundary corners were incorrectly refined to produce an eave line across several roof planes. In some cases, parts of neighbouring buildings were also extracted in the buffered footprint of each building. If a neighbouring building was within 2 m of the building footprint, then sections of the building's roof structure were also extracted. This again resulted in errors of the nDSM boundary segmentation as edges were then extracted from the neighbouring building and connected to the roof-lines from the reconstructed building. An example of this can be seen in Figure 5.13.

5.4.2 LOD2 Reconstruction

The results of LOD2 reconstruction showed that, on average, only 70% of roof corners were correctly extracted, but too many points were constructed as corners, with an extra 30% of total roof corners also extracted. The extraction of too few roof corners can be attributed partly to the Canny edge detector and to the refinement of the edges. This can also be credited to the connectivity ruleset, but this mainly caused the reconstruction of too many corners.

5.4.2.1 Extraction of Roof Geometry

Overall, the results of Canny edge detection and the removal of false positives via scan line segmentation was successful, as was shown in Chapter 4, with most roof edges being extracted. The parameters determined by sensitivity thresholding for the Canny edge detection were particular low and meant that too many edges were extracted and required further refinement. The threshold sensitivity of the Canny edge detector showed that if a higher threshold was chosen, than some roof edges were not extracted from edge detection. There was, thus, a trade-off between extracting false positives and correct edges. The removal of false positives has been shown to be highly successful using scan line segmentation. However, some incorrect edges were still present and not removed.

Although the parameters of the edge detector were set low to extract as many roof edges as possible, some edges were not extracted by the Canny edge detector due to poor image contrast, which produced a weak magnitude gradient value along the roof-lines. As previously discussed, this mainly occurred at the boundary of a roof where shadow was cast. To overcome this, the boundary of the nDSM was included for the refinement of the edge detection to produce boundary edges of the roof. However, the edges of some roof planes were also cast in shadow, which meant they were not detected. The example in Figure 5.19 shows several roof-lines covered by shadow, where an edge could not be detected by the Canny edge detector and was therefore not reconstructed.

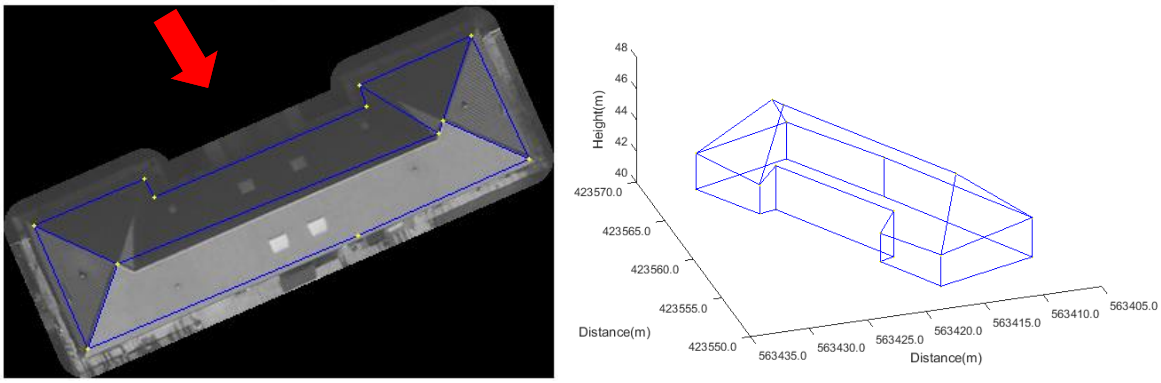


Figure 5.19 – Example of LOD2 reconstructed model of a complex roof.

A number of edges extracted by the Canny edge detector were incorrectly refined using the developed scan line segmentation procedure. An example can be seen in Figure 5.20. The results of Canny edge detection, in Figure 5.20a, shows the extraction of the major roof lines, but also a vast number of false positives that were extracted from roof texture on the two triangular planes of the hipped roof.

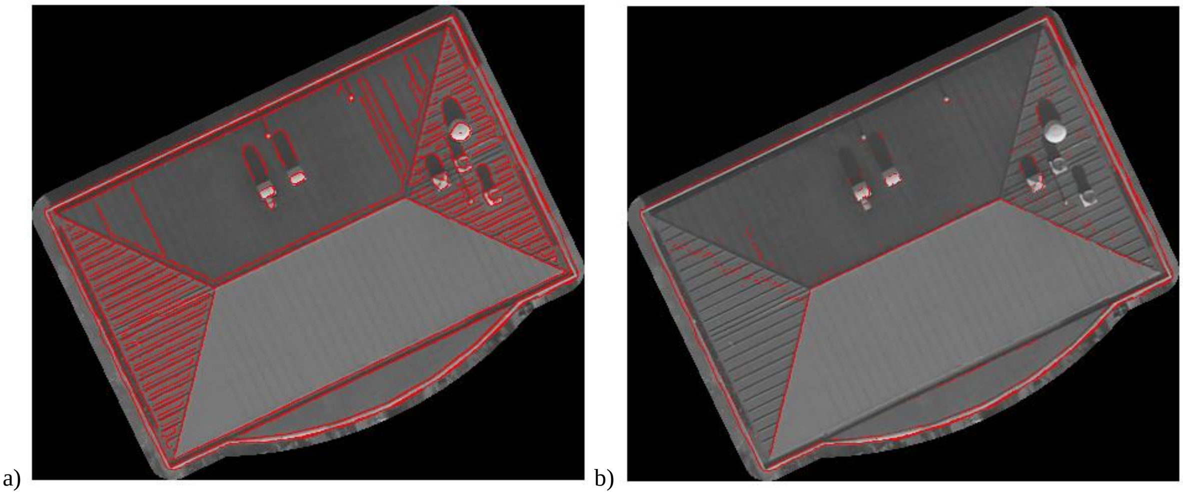


Figure 5.20 – Example of roof geometry extraction for a hipped roof: a) the Canny edge detection; b) edge detection refinement using scan line segmentation.

The results of scan line segmentation, in Figure 5.20b, show how the false positives on the triangular roof planes were removed. Most of the edges detected from roof texture, shadow, chimneys and other roof artefacts were removed using the developed scan line segmentation methodology. However, edges along two hip-lines of the triangular roof planes were also removed. This resulted in a number of fragmented edges, which were extracted with a slight offset from the true edge. This was due to the vast number of false positives extracted from the Canny edge detector on the triangular roof planes. This meant the edge pixels had similar height values, and thus the edges detected on the hip-lines were determined to be below the residual threshold and removed as false positives. The short-fragmented edges, which were extracted at a slight offset to the hip-lines, were removed with the length and orientation thresholds of the roof before the connectivity ruleset. The removal of pixels along the hip-lines, during scan line segmentation, and refinement of the vectorised lines meant no edges could be connected, and resulted in only one roof plane being reconstructed, as shown in Figure 5.6b.

Edges were also removed from the example in Figure 5.20 between the boundary of the hipped roof plane and the small curved flat extension, which was again due to height differences in the pixels being below the threshold. The curved section of the roof was over-generalised in the extraction of boundary corners, which was not able to handle curved edges. This short flat extension was not reconstructed in the results shown in Figure 5.6b. During scan line segmentation, most of the edge pixels were removed at the narrowest part of the curved section. At the widest part of the flat section, edges were kept as breakpoints, where the residual of the fitted line was above the threshold. This again highlights how the scan line segmentation approach struggled to retain pixels on the correct roof-line, when edges were in close proximity.

5.4.2.2 Connectivity Ruleset

Although roof corners were extracted to a high degree of success, it can be concluded that the extraction of roof planes, for any roof shape other than a flat roof, was unsuccessful. The average quality of roof plane reconstruction for all LOD2 models reconstructed from the Newcastle upon Tyne dataset was only 39%. This only slightly improved when planes larger than 10 m² were considered. This shows that regardless of size most roof planes were not reconstructed. This was partly due to only 71% of roof corners being extracted, which meant that some edges of a plane were not reconstructed, and thus each plane being a closed network of roof edge could not be held.

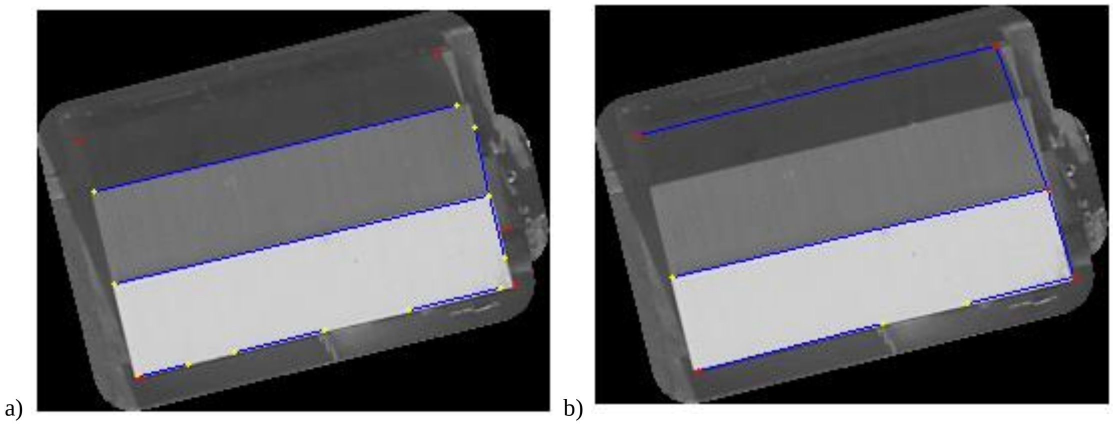


Figure 5.21 – Example results of roof-line connectivity: a) roof-line segmentation before final connectivity; and b) results of connecting roof-lines and boundary corners.

The example in Figure 5.21 shows the intermediate results before and after roof-line connectivity for the reconstructed 3D model in Figure 5.6a. The ridgeline and the step edge roof-line between the roof planes were not reconstructed in Figure 5.6a, but were correctly segmented from the extracted Canny edges, with the endpoints of the lines in approximately the correct location, shown in Figure 5.21a. The two endpoints of the step edge between the boundary of the gabled roof and flat roof extension were incorrectly connected to the boundary corners (red points in Figure 5.21). This was due to the size of the search window used and the ruleset. The length of the step edge can be seen to be one of the longest extracted edges, so the maximum search window size of 6 x 6 m was used to find connecting endpoints. Due to the large window size, incorrect connections were tested, for example between the roof-line endpoint and the boundary corner. To connect a roof-line endpoint to a boundary corner, which were not connected to other roof-lines, the ruleset calculates the angle between neighbouring boundary corners. The endpoint with the angle closest to 90°, or 180° if both angles were larger than 135°, between the previous and next boundary corner in the closed-cycle network, is determined as the correct boundary corner. In this example, the step edge was incorrect connected to the boundary corner due to the angle at the boundary corner being closest to 90°. This occurred for both endpoints of the step edge roof-line, so moved the roof-line endpoints from the correct separation of the two roof levels to the boundary of the building, shown in Figure 5.21b as the blue connected line between the two red points.

The central ridgeline of the gabled roof in Figure 5.6a was also not reconstructed, because one of the endpoints was unconnected after using the connectivity ruleset, as shown in Figure 5.21b, so was therefore removed in the final refinement stage. Again, the maximum search window was used due to the length of the line, but no endpoint was found that satisfied the connectivity ruleset. Errors can also be seen in the nDSM boundary segmentation, where five corners were extracted, represented by five red points in Figure 5.21. Four of the corners were approximately in the correct building boundary corner location, whilst the other was incorrectly connected to the endpoint of the central ridgeline, using the calculated angles between boundary corners, as previously explained for the extracted step edge roof-line from this example. This therefore shows that the proposed connectivity ruleset requires further development, and that the defined search windows were possibly too large to find connecting roof corners.

Whilst the above example shows how planar reconstruction was unsuccessful, due to the failure to connect and reconstruct roof-lines, most of the errors in roof plane reconstruction were due to the extraction of too many points and the reconstruction of incorrect roof-lines. This was caused by the size of the search window for connecting endpoints and the ruleset to connect roof-lines. The connectivity ruleset only considered the geometry of the roof-lines and did not take into account the geometry of the reconstructed planes. Noise and false edges that fitted the criteria of the line geometry were connected and created planes with irregular geometry, examples of which can be seen in Figure 5.9 and Figure 5.10. These irregularly shaped planes, created from edges extracted from false positives, consisted of corners with small acute angles. This shows that by just using orientation and length thresholds to refine the extracted edges, that the issues of irregularly reconstructed geometry cannot be resolved.

To connect roof-line endpoints to other roof-lines and boundary corners, various search windows were defined, dependent on the length of the roof-line, with respect to the longest roof edge. The search windows used were 2 m, 4 m, and 6 m. The idea of various search windows was introduced to try and reduce the connectivity of shorter edges, which may be more likely to have been incorrectly extracted, whilst allowing greater flexibility for the connectivity of longer edges. Whilst the size of the search window was dependent on the length of the line, no consideration was taken into account for the size of the building. Some buildings in the dataset had a footprint smaller than the size of the search window. For example, the roof in Figure 5.9 had maximum dimensions of 17 x 17 m, which when using a 6 x 6 m search window meant that corners over a third of the roof length could be found and tested for connectivity. This could explain why several incorrect connections were made across roof faces for the result in Figure 5.9.

The influence of building footprint size can be seen from the reconstruction results of roof type and area. Industrial and complex roof structures tended to have a much larger building footprint compared to residential buildings, which mainly consisted of gable and hipped roof structures. The results based on roof type in Table 5.3 showed that as the complexity of the roof increased, the percentage of roof planes reconstructed also increased. The reconstruction of roof planes for complex building was shown to reconstruct the highest percentage of roof planes, without regarding flat roofs, for manually selected buildings. When reconstructing by defined area, the planar extraction results of the residential areas, in Table 5.6 were lower than that of the industrial areas, in Table 5.9. This again shows how the size of the search window seems to favour buildings with a larger footprint and larger roof planes. By using a smaller search window, planar extraction from simpler roofs and residential areas may increase by reducing the number of false connections. Although the buildings in the city centre scenes were much larger than the industrial scenes, the complexity of the buildings meant the quality of planar extraction was poor, with some buildings consisting of several small roof planes.

The size of the search window also influenced the connectivity of edges from dormer windows and small roof structures, and created several false connections, with an example of this seen in Figure 5.4. Similar to planar fitting procedures discussed in Section 2.3.1.2, planar reconstruction errors occurred due to small roof artefacts. Edges extracted from roof structures, such as chimneys, could not be refined due to the height difference between the chimney and the roof plane. This caused errors in the connectivity of roof-lines, with examples shown in Figure 5.10.

As previously discussed, corners of dormer windows were not included in the reference data. Whilst there have been many attempts to automate building reconstruction, few have considered the reconstruction of dormer windows, with the focus of reconstruction being on the main roof geometry. For comparison of reconstruction results, it was therefore not the aim of this research to reconstruct dormer windows and other small roof artefacts. However, some of the results have shown how edges and corners were correctly reconstructed for dormer windows. Examples of roof structures with skylights and dormer windows were shown in Figure 5.4. Previous attempts of building reconstruction, which have used planar segmentation and geometry constraints, have struggled to reconstruct small roof structures (Rottensteiner *et al.*, 2014). The results in Figure 5.4 show the potential of this methodology to extract edges from dormer windows and skylights. The edges were successfully extracted using the Canny edge detector. The density of the DSM means that the change in gradient was detectable and measured along cross-sections of the roof to correctly refine the edges using the scan line segmentation approach. To ensure correct reconstruction further consideration is required in the connectivity workflow for these features, which will be discussed further in Section 7.4. The example in Figure 5.4b shows how correctly extracted dormer window edges were incorrectly connected to the central ridgeline and other roof-lines. This resulted in the incorrect separation of roof planes. These incorrect connections could again be due to the size of the search window, finding potentially incorrect corners, or the criteria of the connectivity ruleset. This again highlights that the connectivity criteria need further refinement to prevent these edges connecting.

5.4.3 Vegetation

Both LOD1 and LOD2 reconstructed models were also affected by vegetation. This caused issues in the LOD1 reconstruction, where vegetation was included in the nDSM boundary and caused incorrect segmentation and extraction of corners, and for LOD2 reconstruction, where edges extracted from the vegetation were incorrectly connected to roof-lines. The nDSM only distinguished between ground and non-ground features, so did not distinguish between vegetation and buildings. For LOD2 reconstruction, edges were extracted by the Canny edge detector from vegetation. In many cases vegetation was overhanging the roof surface, an example of which can be seen in Figure 5.9, which extracted edges at a higher elevation compared to the roof surface. This meant that the edges extracted from the vegetation could not be removed using the developed scan line segmentation procedure due to the height difference.

The influence of vegetation can be seen in Figure 5.22. This was one of the worst cases of reconstruction for Newcastle upon Tyne, where an extra 700% of points were extracted. It can be seen on the right of Figure 5.22 that edges extracted from the surrounding vegetation were much higher than those of the roof surface, and caused the incorrect creation of step edges. Lines were connected, which constructed several very small planes around three of the four roof corners. Due to the problems of overhanging and neighbouring vegetation, the removal of vegetation from building extraction can be considered a separate research topic, so was not investigated in this study.

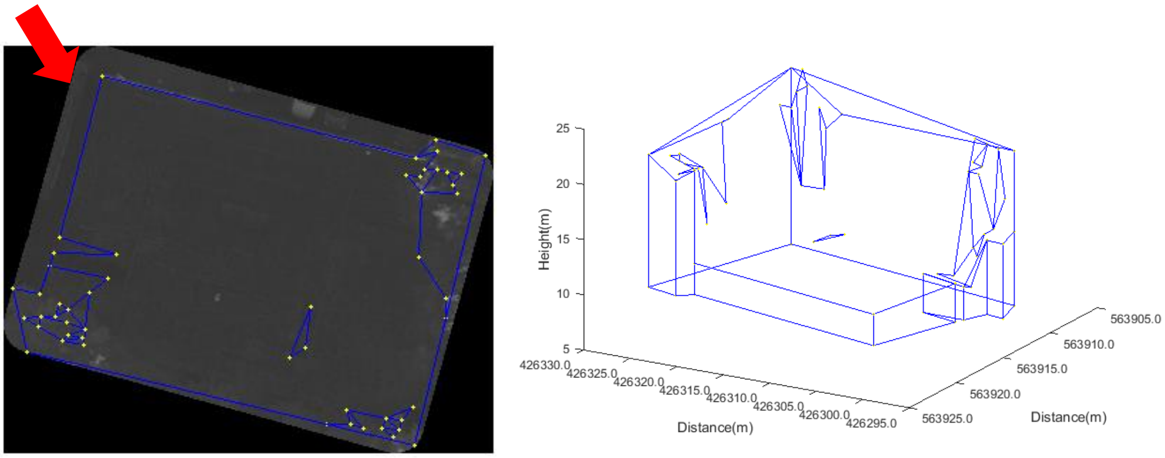


Figure 5.22 – Example of 3D reconstruction where the number of corners extracted is affected by overhanging vegetation.

Therefore, further consideration is required to distinguish vegetation and buildings. This was also concluded in the ISPRS 3D reconstruction benchmark (Rottensteiner *et al.*, 2014). One of the benefits of digital imagery capture is that full colour and NIR imagery can be captured simultaneously. This should allow the classification of vegetation using a measure such as NDVI, which has been used for 3D reconstruction in the past (Awrangjeb *et al.*, 2012; Demir and Baltsavias, 2012; Grigillo and Kanjir, 2012). However, as previously explained in Section 3.5.1, the provided imagery was captured in November, when most of the trees had leaf-off conditions, so do not exhibit green colour, which makes it difficult to classify vegetation using NDVI.

5.5 Summary

The results of reconstruction for the Newcastle upon Tyne dataset were shown to extract approximately 70% of roof corners, for both block model and models with roof geometry, to a positional accuracy of ±0.55 m. The height accuracy of corners was not as high, with an average accuracy of ±1.34 m. This was due to errors in the pixel-to-pixel matching procedure and the extraction of heights from the nDSM. The accuracy achieved shows corners of buildings were reconstructed to the accuracies required for LOD2 models of CityGML, which was the aim of this research. However, nearly all reconstructed roof models suffered from too many points being extracted and constructed, which caused poor planar extraction. The reasons behind these various errors have been explained and attributed to misclassification of the ground from the DTM, the extraction of roof geometry and the connectivity ruleset; in particular, the size of the search window used to find potential connecting endpoints. Whilst issues have been determined in the workflow, the transferability of the method will be tested on the dataset outlined in Section 3.5.2. This should clarify if the reconstruction errors were due to the reasons discussed in this chapter, or may be influenced by the imagery and roof geometry.

Chapter 6. Reconstruction Results of Vaihingen, Germany

6.1 Introduction

To investigate the transferability of the procedure, buildings were reconstructed from imagery captured over a different scene with a different camera system. The camera used to capture imagery of Vaihingen has been outlined in Section 3.5.2, along with the processing of the data. Compared to the Newcastle upon Tyne dataset, imagery was captured at a slightly higher spatial resolution, with a GSD of 9 cm. For this study site, four areas were selected for reconstruction. Area1 and Area3 are included in the ISPRS 3D reconstruction benchmark and are described as an inner city area characterized by dense complex buildings, whilst Area3 is a residential area consisting of small detached house, respectively (Rottensteiner *et al.*, 2013). These two areas allow a direct comparison of the developed method to those currently proposed by other researchers. Another two areas, Area21 and Area32, were selected based on the geometry of the roof shapes. The extracted buildings for each area can be seen in Appendix F. None of the parameter settings from the Newcastle upon Tyne dataset were changed for the 3D reconstruction of buildings from the Vaihingen imagery. This was to determine how well the parameter settings performed with different data.

LOD1 Reconstruction	Average RMSE (m)			Percentage of corners detected (%)	Percentage of total corners detected (%)	Number of corners	Number of corners extracted
	X	Y	Z				
Area1	0.82	0.87	1.80	75.4	169.5	351	596
Area3	0.70	0.65	1.59	74.6	176.3	388	684
Area21	1.00	0.81	2.25	55.5	127.3	256	326
Area32	0.71	0.80	1.22	60.4	166.4	107	178

Table 6.1 – Accuracy and number of corners extracted for LOD1 reconstruction.

LOD2 Reconstruction	Average RMSE (m)			Percentage of points detected (%)	Percentage of total corners detected (%)	Number of corners	Number of corners extracted
	X	Y	Z				
Area1	0.67	0.54	1.41	64.2	237.3	899	2133
Area3	0.73	0.57	1.53	80.9	342.2	557	1906
Area21	0.52	0.48	1.58	73.0	359.0	424	1522
Area32	0.54	0.79	1.14	79.8	317.0	153	485

Table 6.2 – Accuracy and number of corners extracted for LOD2 reconstruction.

The results presented in Table 6.1 and Table 6.2 show the accuracy and the number of extracted corners for LOD1 and LOD2 reconstruction, respectively. The results show a large number of extra points were constructed as corners: on average an extra 214% of points were extracted for the LOD2 reconstruction of the four test sites. Because of this high level of incorrect roof-line construction, the statistical quality of the extracted roof planes could not be measured.

6.2 Area1

This scene is an area of the city centre of Vaihingen with historic buildings consisting of complex shapes (Rottensteiner *et al.*, 2013). 42 buildings were extracted covering a footprint of nearing 40,000 m². LOD1 reconstruction was achieved with 75% of roof corners correctly extracted to a positional accuracy RMSE of ±0.85 m, but a much lower height accuracy RMSE of ±1.59 m. As shown in Figure 6.1 and Figure 6.2, this was the result of neighbouring features such as trees and buildings being included in the reconstruction. Some of the boundary points were extracted near to their true location, but because of overhanging vegetation, the height of the corner could not be extracted accurately. These neighbouring features also caused the over-segmentation of the nDSM boundary, with an extra 70% of points extracted for Area1 LOD1 reconstruction.

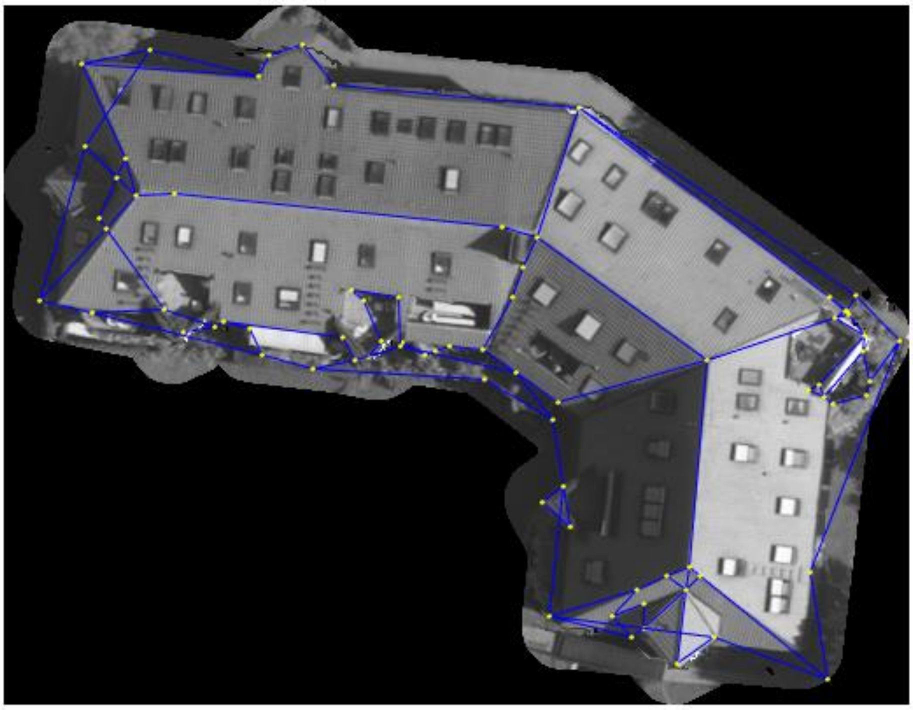


Figure 6.1 – Example plan view of LOD2 Reconstruction for a complex roof structure.

Out of the four Vaihingen test sites for LOD2 reconstruction, it can be seen from Table 6.2 that the lowest percentage of roof corners was extracted for Area1 with only 64% of correct corners reconstructed. The correct corners that were extracted were to a positional accuracy RMSE of ±0.61 m. Again, the height accuracy was much lower with an RMSE of ±1.41 m. The low percentage of corners extracted was due to the complexity of the roof shapes. An example of a building with a complex roof structure can be seen in Figure 6.1. The roof consisted of lots of small extensions, roof texture from skylights, as well as indentations in the roof for balconies. 81% of correct roof corners were extracted to a positional accuracy RMSE of ±0.60 m for this building. The main ridgelines, valley lines and some of the small extensions were successfully reconstructed. However, an extra 137% of points were constructed, which can be seen from neighbouring vegetation and the incorrect connection of small indentations and extensions of the roof.

Area1 also consisted of some smaller residential buildings, which were also reconstructed. An example of this can be seen in Figure 6.2 for a gable roof, which has two dormer windows. The extraction of roof corners for this example was very successful, extracting 95% of the correct roof corners to a positional accuracy RMSE of ±0.46 m, and a height accuracy RMSE of ±0.64 m. The ridgelines of the dormer windows were extracted and the 3D shape reconstructed. The corners of the dormer lines did not connect to the nDSM boundary and meant the connection between the dormers and the roof perimeter was not reconstructed, giving the impression of floating surfaces, as shown in Figure 6.2. As was seen for the reconstruction in Figure 6.1, too many points were extracted, which produced an incorrectly constructed model. The results in Figure 6.2 constructed an extra 109% of corners from incorrect connections along the central ridgeline to dormer windows, and from neighbouring vegetation. Errors in the nDSM boundary also resulted in the incorrect determination of step edges, seen in Figure 6.2, which added to the corner extraction count. This result also highlights the difficulty in measuring planar extraction.

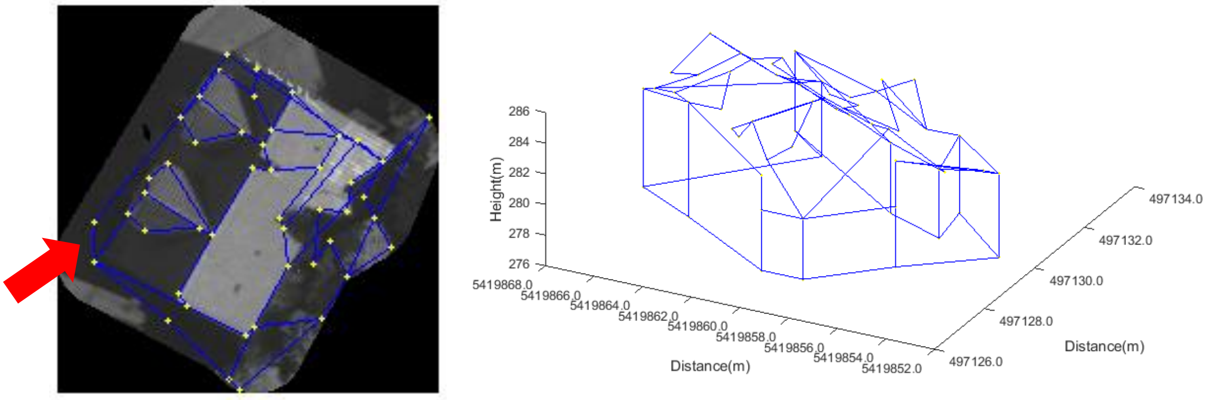


Figure 6.2 – Example LOD2 reconstruction of a gable roof with dormer windows.

6.3 Area3

Area3 was another of the ISPRS benchmark test sites. This site was described as a purely residential area with small detached houses (Rottensteiner *et al.*, 2013). Many of the roofs therefore had very simple roof structures such as the gabled roofs shown in Figure 6.3 through to Figure 6.5. 71 buildings were extracted covering an area of nearly 49,000 m².

The results for LOD1 reconstruction were very similar to those from Area1: 75% of roof corners were reconstructed to a positional accuracy RMSE of ±0.68 m, with a worse height accuracy RMSE of ±1.59 m. Similarly to Area1, an extra 75% of points were constructed, showing incorrect levels of corner point extraction and refinement. These errors can again be attributed to neighbouring features and errors in the ground classification. An example of this can be seen in the incorrect reconstruction of step edges, which were present in all three examples shown for Area3, as well as the small extension of the building in Figure 6.4, which resulted in a corner with an acute angle being reconstructed.

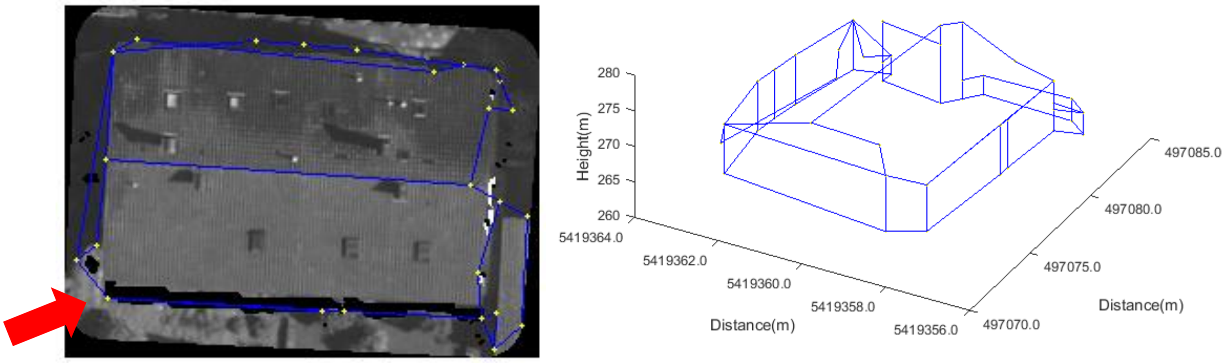


Figure 6.3 – Example LOD2 reconstruction for a gable roof.

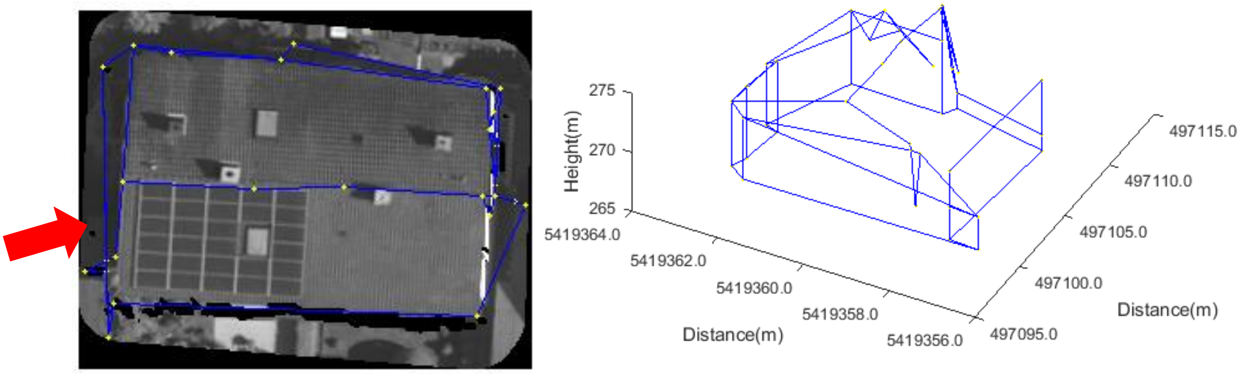


Figure 6.4 – Example LOD2 reconstruction for a gable roof.

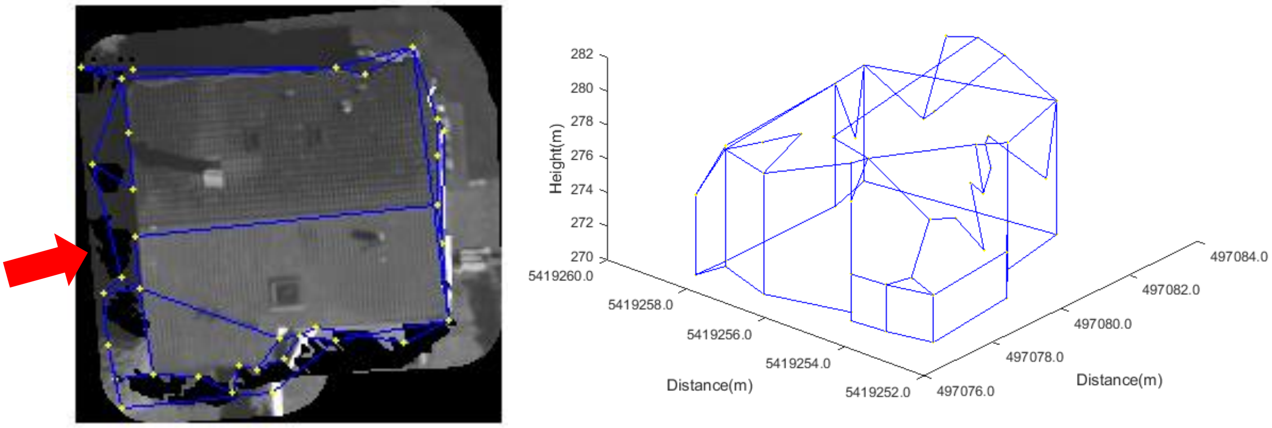


Figure 6.5 – Example LOD2 Reconstruction for a gable roof.

For LOD2 reconstruction, Area3 extracted the highest percent of roof corners of all four test sites, with 81% of corners reconstructed. When compared to Area1, the higher percentage of corners reconstructed was due to the simpler nature of the roof geometry. Examples of 3D reconstruction for three gabled roofs from Area3 can be seen in Figure 6.3 through to Figure 6.5. These three figures show the successful extraction of roof corners, with 100% of corners successfully reconstructed for all three figures, to varying positional and height accuracies. However, all three figures also show the construction of false roof-lines from the extraction of incorrect points. The example in Figure 6.3 constructed an extra 200% of roof corners, whilst the examples in Figure 6.4 and Figure 6.5 extracted over an extra 300% of points as roof corners. The three examples show varying levels of roof texture where corners could have been connected. In most cases, edges were not extracted from the roof texture, with the extra corners being constructed from holes in the true-orthophoto, seen in the three figures as black sections within the imagery, and from the over-segmentation of the nDSM boundary, which also resulted in roof-lines being incorrectly connected to these points. Roof texture caused extra corners to be extracted on the central ridgeline of Figure 6.4, but it can also be seen from this figure that no edges were incorrectly created from the roof texture between the central ridgeline and the roof boundary.

From visual analysis of the results in Figure 6.3, Figure 6.4 and Figure 6.5, the success of the roof plane reconstruction can be seen. As shown with the Newcastle upon Tyne dataset, if one connecting edge was missing, the result was an unclassified roof plane. An example of this can be seen in Figure 6.5, where one roof plane was reconstructed, but a connecting gable line was missing for the opposite plane, so was therefore not reconstructed. Whilst it can visually be seen that roof plane extraction was successful in the three figures, to varying degrees, an extra 288% of points were extracted, on average, for each building which meant statistical analysis of the roof planes was not possible.

6.4 Area21

Area21 consisted mainly of residential buildings but also included some larger industrial buildings. 28 buildings were extracted from an area covering approximately 39,000 m^2. Most of the buildings also had simple roof geometry. Although the roof structures were fairly simple, Area21 was the worst performing site out of all four test sites.

LOD1 reconstruction showed the poorest extraction of roof corners for all four test sites, with only 55% of correct roof corners reconstructed. The corners that were correctly reconstructed had a positional accuracy RMSE of ±0.90 m and height accuracy RMSE of ±2.25 m. Although it was the worst performing site, Area21 extracted the fewest extra points, with an extra 27% of points constructed as corners for LOD1 models. An example of LOD1 reconstruction can be seen in Figure 6.6a, which suffered from positional and height inaccuracies, as well as over-segmentation of the nDSM boundary, due to surrounding vegetation.

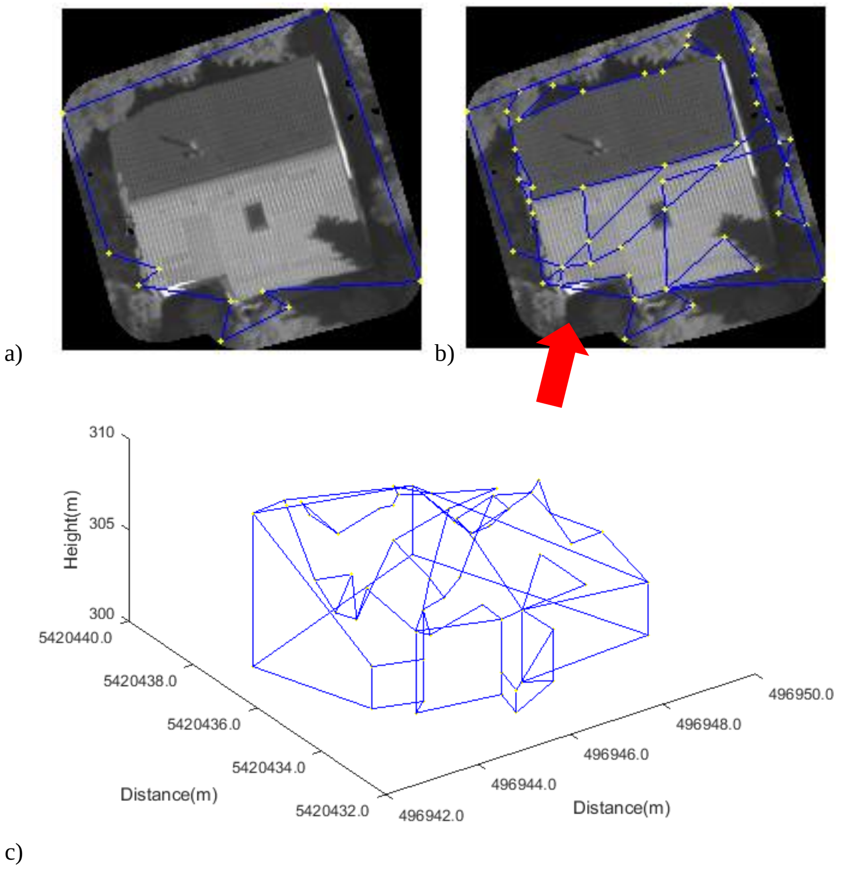


Figure 6.6 – Example 3D reconstruction of a gable roof: a) LOD1 boundary; b) LOD2 reconstructed edges and; c) 3D view of LOD2 model.

The positional accuracy of correctly extracted corners for LOD2 reconstruction was within ±0.5 m of the truth, whilst the method only extracted 73% of the roof corners. However, an extra 258% of points were extracted on average per building. An example of this reconstruction can be seen in Figure 6.6, where the 2D views of LOD1 and LOD2 reconstruction can be seen with the 3D model for LOD2 reconstruction. Due to several trees surrounding the building, the ground classification was not at the boundary of the roof and caused reconstruction errors. Only 50% of boundary corners were extracted for LOD1 reconstruction within the 2 m positional tolerance and with a resulting RMSE of ±0.73 m. Although 100% of the correct roof corners were reconstructed for the LOD2 model, nearly an extra 500% of points were extracted. This resulted in many false connections being constructed, which can be clearly seen in Figure 6.6b, where edges have been constructed across the roof plane and several short edges along a gable line of the roof. The LOD2 model in Figure 6.6b reconstructed all roof corners to within a positional accuracy RMSE of ±0.30 m. However, as can be seen in Figure 6.6c the height accuracy was not as accurate, with an RMSE of ±1.81 m. Incorrect height extraction resulted in one of the ridgeline endpoints having a lower height value than the eave line corners and causing the gable edges to slope downwards instead of upwards to the correct height of the ridgeline. The large amount of falsely constructed roof-lines, particularly across the roof plane, made it almost impossible to measure planar extraction. More examples of incorrect reconstruction from Area21 can be seen in Figure 6.12 and Figure 6.13, which will be discussed later in more detail.

6.5 Area32

Area32 consisted of a few large industrial buildings, mainly with flat roofs or very simple roof geometry. The footprint of the buildings covered a similar sized area as the other test sites, approximately 36,000 m², but only 19 buildings were extracted, due to the large size of the buildings, as seen in Appendix F.4. LOD1 models were reconstructed with 60% of corners correctly extracted to a positional accuracy within ±0.75 m. The percentage of corners reconstructed and the positional accuracy of LOD2 models was slightly greater, compared to LOD1, with 80% of corners reconstructed to a positional accuracy within ±0.66 m.

As was seen with the other three test sites, errors in the nDSM meant that the reconstruction of some boundary corners was worse in the LOD1 model, compared to the LOD2 models. This is evident in both Figure 6.7 and Figure 6.8, where corners at the boundary of the two different height levels were extracted for LOD2 reconstruction. For LOD1 reconstruction, some of these corners were extracted, but not within the position tolerance. The success of the reconstructed boundaries corners for LOD1 modelling was 88% for the example in Figure 6.7a, and 75% for the example in Figure 6.8a.

Both the examples in Figure 6.7b and Figure 6.8b extracted 100% of roof corners for LOD2 reconstruction, to within a positional accuracy of ±0.5 m. The extracted heights were also close to the truth with a height accuracy RMSE of ±0.46 m for the example Figure 6.7b, and ±0.60 m for the example in Figure 6.8b. The difference in the reconstruction of LOD1 and LOD2 models, and the height inaccuracies, resulted in some floating surface, as seen in Figure 6.8c.

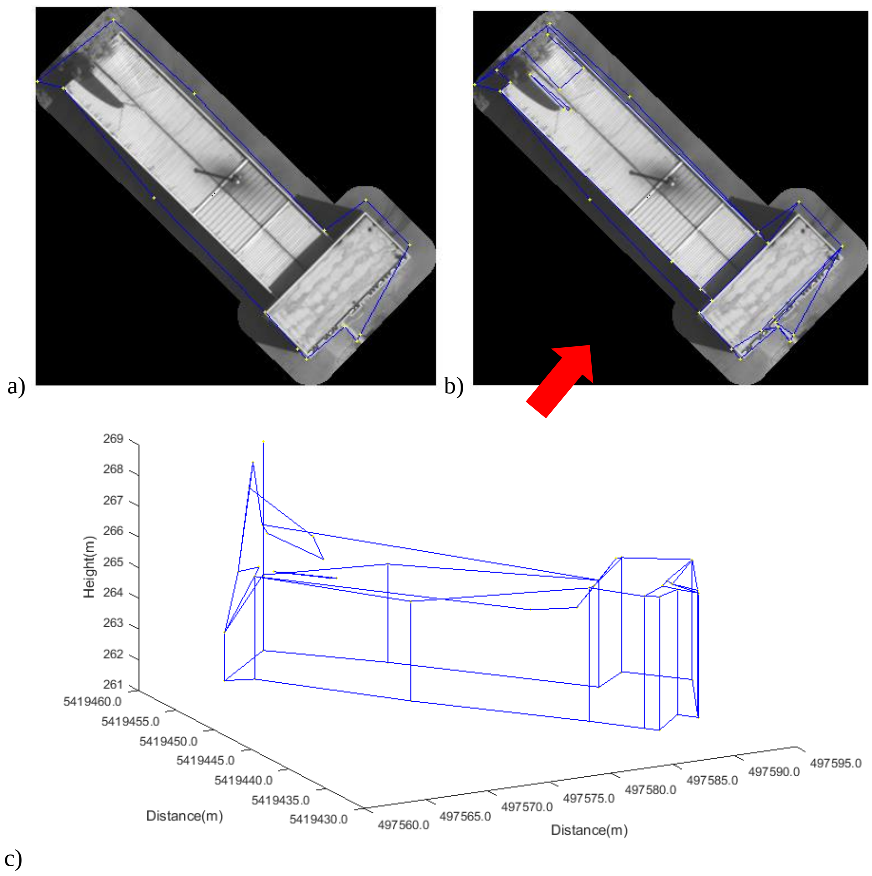


Figure 6.7 – 3D reconstruction of a multi-level flat roof: a) LOD1 roof reconstruction; b) LOD2 reconstruction and; c) 3D view of the LOD2 model.

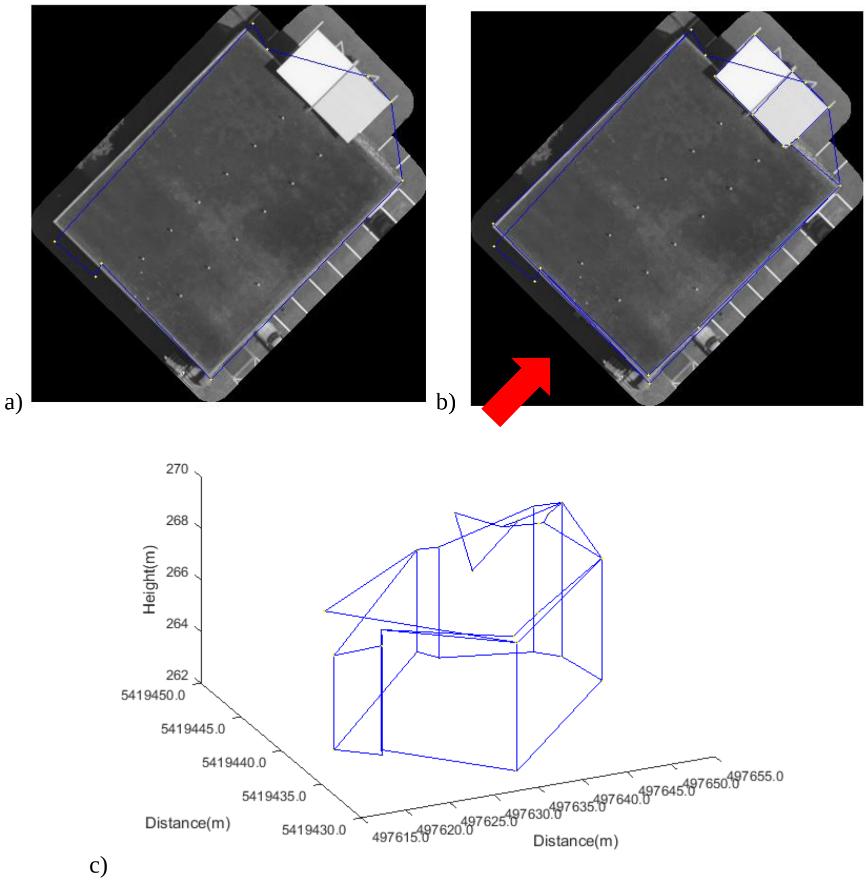


Figure 6.8 – 3D reconstruction of a multi-level flat roof: a) LOD1 reconstruction; b) LOD2 reconstruction and c) 3D view of the LOD2 model.

Both LOD1 and LOD2 models suffered from the extraction of too many points. LOD1 models constructed, on average, an extra 66% of points as corners, whilst LOD2 models constructed, on average, an extra 216% of points as corners. As discussed with the other reconstructed examples, this was due to vegetation, incorrect nDSM boundary segmentation and false positive extraction on the roof surfaces.

6.6 Discussion

Compared to the reconstruction results of Newcastle upon Tyne, the results of 3D building reconstruction of Vaihingen can be seen to have been unsuccessful: the positional accuracy was slightly worse with ±0.8 m for LOD1 and ±0.61 m for LOD2, and the height accuracy was worse compared to individual building selection, but comparable to reconstruction of an area, with an average accuracy of ±1.6 m. The building reconstruction in Vaihingen extracted a far greater number of points as corners when compared to reconstruction in Newcastle upon Tyne. It can be concluded that the parameters used at various stages of reconstruction do not transfer successfully to other datasets, due to differences in the datasets and incorrect parameter settings, which will be further discussed.

6.6.1 LOD1 Reconstruction

Overall, the results of LOD1 reconstruction were unsuccessful. On average, only 66% of roof corners were correctly reconstructed within a positional accuracy of ±0.80 m and a height accuracy of ±1.72 m. Many of the reconstruction errors were the result of incorrect ground classification, and thus resulted in incorrect nDSM boundary segmentation for LOD1 reconstruction. The various examples above show how extra points were extracted from vegetation, nearby buildings and holes in the true-orthophoto. An example of a segmented nDSM boundary can be seen in Figure 6.9a.

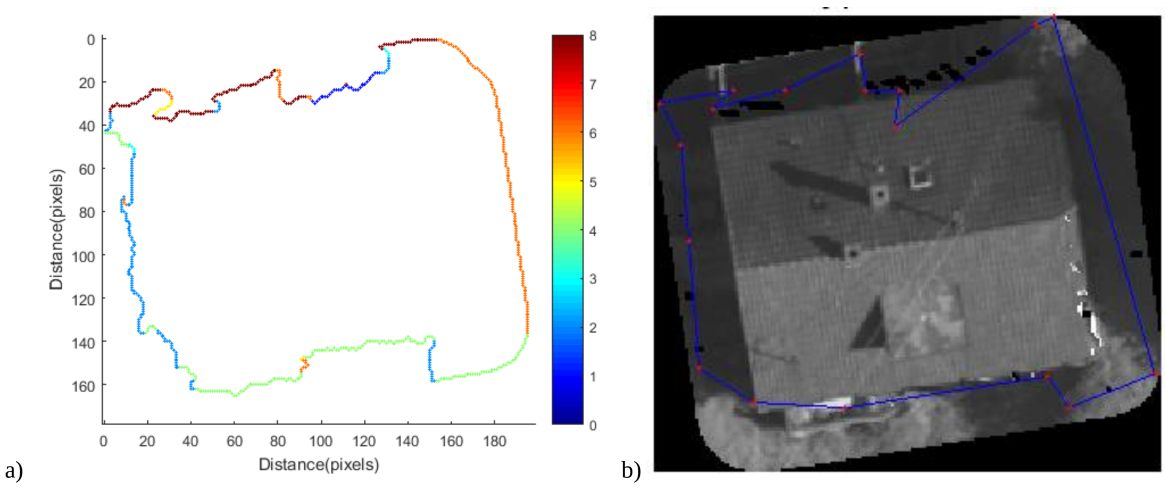


Figure 6.9 – Example results of: a) the segmentation of the nDSM boundary; and b) the resulting corners extracted from the segmentation of the nDSM boundary.

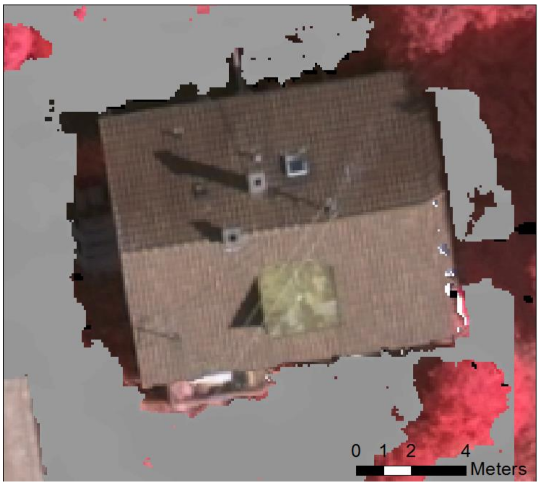


Figure 6.10 – Ground classification used to produce the nDSM boundary in Figure 6.9a.

The nDSM boundary has been heavily over-segmented due to the irregular nature of the edges. This caused lots of short and irregular lines to be extracted, which even after some initial refinement produced incorrect segmentation, as shown in Figure 6.9b. The straight edge of the boundary was produced from the image extent. The irregular nature of the nDSM was caused by the ground classification, which can be seen in Figure 6.10. The ground classification does not coincide with the boundary of the roof, because of neighbouring vegetation and misclassification along one of the roof edges, which caused a jagged edge in the nDSM. The vast amount of over-segmentation of the nDSM boundary, for this example, led to incorrect reconstruction of the LOD1 model, with only 50% of points being extracted within the positional 2 m tolerance. The height accuracy RMSE was ±4.46 m, for this example, due to the associated height at one corner being selected from the ground and not the roof surface.

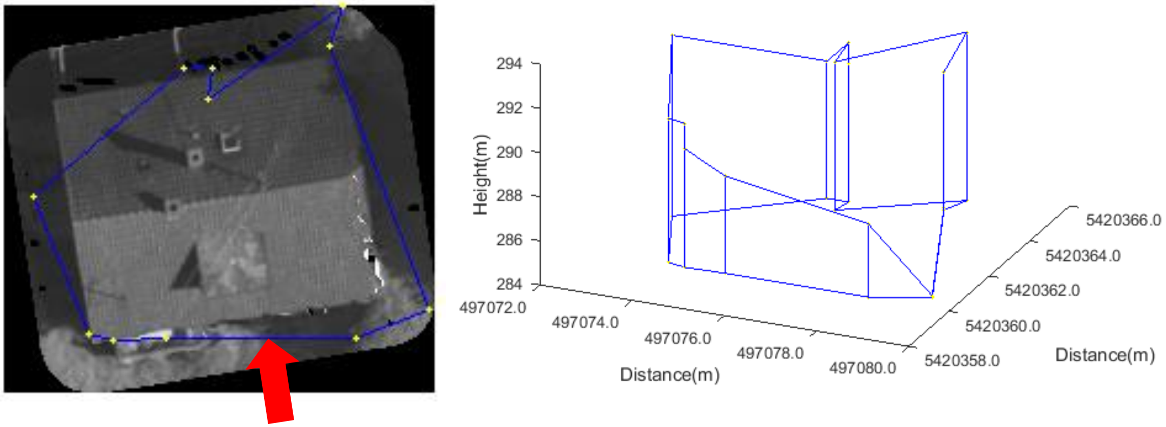


Figure 6.11 – LOD1 Reconstruction for a gabled roof from the resulting segmentation in Figure 6.9a.

The average total percentage of corners extracted for LOD1 reconstruction across the four test sites was 170%. Although this can be linked to the over-segmentation of the nDSM boundary, this also shows that the orientation thresholds were not sufficient to refine the edges. This could be due to the leniency of the thresholds to not limit reconstructed corners to orthogonal constraints, as was shown by several methods in Chapter 2. In this approach eight equally spaced orientation thresholds from the main orientation were implemented to remove incorrect lines. The orientation thresholds do not take into account the angle of the corner. Some of the results showed irregular geometry, with the reconstruction of roof corners at acute angles. This can be seen in the LOD1 model reconstructed in Figure 6.11, where a small acute angle was determined at one corner point using just orientation to refine edges. Other examples of errors caused by refined boundary lines using just orientation and length can be seen in Figure 6.4 and Figure 6.6a for different building types. Therefore, in order to improve LOD1 reconstruction, the computation of the nDSM needs to be improved: to improve the ground classification and remove neighbouring features such as vegetation, as well as developing the connectivity and refinement ruleset of segmented edges.

6.6.2 LOD2 Reconstruction

Compared to LOD1 reconstruction of the four areas, the results from LOD2 models were slightly more successful, in terms of corners extracted and the accuracies achieved. On average, 75% of roof corners were reconstructed to a positional accuracy of ±0.61 m and a height accuracy of ±1.42 m. This again highlights that the positional accuracy of corners was better than the height accuracy. Similarly to the Newcastle upon Tyne dataset, the positional accuracy of corners was close to those defined by CityGML LOD3, and the height accuracy was within those defined for LOD2 (Gröger and Plümer, 2012). However, the total percentage of points reconstructed, when compared to the total number of roof corners was, on average, 335%, which was much higher than reconstruction from the Newcastle upon Tyne dataset. Errors can be associated with the complexity of the scenes used. As described in Rottensteiner *et al.* (2013) the buildings in Area1 and Area3 were of high complexity. Whilst the method has been shown to handle the reconstruction of complex buildings, in Chapter 5, the over extraction of modelling cues, together with the high complexity of the roof shapes meant that buildings were unsuccessfully extracted. The two main causes for significant over extraction of roof corners for LOD2 reconstruction in the Vaihingen dataset were due to the size of the search windows for roof-line connectivity and the Canny edge detector.

As discussed for the Newcastle upon Tyne dataset, the size of the search window to find connecting endpoints was a cause of incorrectly connected edges. The size of the search window was set to 2 m, 4 m, or 6 m, dependent on the length of an unconnected line with respect to the longest extracted roof-line. The size of the search window was constructed using the number of pixels in an image for the relevant size. However, the GSD of the image was not accounted for. The Vaihingen imagery was captured at a GSD of 9 cm, whilst the Newcastle upon Tyne imagery was captured at 10 cm GSD. Therefore, the radius of the windows varied between the two datasets. The search windows used for Vaihingen were thus calculated with radii of 1.8 m, 3.6 m and 5.4 m. Although a slightly smaller search window was used, the results indicate that the search window was still too large. The size of the search window was dependent on the length of the extracted lines and not the building footprint. The building reconstructed in Figure 6.6 had a building footprint of 10 x 10 m, whilst the building reconstructed in Figure 6.2 had a building footprint of 10 x 8 m. The maximum window size used covers more than a quarter of the whole roof structure for the building in Figure 6.6, and nearly the entire width of the building in Figure 6.2. Both examples show the incorrect connection of points across roof faces and from neighbouring features. For these test sites, smaller window sizes, which have a smaller difference between the sizes, should be used. The difference in GSD of the two datasets also influenced the refinement of the run graph vectorisation results for the roof-lines and the extraction of height values around each defined corner.

Errors in the height values can also be associated with the surface model, as was explained with the reconstruction of buildings with no roof geometry in Section 6.6.1. Errors in the ground classification resulted in the incorrect roof boundary being reconstructed, as previously discussed. Although these errors can be attributed to error in the DSM, they were also caused by the ground classification parameters used, which were also shown to have been incorrectly set for some parts of the Newcastle upon Tyne dataset. Errors in the ground classification also resulted in the incorrect reconstruction of step edges, with too many boundary corners being extracted and assigned height values between connecting corners greater than the threshold.

The true-orthophoto was provided using the green, red and NIR colour bands. The inclusion of the NIR band offers better contrast of edge features compared to RGB imagery (Demir and Baltsavias, 2012). The thresholds of the Canny edge detector were set low in order to extract edges of dormer windows from RGB imagery. By changing the combination of bands to produce the Green, Red, Near-Infrared (GRNIR) colour true-orthophoto, the contrast is likely to have been enhanced, and more false positives were likely to be extracted. This could be the reason why too many roof edges were extracted and resulted in the reconstruction of several false roof-lines. If there is better contrast of edges, the Canny edge detector should be able to still detect the correct edges, but with a higher threshold. To test this theory, the buildings of Area21 were reconstructed with an increased maximum threshold for the Canny edge detector of 0.1, instead of 0.05. The results of this reanalysis can be seen in Table 6.3, and compared against the results using the original parameters.

6.6.3 Parameter Enhancement

LOD2 Reconstruction	Average RMSE (m)			Percentage of corners detected	Percentage of total corners detected	Number of corners	Number of corners extracted
	X	Y	Z				
Area21 – Original	0.52	0.48	1.58	73.0	410.1	424	1522
Area21 – adjusted	0.64	0.50	1.61	72.3	335.7	424	1218

Table 6.3 – Reconstruction results for Area21 by adjusting the Canny edge detector.

The positional and height accuracy, as well as the percentage of correct roof corners were similar using both thresholds, but a reduction can be seen in Table 6.3 for the percentage of total corners detected. This shows, on average, that by increasing the Canny maximum threshold, no correct roof corners previously extracted were removed, nor were the accuracies decreased, but the number of total lines constructed was reduced. Two examples of 3D reconstruction are shown with the original Canny threshold and refined threshold in Figure 6.12 and Figure 6.13.

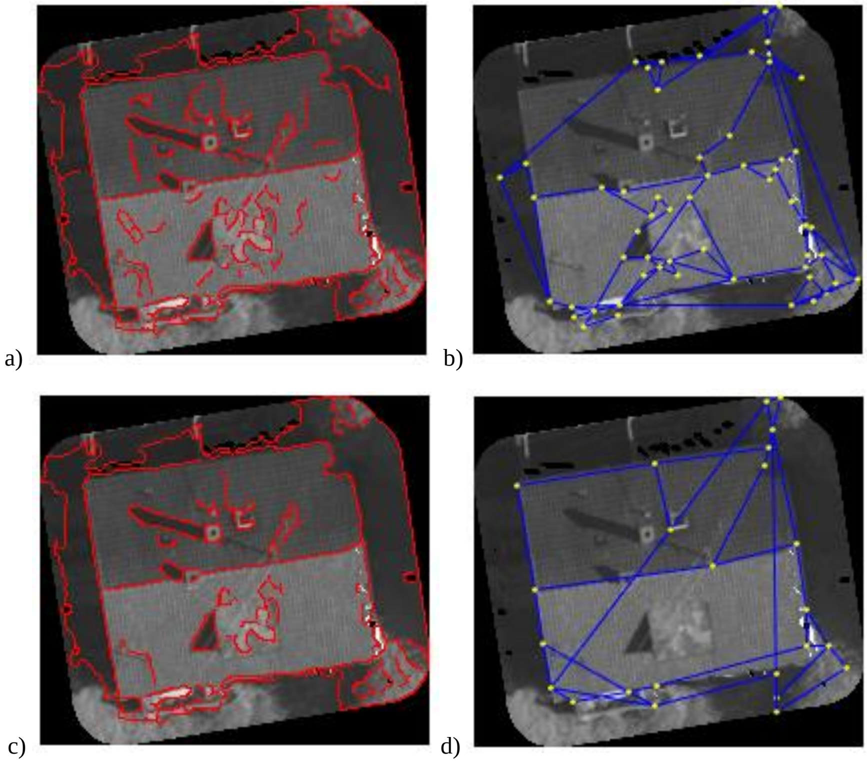


Figure 6.12 – Results of roof geometry reconstruction: a) Canny edges detected with original thresholds; and b) the resulting plan view of the LOD2 model; c) Canny edges detected with the refined threshold; and d) the resulting plan view of the LOD2 model.

The results of Figure 6.12a and Figure 6.12c show that by increasing the maximum threshold of the Canny edge detector, the number of detected edges was decreased. This is noticeable for edges around the dormer window, on neighbouring vegetation and on the two roof faces. The reduction in the number of edges extracted, shows improvements in the percentage of total corners reconstructed. With the original threshold, an extra 883% of points were constructed as roof corners, shown in Figure 6.12b, whereas with the refined threshold an extra 300% of points were constructed, shown in Figure 6.12d. This shows that 60% fewer points were constructed with the refined threshold compared to the original threshold. Another difference between the results of Figure 6.12b and Figure 6.12d was the increase in the number of correct roof corners extracted from 83% to 100%. This was due to the connectivity ruleset connecting an unconnected endpoint to the first corner that matched the defined criteria. The endpoint of the eave line, extracted from edge detection, was evidentially connected to an incorrect edge with the original Canny threshold. With the refined threshold, this incorrect corner was not extracted and the connection was made between the correct edges, and resulted in the correct corner being reconstructed at the boundary corner between the gable line and the eave line of the roof.

The reduction in the roof-line corners extracted also influenced the LOD1 model by reducing the number of corners reconstructed from 300% to 225%. The long diagonal edge in Figure 6.12d, from one boundary corner across the two roof planes to the opposite diagonal corner, was from the LOD1 model. The nDSM boundary and the segmentation results for this building can be seen in Figure 6.9b, where several corners were extracted along the left side boundary edge. The result in Figure 6.12b shows that an extracted gable line was incorrectly connected to one of the segmented boundary corners, which meant that, although the boundary corner was incorrectly extracted and connected, it could not be refined or removed from the LOD1 model. This connection between the boundary corner and gable line was not made in the results of the refined maximum threshold for Canny edge detection, because the edge was not extracted, so the boundary corner was removed in the LOD1 model. Other connections between roof-lines and boundary corners were also not made, which reduced the number of corners for LOD1 reconstruction. The correct extraction of boundary corners was still only 50% for LOD1 reconstruction, with incorrect corners still extracted from errors in the nDSM boundary and the correct corner, previously discussed, along the eave line being reconstructed from the extracted roof-lines.

Another example from the results of increasing the maximum threshold of the Canny edge detector are shown in Figure 6.13 for a pyramid shaped roof, which has a lower-levelled flat roof extension to the building. A clear reduction in the number of extracted edges can be seen between the original threshold results in Figure 6.13a, compared to using refined threshold results in Figure 6.13c. Whilst most of the false positives on the roof planes were removed, there was only a slightly reduction in the edges extracted from overhanging vegetation. The reduction in the number of Canny edges extracted resulted in fewer corners being incorrectly constructed. The number of corners constructed reduced from an extra 700% to an extra 391% of points, and resulted in a reduction of 39% of points being reconstructed. The reduction in corners did not influence the percentage of correct corners reconstructed, which stayed constant at 82%. Improvements can be seen, when comparing the results of Figure 6.13b and Figure 6.13d, in edges extracted along the step edge of the pyramid and flat roof planes, as well as removing all falsely connecting lines on the flat roof surface. The influence of texture from the window at the peak of the pyramid was also reduced using a higher maximum threshold.

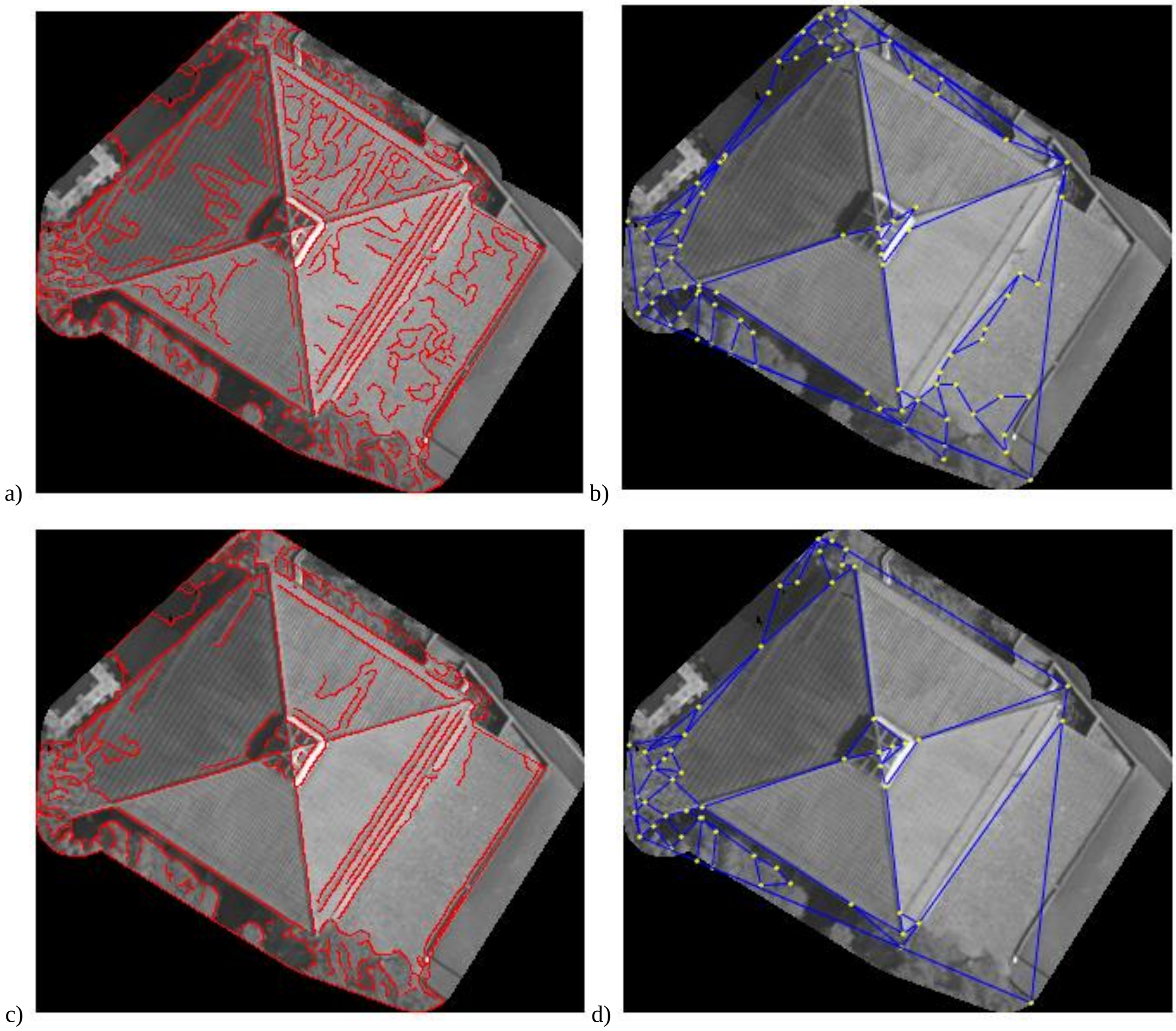


Figure 6.13 – Results of roof geometry reconstruction: a) Canny edges detected with original thresholds; and b) the resulting plan view of the LOD2 model; c) Canny edges detected with the refined threshold; and d) the resulting plan view of the LOD2 model.

The results of increasing the maximum threshold of the Canny edge detector have been shown to improve the reconstruction results. However, the percentage of corners reconstructed was still very high using a greater maximum threshold, with over 335% of corners constructed, a reduction of approximately 22% of incorrectly detected points. Although the threshold goes some way to reducing the incorrect extraction of edges, some incorrect edges will always be extracted, regardless of the threshold. An example of this was where holes were present in the true-orthophoto, as shown in Figure 6.5. These holes were created by errors in the orthorectification of the image, where the pixel could not be orthorectified. Whilst methods have tested the reconstruction of 3D models using this imagery (Rottensteiner *et al.*, 2014), only three out of 14 methods tested in the ISPRS benchmark used the imagery for reconstruction, which may be due to the matching errors highlighted.

6.6.4 Comparison to Newcastle upon Tyne reconstruction

Although different sensors, image configurations and processing software was used for the Newcastle upon Tyne and Vaihingen datasets, some comparisons can be drawn from the geometry of the buildings. Area1 of Vaihingen is a residential area, and when compared against the Residential areas of Newcastle upon Tyne, the results show a similar number of correct corners reconstructed with a similar planimetric accuracy. However, a difference in the height accuracy was seen when comparing the two datasets. This was likely to be due to the processing parameters used to classify the ground classification, where errors have previously been discussed. The poorer height accuracy could also be due to matching errors of stereo-pairs for the reconstruction of the DSM for the Vaihingen dataset (Rottensteiner *et al.*, 2014). Both sites reconstructed a similar percentage of correct corners, but Vaihingen extracted and constructed a much higher percentage of points as corners. As previously discussed, this was attributed to the inclusion of the near-infrared colour band in the true-orthophoto, which increased the contrast of the imagery and caused the Canny edge detector to extract too many edges.

When comparing Area 3 from Vaihingen to the City Centre scenes of Newcastle upon Tyne, several differences were noticed. For the accuracy of the corners reconstructed, Newcastle upon Tyne produced a greater planimetric accuracy, whilst Vaihingen produced a greater height accuracy. Although errors in the height accuracy of corners from Vaihingen have been attributed to the ground classification and errors in the DSM, the accuracy was much better than Newcastle upon Tyne due to the complexity of the buildings. As seen in Figure F.2 Appendix F, many of the roofs in Area 3 were gable-roofs which offer simple geometry, when compared to the roof shapes in Figure E.11 and Figure E.13 of Appendix E for Newcastle upon Tyne. The buildings in Newcastle upon Tyne city centre scenes were also across multiple levels, which also made it more difficult to reconstruct the correct corner height. The percentage of corners reconstructed can also be contributed to the geometry of the buildings. Buildings in Area3 were reconstructed with a higher percentage of correct corners compared to Newcastle upon Tyne, but also with a much higher percentage of points incorrectly constructed as corners. The simpler geometry from Vaihingen meant corners were more likely to be extracted, but similarly to

Area1, the colour band combination of the Vaihingen true-orthophotos meant a greater number of edges were incorrectly extracted.

6.6.5 Comparison to ISPRS Benchmark Results

The accuracy of the reconstructed corners for 3D reconstruction of building in Vaihingen was compared to other researcher's methods. Due to the large amount of incorrectly reconstructed roof-lines, planar extraction could not be measured. Thus, this limited the comparison of methods to the accuracy of corner extraction only. The planimetric accuracy of reconstructed corners was very similar to other reconstruction methods for 3D reconstruction from the Vaihingen dataset. The best achieved average positional accuracy was 0.60 m, but results were only present for Area3 using this method (Zhang *et al.*, 2012; Rottensteiner *et al.*, 2014). The average positional accuracy achieved using the proposed method in this research for reconstruction of Area3 was 0.65 m, which slows close comparison. Of the methods that tested all three areas, the best achieved average positional accuracy was 0.63 m (Sohn *et al.*, 2012; Rottensteiner *et al.*, 2014). The two methods that reconstructed buildings from imagery had positional accuracies of 1.10 m and 0.73 m (Rau and Lin, 2011; Bulatov *et al.*, 2014; Rottensteiner *et al.*, 2014). The average positional accuracy of all methods was 0.81 m, whereas the average for the results for Area1 and Area3 using the method in this research was 0.63 m. This allows the following conclusions to be made. Firstly, the positional accuracy of the corners was similar to all methods in the ISPRS benchmark. Of the methods using imagery for reconstruction, a better positional accuracy was achieved using the proposed method.

However, the height accuracy achieved by other approaches, compared to this approach, was significantly better. The best height accuracy achieved was 0.10 m, but this was again only for one test site (Zhang *et al.*, 2012; Rottensteiner *et al.*, 2014). Of the methods using all three test sites, the best height accuracy achieved was 0.17 m (Oude Elberink and Vosselman, 2011), whilst the worst accuracy was 0.63 m (Rau and Lin, 2011; Rottensteiner *et al.*, 2014). The proposed method in this research achieved an average height accuracy an order of magnitude worse, at 1.7 m, which shows that further investigations for the extraction of the roof height is required. Errors in the ISPRS benchmark for 3D reconstruction from just imagery were found to be from matching errors in the DSM (Rottensteiner *et al.*, 2014). In the proposed method, these errors will have influenced the refinement of extracted edges, via scan line segmentation, and the determination of the corner's height value. The greater height accuracy of some approaches in the ISPRS Benchmark were also due to the use of lidar data, which as shown in Chapter 3, has a slightly higher height accuracy when compared to dense image matching dataset. The higher accuracy of roof points could be influenced by the simpler and less-challenging roof geometry of Area2, which consisted of large flat roof buildings (Rottensteiner *et al.*, 2014). Reconstruction was not tested for Area2 using the proposed approach because building footprints were not available. Compared to Area2, the roof geometry of Area1 and Area3 was described as challenging, due to irregular distribution of buildings and buildings having small roof structures (Rottensteiner *et al.*, 2014).

As previously discussed, the validation of the planar extraction in this research could not be measured for the Vaihingen dataset, due to the large number of constructed false roof-lines. The results from the ISPRS benchmark showed high success rates of planar extraction with most methods offering an average roof plane quality of 75% or better (Rottensteiner *et al.*, 2014). The most common issue found in the ISPRS benchmark with the extraction of planes was under-segmentation, due to the presences of small roof structures. However, the results using the proposed approach have shown that, for example in Figure 6.2, these small structures can be reconstructed, but tend to create false connections across the main roof planes.

6.7 Summary

Overall the results of 3D reconstruction from Vaihingen were poor. LOD1 models were reconstructed with 66% of corners to a positional accuracy of ±0.8 m. The height accuracy was worse with an accuracy of ±1.72 m. LOD2 models were reconstructed with 75% of corners reconstructed to a positional accuracy of ±0.61 m. Again, the height accuracy was worse with a accuracy of ±1.42 m. Although slightly worse than the results from Newcastle upon Tyne, presented in Chapter 5, the accuracies achieved for the corners reconstructed were within the requirements of LOD2 reconstruction (Gröger and Plümer, 2012). However, the total number of corners extracted was much worse for Vaihingen. On average for LOD1 models an extra 62% of points were constructed, whilst for LOD2 nearly an extra 200% of points were constructed. Whilst a few errors can be attributed to the connectivity ruleset developed, it would appear that most of the errors were caused by the input data and the extracted geometry. As found in many other research projects, if there are errors in the extraction of the modelling cues, then these are present in the reconstructed models (Melnikova and Prandi, 2011; Agugiaro, 2016).

A large number of edges were extracted from the Canny edge detector. This was due to a low maximum threshold value in order to try and maximise the number of correct roof edges extracted. The difference in the colour band combination of the true-orthophotos, when compared to the Newcastle upon Tyne true-orthophoto, meant that greater contrast of edges was seen in the Vaihingen dataset. The low thresholds meant false positives were extracted from shadow, roof texture and other unwanted artefacts. The scan line segmentation approach for refining edges has been shown to successfully remove most of these edges, to leave just the wanted edges along major roof structure lines. However, if a large number of false positives are extracted, then incorrect removal of edges can occur and errors in the reconstructed DSM also prevented incorrect edges from being removed.

When comparing the results of Vaihingen and Newcastle upon Tyne, it can be concluded that the transferability of the methodology between different datasets was unsuccessful. Errors can mainly be attributed to the processing parameters used, but differences between the datasets also had some influence on the results. Both sites constructed too many points as corners, which is attributed to the extraction of edges and the connectivity ruleset. The size of the search window has been shown to have been too large for many buildings. Although different sized windows were used dependent on the length of the roof-line, the size of the building was not considered, which meant the accuracy and percentage of correctly reconstructed corners was greater for larger buildings. The difference in the construction of incorrect corners can also be attributed to difference in the datasets. The different colour-band combinations between the datasets meant a much higher percentage of points were constructed as corners for building in Vaihingen. Errors in the DSM, from image matching procedures, meant edges were not correctly refined and that step edges were more frequently constructed for the Vaihingen dataset, compared to Newcastle upon Tyne. Vegetation was also shown in both datasets to cause over-extraction of edges; thus, further research is required to exclude these areas from reconstruction.

Both sites showed poor accuracy of reconstructed corner height, caused by the ground classification and errors in the DSM. The parameters used for ground classification were not tuned for the nature of the scenes and meant that areas around buildings were incorrectly classified. In general, the height accuracy of reconstructed corners from buildings in Vaihingen was worse than those from Newcastle upon Tyne, which could be the result of errors in the DSM (Rottensteiner *et al.*, 2014).

In conclusion, although many of the errors were the results of incorrect processing parameters, differences in the data and the geometry of the buildings have also been shown to cause errors in the reconstruction results. Therefore, more research is required to improve the methodology and transferability of automatic 3D building reconstruction from dense image matching datasets.

Chapter 7. Conclusion and Recommendations

7.1 Summary of Findings

This thesis has presented a new approach for automatic 3D reconstruction of buildings from aerial imagery based on a developed geometry extraction approach and connectivity ruleset. Edge information was extracted from true-orthophotos using a feature-based approach, which can be sensitive to the parameter settings and the imagery. Previous approaches have used criterion such as length to remove edges. However, these criteria can be detrimental to the reconstruction of small roof geometry such as dormer windows. The approach utilised in this research for edge detection refinement was developed upon the theory of segmentation using cross-sections; by incorporating heights from the dense datasets. Heights were applied to each detected edge, and edges were removed if the fit to a fitted least squares linear regression line was below a predefined threshold, which incorporated changes in slope along cross-sections of the roof planes. This approach showed how the edges could be refined to remove false positives from roof planes, whilst keeping edges at the boundaries of roof planes. The results also showed how edges along dormer windows and other small roof features were correctly extracted, which showed the potential of this method to extract small roof geometry, an area in which previous research has struggled. The extracted edges were vectorised into lines to reconstruct a 3D building as a network of corners, connected by roof-lines, which delineated roof planes.

The developed approach has again highlighted the successful reconstruction of 3D building models by using constraints for roof geometry, as was concluded by Nex and Remondino (2012) and other researchers, presented in Chapter 2. Whilst previous approaches have used strict constraints to reconstruct geometry with corners at 90° and planes with an area greater than a predefined threshold, this approach has extended the constraints to increase the complexity of the roof geometry reconstructed. Roof corners were not limited to orthogonal angles, short lines were reconstructed and new constraints such as roof-line intersection were proposed for the reconstruction of roof geometry. The complexity of reconstructed roof geometry was also improved due to the inclusion of adaptive thresholds based on the length of a roof-line. Stricter constraints were applied to shorter lines, compared to longer lines, in order to reduce the influence of roof geometry reconstruction, but also to prevent them incorrectly connecting to other extracted endpoints.

The methodology has been tested on two areas, captured with different camera systems that offered variations in the geometry of the roof structure. The results showed that corners were reconstructed to a greater accuracy than listed in the OGC CityGML standards, with approximately 75% of corners extracted. Whilst the methodology was largely considered successful on the first test site, the results were not as promising when the methodology was transferred to the second test site. This was due to some of the parameter settings identified throughout Chapter 5 and Chapter 6: the extraction of roof geometry information and the size of the search window used to connect roof-lines to other roof-lines, and to boundary corners, but also due to differences between the two datasets.

Errors in the ground classification of the image-based point cloud were transferred to the reconstruction of block model and building models with roof geometry. These classification errors were believed to derive from not classifying vegetation, errors in the building footprint and possibly the parameter settings used. These caused over-segmentation of the nDSM boundary and resulted in increased extraction of corners from the boundary of the building, producing irregular geometry. The low parameter settings of the Canny edge detector meant that many false positives were extracted. The difference in the colour-band combinations of the true-orthophotos between the two test sites resulted in a large difference of edges extracted, which led to a large difference in the number of points constructed as corners. The developed procedure for the refinement of detected edges was successful in removing most of the false positives along roof planes. However, edges detected at features that did not require reconstruction, such as chimneys and vegetation, were not removed due to the difference in height.

The second main error source was the size of the windows for connecting roof-lines. Errors in the script, which did not consider the GSD of the imagery, meant that different search window sizes were used for the two test sites. Although this may have caused some differences in the success of reconstruction, errors were generated from the size of the search windows being too large for both study areas. The size of the search windows performed more successful reconstruction for larger, industrial buildings compared to smaller, residential areas. By taking into account the size of the building footprint, incorrect connectivity of roof-lines across roof planes may be reduced.

Although errors have been noted, the developed methodology has been shown to successfully reconstruct building models at two levels of detail, by extracting roof lines and corners. Corners of the roof were extracted to a high positional accuracy, but errors were seen in the extracted heights of the corners. Corners were extracted within the positional accuracy of CityGML LOD2 building models, and just outside the requirements for LOD3 (Gröger and Plümer, 2012). Although the height accuracy of the corners was not as accurate as the positional accuracy, heights were extracted within the specifications of both LOD1 and LOD2 reconstruction (Gröger and Plümer, 2012). However, not all connecting edges were extracted, which resulted in generally unsuccessful extraction of roof planes. Two errors were present in the reconstruction of roof planes. Missing edges and the incorrect connection of roof-lines across roof planes meant that some planes were not reconstructed or were incorrectly segmented. This can be explained by the size of the search window being too large for some buildings, as previously discussed.

The workflow has shown that by only applying constraints to the connected line, planar extraction was not successful and that further constraints should also be included for the planar geometry. Currently, only the geometry of the lines is considered: whether the line should extend to the corner/endpoint of another roof-line, or whether two lines in close proximity should connect, all in 2D. More constraints can be applied to the line geometry, to include height information, as well as apply planar geometry constraints such as area, slope and pitch. However as shown in Chapter 2, some constraints such as planar area can limit the range and complexity of roof shapes reconstructed. Other constraints could also be applied to the boundary corners of the reconstructed buildings, such as measuring the internal angle of corners. By including more constraints, this should therefore aid reconstruction and improve planar extraction. Further constraints to consider will be explored in Section 7.4.

7.2 Evaluation of Aims and Objectives

The aim of this research was to produce a novel method for the automatic reconstruction of building models from dense image-matching datasets at the LOD1 and LOD2 specifications, as defined by the OGC CityGML standard.

In order to achieve this aim, the following objectives were defined;

i. Investigate the current state-of-the-art methodology in building modelling and rooftop extraction from aerial photography and lidar data, examining levels of automation as well as the relative strengths and weaknesses of different approaches;

ii. Based on objective i, propose and develop an automated procedure for the extraction of 3D rooftop vertices from dense image matching datasets;

iii. Integrate the information extracted from the procedure developed in objective ii in order to produce 3D building models;

iv. Validate and refine the methodology proposed in objectives ii and iii, delivering appropriate metrics on the accuracy and reliability of the developed approach.

In reference to the aims and objectives, it can be determined that the aim of this research has been achieved. An approach to automate reconstruction of building models at the specification of LOD1 and LOD2 models has been successfully formulated, but outputs show somewhat unsatisfactory reconstruction of the building geometry, due to reasons previously explained. The approach was developed based on the results of the first objective to critique the current state-of-the-art methods, reported in Chapter 2. The strengths of aerial imagery, presented in Chapter 3, were used to extracted roof geometry as part of the second objective, and the current limitations of area-based methods, which are predominately used with lidar point cloud data, and generally lead to under-segmentation of the extracted roof planes, were avoided.

The extraction of roof geometry, as part of the second objective, was shown to be sensitive to noise and the parameters of the edge detector. Edges were extracted in 2D, so in order to extract 3D roof geometry, datasets were combined together which allowed the refinement of the extracted geometry. The refinement of the detected edges has shown how an area-based method, which was previously proven unsuccessful for roof modelling cue extraction, can be utilised together with a feature-based extraction method to overcome the limitations of both methods. Information from the true-orthophoto was successfully combined with the DSM to use 3D information to refine 2D feature extraction. The sensitivity threshold test in Chapter 4 showed how the parameters of the edge detector were determined, and how the edges were combined with height information to further refine the edges.

The extracted roof geometry was integrated and refined, as part of the third objective, to reconstruct 3D building models. The developed connectivity rulesets showed how the closed network of roof-lines can be formulated to create individual roof planes. The constraints enforced successfully connected roof-lines to other roof-lines, as well as to boundary corners, but also caused several false connections to be constructed. The refinement and connectivity of the edges and corners were presented in Section 4.6 for LOD1 models and Section 4.7 for LOD2 models.

As part of the final objective, the reconstructed models were quantified against reference data to determine the success of the methodology. Chapter 5 and Chapter 6 show the results of the methodology on the two study areas. The results showed accurate corner extraction, with regards to the outlined specification, but too many points were constructed as corners, which hindered the reconstruction of roof planes.

7.3 The Applicability of Dense Image Matching Datasets for 3D Reconstruction

The proposed workflow has highlighted the strengths of the dense image matching datasets, which include the spatial resolution of the imagery and the DSM.

The high spatial resolution of the captured imagery meant edges were successfully extracted at the boundaries of roof planes and along major roof-lines. However, the high spatial resolution also meant that many false positives were extracted from very fine roof detail, including roof tiles and texture, that was not required for reconstruction. The positional extraction of roof edges was helped by the reconstruction of the true-orthophoto. The density of the DSM meant that all pixels were orthorectified, to remove radial distortion and prevent edges being extracted on the façade of the building.

The DSM, reconstructed from image-matching, produced a dense grid of height values, which allowed the successful refinement of detected Canny edges. Heights were applied to detected Canny edges, which allowed a least squares linear regression line to be fitted to detected edges, successfully removing edges detected on the roof planes. The sharp edges at height discontinuities allowed the successful reconstruction of step edges along roof boundaries. However, the heights at corners were not determined as accurately as the planimetric position. Errors in the matching of images for the creation of the DSM may have caused errors in the height extraction of corners. This was noted in building reconstruction from imagery for the Vaihingen dataset of the ISPRS benchmark (Rottensteiner *et al.*, 2014). Whilst the heights at the corners was not as accurate as the planimetric accuracy, heights assigned using the developed method were shown to be more accurate when compared against methods from the ISPRS benchmark that used imagery for reconstruction. This increase in the vertical accuracy of corners shows the potential of the proposed method and dense image matching dataset to extract heights, which can rival those of lidar surfaces. Approaches for pixel-to-pixel matching have been developed further by software vendors and academia, which will produce higher accuracy datasets from imagery, which should therefore be utilised for 3D reconstruction.

As shown in the ISPRS benchmark for 3D reconstruction, further research is still required to determine the true potential of dense image matching for 3D reconstruction. This research has shown how corner of buildings and roof planes can be extracted to a high accuracy and that a wide range of building roof complexities can be reconstructed. The point cloud created from image-matching in this research was subject to noise, due to all matched rays between stereo-pairs of images being reconstructed for Newcastle upon Tyne, and no image-based point cloud being available for the Vaihingen dataset. Whilst other software packages were tested to reconstruct an image-based point cloud, the same accuracy of the DSM could not be achieved. The provided image-based point cloud was not used for testing reconstruction, and a 2.5D point cloud was instead reconstructed from the DSM for both datasets. The potential of image-based point clouds could therefore not be investigated. By using a correctly matched image-based point cloud, this could improve ground classification, aid the removal of vegetation by combining spectral information to each point and be used more in the reconstruction stage of the methodology.

7.4 Recommendations for Future Research

There are a number of potential future directions for this research to be continued. These include further expansion of the methodology and the investigation of new datasets. As previously discussed, several issues with the current implementation, including the extraction of modelling cues from the data and the size of the search window, should be refined. By improving the extraction of modelling cues, correcting errors in the ground classification, removing vegetation and reducing the number of edges detected, fewer errors should be reconstructed. Also, altering the size of the search windows should reduce the construction of false roof-lines. Further enhancements to the workflow, as well as developments should be undertaken to improve the reconstruction results.

The refinement of the building boundaries has been shown to be unsuccessful by simply measuring the orientation of roof edges. Examples of reconstruction can be seen in Chapter 5 and Chapter 6 where acute and irregular angles have been reconstructed for roofs. The edges, which resulted from these corners, had similar orientation and were within the thresholds. By measuring the interior angle between boundary corners and including an interior angle constraint, the over-segmentation of the building boundary should be reduced by removing any corner point with an irregular angle, which could produce irregular planar geometry.

Whilst suggestions have been made to correct the errors currently in the workflow, there are a number of developments that could be undertaken for the connectivity ruleset:

- Currently, the ruleset connects an unconnected endpoint to the first corner that matches the criteria. Endpoints within the defined neighbourhood were ranked based on the distance from the unconnected endpoint. By developing the ruleset further, a multi-criteria evaluation approach could be implemented. This could test all endpoints found within a search window for a number of criteria: angle, line orientation, distance and possible other constraints. Endpoints could then be ranked based on the criteria to find the best connecting endpoint. This would remove any noise bias found close to the endpoint.

- The ruleset could be developed to include the heights of the corners. By testing changes in slope of an edge, this could prevent false connections being reconstructed. An example of this may be for the connection of dormer window roof-lines. In some cases, reported in Chapter 5, the ridgelines along the top of the dormer window were extended to connect to the ridgeline of the main roof structure. Considering the height of the endpoints would represent a change in the slope of the line; the endpoints of the ridgeline along the top of the dormer window should have similarly height values with a flat slope. By extending this line to the ridgeline of the main roof structure, which is generally at a higher height, would change the slope of the dormer windows ridgeline. A change in slope threshold could therefore be introduced. This would mean that the connection would be tested in 3D instead of only 2D.
- Each rule of the connectivity ruleset was developed in isolation from other rulesets having different parameter settings. The results have shown how a connectivity ruleset of roof-lines can successfully reconstruct roof geometry and 3D building models. However, further refinement of the parameters and the rules may enhance the reconstruction results. By testing the rulesets together, a common threshold value may be found, which proves to be more successful.
- This research has only considered the reconstruction of the main roof geometry, without considering finer details such as dormer windows. The sensitivity testing of parameters for the Canny edge detector and scan line segmentation, presented in Chapter 4, were refined to allow the extraction of roof geometry and small features. However, the aim of the research was to reconstruct the main roof face. Therefore, further research into the reconstruction of dormer windows should be investigated, once other issues have been resolved.

The reconstruction of step edges along extracted roof-lines also needs to be developed. The proposed workflow for the extraction of step edges along the roof boundary has been shown to successfully reconstruct the difference in height at the edge. In some cases, this created additional edges, for example on the façade between multi-levels of a flat roof, which do not exist in reality. The workflow needs to be developed to separate the two roof levels into separate but connected models. To construct step edges, the height difference between connected boundary roof corners was measured. However, this could not be used for step edge reconstruction from roof-lines because sloped edges, for example a gable line between a boundary corner and a ridgeline would be incorrectly reconstructed as a step edge. An approach similar to the extraction of height values could be developed. The numerical gradient of height values surrounding a roof-line corner could be measured to determine a steep gradient and a potential step edge. If the gradient was above a threshold, then a step edge may be defined.

The reconstructed roof model could be merged with the building footprint to reconstruct LOD2 models with roof overhang. Currently, the reconstructed models presume that the edge of the roof is the edge of the building, so the façade is modelled from the edge of the roof to the ground. By combining the OS MM building footprints, which models the outline of the building, with the reconstructed roof model, a 3D building model could be reconstructed. This was investigated and, for a small sample of roofs, successfully reconstructed the roof overhang. However, there were disagreements between the boundaries of most building footprints and the reconstructed roof model, particularly where small outlets were not reconstructed. This meant that roof overhang was not correctly reconstructed. If the number of boundary corners can be correctly reconstructed, and with an accurate height value, then building footprints and roof models could be reconstructed together to more adequately represent the roof overhang.

As well as refining the methodology, new sensors and emerging technology should be investigated and considered. The spatial resolution of sensors is continuously improving as CCD and CMOS technology improves, for example the Leica DMC III aerial camera is capable of capturing images at 3 cm spatial resolution (Chen *et al.*, 2016; Leica Geosystems, 2016c). A move towards multi-view configurations has been noticed within the photogrammetric field as the need for better object space observations grows (Chen *et al.*, 2016; Toth and Jóźków, 2016). Whilst oblique imagery is not a new concept, the development of camera systems has allowed high spatial resolution imagery to be captured from various viewing angles to captures all sides of a building. The increase in data captured means façade geometry and texture mapping can be easily reconstructed with oblique imagery (Xiao *et al.*, 2012). This may potentially allow the development to LOD3 building models, where façade details are modelled.

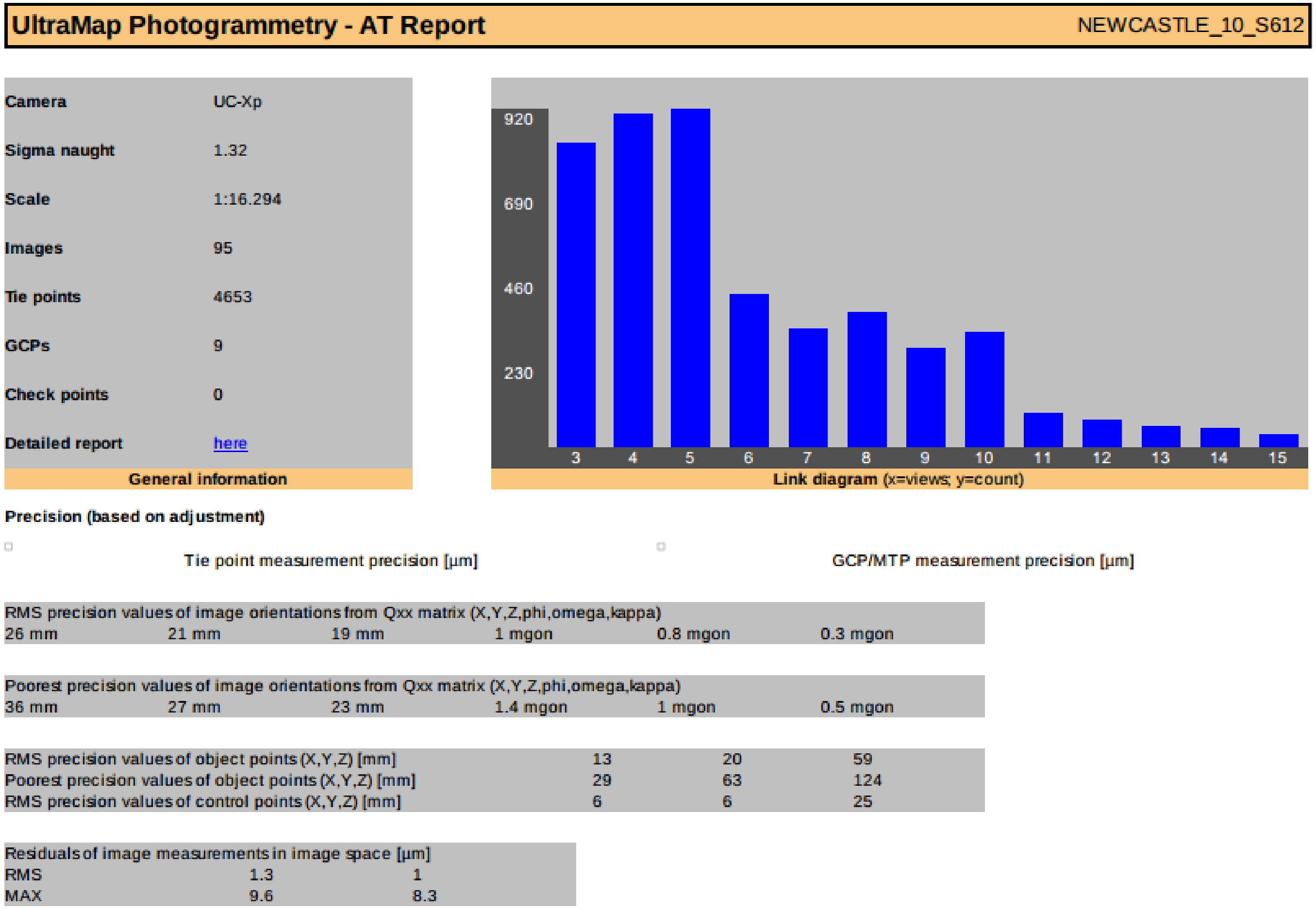
Satellite imagery is also improving. In the past, researchers had preferred not to use satellite imagery due to its poorer spatial resolution. However, this has recently improved, currently up to 0.31 m for WorldView 3 (Chen *et al.*, 2016). The high temporal resolution of the data also means that if a successful 3D reconstruction method can be developed from this data, then applications such as 3D change detection and the rapid update of 3D models can become possible. Hyperspectral imagery is also a growing area which can be used to generate 3D data (Chen *et al.*, 2016).

The development and availability of Unmanned Aerial Systems (UAS) has also grown over the last five years. These platforms currently offer high temporal and spatial resolution imagery, which can be matched together into a dense image matching datasets. Due to regulations, these systems can still not be flown over built-up areas, but have been used for the capture and reconstruction of single complex buildings (Jazayeri *et al.*, 2014).

As well as optical sensors developing, the technology and configurations of active systems are also progressing. Lidar sensors are being duplicated on platforms to increase the pulse rate, as well as currently investigating different parts of the spectrum, to hopefully introduce full colourised point clouds from a lidar system (Toth and Jóźków, 2016). Therefore, as well as enhancing the proposed methodology, new sensors and datasets should be considered for the applicability of 3D scene reconstruction.

Appendix A. Aerial Triangulation Report for Newcastle upon Tyne Imagery

This aerial triangulation report is for the UltraCam Xp imagery, captured over Newcastle upon Tyne and processed using UltraMap v3. The RMSE height accuracy of object points was 59 mm, with the poorest height accuracy to be 124 mm.



UltraMap Photogrammetry - AT Report NEWCASTLE_10_S612

Camera	UC-Xp
Sigma naught	1.32
Scale	1:16.294
Images	95
Tie points	4653
GCPs	9
Check points	0
Detailed report	here

General information

Link diagram (x=views; y=count)

Precision (based on adjustment)

Tie point measurement precision [µm]

GCP/MTP measurement precision [µm]

RMS precision values of image orientations from Qxx matrix (X,Y,Z,phi,omega,kappa)

26 mm	21 mm	19 mm	1 mgon	0.8 mgon	0.3 mgon

Poorest precision values of image orientations from Qxx matrix (X,Y,Z,phi,omega,kappa)

36 mm	27 mm	23 mm	1.4 mgon	1 mgon	0.5 mgon

RMS precision values of object points (X,Y,Z) [mm]	13	20	59
Poorest precision values of object points (X,Y,Z) [mm]	29	63	124
RMS precision values of control points (X,Y,Z) [mm]	6	6	25

Residuals of image measurements in image space [µm]

RMS	1.3	1
MAX	9.6	8.3

Appendix B. Boundary Parameters

B.1 Boundary Orientation Threshold Sensitivity

The four plots below represent the results of boundary orientation threshold sensitivity testing for four different buildings. To overcome issues previously experienced using only orthogonal angles, the main orientation of a roof was calculated as the orientation of the longest length. Orientation intervals of ±45° from the main orientation were then calculated for edge refinement. Threshold sensitivity was performed to determine the orientation tolerance to accept or remove an edge.

In the plots below, a blue dot represents a line of the building boundary, which is located against the length and orientation. The red vertical lines represent the threshold sensitivity windows from the main calculated orientation and intervals from this of ±45°, with thresholds of ±2.5° to ±10°, in intervals of 2.5°, tested.

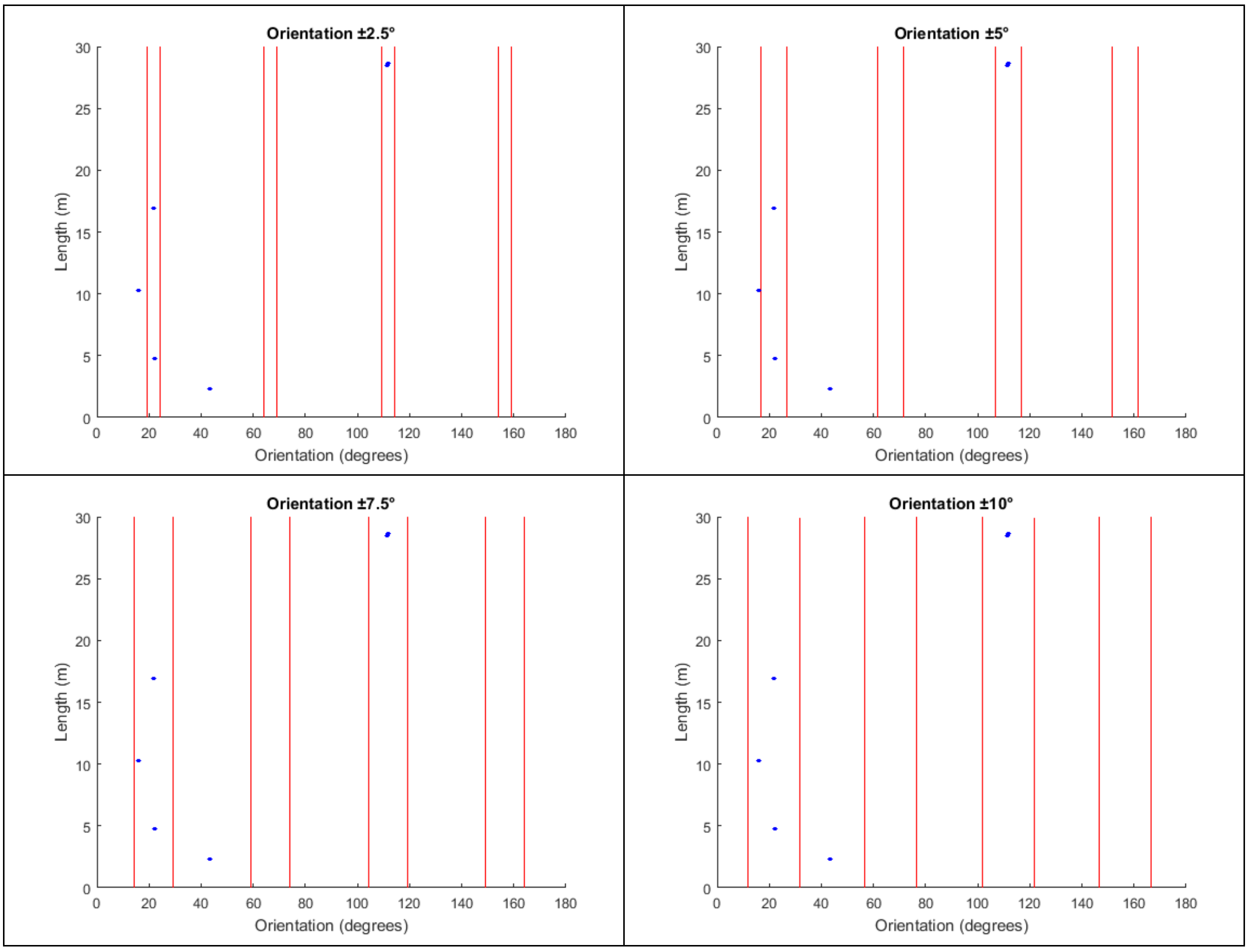


Figure B.1 – Example plots of orientation threshold sensitivity testing for a selected building.

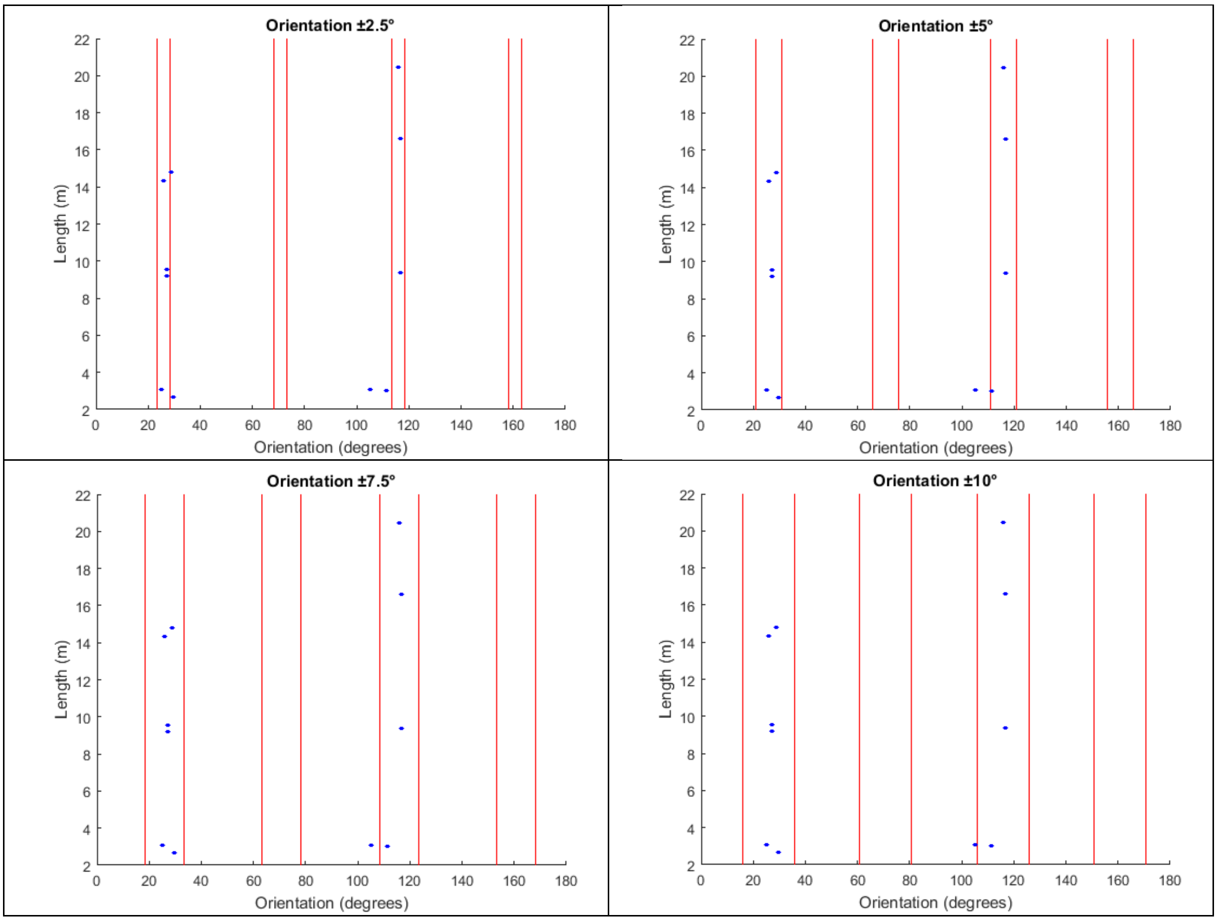


Figure B.2 - Example plots of orientation threshold sensitivity testing for a selected building.

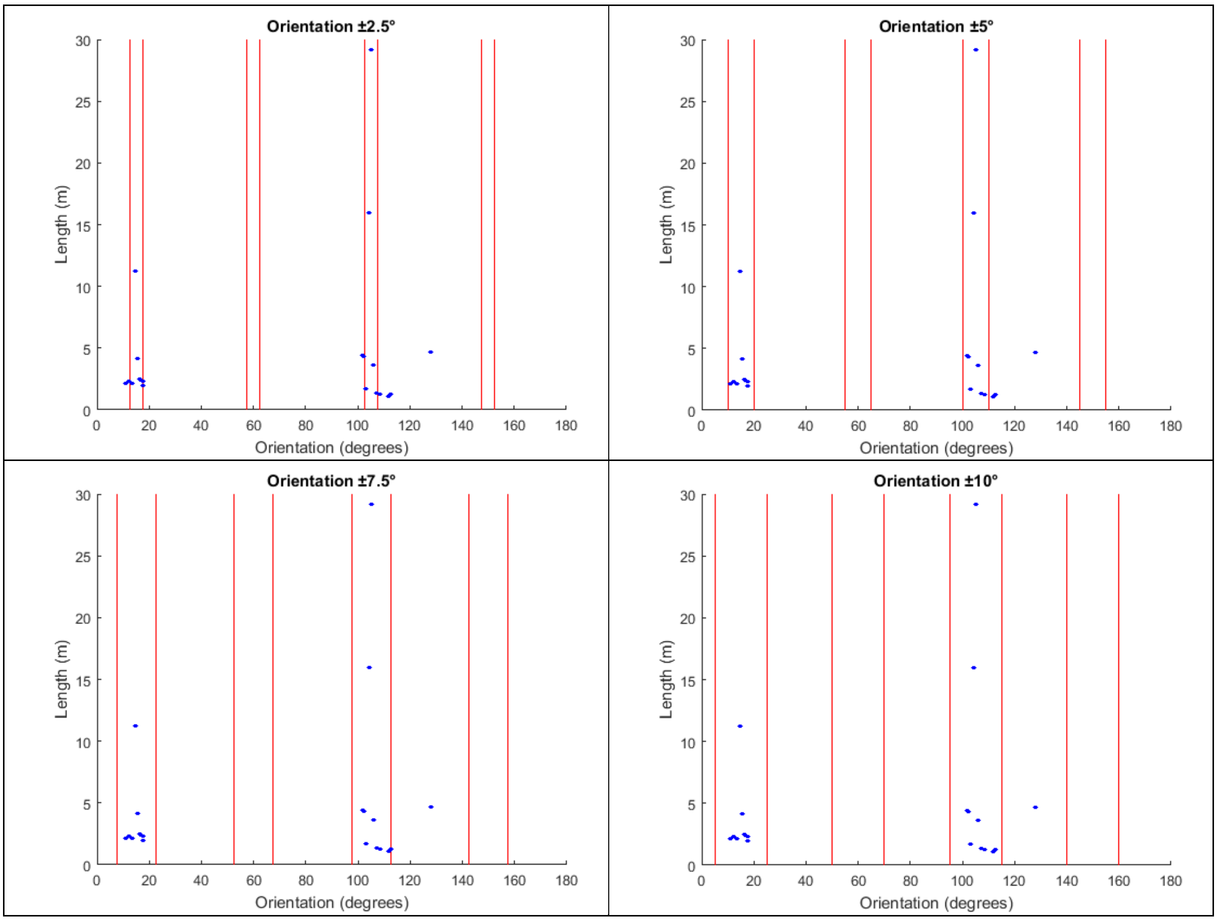


Figure B.3 - Example plots of orientation threshold sensitivity testing for a selected building.

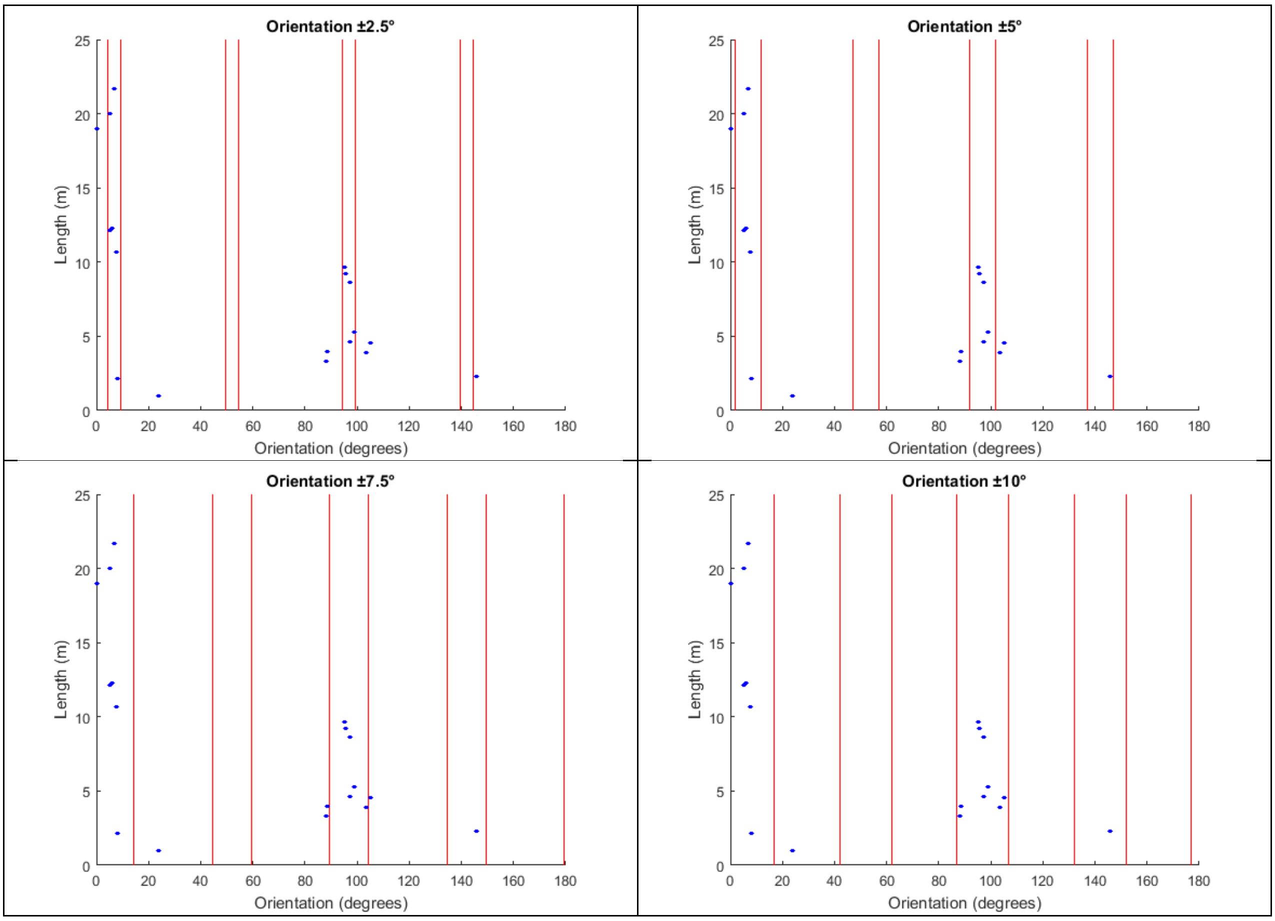
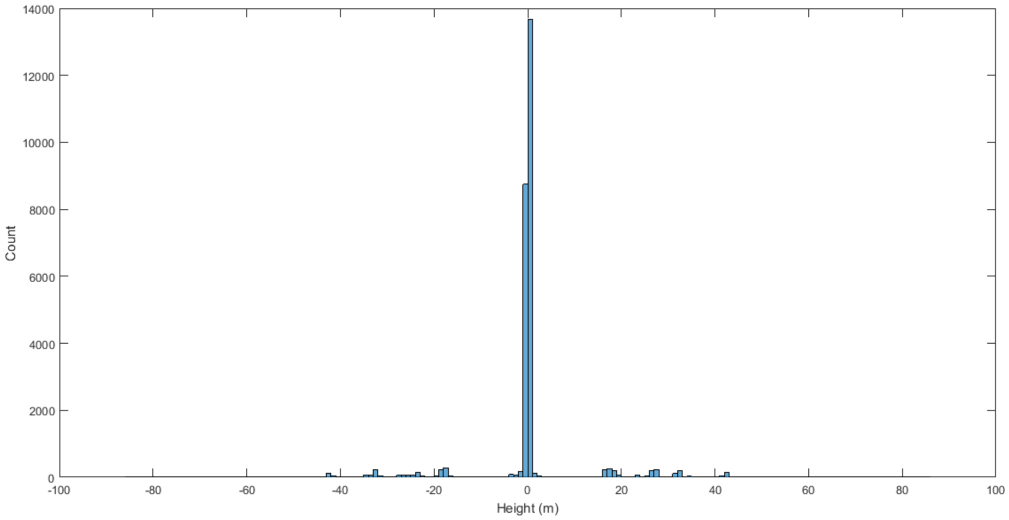


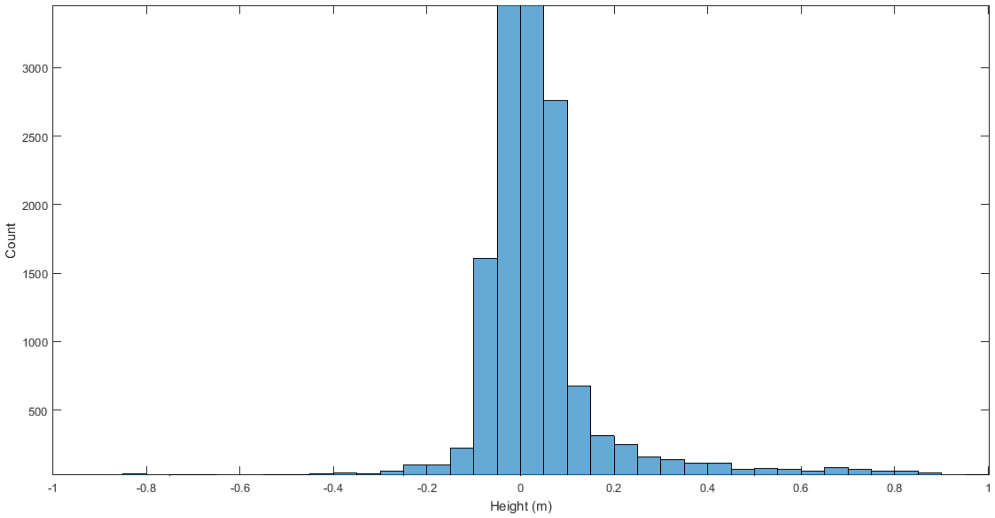
Figure B.4 - Example plots of orientation threshold sensitivity testing for a selected building.

B.2 Height Thresholds

The histograms in Figure B.5 show the accumulative height gradients calculated of all test site boundary corners when assigning a height value to a corner. As can be seen in Figure B.5a, some of the height gradients were very large, which represent the extent of the nDSM. These were excluded using a maximum threshold of 5 m when assigning height values. The zoomed in section of the histogram in Figure B.5b shows a steep bell curved shaped, with the height gradients tailing off at approximately ±0.2 m.



a)



b)

Figure B.5 – Height threshold sensitivity: (a) all height values, and (b) section from Figure B.1a zoomed into the middle of the X axis.

Appendix C. Roof-line Parameters

C.1 Orientation and Length

Similar to the roof boundary, as shown in Appendix B.1, threshold sensitivity testing was also undertaken on the roof-lines; to remove edges shorter than a specific length, that were not within a threshold of the main orientation, and at 45° intervals from this. The four plots below show the results for four different building structures, where each blue point represents a line, plotted at its length and orientation. The red vertical lines are the orientation thresholds. As explained in Appendix B.1 the main orientation was calculated from the longest roof-line, with roof-lines at ±45° intervals from the main orientations being classified as correctly extracted. To allow a tolerance from these intervals, thresholds of ±2.5° up to ±10°, in intervals of 2.5° were tested. The green horizontal lines represent the length thresholds tested. Lengths of 5% and 10% of the longest roof-line were tested as threshold parameters.

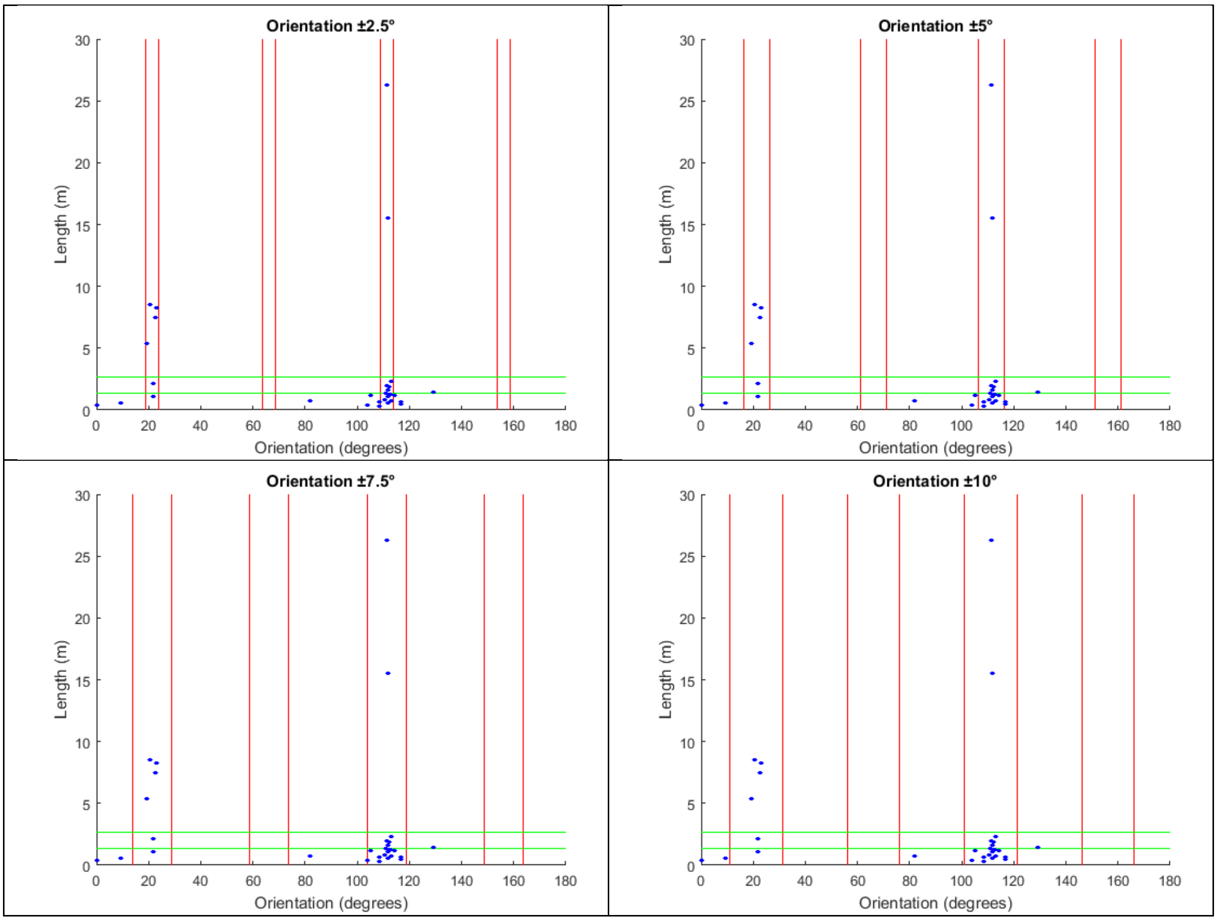


Figure C.1 - Example plots of orientation and length threshold sensitivity testing for a selected building.

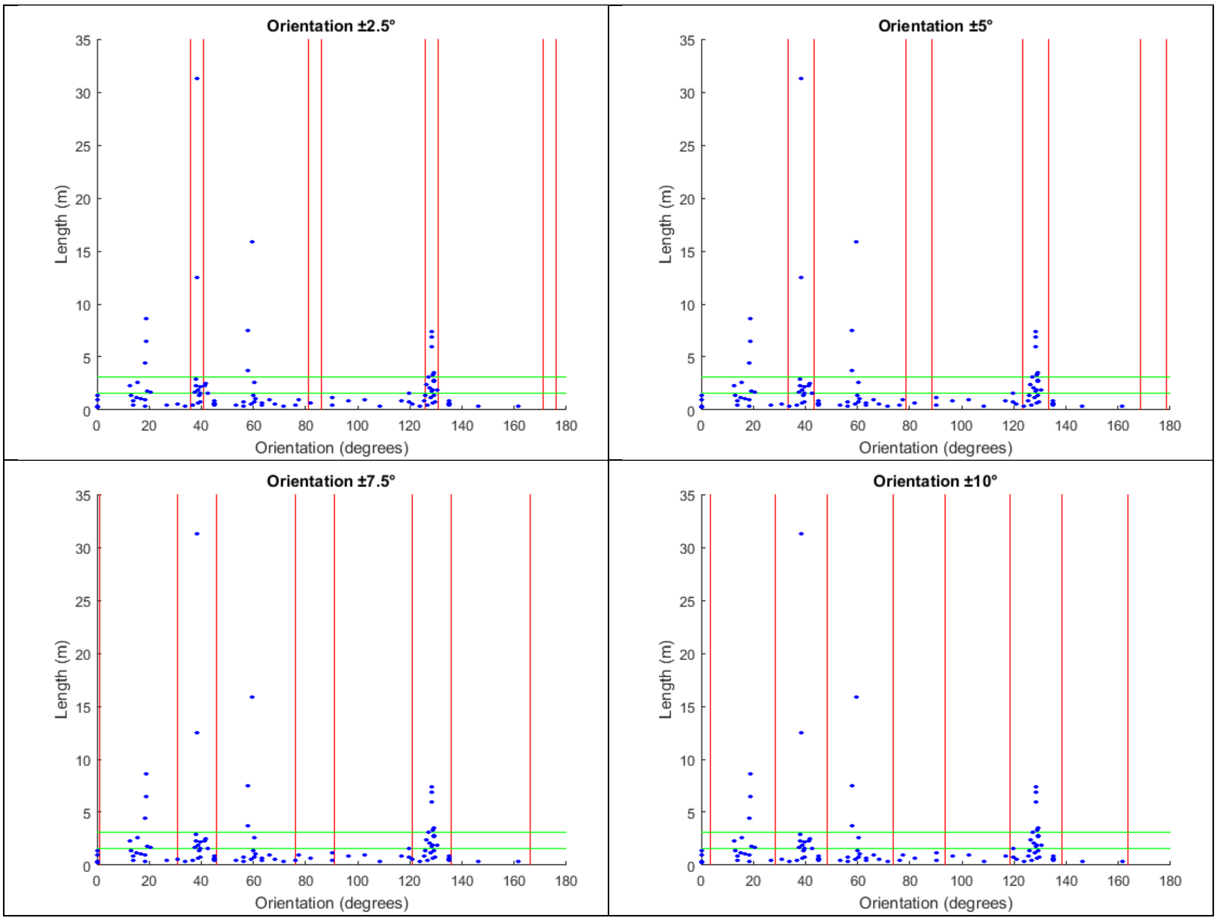


Figure C.2 - Example plots of orientation and length threshold sensitivity testing for a selected building.

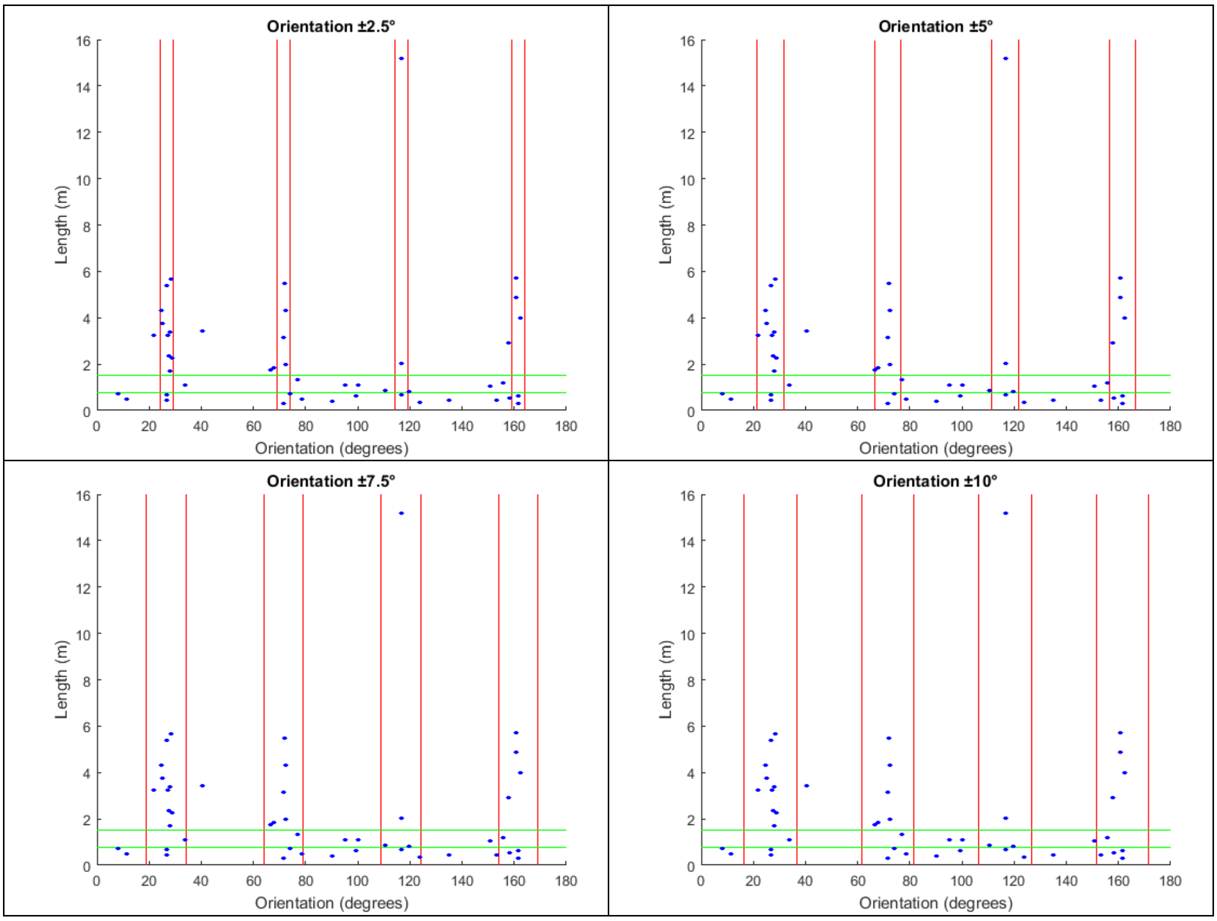


Figure C.3 - Example plots of orientation and length threshold sensitivity testing for a selected building.

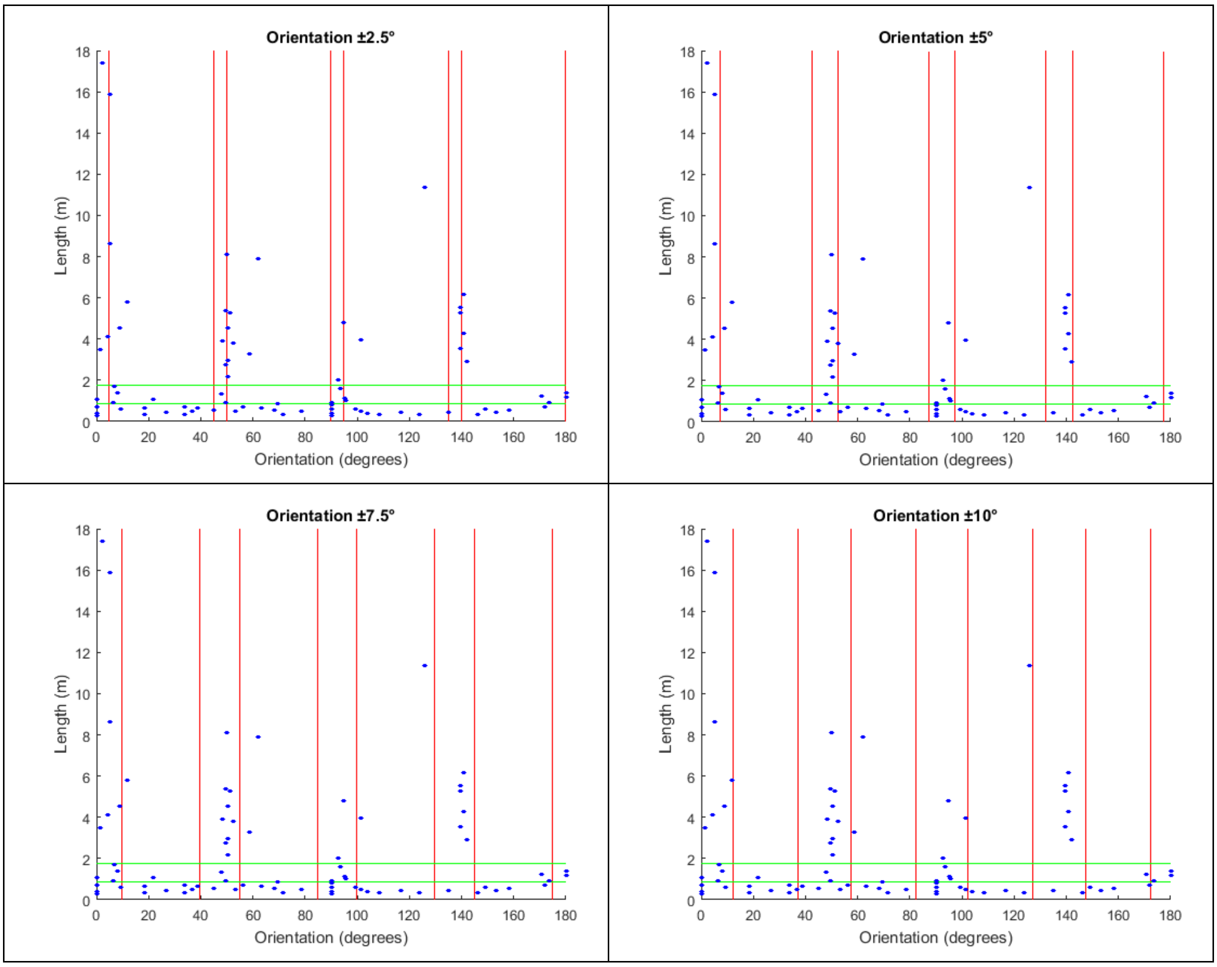


Figure C.4 - Example plots of orientation and length threshold sensitivity testing for a selected building.

C.2 Angular Threshold

To extent a roof-line and connect it to the correct corner position, the angle from an unconnected endpoint to a corner was measured. Ideally the angle at the unconnected endpoint, from the opposing endpoint of the roof-line, to the potentially connecting endpoint should be as close as possible to 180°. A tolerance was applied to this angle to ensure the correct connections were made. If the tolerance was too large, false connections between roof-lines may be constructed, whereas if the tolerance was too strict then correct connections may not have been constructed. The bar chart in Figure C.1 is a histogram of measured angles from unconnected endpoints to potential connecting endpoints.

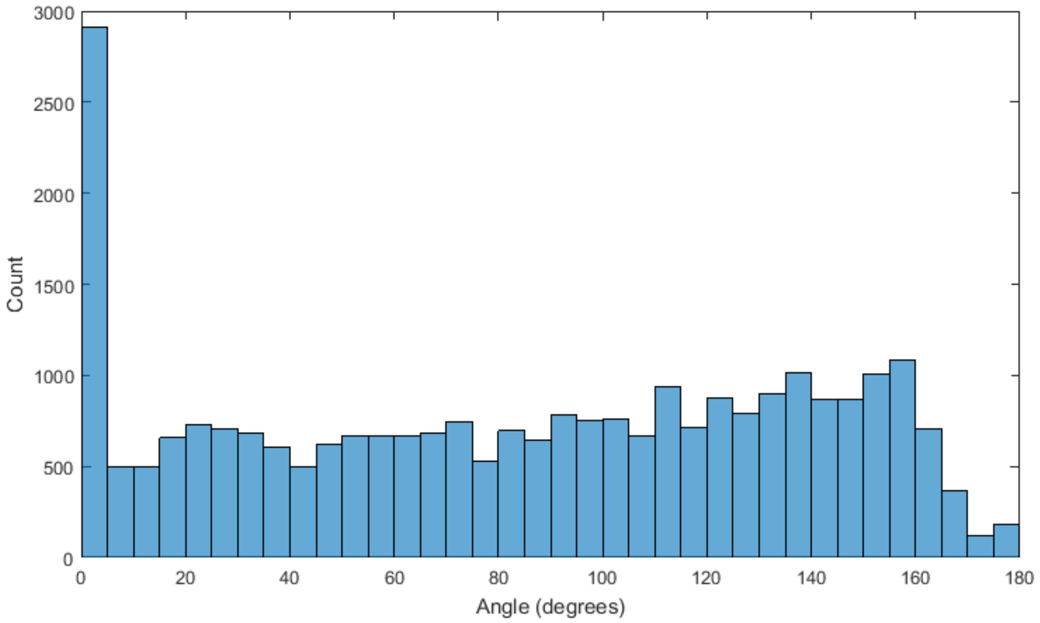


Figure C.5 – Bar chart of accumulative angle measurements

C.3 Distance Threshold

A distance threshold was also applied between endpoints to remove roof-lines shorter than a specific distance. Table C.1 shows some example results of connecting endpoints with a distance of either 0.5 m or 1 m.

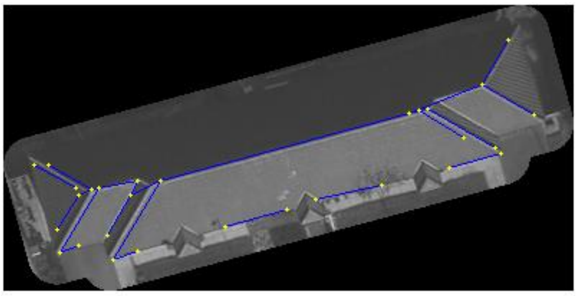
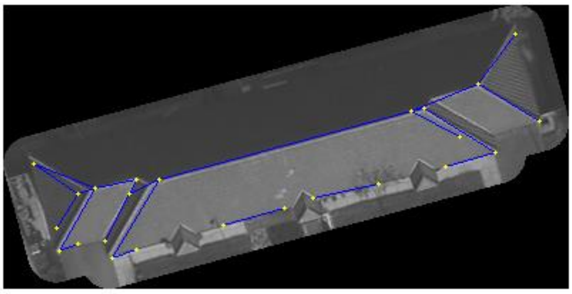
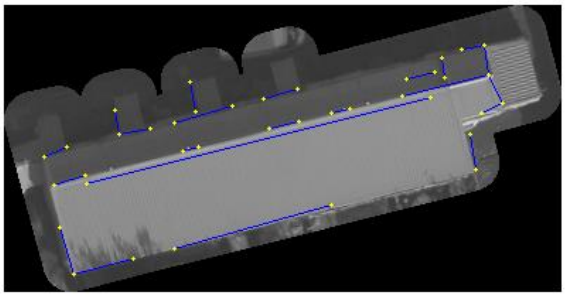
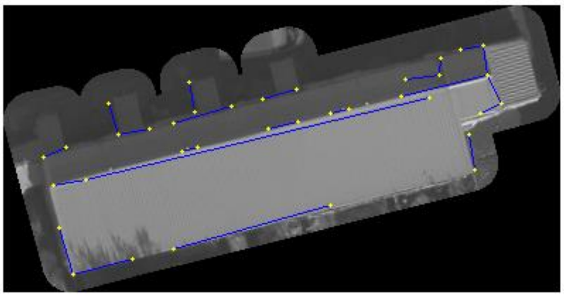
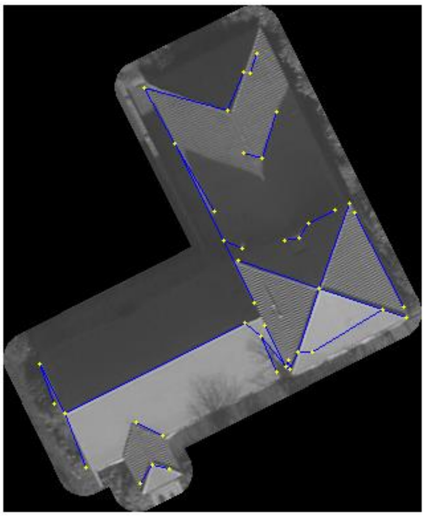
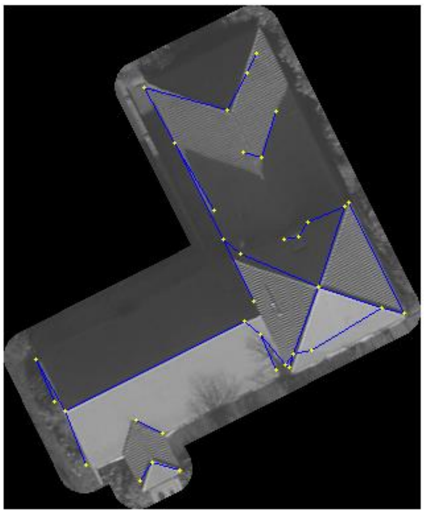
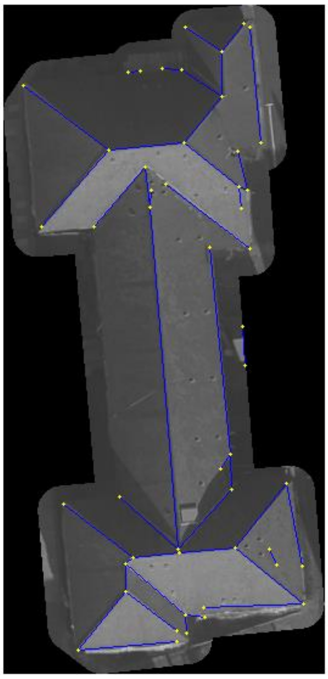
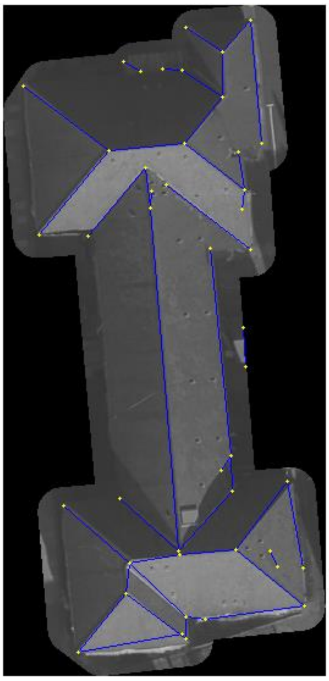
0.5m	1m
	
	
	
	

Table C.1 – Results of distance threshold sensitivity for connecting roof-lines.

Appendix D. Reconstruction by Roof Type

D.1 Flat Roof Reconstruction

A flat roof consisted of a horizontal roof plane with the height of all corners being equal. Variations of the flat roof, including a multi-level flat roof and a shed roof, which has a sloped roof plane with one edge of the roof being higher than the opposing edge, were also included in this analysis.

The results of flat roof reconstruction showed that the positional accuracy RMSE of extracted corners was within ±0.5 m of the reference data, with the height accuracy RMSE slightly worse at approximately ±0.75 m. Overall reconstruction was successful, with nearly 80% of corners extracted for LOD1, and 85% of roof corners extracted for LOD2 reconstruction. The increase in corners extracted for LOD2 reconstruction was due to the presence of step edges in some test sites. A step edge is not reconstructed at LOD1, but should be reconstructed for LOD2 (Gröger and Plümer, 2012). A total of 11 step edges were present in the 10 test sites (noted by the change in the total number of corners between Table 5.1 and Table 5.2), with several of them being reconstructed and yielding a higher percentage of points detected for LOD2, compared to LOD1. However, there was over twice the number of points reconstructed for LOD2 compared to LOD1, which was a result of the extracted lines from the Canny edge detector.

Figure D.1 shows an example of 3D reconstruction for a shed roof. It can be seen that the surface has been correctly reconstructed, with one side of the roof higher than the other. The figure shows the incorrect construction of points along the eave line of the roof; with points extracted from vegetation and a nearby building. In this example, the nDSM has been under-segmented; a small extension to the building has not been reconstructed. The six corners of the largest roof faces were reconstructed within a positional accuracy of ±0.5 m, when compared to reference data. However, for LOD2 reconstruction an extra roof-line was extracted, separating the large and the small roof planes, shown in Figure D.1.

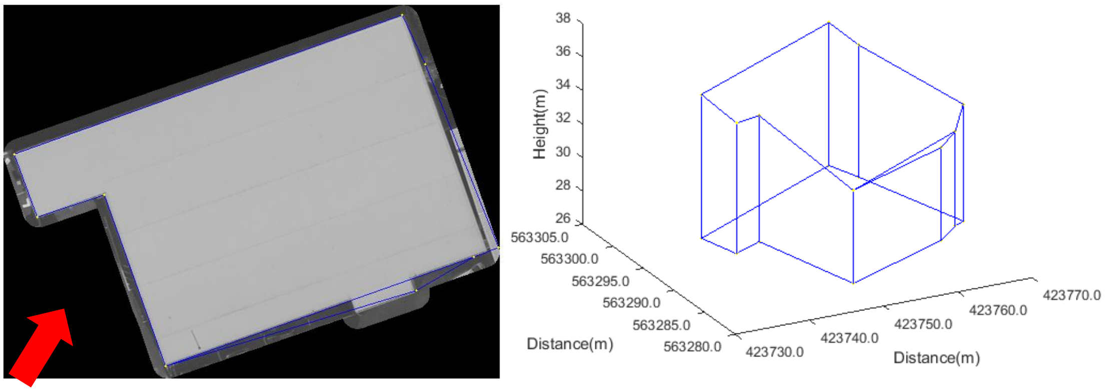


Figure D.1 – 3D reconstruction of a shed roof.

Another example of 3D reconstruction for flat roofs can be seen in Figure D.2, where a flat roof, split over two levels, has been reconstructed. All the boundary corners and roof-line corners were successfully extracted to a positional accuracy of ±0.23 m and a height accuracy of ±0.25 m. However, an extra two points were extracted from shadow between the roof levels, which, due to the difference in slope, could not be removed using scan line segmentation. A sloped roof plane was, therefore, reconstructed between the two height levels. In this example, the step edges were not reconstructed because the height difference between the roof planes along the boundary was below the height threshold. Also, no step edges were reconstructed for the extracted roof-lines, a current limitation of the workflow.

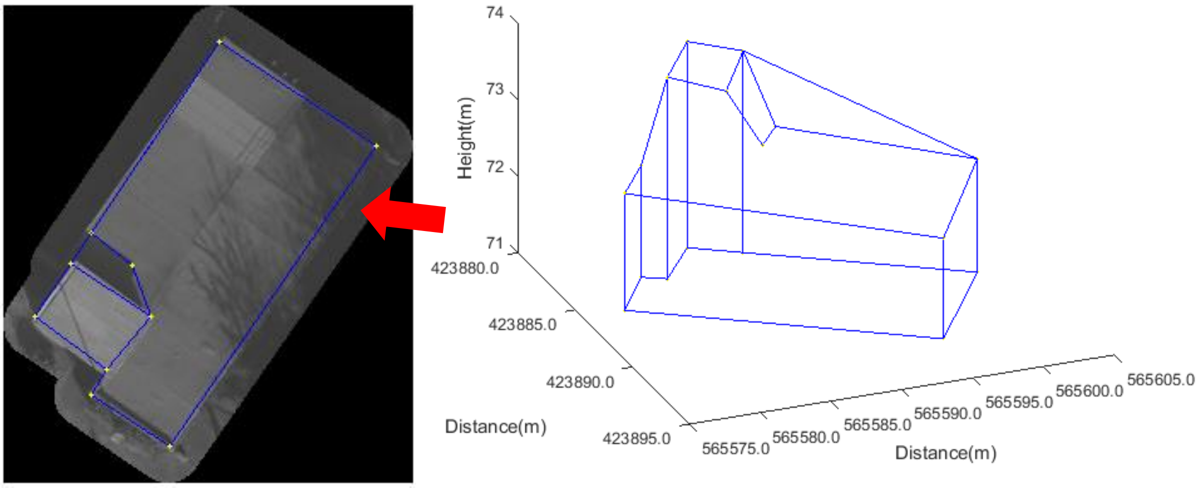


Figure D.2 – 3D reconstruction of a flat roof on two levels separated by a step edge.

As expected with flat roofs, the planar extraction was very successful with a quality percentage of 80%, which rose to 85% when only considering planes larger than 10 m², as shown in Table 5.3. Most flat roofs only have one plane, which were governed by the nDSM boundary, with an example shown in Figure D.1. The extracted roof surface in this example had a quality percentage of 95%, which was hindered by the incorrect extraction of a roof-line, as previously discussed. Some of the buildings included flat roofs on multiple levels, for example, the results shown in Figure D.2. The planar extraction of this example yielded a quality percentage of 88%, with the lower roof surface having a slightly lower quality percentage due to reconstruction of false roof-lines creating a plane around the shadowed area on the roof.

D.2 Gable Roof Reconstruction

A gable roof consisted of two roof planes with opposing slopes that met at a ridgeline. The ridgeline is connected to the boundary corners via gable lines. Gable roofs commonly have the ridgeline along the centre line of symmetry of the roof. Other variations such as saltbox roofs, where the ridgeline is not at the centre, along the line of symmetry, but shifted to produce one large roof plane and a small plane; and M-Shaped roofs, a combination of gable roofs, were also included in the analysis.

The extracted corners of LOD1 and LOD2 for a gable roof show high positional accuracy, with a RMSE of ±0.38 m for LOD1, and ±0.49 m for LOD2. Similar to the flat roof reconstruction, the height accuracy was slightly worse than the positional accuracy, with a height RMSE of approximately ±0.7 m. Nearly all the boundary corners were extracted, with a percentage of correct corner extraction for LOD1 reconstruction at 91%. The extraction of roof-line corners was slightly worse with only 70% of correct corners extracted. However, on a whole, both LOD1 and LOD2 models suffered from the extraction of too many point as corners. This was due to overhanging vegetation, and small roof features such as chimneys causing incorrect connections.

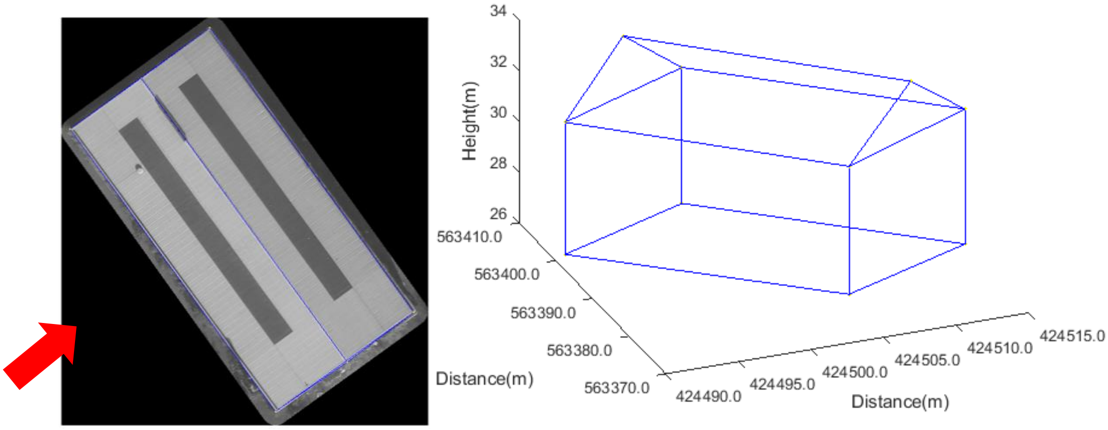


Figure D.3 – Example of successful 3D reconstruction of a gable roof at LOD2.

An example of a successfully reconstructed LOD2 gable roof can be seen in Figure D.3, where all six corner points: four at the roof boundary and two at the endpoints of the ridgeline, were reconstructed. All roof-lines: one ridgeline, four eave lines and the four gable-lines, were successfully extracted to reconstruct the two roof planes. This example yielded a high-quality percentage of the extracted planes. However, for all gabled roofs tested the analysis of the reconstructed planes when compared to reference data, showed unsatisfactory results. The overall quality of the extracted planes was only 32.5%. This was improved marginally when only considering planes over 10 m², with an overall quality of 34.2%. This was due to connecting ridgelines and gable-lines not being extracted in some test sites. An example of this can be seen in Figure D.4, for an M-Shaped roof. Although corner extraction was successful, with 92% of corners reconstructed, planar extraction was only 60% because two corners were not reconstructed. This can be seen in Figure D.4 where one ridgeline was not reconstructed, which resulted in two roof planes not being reconstructed. Erroneous points were extracted for the example from a large vehicle parked next to the building, which was present in the nDSM. This can be seen in Figure D.4 by the incorrect reconstruction of a step edge.

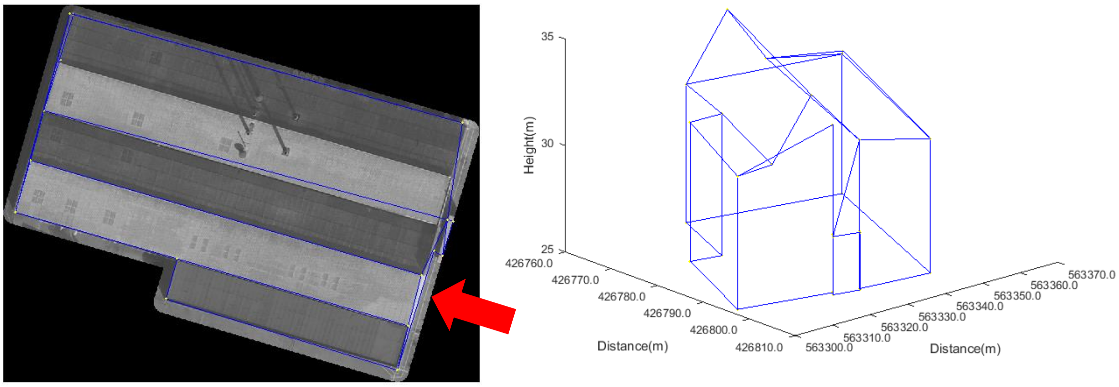


Figure D.4 – Example of unsuccessful reconstruction results of a M-Shaped roof at LOD2.

D.3 Hipped Roof Reconstruction

A hipped roof generally consisted of four roof planes: two triangular planes, with hip lines connecting the central ridgeline to the boundary corners, and two trapezoidal planes separate by a central ridgeline and the two triangular roof planes.

The positional and height accuracies showed agreement with the flat and gable roof structures, with the positional accuracy of the points being within an RMSE of ±0.5 m of the true corner position for both LOD1 and LOD2 models. However, a greater height accuracy was achieved, compared to previous building types, with an RMSE of ±0.38 m. The total number of points extracted for LOD1 was approximately 100%, which suggests that the methodology extracted the correct number of boundary corners for each hipped building. However, the percentage of roof corners detected in their correct location was only 83%, which shows some roof corners were not reconstructed. An example of this can be seen in Figure D.5 where parts of the extension of the building were not reconstructed. This was due to failures in the segmentation of the nDSM boundary, and a clear directional difference not being classified for the edges. Two edges of this small extension were reconstructed, albeit not at the correct orthogonal angle, but the remaining edges were missing and the eave line was over-generalised resulting in only 75% of LOD1 corners being reconstructed. An extra incorrect point was constructed along the boundary from vegetation, shown by the incorrect vertical edge in Figure D.5.

For LOD2 reconstruction, the correct hipped roof structure can be seen to have been correctly reconstructed for the main roof in Figure D.5. None of the corners of the building extension were reconstructed and one corner from the hipped line was missing, which resulted in only 71% of corners being successfully reconstructed. Of the corners successfully reconstructed in Figure D.5, an average positional accuracy of ±0.41 m and an average height accuracy of

±0.25 m was achieved. Small dormer windows did not influence the reconstruction results because no edges were extracted for these features, which in other examples caused erroneous construction of edges.

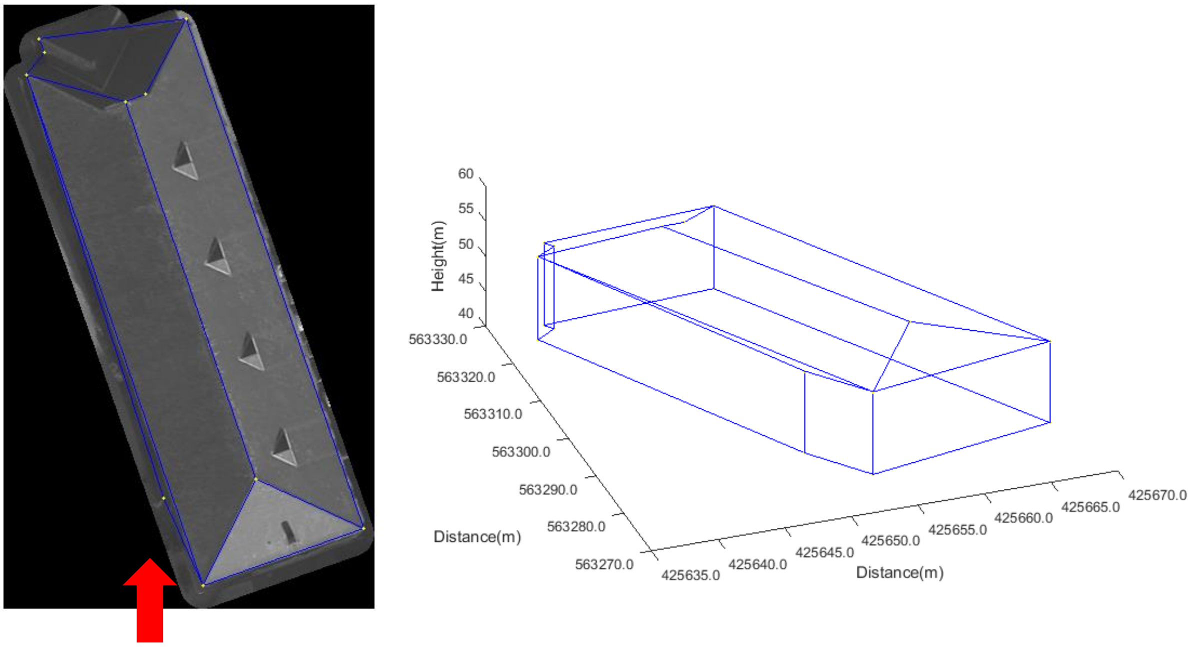


Figure D.5 – Example of a LOD2 reconstructed hipped roof.

Similar to the gable roofs, the percentage of extracted roof planes was very low with a quality percentage of 35%, which increased slightly when only planes larger than 10 m² were considered. As explained for the gable roofs, this was due to some connecting roof-lines not being extracted, an example of which is shown in Figure D.6. One corner was not extracted, which caused two edges to not be connected and meant two roof planes were not reconstructed. This produced a planar quality percentage of only 42%. Although two out of four planes were reconstructed, planar extraction was below 50% due to one corner being extracted in the wrong location. This can be seen at the top of the left image in Figure D.6, where the corner was extracted along the ridgeline. This corner was ±1.4 m from its true location, so was classified as being correctly reconstruction, with regards to the LOD2 specification.

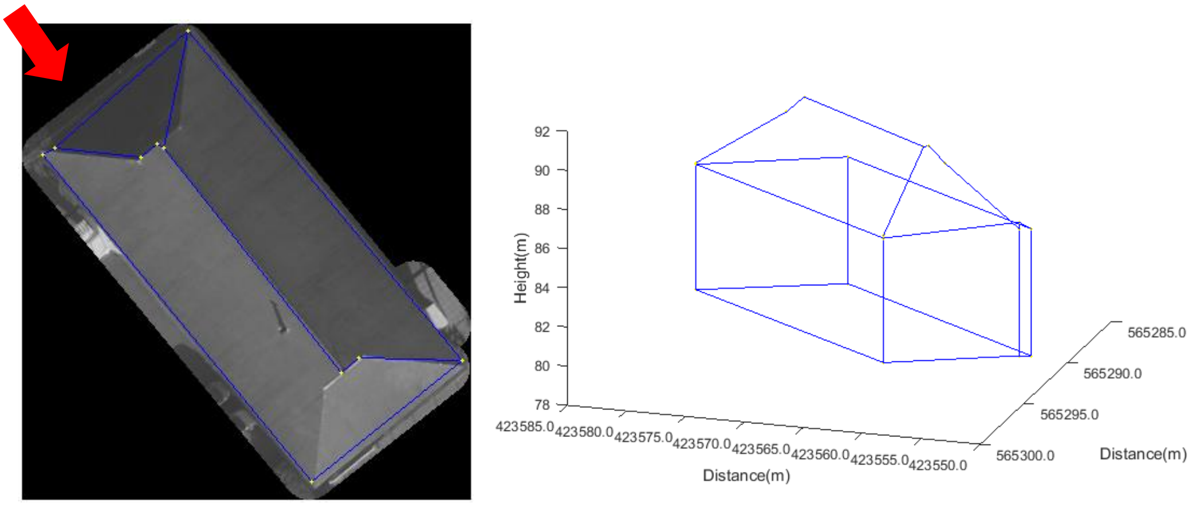


Figure D.6 – Example of LOD2 reconstruction of a hipped roof, showing under-segmentation of the roof planes.

D.4 Cross-Gabled Roof Reconstruction

Cross-gabled roofs are structures that have two or more ridgelines, which are usually perpendicular or parallel to each other, with a slight offset, and connected with a series of valley lines. The positional and height accuracies achieved were again approximately ±0.5 m for LOD1 and LOD2 reconstruction. The extraction of corners was seen to be slightly worse for LOD1 reconstruction when compared to LOD2 reconstruction. This was likely to be due to the complexity of the building boundary. An example of a reconstructed cross-gabled LOD2 model can be seen in Figure D.7, where 81% of corners were successfully reconstructed for LOD2. Errors can be seen in the extraction of points from neighbouring vegetation, which caused the incorrect construction of step edges, due to the height difference between the roof and the vegetation. One boundary corner was not extracted due to the short edge of the nDSM boundary being under-segmented. Another boundary corner was reconstructed in the LOD2 model but not the LOD1 model, due to the edge being vectorised from the Canny edge detection results.

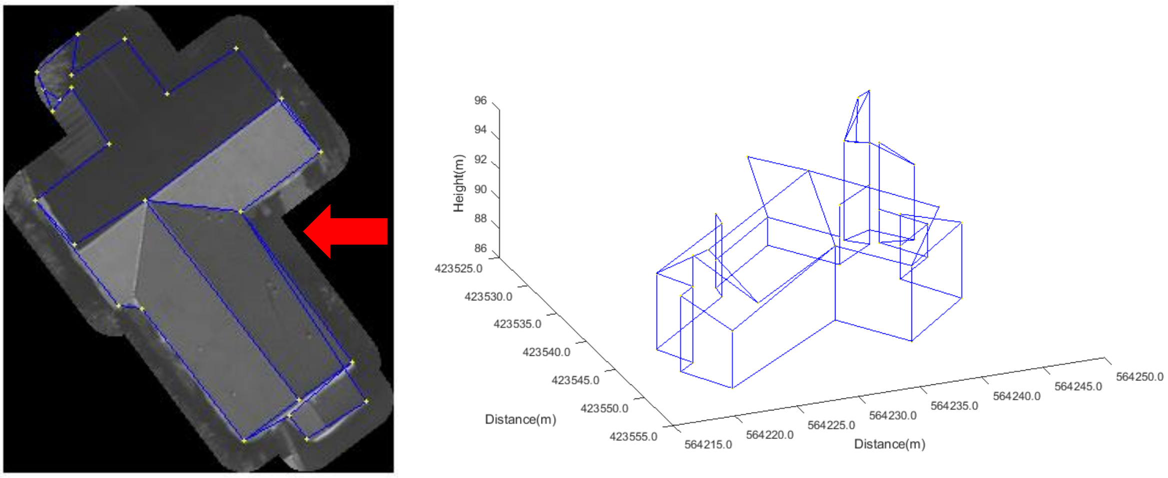


Figure D.7 – Example of a reconstructed LOD2 cross-gabled model.

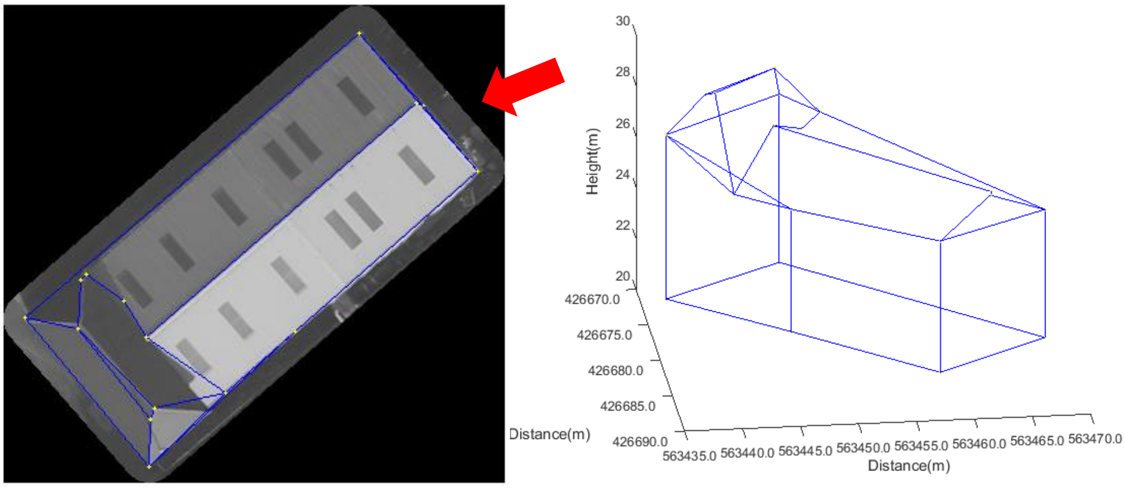


Figure D.8 – Example of 3D reconstruction of a cross-gabled roof.

Overall, the total number of corners extracted for cross-gabled roofs was higher than the number of roof corners, where on average, an extra 20% of points were extracted. An example of this can be seen in Figure D.8, where LOD1 reconstruction extracted one extra point, and LOD2 reconstruction extracted an extra three points. This led to the formation of false planes and incorrect roof-line reconstruction.

The extraction of roof planes, as a whole, was again unsatisfactory with an overall quality of 38%. This was again evident from the misidentification of roof-lines. The example in Figure D.7 extracted 53% of planes, which was due to several missing corners and roof-lines. The example shown in Figure D.8 reconstructed roof planes to an overall quality of 68%, reconstructing four out of six planes. Although 100% of roof corners were extracted, one corner from the hipped-line of the hipped roof section was not connected to the eave line, which meant two roof planes were not reconstructed. The location of some corners was also incorrect, which meant that planes were not the correct size.

D.5 Complex Roof Reconstruction

A complex roof was defined as a roof that did not fit into any of the other four categories, with roof geometry that was irregular and could not be easily described as a predefined shape. As was the case with the other roof types tested, the positional accuracy of extracted points was within ±0.5 m of the true position. However, the height accuracy was much worse, with an RMSE of over 1 m for LOD1 models, and, on average, ±0.74 m for LOD2 models. Similar to cross-gabled roofs, the extraction of corners was worse for LOD1 reconstruction compared to LOD2 reconstruction. The under-segmentation of the nDSM boundary and the poor height accuracy were due to the complexity of the building boundary, with small extensions to the boundary, which were over-generalised by the nDSM boundary.

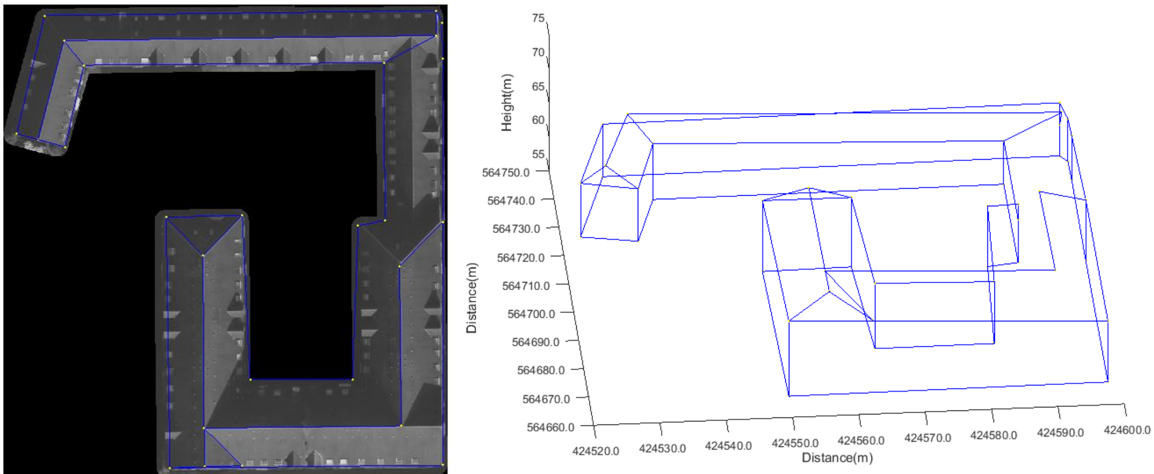


Figure D.9 – Example of LOD2 reconstructed model of a complex roof.

An example of a complex roof can be seen in Figure D.9, which extracted 61% of all roof corners, with an average positional accuracy of ±0.82 m and height accuracy of ±0.54 m. Although the extraction of boundary corners was lower than the extraction of roof-line corners for complex roofs, the example in Figure D.9 shows all boundary corners were reconstructed with some corners missing from valley and ridgelines. This example shows how corners can be reconstructed at non-orthogonal corners, using the defined criteria.

The overall quality of extracted roof planes for complex buildings was 43%. This was higher than the extraction of other roof shapes, apart from flat roofs, but is still unsatisfactory for automated 3D building reconstruction. Most of these complex buildings were large buildings with easily distinguishable roof planes. This can be seen in Figure D.9, where many of the large planes have been reconstructed. However, due to a number of edges not being extracted, planar extraction of this example was only 20%. The increase in planar quality for more complex buildings may suggest the parameter settings used are more suitable to larger buildings.

A much more complex roof structure can be seen in Figure D.10, where the building consists of several flat roofs, on multiple levels. From the 3D reconstruction plot in Figure D.10, it can be seen that the roof edges were created at multiple levels with the introduction of step edges. However, some step edges were incorrectly included due to errors in the segmentation of nDSM boundary. Errors in the OS MM building footprint, used to extract the initial search area, can be seen from the underlying true-orthophoto in Figure D.10. Errors in the building footprint influenced the ground classification procedure to include an extra four boundary points, which caused the incorrect extraction of a step edge. Over-segmentation of the classified directions along separate boundary edges caused the extraction of another incorrectly determined step edge, due to incorrect height extraction. The overall number of roof corners extracted was almost twice the required amount for reconstruction, but several of the edges extracted were not along a main roof edge. A clear example can be seen of an edge being detected from a shadow across a roof surface in Figure D.10. Although there were several failings in the extraction of roof corners, roof planes were extracted to a quality of 85%, as the five roof planes were extracted at the various height levels, shown in Figure D.10. The 3D reconstruction also highlights the need to develop the ruleset to determine step edges from roof-lines, as can be seen from a floating plane in Figure D.10, which has one corner incorrectly connected to the roof boundary.

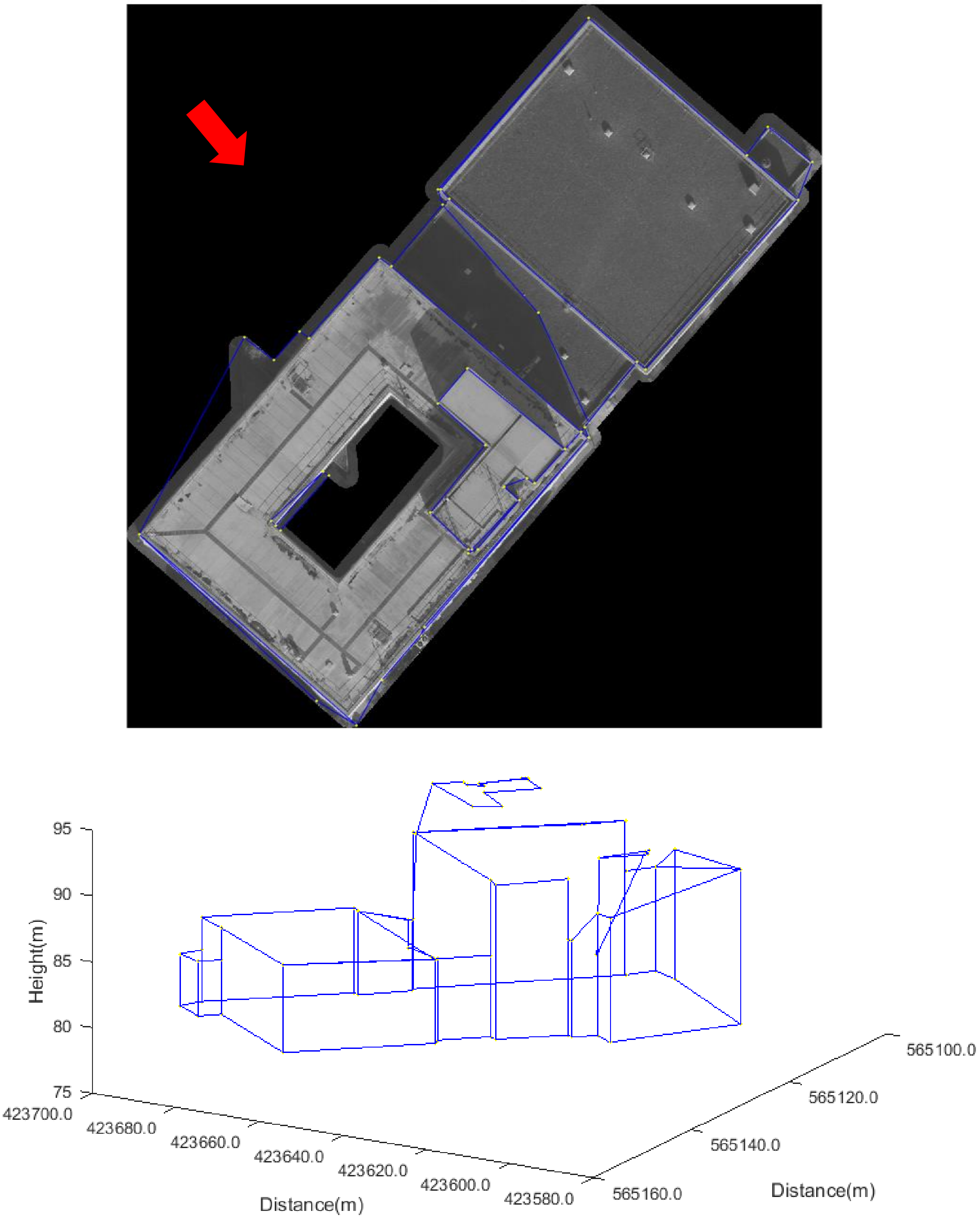


Figure D.10 – Example of 3D reconstruction of a multi-levelled complex flat roof.

Appendix E. Reconstruction by Selected Scene from Newcastle upon Tyne

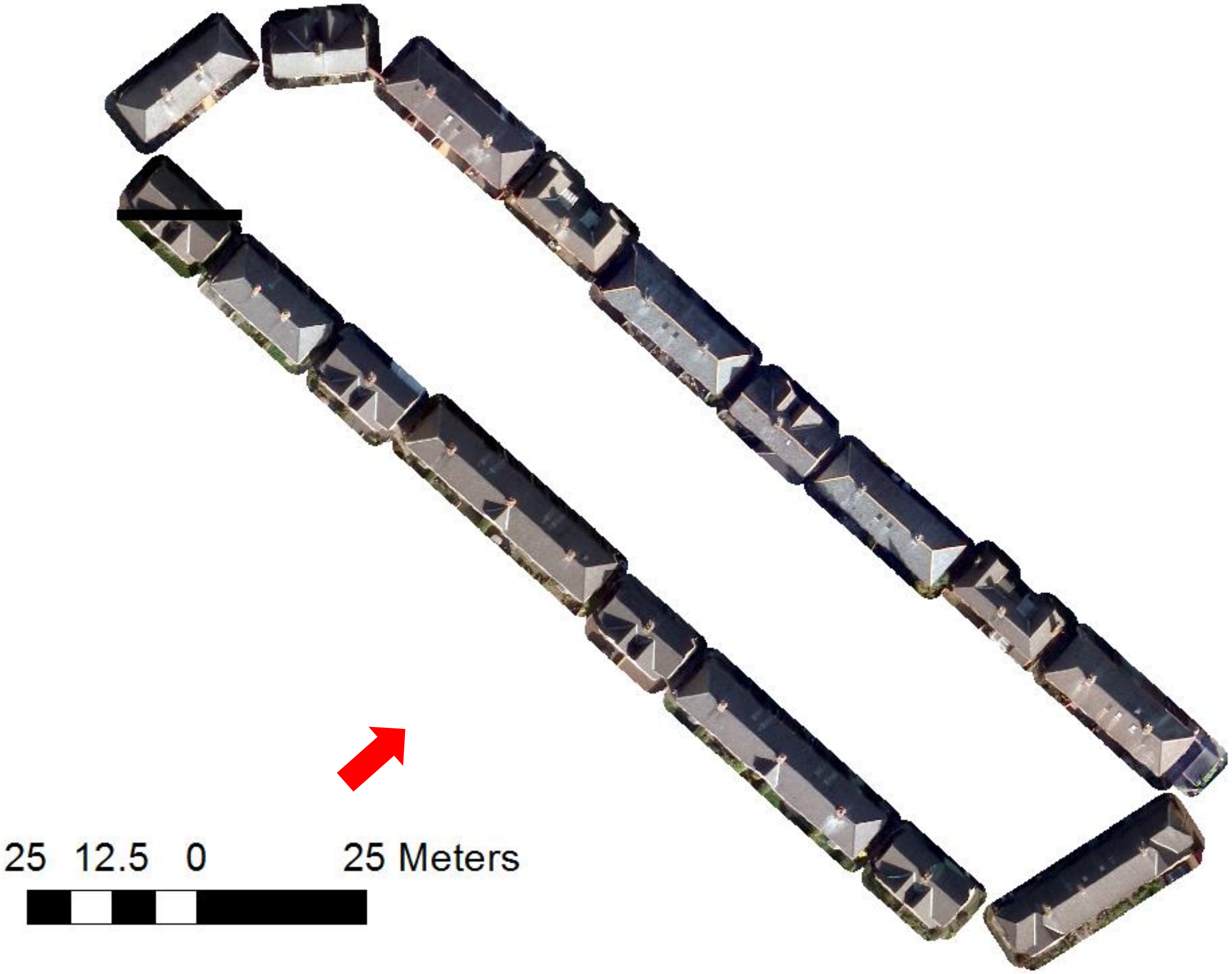


Figure E.1 – Residential1 buildings extracted from the true-orthophoto.

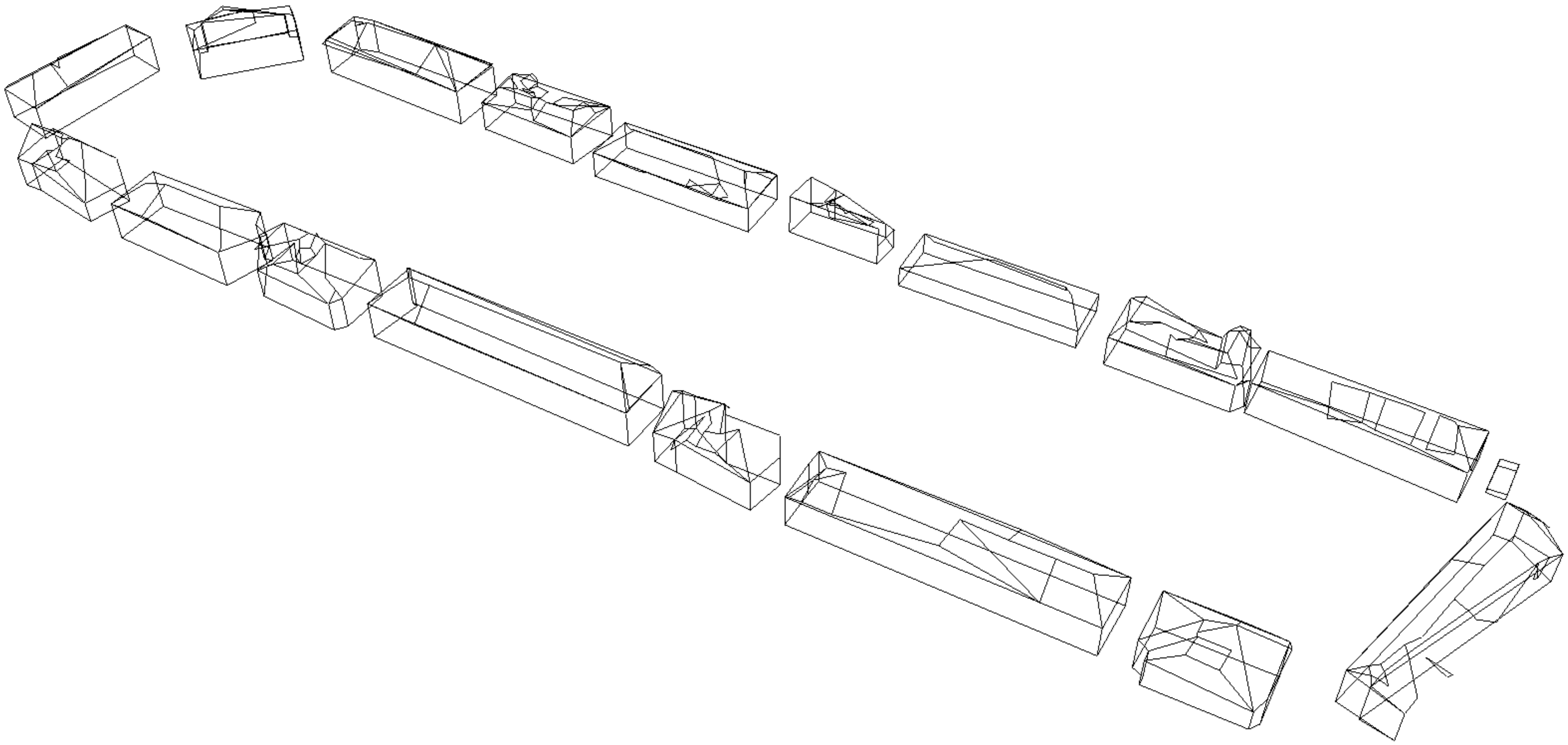


Figure E.2 – Reconstructed 3D models for Residential 1.

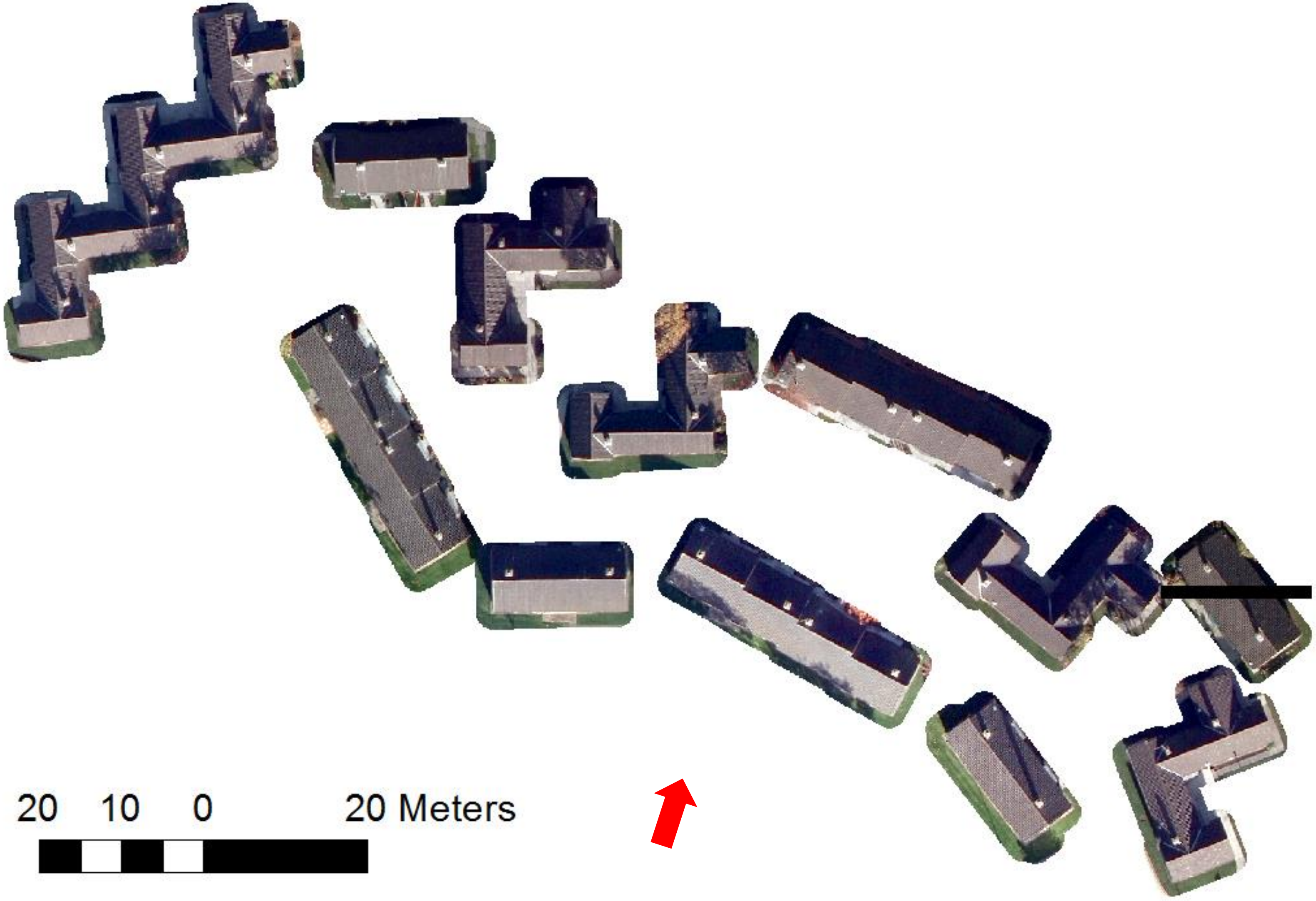


Figure E.3 – Residential2 buildings extracted from the true-orthophoto.

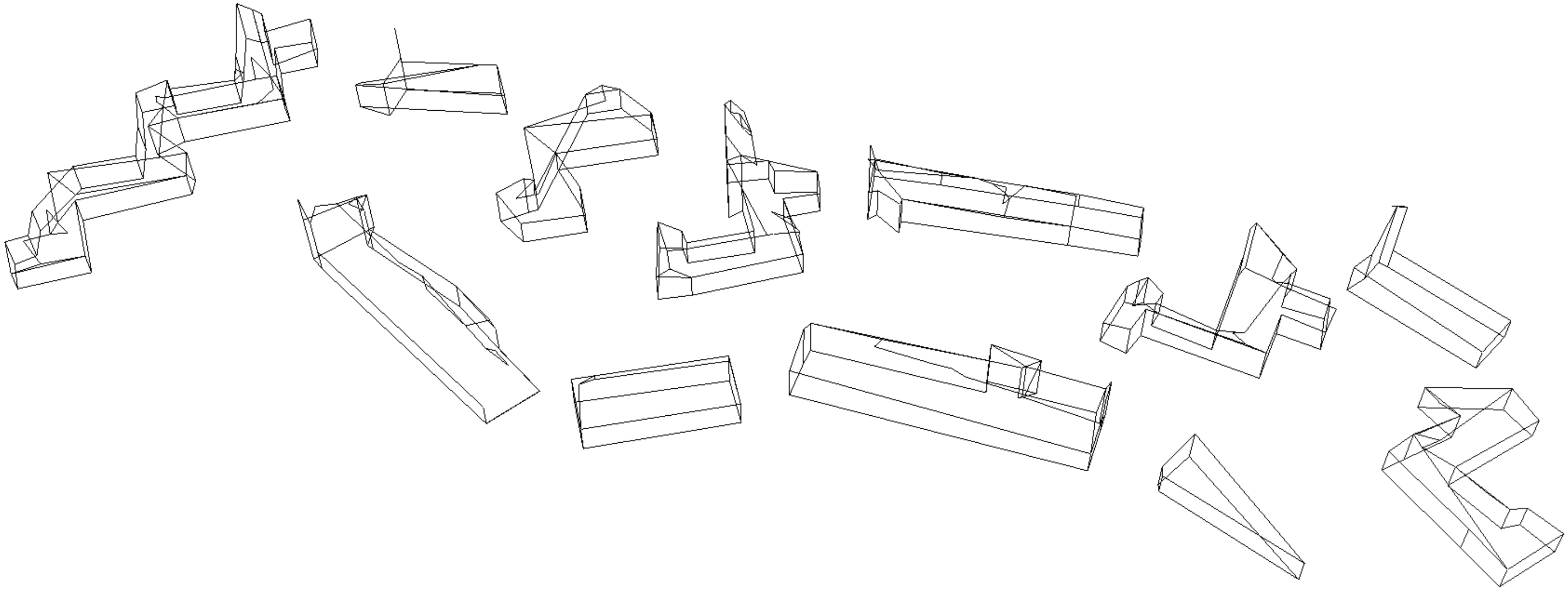


Figure E.4 – Reconstructed 3D models for Residential2.

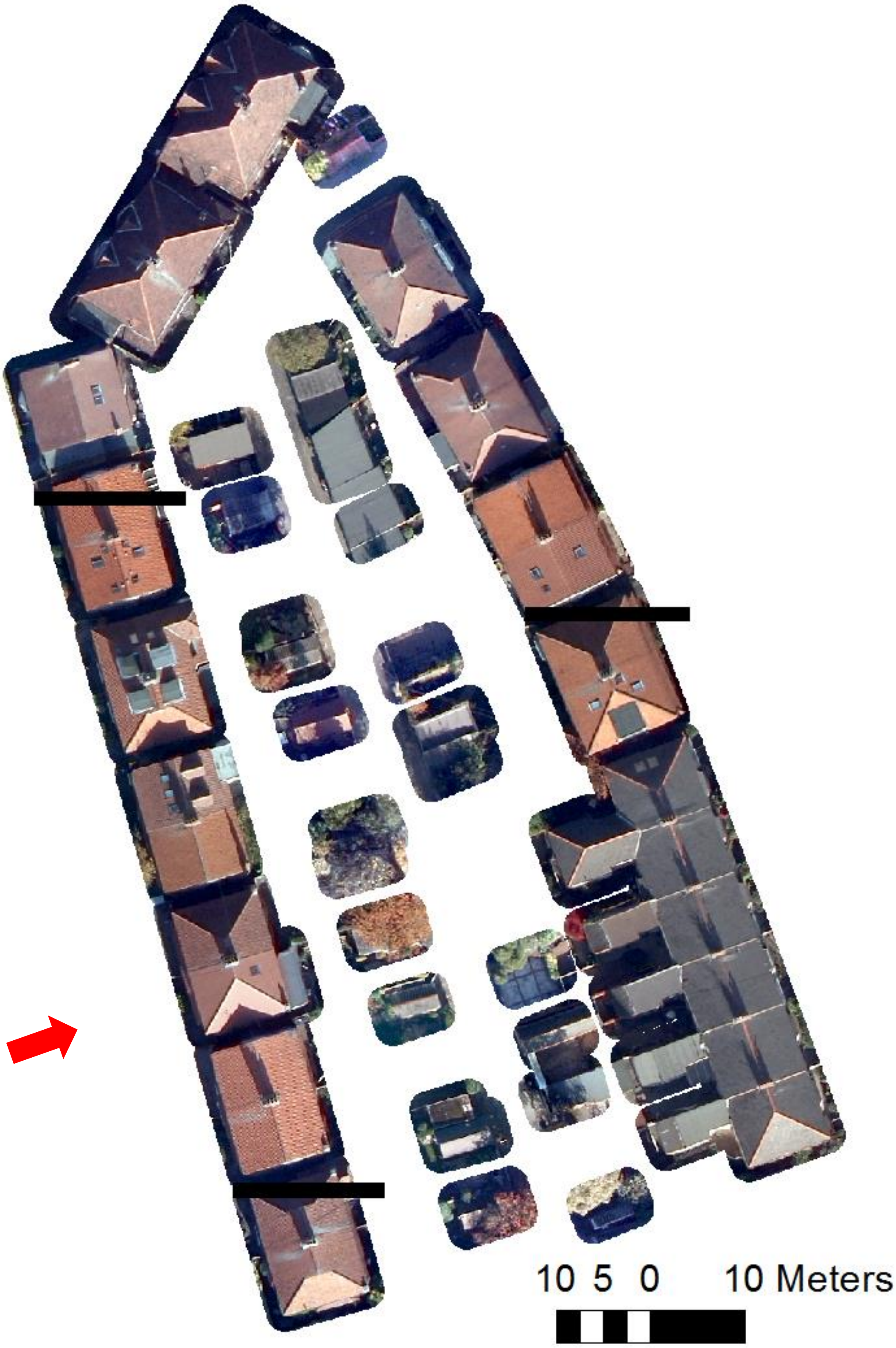


Figure E.5 – Residential3 buildings extracted from the true-orthophoto.

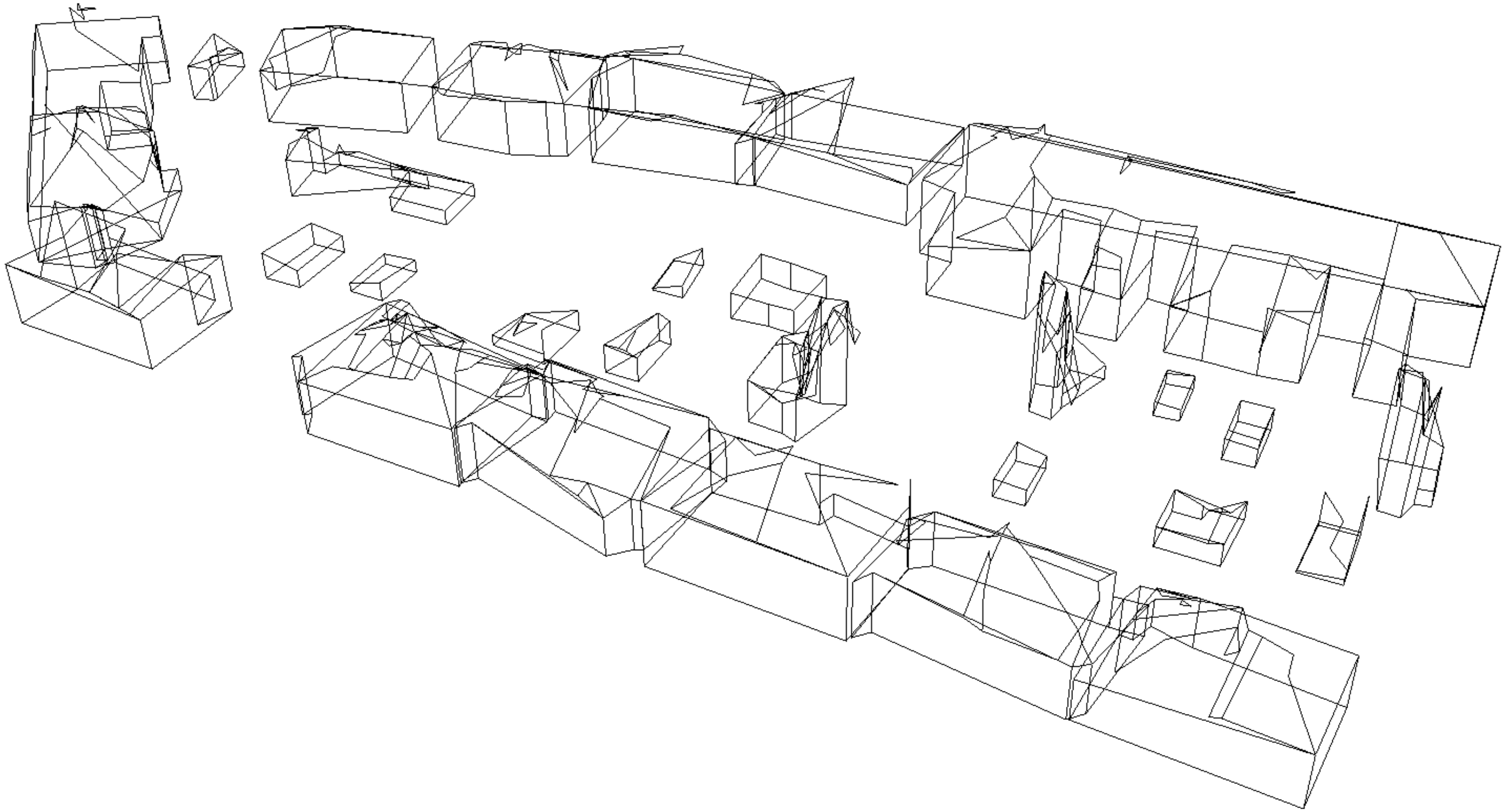


Figure E.6 – Reconstructed 3D models for Residential3.

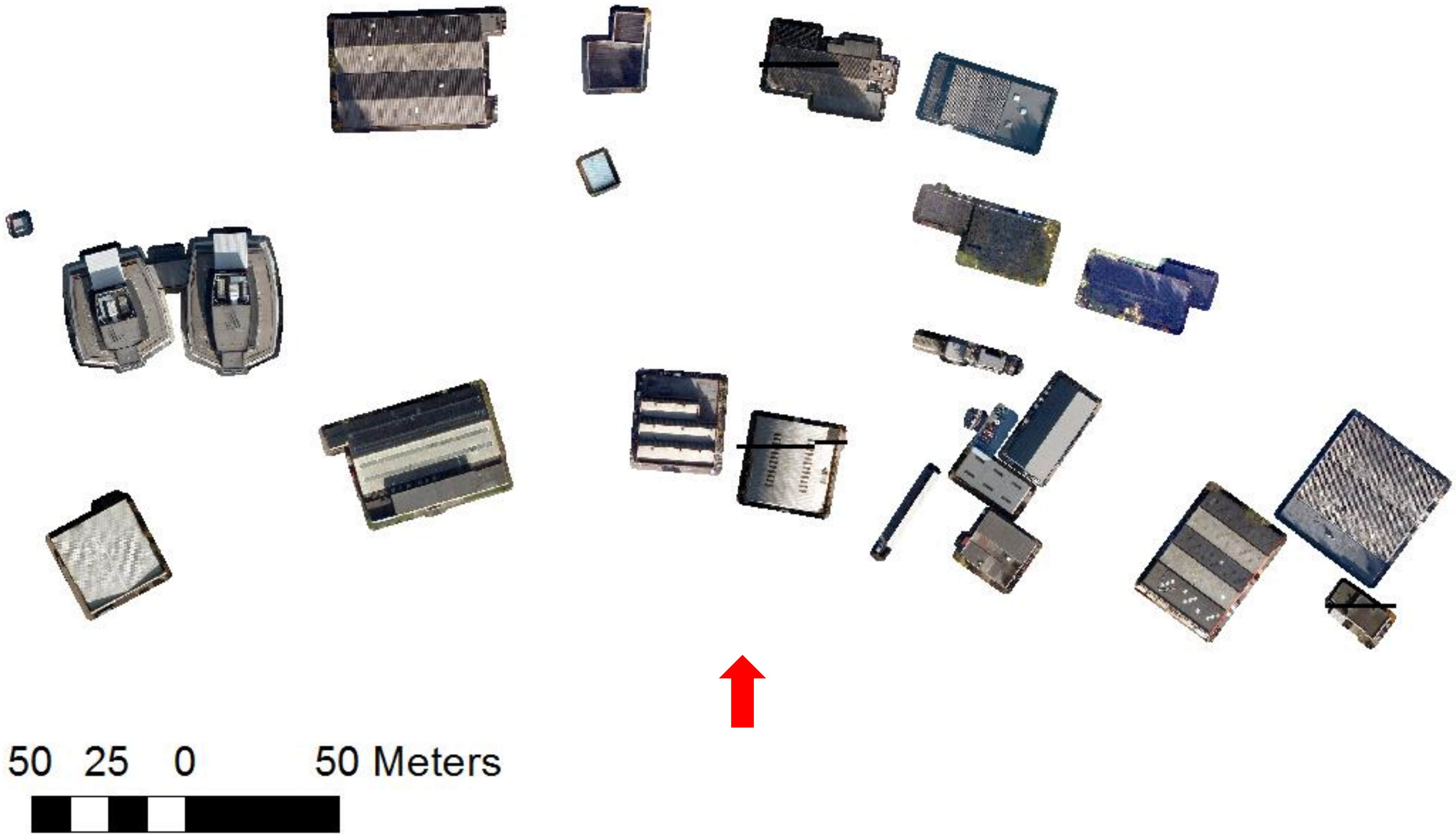


Figure E.7 - Industrial1 buildings extracted from the true-orthophoto.

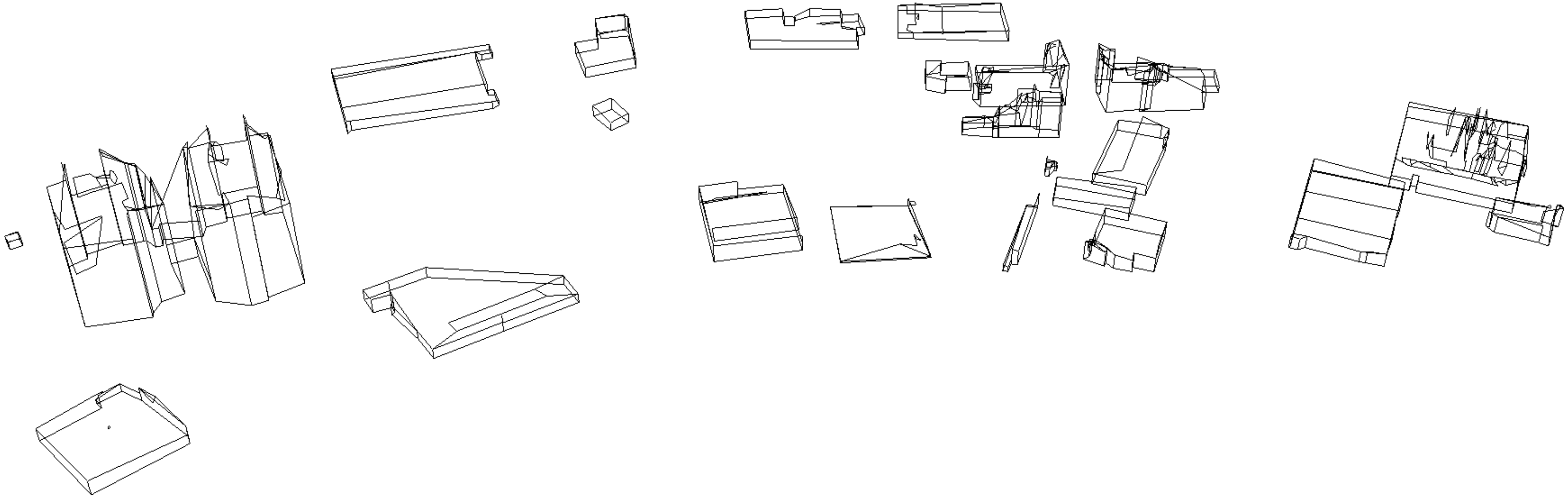


Figure E.8 – Reconstructed 3D models for Industrial1.

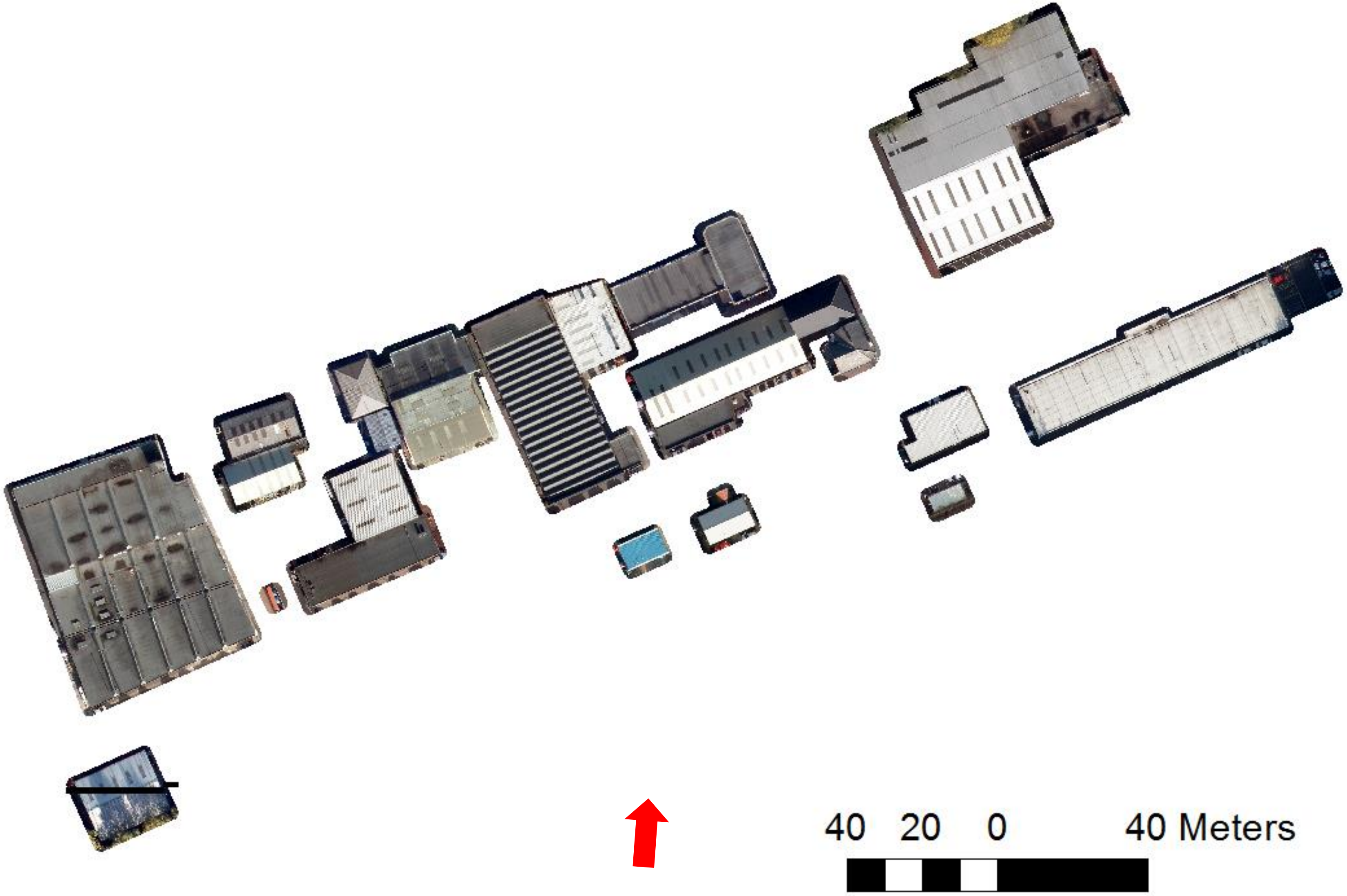


Figure E.9 – Industrial2 buildings extracted from the true-orthophoto.

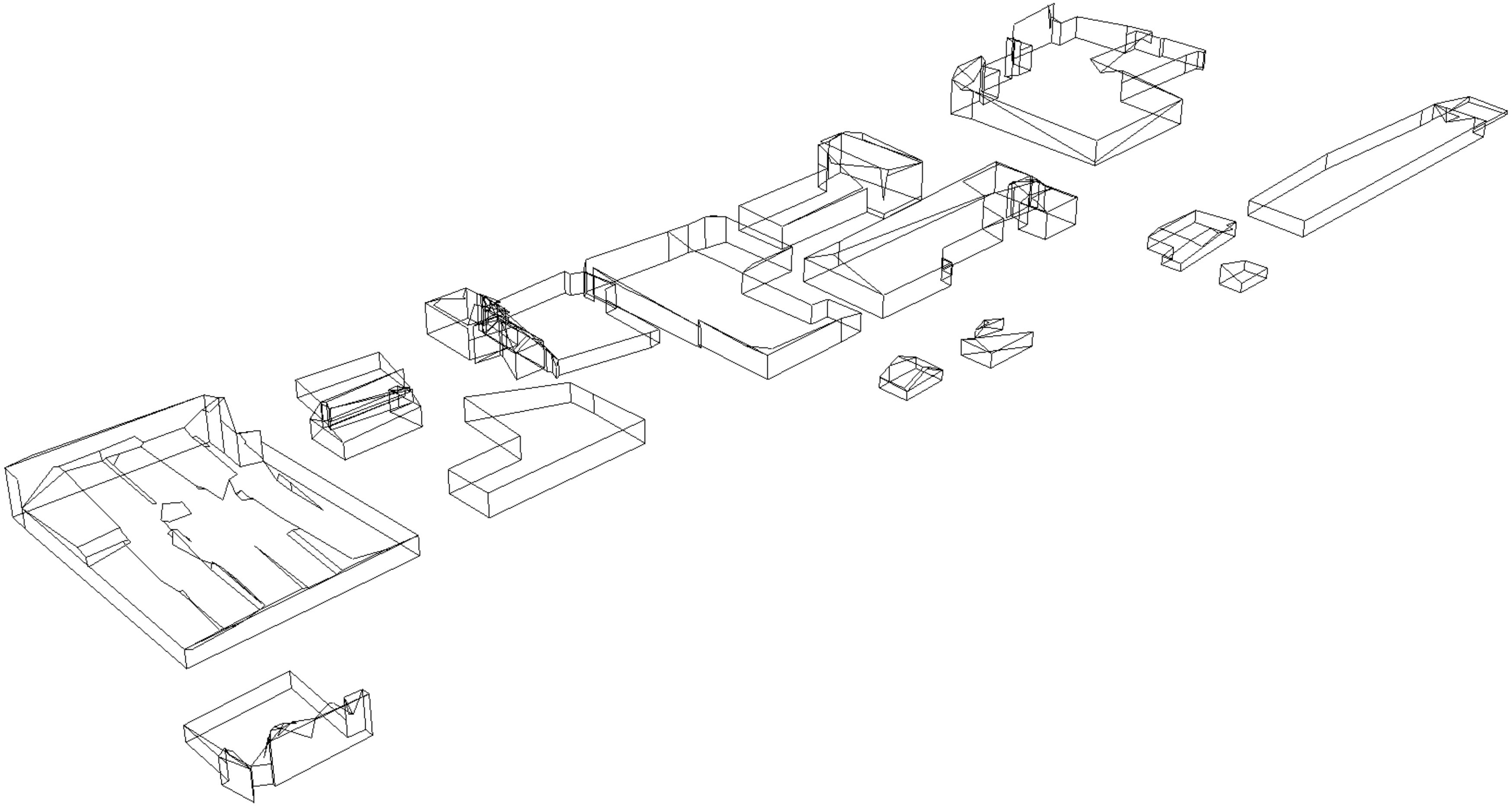


Figure E.10 – Reconstructed 3D models for Industrial2.

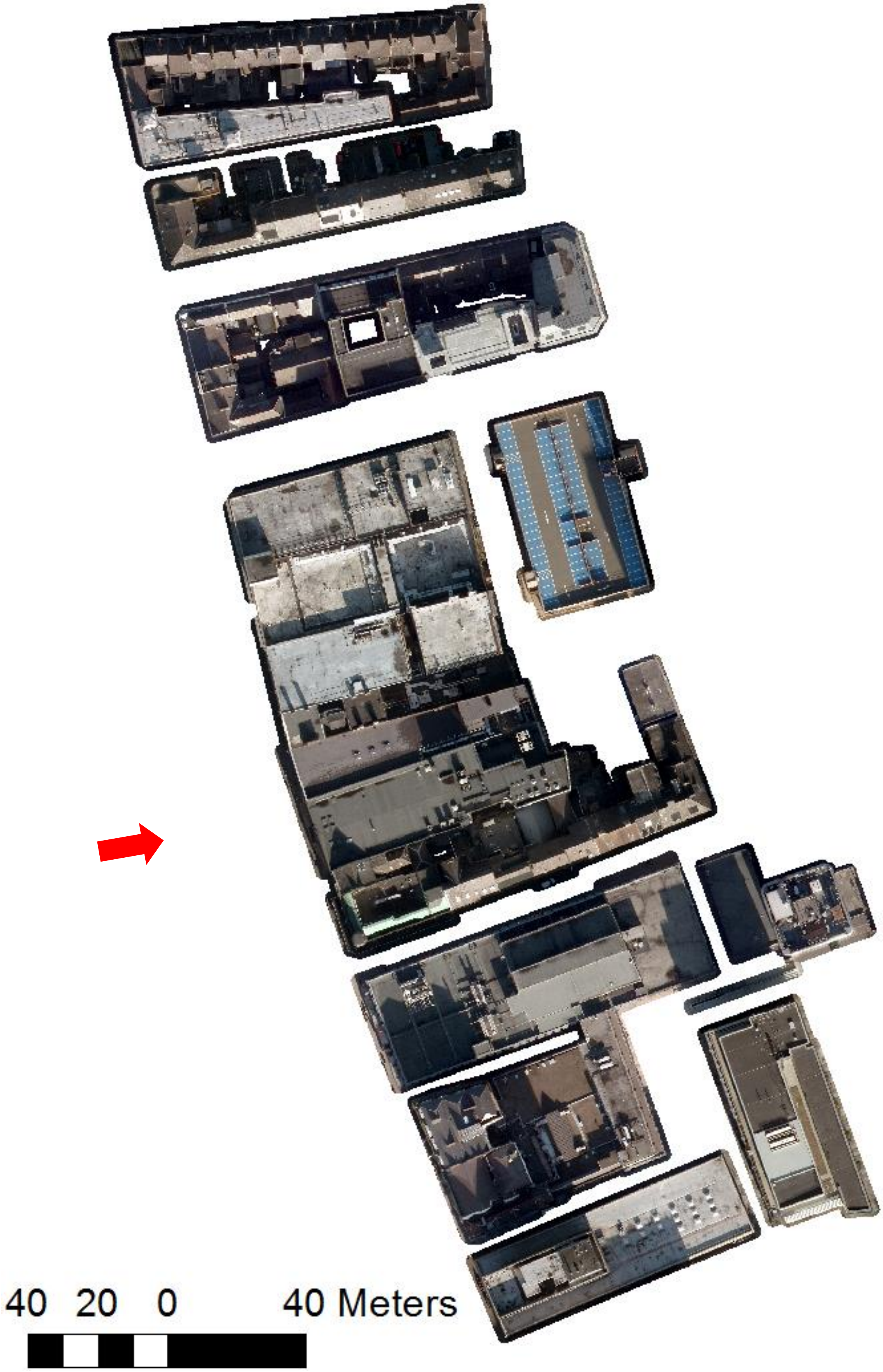


Figure E.11 – CityCentre1 buildings extracted from the true-orthophoto.

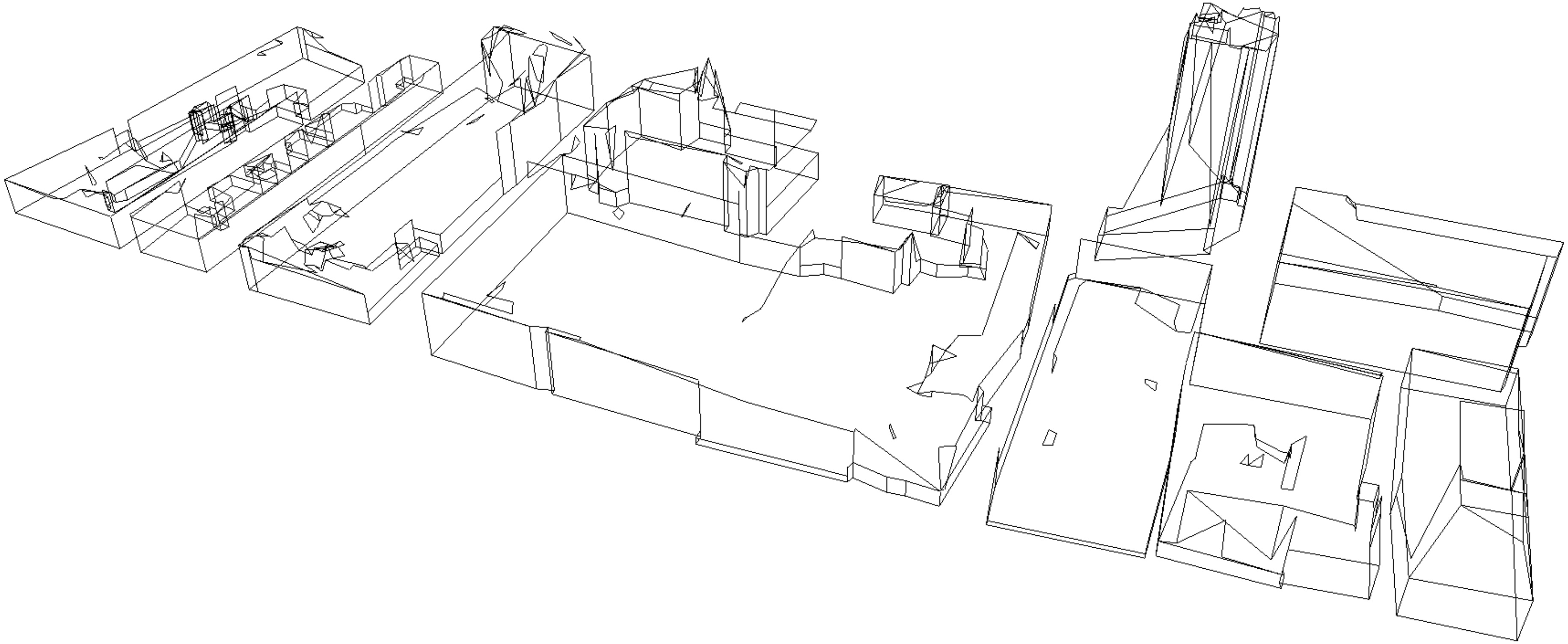


Figure E.12 – Reconstructed 3D models for CityCentre1.

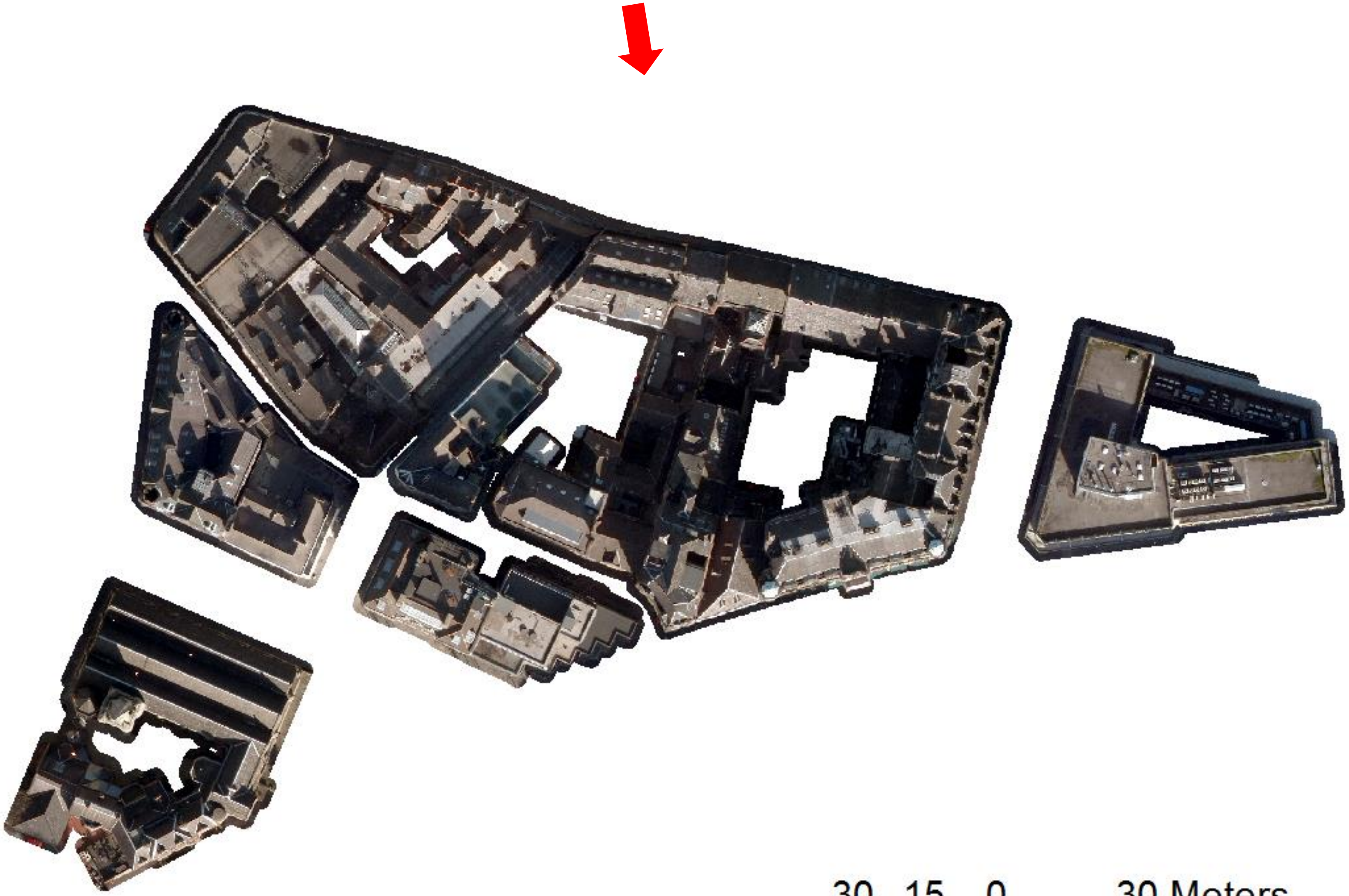


Figure E.13 – CityCentre2 buildings extracted from the true-orthophoto.

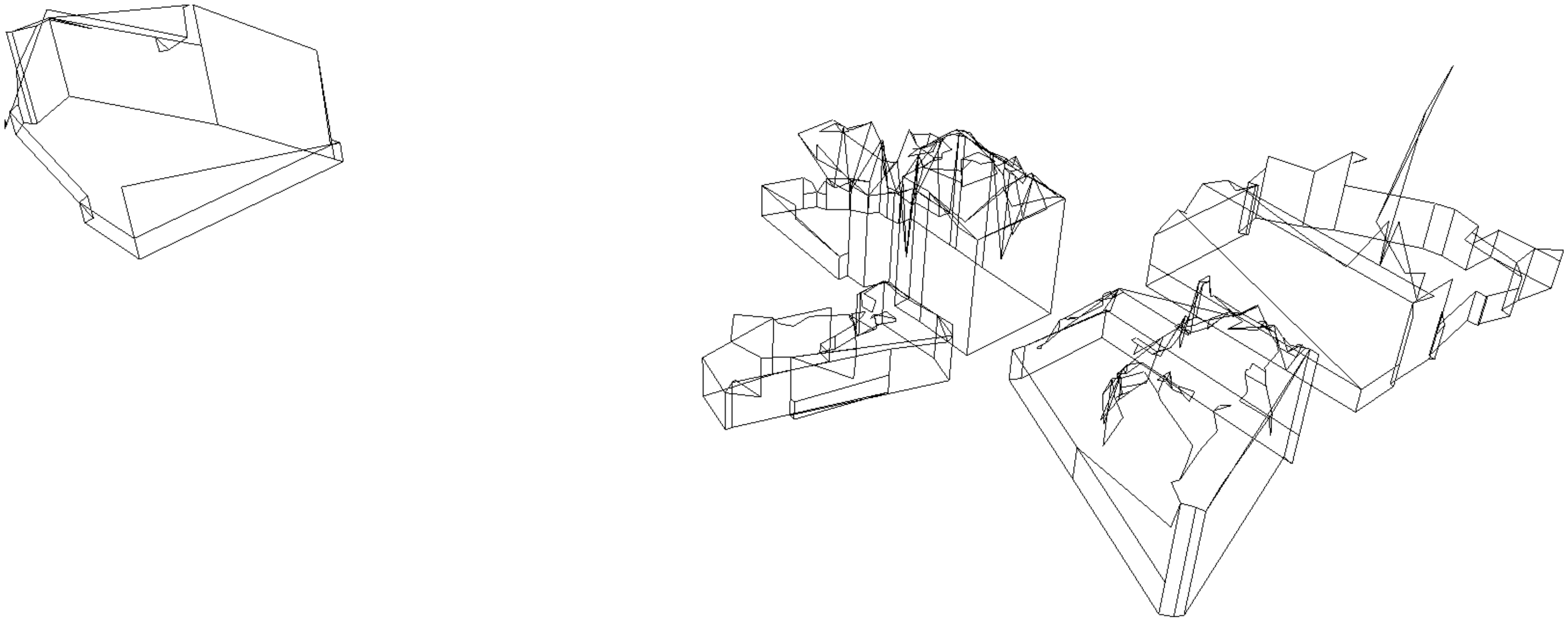


Figure E.14 – Reconstructed 3D models for CityCentre2.

Appendix F.Vaihingen Extracted Buildings

F.1 Area1

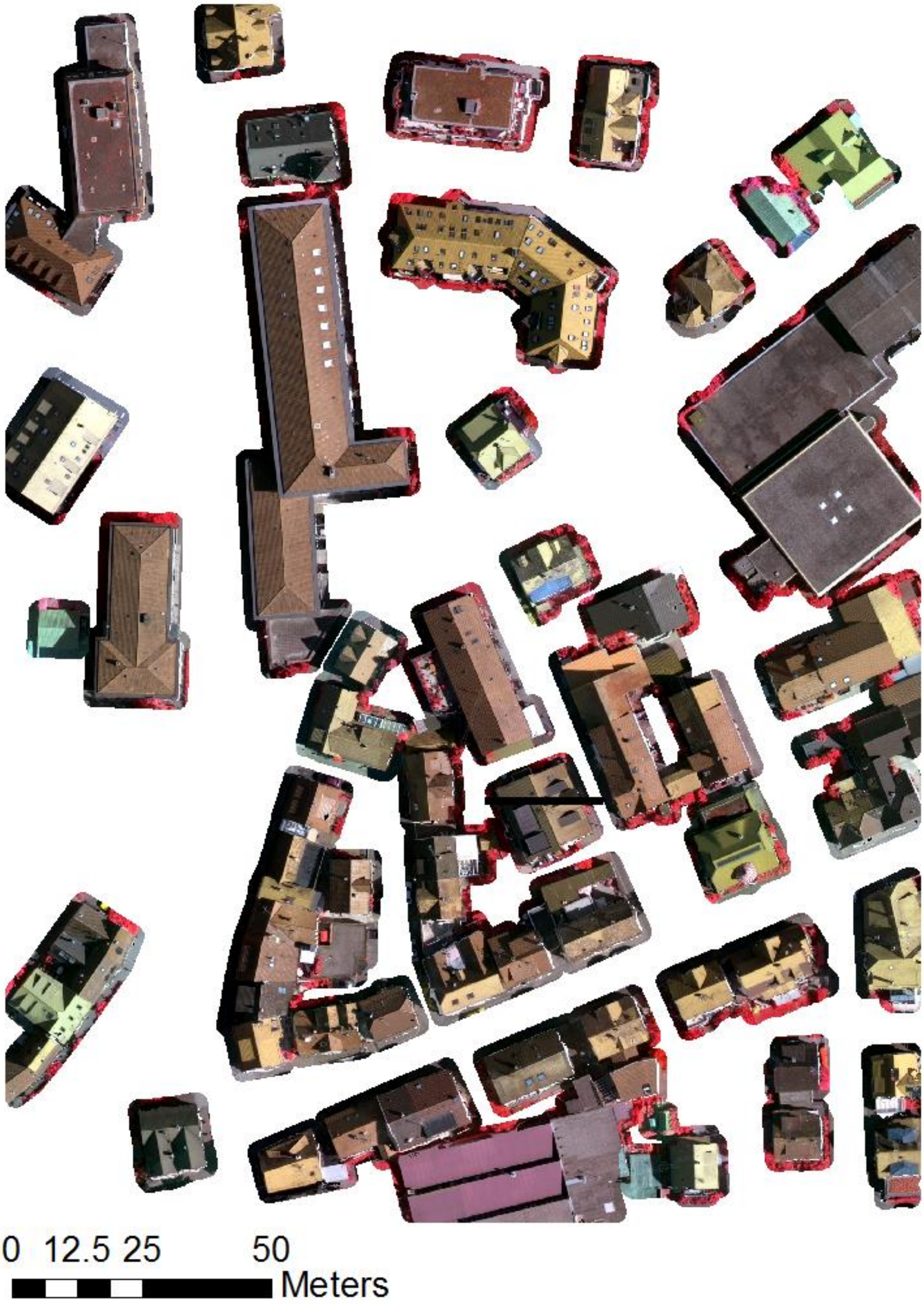


Figure F.1 – Buildings extracted by footprints in Area1.

F.2 Area3

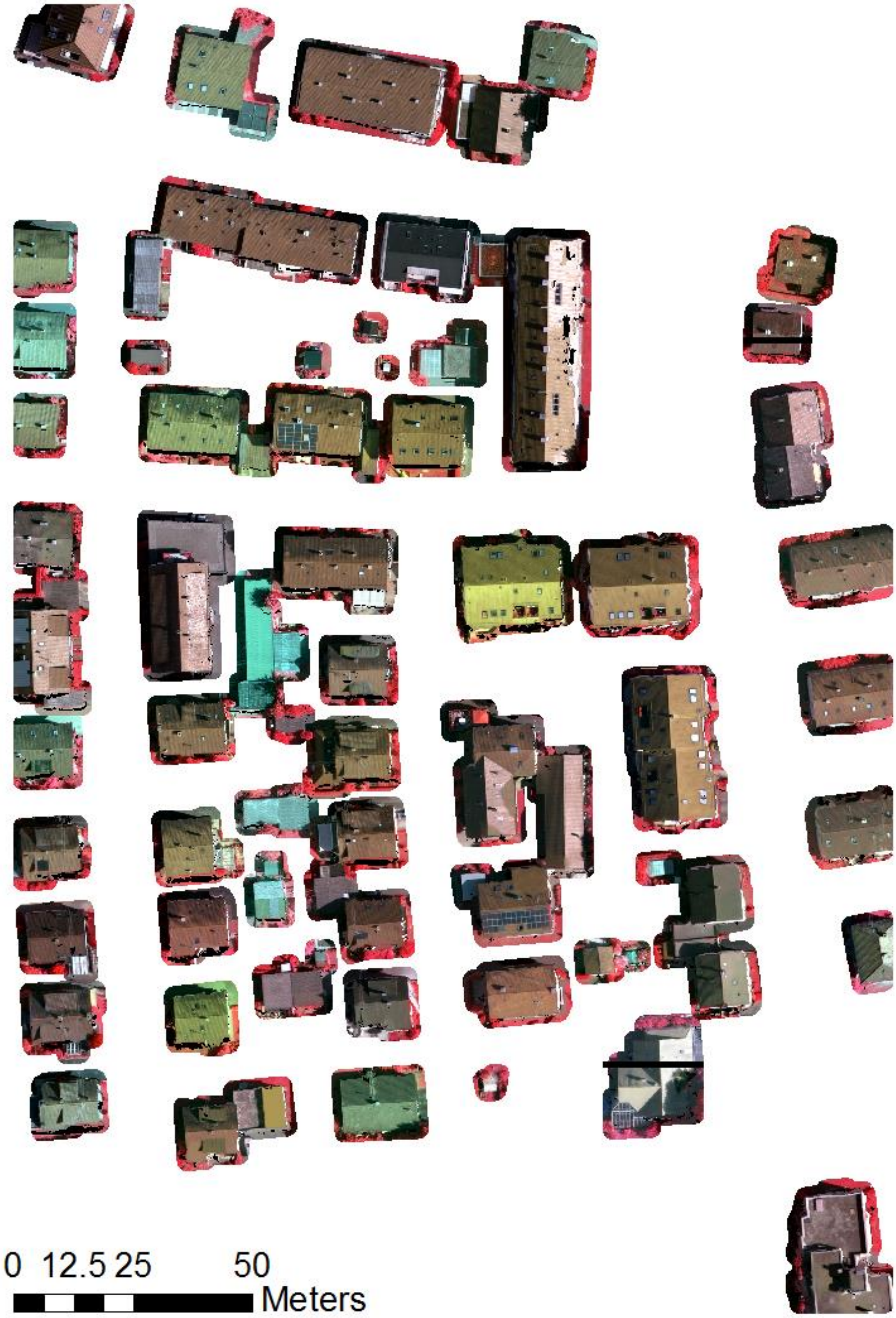


Figure F.2 - Buildings extracted by footprints in Area3.

F.3 Area21

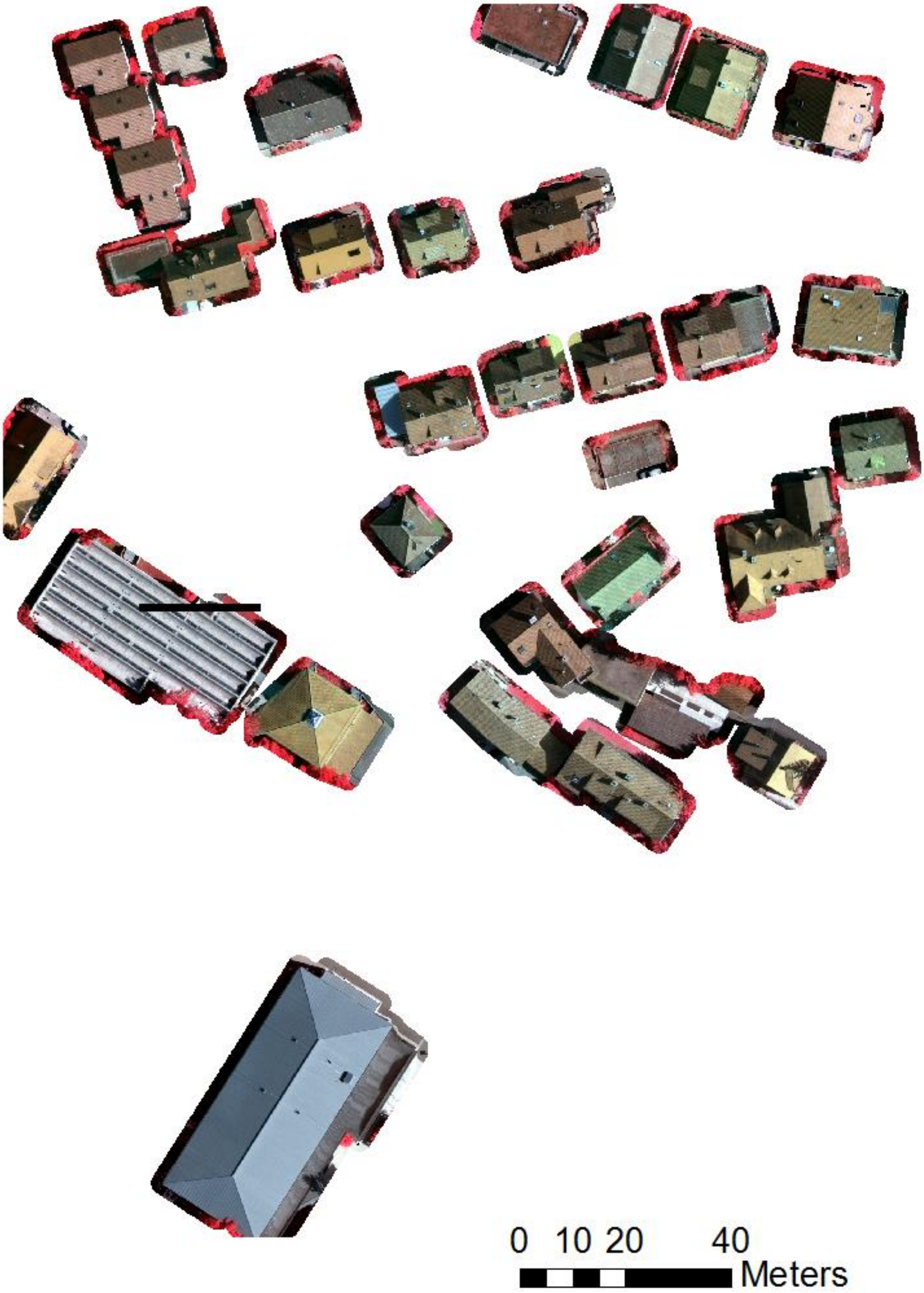


Figure F.3 - Buildings extracted by footprints in Area21.

F.4 Area32

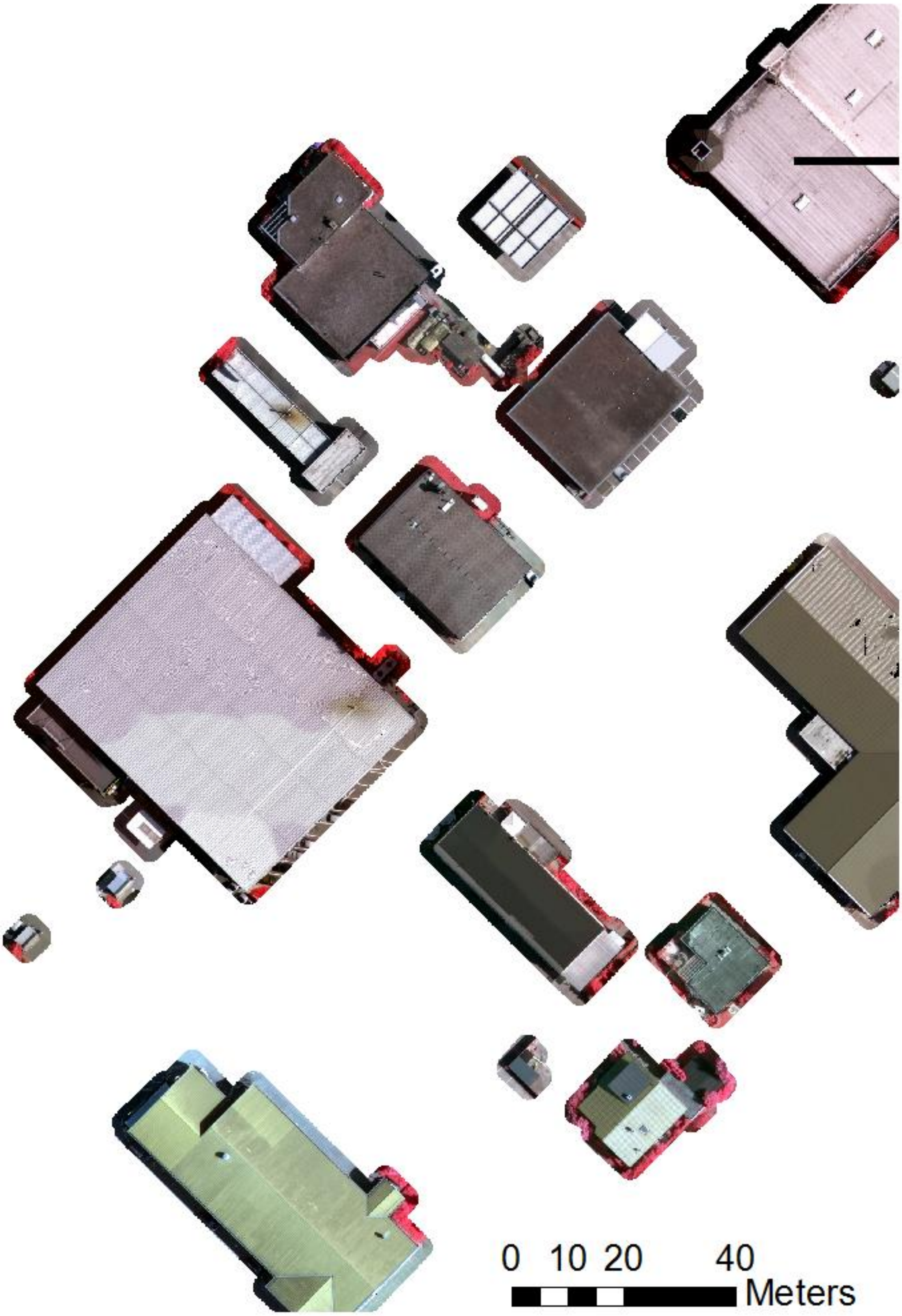


Figure F.4 - Buildings extracted by footprints in Area32.

References

Adelaide City Council (2016) *3D City Model*. Available at: http://www.adelaidecitycouncil.com/planning-development/city-planning/3d-city-model/ (Accessed: 07/05/16).

Agugiaro, G. (2016) 'First steps towards an integrated CityGML-based 3D model of Vienna', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* 3(4), pp. 139-146.

Akca, D., Freeman, M., Sargent, I. and Gruen, A. (2010) 'Quality assessment of 3D building data', *The Photogrammetric Record*, 25(132), pp. 339-355.

Alharthy, A. and Bethel, J. (2004) 'Detailed building reconstruction from airborne laser data using a moving surface method', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35(3), pp. 213 - 218.

Arefi, H., Engels, J., Hahn, M. and Mayer, H. (2008) 'Levels of Detail in 3D Building Reconstruction for LiDAR Data', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* 37(B3b), pp. 485 - 490.

Avrahami, Y., Raizman, Y. and Doytsher, Y. (2007) 'Semi-automatic approach toward mapping of flat-roofed buildings within a non-stereoscopic environment', *The Photogrammetric Record,* 22(117), pp. 53-74.

Awrangjeb, M., Zhang, C. and Fraser, C.S. (2012) 'Automatic reconstruction of building roofs through effective integration of lidar and multispectral imagery', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* 1(3), pp. 203-208.

BAE Systems (2011) *SOCET GXP* (Version 4.1.0) [Computer program]. BAE Systems.

Berlin Business Location Centre (2009) *12th IAAF World Championships in Athletics berlin 2009*. Available at: http://www.businesslocationcenter.de/en/berlin-economic-atlas/the-project/project-examples/marathon (Accessed: 03/06/16).

Berlin Business Location Centre (2013) *smartMap Berlin - the 3D model of the city goes mobile*. Available at: http://www.businesslocationcenter.de/en/smartmap (Accessed: 09/05/16).

Berlin Business Location Centre (2016a) *3D City Model*. Available at: http://www.businesslocationcenter.de/en/berlin-economic-atlas/the-project (Accessed: 09/05/16).

Berlin Business Location Centre (2016b) *The Course of the Berlin Wall*. Available at: http://www.businesslocationcenter.de/en/berlin-economic-atlas/the-project/project-examples/berlin-wall (Accessed: 09/05/16).

Bhardwaj, S. and Mittal, A. (2012) 'A Survey on Various Edge Detector Techniques', *Procedia Technology,* 4, pp. 220-226.

Biljecki, F., Ledoux, H., Du, X., Stoter, J., Soon, K.H. and Khoo, V.H.S. (2016a) 'The Most common Geometric and Semantic Errors in CityGML Datasets', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W1, pp. 13-22.

Biljecki, F., Ledoux, H. and Stoter, J. (2016b) 'Generation of Multi-Lod 3D City Models in CityGML with the Procedural Modelling Engine RANDOM3DCITY', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W1, pp. 51-59.

Biljecki, F., Ledoux, H. and Stoter, J. (2016c) 'An improved LOD specification for 3D building models', *Computers, Environment and Urban Systems*, 59, pp. 25-37.

Biljecki, F., Ledoux, H., Stoter, J. and Vosselman, G. (2016d) 'The variants of an LOD of a 3D building model and their influence on spatial analyses', *ISPRS Journal of Photogrammetry and Remote Sensing*, 116, pp. 42-54.

Biljecki, F., Ledoux, H., Stoter, J. and Zhao, J. (2014) 'Formalisation of the level of detail in 3D city modelling', *Computers, Environment and Urban Systems*, 48, pp. 1-15.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S. and Çöltekin, A. (2015) 'Applications of 3D City Models: State of the Art Review', *ISPRS International Journal of Geo-Information*, 4(4), pp. 2842 - 2889.

Boatto, L., Consorti, V., Del Buono, M., Di Zenzo, S., Eramo, V., Esposito, A., Melcarne, F., Meucci, M., Morelli, A. and Mosciatti, M. (1992) 'An interpretation system for land register maps', *Computer*, 25(7), pp. 25-33.

Boeters, R., Arroyo Ohori, K., Biljecki, F. and Zlatanova, S. (2015) 'Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry', *International Journal of Geographical Information Science*, 29(12), pp. 2248-2268.

Boland, J., Ager, T., Edwards, E., Frey, E., Jones, P.A., Jungquist, R.K., Lareau, A.G., Lebarron, J., King, C.S., Komazaki, K., Toth, C., Walker, S., Whittaker, E.F., Zavattero, P. and Zuegge, H. (2004) 'Cameras and Sensing Systems', in McGlone, J.C. (ed.) *Manual of Photogrammetry*. Fifth edn. Maryland, United States of America: American Society of Photogrammetry and Remote Sensing, pp. 581 - 676.

Bretar, F. (2008) 'Feature Extraction from LiDAR Data in Urban Areas', in Shan, J. and Toth, C. (eds.) *Topographic Laser Ranging and Scanning*. Boca Raton, Florida: CRC Press, pp. 403-420.

Brovelli, M.A., Hogan, P., Prestifilippo, G. and Zamboni, G. (2016) 'NASA Web WorldWind: multidimensional virtual globe for geo big data visualistion', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41(B2), pp. 563-566.

Buhur, S., Kersten, T., Buyuksalih, G., Jacobsen, K., Baz, I., Dursun, S. and Sagir, D. (2008) '3D City modelling of Istanbul historic peninsula by combination of aerial images and terrestrial laser scanning data', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B7), pp. 1239 - 1246.

Bulatov, D., Häufel, G., Meidow, J., Pohl, M., Solbrig, P. and Wernerus, P. (2014) 'Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks', *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 157-170.

Bulatov, D., Solbrig, P., Gross, H., Wernerus, P., Repasi, E. and Heipke, C. (2012) 'Context-based urban terrain reconstruction from UAV-videos from geoinformation applications', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1), pp. 75-80.

Canny, J. (1986) 'A Computational Approach to Edge Detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), pp. 679-698.

Capstick, D. and Heathcote, G. (2006) 'Moving Towards 3D — From a National Mapping Agency Perspective', in Abdul-Rahman, A., Zlatanova, S. and Coors, V. (eds.) *Innovations in 3D Geo Information Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 491-500.

Carpenter, J. and Snell, J. (2013) *Future trends in geospatial information management: the five to ten year vision*. Ordnance Survey: UN-GGIM Ordnance Survey.

Chaturvedi, K. and Kolbe, T.H. (2016) 'Integrating Dynamic Data and Sensors with Semantic 3D City Models', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W1, pp. 31-38.

Chen, J., Li, S., Mills, J., Paparoditis, N., Rottensteiner, F., Sester, M., Toth, C. and Trinder, J. (2016) 'Information from imagery: ISPRS scientific vision and research agenda', *ISPRS Journal of Photogrammetry and Remote Sensing*, 115, pp. 3-21.

Cheng, L., Gong, J., Li, M. and Liu, Y. (2011) '3D Building Model Reconstruction from Multi-view Aerial Imagery and Lidar Data', *Photogrammetric Engineering & Remote Sensing*, 77(2), pp. 125-139.

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

Dahlke, D. and Linkiewicz, M. (2016) 'Comparison between two generic 3D building reconstruction approaches - point cloud based vs. image processing based', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41(B3), pp. 599-604.

Dal Poz, A.P. and Fernandes, V.J.M. (2016) 'Extraction of roof lines from high-resolution images by a grouping method', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41(B3), pp. 853-857.

de la Losa, A. and Cervelle, B. (1999) '3D Topological modeling and visualisation for 3D GIS', *Computers & Graphics*, 23(4), pp. 469-478.

Demir, N. and Baltsavias, E. (2012) 'Automated modeling of 3D building roofs using image and lidar data', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(4), pp. 35-40.

Dorninger, P. and Pfeifer, N. (2008) 'A Comprehensive Automated 3D Approach for Building Extraction, Reconstruction, and Regularization from Airborne Laser Scanning Point Clouds', *Sensors*, 8(11), pp. 7323-7343.

Durupt, M. and Taillandier, F. (2006) 'Automatic Building Reconstruction from a Digital Elevation Model and Cadastral Data : An Operational Approach', *Symposium of ISPRS Commission III*. Bonn, Germany, 20 - 22 September. ISPRS, p. 6.

Eireiner, S. (2004) *RLE de/encoding* (Version 1.0) [Computer program]. Available at: http://uk.mathworks.com/matlabcentral/fileexchange/4955-rle-de-encoding (Accessed: 09/03/15).

Emem, O. and Batuk, F. (2004) 'Generating precise and accurate 3D city models using photogrammetric data', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35(B4), pp. 431 - 436.

Engels, J., Arefi, H. and Halm, M. (2008) 'Generation of roof topologies using plane fitting with RANSAC', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B3a), pp. 119 - 126.

ESRI (2016a) *ASCII 3D to Feature Class* (Version 10.3) [Computer program]. Available at: http://pro.arcgis.com/en/pro-app/tool-reference/3d-analyst/ascii-3d-to-feature-class.htm.

ESRI (2016b) *Extract by Mask* (Version 10.3) [Computer program]. Available at: http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/extract-by-mask.htm.

Fan, H., Yao, W. and Fu, Q. (2014a) 'Segmentation of Sloped Roofs from Airborne LiDAR Point Clouds Using Ridge-Based Hierarchical Decomposition', *Remote Sensing*, 6(4), pp. 3284-3301.

Fan, H., Zipf, A., Fu, Q. and Neis, P. (2014b) 'Quality assessment for building footprints data on OpenStreetMap', *International Journal of Geographical Information Science*, 28(4), pp. 700-719.

Forlani, G., Nardinocchi, C., Scaioni, M. and Zingaretti, P. (2006) 'Complete classification of raw LIDAR data and 3D reconstruction of buildings', *Pattern Analysis and Applications*, 8(4), pp. 357-374.

Forlani, G., Roncella, R. and Nardinocchi, C. (2015) 'Where is photogrammetry heading to? State of the art and trends', *Rendiconti Lincei*, 26(1), pp. 85-96.

Forstner, W. (1999) '3D-city models: automatic and semiautomatic acquisition methods', *Photogrammetric Week 99*. Stuttgart, Germeny. Wichmann Verlag, pp. 291 - 303.

Freeman, H. (1961) 'On the Encoding of Arbitrary Geometric Configurations', *IRE Transactions on Electronic Computers*, 10(2), pp. 260-268.

Fu, C.-S. and Shan, J. (2008) 'Three-dimensional Building Reconstruction Using Point Primitives: a Geometric Graph Approach', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B2), pp. 203 - 208.

Gehrke, S., Morin, K., Downey, M., Boehrer, N. and Fuchs, T. (2010) 'Semi-global matching: An alternative to LIDAR for DSM generation', *The 2010 Canadian Geomatics Conference and Symposium of Commission I, ISPRS Convergence in Geomatics – Shaping Canada's Competitive Landscape*. Calgary, Alberta, Canada, June 15-18.

Ghassoun, Y., Löwner, M.-O. and Weber, S. (2015) 'Exploring the Benefits of 3D City Models in the Field of Urban Particles Distribution Modelling—A Comparison of Model Results', in Breunig, M., Al-Doori, M., Butwilowski, E., Kuper, V.P., Benner, J. and Haefele, H.K. (eds.) *3D Geoinformation Science: The Selected Papers of the 3D GeoInfo 2014*. Cham: Springer International Publishing, pp. 193-205.

Goebbels, S. and Pohle-Fröhlich, R. (2016) 'Roof reconstruction from airborne laser scanning data based on image processing methods', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3(3), pp. 407-414.

Goodheart-Willcox (2009) 'Chapter 19 - Roof Designs'. http://www.slideshare.net/stootypal/roof-types/7: SlideShare. Available at: http://www.slideshare.net/stootypal/roof-types/7.

Grigillo, D. and Kanjir, U. (2012) 'Urban object extraction from digital surface model and digital aerial images', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(3), pp. 215-220.

Gröger, G. and Plümer, L. (2012) 'CityGML – Interoperable semantic 3D city models', *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, pp. 12-33.

Gruber, M. and Wiechert, A. (2009) 'UltraCam Xp, the new digital aerial camera system by Vexcel Imaging/Microsoft', *Proceedings of the Remote Sensing and Photogrammetry Society (RSPSoc) Annual Conference 2009*. Leicester, UK, 8 - 11 September. RSPSoc.

Gruen, A. (2012) 'Development and Status of Image Matching in Photogrammetry', *The Photogrammetric Record*, 27(137), pp. 36-57.

Gruen, A. and Wang, X. (1998) 'CC-Modeler: a topology generator for 3-D city models', *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(5), pp. 286-295.

Haala, N. (2011) 'Multiray Photogrammetry and Dense Image Matching', *Photgrammetric Week '11*. Universtiy of Stuggart, 5 - 9 September. Wichmann, pp. 185 - 197.

Haala, N. (2013) *Dense Image Matching Final Report*. EuroSDR.

Haala, N. (2014) 'EuroSDR EduServ 2014', *High denisty image matching*. Trento, Italy, 3rd March 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*, 2010(2), pp. 99-115.

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

Habib, A. (2008) 'Integration of LiDAR and Photogrammetric Data', in Shan, J. and Toth, C. (eds.) *Topographic Laser Ranging and Scanning*. Boca Raton, Florida: CRC Press, pp. 371-402.

Harris, C. and Stephens, M. (1988) 'A combined corner and edge detector', *Alvey vision conference*. University of Manchester, United Kingdom, 31st August - 2nd September. pp. 147 - 152.

He, Y., Zhang, C. and Fraser, C.S. (2013) 'A line-based spectral clustering method for efficient planar structure extraction from LiDAR data', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(5/W2), pp. 103-108.

Hebel, M. and Stilla, U. (2008) 'Pre-classification of points and segmentation of urban objects by scan line analysis of airborne LiDAR data', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B3a), pp. 105-110.

Helbich, M., Jochem, A., Mücke, W. and Höfle, B. (2013) 'Boosting the predictive accuracy of urban hedonic house price models through airborne laser scanning', *Computers, Environment and Urban Systems*, 39, pp. 81-92.

Henn, A., Gröger, G., Stroh, V. and Plümer, L. (2013) 'Model driven reconstruction of roofs from sparse LIDAR point clouds', *ISPRS Journal of Photogrammetry and Remote Sensing*, 76, pp. 17-29.

Herbert, G. and Chen, X. (2015) 'A comparison of usefulness of 2D and 3D representations of urban planning', *Cartography and Geographic Information Science*, 42(1), pp. 22-32.

Hirschmüller, H. (2008) 'Stereo processing by semiglobal matching and mutual information', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), pp. 328-341.

Hirschmüller, H. (2011) 'Semi-Global Matching - motivation, developments and applications', *Photogrammetric Week '11*. Stuttgart, Germany, 5 - 9 September.

Hirschmuller, H. and Bucher, T. (2010) 'Evaluation of digital surface models by semi-global matching', *DGPF 2010*. Vienna, Austria, 1 - 3 July. DGPF, pp. 571 - 581.

Honkavaara, E., Arbiol, R., Markelin, L., Martinez, L., Cramer, M., Bovet, S., Chandelier, L., Ilves, R., Klonus, S., Marshal, P., Schläpfer, D., Tabor, M., Thom, C. and Veje, N. (2009) 'Digital Airborne Photogrammetry—A New Tool for Quantitative Remote Sensing?—A State-of-the-Art Review On Radiometric Aspects of Digital Photogrammetric Images', *Remote Sensing*, 1(3), p. 577.

Huang, H., Brenner, C. and Sester, M. (2013) 'A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data', *ISPRS Journal of Photogrammetry and Remote Sensing*, 79, pp. 29-43.

INSPIRE (2013) *D2.8.III.2 Data Specification on Buildings – Technical Guidelines*. European Commission Joint Research Centre.

ISO (2016) *ISO/TC 211 - Geographic information/Geomatics*. Available at: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=54904 (Accessed: 07/09/2016).

Jacobsen, K., Cramer, M., Ladstädter, R., Ressl, C. and Spreckels, V. (2010) 'DGPF-project: evaluation of digital photogrammetric camera systems–geometric performance', *Photogrammetrie-Fernerkundung-Geoinformation*, 2010(2), pp. 83-97.

Jaw, J.J. and Cheng, C.C. (2008) 'Building Roof Reconstruction by Fusing Laser Range Data and Aerial Images', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(Part B3b), pp. 707 - 712.

Jazayeri, I., Rajabifard, A. and Kalantari, M. (2014) 'A geometric and semantic evaluation of 3D data sourcing methods for land and property information', *Land Use Policy*, 36, pp. 219-230.

Jiang, X. and Bunke, H. (1994) 'Fast segmentation of range images into planar regions by scan line grouping', *Machine Vision and Application*, 7(2), pp. 115-122.

Jochem, A., Höfle, B., Rutzinger, M. and Pfeifer, N. (2009) 'Automatic Roof Plane Detection and Analysis in Airborne Lidar Point Clouds for Solar Potential Assessment', *Sensors*, 9(7), pp. 5241-5262.

Juneja, M. and Sandhu, P.S. (2009) 'Performance evaluation of edge detection techniques for images in spatial domain', *International Journal of Computer Theory and Engineering,* 1(5), pp. 614 - 622.

Kaartinen, H. and Hyyppa, J. (2006) 'Evaluation of building extraction', *EuroSDR Project*, 50, pp. 9 - 77.

Kabolizade, M., Ebadi, H. and Mohammadzadeh, A. (2012) 'Design and implementation of an algorithm for automatic 3D reconstruction of building models using genetic algorithm', *International Journal of Applied Earth Observation and Geoinformation*, 19, pp. 104-114.

Kada, M. (2009) 'The 3D Berlin Project', *Photogrammetric Week 09*. Hannover, Germany, 7 - 11 September. pp. 331-340.

Kendall, A. (2010) *count_unique* (Version 1.5) [Computer program]. MathWorks. Available at: http://uk.mathworks.com/matlabcentral/fileexchange/23333-determine-and-count-unique-values-of-an-array.

Kim, K. and Shan, J. (2011) 'Building roof modeling from airborne laser scanning data based on level set approach', *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(4), pp. 484-497.

Kim, Z. and Nevatia, R. (2004) 'Automatic description of complex buildings from multiple images', *Computer Vision and Image Understanding*, 96(1), pp. 60-95.

Kolbe, T.H. (2009) 'Representing and Exchanging 3D City Models with CityGML', in Lee, J. and Zlatanova, S. (eds.) *3D Geo-Information Sciences*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 15-31.

Kraus, K. (2007) *Photogrammetry Geometry from Images and Laser Scans*. 2 edn. Berlin, New York: Walter de Gruyter.

Lafarge, F., Descombes, X., Zerubia, J. and Pierrot-Deseilligny, M. (2010) 'Structural Approach for Building Reconstruction from a Single DSM', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1), pp. 135-147.

Leberl, F., Bischof, H., Tock, T., Irschara, A. and Kluckner, S. (2010a) 'Aerial Computer Vision for a 3D Virtual Habitat', *Computer*, 43(6), pp. 24 - 31.

Leberl, F., Irschara, A., Pock, T., Meixner, P., Gruber, M., Scholz, S. and Weichert, A. (2010b) 'Point Clouds: Lidar versus 3D Vision', *Photogrammetric Engineering and Remote Sensing*, 76(10), pp. 1123-1134.

Leberl, F., Meixner, P., Wendel, A. and Irschara, A. (2012) 'Automated Photogrammetry for Three-Dimensional Models of Urban Spaces', *Optical Engineering*, 51(2), p. 021117.

Leica Geosystems (2016a) 'Airborne Imaging'. Available at: http://www.leica-geosystems.co.uk/en/Airborne-Imaging_86816.htm.

Leica Geosystems (2016b) 'Leica ADS100' Leica Geosystems *Airborne Digital Sensor - Airborne Evolution*. Switzerland. Available at: http://www.leica-geosystems.co.uk/downloads123/zz/airborne/ADS100/brochures-datasheet/Leica%20ADS100_DS_en.pdf.

Leica Geosystems (2016c) 'Leica DMC III' Leica Geosystems *Breaking New Ground. Always.* Switzerland.

Li, J. and Guan, H. (2011) '3D Building Reconstruction from Airborne Lidar Point Clouds Fused with Aerial Imagery', in Yang, X. (ed.) *Urban Remote Sensing: Monitoring, Synthesis and Modeling in the Urban Environment*. Chichester, UK: John Wiley & Sons, Ltd, pp. 75-91.

Lillesand, T., Kiefer, R.W. and Chipman, J. (2008) *Remote Sensing and Image Interpretation*. Sixth Edition edn. United States of America: Wiley.

Lopez-Molina, C., De Baets, B. and Bustince, H. (2013) 'Quantitative error measures for edge detection', *Pattern Recognition*, 46(4), pp. 1125-1139.

Löwner, M.O. and Gröger, G. (2016) 'Evaluation Criteria for Recent LoD Proposals for City-GML Buildings', *Photogrammetrie - Fernerkundung - Geoinformation*, 2016(1), pp. 31-43.

Löwner, M.O., Gröger, G., Benner, J., Biljecki, F. and Nagel, C. (2016) 'Proposal for a New LOD and Multi-Representation Concept for CityGML', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W1, pp. 3-12.

Maini, R. and Aggarwal, H. (2009) 'Study and comparison of various image edge detection techniques', *International Journal of Image Processing*, 3(1), pp. 1-11.

Maltezos, E. and Ioannidis, C. (2016) 'Automatic extraction of building roof planes from airborne lidar data applying an extended 3D randomized hough transform', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3(3), pp. 209-216.

Mannini, A. (2010) *Freeman Chain Code* (Version 1.0) [Computer program]. Available at: http://uk.mathworks.com/matlabcentral/fileexchange/29518-freeman-chain-code (Accessed: 14/03/15).

Markelin, L., Honkavaara, E., Peltoniemi, J., Ahokas, E., Kuittinen, R., Hyyppä, J., Suomalainen, J. and Kukko, A. (2008) 'Radiometric Calibration and Characterization of Large-format Digital Photogrammetric Sensors in a Test Field', *Photogrammetric Engineering & Remote Sensing*, 74(12), pp. 1487-1500.

Martin, A.F. and Robert, C.B. (1981) 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography', *Communications of the ACM*, 24(6), pp. 381-395.

MathWorks (2006) *Edge Detection* [Computer program]. MathWorks. Available at: http://uk.mathworks.com/help/images/ref/edge.html?searchHighlight=edge (Accessed: 13/06/16).

MathWorks (2016) *RGB2GRAY* (Version R2016B) [Computer program]. Available at: http://uk.mathworks.com/help/matlab/ref/rgb2gray.html?s_tid=gn_loc_drop.

MATLAB (2017) *Matrices and Arrays*. Available at: https://uk.mathworks.com/help/matlab/learn_matlab/matrices-and-arrays.html (Accessed: 09/04/17).

Maxfield, C. and Brown, A. (2006) *1-Bit, 8-Bit, 15-Bit, 16-Bit, 24-Bit, 30-Bit, or 32-Bit Color* [Image]. Available at: https://www.clivemaxfield.com/diycalculator/popup-h-console.shtml (Accessed: 23/04/2017).

McClune, A.P., Miller, P.E., Mills, J.P. and Holland, D. (2014) 'Automatic urban 3D building reconstruction from multi-ray photogrammetry', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(3), pp. 219-226.

Meidow, J., Hammer, H., Pohl, M. and Bulatov, D. (2016) 'Enhancement of generic building models by recognition and enforcement of geometric constraints', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3(3), pp. 333-338.

Melnikova, O. and Prandi, F. (2011) '3D Buildings extraction from aerial images', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(4/W19), pp. 193-197.

Microsoft (2014) 'Ultracam Eagle' Microsoft *UltraCam*. Austria: Microsoft.

Mikhail, E.M., bethel, J.S. and McGlone, C. (2001) *Introduction to Modern Photogrammetry*. New York, USA: John Wiley & Son.

Montero, A.S., Nayak, A., Stojmenovic, M. and Zaguia, N. (2009) 'Robust line extraction based on repeated segment directions on image contours', *IEEE Symposium on Computational Intelligence for Security and Defense Applications*. Ottawa, Canada, 08 - 10 July. IEEEXplore, pp. 1-7.

Moravec, H. (1980) *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Carnegie-Mellon University, Robotics Institute.

Nadernejad, E., Sharifzadeh, S. and Hassanpour, H. (2008) 'Edge Detection Techniques: Evaluations and Comparisons', *Applied Mathematical Sciences*, 2(31), pp. 1507 - 1520.

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

Nizar, A.A., Filin, S. and Doytsher, Y. (2006) 'Reconstruction of buildings from airborne laser scanning', *ASPRS 2006 Annual Conference*. Reno, Nevada, 1 - 5 May. ASPRS.

OGC (2012) *OGC City Geography Markup Language (CityGML) En-coding Standard*. Open Geospatial Consortium.

Omidalizarandi, M. and Saadatseresgt, M. (2013) 'Segmentation and Calssification of Point Clouds from Dense Aerial Image Matching', *The International Journal of Multimedia & Its Applications*, 5(4), pp. 33 - 50.

Ordnance Survey (2014a) 'How to create models of towns with building height', *GI explained*. Available at: https://www.ordnancesurvey.co.uk/blog/2014/06/how-to-create-models-of-towns-with-building-heights/ (Accessed: 21/10/14).

Ordnance Survey (2014b) *OS MasterMap Topography Layer user guide contents*. Ordnance Survey. [Online]. Available at: https://www.ordnancesurvey.co.uk/docs/user-guides/os-mastermap-topography-layer-user-guide.pdf.

Oude Elberink, S., Stoter, J., Ledoux, H. and Commandeur, T. (2013) 'Generation and dissemination of a national virtual 3D city and landscape model for the Netherlands', *Photogrammetric Engineering & Remote Sensing*, 79(2), pp. 147 - 158.

Oude Elberink, S. and Vosselman, G. (2009) 'Building reconstruction by target based graph matching on incomplete laser data: Analysis and limitations', *Sensors*, 9-8(8), pp. 6101-6118.

Oude Elberink, S. and Vosselman, G. (2011) 'Quality analysis on 3D building models reconstructed from airborne laser scanning data', *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2), pp. 157-165.

Paparoditis, N., Souchon, J.-P., Martinoty, G. and Pierrot-Deseilligny, M. (2006) 'High-end aerial digital cameras and their impact on the automation and quality of the production workflow', *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(6), pp. 400-412.

Parker, J.R. (2010) *Algorithms for Image Processing and Computer Vision*. Second edn. Indianapolis: Wiley.

Passini, R., Jacobsen, K. and Day, D. (2012) 'Accuracy and Radiometric Study on Latest Generation Large Format Digital Frame Cameras', *JACIE 2012 Civil Commercial Imagery Evaluation Workshop*. Fairfax, USA, 17 - 19 April.

Perera, G.S.N. and Maas, H.-G. (2014) 'Cycle graph analysis for 3D roof structure modelling: concepts and performance', *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 213-226.

Peternell, M. and Steiner, T. (2004) 'Reconstruction of piecewise planar objects from point clouds', *Computer-Aided Design*, 36(4), pp. 333-342.

Petrie, G. and Toth, C. (2008) 'Airborne and Spaceborne Laser Profilers and Scanners', in Shan, J. and Toth, C. (eds.) *Topographic Laser Ranging and Scanning*. CRC Press, pp. 29-86.

Piepereit, R., Schilling, A., Alam, N., Wewetzer, M., Pries, M. and Coors, V. (2016) 'Towards Automatic Processing of Virtual City Models for Simulations', *ISPRS Annals of the Photogrammetry Remote Sensing and Spatial Information Sciences*, IV-2/W1, pp. 39-45.

Prandi, F., Soave, M., Devigili, F., Andreolli, M. and De Amicis, R. (2014) 'Services Oriented Smart City Platform Based On 3d City Model Visualization', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(4), pp. 59-64.

Rau, J.-Y. and Lin, B.-C. (2011) 'Automatic roof model reconstruction from ALS data and 2D ground plans based on side projection and the TMR algorithm', *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6, Supplement), pp. S13-S27.

Rau, J.Y. (2012) 'A line-based 3D roof model reconstruction algorithm: TIN-merging and reshaping (TMR)', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(3), pp. 287-292.

Read, R. and Graham, R. (2002) *Manual of Aerial Survey*. 1 edn. Scotland: Whittles Publishing.

Reitinger, B., Sormann, M., Zebedin, L., Schachinger, B., Hoefler, M., Tomasi, R., Lamperter, M., Gruber, B., Schiester, G., Kobald, M., Unger, M., Klaus, A., Bernoegger, S., Karner, K., Wiechert, A., Ponticelli, M. and Gruber, M. (2012) 'UltraMap V3 – A revolution in aerial photogrammetry', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(B4), pp. 149-152.

Rothermal, M., Wenzel, K. and Fritsch, D. (2012) 'SURE: Photogrammetric surface reconstruction from imagery', *Low-Cost 3D: Sensors, Algorithms, Applications workshop*. Berlin, Germany, 6 - 7 December.

Rothermel, M. and Haala, N. (2012) 'Potential of Dense Matching for the Generation of High Quality Digital Elevation Models', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(4/W19), pp. 271-276.

Rottensteiner, F. (2000) 'Semi-Automatic Building Reconstruction Integrated in Strict Bundle Block Adjustment', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 33(B2), pp. 461 - 468.

Rottensteiner, F., Sohn, G., Gerke, M. and Wegner, J.D. (2013) *ISPRS Test Project on Urban Classification and 3D Building Reconstruction*. http://www.commission3.isprs.org/wg4/: ISPRS.

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

Rutzinger, M., Rottensteiner, F. and Pfeifer, N. (2009) 'A comparison of evaluation techniques for building extraction from airborne laser scanning', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2(1), pp. 11-20.

Sampath, A. and Shan, J. (2010) 'Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds', *Geoscience and Remote Sensing, IEEE Transactions on*, 48(3), pp. 1554-1567.

Sandau, R. (2010) *Digital Airborne Camera, Introduction and Technology*. Springer.

Sargent, I., Holland, D. and Harding, J. (2015) 'The Building Blocks of User-Focused 3D City Models', *ISPRS International Journal of Geo-Information*, 4(4), p. 2890.

Scharstein, D. and Szeliski, R. (2002) 'A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms', *International Journal of Computer Vision*, 47(1), pp. 7-42.

Scharstein, D., Szeliski, R. and Hirschmüller, H. (2016) *Middlebury Stereo Evaluation - Version 3*. Available at: http://vision.middlebury.edu/stereo/eval3/ (Accessed: 28/05/2016).

Sheppard, S.R.J. and Cizek, P. (2009) 'The ethics of Google Earth: Crossing thresholds from spatial data to landscape visualisation', *Journal of Environmental Management*, 90(6), pp. 2102-2117.

Sohn, G., Huang, X. and Tao, V. (2008) 'Using a Binary Space Partitioning Tree for Reconstructing Polyhedral Building Models from Airborne Lidar Data', *Photogrammetric Engineering & Remote Sensing*, 74(11), pp. 1425 - 1438.

Sohn, G., Jwa, Y., Jung, J. and Kim, H. (2012) 'An implicit regularization for 3D building rooftop modeling using airborne lidar data', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3(3), pp. 305-310.

Spreckels, V., Syrek, L. and Schlienkamp, A. (2010) 'DGPF-Project: Evaluation of Digital Photogrammetric Camera Systems–Stereoplotting', *Photogrammetrie-Fernerkundung-Geoinformation*, 2010(2), pp. 117-130.

Stoter, J., Ledoux, H., Zlatanova, S. and Biljecki, F. (2016a) 'Towards sustainable and clean 3D Geoinformation', in Kolbe, T.H., Bill, R. and Donaubauer, A. (eds.) *Geoinformationssysteme 2016: Beiträge zur 3. Münchner GI-Runde* Munich, Germany: Wichmann Herbert, pp. 100 - 113.

Stoter, J., Roensdorf, C., Home, R., Capstick, D., Streilein, A., Kellenberger, T., Bayers, E., Kane, P., Dorsch, J., Woźniak, P., Lysell, G., Lithen, T., Bucher, B., Paparoditis, N. and Ilves, R. (2015) '3D Modelling with National Coverage: Bridging the Gap Between Research and Practice', in Breunig, M., Al-Doori, M., Butwilowski, E., Kuper, V.P., Benner, J. and Haefele, H.K. (eds.) *3D Geoinformation Science: The Selected Papers of the 3D GeoInfo 2014*. Cham: Springer International Publishing, pp. 207-225.

Stoter, J., Vallet, B., Lithen, T., Pla, M., Wozniak, P., Kellenberger, T., Streilein, A., Ilves, R. and Ledoux, H. (2016b) 'State-of-the-art of 3D national mapping in 2016', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41(B4), pp. 653-660.

Stoter, J., Vosselman, G., Goos, J., Zlatanova, S., Verbree, E., Klooster, R. and Reuvers, M. (2011) 'Towards a National 3D Spatial Data Infrastructure: Case of The Netherlands', *Journal of Photogrammetry, Remote Sensing and Geoinformation Processing*, 6, pp. 405 - 420.

Suveg, I. and Vosselman, G. (2004) 'Reconstruction of 3D building models from aerial images and maps', *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(3–4), pp. 202-224.

Synder, G.I. (2012) 'The 3D Elevation Program--Summary of Program Direction: U.S. Geological Survey Fact Sheet 2012-3089' USGS. p. 2. Available at: http://pubs.usgs.gov/fs/2012/3089/.

Szeliski, R. (2010) *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P. and Koehl, M. (2007) 'Model-driven and data-driven approaches using LIDAR data: Analysis and comparison', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/W49A), pp. 87-92.

TerraSolid Limited (2015) *TerraScan User's Guide*. Available at: https://www.terrasolid.com/download/tscan.pdf (Accessed: 28/03/15).

Thompson, E.M., Greenhalgh, P., Muldoon-Smith, K. and Charlton, J.D., Michael (2016) 'Planners in the Future City: Using City Information Modelling to Support Planners as Market Actors', *Urban Planning*, 1(1), pp. 79 - 94.

Toronto City Planning (2015) *3D Massing*. Available at: http://www1.toronto.ca/wps/portal/contentonly?vgnextoid=d431d477f9a3a410VgnVCM10000071d60f89RCRD (Accessed: 07/05/16).

Toth, C. and Jóźków, G. (2016) 'Remote sensing platforms and sensors: A survey', *ISPRS Journal of Photogrammetry and Remote Sensing*, 115, pp. 22-36.

Tseng, Y.-H. and Wang, S. (2003) 'Semiautomated building extraction based on CSG model-image fitting', *Photogrammetric engineering and remote sensing*, 69(2), pp. 171-180.

Ulm, K. (2003) 'Improved 3D city modeling with CyberCity Modeler using aerial-, satellite imagery and laser scanner data', *WG V/6 International Workshop "Visualization and Animation of Reality-based 3D Models"*. Tarasp-Vulpera, Engadin, Switzerland, 24-28 February.

United Nations (2015) *Goal 11: Making cities inclusive, safe, resilient and sustainable*. Available at: http://www.un.org/sustainabledevelopment/cities/ (Accessed: 19/08/16).

Vallet, B., Pierrot-Deseilligny, M., Boldo, D. and Brédif, M. (2011) 'Building footprint database improvement for 3D reconstruction: A split and merge approach and its evaluation', *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(5), pp. 732-742.

van den Brink, L., Stoter, J. and Zlatanova, S. (2013) 'Establishing a national standard for 3D topographic data compliant to CityGML', *International Journal of Geographical Information Science*, 27(1), pp. 92-113.

Varduhn, V., Mundani, R.-P. and Rank, E. (2015) 'Multi-resolution Models: Recent Progress in Coupling 3D Geometry to Environmental Numerical Simulation', in Breunig, M., Al-Doori, M., Butwilowski, E., Kuper, V.P., Benner, J. and Haefele, H.K. (eds.) *3D Geoinformation Science: The Selected Papers of the 3D GeoInfo 2014*. Cham: Springer International Publishing, pp. 55-69.

Verma, V., Kumar, R. and Hsu, S. (2006) '3D Building Detection and Modeling from Aerial LIDAR Data', *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. New York, 17-22 June. IEEE Computer Society, pp. 2213-2220.

Vexcel Imaging (2016) *UltraCam Eagle Mark 2*. Available at: http://www.vexcel-imaging.com/products/ultracam-eagle/ (Accessed: 02/09/2016).

Vosselman, G. (2012) 'Automated planimetric quality control in high accuracy airborne laser scanning surveys', *ISPRS Journal of Photogrammetry and Remote Sensing*, 74, pp. 90-100.

Vosselman, G. and Dijkman, S. (2001) '3D building model reconstruction from point clouds and ground plans', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(3/W4), pp. 37-44.

Vosselman, G., Gorte, B.G.H., Sithole, G. and Rabbani, T. (2004) 'Recognising structure in laser scanner point clouds', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46(8), pp. 33-38.

Walter, V. (2014) *A Survey on state of the art of 3D Geographical Information Systems*. EuroSDR EuroSDR. [Online]. Available at: http://www.eurosdr.net/sites/default/files/uploaded_files/eurosdr_no64_b_0.pdf.

Wang, Y. (2012) 'Automatic extraction of building roofs from Pictometry's orthogonal and oblique images', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(B3), pp. 361-365.

Wolf, P.R., Dewitt, B.A. and Wilkinson, B.E. (2014) *Elements of Photogrammetry with Applications in GIS*. Fourth edn. United States of America: McGraw-Hill Education.

Wong, K. and Ellul, C. (2016) 'Using Geometry-Based Metrics As Part of Fitness-For-Purpose Evaluations of 3D City Models', *ISPRS Annals of the Photogrammetry Remote Sensing and Spatial Information Sciences*, IV-2/W1, pp. 129-136.

Woo, D.-M., Nguyen, Q.-D., Tran, Q.-D.N., Park, D.-C. and Jung, Y.-K. (2008) 'Building Detection and Reconstruction from Aerial Images', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37 (B3b), pp. 713 - 718.

Xiao, J., Gerke, M. and Vosselman, G. (2012) 'Building extraction from oblique airborne imagery based on robust façade detection', *ISPRS Journal of Photogrammetry and Remote Sensing*, 68, pp. 56-68.

Xiong, B., Oude Elberink, S. and Vosselman, G. (2014) 'A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds', *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 227-242.

Xiong, B., Oude Elberink, S. and Vosselman, G. (2016) 'Footprint map partitioning using airborne laser scanning data', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3(3), pp. 241-247.

Yan, J., Jiang, W. and Shan, J. (2012) 'Quality analysis on RANSAC-based roof facets extraction from airborne lidar data', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39(B3), pp. 367-372.

Yastikli, N., Bayraktar, H. and Erisir, Z. (2014) 'Performance Validation of High Resolution Digital Surface Models Generated by Dense Image Matching with the Aerial Images', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(1), pp. 429-433.

Zhang, W., Grussenmeyer, P., Yan, G. and Mohamed, M. (2012) 'Primitive-based building reconstruction by integration of lidar data and optical imagery', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(5/W12), pp. 7-12.

Zhang, W., Wang, H., Chen, Y., Yan, K. and Chen, M. (2014) '3D Building Roof Modeling by Optimizing Primitive's Parameters Using Constraints from LiDAR Data and Aerial Imagery', *Remote Sensing*, 6(9), p. 8107.

Zhou, Q.-Y. and Neumann, U. (2008) 'Fast and extensible building modeling from airborne LiDAR data', *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. Irvine, California, 5 - 7 November. ACM, p. 7.

Zlatanova, S., Rahman, A.A. and Pilouk, M. (2002) '3D GIS: current status and perspectives', *ISPRS Comission IV Symposium 2002: Geospatial Theory, Processing and Applications*. Ottawa, Canada, 9-12 July. ISPRS.