
Abstract

Automatic three-dimensional (3D) building model reconstruction using remote sensing data is crucial in applications which require large-scale and frequent building model updates, such as disaster monitoring and urban management, to avoid huge manual efforts and costs. Recent advances in the availability of very high-resolution satellite data together with efficient data acquisition and large area coverage have led to an upward trend in their applications for 3D building model reconstructions.

In this dissertation, a novel multistage hybrid automatic 3D building model reconstruction approach is proposed which reconstructs building models in level of details 2 (LOD2) based on digital surface model (DSM) data generated from the very high-resolution stereo imagery of the WorldView-2 satellite. This approach uses DSM data in combination with orthorectified panchromatic (PAN) and pan-sharpened data of multispectral satellite imagery to overcome the drawbacks of DSM data, such as blurred building boundaries, rough building shapes unwanted failures in the roof geometries.

In the first stage, the rough building boundaries in the DSM-based building masks are refined by classifying the geometrical features of the corresponding PAN images. The refined boundaries are then simplified in the second stage through a parameterization procedure which represents the boundaries by a set of line segments. The main orientations of buildings are then determined, and the line segments are regularized accordingly. The regularized line segments are then connected to each other based on a rule-based method to form polygonal building boundaries. In the third stage, a novel technique is proposed to decompose the building polygons into a number of rectangles under the assumption that buildings are usually composed of rectangular structures. In the fourth stage, a roof model library is defined, which includes flat, gable, half-hip, hip, pyramid and mansard roofs. These primitive roof types are then assigned to the rectangles based on a deep learning-based classification method. In the fifth stage, a novel approach is developed to reconstruct watertight parameterized 3D building models based on the results of the previous stages and normalized DSM (nDSM) of satellite imagery. In the final stage, a novel approach

is proposed to optimize building parameters based on an exhaustive search, so that the two-dimensional (2D) distance between the 3D building models and the building boundaries (obtained from building masks and PAN image) as well as the 3D normal distance between the 3D building models and the 3D point clouds (obtained from nDSM) are minimized. Different parts of the building blocks are then merged through a newly proposed intersection and merging process.

All corresponding experiments were conducted on four areas of the city of Munich including 208 buildings and the results were evaluated qualitatively and quantitatively. According to the results, the proposed approach could accurately reconstruct 3D models of buildings, even the complex ones with several inner yards and multiple orientations. Furthermore, the proposed approach provided a high level of automation by the limited number of primitive roof model types required and by performing automatic parameter initialization. In addition, the proposed boundary refinement method improved the DSM-based building masks specified by 8% in area accuracy. Furthermore, the ridge line directions and roof types were detected accurately for most of the buildings. The combination of the first three stages improved the accuracy of the building boundaries by 70% in comparison to using line segments extracted from building masks without refinement. Moreover, the proposed optimization approach could achieve in most cases the best combinations of 2D and 3D geometrical parameters of roof models. Finally, the intersection and merging process could successfully merge different parts of the complex building models.

Zusammenfassung

Eine automatisierte dreidimensionale (3D) Gebäudemodell Rekonstruktion unter Verwendung von Fernerkundungsdaten ist wichtig für Anwendungen, die umfangreiche und häufige Aktualisierungen von Gebäudemodellen erfordern, wie z.B. Katastrophenüberwachung und Stadtplanung, auch um großen manuellen Aufwand und Kosten zu vermeiden. Die jüngsten Fortschritte bei der Verfügbarkeit geometrisch sehr hochauflösender Satellitendaten in Verbindung mit deren effizienter Datenerfassung und großflächigen Abdeckung, haben zu einer deutlichen Verstärkung der Anwendungen für 3D-Gebäudemodell Rekonstruktionen geführt.

In dieser Dissertation wird ein neuartiger mehrstufiger hybrider automatischer 3D-Gebäudemodell Rekonstruktionsansatz vorgeschlagen, der Gebäudemodelle in Detailstufe 2 (LOD2) auf der Grundlage von digitalen Oberflächenmodellen (DSM) rekonstruiert, die aus sehr hochauflösenden (VHR) Stereobildern des WorldView-2 Satelliten generiert wurden. Dieser Ansatz verwendet DSM-Daten in Kombination mit orthorektifizierten panchromatischen (PAN) und pangeschärften (PS) Daten von multispektralen (MS) Satellitenbildern, um die Nachteile der DSM-Daten, wie beispielsweise unscharfe Gebäudegrenzen, zu kompensieren.

In der ersten Stufe werden die groben Gebäudegrenzen in den DSM-basierten Gebäudemasken verfeinert, indem die geometrischen Merkmale der entsprechenden PAN-Bilder klassifiziert werden. Die verfeinerten Grenzen werden dann in der zweiten Stufe durch ein Parametrisierungsverfahren vereinfacht, welches die Gebäudegrenzen durch einen Satz von Liniensegmenten repräsentiert. Anschließend werden die Hauptausrichtungen des Gebäudes festgelegt und die Liniensegmente entsprechend angeordnet. Die regularisierten Liniensegmente werden dann nach einem regelbasierten Verfahren zu polygonalen Gebäuderändern miteinander verbunden. In der dritten Stufe wird eine neuartige Technik vorgeschlagen, um die Gebäudepolygone in eine Reihe von Rechtecke zu zerlegen, unter der Annahme, dass Gebäude in der Regel aus rechteckigen Strukturen bestehen. In der vierten Stufe wird eine Modellbibliothek von Dachformen definiert, welche Flach-, Giebel-, Halbwaln-, Waln-, Pyramiden- und Mansarddächer umfasst. Diese Dachtypen wer-

den dann den Rechtecken nach einer Deep Learning basierten Klassifikationsmethode zugeordnet. In der fünften Stufe wird ein neuartiger Ansatz zur Rekonstruktion voll parametrisierter 3D-Gebäudemodelle entwickelt, der auf den Ergebnissen der vorangegangenen Stufen und dem normalisiertem DSM (nDSM) aus Satellitenbildern basiert. Schließlich wird ein neuer Ansatz zur Optimierung der Gebäudeparameter auf der Grundlage einer umfassenden Parameteroptimierung entwickelt. Schließlich wird ein neuartiger Ansatz zur Optimierung der Gebäudeparameter auf der Grundlage einer umfassenden Parameteroptimierung entwickelt. Hierbei wird der zweidimensionale (2D) Abstand zwischen den 3D-Gebäudemodellen und den Gebäudebegrenzungen (erhalten aus Gebäudemasken und PAN-Bild) sowie der 3D-Abstand zwischen den 3D-Gebäudemodellen und den 3D-Punktwolken (erhalten aus nDSM) minimiert.

Alle entsprechenden Experimente wurden auf vier Gebieten innerhalb der Stadt München mit insgesamt 208 Gebäuden durchgeführt und die Ergebnisse qualitativ und quantitativ ausgewertet. Die Ergebnisse zeigen, dass der vorgeschlagene Ansatz 3D-Modelle von Gebäuden recht genau rekonstruieren kann, und sogar komplexe Gebäuden mit mehreren Innenhöfen und mehreren Orientierungen gut abbildet. Darüber hinaus bietet der vorgeschlagene Ansatz einen hohen Automatisierungsgrad durch die begrenzte Anzahl der erforderlichen Dachmodelltypen und durch die Durchführung automatischer Parameterinitialisierungen und -optimierungen. Darüber hinaus verbessert das vorgeschlagene Optimierungsverfahren die DSM basierten Gebäudemasken um 8 % bezüglich der Flächengenauigkeit. Darüber hinaus wurden die Firstlinienrichtungen und Dachtypen für die meisten Gebäude korrekt erfasst. Die Kombination der ersten drei Stufen verbesserte die Genauigkeit der Gebäudebegrenzungen um 70 % im Vergleich zur Verwendung von Liniensegmenten, die aus Gebäudemasken die ohne Verfeinerung extrahiert wurden. Schließlich kann der vorgeschlagene Optimierungsansatz in den meisten Fällen eine sehr gute Kombination von 2D- und 3D Geometrieparametern von Dachmodellen berechnen. Bei komplexen Gebäudemodellen kann der vorgeschlagene Merging-Prozess die berechneten Gebäudeteile erfolgreich zusammenführen.

Acknowledgments

First, I would like to emphasize my gratitude to Prof. Dr. Peter Reinartz for giving me the opportunity to perform my PhD on this very interesting topic at the Remote Sensing Technology Institute (IMF) of DLR and carry out the thesis at the Osnabrueck University. He has supported, encouraged, and motivated me during my PhD study. All of his insightful and constructive comments have been great helps for progressing my research and improving my papers and also this dissertation.

I gratefully acknowledge the German Academic Exchange Service (DAAD) for financial support and the German Aerospace Center (DLR) for providing facilities and datasets during my PhD study.

I would like to extend my gratitude to Prof. Dr. Friedrich Fraundorfer for his invaluable scientific advice, guidance and stimulating discussions throughout the last two years of my PhD study. His brilliant insights, expertise, and guidance showed me a different view of my research.

I would like to thank Dr. Reza Bahmanyar for the favor of proofreading this dissertation and our papers. He gave me helpful comments about the language issues and the structure of my dissertation. His contribution concerning the refinement of the noisy building mask was very helpful to expand and enhance my methodology in this part of my research.

I would like to thank Dr. Hai Huang at the group of Applied Computer Science in Bundeswehr University for fruitful discussions and providing me a code to calculate the vertices of the building roof planes from their geometric parameters which have been used to reconstruct 3D parametric building roof model.

I would like to thank Dr. Janja Avbelj for her support by providing a code

to calculate PoLiS distance measurement. I would like to thank Dr. Pablo d'Angelo for the valuable discussions and generating all of the DSMs and datasets required for this research. I am also grateful to Thomas Krauß for the valuable discussions and providing building masks and footprints.

I would like to thank Peter Schwind, Dr. Franz Kurz, Dr. Emiliano Carmona, and Maximilian Langheinrich my colleagues in the IT team of IMF at DLR who selflessly gave their time for helping me.

Many thanks are given to all my colleagues and friends at DLR: Dr. Shiyong Cui, Dr. Jiaojiao Tian, Dr. Daniela Espinoza Molina, Ambar Murillo Montes de Oca, Dr. Xiangyu Zhuo, Ksenia Bittner, Dr. Nina Merkle, Zeinab Gharib Bafghi, Dr. Daniele Cerra, Dr. Kevin Alonso Gonzalez, Dr. Jagmal Singh, Dr. Jakub Bieniarz, Dr. Charoula Andreou, and Oliver Meynberg. Many thanks to Ms. Theresia Hantel and Ms. Sabine Knickl for their helps during my PhD research study at DLR.

Moreover, I would like to thank Adam Fathalrahman, Seyed Majid Azimi, and Dimitris Marmanis for helping me by valuable programming support and helpful discussions to go through the field of Deep Learning.

I would like to express my gratitude towards Dr. Hossein Arefi (Department of surveying and Geomatics engineering, Tehran university) for his invaluable help, advice, discussions, and motivation in the early stage of my PhD. His precious support helped me to get to know Germany very soon and have a good starting of my PhD research.

I would like to express my gratitude towards Prof. Dr. George Vosselman for accepting me as an academic visitor in his research group for two months, and I would like to thank Dr. Biao Xiong and Dr. Sander Oude Elberink for their guidance and suggestions during my stay at ITC Netherland.

I would like to thank my previous supervisor during my master's study, Prof. Farhad Samadzadegan (Department of surveying and Geomatics engineering, Tehran University) for supporting, motivating and encouraging me in the early stage to do PhD research in Germany.

I would like to thank all my friends in Munich for their love and support during my stay in Germany.

Most importantly, none of this would have been possible without the love, trust, and patience of my family and my husband's family. My parents and brothers, who have always been encouraging me on the ways I have chosen.

Last but not least, my deepest and heartfelt gratitude to my beloved husband Mohammad, who has always been by my side with his full understanding, faithful support, and unconditional love.

Contents

Abstract	iii
Zusammenfassung	v
Acknowledgements	vii
1 Introduction	1
1.1 Motivation	1
1.2 Research Challenges	4
1.3 Research Contributions	5
1.4 Related Works	7
1.4.1 Building Detection	7
1.4.1.1 Model-driven Approaches	7
1.4.1.2 Data-driven Approaches	8
1.4.2 Building Outline Extraction	11
1.4.2.1 Model-driven Simplification Approaches	12
1.4.2.2 Data-driven Simplification Approaches	13
1.4.2.3 Building Outline Regularization	15
1.4.3 3D Building Model Reconstruction	16
1.4.3.1 Model Reconstruction and Geometric Shapes	16
1.4.3.2 Model Reconstruction and Various Data Sources	17
1.4.3.3 Model Reconstruction and Different Automation Levels	19
1.4.3.4 Model Reconstruction and Data Processing Methods	19
1.4.4 Summary	29
1.5 Thesis Overview	30

2	Building Boundary Extraction	31
2.1	Building Mask Refinement	33
2.1.1	Morphological Filtering of the DSM-based Mask	34
2.1.2	Primitive Feature Extraction	35
2.1.3	Feature Classification	35
2.1.4	Mask Postprocessing	36
2.2	Building Outline Extraction and Simplification	36
2.2.1	Line Segment Extraction	37
2.2.2	Line Segment Regularization	39
2.2.3	Line Segment Adjustment	42
2.2.3.1	Mathematical Model for Line Segment Adjustment	42
2.2.4	Line Segment Intersection and Connection	47
2.3	Experimental Results and Discussion	48
2.3.1	Mask Refinement	50
2.3.2	Building Outline Extraction	53
2.3.2.1	Qualitative Evaluation	53
2.3.2.2	Comparison to an Existing Line Simplification Method	56
2.3.2.3	Quantitative Evaluation	57
2.4	Summary and Conclusion	60
3	Building Boundary Generalization and Decomposition	61
3.1	Building Boundary Generalization	63
3.2	Line Segment-based Building Polygon Decomposition	68
3.2.1	Creating Decomposition Rectangular Shapes	68
3.3	A Heuristic Approach for Selecting a Combination of Rectangles	70
3.4	Experimental Results and Discussion	72
3.4.1	Quality Evaluation of the Generalized Building Polygon	76
3.5	Summary and Conclusion	80
4	Roof Type Classification	81
4.1	Building Roof Library Definition	82
4.2	Dataset Generation Based on Roof Model Library	87
4.3	Image-based Roof Type Classification	89
4.3.1	Pretrained VGGNet	92
4.3.2	Pretrained ResNet	92
4.3.3	Fine-tuning Pretrained VGGNets 16/19 and ResNets (50/101/152) for Roof Type Classification	94
4.3.4	Pretrained CNN-based Features	96
4.4	A Conventional DSM-based Roof Type Classification	97
4.5	Experimental Results and Discussion	100
4.5.1	Experiment Setup	100
4.5.1.1	Training the Fine-tuning Approaches	101

4.5.1.2	Training SVM on the DeCAFs	101
4.5.1.3	Training SVM and RF on the RGB Features	102
4.5.2	Discussion of Results	102
4.6	Summary and Conclusion	109
5	3D Building Model Reconstruction	111
5.1	Geometrical Structure of the Roof Model	113
5.2	Enhancement of 2D Roof Model Rectangles	114
5.2.1	Enhancement of the Rectangle Parameters	114
5.2.2	Decomposition of the Rectangles based on the Roof Types	118
5.2.3	Detection of the Height Discontinuities	118
5.3	Selection and Initialization of Roof Model Parameters	119
5.3.1	Detection of Building Roof Model Orientation	119
5.3.2	Improvement of Roof Type Classification Results	121
5.3.3	Initialization of Roof Model Parameters	125
5.4	3D Building Model Reconstruction	126
5.4.1	Optimization	126
5.4.1.1	Finding the Interaction of the Rectangles in One Building Block	128
5.4.1.2	2D Optimization	130
5.4.1.3	3D Optimization	130
5.4.2	Intersection and Reconstruction	132
5.5	Result and Discussion	133
5.5.1	Qualitative Evaluation	133
5.5.1.1	Challenges and Difficulties	135
5.5.1.2	Comparing Height Maps and Profiles to LiDAR Data	148
5.5.2	Quantitative Evaluation	149
5.5.2.1	Roof Type Classification	153
5.5.2.2	Ridge Orientation Detection	153
5.5.2.3	3D Building Models	154
5.6	Summary and Conclusion	158
6	Summary, Conclusions, and Future Work	161
6.1	Summary	161
6.2	Conclusions	163
6.3	Future Work	165
A	Datasets	167
A.1	DSM from Stereo Satellite Imagery	167
A.2	DSM based LiDAR Data	171
B	Tables of State-of-the-Art	173

C Support Vector Machine (SVM)	177
C.1 Binary SVM Classifier	177
C.2 Soft-margin	179
D Algorithms	181
D.1 Improvement of Roof Type Classification Results	181
D.2 2D Optimization	181
D.3 3D Optimization	181
Acronyms	193
List of Symbols	202
List of Figures	209
List of Tables	212
References	229

1.1 Motivation

Having access to accurate three-dimensional (3D) building models is becoming highly demanded in various applications, such as urban planning, mobile communication network planning, mobile navigation systems, virtual tourism, cartography and disaster monitoring. In urban planning, 3D building models can support decision-making in urban development projects [1]. According to a report of the United Nations (UN) organization [2], more than 68% of the world's population will live in urban areas by 2050, which highlights the growing needs for the development of urban environments. In mobile communication network planning, 3D models can help to generate the wave propagation models to guide the waves between transmitters [3]. Navigation systems usually rely on two-dimensional (2D) maps in their interfaces, which sometimes cause difficulties for the users when linking the maps to the real environment. Thus, visualizing 3D building models in navigation systems could assist users in previsualizing the situations [4]. In addition, 3D building models can be employed in disaster monitoring, as they provides a comprehensive view of structures and activities leading to better and quicker responses [5].

The 3D building models have traditionally been obtained through topometric surveys [6] or based on a geographical information system (GIS) and photogrammetry approaches [7], which usually require huge manual effort and costs, and, therefore, are not applicable to large-scale 3D building model reconstructions. In addition, it is necessary to update urban models frequently which costs too much using manual approaches. Altogether, the interest in moving from manual to automatic 3D building model reconstructions has been growing over the last few years. The availability of remote sensing data has increased recently due to efficient and frequent data acquisitions and significant advances in the areas covered by aerial and satellite images. The existing 3D building reconstruction approaches using remote sensing data usually rely on optical imagery, laser data and radar data, together with their products, such as 3D point clouds of stereo image matching [6].

These approaches detect the existing buildings in scenes and extract them from the objects, such as cars, trees, roads and sidewalks, surrounding them. The extracted buildings are then reconstructed geometrically and their outlines are attributed, leading to 3D building models [8]. Dense 3D point clouds, which can be provided by airborne laser scanning techniques, such as light detection and ranging (LiDAR) or satellite/aerial stereo image matching, allow easy discrimination of elevated objects (e.g. buildings) from their ground-level neighbors [9]. The dense 3D point clouds are employed for building detection and extraction either directly or after being processed. Digital surface models (DSM), for example, which are derived by applying interpolations to irregular dense 3D point clouds, have proved to be more efficient than the direct use of dense 3D point clouds [10]. Symmetric neighborhoods around 3D structured data points can be defined using the DSM of LiDAR data and sharp linear features (e.g. building boundaries) can be extracted easily by image-processing algorithms, such as morphological operators and gradient-based edge detectors [10, 11, 12].

Although LiDAR provides accurate and direct 3D surface information, it cannot represent the semantic information of the captured surfaces due to its sensor specifications. In addition, the accuracy of the object boundaries extracted based on LiDAR data depends highly on the density of the point clouds. Dealing with these limitations, a number of previous works proposed combining accurate height information from LiDAR and accurate planimetric information from aerial images [13, 14, 15]. As an alternative to LiDAR data, highly overlapped aerial image sequences can be utilized to generate dense 3D point clouds by applying stereo image-matching techniques. It has been demonstrated that the qualities of the resulting point clouds and the point clouds obtained based on LiDAR data are similar [16]. In addition, it has been reported that the 3D point clouds derived from aerial imagery lead to a reliable DSM, even though they suffer from a low degree of automation due to occlusion, artifacts and errors during image-matching processes [17].

Despite producing dense and accurate 3D point clouds, the availability of LiDAR data and aerial images is limited to a few specific locations in the world because of restrictions in aerial data acquisition, authorization constraints and high acquisition costs. By contrast, satellite images provide a coverage of the whole globe with a high acquisition frequency and fair acquisition costs. Furthermore, the rich semantic information contents of satellite images can enhance building detection and classification results [18].

The availability of satellite images with high spatial resolution has led recently to an increasing interest in developing algorithms for 3D point cloud generation by applying stereo image-matching techniques to the satellite images. Although the 3D point clouds obtained from satellite images are generally less accurate than those obtained from LiDAR data or aerial images, they are accurate enough for the recognition and reconstruction of large buildings [19].

Extracting buildings and reconstructing their 3D models have been traditionally

performed manually by human operators, which is an expensive and time-consuming approach. Therefore, automatizing the whole process chain has been always a focus of attention of many researchers over the past few decades [20]. Designing fully automatic processing chains is a challenging issue due to the data types, scales being used, the complexity of the buildings and the level of details required. Therefore, semi-automatic approaches for 3D building model reconstruction are more feasible and have been addressed by a number of previous research works [21, 22, 23, 24]. In semi-automatic approaches, models are selected by operators and then computers automatically fit the model to the image or 3D point clouds [21, 25]. The operators can also intervene in the process by deleting errors and editing models. Although this intervention improves the results, it increases the total elapsed time for 3D building model reconstruction.

Many methods have been proposed by several recent research works to increase the automation level and decrease the human interventions [26, 27, 28]. The existing automatic 3D building model reconstruction approaches can be generally categorized into model-driven, data-driven and hybrid approaches. Model-driven approaches select the best fitting model to the 3D point clouds or DSM data from a building library to represent the 3D model of buildings [26]. The differences between the building model and 3D point clouds are calculated based on normal or vertical distance. Nevertheless, data-driven approaches extract geometrical components (e.g. lines, corners, planes) from 3D point clouds or DSM data provided and, subsequently, consider some geometrical topology between these components to form 3D building models [29]. Hybrid approaches integrate the two former approaches. In hybrid approaches, a data-driven approach generally extracts the building roof features, such as ridge/eave lines, and other prior knowledge for a subsequent model-driven approach [30].

To the best of our knowledge, satellite images have been used by only a few previous works for 3D building model reconstruction, despite their great advantages. In other words, using satellite images for 3D building model reconstruction is an active research topic which is still in its early stages. Additionally, there is still considerable space for improving fully automatic approaches. In this dissertation, we propose a hybrid automatic 3D building model reconstruction approach within the level of details (LOD)1 and LOD2 (while LOD1 provides prismatic models generated from the extrusion of building outlines in their 3rd dimensions, LOD2 includes roof details, structures and textures). Consequently, we use DSM data obtained from satellite imagery together with very high-resolution (VHR) panchromatic (PAN) and multi-spectral (MS) optical satellite images.

1.2 Research Challenges

We face a few challenges in reconstructing 3D building models based on satellite imagery, including the problems caused by the image characteristics and the complexities of scenes and building boundaries. The low pixel resolution in comparison to aerial images (approx. 0.5 to 1 m PAN and 2 to 4 m MS images) and low signal to noise ratio (SNR), and the wide baseline of the satellite images can particularly influence the accuracy of DSM data and, consequently, the accuracy of the automatic building reconstruction approaches. The wide baseline, for instance, may lead to accurate depth information through stereo matching but causes occlusion problems, especially in very dense and complex urban areas.

Noise, artifacts, imperfections and mismatched areas in DSMs cause some gaps particularly at the building boundaries [31, 32]. A number of previous works [33, 34] have applied interpolation methods to DSM data in order to fill the gaps; however, the results are still unsatisfactory due to the deficiencies of interpolation (e.g. decreasing sharpness of the building boundaries).

Furthermore, noise in DSM data makes discovering meaningful patterns from neighboring pixels difficult. The geometric parameters (e.g. the slopes and normal vectors), for instance, of the neighboring pixels in the same roof plane can be significantly different in some regions. In order to overcome these challenges, the previous works [12, 32, 18, 35, 36] recommend combining DSM data with VHR satellite images.

Urban scenes consist of a variety of objects, such as buildings, roads, sidewalks, bridges, hedges, cars, and even rivers and lawns. Therefore, discriminating the buildings from their surroundings to generate the building masks is a necessary and preliminary step for 3D building model reconstruction. Two-dimensional ground plans are typically used (if they are available for the areas) for this purpose or classification methods are applied to the scene image. The resulting masks derived from classification methods usually suffer from imperfections, especially at the building boundaries, because the objects neighboring the buildings, such as trees, disturb a satisfactory detection of the building boundaries.

After recognizing the buildings, their corresponding parameterized boundaries are determined as simplified building polygons, which are not always simple rectangular shapes with one main orientation. Although it is conventionally assumed that buildings have mutually perpendicular orientations, in reality, complex buildings usually have more than one main orientation, which might be non-perpendicular.

1.3 Research Contributions

The focus of this dissertation is to propose a fully automatic building extraction and 3D building model reconstruction approach using DSM data together with VHR optical satellite images (orthorectified PAN and pan-sharpened (PS) MS images) ¹, as shown in the first row of Figure 1.1. More specifically, a new building boundary extraction method and 3D building model reconstruction method are proposed to deal with the challenges mentioned previously:

- The building boundary extraction is performed on the building masks derived from DSM data. We propose an approach based on VHR satellite images (PAN images of the WorldView-2 satellite with 0.5 m resolution) to refine the building masks, particularly at the building boundaries (the second row and left side of Figure 1.1) to deal with the imperfections of the building masks. We develop a heuristic approach for extracting parameterized building polygons with multiple orientations from the refined building masks [37] (the third row and left side of Figure 1.1).
- We propose a method based on deep learning for the classification of the roof types into flat, half-hip, gable, hip, pyramid and mansard roof categories using PS image data and footprints which are obtained from digital cadastral maps as shown in roof type classification part in Figure 1.1. To generate training and test patches, the cadastral footprint mask is used instead of the refined building mask due to their great similarity, their availability in a large-scale and being easy to use. The results of roof types classification are a later stage used as preknowledge about the roof types and fused with height information obtained from a DSM in a Bayesian framework to make the final decision about the roof type (updated roof types in Figure 1.1). This knowledge combination helps one to take advantage of both spectral and spatial information in the classification of roof types [38].
- We propose a decomposition procedure for simplifying the reconstruction of complex roof shapes which may contain different roof types. More precisely, the buildings with complex shapes are decomposed into smaller blocks with simpler rectangular shapes (decomposition and 2D rectangle models in Figure 1.1). The roof type is identified for each rectangle and a building model is selected from a specified building model library (updated roof types in Figure 1.1). Overlaps between neighboring rectangles are considered to reconstruct intersection parts of neighboring models (connecting roofs models) and, consequently, have a continuous 3D model [39].

¹To know about satellite sensor characteristic and the DSM based stereo satellite imagery generation procedure in this dissertation please refer to Appendix A

- The previous contributions provide preknowledge of roof parameters and lead to the initialization of roof parameters. We introduce an approach for generating the 3D model of the buildings by performing an exhaustive search within the building parameter space to find the best fitting 3D models to reach the final 3D building model reconstruction. Searching the parameters is carried out in two steps for 2D and 3D parameters in sequence [39]. While 2D parameters determine the building outlines, the 3D parameters define the planes forming the 3D geometry of the buildings (fourth and fifth row of Figure 1.1).

1.4 Related Works

This section explains the three main steps of 3D building model reconstruction approaches briefly, including building detection, outline extraction and model reconstruction, regarding the related previous works in the domain. The existing methods are categorized into model- and data-driven methods for each step.

1.4.1 Building Detection

The preliminary steps in reconstructing 3D building models from remote sensing data are detecting and discriminating building areas from other existing objects in the image scenes, such as vegetation and roads.

1.4.1.1 Model-driven Approaches

In model-driven building detection approaches, a library consists of different building model shapes is defined. The buildings are then detected and extracted by fitting the building model from the library to the object in the scene of image or LiDAR data. Extracted building shapes are limited to the pre-defined shapes in the library.

Zhou *et al.* [40] proposed a model-driven approach in which aerial images and LiDAR data have been fused for building detection and extraction. In this approach, a library including eleven 3D primitives is defined. These primitives are projected in 2D to define aspects and aspect graphs. In the aspect graph, nodes present the house faces and arcs are described by the attributes which are formulated using coding regulations. The object boundaries are firstly extracted from the aerial images and then some descriptive features such as corners and lines are detected from the boundaries. Then the aspects, aspect graphs, and coding regulations are created on the basis of aerial image processing results and the results of LiDAR data processing. Next, the aspects are co-registered with LiDAR data using a correlation operator. By interpreting the aspects and aspect graphs, the houses are extracted and then fitted using a planar equation to generate digital building models. In this method,

data fusion is carried out in each process of house detection, which is called seamless fusion. Fusing the aerial images and LiDAR data allows this method to combine detailed information about the geometrical structure of buildings and the height information for better detecting and discriminating buildings. Nevertheless, this method fails to detect buildings when they are occluded by surrounding objects (e.g., trees) or covered by less than three LiDAR footprint shots due to computing three unknown fitting plane parameters.

Karantzalos *et al.* [41] developed a novel model-driven variational method to enhance usual building extraction methods by considering shadows and occlusions and consequently, narrowing the solution space by a novel grammar representation and an energy formulation. They fused images and Digital Elevation Models (DEM) to compute the grammar-based building priors. This method fails if the existing buildings are not sufficiently described by the building models within the used building library.

1.4.1.2 Data-driven Approaches

In data-driven building detection approaches, buildings are distinguished from other objects by performing image filtering, segmentation and classification. Since these approaches need no prior information of building structures, buildings with complex shapes can be also detected and extracted.

Some data-driven methods have been proposed which apply a height threshold to 3D point clouds to detect buildings as elevated objects. These methods extract digital terrain models (DTM) and subtract them from a DSM to obtain the elevated objects (e.g. buildings and trees). Weidner and Förstner [10] proposed a method based on morphological opening to approximate the DTM. This method subtracts the resulting DTM from a DSM to obtain a normalized DSM (nDSM) followed by a sequential height thresholding. The resulting binary image is then analyzed regarding the size of the connected components to remove the small segments, which are presumably trees. However, using only the object size is not sufficient for separating buildings from other objects. Some methods have been proposed which considered the building roofs as planar surfaces, where their surface normal vectors are constant or vary slowly. These methods compare the magnitude of the eigenvalues obtained from covariance analysis of the 3D point clouds using a planarity criterion to discriminate non-planar surfaces (e.g. trees, fences and poles) and planar surfaces (e.g. ground and roof planes). They then perform connected component analyses to separate the ground points from the roof plane points [42, 43].

Sun and Salvaggio [44] proposed a method to separate vegetation from roof planes and terrain by applying a graph-cut segmentation method to the normal vector calculated from the local neighborhood of the LiDAR data. The vegetation areas are then discarded, and the roof planes are extracted using a novel hierarchical

Euclidean clustering method.

A number of earlier works have used plane-based segmentation of 3D point clouds to separate the roof plane class from the other classes [27, 45, 46, 47]. However, these methods usually suffer from under-segmentation (e.g. considering a road connected to a lawn in the same segment). Dealing with this problem, Gerke and Xiao [48] proposed using images' spectral (color) information in addition to the height information of the 3D point clouds from LiDAR data. They further improved the segmentation results by selecting optimum features and performing a pre-segmentation process. Subsequently, the classification was performed using the segments obtained from the previous step. In another work, Gerke and Xiao [49] used the combination of height and spectral information further for segmentation-based region growing. In order to distinguish buildings from other objects, they employed a supervised classification, namely Random Trees (RTrees), and an unsupervised classification approach embedded in a Markov random field (MRF) framework with graph-cut for energy optimization.

Hermosilla *et al.* [50] proposed two approaches for automatic building extraction and detection using high spatial resolution images and LiDAR data based on thresholding and object-based classification. Building masks using the thresholding-based approach were detected by intersecting an nDSM and vegetation masks obtained from the normalized differenced vegetation index (NDVI). This process was followed by morphological opening and closing operators to eliminate small objects (i.e. noise). A region growing segmentation method was applied to the nDSM data using the object-based classification approach. A decision-tree and a boosting multi-classifier were then applied to the segmented areas to distinguish buildings from other objects using four different features (spectral, texture, shape and height) extracted from the multispectral image and nDSM data. The results showed that the thresholding-based building detection performed better than the object-based classification in different urban areas. Moreover, the former is less complex than the object-based classification due to fewer thresholding parameters. Additionally, the thresholding-based approach can detect buildings with a higher accuracy, especially when its parameters are adjusted based on the type of urban landscape.

Lafarge and Mallet [27] classified four classes, namely, buildings, vegetation, ground and clutter. Subsequently, suitable features, such as local non-planarity, elevation, scatter and regular grouping, were extracted from laser-based 3D point clouds or DSMs from multi-view stereo (MVS) imagery. An unsupervised classification was applied to the features mentioned previously based on a graph-cut optimization in a MRF framework. The use of an energy model and its simplicity allowed this approach to have promising results in large-scale urban areas. Additionally, this approach avoids local errors on the exterior border of classes (e.g. roof planes, trees and ground) by using a planimetric arrangement procedure which can arrange both geometric shapes (e.g. planes, cylinders, spheres and cones) and urban components (e.g. ground, vegetation and roofs). Despite its advantages, this ap-

proach does not perform well when the altimetric (height) accuracy and the density of the 3D point clouds are low.

Rottensteiner *et al.* [51] proposed an approach in which they detected the initial regions of the buildings by using a DTM and DSM generated from the LiDAR data and NDVI computed from MS information. They then applied an unsupervised classification based on the Dempster-Shafer theory [52, 53] to the initial building regions in order to detect building areas. The Dempster-Shafer theory can handle the union of classes and mixed pixel problem (i.e. different land covers have similar spectral information) better compared to other classification methods. Thus, this approach can achieve satisfactory results for the buildings with very inhomogeneous shapes. However, the accuracy of the results depends greatly on the size of the buildings and the resolution of LiDAR data.

Bittner and Reinartz [54] proposed a method which applies fully convolutional neural networks (FCN) to the nDSM of VHR satellite imagery. This method can extract complex buildings without any prior assumption about building sizes and shapes. Nevertheless, it cannot detect low rise buildings and sharp boundaries perfectly. As an improvement, using spectral information, such as PAN and MS images, can help to overcome these deficiencies [18]. Krauss *et al.* [19] proposed an approach in which they firstly discriminated the ground level and elevated objects using the nDSM data of VHR satellite imagery. The buildings were then extracted from the surrounding patterns and clutter based on advanced rule-based fuzzy spectral classification on PS VHR satellite images (WorldView-2) followed by a height thresholding.

In this dissertation, the data-driven approach proposed by Krauss *et al.* [19] is used at the beginning to detect buildings and generate building masks for further developments and processing. This approach is privileged due to its fast and fully automatic procedure in generating building masks. It takes advantage of PS VHR satellite imagery in the classification procedure and the height information from the DSM data generated based on the stereo matching of the PAN bands of the satellite images. Figure 1.2(b) shows an exemplary classification result of the proposed method applied to the WorldView-2 satellite imagery acquired from an area in Munich, Germany.

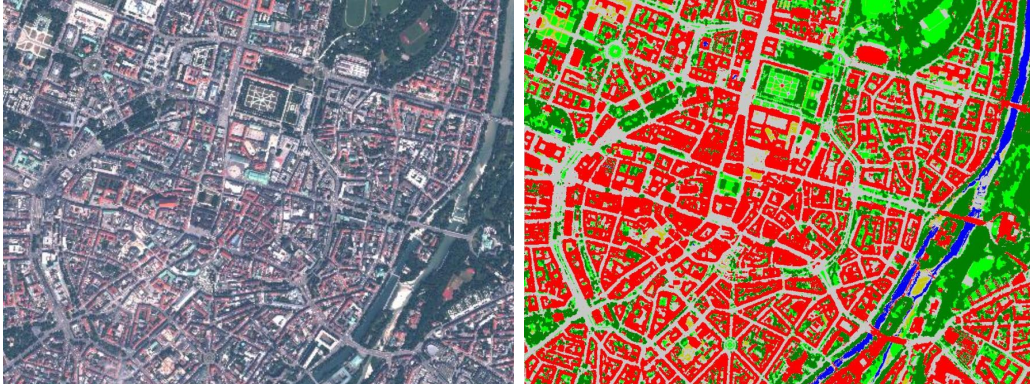


Figure 1.2: (a) Exemplary image covering an area of $2500m \times 2000m$ in central Munich acquired by the Worldview-2 satellite, (b) classification of the image using the proposed approach by Krauss *et al.* [19]

1.4.2 Building Outline Extraction

After detecting buildings, their 2D outlines are extracted to be used in 3D building model reconstruction. Building outline extraction comprises three main steps: 1) Detecting the points (pixels) on building boundaries, 2) reducing the detected points to retrieve only the relevant edges for generating the building polygons, and 3) regularizing the polylines to enforce parallelism and rectangularity which are typical constraints in man-made structures.

Detecting buildings based on DSM data results in a set of connected components. The pixels on the boundaries of their components can be traced to extract building polygons. Since the process of converting point clouds to DSM data requires an interpolation step, the height jumps on the building outlines are usually smoothed, which affects the accuracy of building boundary detection. In order to avoid the smoothing effect, depending on the data types (e.g. DSM and 3D LiDAR point cloud), different methods have been suggested, such as extracting boundaries directly from point clouds and fusing DSMs or point clouds with auxiliary data (e.g. high spatial resolution images). In order to extract boundary pixels directly from point clouds, the 3D points are projected onto a horizontal plane, and then a 2D Delaunay triangulation is employed to find the correspondences between the neighboring points. The triangle vertices which are not neighboring to any other triangle are then considered as the building edge points [55]. As an alternative method for triangulation, Matei *et al.*[56] and Verma *et al.* [42] used a ball-pivoting algorithm, which grows a seed triangle using a ball with a fixed radius. It pivots the front edge of the current active boundary until the geometry and topology of the whole object is captured. The results are the triangles generating the mesh. This algorithm can extract a complete boundary in a time-efficient manner; however, the generated mesh surfaces could suffer from holes in the places where sample point densities

are low. Similarly, the alpha-shape algorithm, which has been used by a number of previous works [46, 57, 58], generates building roof outlines.

In order to extract building boundaries by fusing DSMs and high spatial resolution images, firstly, the rough building boundaries from DSMs are extracted. Then the Hough transform [59], random sampling consensus (RANSAC [60]), active box [32] or other model fitting methods are applied to the building edges on the high-resolution images to extract line segments. Finally, the rough boundaries from DSMs are enhanced by the line segments [37, 32]. The building boundaries extracted usually contain numerous jagged edges. Therefore, as the next step, a polygon simplification method is employed to reduce the number of edges and keep the most relevant ones. The polygon simplification methods are categorized into model- and data-driven approaches.

1.4.2.1 Model-driven Simplification Approaches

These approaches simplify polygons based on rectangular structures. Gerke *et al.* [61] proposed a hierarchical method to fit rectilinear polygons to building boundaries. This method computes the main orientation of each building and fits a rectangle to the building outline based on its main orientation. After that, it iteratively subtracts the fitted rectangle from the building mask and fits a new rectangle to the remaining parts until the whole building area is covered by the rectangles. Although this method performs well for rectangular buildings, it fails for the non-rectangular ones. Dutter [62] proposed a similar method based on the minimum bounding rectangle (MBR). As opposed to the previous method, this method can fit polygons to the outlines of the buildings with T, L and Z shapes. Arefi *et al.* [63] proposed a method based on a combined MBR (CMBR) (which has been introduced mainly for the regularization of non-rectilinear polygons) and RANSAC which can compute the outlines of complex buildings, such as the buildings with non-perpendicular edges. Kwak [15] proposed a recursive MBR (rMBR), which is similar to the method proposed by the previous authors; however, it deals with boundary points instead of building masks. In addition, the iterative rectangle fitting can be stopped according the required level of details. All the methods mentioned previously need predefined main orientations of buildings. They also continue rectangle fitting until they cover the whole building area and remove small rectangles with areas smaller than a threshold. Janja Avbelj *et al.* [64] proposed an improvement to iterative MBR (iMBR) by introducing a cost function on the basis of Euclidean distance to compute the number of iterations automatically. Furthermore, resulting building polygons were refined by finding the rectilinear building polygons in the sense of a least-squares (LS) method.

Brédif *et al.* [65] developed a method for extracting building boundaries directly from DSMs without any building detection step. Rectangular building outlines were extracted from DSM data using a marked point process of rectangles. The optimum

configuration of the rectangles was then found based on reversible-jump Markov chain Monte Carlo (RJMCMC) [66] coupled with a simulated annealing algorithm. This optimization process minimizes an energy function which fits rectangle edges to the height discontinuities of the DSM data and penalizes overlaps of different objects. As a next step, the rectangle edges were elongated to intersect each other and form line segments. A graph of the line segments was then generated. Finally, the line segments supporting the rectangle edges were arranged using a graph-cut technique to create the final building outlines. Since the resulting boundaries are jagged, this method needs a post-processing step for the generalization and simplification of the boundary lines. The advantage of this method is that it requires no preprocessing step for building mask extraction.

Sirmacek *et al.* [32] proposed a method to extract building boundaries from DSM data of high and low resolution satellite images using a novel active shape detection approach. Their proposed method detects building junctions and endpoints from building skeletons. The buildings are then divided into various pieces and a box-fitting algorithm is run on each piece, starting from the building piece's center point. The active rectangular shape grows until an energy function, considering the distance between the rectangle and a set of previously extracted canny edges, is minimized. This algorithm can approximate building shapes even if the building edges are not completed or closed; however, it is very slow and not very satisfactory for large and complex buildings.

1.4.2.2 Data-driven Simplification Approaches

These approaches simplify polygons through reducing the vertices of the polygons' line segments. The Douglas Peucker (DP) algorithm [67] is a well-known polygon simplification method which can be employed for building outlines [58]. In order to simplify the outline of a building, it starts from the furthest connection between the building outline vertices. Among the other vertices, the vertex with a distance greater than a threshold to this connection is taken. This step is repeated till a complete simplification of the polygon is obtained. A problem with this method is the presence of outliers in the final result. Furthermore, if the original points are missing, this method cannot recover them since the vertices obtained are a subset of the original points. This can affect the final results greatly, especially at the critical points (e.g. corners).

Vosselman [68] used a sequential algorithm to approximate building boundaries. In this algorithm, firstly, the connected components of all the points of the building roof faces are determined. After that, a boundary line is defined based on the first two points of the contour of each connected component. This line is then updated using the succeeding points until the distance of the succeeding point to the line exceeds a threshold. A new line is then initialized by the last point of the line already extracted. The extraction of the new line is the same as the previous one

and its direction is perpendicular to the previous line. The final line segment is positioned between the outermost and average points. Kada and Wichmann [58] extracted the boundaries of building segments using a method based on 2D alpha-shapes. Since the boundaries extracted by the alpha-shapes method are very jagged, they are generalized and simplified by the DP algorithm. A quadratic LS adjustment is then applied to the remaining line segments and to the discarded points from the DP algorithm. Dorninger and Pfeifer [46] simplified the resulting polygons from alpha-shapes by considering the angular deviation between subsequent segments. If the angular deviation is smaller than a threshold, the segments are assigned to the same line segment. Afterwards, the mean angular deviation of the line segment is calculated and compared to the angular deviation of its subsequent segment. This process is terminated when the difference between the mean angular deviation and the angular deviation of the subsequent segment is larger than a threshold.

Sampath and Shan [69] proposed a modified version of the Jarvis March algorithm [70] based on the convex hull (CH) approach to trace the segment boundaries of sets of points. The algorithm starts by selecting the left-most point as a boundary point. The point which has the smallest angle with the previously selected boundary point within a local neighborhood is then selected as the next boundary point. This procedure iterates until the first point is visited again. The performance of this approach depends on the neighborhood used in the tracing process. Similar to the previous approach, Orthuber and Avbelj [71] used an iterative CH collapsing procedure to create a polygonal boundary of roof segments. Each line segment of the CH is refined based on a minimum distance measure. The collapsing stops when the line segment is shorter than a directional-based threshold. This threshold is defined based on the LiDAR point spacing in across-track and along-track sampling directions.

Similar to the previous approaches, Zhou and Neumann [72] considered a grid on point clouds. The LiDAR points which are close to the grid sides are then taken as boundary points. Because the segments boundaries are generated independently, a gap between neighboring planes can result in missing some of the planar segments. To overcome this problem, Rottensteiner *et al.* [11] used a Voronoi diagram of the planar segments to create boundary polygons. The advantage of this method [72] is that after marking cells as background or object, the grid is treated as a monochrome image and, therefore, a morphological operator can remove artifacts. Morgan and Habib [73] employed a global approach, such as the Hough transform algorithm which extracts straight lines through identification of the peaks in the Hough space. The straight lines are then intersected to form the building polygon.

As another global approach, Sester and Niedhart [74] used the RANSAC algorithm to extract building outlines from LiDAR data. Inspired by this work, Guercke and Sester [75] later proposed a new method which employs the Hough analysis instead of RANSAC for extracting line segments. Chen *et al.* [76] detected the initial building outlines by the canny edge detector [77] on the raster form of LiDAR point

clouds data. They then employed a 2D line-tracking technique with a length threshold to filter the short lines. The remaining lines (rough edges) were refined through the Hough transform algorithm which detects lines in the image space.

1.4.2.3 Building Outline Regularization

Man-made structures, such as buildings, typically follow certain principles and shape regularities. Nevertheless, the simplified building polygons are usually irregular and include short line segments with large direction variations. Therefore, a number of building boundary regularization methods have been proposed by previous works based on main orientations of building.

Vosselman [68] intersected the adjacent building faces obtained from LiDAR data to determine the main building orientations. If the adjacent faces have no intersection, the main orientation was extracted from building contours. Alharthy and Bethel [78] estimated main orientations of building using a histogram of angles computed from cross-correlation matching between building footprint images and a rotating template for every two degrees, starting from 0 to 360. All line segments are only allowed to be either parallel or orthogonal to the main orientations determined for building boundary regularization. Regarding line segment generalization, firstly, a histogram of the boundary points is obtained in parallel and orthogonal orientations to the main orientation of the building. The boundary points are then clustered as belonging to the same lines if they are more than a threshold in the limited space.

Sampath and Shan [69] used a threshold to classify all boundary lines as being parallel or perpendicular to the main orientations of building, which are commonly taken as the orientation of the longest line segments [79]. The boundary lines are then aligned according to their class using a LS adjustment algorithm. Similarly, Sester and Neidhart [74] improved the extracted line segments from the RANSAC method by applying a LS adjustment on all connected points. An overall LS estimation is performed to have a closed polygon by imposing right angle and parallelism constraints. Since the parallelism constraint is introduced, if the angles are adequately close, the non-rectangular building polygons can also be extracted. Dorninger and Pfeifer [46] regularized line segments by enforcing parallelism and orthogonality to the main orientations of building.

Zhou and Neumann [72] used a statistical analysis on the tangent directions for all boundary points in local neighborhoods and built direction histograms. The histogram peaks showed the principal directions of the buildings. The line segment and neighboring segment are then aligned to the principal orientation. Similarly, Kada and Wichmann [58] oriented the line segments using a histogram of orientations. A histogram of discrete directions in modulo 45° was generated in which the number of points composing each line segment was considered as weight values. The parallelism and perpendicularity to the main orientation of building was then enforced

to the lines whose angular deviations to one of these conditions (i.e. parallelism and perpendicularity) were below a threshold.

1.4.3 3D Building Model Reconstruction

After detecting buildings and extracting their boundaries, their 3D models are created using building model reconstruction approaches. The existing approaches can be categorized based on different aspects, such as the geometric shape of building models (e.g. generic or parametric models), data sources, their degree of automation (e.g. manual, semi-automatic or fully automatic) and their data processing strategies (data-driven, model-driven or hybrid).

1.4.3.1 Model Reconstruction and Geometric Shapes

Regarding the geometric shape, building model reconstruction approaches are categorized into parametric and generic ones. The parametric approaches use a number of predefined model primitives, including the rectangle, gable, hip, L-shape, I-shape and T-shape, with a fixed set of variable parameters, such as orientation, position, height, and length and width of rectangles. Since building shapes are usually irregular and complex, building model reconstructions using predefined model primitives is challenging. In contrast to the parametric approaches, for the generic ones, model primitives are not predetermined. In these approaches, modeling cues such as corners, edges and planes are extracted based on feature extraction algorithms, which allow them to reconstruct the buildings with irregular shapes. However, since there is no prior knowledge of the building shapes available, generic approaches perform poorly when building parts are occluded by nearby objects (e.g. trees, cars), which is a common problem in urban areas.

There are generally two groups of generic approaches that are based on prismatic or polyhedral models [55]. The prismatic-based approaches represent each building shape by a polygon and a height value for each polygon edge. These approaches cannot represent the roof structures. The polyhedral-based approaches represent roof structures as planar patches or curved surfaces [80]. They model internal geometric relations between roof components by intersecting the polygonal patches. Both groups of generic approaches model only a single building at a time and do not consider building blocks with internal geometric and topological relationships. Constructive solid geometry (CSG) is an approach for dealing with buildings which have complex shapes [25, 81]. It models complex buildings by integrating different simple model primitives of the parametric approaches. Building boundaries can be easily driven from the CSG models. Boundary representation (B-rep) is another approach to present complex building models which are defined by nodes and edges from intersecting planar faces [25, 81]. A set of topological relations is determined to connect faces, edges and vertices.

1.4.3.2 Model Reconstruction and Various Data Sources

The 3D building model reconstruction approaches extant use various data sources, such as aerial images, VHR satellite images, point cloud acquired from LiDAR data or driven from stereo image matching, and 2D ground plans. Some studies have been conducted using a combination of these data sources to use their advantages. Walls and shadows are usually utilised together with the direction of sun and camera models to reconstruct 3D building models from a single image [82]. These approaches depend greatly on selecting hypotheses based on line segments detected from a single image and the use of shadow information, which are not reliable. Therefore, these approaches usually fail to find reliable hypotheses in noisy datasets. Dealing with these issues, stereo or multiple images are used to generate 3D point clouds or DSMs through image-matching processes. Baillard *et al.* [83], for example, extracted 3D building features (i.e. lines and planes) through image matching to reconstruct 3D building models.

Building model reconstructions using the height values of DSMs generated by stereo image matching is influenced by occlusion, illumination (shadow and reflection), surface material or texture, object geometry and anomalies of the surface heights [33]. By contrast, LiDAR data provides high-density 3D point clouds that avoid the limitations of stereo matching and are beneficial for 3D planar patch detection; however, the discrete and irregular distribution of the points can cause the low geometrical accuracy of the building models in break lines, such as large discontinuities, ridge lines and the building boundaries [84]. In addition, due to the irregularity of its resulting 3D points, the information on the building edges is usually missing due to the irregularity of its resulting 3D points. In order to generate more accurate building models using LiDAR point clouds or DSMs of stereo imagery, they are usually integrated with additional data sources [12]. Furthermore, geometric assumptions on building shapes (e.g. parallelism, rectangularity, symmetry) are usually considered for solving the geometrical problems of the extracted building outlines caused by the limitations mentioned previously [74, 69]. Brenner [85] proposed utilizing GIS datasets from cadastral maps (CM) as auxiliary data in building reconstruction approaches. The ground plans not only help to detect outer roof face edges accurately, but also reveal information about the structure of a building's roofs. Suveg and Vosselman [86] proposed a 3D building reconstruction method that combines aerial images, 2D GIS datasets and domain knowledge. The authors in [81, 87] integrated LiDAR data and a 2D ground plan to determine the main orientations of building and improve the outlines of roof faces.

Although 2D ground plans reduce the search space in reconstruction processes, they suffer from some deficiencies, such as being outdated, incomplete and misaligned with other remotely sensed datasets. Dealing with these problems, it has been proposed to integrate DSM or LiDAR data with aerial or satellite imagery [11, 12, 31, 32, 37, 50, 76, 88, 89, 90].

Rottensteiner *et al.* [11, 51] computed edges from aerial images and used them to improve the geometric quality of the initial boundary polygons extracted from LiDAR data. Cehn *et al.* [76] reconstructed 3D building models from the accurate 3D edges provided by the Split-Merge-Shape (SMS) method in [79]. First of all, they extracted initial building edges from LiDAR data using a canny operator and a 2D line-tracking method. They then employed a Hough transform to compute straight edges within the image space. Finally, they obtained 3D edges in the object space based on exterior orientation parameters, given the image coordinates and the height information from 3D planes. Awrangjeb *et al.* [89] integrated LiDAR data and MS imagery for automatic 3D roof extraction. They extracted lines (e.g. roof edges and ridge lines) from the gray scale of VHR ortho-aerial imagery. The lines extracted were then used as baselines to detect the nearby LiDAR points of neighboring planes. Using the non-ground LiDAR points close to the baselines, a robust region seed was selected for each plane and its nearby region was iteratively grown to extract the complete roof plane. Finally, a rule-based procedure was applied to remove the planes belonging to non-building objects. Habib *et al.* [88] integrated aerial imagery and LiDAR data for reconstructing 3D building models. They adjusted initial model primitives derived from LiDAR data by detecting edges from stereo aerial imagery. Sirmacek *et al.* [32] proposed a novel active shape detection to extract building boundaries from the DSM of satellite imagery. They also extracted ridge lines from DSMs by calculating derivatives over DSMs. The Harris corner detection algorithm was then applied to the PAN images of the roof (except the ridge lines) to extract the roof polygons. Arefi *et al.* [12] used orthorectified PAN images to improve the ridge lines derived from the DSM of satellite imagery. They extracted surface normal and regional maxima from DSMs and edge information by applying the canny edge detector on PAN images.

Inspired by the previous approaches, a new parametric CSG based approach is proposed in this dissertation to reconstruct 3D building model from DSMs of satellite imagery which has low quality. Additional data sets such as high spectral and spatial resolution satellite imagery are used to overcome the deficiencies of the DSM in 3D building modeling. This approach refines building masks based on high spatial resolution MS satellite images (e.g. WorldView-2 images). Subsequently, it employs a novel building boundary delineation method to extract building polygons from the refined building masks [37]. In addition, a new decomposition method is proposed to decompose complex building shapes into rectangular shapes using the extracted building outlines of the previous step. After that, the roof types are distinguished based on a deep learning classification method applied to PS images. Furthermore, the ridge lines of the buildings are detected during the model fitting process based on DSMs and using the neighborhood relations between models.

1.4.3.3 Model Reconstruction and Different Automation Levels

Developing fully automatic approaches is in high demand for 3D building model reconstruction. Although a number of fully automatic 3D building model reconstruction approaches have been proposed recently [26, 91], these approaches do not usually achieve promising results due to the data types, the scale being used, the object complexities and the level of details required. They typically require pre- and postprocessing steps. Therefore, various semi-automatic approaches have been proposed which allow users to intervene manually in failure cases by, for example, selecting building model types [21, 24].

1.4.3.4 Model Reconstruction and Data Processing Methods

The 3D building model reconstruction approaches can be categorized into model-driven, data-driven and hybrid approaches. Model-driven approaches define a library of parametric models. The topological relations between model primitives are fixed and their geometrical relations are evaluated through fitting the models into selected point clouds. By contrast, data-driven approaches do not make any assumption about building shapes, which make them more flexible for reconstructing complex building models compared to the model-driven ones. They only consider a number of constraints and rules on rectangularity and parallelism [29, 92]. A drawback of data-driven methods is that any incompleteness of point clouds arising from shadow, occlusion or any other missing information, results in incorrect models. In practice, hybrid 3D building model reconstruction approaches are normally used to combine the flexibility of data-driven approaches and the robustness of model-driven approaches [8, 30].

1.4.3.4.1 Model-driven Methods

Model-driven approaches use parameterized model libraries and determine the best fitting roof models to point clouds. These approaches normally utilize additional data, such as a footprint, to decompose the roofs into the basic ones for the roofs with complex shapes. Furthermore, building footprints can help to localize the region of interest on images and point clouds to reduce the search space and focus on building regions. They also help the decomposition of complex roofs by providing hypotheses. If the footprint is not accurate enough, the building boundaries extracted, as described in Section 1.4.2, are employed [93].

In footprint decomposition, complex roofs are decomposed into simple rectangular structures, the so-called building primitives. Position, orientation and horizontal extension of roof primitives are defined by the rectangle, and type, height and slope of the building primitives need to be determined [81]. Suveg and Vosselman [86] performed footprint decomposition through partitioning and selecting the best par-

tioning based on the minimum description length (MDL) principal. Kada and McKinley [94] proposed a method to decompose building footprints into small sets of nonintersecting quadrilateral-shaped polygons, the so-called cells. Adequate subset of lines which are long enough in an assumed buffer are found for their proposed methodology and extended infinitely. Later, Henn *et al.* and Zheng *et al.* [95, 96] used a similar footprint decomposition method. Vallet *et al.* [97] introduced a decomposition method based on minimizing an energy function in the roof area with height discontinuities using DSMs of aerial imagery. They also proposed a greedy search to merge neighboring cells with a low gradient in their common edges. Lafarge *et al.* [98] decomposed the rectangles extracted [99] during the regularization process by transforming neighboring rectangles into sets of connected quadrilateral or triangles by incorporating neighboring rectangles. They then found the best configuration of the neighboring rectangles by the simulated annealing technique. The final quadrilaterals are more partitioned by detecting the height discontinuities from their inner DSM. Arefi and Reinartz [12] decomposed footprints into rectangles based on ridge lines detected from satellite nDSMs and canny edge points of orthorectified PAN images. Zheng *et al.* [30] decomposed building footprints into sub-footprints by detecting the step edges from canny points on the nDSM of LiDAR data. They then decomposed the sub-footprints by the ridge lines identified using watershed analysis and stream order algorithms from the nDSM of LiDAR data and VHR aerial images.

Parametric models describe roof primitives in libraries. They are created by using a few parameters, as shown in Figure 1.3. Suveg and Vosselman [86] defined a set of basic building roof primitives, such as flat, symmetrical gable and non-symmetrical gable roofs. They reconstructed the buildings with complex roofs by integrating multiple primitives using CSG tree fitting. Lafarge *et al.* [26] developed a rather large library to enhance its generalization ability and reconstruct buildings with a wide range of roof shapes. Their library consisted of two parts: The roof shapes (including monoplane, multiplane and covered roofs with specific paramters) and the ending form of the roofs (including hipped and straight). They also presented the roof orientation with respect to the 2D outlines [98]. In order to reconstruct the buildings with complex roof shapes, they proposed a method based on the structural concept which put the primitives in a way similar to the LEGO kit. Poullis and You [100] defined nine roof primitives for a single building, such as flat, gable, hip, pyramid, mansard and saltbox, by varying the scales of width and length of building rectangles. They also introduced a flexible polygonal primitive for reconstructing the buildings with complex roof shapes. Since this method optimizes the model for the entire building as a single object, it reduces the number of parameters and results in watertight roofs. Huang *et al.* [101] defined a simple primitive library with a low number of parameters to have more efficient reconstructions. Their library also contained planar shapes with a rectangular footprint. Their library contained single-plane roofs (e.g. flat and shed roofs)

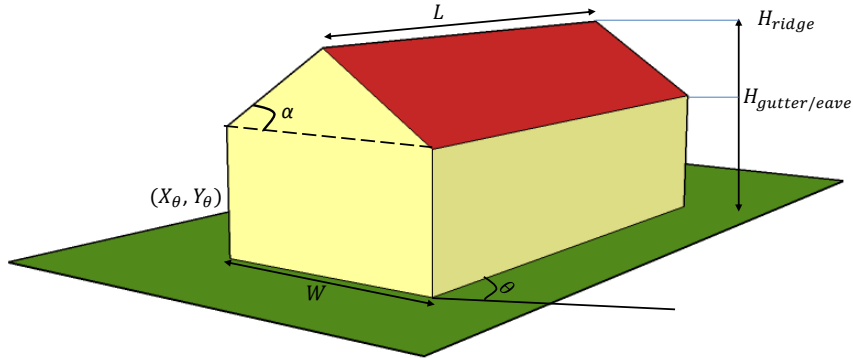


Figure 1.3: Parametric gable model (coordinate origin (X_θ, Y_θ) , orientation (θ) , slope of roof plane (α) , length (L) , width (W) , heights of gutter/eave $(H_{gutter/eave})$ and ridge lines (H_{ridge}))

and multipane roofs, such as hip, pyramid, half hip, gable, asymmetric, mansard, gambrel, Dutch gable and salt-box. They also employed a merging strategy to reconstruct non-rectangular footprints. Henn *et al.* [95] proposed a hierarchical structured roof model with prototypical roofs, as defined in City Geography Markup Language (CityGML) LOD2. Hierarchical structured roof models start from single and multipanes. Flat and shed roofs are located in the single plane category and gable, hip and pyramid roofs are located in the multipane category. The 2D parameters of the model, such as position and orientation, are computed from footprints. They used the minimum number of points in each plane for the initialization of the 3D parameters, such as heights and slope of the roof. A gable roof, for instance, can be determined by two points with different heights. Simple models from a defined library are selected for each cell obtained from footprint decomposition for reconstructing complex roofs [94]. Later, Zheng *et al.* [96] introduced a similar roof library which contained seven roof types in a hierarchical manner. In this library, the roofs are categorized into single-plane roofs (including flat roof) and multi-plane roofs such as gable, cross gable, intersecting, hip, half hip and pyramid roofs. Zheng *et al.* [30] defined a library which contained flat and shed roofs, cone and cylinder roofs and six commonly seen types of sloped roofs, such as gable, cross-gable, intersecting, hip, half-hip and pyramid roofs. Ngattem *et al.* [102] introduced a library of the gable, hip and mansard roofs.

Roof model selection is the process of fitting models into point clouds and selecting the most appropriate model from a library, which minimizes a cost function. Several methods have been proposed for model selection and estimation. Mass and Vosselman [103] proposed a new method for gable roof reconstruction by computing their parameters using the analysis of invariant moments of the 3D point clouds of LiDAR data. The information on the roof type and shape parameters is computed

by using the heights of the point clouds as weight functions in moment equations. Haala *et al.* [81] estimated roof plane parameters by segmenting the DSM of aerial images and analyzing the surface normals and ground plane orientations of the segments. They determined the roof types according to the number of segments. Zheng *et al.* [30] used a similar approach, subsequently, for roof type identification, in which the root mean square error (RMSE) between the DSM and corresponding points from the candidate roof model determines the quality of reconstruction. Suveg and Vosselman [86] proposed a method based on a combination of aerial images, 2D GIS databases and domain knowledge. They generated a number of hypotheses by fitting models to images using an iterative LS method and CSG tree, where the hypotheses were the nodes of the tree. The model parameters were estimated through the fitting process and the model with the maximum mutual information to the image was selected. Kada and McKinley [94] determined the roof type of each resulting cell from the decomposition step by computing the percentage of normal vector orientations from LiDAR points of each cell and comparing them to the possible shapes of roof faces. Poullis and You [100] presented an extendable parameterized geometric primitive for the automatic identification of building roof types in the modeling process from airborne LiDAR data. The roof model parameters were determined using a nonlinear bound-constraint minimization. During this optimization, a Gaussian mixture model (GMM) was used to detect and exclude outliers from the fitting plane, where the parameters of GMM were estimated using an expectation-maximization (EM) algorithm. Lafarge *et al.* [26] proposed a stochastic method for reconstructing 3D building models from the DSM of satellite imagery (PLEIADES satellite data simulations with resolutions of 0.7 m). They used a Bayesian algorithm based on RJMCMC to decide the building model which best fitted the DSM data. Huang *et al.* [101] utilized generative statistical models to reconstruct 3D building models from LiDAR data. The method finds the optimum combination of parameters by a stochastic search. Nguatem *et al.* [102] proposed a method to fit building models to 3D point clouds using sequential importance sampling (SIS). The model fitting was performed within a Bayesian framework and the solutions were found using a sequential Monte Carlo (SMC) approach. Henn *et al.* [95] proposed a strategy for 3D building reconstruction from a small number of LiDAR data points. This method estimates roof parameters by fitting the roof models and estimating their parameters by M-estimator sample consensus (MSAC). It determines the most probable roof model by a support vector machines (SVM). Zheng and Weng [96] proposed a method based on LiDAR data and building footprints. They computed some morphological and physical parameters from each cell resulting from the building decomposition step. They then applied a decision tree-based classifier to these statistical features to classify the building footprints into seven roof types. According to the roof type, they calculated the roof model parameters based on the statistical moments of the points within the cells.

1.4.3.4.2 Data-driven Methods

Data-driven approaches extract roof segments from data and then delineate building boundaries. They then determine the topologies of the adjacent segments by detecting ridge lines, step edges and intersection lines. Next, they generate roof topology graphs in which the nodes are segments and the edges are labeled according to the topology determined between the segments. Finally, they reconstruct the 3D roof models based on the topology graphs. Data-driven approaches use subsequent regularization steps to ensure the simplicity, parallelity and orthogonality of the final models. In contrast to the model-driven approaches, the data-driven ones are highly flexible in reconstructing complex buildings and rooftop superstructures given dense enough datasets. Nevertheless, the reconstruction quality depends on the algorithms being used for the roof plane segmentation. Furthermore, these approaches are sensitive to occlusions.

Most of the efforts in data-driven roof model reconstruction have been focused on the roof plane segmentation [27, 42, 87]. The point cloud segmentation methods, such as surface growing [58], triangulation irregular network (TIN) growing [103, 73, 71] and surface fitting techniques [104], turn roof surface points into planar, cylindrical and spherical structures [105]. Surface growing methods start from selected seed points and compare their properties to those of their adjacent points, expanding the region of planar segments around the seed points to the similar adjacent points. According to how the seed points, the adjacent points and the similarity between them are selected and calculated, various segmentation methods have been proposed in the literature [11, 43, 44, 87].

Vosselman and Dijkman [87] proposed a surface growing method to segment the building point cloud into homogenous patches called roof planar segments. In this method, 3D Hough transform is applied to the neighboring points of a random point. A seed surface is then determined where the neighboring points of the random point are on a plane (planar). The seed surface is extended by the adjacent points lying at a predefined distance from the plane. The surface is grown iteratively until no more points can be added to the seed surface. The parameters of the final plane are calculated by fitting a plane to all the seed surface points using a LS technique. Elberink and Vosselman [23] and Xiong *et al.* [28] used this method to obtain roof planes from LiDAR data points. Rottensteiner *et al.* [11] suggested extracting roof planes initially by means of region growing segmentation and considering the local homogeneity of surface normal vectors of DSMs. Thereafter, co-planar roof segments were merged based on the Fisher distribution (F-distribution) ratio for each pair of neighboring planes. Verma *et al.* [42] proposed a method which fits the planar patches to the roof points identified and merges them based on the normal of the surface patches.

Dorninger and Pfeifer [46] proposed a new method that selects the seed points

by applying a hierarchical clustering algorithm to the feature space defined by four parameters of local regression planes. The points are added to a cluster if their distances to the fitted plane are within a certain threshold. Sampath and Shan [43] analyzed the eigenvalues of each building points in their Voronoi neighborhood to calculate the normal vectors and separate LiDAR points into planar and non-planar points. A fuzzy k-means clustering is then applied to the normal vector of the planar points. Finally, parallel and co-planar segments are separated based on the connectivity and distance conditions. Kada and Wichmann [58] proposed selecting seed points randomly and with a low RMSE distance from a locally estimated plane for them. The seed surface is then grown if the perpendicular distance of the adjacent points to the plane and the angle differences between the segment's normal vector and the locally estimated normal of the point are below a certain threshold. Sun and Salvaggio [44] proposed selecting seed points with the smallest curvature values. The difference between the direction of normal vectors of the seed points and their neighboring points are calculated to grow roof planar regions.

Triangulation irregular network (TIN) growing is a segmentation method which, in contrast to the surface growing methods, allows the segments to grow into continuous shapes. Mesh models are then combined with region growing algorithms to extract planar patches. In contrast to the surface growing algorithms, TIN-based region growing algorithms start from a triangle instead of a seed point, which speeds up the region growing algorithm, since each triangle consist of several points with the same label [71].

Surface fitting techniques, in which RANSAC and Hough transform are the most well-known methods, are also used for segmentation. The RANSAC is an iterative and non-deterministic algorithm which can fit the best planes to 3D point clouds. It has been suggested by a number of previous works for detecting roof planes [29, 85, 104, 106, 107, 108, 109, 110, 111]. The Hough transform approach is usually applied to both raster and vector data for detecting 2D and 3D geometric primitives (e.g. 2D and 3D lines, circles, ellipses, 2D and 3D planes). In terms of 3D point clouds for building reconstructions, it can be employed for detecting ridges, building boundaries and 3D roof planes. Overby *et al.* [112] and Sohn *et al.* [113] utilized the 3D Hough transform to detect planes iteratively from building point clouds. Furthermore, Rau and Lin [79] used 2D Hough transform for detecting planes indirectly from the projection of 3D point clouds. This approach has been extended by Rabbani and Van den Heuvel [105] for the extraction of other geometric primitives (e.g. cylinders). Jibrini *et al.* [114] proposed an automatic 3D building reconstruction process from high-resolution aerial stereo-pairs and 2D CMs. Their approach is based on two main processes: Extracting roof planar hypotheses by 3D Hough transform on a fuzzy-based DSM and repositioning the cadastral 2D segments in 3D using the planar hypotheses obtained from the previous step and

image contrasts.

Once the planar roof point segments are available, roof boundaries can be extracted from the boundary points of the segments. Polyhedral building models can then be reconstructed by generalizing and adjusting the roof boundaries [20]. However, the resulting boundaries are usually irregular and consist of short line segments with high orientation variability that cannot be used for modeling. Regularization procedures are applied to the extracted boundaries to reduce these irregularities (i.e. to minimize and to simplify the polygons with fewer vertices). The previous research works mentioned are reviewed in Section 1.4.2.

After detecting the roof segment regions, the adjacency between them is determined for modeling. Verma *et al.* [42] considered two roof segments to be adjacent if their edges are close enough. Sampath and Shan [43] suggested calculating the minimum distance of all possible combinations of the points between two segments and define it as the distance between the segments. Milde *et al.* [115] calculated the perpendicular distance between the points of two segments and the segments' intersection lines. Elberink [116] and Xiong *et al.* [28] determined an intersection line for the adjacent segments by considering a threshold for the length of the intersection line.

Ridge lines and step edges can be extracted regarding planar segments, adjacency and topological relations. Intersection lines (ridge lines) indicate the connection between two adjacent roof segments, while step edges correspond to the positions of maximum height changes and height discontinuities between neighboring segments. The step edge between two segments can be determined by analyzing the height differences on an orthogonal profile to the boundary of the segments [11]. Elberink [116] detected step edges by analyzing 2D and 3D relations between the adjacent segments which have no intersection line. Sohn *et al.* [113] developed a step line extractor, the so-called compass line filter (CLF), which tracks all boundary points with height discontinuities on the adjacent clusters on the TIN. This process continues with thinning and computing the directions of the step edge lines. Rottensteiner and Briesse [117] computed the intersection lines as the lines with the smallest RMSE of the edge pixels between two adjacent segments. Arefi and Reinartz [12] combined the DSMs of satellite imagery with the edge information from orthorectified PAN images to obtain high quality ridge lines. They extracted ridge lines by RANSAC. The ridge lines and the step edge lines are used in the next step to determine a roof topology graph.

Roof topologies have to be determined to identify the relationships between roof planes, which can be represented as roof topology graph (RTG) [23, 42, 110, 115] or adjacency matrix [43]. Verma *et al.* [42] determined roof topologies based on the direction of the projected normal of the polygons on X-Y planes. Elberink [23]

and Milde *et al.* [115] labeled the edges of graphs based on the geometry of roof segments (e.g. same/opposite normal directions, convex/concave, tilted/horizontal). Schwable *et al.* [118] discovered topologies by projecting roof points orthogonal to the main orientation of the roof planes obtained based on the bin analysis of height histograms and ground plans. Ridge lines in the 2D projections are detected to construct the topology and generate the roof planes.

Using roof segments, boundaries, intersection lines, step edges and roof topology graphs are necessary components for constructing polyhedron models. Mass and Vosselman [103] proposed generating vertical wall planes from all outlines where their heights were determined by intersecting them to the roof planes. This method has been extended later by Vosselman [68] for the presence of step edges between buildings. Rottensteiner *et al.* [11] combined roof polygon segments by checking the consistency between pair-wise polygons. If the intersection line of two segments changes the direction of one of the segments, a step edge is inserted between the end points of common lines of the two segments. Dorninger and Pfeifer [46] proposed adding a step edge in the case of no closed roof boundary. Zhou and Neumann [119] proposed a novel method named 2.5D dual contouring, which estimates the vertices of building polygons in grid cells overlaying the segmented LiDAR points in the x-y plane. In this method, building polygon vertices are estimated by minimizing their distances to the boundary lines, which separate different LiDAR segment points in each grid cell. Sun and Salvaggio [44] proposed a similar method to [119] which can model complex roof shapes including non-planar roof segments; however, it cannot recognize the step edge within one height level of roof, which causes imperfections in the modeling of superstructures. In order to remedy this deficiency, Orthuber and Avbelj [71] added detailed segmentation to the dual contouring approach. Additionally, they employed an adaptive 2D dual contouring approach to consider step edges between segments within one roof height level. Lafarge and Mallet [27] segmented 3D point clouds by assigning labels to the points (e.g. roof, ground). They then used a MRF algorithm for label propagation on a grid under some geometric constraints. The propagation was restricted by 3D lines, which were extracted from the segments. The final model was then reconstructed by mesh patches created by triangulation of the grid cells containing the roof points. Taillandier and Deriche [120] proposed an automatic building reconstruction method based on multiple aerial images. This method considers planes as primitives and determines their arrangements using a 3D graph of the facets, where its nodes represent initial facets and the edges indicate that the adjacent nodes belong to at least a common hypothesis of a building. Maximal cliques of the graph provide all hypotheses of the buildings. The best hypothesis is then selected using a Bayesian algorithm. Taillandier [111] introduced a 3D building reconstruction method for large and dense urban areas based on CM and aerial images. They considered fixed slopes (45°) for the corresponding planes to gutter (eave) segments. This method was improved later by Durupt and Taillandier [121] by extracting planes from DSMs using RANSAC and extracting principal directions

based on CMs, which helped to extract the principal orientations. This method builds a 3D graph considering all possible models, where the best model is then selected by a Bayesian algorithm.

The methods based on RTGs or adjacency matrices model roof shapes through matching the step edges and intersection lines of RTGs with a target graph which consists of a set of very simple roof types. These methods model complex roof shapes by putting the simple roof shapes through sub-graph matching processes [42]. Milde *et al.* [115] utilized formal grammar, which is used to express valid roof types on the roof primitive shapes selected based on sub-graph matching, to reconstruct complex roofs as a combination of several primitive roof shapes. Elberink and Vosselman [23] defined a larger number of model shapes in the target graph library compared to the previous works. In this method, the incomplete segments that do not fit an existing target roof topology are corrected by human operators. Pears and Mass [8] suggested using cycle graphs to represent 3D corners better. Xiong *et al.* [122] used the simplest target graphs in a predefined library to make the complex building modeling more efficient. Furthermore, they introduced a graph edit dictionary containing pairs of erroneous sub-graphs of the roof topology graph and their corresponding corrected sub-graphs to correct the errors of the RTG automatically.

Roof boundaries, intersection lines and step edges derived from point clouds are usually noisy, causing ambiguities at corner points and forming closed polygons. Therefore, these features are rectified by enforcing orthogonality and parallelism to dominant orientations [46] or by using 2D ground plans [87, 85, 116]. In addition, if any three planes intersect at more than one point, this method regularizes the 3D model by merging them into a single point considering that they are closer than a certain threshold. [46]. Rottensteiner [123] used geometric regularization as a soft constraint in the intersection of vertices and walls for estimating building parameters. Verma [42] enforced certain regularities, such as symmetries of the faces which share a ridge segment parallel to the outline in parametric models, to improve the appearance and accuracy of building models in the presence of noisy data.

1.4.3.4.3 Hybrid Methods

Hybrid approaches follow “hypothesize and verify,” thereby combining data-driven and model-driven approaches [55, 15, 81]. First of all, hypotheses are formulated from primitives generated by data-driven approaches, and then these hypotheses are verified by model-driven approaches. In this way, these approaches combine the flexibility of the data-driven approaches and the robustness of the model-driven ones [15]. Sirmacek *et al.* [32] reconstructed building models based on DSMs of satellite imagery. They reconstructed complex roofs by detecting building junctions and endpoints from building skeletons. They divided the buildings into several pieces and applied an active box fitting algorithm to each piece to extract building outlines. Next, they detected ridge lines based on derivatives of the DSM data in different directions and then filtered the results. In the next step, they identified the tower and other superstructures on the rooftops by employing a height threshold. Finally, they classified the roof types based on building ridge lines and outlines and they reconstructed the 3D building model together with the tower. Lin *et al.* [109] proposed a new hierarchical method to decompose and reconstruct low-rise buildings from 3D point clouds of ground-based LiDAR data. Their method segments building point clouds into the walls, roofs and columns. It then decomposes 3D building point clouds into the basic blocks by introducing planarity, symmetry and convexity constraints and grouping the connected primitives. Arefi and Reinartz [12] proposed a novel side projection method applied to the DSM of WorldView-2 images for reconstructing building models. This method projects the 3D points on 2D planes in the direction of ridge lines. In order to extract ridge lines, DSMs of satellite imagery are combined with the edge information from VHR orthorectified PAN images and RANSAC line fitting is then employed. Subsequently, the buildings are modeled as hipped, gable or flat roofs. Wang *et al.* [110] proposed a method based on the semantic decomposition of LiDAR point clouds and MS aerial imagery. They used the graph adjacency between planar patches and their normal direction to create attributed graphs. The latter are then decomposed into sub-graphs where the antisymmetric planar patches exist. The sub-graphs are then used to recognize the roof type primitives. After that, 2D corners are extracted from aerial images corresponding to the primitives selected and the final models are reconstructed using nonlinear LS under some constraints obtained from LiDAR data and aerial images. Zheng *et al.* [30] proposed a method based on LiDAR DSMs, which selects the roof types using the aspect value of the pixels within the sub-footprints obtained from the building decomposition step. An appropriate model is selected from the library and placed on the sub-footprints to reconstruct 3D building models using regular sub-footprints. For a reconstruction using irregular sub-footprints, they are decomposed by the ridge lines and then extruded by the height values from LiDAR DSMs. This method can produce 3D models of complex buildings at LOD2 for an entire city.

A comparison of existing methods for 3D building model reconstruction are summarized in Tables B.1, B.2 and B.3.

1.4.4 Summary

In summary, most of the methods mentioned previously (especially the data-driven ones) use LiDAR data or the DSM of aerial imagery, which provide denser and more accurate information in comparison to the DSM of satellite imagery. Nevertheless, we use the DSM of satellite imagery in this dissertation, which allows us to reconstruct 3D building models in more different and larger areas compared to the LiDAR or aerial images. In order to deal with the lower resolution and SNR of the DSM of satellite imagery, we use it together with high spatial resolution PAN images in building outline extraction and PS images (a fusion of PAN and MS images) in roof type classification. Our building model reconstruction approach is a hybrid method that brings the flexibility of the data-driven and the robustness of the model-driven methods together. Furthermore, it allows the use of DSM data directly without any refinement process. Our approach is similar to the ones proposed in [12, 26, 32] who used the DSM of satellite imagery for building model reconstructions. However, it differs from Lafarge *et al.* [26] regarding the building boundaries extraction, size of the libraries used, roof type classification and 3D building reconstruction methods. They extracted building boundaries using only the DSM of satellite imagery without any high spatial image information. Furthermore, they used a larger library than the one we propose and employed only DSM data for roof type classification, which could increase the reconstruction error. They used RJMCMC algorithm jointly with a simulated annealing optimization algorithm to select roof types and fitting roof models to the 3D point clouds to avoid falling into a local minimum in parameters continuous search space, which is comparable to our modified exhaustive search in finding the global maximum in predefined parameters of discrete search space. In addition, our work is different from Arefi and Reinartz [12] regarding the boundary extraction and model reconstruction steps. They employed the CMBR technique for boundary extraction, which could fail in the buildings with multiple orientations. Moreover, they utilized a method based on ridge line extraction in the reconstruction step, which could fail in the roof areas with a low image contrast and poor DSM data. Simacek *et al.* [32] improved the outline extraction of [12] to handle boundary extraction of the buildings with multiple orientations. However, they used similar features (e.g. ridge lines) as [12] for the reconstruction step, which has the same drawbacks as the approach by [12]. The work of Dudan and Lafarge [124] is different from our method regarding the level of details of the reconstructed models and the model representation. They only reconstructed the mesh-based models of the high-rise buildings with flat roof type (LOD1).

1.5 Thesis Overview

The objective of this research is to present fully automatic building extraction and model reconstruction by integrating model-driven and data-driven strategies while taking advantage of using high spatial resolution satellite images and DSMs. Firstly, the DSM-based building mask is refined by classifying the geometrical features of the VHR PAN satellite images. Chapter 2 describes our proposed methods for refining the DSM-based mask and presents a new heuristic method for extracting parameterized building outlines. Chapter 3 propose a new side-based method for the decomposition of the building polygon into the basic rectangular shapes. In Chapter 3 we further propose a new method for the selection of the relevant rectangle among abundance rectangles obtained from the previous step. Chapter 4 introduces a new method for the selection of the roof type based on the patch-wise deep learning-based classification using PS images. Chapter 5 proposes a model-driven method to reconstruct a 3D building model from the DSM of satellite imagery. In this chapter, we develop a new method to detect the height discontinuities and decompose the rectangles obtained from Chapter 3 based on these height discontinuities and also roof types. Chapter 5 makes further final decisions about the roof types by fusing the classification results obtained from Chapter 4 and the DSM in the Bayesian formula. This chapter also presents a robust method to detect the ridge orientation based on the local maximum of the DSM within the rectangles and the neighborhood relation of the rectangles. After all the initialization mentioned above, a roof model is selected and fitted to the DSM within a rectangle using Huber distance cost function by an exhaustive search approach in a tight range for each roof model parameter. The reconstructed rectangular building models are finally intersected and merged to also reconstruct complex buildings and indicate them in a CSG representation. Chapter 6 summarizes and concludes this dissertation and discusses directions for future research.

Building Boundary Extraction

Building boundary extraction is a key step for automatic 3D building model reconstruction if the building footprint is not available. As we mentioned in Chapter 1, the DSM data suffers from imperfections and causes difficulties in detecting and extracting buildings. In this chapter, an improved framework is introduced, in comparison to the existing approaches, focusing on the refinement of DSM-based building masks and simplification of the building outlines extracted. We use the advantages of DSM data while remedying its imperfections by integrating DSM data with high spatial resolution PAN images to improve the building masks, especially at the building boundaries. The framework introduced concentrates on the DSM derived from satellite data, where the DSM quality is low. Using the refined masks, we further develop a novel automatic method for an optimized extraction and simplification of building outlines. The main steps of our proposed method are shown in Figure 2.1.

In this chapter, Section 2.1 describes the mask refinement step (the blue blocks in Figure 2.1). Regarding the mask refinement, building boundaries are enhanced by applying a classification method (Section 2.1.3) to the primitive geometrical features of their corresponding high-resolution PAN images (Section 2.1.2). Section 2.2 then explains the newly developed data-driven procedure for parameterized building outline extraction and simplification (the green block in Figure 2.1). In order to extract a building's outline, the building boundary points are traced on its corresponding refined mask and a set of line segments is fitted to them (Section 2.2.1). The line segments obtained are then regularized by finding the building's main orientations (Section 2.2.2) and assigning all the line segments to their appropriate main orientations based on arc length differences between main orientations and line segments. Line segments are then aligned to their assigned main orientations based on LS adjustment (Section 2.2.3). As the final step, the sequential line segments are connected based on a set of rules (Section 2.2.4). The proposed methodology is applied to the different datasets and the results will be discussed in Section 2.3 for validation.

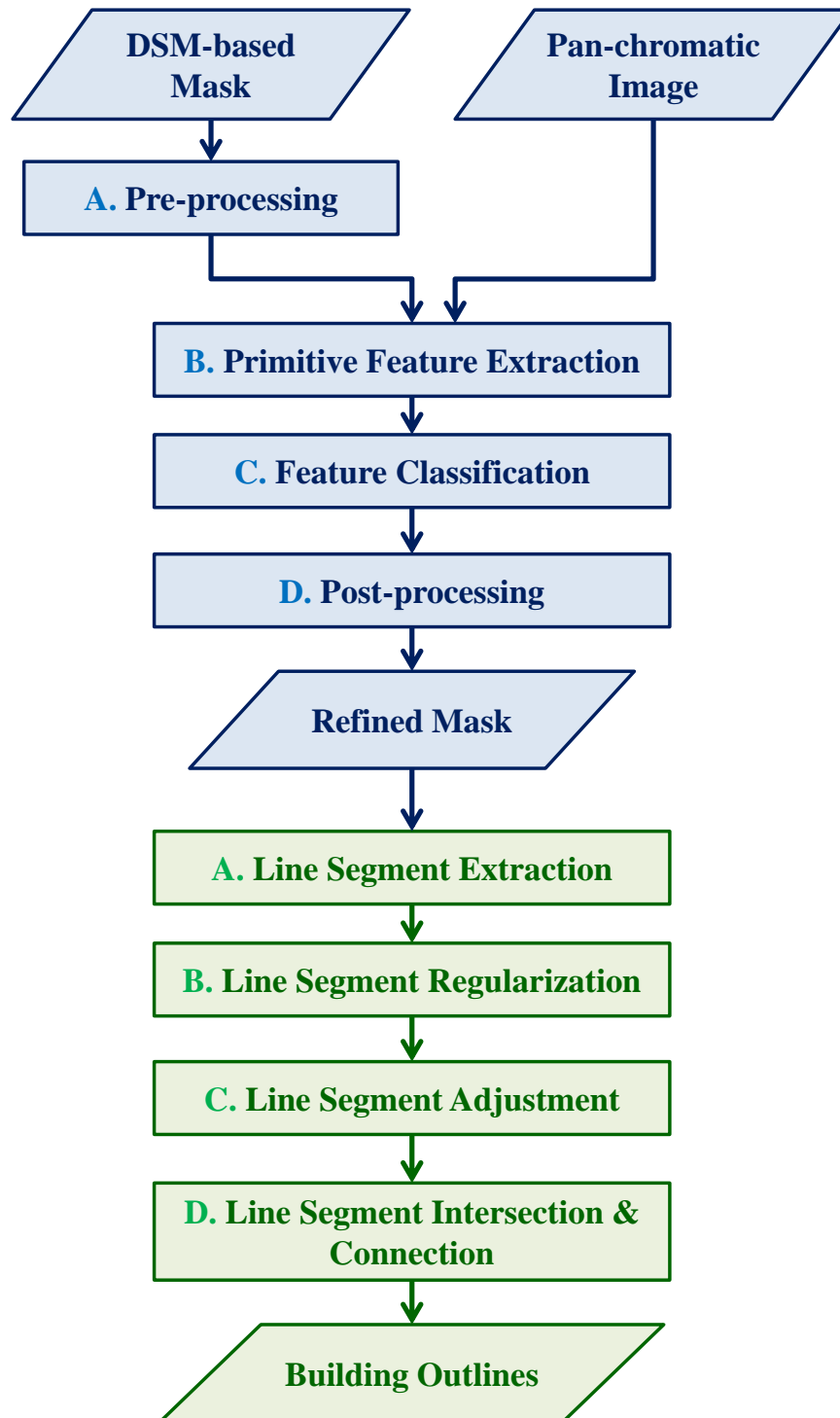


Figure 2.1: Overview of our proposed building outlines extraction method.

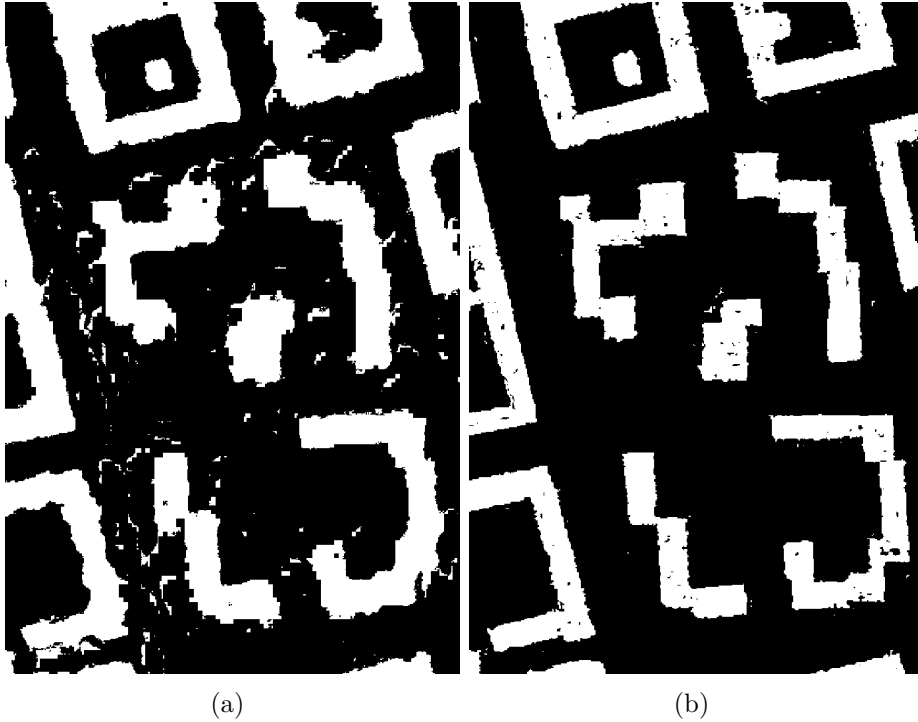


Figure 2.2: (a) DSM-based building mask. (b) Building mask after refinement.

2.1 Building Mask Refinement

In this dissertation, the building masks are generated based on the DSM of satellite imagery using the method explained in [19]. As illustrated in Figure 2.2. (a), this DSM-based building mask presents rough building boundaries. Some patterns and objects, such as shadows and trees, are incorrectly considered as building elements due to their similar characteristics or heights. Therefore, a preprocessing step is needed to eliminate these small noisy objects. Firstly, considering the minimum possible building size in the images provided, the groups of building points from the DSM-based masks which are smaller than a threshold are removed. The threshold in our experiments is set to 300 pixels, because at the 0.5 m resolution of our PAN images, we assume only buildings that cover areas larger than 75 m^2 . A preprocessing step based on morphological filters is then performed to make the building mask large enough to cover the probable missing building edges and some surrounding areas, which will be used later as negative samples for the classification phase.

2.1.1 Morphological Filtering of the DSM-based Mask

Mathematical morphology consists of a set of image processing operations that are used to extract the image components useful in the representation and description of the shape of the region, such as boundaries, skeletons and CHs. Pesaresi and Benediktsson [125] used morphological filters to process remotely sensed data. The basic mathematical operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image (Equation 2.1), while erosion removes pixels on object boundaries (Equation 2.2). The number of pixels added or removed from objects in an image depends on the size and shape of the structuring element. The structuring element should be selected based on the knowledge about the size, shape and orientation of objects which are to be filtered in an image.

The structuring element can be selected as a disk, a square, or a cross. The dilation of BI by the structuring element SE is defined by:

$$BI \oplus SE = \bigcup_{se \in SE} BI_{se}, \quad (2.1)$$

where the erosion of the binary image BI by the structuring element SE is defined by:

$$BI \ominus SE = \bigcap_{se \in SE} BI_{-se} \quad (2.2)$$

Opening and Closing are the most important morphological operators derived from erosion and dilation. Opening preserves the foreground pixels of regions that have a similar shape to the structuring element or that can contain the structuring element completely, while removing all other regions of foreground pixels. In contrast to Opening, Closing fills all the background pixels of a region that does not contain the structuring element.

The opening of BI by SE is obtained by the erosion of BI by SE , followed by dilation of the resulting image by SE ,

$$BI \circ SE = (BI \ominus SE) \oplus SE \quad (2.3)$$

The closing of BI by SE is obtained by the dilation of BI by SE , followed by erosion of the resulting image by SE ,

$$BI \bullet SE = (BI \oplus SE) \ominus SE \quad (2.4)$$

Since building masks should be refined on the boundaries, DSM-based and footprint-based masks are compared on the building outline. Pixels at the building outline are then labeled as positive, whereas non-building samples are labeled as negative (Figure 2.3). Consequently, we apply a morphological Opening followed by a Closing to the mask with the structuring element for the Opening larger than that of the Closing. This enlarges the mask enough to take both positive and negative samples. We use a disc with a diameter of three pixels for Opening and a disc with a diameter of two pixels for Closing.

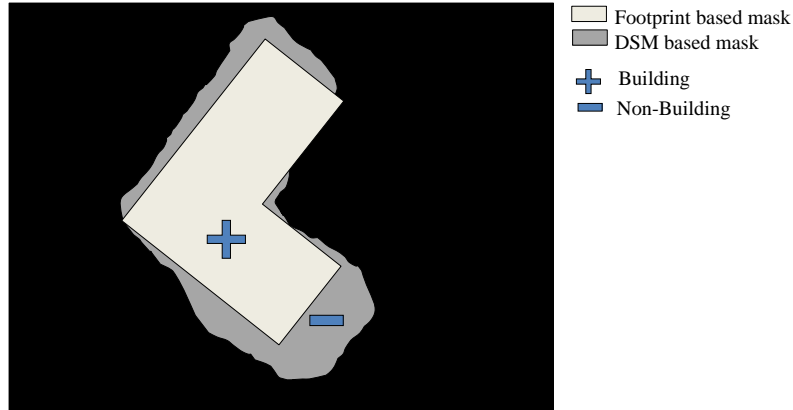


Figure 2.3: Illustration of building and non-building pixels as positive and negative pixel labels, respectively

2.1.2 Primitive Feature Extraction

Each pixel within the masked area of PAN images is described by a vector of its important features using the scale invariant features transform (SIFT) method to classify image regions into building and non-building [126, 127]. The reason for using a SIFT feature descriptor rather than other feature descriptors, such as pixel intensity (intensity histogram (InHist)) and texture (Weber local descriptor (WLD)) [128], is its capability to extract linear geometrical structures, such as line and corner, and its robustness against the noise which exists in PAN satellite image [129]. Another reason is that they are invariant to image scale, rotation and illumination changes. In the experiment, SIFT feature descriptors are extracted in a dense manner for each pixel of building and non-building. The SIFT generates 16 orientation histograms on the 4×4 neighborhood of the pixels of interest. Each histogram consists of eight bins, corresponding to eight different directions. These histograms are concatenated forming a feature vector with 128 elements.

2.1.3 Feature Classification

In this step, extracted features are classified to improve the building mask on the building outline. Using SIFT feature descriptors and then classification of them for the purpose of building mask refinement are developed in this research that have not been represented in previous researches. To this end, 30 % of the building and non-building candidate pixels in each image are randomly sampled as positive and negative samples, respectively, to discriminate the area covered by buildings from the rest of the image. We then train the SVM on the sample points. The SVM classifier has been used in many applications, such as remotely sensed data classification, during the past decade. It is very attractive to provide high accuracy in the

presence of heterogeneous classes for which only a few training samples are available. It also has a good performance using a high dimensional features space. Please refer to appendix C for more mathematical details. In the experiments, the model parameters of SVM classifier are selected empirically by varying the parameters and validating the resulting model on a small set of the sample points. The parameters leading to the best model are then considered for further experiments.

2.1.4 Mask Postprocessing

The SIFT descriptors are highly sensitive to building edges compared to the other building features (e.g. patterns on rooftops). Since SIFT is computed for a window of size 16x16 pixels, the pixels within an eight-pixel distance (i.e. half the SIFT window size) of building edges still contain edge information. Thus, these pixels are usually classified into the positive class forming building silhouettes with precisely determined contours. However, the pixels within building areas which are too far from building edges to contain building edge information are usually classified into the negative class. This results in gaps within building boundaries. Figure 2.4. (b), shows a sample classification result in which the positive class is depicted in bright green and the negative class in dark green and red, where dark green refers to the gaps within building boundaries. Therefore, after discriminating the regions covered by buildings, we perform a morphological Closing (with a disc of three pixels diameter) followed by a gap filling method (with a threshold of 400 pixels) to fill in the small imperfections and gaps within the building boundaries, the dark green areas in Figure 2.4. (b). The resulting refined masks will be used further in the building outline extraction process. Figure 2.2. (b) and Figure 2.4. (b) demonstrate building masks after refinement. Although there are still small gaps in the building areas, since they do not usually lie on building edges, they will not influence the results of the building outline extraction.

2.2 Building Outline Extraction and Simplification

In this section, we model the refined building masks as polygons to extract parameterized building outlines. The refined masks approximate the true position of the building boundaries, as our experimental results have shown (Section 2.1). Therefore, tracing the boundary points of a building according to the refined mask allows us to detect the building's outline. Consequently, we use the modified Moore-Neighbor tracing algorithm [130] which detects boundary points in a consecutive manner. Moore-Neighbor tracing is an efficient algorithm to extract the contour of any pattern no matter what its connectivity is. It starts scanning from the top-left position and scans each pixel from left to right downwards. When a pixel of

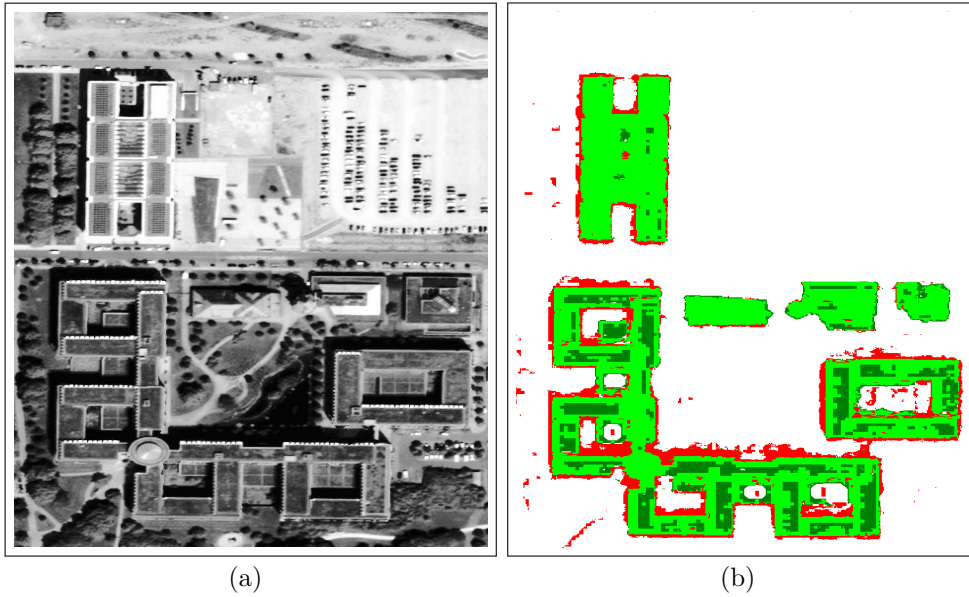


Figure 2.4: (a) Original PAN image and (b) its refined mask, bright and dark green areas, together with the areas removed from the mask, red areas.

contour is found, the contour is traced by checking the eight neighbors, named Moore-Neighbors, and the possibility of having another pixel of contour. The neighbors are checked in a clockwise manner. Every time a pixel of the contour is found, clockwise sequence tracing is started from the pixel that has been entered to the currently detected pixel of the contour. This algorithm is repeated till the first pixel of the contour is visited again. This stopping criterion is called the Jacob-Eliosof criterion. After extracting the boundary points, a consecutive line-fitting procedure based on the combination of RANSAC and LS adjustment is applied to each group of the boundary points to extract line segments.

2.2.1 Line Segment Extraction

A complete overview of the process chain followed for this step is depicted in Figure 2.5, which starts with an initial group of three consecutive boundary points. The RMSE distance of a next consecutive point is then measured from the last point of the initial group. If the RMSE is less than a predefined threshold, the point will be grouped with the previous point. Otherwise, it will be used for initiating a new group and a line is fitted locally to all the points within the previous group. The line is centered at the mean of the points and is extended in the directions of the points' principal eigenvector, with the scale determined by the corresponding eigenvalue to the principal eigenvector. This process iterates until all the boundary points are represented by line segments. The line segments shorter than five pixels

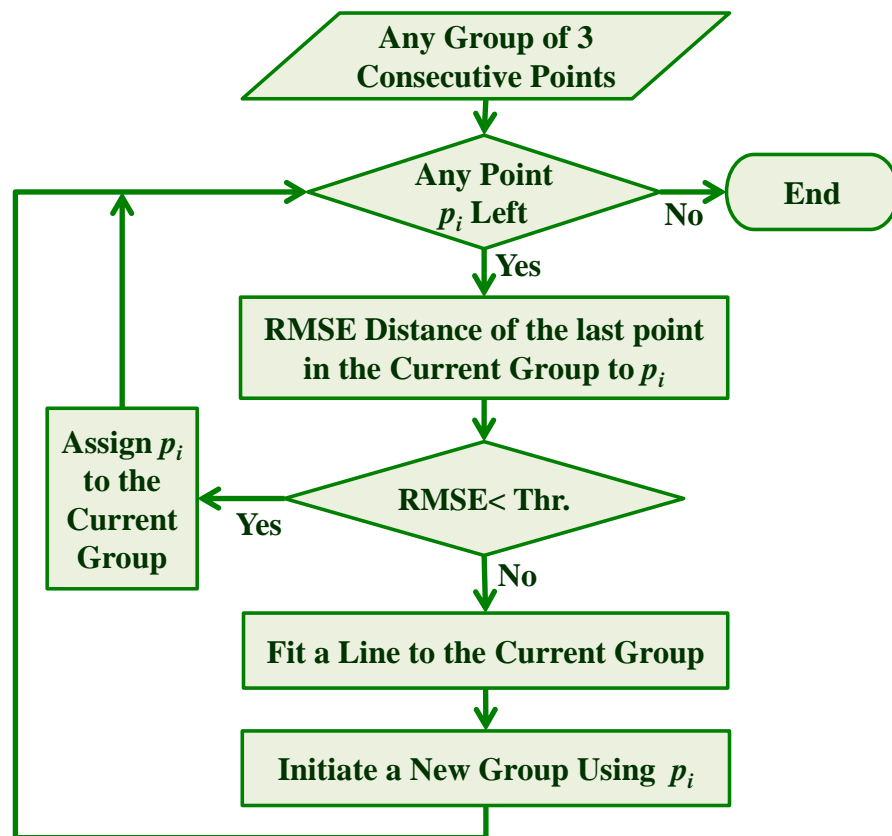


Figure 2.5: Line segment extraction process.

are then removed, because at the 0.5 m resolution of our PAN images, a building length cannot be smaller than five pixels. This method was initially developed by Vosselman [68]. In our research, we improved this method and make it flexible for any direction. The direction of the new group of points, for instance, does not have to be perpendicular to the previous group and it can accept any direction. The line segments need to be regularized due to the irregular shape of the structures formed by the remaining line segments caused by the existing noise and artifacts. Figure 2.6. (a) illustrates the extracted line segments for an example building.

2.2.2 Line Segment Regularization

We determine the main orientation of each building to refine the line segments extracted. A conventional assumption is that buildings have mutually perpendicular orientations. However, complex buildings usually have more than one main orientation, which might be non-perpendicular. In this dissertation, we develop a new method that considers multiple main orientations for each building, regardless of the perpendicularity of the orientations. We determine the building’s main orientations based on the length of the line segments and their arc lengths. We select the orientation of each building’s longest line segment as the building’s main orientation. The other line segments which are either parallel or perpendicular to this main orientation will be grouped together. We measure the arc lengths of the longest line segment and the other line segments, based on a predefined threshold (five pixels in our experiments), to check the parallelism and perpendicularity. With this threshold, the angle distance would be smaller than 20° , which is a good assumption for parallelism. The longest of the remaining line segments in the next iteration is selected to determine another main orientation of the building. This process iterates until all line segments are grouped with a main orientation of the building. Figure 2.6. (b) shows three main orientations of the building depicted by magenta, green and blue colors. The assignment of the other line segments to these three main orientations is demonstrated in Figure 2.6. (c).

In addition to the orientations of the line segments within each orientation class, their locality is also important. In other words, the line segments within each class should also lie in the same neighborhood. However, as illustrated in Figure 2.6. (c), there are a number of line segments misclassified due to the presence of noise and artifacts. The line segments l_1 , l_2 , l_3 and l_4 in Figure 2.6. (c), for example, should be assigned to the magenta orientation class, whereas l_5 should be assigned to the blue orientation class. Dealing with this problem, we verify the main orientation assigned to the line segments in each neighborhood and assign the misclassified line segments to a correct orientation class. Figure 2.7 represents the main steps of this procedure.

We start by splitting each building into several segments (CHs) by extracting the points on the skeletons of the building masks. The points are then placed at

2. Building Boundary Extraction

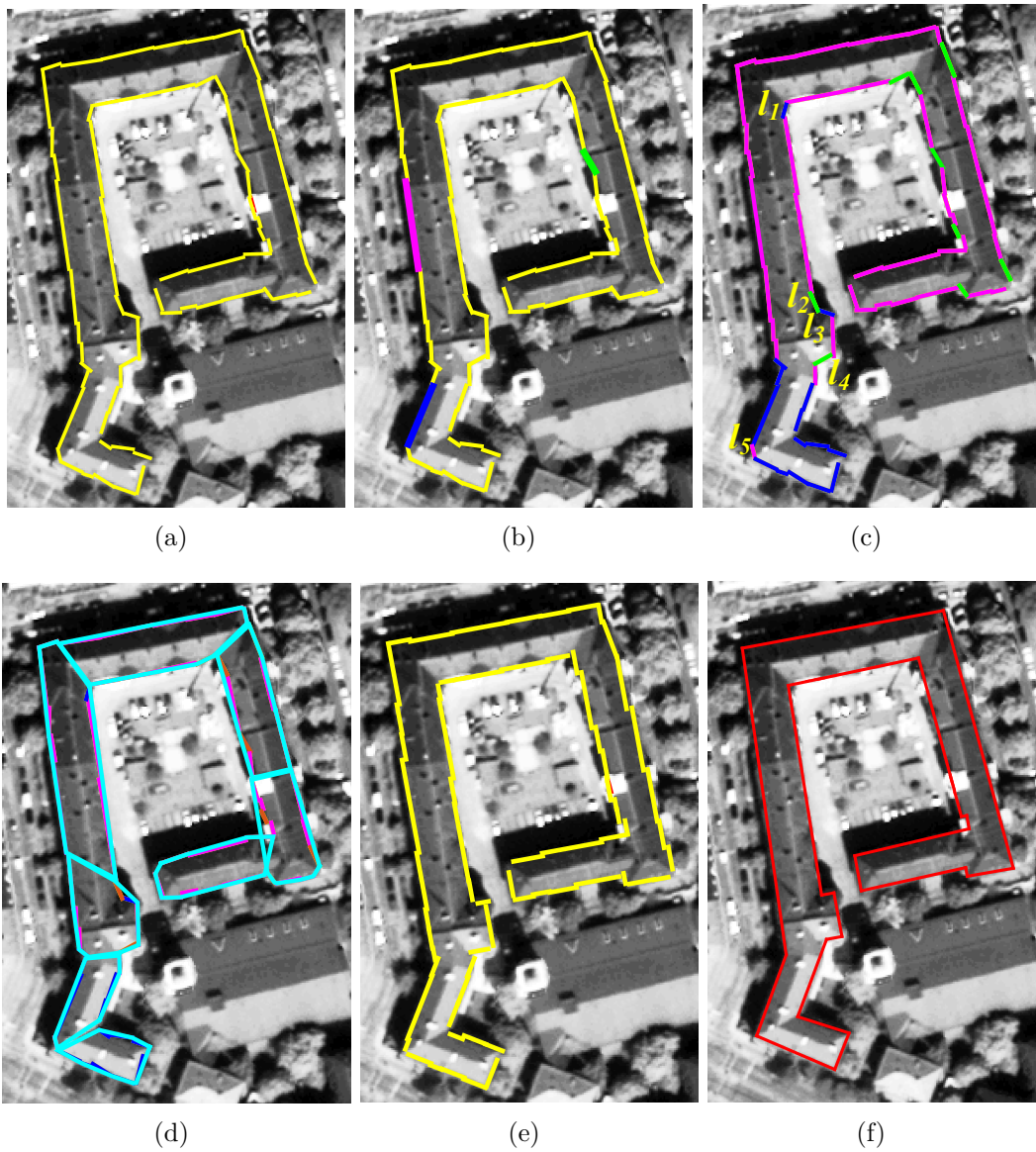


Figure 2.6: (a) Extracted line segments for an example building. (b) Three main orientations of the building depicted by magenta, green and blue colors. (c) Assignment of the line segments to the building's main orientations. (d) Splitting the building into several CHs. (e) Extracted line segments after regularization and adjustment. (f) Outline of the building after intersecting and connecting the line segments.

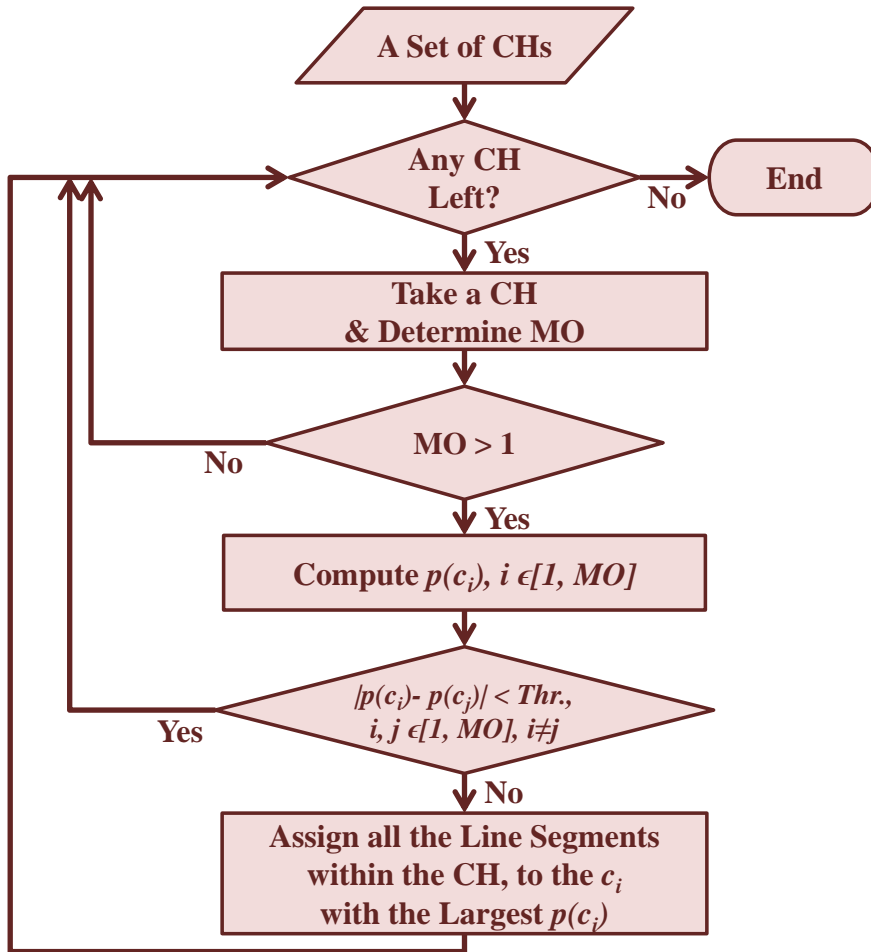


Figure 2.7: Process chain of the line segment regularization, where CH denotes a convex hull and MO indicates the number of main orientation classes c_i in each CH.

a five-pixel distance from each other and the ten nearest neighboring line segments are selected for each point and a CH is fitted (please refer to Figure 2.6. (d)), where the number of nearest neighbors are determined empirically. The orientations of the line segments in each CH are compared and the probability of every existing main orientation class c_i is computed as:

$$p(c_i) = \frac{\sum_j l_{ij}}{\sum_i \sum_j l_{ij}}, \quad (2.5)$$

where l_{ij} indicates the length of the line segment j which belongs to the orientation class c_i . If the probabilities of the existing main orientation classes in a CH are close enough, no change will be imposed. Otherwise, all the line segments will be grouped into the main orientation class with the largest probability.

2.2.3 Line Segment Adjustment

A further goal is to adjust the line segments within each orientation class to align them as either parallel or perpendicular to the main orientation of the class, based on their arc lengths to the main orientation. Therefore, the line segment adjustment problem can be seen as fitting a rectilinear to the points of the line segments in the sense of LS method. This is a minimization problem which finds the best parameters of the line segment by minimizing the distance between the observations (building boundary pixels) and rectilinear building polygon. In addition, it can align the line segments based on their class orientation at the same time. The relation between observations and parameters are defined in the functional model. More than one functional model is needed in the Cartesian coordinate system to minimize the distance between the building boundary pixels and rectilinear building polygon.

2.2.3.1 Mathematical Model for Line Segment Adjustment

There are generally two functional models, called the Gauss–Markov (GM) and Gauss–Helmert (GH) models, which are used utilizing the LS method to estimate the unknown parameters to solve this adjustment problem of rectilinear polygon fitting [64]. In this dissertation, the GH is used as the functional model for line segment adjustment.

Least Squares (LS) Method The LS theory is a well-known general framework used to determine unknown parameters based on observations provided. This general concept allows the integration of different constraints to solve the complex problem. This optimization technique is popular in mathematics and geodesy. The functional or stochastic models should be defined for describing the problem in LS adjustment. In the functional model, the observations \mathbf{L} are described as the function of unknown parameters \mathbf{X} .

$$\mathbf{L} = F(\mathbf{X}), \quad (2.6)$$

This function is in a linear or nonlinear form that can be collected in a vector $\mathbf{F} = [F_1(\cdot), F_2(\cdot), \dots]^T$. If these functions are nonlinear, they should be linearized regarding the approximate value \mathbf{X}^0 of unknown parameters. These functions are linearized by taking the first order of the Taylor series expansion around approximate values of unknown parameters. The Jacobian matrix $\mathbf{A}(\mathbf{J} = \partial\mathbf{F}/\partial\mathbf{X})$ is determined based on the first order partial derivative of observation equations, and evaluated at \mathbf{X}^0 . Observations in the stochastic model have accuracies which are introduced into the system by the weight matrix \mathbf{W} . The matrix \mathbf{W} has a diagonal structure if there are no correlations among the observations. If the number of observations is more than the number of the unknown parameters, in the presence of observation errors, it is not possible to have unique results. The aim of adjustment theory is to find an optimal solution that satisfies the constraints through minimizing the corrections (residuals \mathbf{v}) to the observations [64, 131].

$$\mathbf{L} + \mathbf{v} = \mathbf{A}\mathbf{X}, \quad (2.7)$$

LS method minimizes the quadratic form of $\mathbf{v}^T\mathbf{W}\mathbf{v}$ to find the $\hat{\mathbf{x}}$ and then finally estimate unknown parameters $\hat{\mathbf{X}}$.

$$\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}(\mathbf{L} - F(\mathbf{X}^0)) \quad (2.8)$$

$$\hat{\mathbf{X}} = \mathbf{X}^0 + \hat{\mathbf{x}}, \quad (2.9)$$

In the case of linearization, the LS estimation is iterated and every iteration \mathbf{X}^0 is updated until the corrections to the unknown $\hat{\mathbf{x}}$ are minimal. Initial values in the LS adjustment are selected in different ways depending on the problem. In the case of the line segment extraction problem, *singular-value decomposition (SVD)* is used for estimating initial values of line parameters to speed up adjustment.

Singular-value Decomposition is the factorization of \mathbf{A} into the product of three matrices $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where the columns of $\mathbf{U}(m \times m)$ and $\mathbf{V}(n \times n)$ are orthonormal and the matrix $\mathbf{\Sigma} = \text{diag}(\lambda_1, \dots, \lambda_n)(m \times n)$ is diagonal with positive real entries $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$. Based on this property, the problem of $\|\mathbf{A}\|_2 = \min$, subject to $\|\mathbf{x}\|_2 = 1$ has the solution $\mathbf{X} = \mathbf{v}_n$ and the value of minimum is $\min_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 = \lambda_n$. Line parameters can be found very easily by using this theorem.

As mentioned, the problem of fitting lines is minimizing the sum of the squares of the distances to specified points. As Figure 2.8 shows, the straight line can be represented by equations

$$b + \mathbf{n}_i \mathbf{l} = 0 \quad (2.10)$$

$$n_x^2 + n_y^2 = 1 \quad (2.11)$$

where \mathbf{n}_i ($i \in 1, \dots, 6$) is the unit normal vector of parallel or perpendicular lines of each building block which constraint to Equation (2.11). The sign and components of each normal vector are defined based on each direction. For example, in Figure 2.8, $\mathbf{n}_3 = -\mathbf{n}_1$, $\mathbf{n}_6 = -\mathbf{n}_2$, $\mathbf{n}_3 \& \mathbf{n}_5 = [n_x, n_y]^T$, and $\mathbf{n}_2 \& \mathbf{n}_4 = [n_y, -n_x]^T$ in which n_x and n_y are the normal vector components of aforementioned unit normal vector \mathbf{n}_i of line i . The \mathbf{l} is observations vector of line points coordinates (x, y) .

Equation 2.10 is true, if the coordinates (x, y) lie on the line. Otherwise, $P = (x_p, y_p)$ correspond to the coordinates of the points located at the distances $\|r\|$ from the line \mathbf{l}_p .

$$r = b + \mathbf{n}_i \mathbf{l}_p \quad (2.12)$$

Therefore, the sum of squares of the distances from the line should be minimized to obtain the line parameters, which is solved in a constrained LS problem.

$$\begin{pmatrix} 1 & x_{p1} & y_{p1} \\ 1 & x_{p2} & y_{p2} \\ \vdots & \vdots & \vdots \\ 1 & x_{pm} & y_{pm} \end{pmatrix} \begin{pmatrix} b \\ n_x \\ n_y \end{pmatrix} \approx \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (2.13)$$

Let \mathbf{A} be the Jacobian matrix of linear equation of 2.13. Using the **QR** decomposition [132], where \mathbf{Q} is an orthogonal matrix and $\mathbf{A} = \mathbf{QR}$, the linear equation can be reduced to an upper triangular matrix \mathbf{R} as $\mathbf{RX} \approx \mathbf{0}$ [133].

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix} \begin{pmatrix} b \\ n_x \\ n_y \end{pmatrix} \approx \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.14)$$

The normal vectors are normalized with the constraint subject to Equation 2.11. Finally, the constrained LS problem

$$\begin{pmatrix} r_{22} & r_{23} \\ 0 & r_{33} \end{pmatrix} \begin{pmatrix} n_x \\ n_y \end{pmatrix} \approx \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \|\mathbf{n}_i\|_2 = 1 \quad (2.15)$$

is solved based on *SVD* and the theorem mentioned above. Therefore ev_n will be eigenvector in the last column of \mathbf{V} corresponding to minimum singular value (λ_n). After computing ev_n , \mathbf{b} will be computed according to first row of Equation 2.14 [134]. After initialization of line parameters using *SVD*, the GH model can improve parameters iteratively through updating observation and unknown parameters.

The GH is known as a combined or conditional model with parameters. It considers unknown parameters and observations as random variables, which are collected in the set of condition equations in the form of $\mathbf{F}(\hat{\mathbf{L}}, \hat{\mathbf{X}}) = 0$. It means that it can estimate unknown parameters and observations jointly [131]. The functional model for the line fitting based on Equation 2.12 can be expressed as the following target functions:

$$F_1 = a_1x + a_2y + b, \quad (2.16)$$

$$F_2 = a_1^2 + a_2^2 - 1, \quad (2.17)$$

where F_1 is a line equation with the coefficients a_1 , a_2 , and b . Since a_1 and a_2 are the normal vector components of each line segment, F_2 is a constraint equation to normalize the normal vectors of the line segments. F_2 is a nonlinear function; therefore, we linearize it using Taylor's series expansion so that it be used in the LS adjustment procedure. The GH's functional model can, thus, be written as:

$$\mathbf{F}(\hat{\mathbf{L}}, \hat{\mathbf{X}}) = \mathbf{w} + \mathbf{B}\mathbf{v} + \mathbf{A}\hat{\mathbf{x}}, \quad (2.18)$$

where \mathbf{A} and \mathbf{B} are matrices containing the derivatives of the target functions (F_1 and F_2) regarding the unknown parameters (a_1 , a_2 , and b) and the observed parameters (x and y), respectively. Furthermore, \mathbf{w} is the vector of misclosure.

Figure 2.8 illustrates an example of fitting a model to a set of points using LS adjustment. In this figure, n_1 , n_2 , n_3 , n_4 , n_5 , and n_6 indicate the normal vectors of the parallel and perpendicular lines, respectively, while b_j are the line segment intercepts. Assuming only one main orientation, the matrices \mathbf{A} and \mathbf{B} , and the vector \mathbf{w} are as follows:

$$\mathbf{A} = \frac{\partial \mathbf{F}(\mathbf{L}, \mathbf{X}^0)}{\partial \mathbf{X}^0} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & x_{K_1(l_1)} & y_{K_1(l_1)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & 0 & 0 & 0 & x_{K_{N_1}(l_1)} & y_{K_{N_1}(l_1)} \\ 0 & 1 & 0 & 0 & 0 & 0 & y_{K_1(l_2)} & -x_{K_1(l_2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & y_{K_{N_2}(l_2)} & -x_{K_{N_2}(l_2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 & y_{K_1(l_6)} & -x_{K_1(l_6)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 & y_{K_{N_6}(l_6)} & -x_{K_{N_6}(l_6)} \\ 0 & 0 & 0 & 0 & 0 & 0 & (2a_1)^0 & (2a_2)^0 \end{pmatrix}, \quad (2.19)$$

$$\mathbf{B} = \frac{\partial \mathbf{F}(\mathbf{L}, \mathbf{X}^0)}{\partial \mathbf{L}} = \begin{pmatrix} a_1^0 & a_2^0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & a_1^0 & a_2^0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}, \quad (2.20)$$

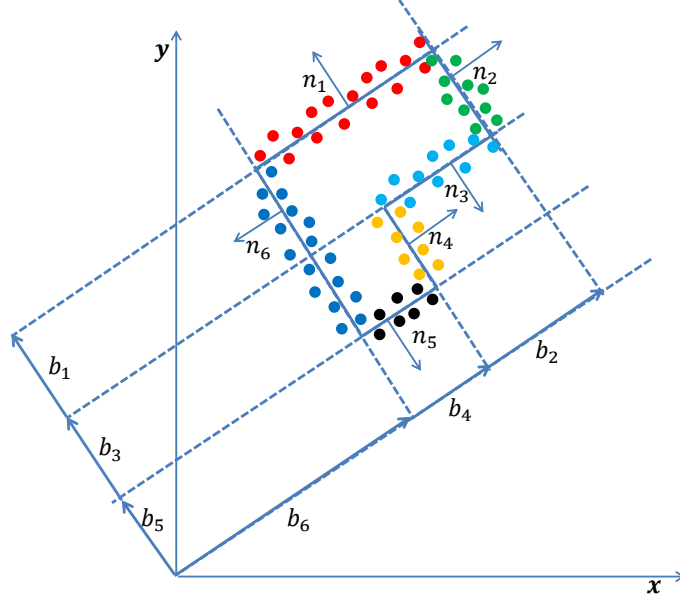


Figure 2.8: Fitting a model to a set of points using LS adjustment.

$$\mathbf{w} = \mathbf{F}(\mathbf{L}, \mathbf{X}^0) = \begin{pmatrix} a_1^0 * x_{K_1(l_1)} + a_2^0 * y_{K_1(l_1)} + b_{l_1}^0 \\ \vdots \\ -a_2^0 * x_{K_T(l_6)} + a_1^0 * y_{K_T(l_6)} + b_{l_6}^0 \\ (a_1^0)^2 + (a_2^0)^2 - 1 \end{pmatrix}, \quad (2.21)$$

where \mathbf{X}^0 indicates the initial unknown parameters, which are considered as a_1^0 , a_2^0 and b^0 , and computed based on *SVD*. Furthermore, \mathbf{L} is the observation vector which represents the point coordinates of the line segments. Moreover, $x_{K_g(l_j)}$ and $y_{K_g(l_j)}$ are the x and y coordinates of the g -th point of the line segment l_j (i.e., $K_g(l_j)$, $g \in [1, N_j]$).

In Equation (2.18), \mathbf{v} indicates the residuals of the observations and $\hat{\mathbf{x}}$ is the difference between the estimated values of the unknown vector and its initial values. These two vectors are computed as:

$$\mathbf{v} = \hat{\mathbf{L}} - \mathbf{L}, \quad (2.22)$$

$$\hat{\mathbf{x}} = \hat{\mathbf{X}} - \mathbf{X}^0. \quad (2.23)$$

In order to solve the LS adjustment problem, we express the unknown parameters

by:

$$\begin{pmatrix} \mathbf{k} \\ \hat{\mathbf{x}} \end{pmatrix} = - \begin{pmatrix} \mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T & \mathbf{A} \\ \mathbf{A}^T & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \mathbf{w} \\ 0 \end{pmatrix} \quad (2.24)$$

and the residuals are calculated by:

$$\mathbf{v} = \mathbf{W}^{-1}\mathbf{B}^T\mathbf{k}, \quad (2.25)$$

where \mathbf{k} is correlation vector and \mathbf{W} is a matrix determining the weight of the line segment. Since we found out empirically that using various weightings does not significantly influence the resulting building outlines, all the line segments in our experiments are treated equally and, therefore, \mathbf{W} is an identity matrix. In Equation 2.19, the last two columns correspond to one main orientation. In the case of having more than one main orientation, two extra columns are added to the matrix \mathbf{A} for each additional main orientation, which contains the coordinates of the additional main orientation. Moreover, one additional row is added to the matrix \mathbf{A} as a normalization constraint. Furthermore, for each additional main orientation, one zero row is inserted at the end of the matrix \mathbf{B} and a normalization constraint is inserted to the end of vector \mathbf{w} . Changing the matrices mentioned above based on a new main orientation and adjusting all the building outlines in a different orientation at once discriminates the method developed in this dissertation from other similar methods, such as [64, 135].

2.2.4 Line Segment Intersection and Connection

In the last step, we intersect and connect each line segment to its closest neighbor according to the following rules:

1. Any two parallel line segments with a perpendicular distance larger than five pixels are joined by inserting a connection line perpendicular to both line segments (please refer to Figure 2.9. (a)). If the endpoints of the line segments are not aligned (e.g. \mathbf{l}_1 and \mathbf{l}_3 in Figure 2.9. (a)), one of the line segments is selected randomly and its length is adjusted (e.g. either elongated or shortened) to make its endpoint align to the endpoint of the other line segment. The threshold of five pixels is selected according to the resolution of the specific PAN image and the DSM data (about 0.5 m). In this resolution, five pixels represent 2.5 m, which is negligible for urban buildings within the generalization tolerance.
2. For the parallel line segments with perpendicular distance smaller than or equal to five pixels, we insert a line with a minimum distance from the other line segments and extend it to cover the extent of all the line segments (please refer to Figure 2.9. (b)).

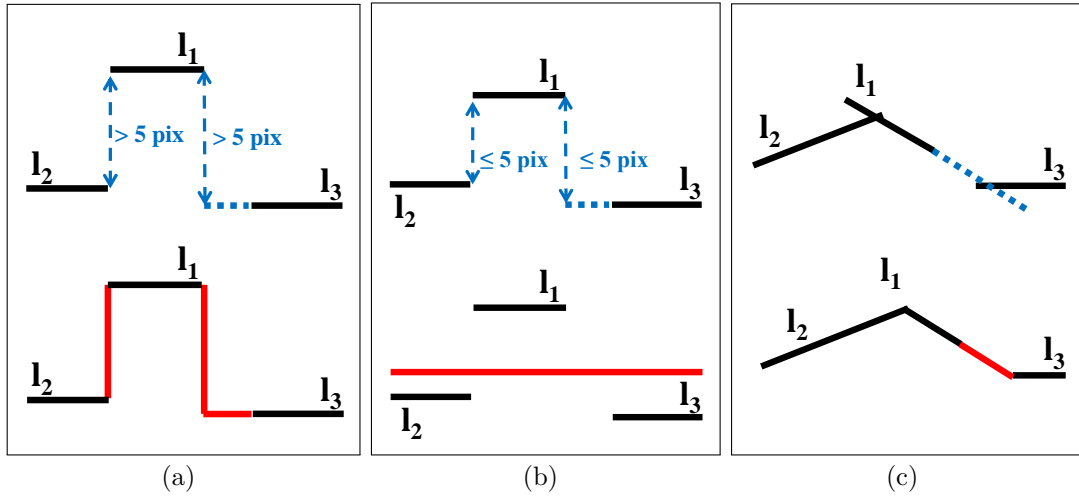


Figure 2.9: Each sub-figure shows a condition and its corresponding constraint for connecting the line segments and the result of the line segment connection. (a) Join parallel lines with a perpendicular distance larger than five pixels. (b) Join parallel lines with a perpendicular distance smaller than or equal to five pixels with the assumption, $l_2 > l_1, l_3$. (c) Join intersecting lines.

3. In the case of intersecting line segments, we adjust their length until their neighboring end points meet (please refer to Figure 2.9. (c)).

2.3 Experimental Results and Discussion

In this section, we evaluate the precision of the building outlines extracted using our proposed method both qualitatively and quantitatively. Consequently, we analyze 10 regions of a PAN image of Munich, acquired by the WorldView-2 satellite (Figure 2.10 and Table 2.1). The regions are selected so that they contain a variety of building structures with different degrees of complexity (e.g. multiple main orientations, buildings with inner yards, different roof types). We perform all the steps developed in Sections 2.1 and 2.2 to extract a building's outline.

Our proposed method relies on some parameters in its various steps. Table 2.2 shows the parameters present in each step and the values assigned to them in our experiments. As described in the corresponding sections, most of these parameters depend on the resolution of the PAN image being used. Therefore, they can be used as they are for new images with the same resolution and they should be adapted accordingly for new images with a different resolution.

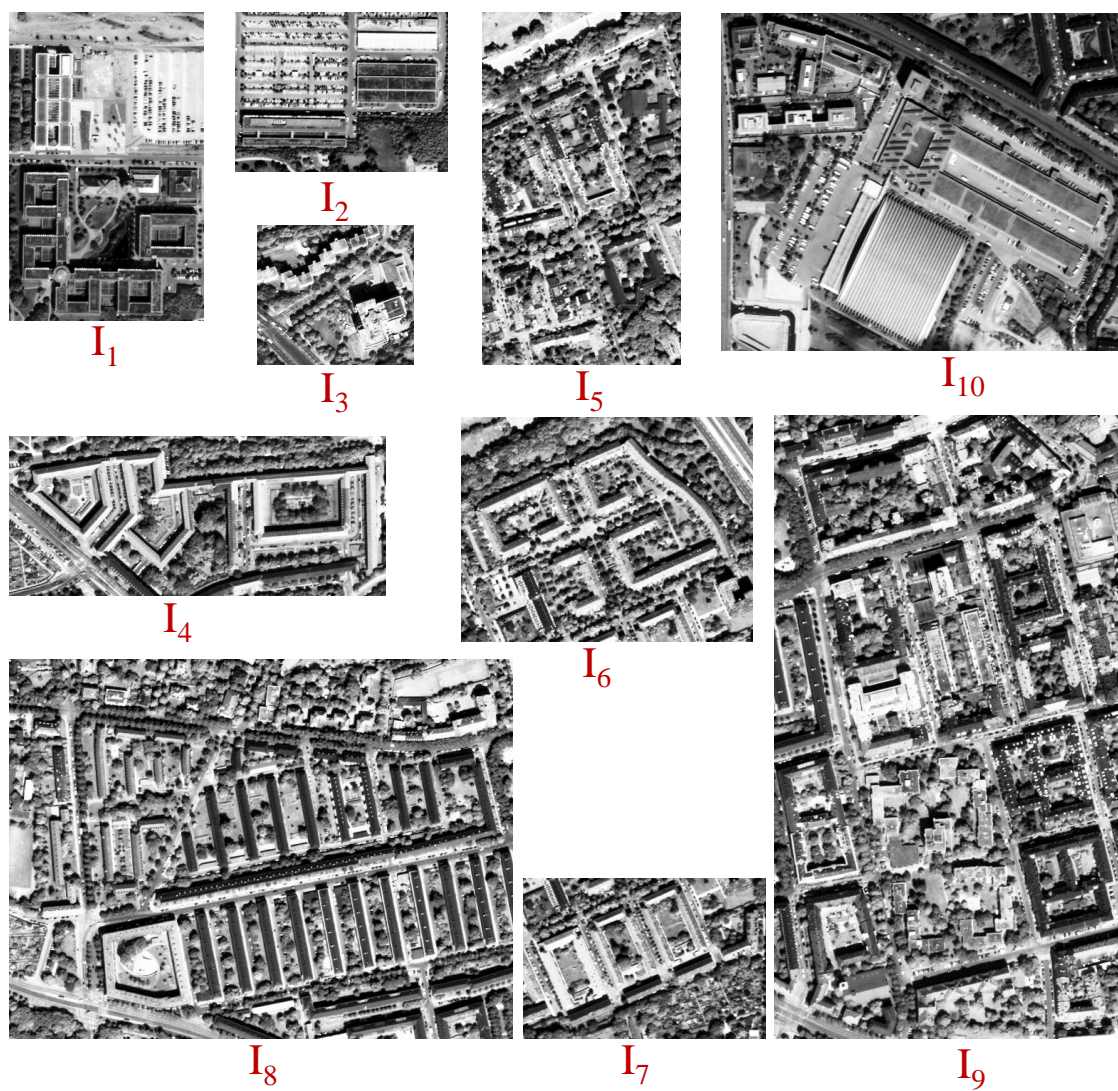


Figure 2.10: Orthorectified PAN images used in our experiments.

2. Building Boundary Extraction

Image	Image size (pixels)	GSD	Roof Types
I_1	745×470	50 <i>cm</i>	Flat
I_2	400×530	50 <i>cm</i>	Flat
I_3	350×390	50 <i>cm</i>	Flat
I_4	400×950	50 <i>cm</i>	Pitched
I_5	800×450	50 <i>cm</i>	Pitched and Flat
I_6	500×650	50 <i>cm</i>	Pitched
I_7	400×590	50 <i>cm</i>	Pitched
I_8	950×1300	50 <i>cm</i>	Pitched
I_9	1350×750	50 <i>cm</i>	Pitched and Flat
I_{10}	820×1070	20 <i>cm</i>	Pitched and Flat

Table 2.1: Specifications of the images used in our experiments.

Steps	Parameters	Values
Mask preprocessing (Section 2.1.1)	Threshold for removing small masked areas	300 pixels
	Morphological opening disc size	3 pixels
	Morphological closing disc size	2 pixels
Feature Classification (Section 2.1.3)	SVM parameters	Empirically selected
Mask Postprocessing (Section 2.1.4)	Morphological closing disc size	3 pixels
	Threshold for gap filling	400 pixels
Line Segment Extraction (Section 2.2.1)	Short line extraction threshold (RMSE)	1 pixels
	Length threshold for deleting small line segments	5 pixels
Line Segment Regularization (Section 2.2.2)	Arc length distance for main orientation computation	5 pixels
	Point distance of skeleton for building partitioning	5 points
	Number of nearest neighboring line segments to a selected point on skeleton	10 lines
Line Seg. Intersec. and Connec. (Section 2.2.4)	Distance between line segments	5 pixels

Table 2.2: Parameters Used in Our Approach

2.3.1 Mask Refinement

In order to conduct a qualitative evaluation of the mask refinement process, Figure 2.11 exemplifies the refinement applied to the masks for I_1 , I_3 , I_4 and I_5 . In this figure, the resulting refined building area is depicted in green (both light and dark green), where the dark green depicts the gaps in the masks which have been filled during the refinement process. The regions depicted by red have been removed by the refinement process from the masks specified. According to the results, building edges in the refined masks are finer and more precise than in the other masks.

In order to evaluate the refined building masks quantitatively, we compare them pixel by pixel to the footprint-based mask as reference data which is generated manually and compute the *Quality* measure,

$$Quality = \frac{TP}{TP + FP + FN}. \quad (2.26)$$

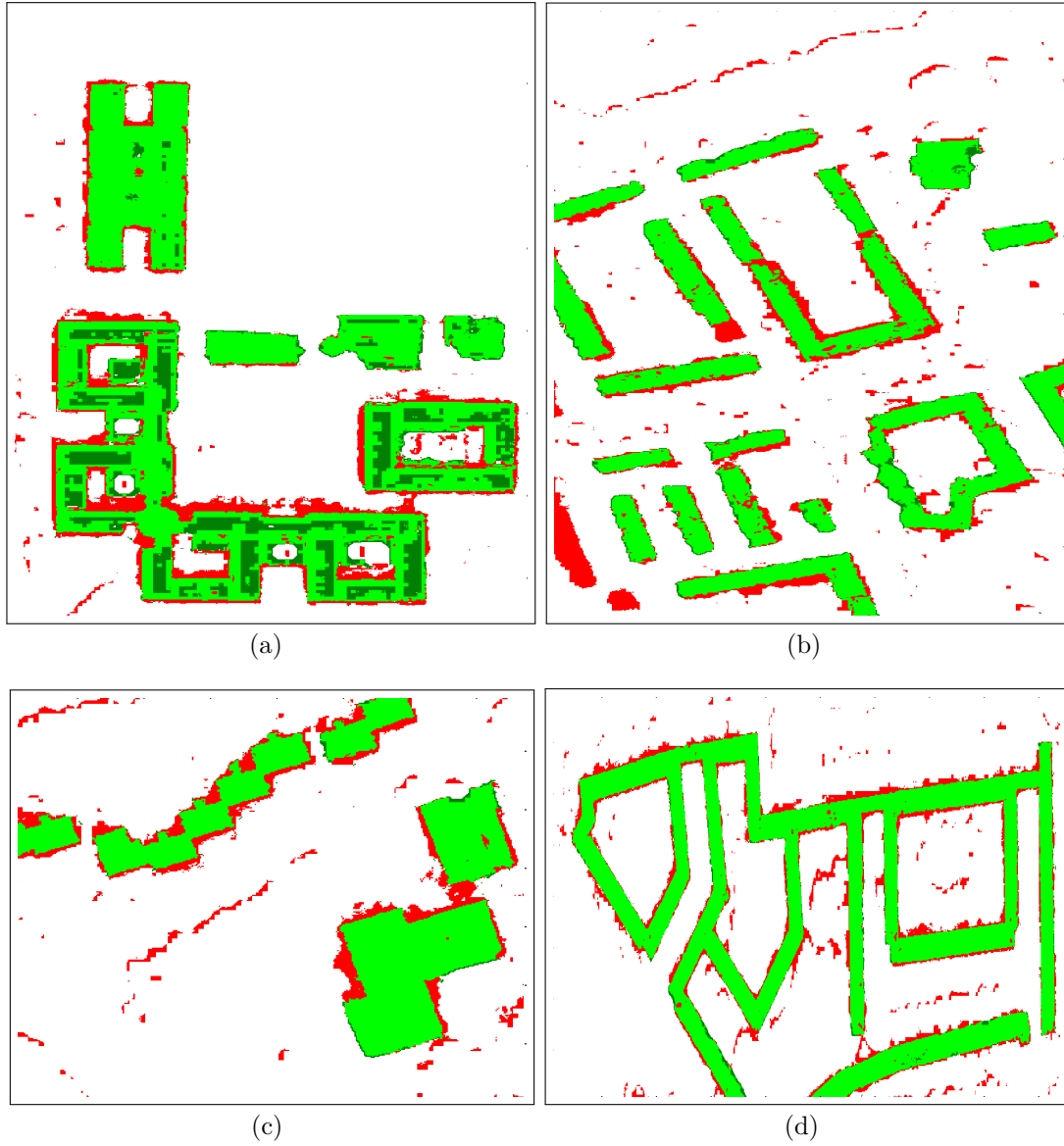


Figure 2.11: Refined masks (bright and dark green areas) together with the areas removed from the masks specified (red areas). (a) I_1 , (b) I_5 , (c) I_2 , (d) I_4 .

2. Building Boundary Extraction

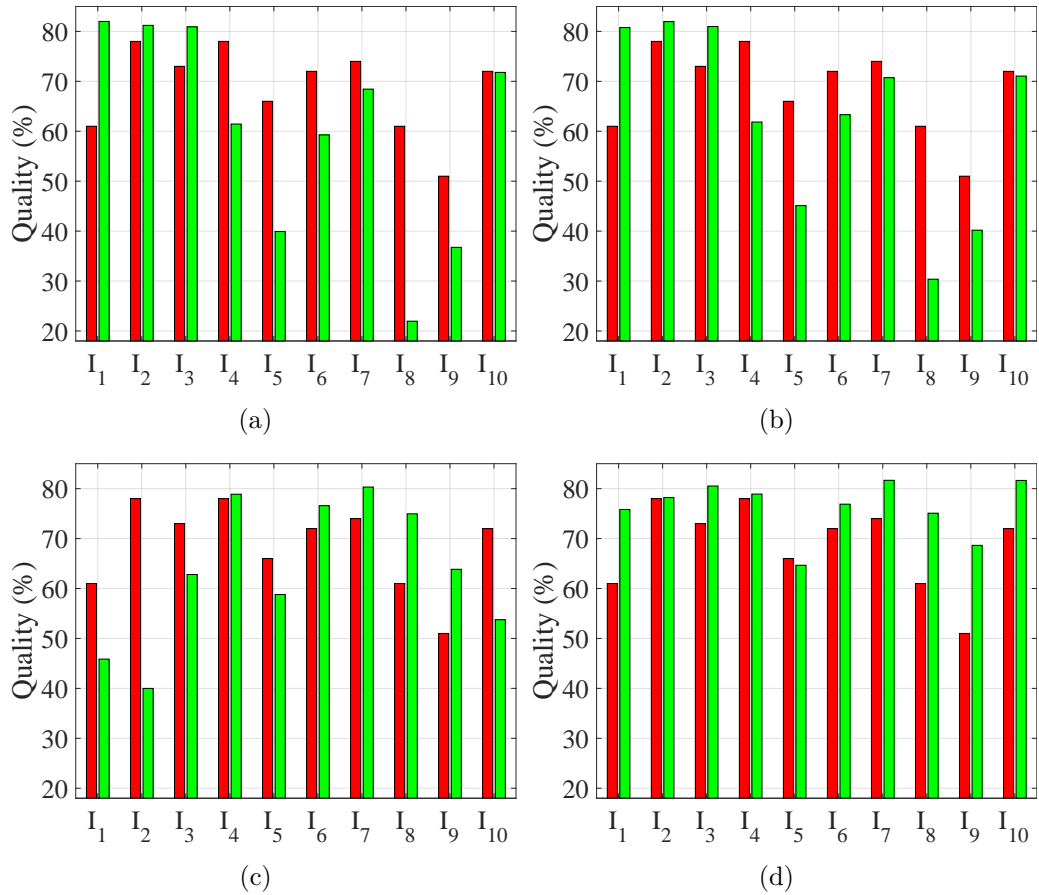


Figure 2.12: The Quality measure of the **given** and refined masks of the images $I_1 - I_{10}$ using SVM classification. Regarding training, 30 % of the pixels within the building boundaries, indicated by the masks, is sampled from images: (a) I_1 and I_2 , (b) I_1 , I_2 and I_3 , (c) I_4 , I_6 and I_8 , (d) I_1 , I_3 , I_4 and I_8 .

This measure provides an overall quality degree of the masks by considering the building detection and boundary delineation. In Equation (2.26), TP (*True Positive*) and FP (*False Positive*) are the number of pixels which are correctly and incorrectly considered as building by the mask, respectively. In addition, FN (*False Negative*) indicates the number of building pixels which are missed by the mask.

Figure 2.12 shows the *Quality* of the specified and the refined masks. In this figure, each column shows an experiment in which the training data is sampled from specific images of our dataset. The SVM for each experiment is trained and tested three times using three different pixel samplings (where the SVM parameters are set empirically) to cross-validate the classification. We test all three SVM models on every images of our dataset, resulting in three sets of refined masks for each

experiment. The three sets of masks are then evaluated and the average Quality value is considered for each image.

In Figure 2.12. (a), the training data is sampled from images I_1 and I_2 which contain only flat roof buildings. As the results show, the refined masks outperform only for the images containing flat roof buildings (e.g. I_1, I_2, I_3), whereas the refined masks are even worse than the other masks in the case of pitched roofs. In Figure 2.12 (b), we train the SVM model on samples from I_3 (which also contains only flat roof buildings) in addition to I_1 and I_2 to demonstrate that adding more similar knowledge does not help to compensate the SVM model’s bias. The results indicate that there is practically no change in the classification performance (i.e. mask quality) compared to the previous experiment. Figure 2.12. (c) illustrates the model bias when it is trained on pitched roof buildings. As the figure illustrates, while the refined masks outperform the original masks for the pitched roofs (e.g. I_4, I_6, I_7, I_8, I_9), they are worse for the flat roofs (e.g. I_1, I_2, I_3).

Considering the results of these experiments, we conclude that SVM models should be trained by various building roof types and surroundings to avoid model biases. Figure 2.12. (d) shows the quality of the refined masks using an SVM model trained on both flat and pitched roof buildings. The results indicate that the quality of the refined mask for most of the images is significantly higher than that of the other mask. Therefore, we will use the refined masks resulting from this experiment for the building outline extraction task.

2.3.2 Building Outline Extraction

In this section, we extract outlines of the buildings through the line segment extraction, regularization, adjustment, intersection and connection steps using the refined masks.

2.3.2.1 Qualitative Evaluation

Figure 2.13 and 2.14 show the final extracted building outlines depicted on our ten sampled PAN images. As demonstrated by the results, the extracted outlines are significantly close to the buildings’ original edges despite the complexity of their structures, such as the existence of inner yards. We extract the line segments of the buildings’ outer boundaries and those of their inner yards separately to delineate the edges of buildings with inner yards. We then consider all the extracted line segments together to determine the buildings’ main orientations and regularize the line segments. Since the edges of inner yards are usually aligned with a building’s outer edges, considering them together increases the robustness of the resulting building outlines.

In addition to its superior results, there are cases where our approach fails to extract building outlines correctly. Figure 2.15 exemplifies these cases. In Figure

2. Building Boundary Extraction

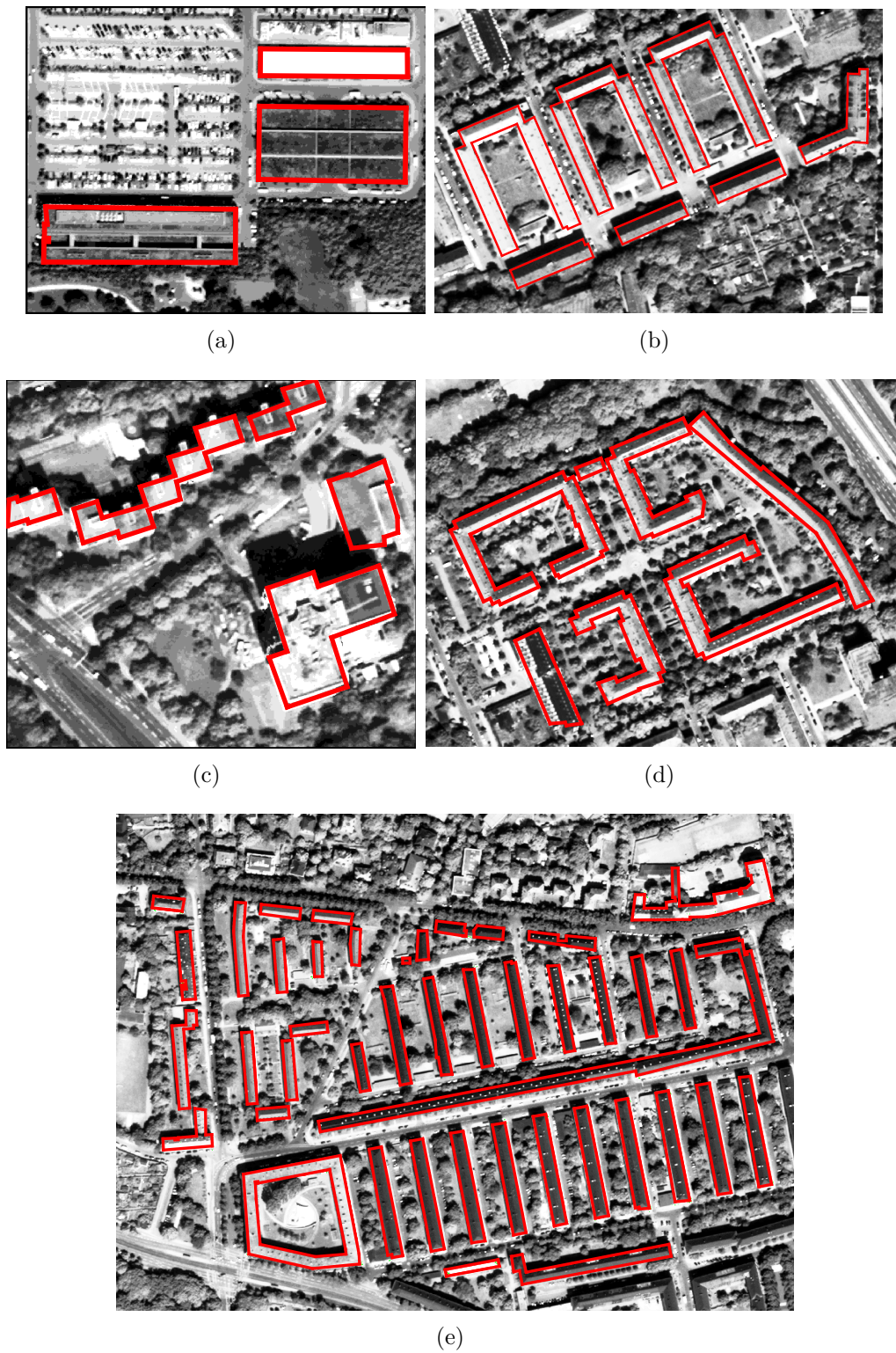


Figure 2.13: Extracted outlines of the buildings within the images (a) I_2 , (b) I_7 , (c) I_3 , (d) I_6 and (e) I_8 using our proposed approach.

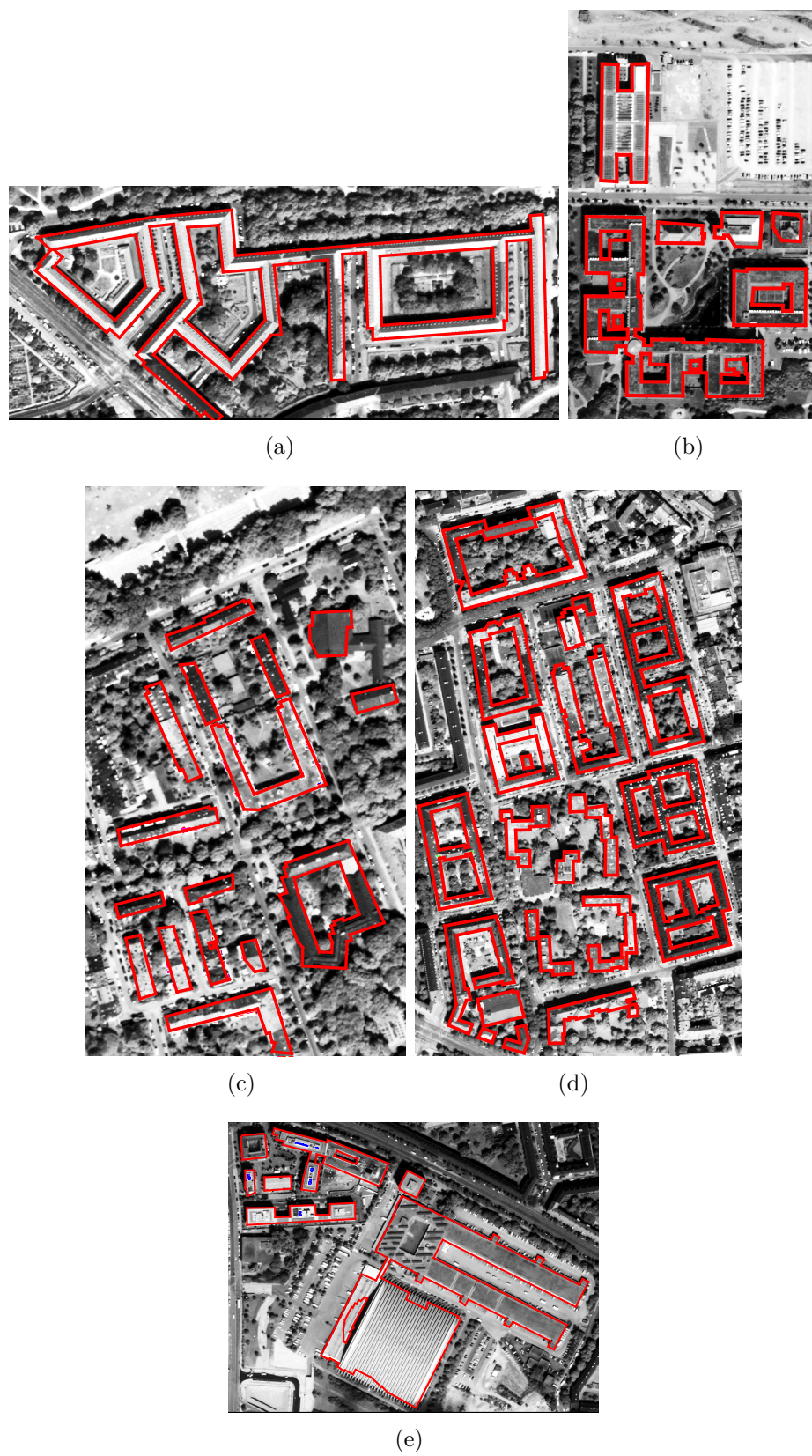


Figure 2.14: Extracted outlines of the buildings within the images (a) I_4 , (b) I_1 , (c) I_5 , (d) I_9 and (e) I_{10} using our proposed method.

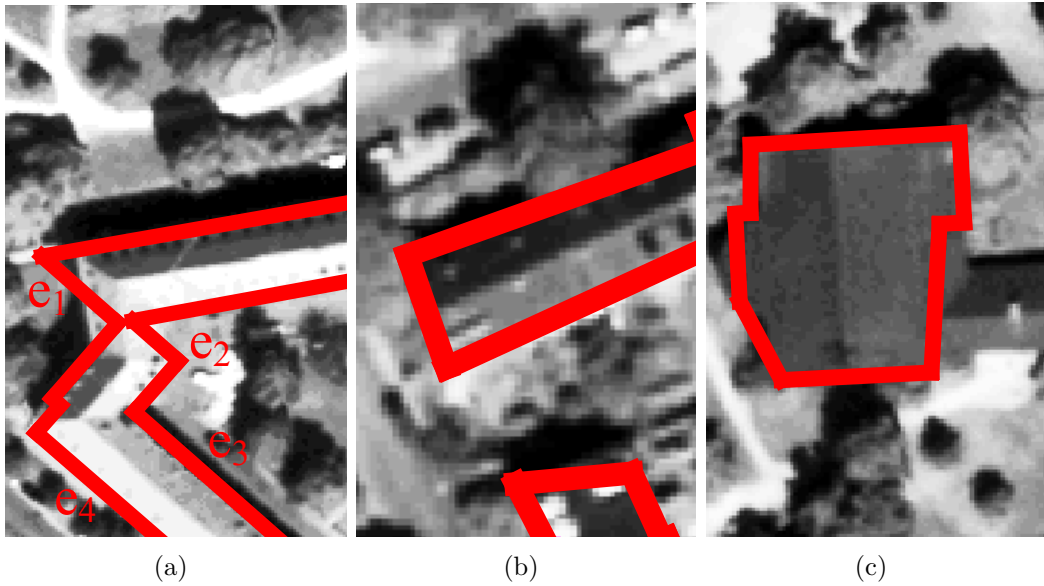


Figure 2.15: Failure case examples of the proposed approach.

2.15. (a), the building edges e_1 and e_2 were not correctly delineated, probably due to the imperfection of the building skeleton computed, caused by the selected threshold for the morphological process in the regularization step (see Section 2.2.2). This affects the building decomposition (see Section 2.2.2) in which those edges were grouped into the main orientation class of the e_3 and e_4 . In Figure 2.15. (b) and (c), the extracted building outlines were affected by the incompleteness of the building masks caused by, for example, occlusion (e.g. trees), shadow and the height threshold used for mask generation, but the representation is still not far away from reality.

2.3.2.2 Comparison to an Existing Line Simplification Method

Line simplification methods, such as DP [67] have been conventionally used to extract building outlines. However, due to their shortcoming in delineating fine details on building, such as sequential corners, in this dissertation, we developed a new line segment extraction method, described in Section 2.2.1. Figure 2.16 compares the building outlines derived by our proposed method and the DP method with the building footprints. The results for both methods are presented after regularization, as explained in Section 2.2.2. As the figure shows, the building outlines obtained based on our proposed method are usually closer to the footprint compared to the results based on the DP method.

In Figure 2.16. (a), due to the small size of the building which limits the number of the available line segments together with its multiple main orientations, the building outlines computed based on DP [67] cannot meet the building's original

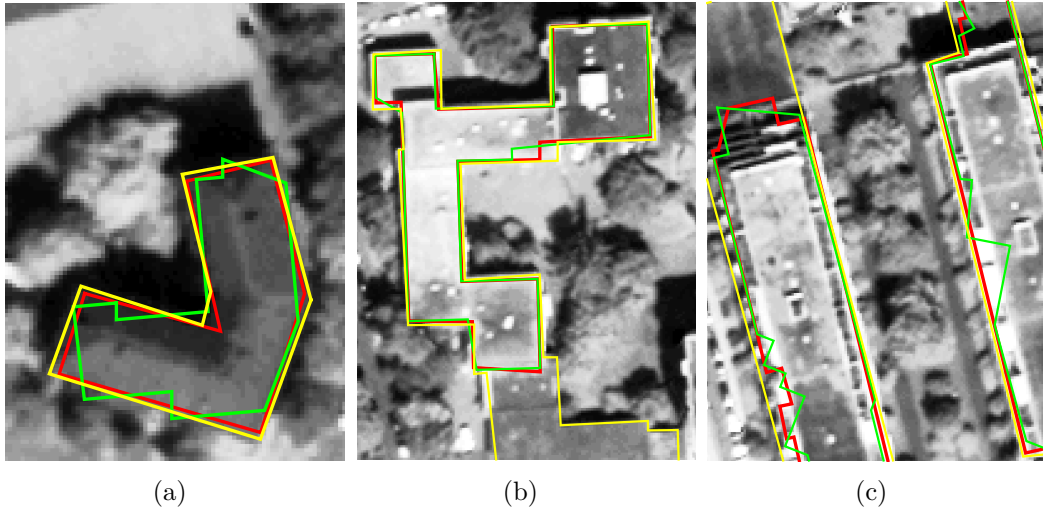


Figure 2.16: Comparison of the building outlines based on our proposed line segment extraction method (red) and the DP algorithm (green), after the regularization step. The yellow lines depict the building footprints.

boundaries. However, our proposed method results in a more accurate building outline because it relies on every single line segment. In Figure 2.16. (b), the results of both methods are quite similar and pretty close to the building boundaries. However, a part of the building (the bottom part) remains undetected by both methods. This is due to the incompleteness (caused, e.g., by occlusion, or shadow) of the building mask. Figure 2.16. (c) shows that the DP method performs poorly in delineating building corners composed by an intersection of sequences of small line segments, whereas our proposed method can extract these details precisely.

2.3.2.3 Quantitative Evaluation

We compare the building outlines extracted to the reference data using a newly introduced metric, namely *polygons and line segment (PoLiS)* [136], to obtain a quantitative evaluation. The PoLiS metric has been proposed for measuring the similarity of any two polygons [136]. It is a positive-definite and symmetric function which satisfies the triangle inequality. The PoLiS distance changes linearly regarding small translation, rotation and scale changes between the two polygons.

In our experiments, we compute the average distance between the polygon vertices extracted and their closest vertices on the reference polygon to measure the PoLiS distance between a building's extracted outline (polygon) and its corresponding reference polygon. The average distance between the reference polygon's vertices and their closest vertices on the polygon extracted is then computed. The PoLiS distance between the two polygons is computed as the summation of the two average distances. Let $v1_j \in \mathbf{V1}, j = 1, \dots, J$ in each vertex of $\mathbf{V1}$ and $v2 \in \partial\mathbf{V2}$ be its

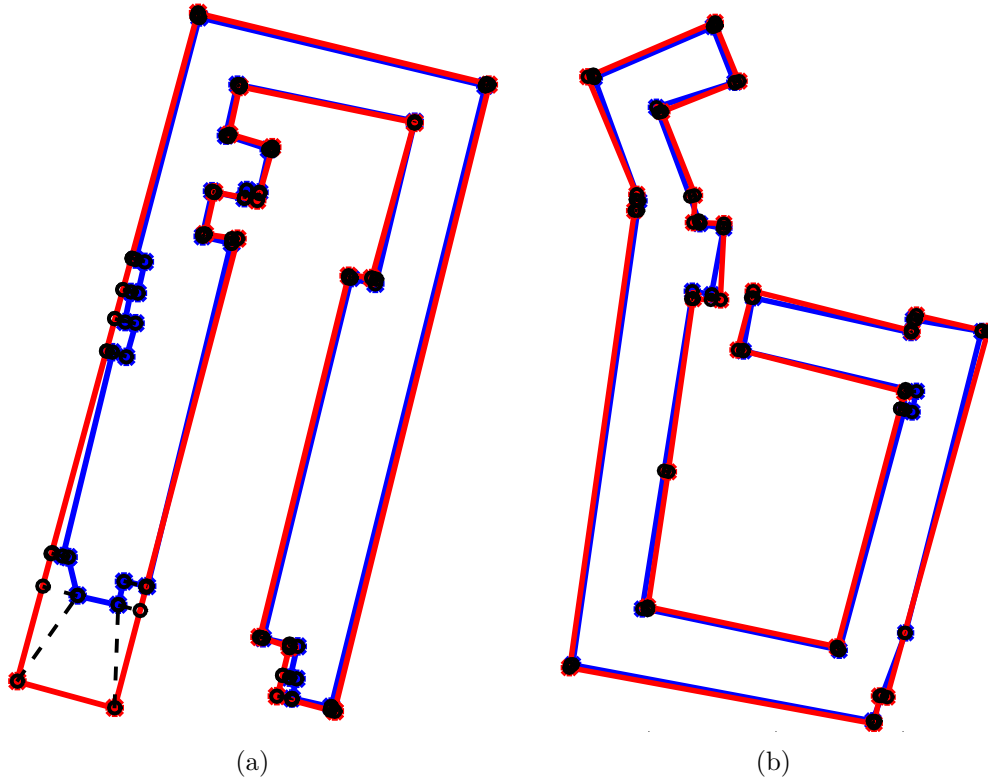


Figure 2.17: Illustrations of the PoLiS distance of the extracted (blue) building polygons to the reference (red) ones. (a) PoLiS = 5.19, (b) PoLiS = 1.60.

closed point (not necessarily a vertex) on the polygon $\mathbf{V2}$. The average distance between $v1_j$ and $v2$ is then a directed PoLiS distance $\vec{d}_p(\mathbf{V1}, \mathbf{V2})$ between polygons $\mathbf{V1}$ and $\mathbf{V2}$ and is defined as

$$\vec{d}_p(\mathbf{V1}, \mathbf{V2}) = \frac{1}{J} \sum_{v1_j \in \mathbf{V1}} \min_{v2 \in \partial \mathbf{V2}} \|v1_j - v2\|. \quad (2.27)$$

Since the directed PoLiS distance \vec{d}_p is made symmetrically, PoLiS metric is defined by summing and normalizing the directed distances as a relationship as follows:

$$\vec{d}_p(\mathbf{V1}, \mathbf{V2}) = \frac{1}{2J} \sum_{v1_j \in \mathbf{V1}} \min_{v2 \in \partial \mathbf{V2}} \|v1_j - v2\| + \frac{1}{2K} \sum_{v2_k \in \mathbf{V2}} \min_{v1 \in \partial \mathbf{V1}} \|v2_k - v1\|. \quad (2.28)$$

Figure 2.17 exemplifies the PoLiS distance measurement. The average PoLiS distances of the extracted building outlines in our ten images are displayed in Figure 2.18. (a). In this figure, the red bars indicate the PoLiS distance when only the

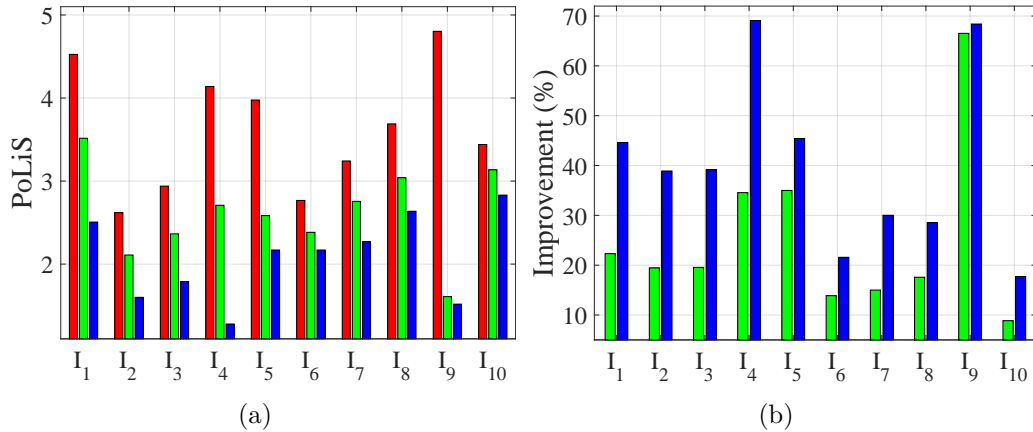


Figure 2.18: (a) PoLiS distance between the extracted and the reference building outlines when only the DSM-based masks specified (the red bars), refined masks (the green bars), and regularization and line segment adjustment (the blue bars) are used. (b) Degree of improvement to the building outlines extracted using the DSM-based masks specified when the refinement step (the green bars), and regularization and line segment adjustment steps (the blue bars) are employed.

specified masks are used, the green bars represent the PoLiS distance after mask refinement, and the blue bars demonstrate the PoLiS distance when the refined masks are used coupled with line segment regularization, adjustment, intersection and connection steps. As demonstrated by the results, using the mask refinement step reduces the PoLiS distance between the extracted outlines and the reference data to a large degree. Moreover, applying the regularization, adjustment, intersection and connection steps further reduces the PoLiS distances significantly.

Figure 2.18. (b) indicates up to which degree the mask refinement (the green bars), and the line segment regularization, adjustment, intersection and connection steps (the blue bars) improve the building outlines obtained by using only the masks specified. According to the results, our proposed additional modules can increase the precision of the extracted building outlines up to 70 % for some images (e.g., I_4 and I_9). Refining the building mask improved the building outlines for I_4 by about 35%, while the results' precision increased up to about 70 % after the line segment regularization, adjustment, intersection and connection steps. This shows that the imperfections of the building outlines are caused by both the faults in the building masks and the complexities of the building structures. However, the improvement for I_9 is already almost achieved by refining the building mask, which indicates that the building complexities in I_9 are less problematic than in I_4 .

Table 2.3 represents the PoLiS distance of the building outlines obtained based on our line segment extraction method and the DP method for the buildings shown in Figure 2.16. Results indicate that the higher accuracy of the outlines extracted are based on the proposed method.

Image	Our method	Douglas-Peucker (DP)
Figure 2.16. (a)	1.34	3.17
Figure 2.16. (b)	9.11	9.24
Figure 2.16. (c)	5.19	5.54

Table 2.3: Comparing the PoLiS distance of the building outlines extracted using our method and DP

2.4 Summary and Conclusion

This chapter proposes a heuristic approach for extracting and simplifying building outlines. It detects buildings and generates their mask using DSM data. The resulting masks are further refined based on high spatial resolution PAN images to represent the building edges closer to the original ones. The refined masks are then used in a new method composed of line segment extraction, regularization, adjustment, intersection and connection steps to extract building outlines as polygons. The main contribution of this chapter of the dissertation is extracting very complex rectilinear polygons of buildings (with different orientations and including inner yards). Experimental results show that the building outlines extracted are close to the building’s original edges to a high degree. Moreover, our method can be generalized to various types of buildings and is robust against complexities of building structures, such as the existence of inner yards and multiple main orientations. The comparison with the state-of-the-art methods show clearly the advantages and importance which are gained through our methodology.

3

Building Boundary Generalization and Decomposition

In the last chapter, the building outlines were extracted for ten different areas of satellite image. They were extracted from refined DSM-based building masks which sometimes have inaccurate details because of the existing noise in the boundary of the buildings in the DSM and its related mask. On the other hand, footprints generated from high-resolution CM can be very detailed which is not desirable for many applications. A high degree of detail requires large storage space and high computational power, which cannot be provided in every application. Employing generalization and decomposition algorithms on the building outlines have been proposed in the literature to enhance the building outlines and, simultaneously, reduce the complexity of the 3D model of an urban area [137, 138, 139, 140]. The main goal of the generalization algorithms is to remove unnecessary details while preserving the general structure and visual impression of the spatial situations. Additionally, the generalization of the building footprint reduces the artifacts which appear in the decomposition process. Since the decomposition of a building polygon into simple rectangular shapes is based on the line segments of the building outline, many details on the building outline results in an over-decomposition (i.e. small rectangular shapes which are not meaningful for 3D building modeling). Therefore, generalization is an important step before footprint decomposition in our procedure. The main work in the generalization and decomposition of a building footprint has been carried out by Kada and Luo [138] using half-spaces, which was improved later by [141] for preserving the main building shape and facade lines. Sester [142] presented some general rules for the simplification of building polygons. Inspired by the last method, we generalize building footprints based on their rules with small improvements, which are described in Section 3.1. The order of the steps within the generalization and decomposition process are shown in Figure 3.1. In this diagram, the building boundaries include building outlines which are the building outlines extracted in Chapter 2 and the footprint which is from a small area of the CM. The

3. Building Boundary Generalization and Decomposition

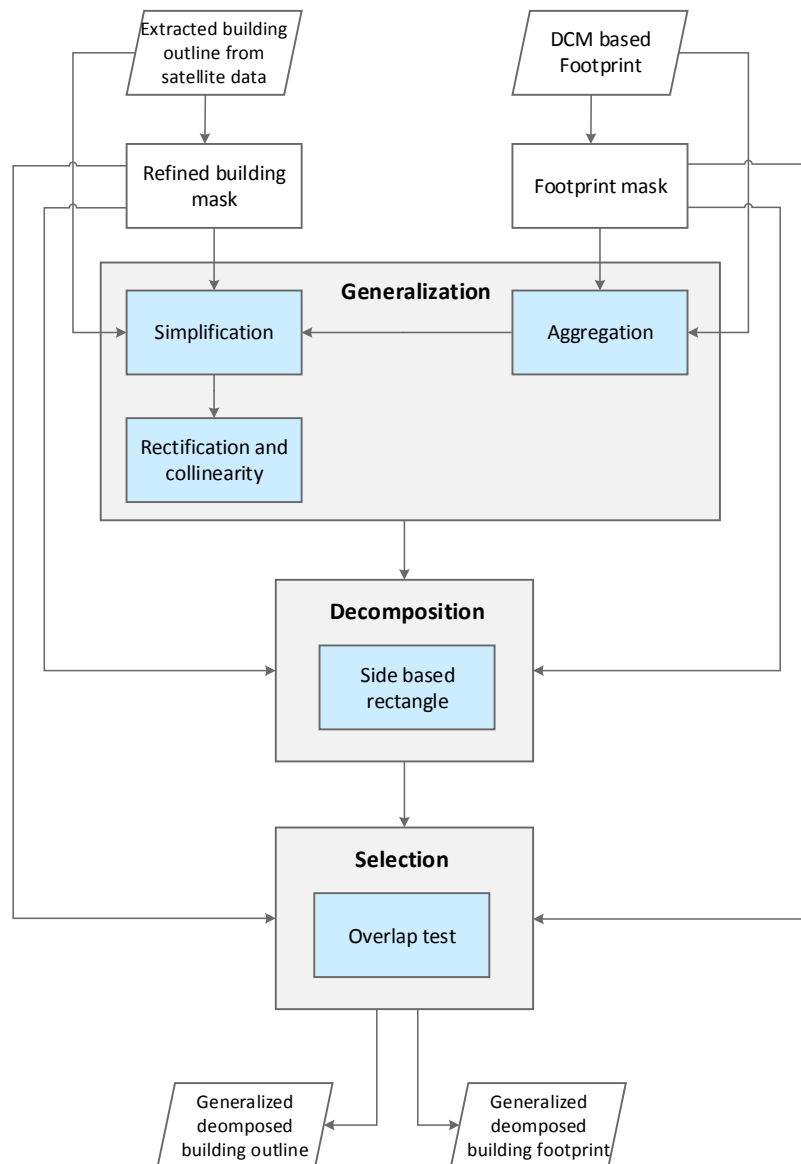


Figure 3.1: Generalization and decomposition process

building masks are binary images generated from the building outline extracted and footprint. Extracted building outlines, footprints and their corresponding building masks are used in all the generalization, decomposition and selection process, as shown in Figure 3.1. After generalization, the building polygon is decomposed into rectangular shapes using the line segments of the building outline (Section 3.2). Subsequently, those among the rectangles generated that have maximum overlap with the building mask and minimum overlap with each other are selected to cover the whole building area (Section 3.3). Finally, the procedure is employed on the building outline extracted and footprint of different areas (Section 3.4).

3.1 Building Boundary Generalization

As Figure 3.1 shows, the generalization process consists of the aggregation (only used for the footprint), simplification and rectification steps. The generalization process is a simplification of the building boundary to persevere with the minimum length (or generalization threshold) of the building facade by discarding the line segments of building outlines which are shorter than this minimum length. Simplification brings the building polygon closer to a rectangular shape. The building polygons become totally rectangular and approximately collinear lines become straight by utilizing the rectification step.

Footprints of a city usually include adjacent buildings, while their distance is not noticeable at the resolution given or even share common line segments (topological adjacency). The adjacent buildings are aggregated by removing the common line segment to generate a single building block which eases the simplification process and reduces the amount of data significantly. In order to find the common line segments, the common vertices of the rectilinear adjacent buildings' polygons are obtained by detecting the junction points as the ones which are repeated more than three times in the coordinate vector of the building polygons (the yellow points on Figure 3.2). The line segments of the building outlines related to the junction points are then found (the blue lines in Figure 3.2). As can be seen in Figure 3.2, the line segments detected refer not only to the common line segments, but also to the outer part of the building outline.

Therefore, in the next step, common line segments are separated from the outer line segments. The building outlines lie on the building edges, which have high gradient magnitude and gradient direction changes on the image or on the building mask, in contrast to the common line segments, as Figure 3.3 (a) shows. Therefore, measuring the gradient on the detected line segments of the building outlines allows us to discriminate outer parts of the building outline and common line segments. In our experiments, the buffer is split into N bins with the size of 2×2 pixels on each side of the buffer. The method proposed in [122] is used to calculate the gradient. This method considers a buffer around line segments of building outlines, where the



Figure 3.2: Illustration of common line segments and wrongly selected common line segments

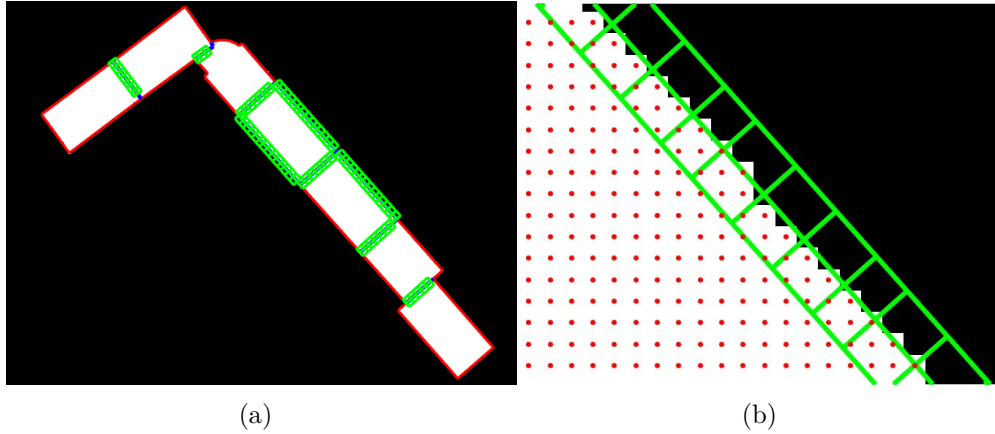


Figure 3.3: Buffer zone around each line which has been introduced as a common line; (a) buffer's bins around a common line; (b) illustration of building mask points inside the buffer's bins

buffer is split into smaller bins, as shown in Figure 3.3 (b). Equation 3.1 shows how to calculate the gradient G using the buffer. According to this equation, the gradient is the sum of the rates between the number of points within the first Bin_1 and the second Bin_2 sides of the buffer of each bin. In order to keep the ratio between 0 and 1, the minimum point count is always divided by the maximum point count for every bin [122].

$$G = 1 - 1/N \sum_{i=1}^N \frac{\min(Bin_{i1}, Bin_{i2})}{\max(Bin_{i1}, Bin_{i2})} \quad (3.1)$$

When the gradient of the line segment of the building outline equals or is close to 1, the line segment is classified as an outer part of the building outline.



Figure 3.4: Aggregated footprint

After removing the common lines, the order of the building polygon vertices is arranged again in a clockwise direction. The final result after removing common lines is shown in Figure 3.4.

Three rules should be considered, depending on the directions and lengths of the predecessor (l_{n-1}) and successor (l_{n+1}) of a short line l_n to perform a simplification of the building outlines extracted or CM-based footprints. These rules are based on three cases introduced in [142] and a modification is proposed in this dissertation in which several sequential lines with small difference angles are replaced with a single line as follows:

1. The predecessor and successor lines of the small line segment l_n have the same direction. The longer predecessor l_{n-1} is intersected with the next approximately orthogonal edge l_{n+2} (Figure 3.5. (a)). In the case of a longer successor l_{n+1} , it is intersected with the previous approximately orthogonal edge l_{n-2} , so that l_n and l_{n-1} are eliminated.
2. The predecessor and successor of the small line segment l_n have opposite directions; this is called an extrusion or intrusion. An extrusion cut the shortest edge of predecessor and successor. In Figure 3.5. (b), for instance, l_{n+1} is shorter, therefore, l_{n+2} is intersected by l_{n-1} , and l_n and l_{n+1} are eliminated.
3. Two sequential lines l_n and l_{n-1} , which have small orientation differences, are replaced by one line. The longer line is extended to the first point of the next line (Figure 3.5. (c)).

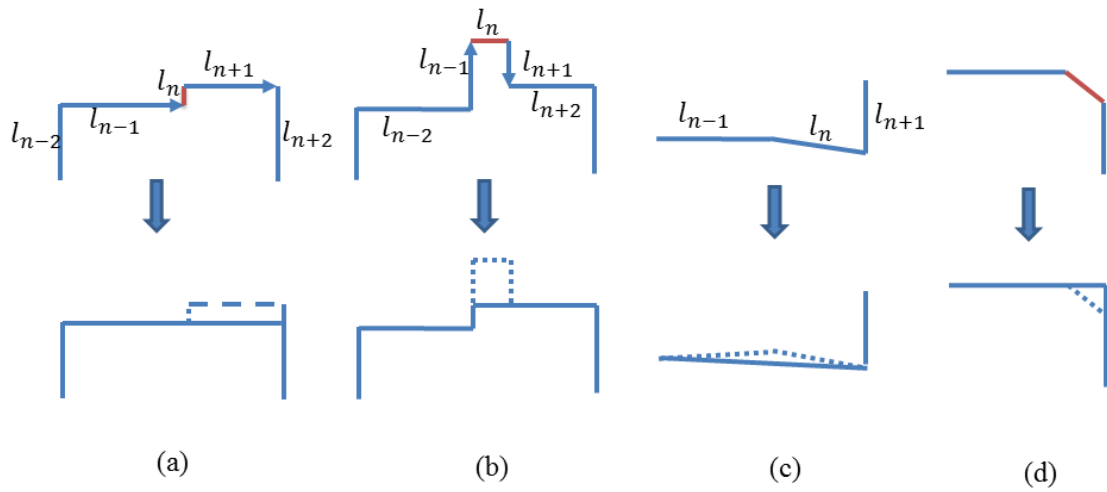


Figure 3.5: Simplification rules. (a) Unidirectional lines with small offset l_n , (b) extrusion, (c) approximately parallel lines and (d) corners

4. The predecessor and successor are intersected with each other for corners with non-orthogonal small line segment l_n (Figure 3.5. (d)).

This operation is iteratively employed to all edges, including those which are generated through the process.

After the simplification of the building outlines, sequential line segments still may have small deviations from rectangularity or collinearity. Rectification is employed on the line segments of the building polygon to remove these deviations. The rectification is performed by determining main orientations and SVD-based line fitting, which were explained in Chapter 2. Figures 3.6 and 3.7 show aggregation, simplification and rectification of the extracted building outlines and footprint.

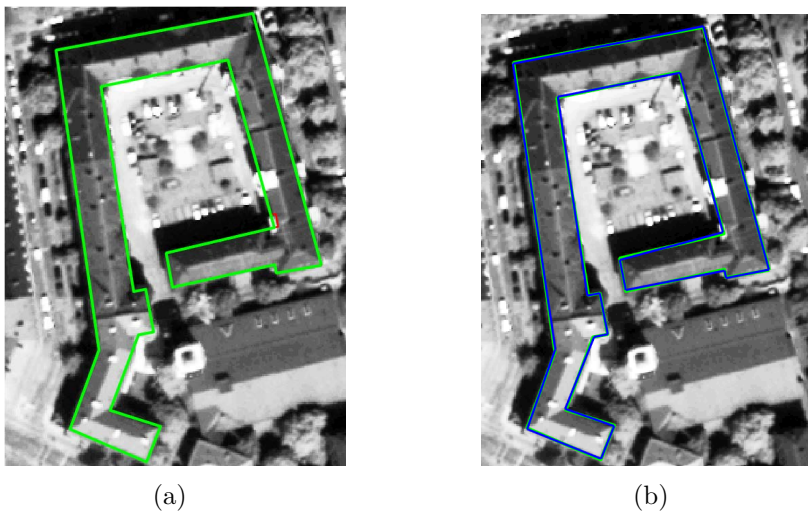


Figure 3.6: (a) Outline simplification red: Before simplification green: After simplification (b) rectified building outline green: before rectification blue: after rectification

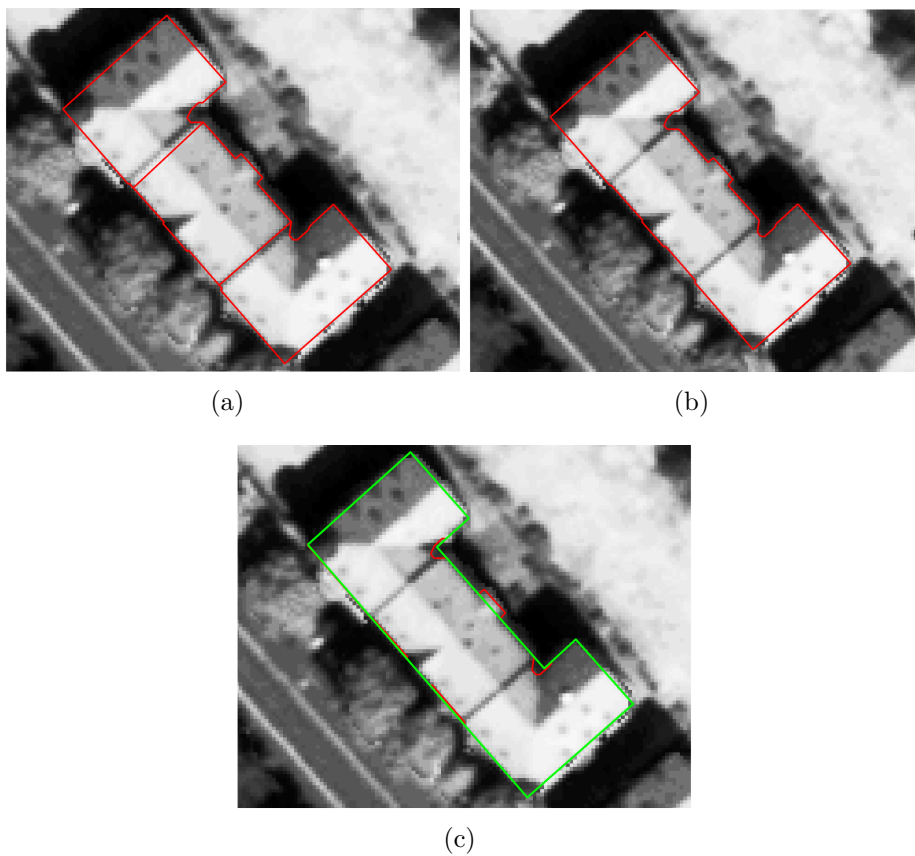


Figure 3.7: (a) Footprint (b) aggregated footprint (c) red: before simplification green: after simplification

3.2 Line Segment-based Building Polygon Decomposition

Buildings in urban areas usually have rectangular shapes, in some cases, with more than one main orientation, as mentioned in Section 2.2.2. Footprint decomposition is an important prerequisite step for model-driven methods to reconstruct 3D building models [65, 94]. Decomposition of a building outline splits it into rectangular shapes. Two different methods have been proposed for this purpose: Fitting rectangular shapes [65] and extending the line segments of the building outlines [94]. A new decomposition method is proposed in this dissertation based on the idea that each line segment of a building outline is a potential component for generating a rectangle. Therefore, every line segment is used for splitting the 2D building outline into rectangular shapes. A heuristic approach is then used to select the best rectangles by which not only the whole building mask is covered, but they also show a minimum overlap with each other.

3.2.1 Creating Decomposition Rectangular Shapes

The decomposition is performed by creating a rectangle from each generalized line segment of the building outline. Creating rectangles starts from the longest line segment. Each line segment is moved in the direction of the pixels of the building mask iteratively until it meets another parallel line segment of the building outline or footprint (Figure 3.8). Consequently, a buffer is considered around the line segments. If the moving line segment lies completely inside one of the buffers, it is considered as meeting the corresponding line segment. A rectangle is generated using these two parallel line segments. In Figure 3.9, for instance, after moving the line segment several times (shown in blue), it lies in the buffer of another line segment of the building outline (yellow rectangle in Figure 3.9). Sometimes, the buffers cannot cover the two endpoints of the moving line segment but only one of them. In this case, the segment which meets the line is accepted if its buffer overlaps more than a certain threshold with the buffer of the moving line segment.

The rectangles generated usually overlap and some of them are not representative enough for the building modeling. In Figure 3.10. (b), for example, the blue rectangles are not good candidates for building modeling because a part of them is outside the building mask. Thus, a combination of the rectangles needs to be selected in which the rectangles have a minimum overlap with each other, cover the whole building footprint and represent the main parts of the building.

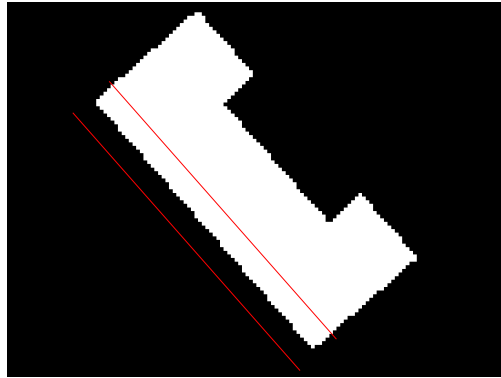


Figure 3.8: Direction of movement based on a footprint-based building mask

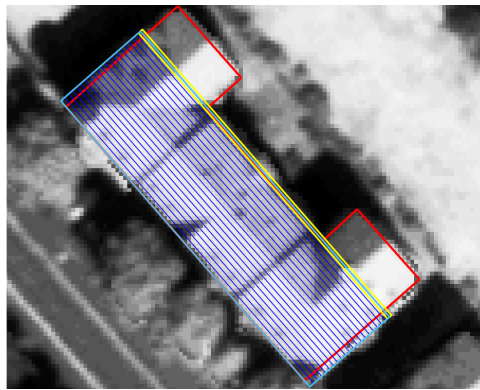


Figure 3.9: Rectangle creation using a moving line segment (red: building outline, blue: moving line segment, yellow: buffer of line segment)

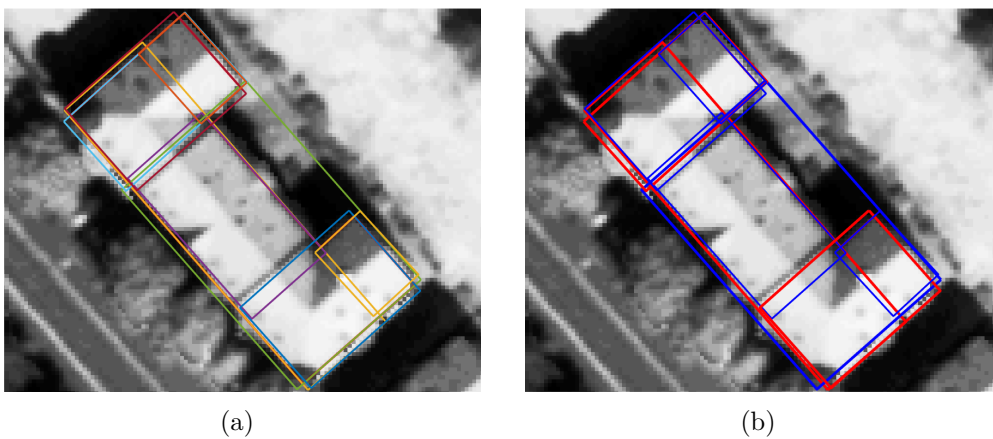


Figure 3.10: Rectangle-based decomposition. (a) Rectangle extraction based on building polygon line segments. (b) Illustration of rectangles (blue: redundant rectangles, red: relevant rectangles)

3.3 A Heuristic Approach for Selecting a Combination of Rectangles

Since a rectangle is generated for each line segment, there are some redundant rectangles which should be removed. These rectangles are either a subset of the relevant rectangles or partially cover the area outside the building mask (blue rectangles in 3.10). Two overlap tests are employed to identify the relevant rectangles. These tests are inspired by the work of Kada and Luo [138], in which the authors decomposed a footprint into several cells based on extending and intersecting the half-space planes. They further computed the overlap between the decomposed cells and the original footprint and compared it with a threshold to see which cells are meaningful and belong to the building. In this dissertation, instead of using the decomposed cells, the relevant rectangles are selected by computing the overlap between rectangles and between every rectangle and the building mask. Thus, the rectangles are sorted based on the length of their related line segments, beginning with the rectangle with the longest line segment. The relevant rectangles are then selected by computing the percentage of the overlap of every rectangle with the building mask:

$$Overlap_1 = \frac{Area(Mask_{rectangle} \cap Mask_{building})}{Area(Rectangle)} > 0.8. \quad (3.2)$$

After selecting the first relevant rectangle based on Equation 3.2, the overlap of the other candidate rectangles with the relevant rectangle is measured. The rectangle with the largest overlap is selected as the relevant rectangle, which is then considered as a reference rectangle to compute its overlap with the other remaining candidate rectangles according to the following equation:

$$Overlap_2 = \frac{Area(R_{search} \cap R_{reference})}{Area(R_{search})} > 0.8, \quad (3.3)$$

where R_{search} is the candidate rectangle and $R_{reference}$ is the relevant rectangle. The value of thresholds is changed based on the complexity of the area and dataset (footprint-based CM or extracted building outline). It turns out, based on empirical investigation, that the value of 0.8 is a good compromise. Figure 3.11 and Figure 3.12. (a) show some results of the rectangle selection process.

After selecting the relevant rectangles, the redundant ones need to be removed (Figure 3.12. (b)). These rectangles are found based on their overlap with the building mask. More precisely, as soon as a set of rectangles cover the whole building mask, the remaining ones are considered as redundant and are removed from the set of the relevant rectangles (Figure 3.12. (c)).

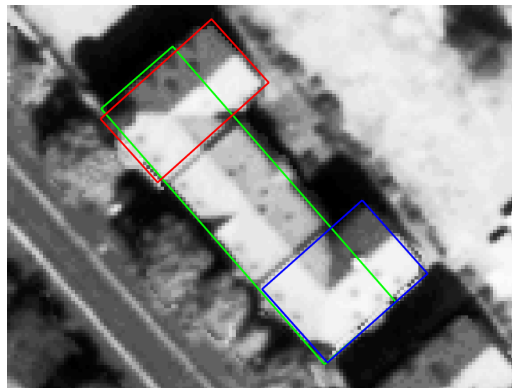


Figure 3.11: Selected rectangles

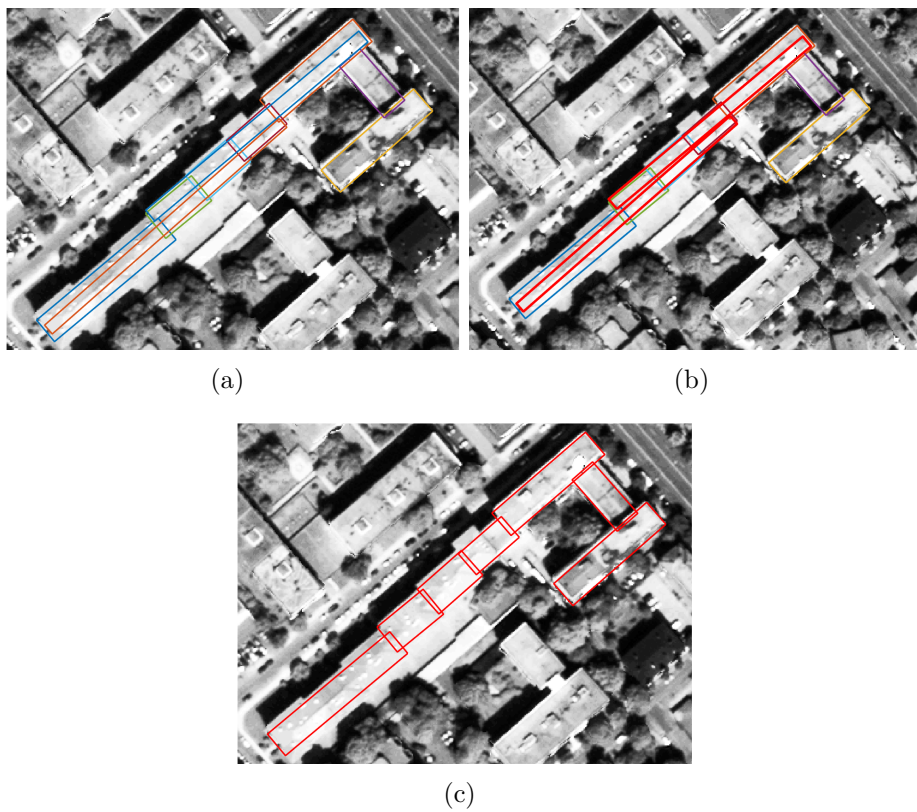


Figure 3.12: Rectangle selection. (a) Selected rectangles after first verification, (b) selected rectangles after second verification (red rectangles), (c) final result of selection after using (a) and (b) verifications

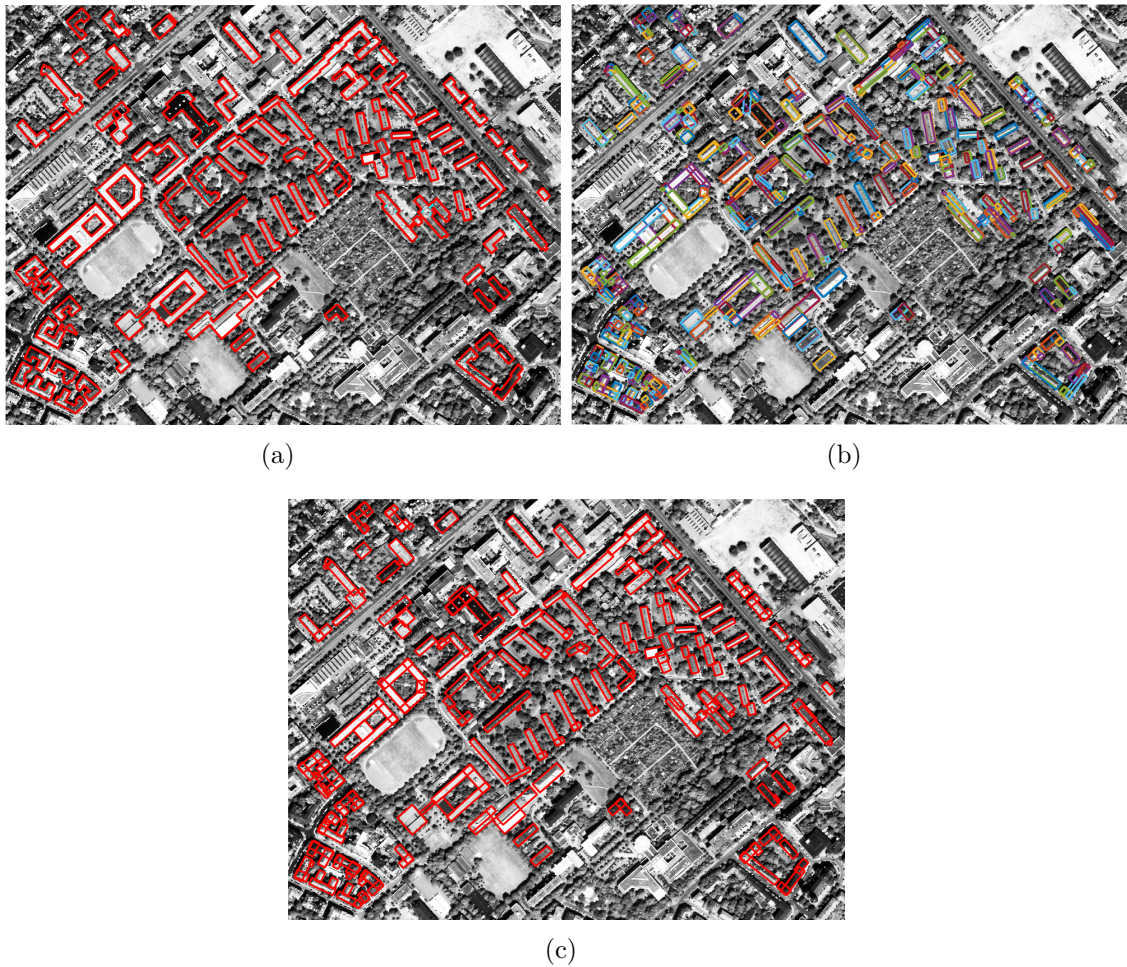


Figure 3.13: Illustration of building footprint generalization, decomposition and selection: (a) the generalized footprint, (b) the decomposed footprint into rectangular shapes (c) and the relevant rectangles selected

3.4 Experimental Results and Discussion

In this section, we evaluate the generalized building footprints and the extracted building outlines obtained by the proposed method in this chapter, both qualitatively and quantitatively. In addition to the extracted building polygon of the ten test areas used in Chapter 2, the building footprints from the CM of a new test area is used for experiments on generalization, decomposition and selection. Figure 3.13 (a)- (c) visualizes the results of the proposed methods for building footprint generalization, decomposition into rectangles and relevant rectangle selection in the new test area, respectively. Results are visualized for the ten areas (Section 2.3) and are presented in Figures 3.14- 3.16.

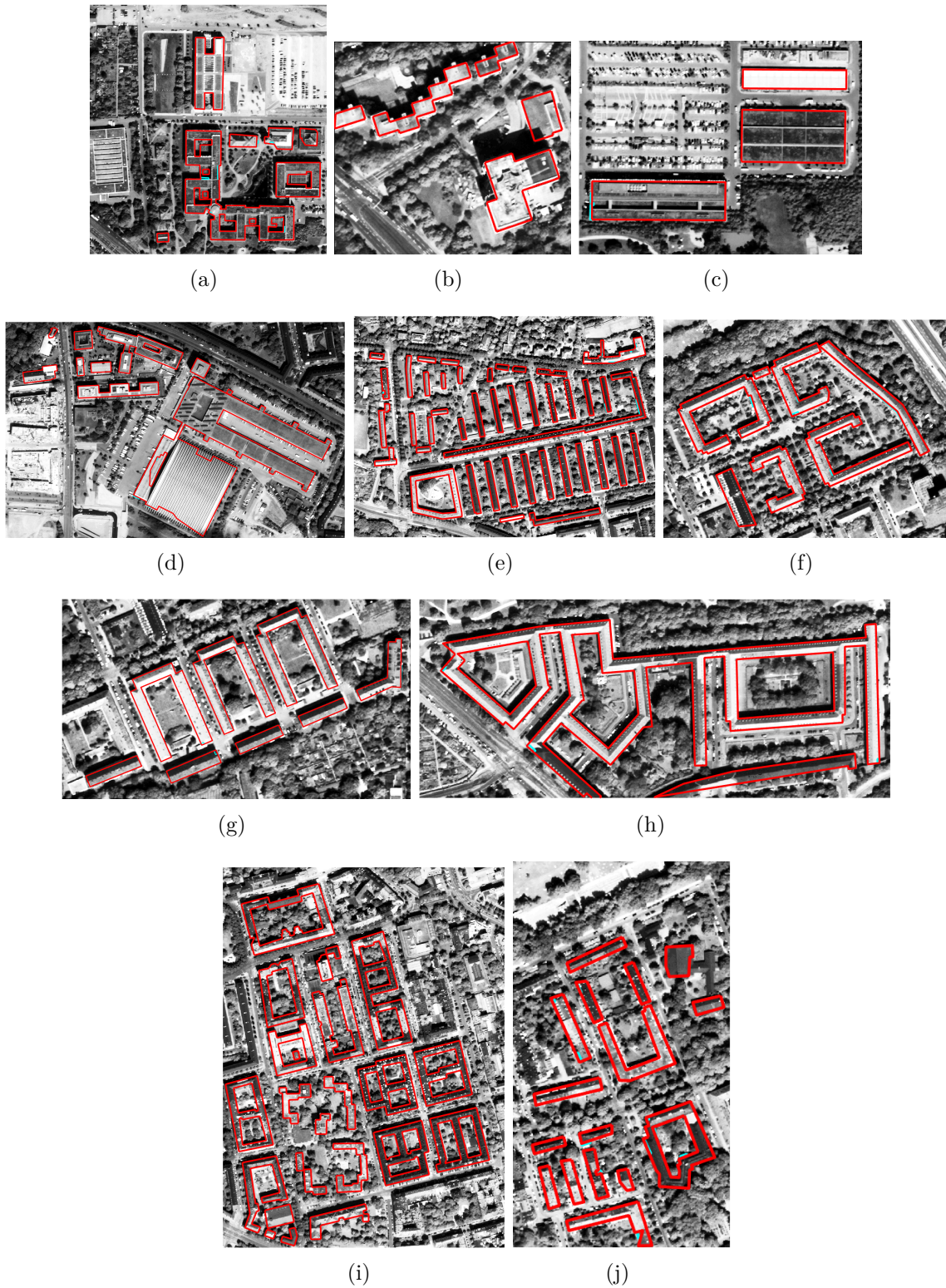


Figure 3.14: Illustration of generalized extracted building outlines, (a) I_1 , (b) I_3 , (c) I_2 , (d) I_{10} , (e) I_8 , (f) I_6 , (g) I_7 , (h) I_4 , (i) I_9 , (j) I_5

3. Building Boundary Generalization and Decomposition



Figure 3.15: Illustration of decomposed building outlines into rectangular shapes, (a) I_1 , (b) I_3 , (c) I_2 , (d) I_{10} , (e) I_8 , (f) I_6 , (g) I_7 , (h) I_4 , (i) I_9 , (j) I_5

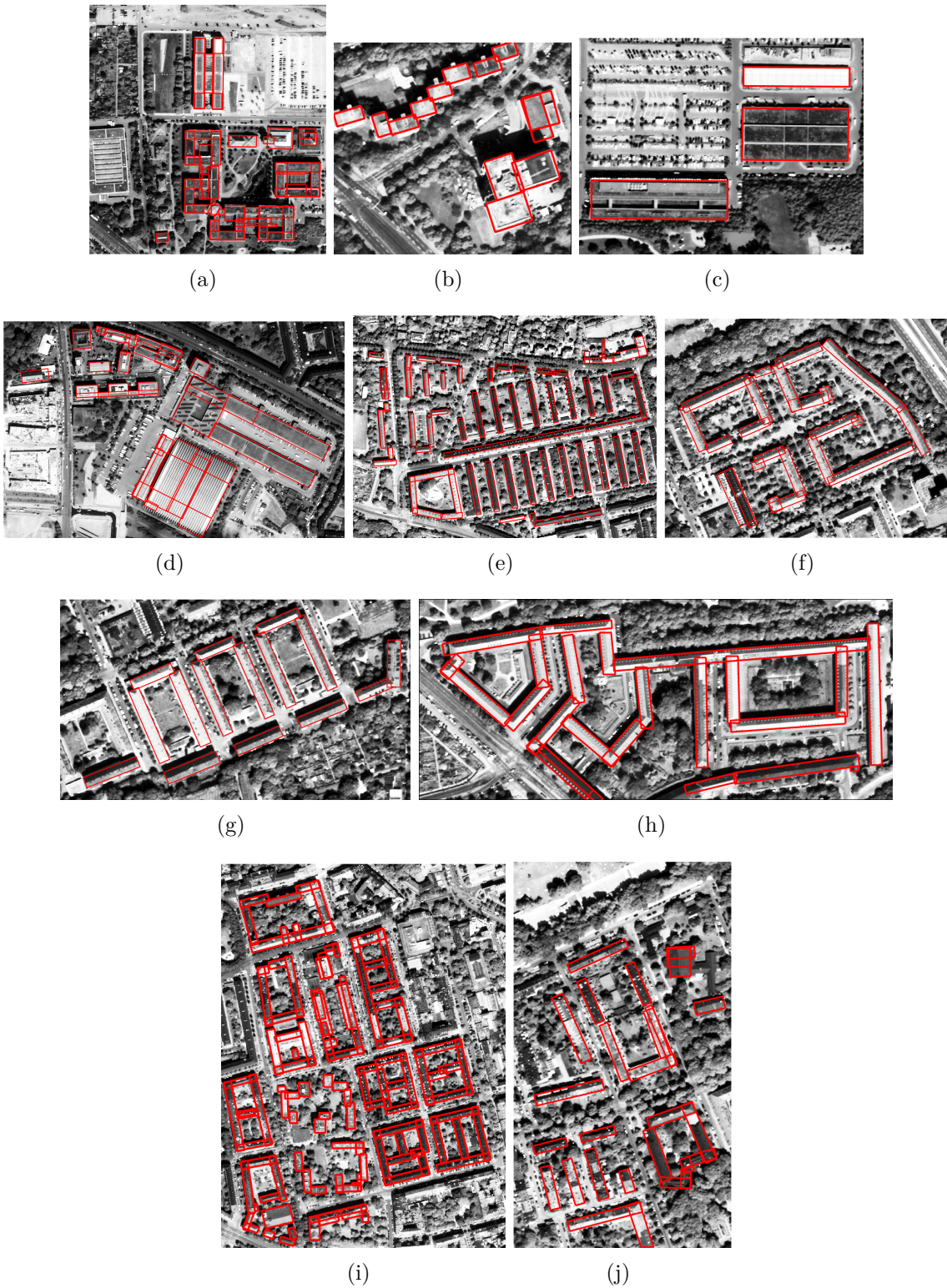


Figure 3.16: Illustration of the relevant rectangles selected, (a) I_1 , (b) I_3 , (c) I_2 , (d) I_{10} , (e) I_8 , (f) I_6 , (g) I_7 , (h) I_4 , (i) I_9 , (j) I_5

3.4.1 Quality Evaluation of the Generalized Building Polygon

In this section, the quality of generalization is evaluated quantitatively. After the generalization process, the shape of the building outline changes and deviates from the original one in reality (ground truth), because generalization can delete or add some vertices to the building outline. Therefore, the generalized building outline is not compared to a ground truth for an evaluation. Instead, it is compared to the building outline before generalization.

The PoLiS metric is a criterion which is used in this section to measure the difference between the original building outline and the generalized one, based on averaging the minimum distances between the vertices of the polygons of the two building outlines (details are explained in Section 2.3.2.3). Thus, PoLiS can detect and consider the omitted outline parts after generalization and can, therefore, be used as a primary control of correctness of the generalization.

In order to investigate the feasibility of the generalization on the proposed extracted building outlines in Chapter 2, they are compared with ground truth before and after generalization using the PoLiS metric (Figure 3.17). As Figure 3.17 shows, the PoLiS metric for areas I_1 , I_5 and I_{10} is higher than the other areas. This is due to incompleteness or absence of some parts of the building mask compared to the ground truth. This difference is even more evident in the area I_1 where the building has several inner yards and height discontinuities and the grass and vegetation cover the building's roofs. In Figure 3.17, green and blue bars depict the PoLiS values before and after the generalization step. The green bar has very small differences compared to the blue bar in each area, as is shown. Due to the simplification of the building outlines in Section 2.2, the proposed generalization step in this chapter can slightly improve the extracted building outlines.

In addition to the PoLiS metric, four different quality parameters are also used and are then combined into a single quality parameter, the so-called total quality (TQ). These quality parameters quantify the differences between the original and generalized objects on the basis of their contours and the areas of their overlapping part [139]. They are defined based on the two goals of the generalization process: Reducing the number of building outline vertices and keeping the input building shape similar to the output one. Therefore, the quality parameters are categorized into two groups; the first group is called the building polygon vertices count (V_N) and the second group includes contour trueness (CT), symmetric difference (SD) and area difference (AD) which focus on different geometric aspects. The outcomes of these quality parameters are limited between 0 and 1, where the larger value indicates the better quality.

- Building outline vertices count (V_N): This quality parameter shows the effect of the generalization on the shape of the building. This criterion is the ratio

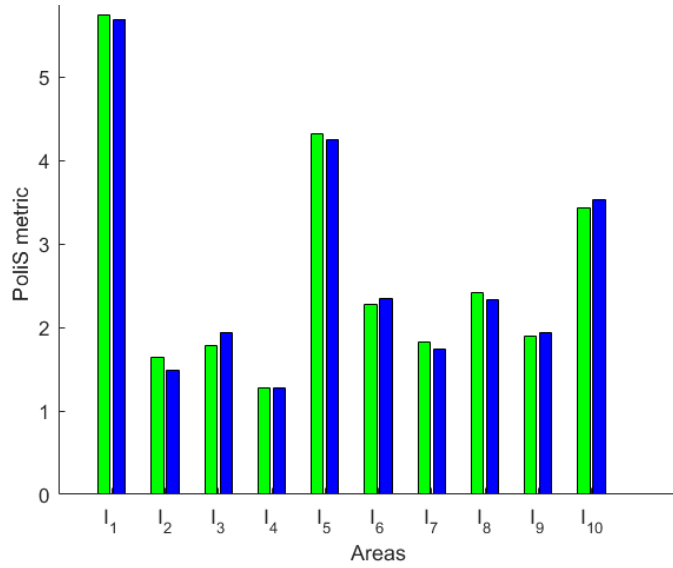


Figure 3.17: Comparison of the PoLiS metric between building outlines and ground truth; green: before generalization, blue: after generalization

between the number of building outline vertices before (N_O) and after (N_G) generalization.

$$V_N = 1 - \frac{N_G}{N_O} \quad (3.4)$$

- **Trueness of building polylines (CT):** This quality parameter compares the overlapping parts of the building outlines before and after generalization to show the shape deviation between the original and generalized building outlines. The geometry of the generalized building outline usually deviates from that of the building polygon before generalization due to its collinearity, parallelism and rectangularity being changed by the generalization process. Thus, a small buffer is created around each line segment of the original building outlines as a tolerance region to compute the CT . The width of buffers should be set smaller than the generalization threshold. The rate of boundary similarity of the original (OR) and generalized (GE) is calculated by the Equation 3.5 which is based on the length of generalized outlines GE surrounded by the buffer of original outlines [139].

$$CT = \frac{Buffer(OR) \cap Perimeter(GE)}{Perimeter(GE)} \quad (3.5)$$

- **Symmetric difference (SD):** This quality parameter is calculated based on the changes in the area of the building outlines after generalization. It considers

3. Building Boundary Generalization and Decomposition

extrusion and intrusion, where they represent the area removed and added after generalization, respectively [139].

$$SD = 1 - \frac{R_{intrusion} + R_{extrusion}}{Area(OR)}, \quad (3.6)$$

where $R_{intrusion}$ and $R_{extrusion}$ are:

$$R_{intrusion} = \frac{Area(OR \cap GE)}{Area(OR)}, \quad (3.7)$$

$$R_{extrusion} = \frac{Area(OR \cap GE)}{Area(GE)}. \quad (3.8)$$

- Area differences (AD): This quality parameter shows the overall area change of the building polygon after generalization as the following,

$$AD = \begin{cases} \frac{Area(GE)}{Area(OR)} & \text{if } Area(GE) \leq Area(OR) \\ 2 - \frac{Area(GE)}{Area(OR)} & \text{if } Area(GE) > Area(OR) \end{cases}$$

The four quality parameters (QP) mentioned previously are then combined to form the (TQ) of the generalized building polygon. The final (TQ) is normalized and represented as a percentage [139].

$$TQ = \sqrt{\sum_{i=1}^n QP_i^2} \quad (3.9)$$

Table 3.1 and Table 3.2 show the PoLiS metric value, total quality and normalized total quality of the footprint¹ and extracted building outlines of the ten areas², respectively. Since the total quality is calculated for each building and there are several buildings in one area, the average of (TQ) are calculated for each area. All the measurements were performed with (+) and without (-) considering (V_N) to evaluate the quality of the generalization and observe its influence by reducing the number of vertices. As both tables show, TQ is increased when V_N is included. This proves that the generalization step simplifies the footprint by decreasing the number of vertices, meanwhile, preserving the building area, which is consistent with the aim of the generalization. In Table 3.1, the normalized total quality is more than 80 %, which means the generalization process does not change the area of the building polygons. On the other hand, considering the V_N criteria can improve the

¹In this chapter, the footprint consists of building polygons of a small area of the CM, which is not considered as ground truth

²The final results of Chapter 2

	Footprint	
TQ	- V_N	1.64
	+ V_N	1.75
Normalized TQ $((TQ-0)/2*100)$	- V_N	82.31
	+ V_N	87.56
PoLiS metric (m)	0.38	

Table 3.1: Quality assessment of the generalized building footprint

quality of generalization more than 5%. The PoliS metric value ($= 0.38m$) between the footprint before and after generalization is less than one pixel, which is acceptable in the generalization threshold (5 pixels) and means the generalization does not discard a large part of the building polygons.

As Table 3.2 shows, all the calculated normalized total qualities of the ten areas are more than 80%, which verifies the changes of the extracted building outlines after generalization. The highest normalized total quality value is related to the area I_2 and the lowest one is related to I_6 , which is due to the complexities of building outlines. As Figure 3.14 (f) shows, there is a curved building outline on the right side of area I_6 . While the generalization process decreases the vertices of the curved building polygon, it may not preserve its area and, subsequently, the normalized total quality value is decreased. Nevertheless, the normalized total quality value is more than 80% in the five-pixel generalization threshold, which is acceptable as the building polygon does not lose its general shape. As Table 3.2 shows, all the PoLiS values of the ten areas are lower than the generalization threshold. The lowest and highest PoLiS values are related to I_4 and I_5 respectively. The reason may be perceived by looking at Figure 3.14 (j), where I_4 has only one large building with inner yards and it was adequately simplified in Chapter 2. Even though the area of I_5 has less complexity in comparison to $I_8 - I_{10}$, its PoLiS metric value is slightly different, which is negligible and below the generalization threshold. This difference may be due to a few missing parts of the building mask and a noisy boundary; the generalization step can improve the building outline slightly by eliminating the outline vertices.

In some areas, the mask refinement did not achieve reasonable results due to misleading classification results or less quality image contrast, which leads to missing some of the building boundaries. Thus, the generalization step could not show its efficiency and its values have been slightly changed compared to the PoLiS metric values.

3. Building Boundary Generalization and Decomposition

		I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
TQ	- V_N	1.71	1.71	1.68	1.72	1.7	1.68	1.72	1.72	1.71	1.72
	+ V_N	1.72	1.75	1.71	1.73	1.72	1.7	1.73	1.74	1.72	1.73
Normalized TQ ((TQ-0)/2*100)	- V_N	85.56	85.74	84.16	86.22	85.03	83.94	86.25	86	85.45	85.87
	+ V_N	86.24	87.47	85.33	86.27	85.96	84.79	86.51	86.8	85.98	86.45
PoLiS metric (m)		0.2611	0.2819	0.3357	0.0539	0.4094	0.3677	0.1435	0.1702	0.223	0.2319

Table 3.2: Quality assessment of extracted building outline for ten areas (I_1 to I_{10})

3.5 Summary and Conclusion

This chapter proposes improved approaches for the simplification of the building outlines. The first step of the simplification is the generalization of the building outlines to reduce their vertices, while preserving the main shape of the building. Since the buildings in urban areas are usually rectangular, the second step in the simplification is the decomposition of each building into the basic rectangular shapes. State-of-the-art methods decompose footprints into several cells with different shapes by extending and intersecting the building edges. The resulting cells are merged based on some criteria, such as the homogeneity of their normal vectors. The previous methods are not efficient for model-driven approaches in which satellite images and DSMs of satellite images are used for modeling, because the normal vector of the plane computed using the DSM is noisy and, therefore, unreliable. To cope with this problem, a decomposition method is proposed in this chapter based on the idea that each line segment of the building outline represents a rectangle. This simplification step can ease the building modeling because each rectangle is a part of the building model. Finally, a heuristic approach is proposed to select a combination of the relevant rectangles among all rectangles created for each building block. Experimental results show that the proposed methods perform well for the generalization and decomposition of building footprints and building outlines in different areas and even for complex building polygons including inner yards and different main orientations.

Roof Type Classification

In the previous chapter, complex building polygons are decomposed into basic rectangular shapes. In this chapter, the roof type of each rectangular shape is recognized in order to reconstruct the 3D model of the buildings. Since the roof model is the most important component of a building, accurate recognition of roof types provides helpful information for the process to reconstruct its LoD2. In this dissertation, roof type recognition is considered as a supervised classification problem, where different roof type categories are introduced in Section 4.1 according to their visibility of geometrical structures in the image and DSM. Additionally, due to the focus of the experiments, only existing roof types in the Munich city area are considered. In Section 4.2, the training and test set is generated based on new semi-automatic methods for roof type classification. A new image-based method is proposed as a classifier based on deep learning algorithms which use the geometrical information of the building roofs in satellite images (Section 4.3). A conventional DSM-based roof type classification is also performed using main curvatures to discriminate flat and non-flat roofs simply to realize the capability of a DSM for recognizing the roof type classes (Section 4.4). We show that this classification can be used as a complementary classification when the image-based classification fails due to low instances of the training data. The results of the classification methods are evaluated and discussed in Section 4.5.

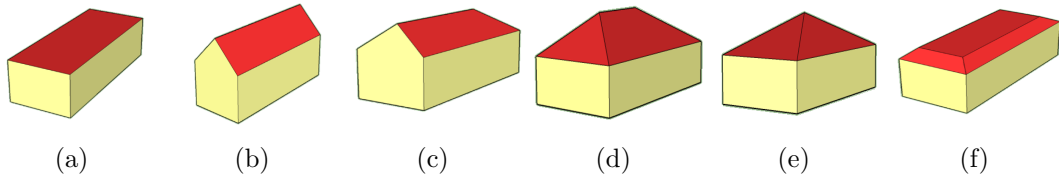


Figure 4.1: Roof model library (a) flat, (b) gable, (c) half-hip, (d) hip, (e), pyramid, and (f) mansard roof models

4.1 Building Roof Library Definition

Defining a roof type library is necessary in model-based 3D building reconstruction. The types of the roofs existing in urban areas depend on various factors, such as climate and social influences. The roofs in industrial areas for example, usually have flat structures, whereas the buildings in downtown areas in Europe usually have multiplane roof styles, such as gambrel roofs. Our experiments are focused on buildings in the city of Munich. Thus, we define our roof model library based on the most common roof types in this area. In addition, the resolution of the DSM is an important criterion when selecting roof types for the library. Since a 3D parametric roof model is reconstructed from the DSM of satellite imagery, the roof types selected should be visible in the satellite images and DSM.

All kind of roof types can be distinguished from the 3D point clouds of LiDAR data or the DSM of aerial imagery [28], whereas distinguishing roof types from the DSM of satellite imagery could be limited. Therefore, it is necessary to consider the potential of the data in representing roof features before creating a library (illustrated in Figure 4.1). The longitudinal and latitudinal 2D profiles of all possible building roof types in Munich (presented in Figures 4.2, 4.3, and 4.4) are analyzed to evaluate the potential of the satellite’s DSM.

The figure illustrates the height changes of buildings in a cross-section of the DSM as orange profiles, the height changes of building in a longitudinal section of the DSM as red profiles, and the height changes of the building which best fit to the DSM profile and reality as blue profiles. The satellite’s DSM allows us to discriminate building boundaries from rooftops, as can be seen in all the profiles. According to the longitudinal and latitudinal profiles, flat roof can be distinguished simply from a non-flat roof, which is added to the library. The simplest non-flat roof is a shed roof with only one sloped plane. The latitudinal profile of this roof type is different from that of flat roofs, however, this difference is not visible in the images, therefore, the shed roof type is not considered as an individual model in the library. Instead, it is taken as a flat roof with its sides of different heights. The gable, half-hip and hip roof profiles differ in the degrees of height changes at the end of the ridge line in the longitudinal profile while the heights change very similarly in their latitudinal profiles. These roof models are also added to the

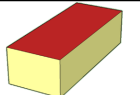

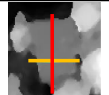

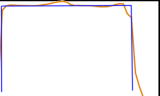
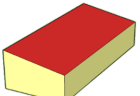

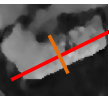
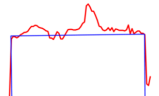
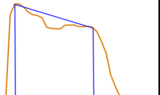
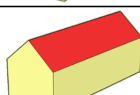

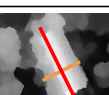
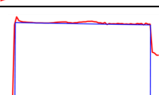
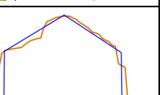
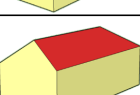

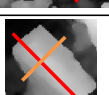
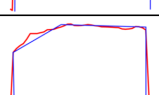
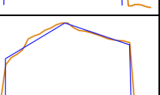
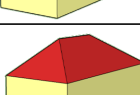

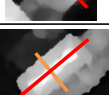
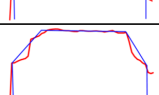
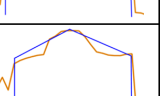
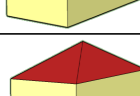

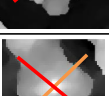
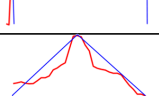
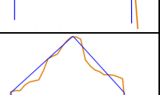
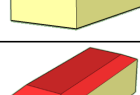
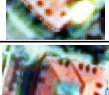
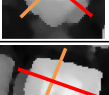
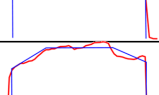
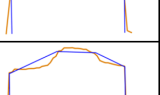
	Roof models	Pan-sharpened image	DSM	Longitudinal profile	Latitudinal profile
Flat roof					
Shed roof					
Gable roof					
Half-hip roof					
Hip roof					
Pyramid roof					
Mansard roof					

Figure 4.2: Illustration of different roof models with their PS image patch, DSM, latitudinal profile and longitudinal profile (Basic roof types)

4. Roof Type Classification

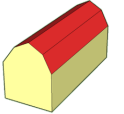


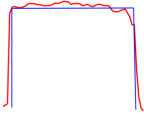
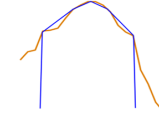
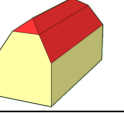

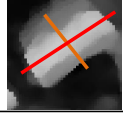
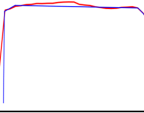
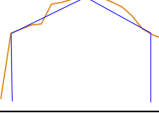
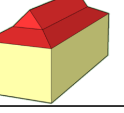
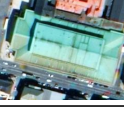
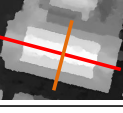

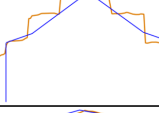
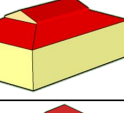

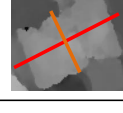
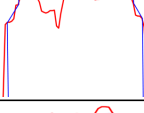

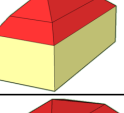


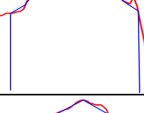

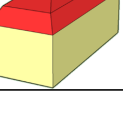


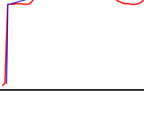
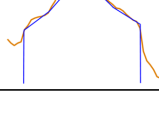
	Roof models	Pan-sharpened image	DSM	Longitudinal profile	Latitudinal profile
Gambrel roof					
Jerkinhead roof					
Bonnet roof					
Dutch gable roof					
Mansard-hip roof					
Mansard-pyramid roof					

Figure 4.3: Illustration of different roof models with their PS image patch, DSM, latitudinal profile and longitudinal profile (Combinatorial roof types)






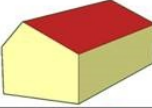




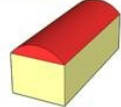

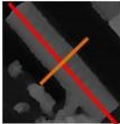
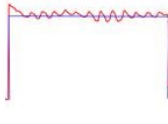

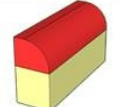
	Roofmodels	Pan-sharpened image	DSM	Longitudinal profile	Latitudinal profile
Saltbox roof					
Dissymmetric roof					
Elliptic roof					
					

Figure 4.4: Illustration of different roof models with their PS image patch, DSM, longitudinal and latitudinal profiles profile (Less frequent roof types)

library. Pyramid and mansard roof models are different from the other roof types, and their differences are visible in the image, and in the latitudinal and longitudinal profiles of the DSM; therefore, they are also added to the library.

There are other roof types in Munich whose differences are visible in the images, as Figures 4.2, 4.3 and 4.4 show. However, these roof types are created by either adding some planes to the basic roof types or combining two basic roof types which have already been added to the library. Gambrel roofs, for example, are created by adding two sloped planes onto the gable roofs, and jerkinhead roofs are a hip variant of gambrel roofs. These added planes cannot make any difference in the longitudinal and latitudinal profiles of a building in the DSM. The bonnet roof and Dutch gable roof are combinations of the gable and mansard roofs. The Dutch roof's longitudinal and latitudinal profiles are very similar to those of hip roofs. The mansard-hip and mansard-gable roofs are categorized as hip and pyramid roof models, respectively, which have similar longitudinal and latitudinal profiles. Since the frequency of the roof types in Munich is important in generating the library, other roof types which are less frequent, such as bonnet and elliptic roofs, have not been added to the library. They are categorized to the basic roof types with similar profiles. The saltbox and dissymmetric gable roofs, which have a similar longitudinal and latitudinal profile to the profiles of gable roofs, are categorized as a gable roof model. Finally, six types of roofs: Flat, gable, half-hip, hip, pyramid and mansard roofs, generate the roof model library.

4. Roof Type Classification

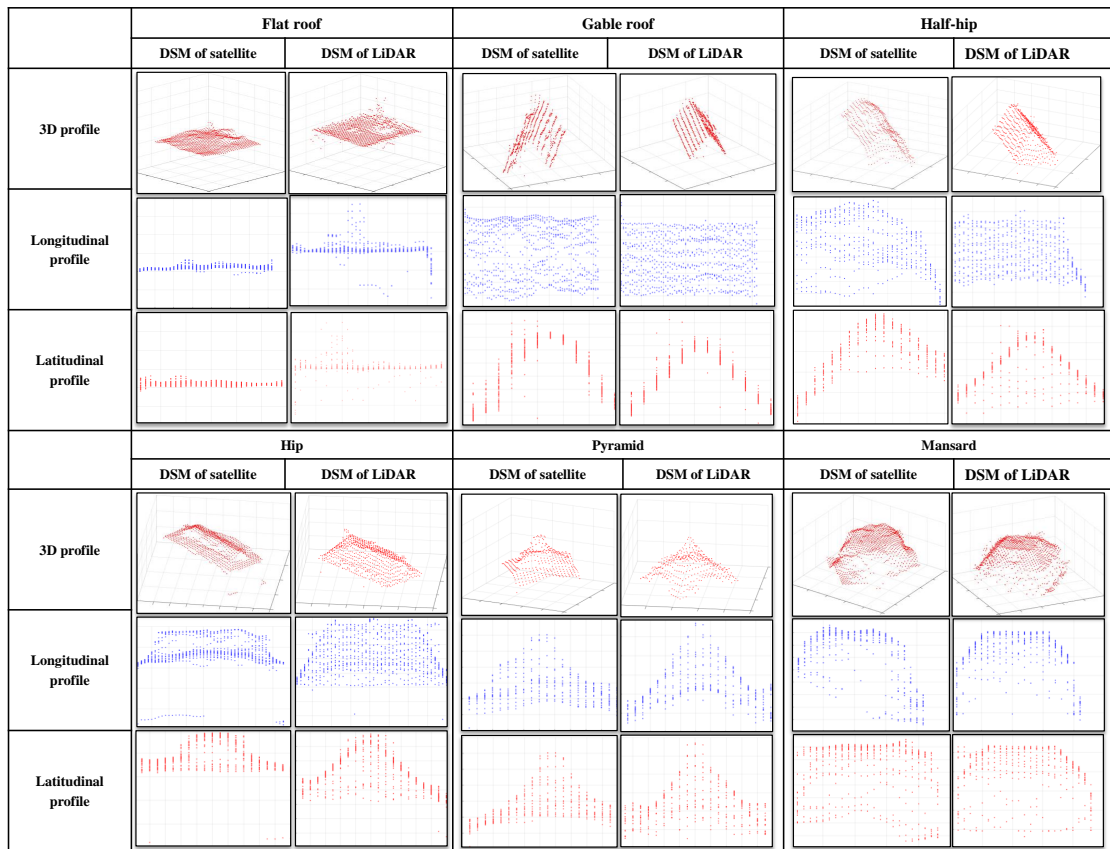


Figure 4.5: Illustration of 3D profile, 2D longitudinal profile and 2D latitudinal profile of the satellite's DSM and LiDAR's DSM data for the various roof types

The longitudinal and latitudinal profiles of the different roof types are illustrated in Figure 4.5 to compare the quality of the satellite's DSM and the DSM of LiDAR data. In this figure, data was sampled every two pixels for projection. As Figure 4.5 shows, the regularity of the roof types in both datasets is comparable, as roof types in LiDAR data are more distinguishable than the ones in DSM, therefore, a model-driven method for the reconstruction of roof model is more reliable than the data-driven method on the satellite's DSM. This profile-based verification is performed for all types of roofs defined in the roof model library created, including the six roof types: Flat, gable, half-hip, hip, pyramid and mansard roof models.

4.2 Dataset Generation Based on Roof Model Library

A training and test image patch dataset was generated to classify roof types and validate their results. The datasets are composed of six classes representing the six roof types in the library, as shown in Section 4.1. Each patch contains the whole or part of only one roof. Additionally, the roofs are masked using building masks extracted from the footprint of CM to reduce the impact of their surrounding objects (e.g. trees and asphalt). Since CM-based footprints are available and the refined building masks from Chapter 2 are very close to them, they are used for generating both training and test patches.

The geometrical structures of the roof type are important points for roof type classification; therefore, one roof type should lie completely on each patch. In this dissertation, a new semi-automatic method based on building mask skeleton is proposed for generating roof image patch datasets. Figure 4.6 shows the steps of generating the training and test patches using a building mask skeleton. Firstly, the skeleton of the building mask which is extracted from cadastral building footprints is computed by a morphological operator to extract the patches. The junction points of the skeleton are then projected on PS satellite images. Three channels of PS images are used to generate the patches. A square box with a fixed size crops the image around each junction point of the skeleton. The size of the square box is selected so that contains only the buildings selected. Based on the experimental investigations, a fixed size of $56 \times 56 \times 3$ pixels is considered proper. The library of the roof patches is shown in Figure 4.7 based on the roof model library in Figure 4.1. After generating the patches, each is manually labeled according to its roof type. The main difference between this approach and previous image patch dataset generation approaches [143, 144] is that the main orientation of each roof is also considered inside the patch. Therefore, the quality of roof patches cannot be degraded by rotation and resizing. Satellite data exhibit great challenges even for visual analysis tasks. Firstly, the quality of some satellite images is degraded due to blurring which occurs during the capturing process. Secondly, roofs are covered by various kinds of facilities, such as air conditioners, chimneys and other roof build-ups. In addition, some roofs in satellite imagery are occluded by shadow and trees. These imperfections are significant obstacles for a reliable visual and computerized analysis task. Furthermore, the number of instances of some roof classes is extremely low in comparison to other classes (such as mansard and pyramid roofs). Augmentation methods are used to increase the training patches of these roof types to deal with the small numbers of samples for two roofs, pyramid and mansard roofs, and in order to balance the number of samples for all roof type categories.

Converting to HSV color space, flipping the image to the right side and rotation of 45° are among several different augmentation methods in previous deep learning-

4. Roof Type Classification

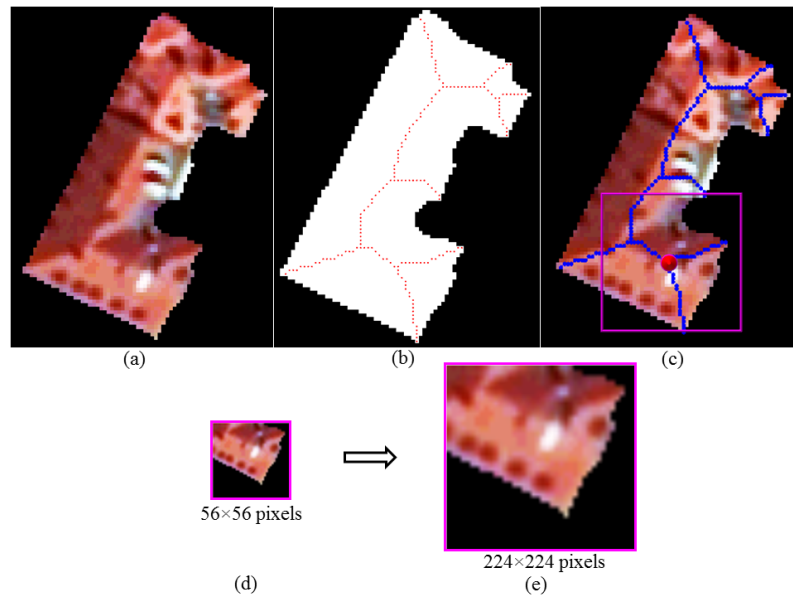


Figure 4.6: (a) PS image of a building roof, (b) building mask and overlapping skeleton points, (c) projected skeleton points on the PS image of a building roof, (d) extracting the patches for one skeleton point in a size of 56×56 pixels, (e) resize roof patch to the proper size of pretrained convolutional neural networks (CNNs) (224×224 pixels)

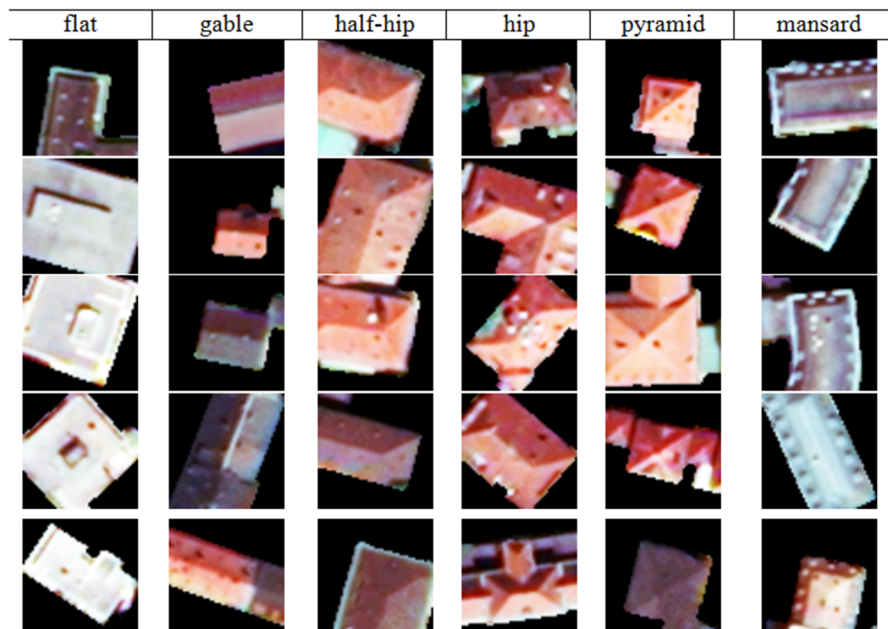


Figure 4.7: The library of roof patches including six roof types

Roof type	Data Augmentation	Training #	Test #
Flat	no	980	1158
Gable	no	1122	1760
Half hip	no	790	970
Hip	no	979	183
Pyramid	yes/no	1149/207	32
Mansard	yes/no	1112/40	27

Table 4.1: The distribution of the training and test sets used in the experiment

based researches used in this dissertation. These augmentation methods can also increase the robustness of the results against rotation and color changes. The distribution of the training and test patches before and after augmentation are shown in Table 4.1.

4.3 Image-based Roof Type Classification

According to Section 4.1, roof type classification depends on the resolution and quality of the DSM. The flat roofs can be discriminated simply from non-flat ones, as Figures 4.2, 4.3, 4.4 and 4.5 show, however, when reconstructing LoD2, categorizing all building roofs into two types is not enough and close to reality. Therefore, the library defined in the Section 4.1 is used for image-based classification. The geometry features play an important role in the image-based roof type classification. Most recent methods for object recognition and classification of the remote sensing images normally use engineered feature descriptors (e.g. HOG and SIFT). However, it has been shown in a number of recent works that deep learning has performed more effectively in satellite image classification and object detection [18, 145]. The convolutional neural networks (CNNs) can automatically learn structured and representative features through layer-to-layer hierarchical propagation schemes as high-level features learn from lower-level ones. The resulting features are invariant to rotation, occlusion, scale and translation. These invariant features are beneficial for the wide variety of object detection and classification tasks, which reduces the need for designing complicated engineered features [146, 147]. Inspired by these previous works, in this dissertation, we evaluate two common strategies based on pretrained CNN models for classifying the roofs into the six roof types of our defined library.

A CNN is a feedforward neural network which is composed of convolution (conv), nonlinearity (ReLU), pooling (subsampling), fully connected (FC) and classification layers. The conv layers are the main parts of the CNN architecture. The layers

consist of a set of learnable filters which are convolved across the width and height of input images with a predefined stride and produce 2D feature maps. A conv layer is formed by stacking the feature maps of all filters along the depth of the input volume. Since the real-world data is usually nonlinear and conv layers are linear operators (i.e. element-wise matrix multiplication and addition), it is required to introduce nonlinearity to the model, which is performed by the ReLu function. The Relu function keeps non-negative values in the feature maps and replaces the negative value by zero. Another important part of CNN is pooling, which can be seen as a nonlinear downsampling. Among several functions for implementing pooling (e.g. average pooling, L_2 -Norm pooling, max-pooling), max-pooling is the most commonly used one. The pooling layer reduces the spatial size of the representation while keeping its most important information. Subsequently, it reduces the number of parameters in the network to control overfitting and reducing the computation time. The conv and max-pooling layers are followed by FC layers. Neurons in an FC layer are fully connected to all feature maps in the previous layer. The last FC layer holds the output as the class scores. The loss layer is the last layer of a CNN architecture which penalizes the deviation between the predicted and true labels during the network training phase [148, 149]. Loss functions are usually selected based on the task specified. Cross-entropy loss, for example, is the most widely used loss function in object recognition tasks.

In detail, CNN is trained by minimizing a negative log likelihood regarding the weights and biases over the whole training set using a stochastic gradient descent (SGD). Mathematically, the following cost function is minimized:

$$\min_{\Theta} \mathcal{L}(\text{softmax}(\Theta^l \text{ac}(Patch_i(I)_{m \times m}), \Theta^{l-1}), y_i), \quad (4.1)$$

where Θ^l are the parameters of the last layer, Θ^{l-1} are the parameters of the previous layer, $\text{ac}(\cdot)$ is an activation function and y_i represents the correct label of the patches specified ($Patch_i$) [54]. In the classification layer, softmax is the most commonly used method to assign output probability to each object class in order to predict a single class out of several possible classes. The softmax function is defined as the following:

$$\text{softmax}(\mathbf{z}) = \frac{\exp(\mathbf{z})}{\|\mathbf{z}\|_1}, \quad (4.2)$$

and the loss function is computed by

$$\mathcal{L}(\Theta = (\mathbf{W}, \mathbf{b}), D) = - \sum_{i=0}^{|D|} \log P(Y = y^{(i)} | x^{(i)}, \mathbf{W}, \mathbf{b}), \quad (4.3)$$

where D is the data set, \mathbf{W} and \mathbf{b} are the weights and biases, respectively, parameterizing the network. In this equation, $\log P(Y = y^{(i)} | x^{(i)}, \mathbf{W}, \mathbf{b})$ denotes the

probability of the i^{th} image to be classified correctly as label y . For each layer of the network, the weights are updated at each iteration t as follows:

$$\mathbf{V}_{t+1} = \nu \mathbf{V}_t - \eta_t \tau \nabla \mathcal{L}(\mathbf{W}_t), \mathbf{W}_{t+1} = \mathbf{W}_t + \mathbf{V}_{t+1} \quad (4.4)$$

where \mathbf{V}_{t+1} is the velocity which is updated by a linear combination of the negative gradient $\nabla \mathcal{L}(\mathbf{W}_t)$ and the velocity from the previous iteration \mathbf{V}_t . \mathbf{W}_{t+1} is the updated weights according to the updated velocity and previous weight \mathbf{W}_t . The learning rate τ weights the negative gradient and η is the scheduling rate which reduces the learning rate at the end of specific iterations. The momentum ν is the coefficient of velocity from the previous iteration [150].

The training process starts with the initialized weights for each conv layer, which are sampled from a normal distribution with a zero mean and small standard deviation. Due to the large number of parameters in each layer (often in the range of millions), the training process requires a large amount of labeled data to update the parameters iteratively; otherwise, it gets stuck in an undesirable local minimum resulting in an over- or underfitting. Therefore, when the training dataset is not large enough, an existing pretrained network on large datasets, such as ImageNet (1.2 M labeled images), is fine-tuned by continuous training (i.e. by running back-propagation) on the new small available dataset. Fine-tuning starts by transferring the weights from the pretrained network to the new network. This process called “transfer learning.” This is motivated by the observation that the earlier layers of different CNN models learn almost the same basic features, such as edge, color or gradient, for different visual recognition tasks, while the upper layers have more specific features related to the classes and properties of the original datasets. Therefore, the upper layers can be fine-tuned and matched to a different but correlated problem. It has been shown in previous works that the pretrained models for CNN networks, such as Alexnet [151], VGGNet [152], GoogleNet [153] and ResNet [154], that have been trained on large datasets, such as ImageNet, are very applicable for other visual recognition tasks without any requirement for training their first few layers.

A common practice for fine-tuning is to replace the last FC layer of the pretrained CNN with a new FC layer that has as many neurons as the number of classes desired in the new task. The fine-tuning of the new network starts with weight initialization of the last FC layer. All the other layers are then fine-tuned. The important hyper-parameter which is changed for fine-tuning pretrained CNN is the “learning rate.” Since the pretrained weights are expected to be much better than the randomly initialized weights, the learning rate is usually chosen ten times smaller than the one used for training the network from scratch to avoid distorting them a lot and quickly.

Fine-tuning is very useful for classification tasks in remote sensing, where the acquisition of large sets of training data needs a huge effort and cost [145]. Deep

CNNs, such as VGGNet, have shown effective performance in the classification of remote sensing images [155, 156] and using deeper networks, such as ResNet, can increase the accuracy of the classification even more [157, 158]. Therefore, we used the pretrained models of these networks in our experiment. We add a new FC layer with six neurons to the two pretrained models for the classification of roofs into six categories. The networks are then fine-tuned in our dataset.

4.3.1 Pretrained VGGNet

The VGG network architecture was introduced by Simonyan and Zisserman in 2014 [152] and achieved a very good performance on the ImageNet ILSVRC-2014 submission for localization and classification. The VGG shows that the depth of the network plays an important role and gives better results. Among many pretrained models, VGGNet adopts the simplest kernel and pooling windows. Only 3×3 convolutions (the convolution stride is set to 1) are stacked on top of each other in increasing depth. Reducing volume size is handled by 2×2 max pooling throughout the whole network. This stack of conv layers is followed by three FC layers: The first two have 4096 nodes for each and the third has 1000 nodes (one for each class of ImageNet dataset).

The VGG 16 (with 13 conv and 3 FC layers) and VGG 19 (with 16 conv and 3 FC layers) are two very deep versions of VGGNet with 16 and 19 weight layers, respectively [152]. The width of the conv layers (the number of nodes) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer until it reaches 512 [152]. The network structure of VGG 16 and VGG 19 are shown in Figure 4.8.

4.3.2 Pretrained ResNet

Deep networks increase the classification performance due to having enriched features which are obtained by the number of stacked layers (depths). He *et al.* [154] showed that stacking deeper layers in plain networks, such as VGG 16/19, leads to a degrading problem resulting in a deduction in their classification performance. This degradation is an optimization problem as deeper models are harder to optimize. He *et al.* [154] proposed a solution based on a deep residual learning framework. It uses the stacked network layers to fit residual mapping $F(\beta) = H(\beta) - \beta$ instead of the mapping $H(\beta)$ directly, where β is an identity function. If the identity mapping is optimal, the solvers may drive the weights of the multiple nonlinear layers toward zero to approach identity mapping. Figure 4.9 shows the difference between the plain net and the residual net with two stacked conv layers.

Using the residual learning formulation, extremely deep nets are easy to optimize and can benefit the accuracy obtained from their depth.

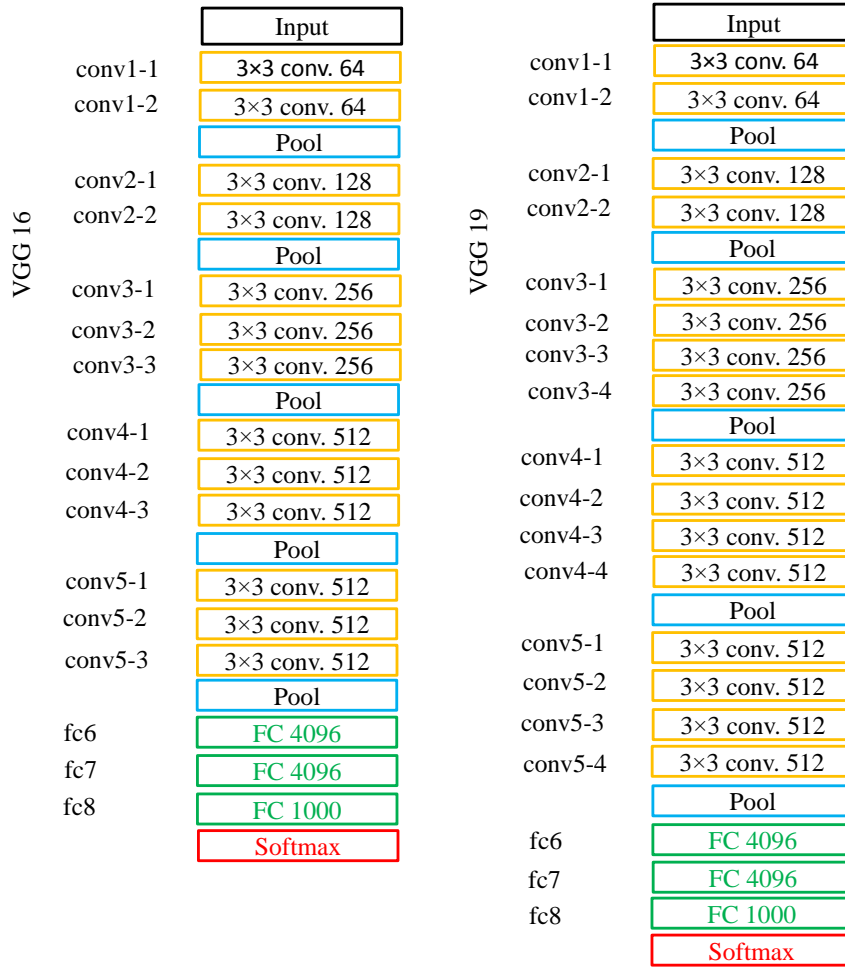


Figure 4.8: The network structure of the VGG 16 and VGG 19

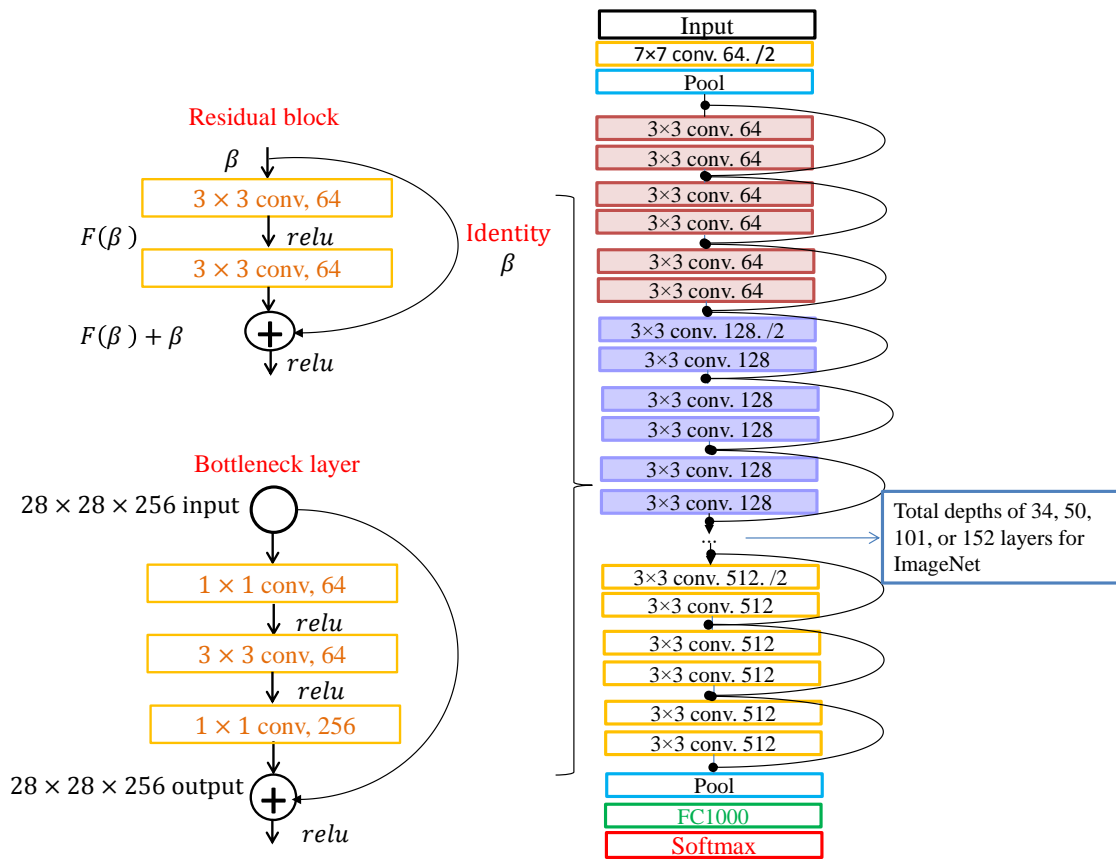


Figure 4.10: Residual block (up and left), bottleneck layer (bottom and left), ResNet architectures (right)

4. Roof Type Classification

hyper-parameters	VGGNet	ResNet
Mini-batch size	10	8
Test iteration (number of validation /mini-batch size)	130	155
Test Interval	1000	1000
Learning rate (τ)	0.001	0.001
Maximum iteration	100000	100000
Scheduling rate (η)	0.1	0.1
Momentum (ν)	0.9	0.9
Weight decay	0.0005	0.0001

Table 4.2: The model hyper-parameters of VGGNet-16 and 19 and of ResNet 50/101/152

16/19 and ResNets. This layer has six classes corresponding to six roof types. The first few layers represent low-level features, such as curves and edges that are important for roof type classification. Therefore, their weights are kept intact for our experiments. The subsequent layers, however, represent dataset-specific features. Thus, their pretrained weights are updated through a back propagation to become fine-tuned for our dataset.

As mentioned above, the same structures of the networks for training from scratch are used in fine-tuning, except the last layer. For our experiments, only the hyper-parameters of the models, such as learning rate, mini-batch size and test iteration, are adjusted; these can be seen in Table 4.2.

4.3.4 Pretrained CNN-based Features

In addition to the fine-tuning approach, a pretrained CNN can be treated as a fixed feature extractor. In this approach, the classification layer of the network is removed, and the rest of the CNN is treated as a feature extractor for the new dataset. A classifier, such as SVM or softmax, is then applied to the extracted features to classify images. These features are known as deep convolutional activation features (DeCAFs) [159].

The last FC layers of the pretrained VGG 16/19 and ResNet are removed for roof type classification in this dissertation. The rest of the networks are then used for extracting features from the training and test image patches. The features extracted are obtained from FC6 and FC7 of VGG 16/19 and from FC1000 of ResNet (50/101/152), and have 4096 and 1000 dimensions, respectively. An SVM classifier using a radial basis function (RBF) kernel is then applied to these features to classify the roof types.

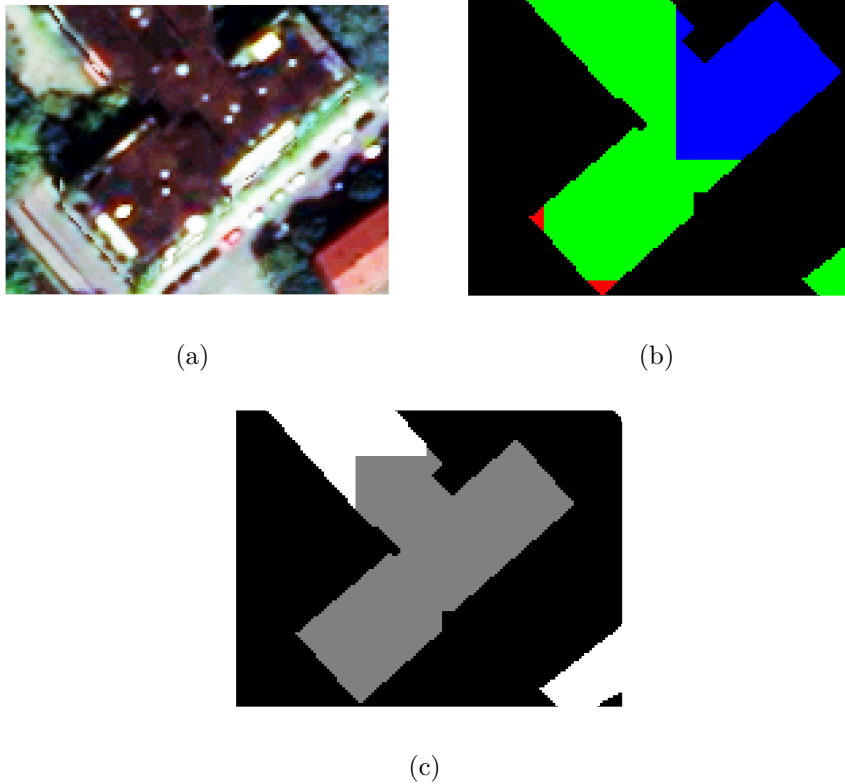


Figure 4.11: Example of a drawback of image-based classification, (a) PS image, (b) image-based classification, (red: flat roof, green: gable roof, and blue: half-hip roof), c) DSM-based roof type classification, (gray: flat roof, white: non-flat roof)

4.4 A Conventional DSM-based Roof Type Classification

Although using pretrained CNNs is a powerful method for roof type classification, it requires a large dataset with a high variety of patches in texture, color, rotation and roof structure for fine-tuning. Consequently, there are some misclassifications of the roof types which lack enough training samples. In Figure 4.11, the flat roof type is classified as a sloped roof (green represents a gable roof and blue represents a half-hip roof), while a flat roof is classified correctly in a conventional DSM-based classification method. Since this kind of misclassification is not large, the result of DSM-based classification can be used as complementary information for classification of roof types when the image-based classification fails. According to Figures 4.3- 4.5, the flat roof structure can be restored simply from DSM. Therefore, a preliminary classification of the DSM can discriminate the flat and non-flat (sloped) roofs. In literature, the classification of roof types from DSM is usually performed by extract-

ing important features, such as gradient [160], surface normals [94], and Gaussian and mean curvatures [161]. A conventional method for DSM-based roof type classification is using the features of the surface, such as mean curvature and Gaussian curvature. This method was used, firstly, for surface-based segmentation of range and intensity images [162]. It was subsequently used for extracting linear features, such as ridge lines and eave lines, of the DSM of LiDAR data [163]. Reference [161] used the features and the classification way proposed in the research mentioned previously to classify roof types into the flat and non-flat roofs from the DSM of LiDAR data. In this dissertation, mean curvature and Gaussian curvature are calculated by the method employed in [162, 164].

Curvature is a useful feature in surface classification. It represents how a tangent vector changes along a curve on the surface. The maximum and minimum curvatures, the so-called principal curvatures, allow us to compute various features that provide some information about the local shape of the surface. The most important curvature-based features are mean curvature (\mathcal{H}) and Gaussian curvature (\mathcal{K}), which represent the shape of the curve and surface, respectively,

$$\mathcal{H} = \frac{k_1 + k_2}{2}, \quad \mathcal{K} = k_1 k_2, \quad (4.5)$$

where k_1 and k_2 are minimum and maximum curvatures, respectively. These two features are scalar and invariant to rotation, translation and the changes in parameterization. There are eight basic types of surface that can be specified using the sign of the mean and Gaussian curvatures, as shown in Table 4.3. Among different ways to calculate mean and Gaussian curvatures, in this dissertation, a quadratic surface z is fitted to the points in each local region of the surface and the curvatures of the fitted surface are computed analytically [164].

$$z = q(x, y) = a_{11}x^2 + a_{22}y^2 + a_{12}xy + a_{10}x + a_{01}y + a_{00} \quad (4.6)$$

The coefficients a_{11} , a_{22} , and a_{12} of the quadratic surface are estimated by a LS-based surface fitting. Let p be a point on the smooth surface S and n_p be the unit normal to S at p . Suppose $q(x, y)$ is a local parameterization of S in a neighborhood of p . Then using $q_x(p)$, $q_y(p)$, and n_p as a local coordinate system, the principal curvature and subsequently mean and Gaussian curvatures are calculated based on a Hessian matrix, which is made with entries of second derivatives of Equation 4.6.

$$\begin{pmatrix} q_{xx}(p) & q_{xy}(p) \\ q_{yx}(p) & q_{yy}(p) \end{pmatrix} = \begin{pmatrix} 2a_{11} & a_{12} \\ a_{12} & 2a_{22} \end{pmatrix} \quad (4.7)$$

4.4. A Conventional DSM-based Roof Type Classification

	$\mathcal{K} > 0$	$\mathcal{K} = 0$	$\mathcal{K} < 0$
$\mathcal{H} < 0$	Peak $\mathcal{T} = 1$	Ridge $\mathcal{T} = 2$	Saddle Ridge $\mathcal{T} = 3$
$\mathcal{H} = 0$	(none) $\mathcal{T} = 4$	Flat $\mathcal{T} = 5$	Minimal Surface $\mathcal{T} = 6$
$\mathcal{H} > 0$	Pit $\mathcal{T} = 7$	Valley $\mathcal{T} = 8$	Saddle Valley $\mathcal{T} = 9$

Table 4.3: Surface type determined based on the surface curvature sign

The eigenvalues of the Hessian matrix of the quadratic surface, λ_1 and λ_2 ($\lambda_1 < \lambda_2$), are the minimum and maximum principal curvatures, respectively. Thus, mean and Gaussian curvatures are calculated using Equation 4.5. Based on the sign of mean and Gaussian curvatures, the type of surface in each DSM point is determined using signum function of Equations 4.8 and Equation 4.9 [162].

$$sgn_{\epsilon}(x) = \begin{cases} +1 & x > \epsilon \\ 0 & |x| \leq 0 \\ -1 & x < -\epsilon \end{cases}, \quad (4.8)$$

$$\mathcal{T}(i, j) = 1 + 3(1 + sgn_{\epsilon_{\mathcal{H}}}(\mathcal{H}(i, j))) + (1 - sgn_{\epsilon_{\mathcal{K}}}(\mathcal{K}(i, j))), \quad (4.9)$$

where $\epsilon_{\mathcal{H}}$ and $\epsilon_{\mathcal{K}}$ are thresholds which are selected as 0.005 and 0.003, respectively, for 5×5 pixel windows in our experiments.

4.5 Experimental Results and Discussion

The classification results of the CNN-based approaches are presented and discussed in this section. In addition, they are compared to two conventional image-based methods, SVM and random forests (RF), using engineered feature vectors, and a conventional DSM-based classification method.

4.5.1 Experiment Setup

The datasets for image-based classification are generated from PS images of WorldView-2 satellite to take advantage of the spatial resolution of PAN images together with the spectral resolution of MS images. The PS images are generated by fusion of the PAN images, with a spatial resolution of 50 cm, and the orthorectified MS images, with a spatial resolution of 2 m and eight channels. Thus, the resulting PS image has a spatial resolution of 50 cm and a spectral resolution of eight channels. Since the pretrained CNN models perform with three-channel images, the red, green and blue channels (the 5th, 3th, and 2nd channels) of the PS image are selected to form RGB images. A number of datasets were generated from the DSM for DSM-based classification. Additionally, a building mask is obtained from a building CM-based footprint and is projected on images and DSM to separate buildings from other objects. The training and test datasets were selected from different areas of Munich and, therefore, they are totally independent from each other.

Fixed-size patches of $224 \times 224 \times 3$ pixels are required for fine-tuning and extracting features of the pretrained VGGNet 16/19 layers and ResNet model architectures. Since the size of the training and test image patches are $56 \times 56 \times 3$ pixels, they are resized to the required input size of the networks, as Figures 4.6 (d) and (e) show. The CNN models were pretrained on the ImageNet ILSVRC benchmark [165] dataset with 1.2 million images with 1000 different categories. Therefore, training and test image patches for roof type classification should have the same properties as the ImageNet dataset. Thus, all of the training and test data are subtracted from the mean image computed from the ILSVRC challenge dataset.

The RF and SVM classifiers are applied to the RGB feature vector as conventional image-based methods. For these methods, there is no need to resize the image patches.

4.5.1.1 Training the Fine-tuning Approaches

In order to fine-tune the CNN models, 20 % of each class of the training data is randomly selected to be used as validation data. The pretrained CNN models are then fine-tuned based on the rest of the training data. Figure 4.12 shows training loss, validation loss, and top-1 and top-5 validation accuracy. Top-1 accuracy is the percentage of the time that the classifier gives a correct class for the class with the highest probability and top-5 accuracy is the percentage of the time that classifier gives a correct class which is among the five classes with the highest probabilities. As we can see in Figure 4.12, top-1 accuracy is close to top-5, which is almost equal to 1. This means that, most of the time, the classes are predicted correctly in top-5. As Figures 4.12 (a) and (b) show, the trend of training and validation losses are similar for VGGNet-16 and VGGNet-19. By comparing training and validation losses, we can see that the pretrained models on the ImageNet can transfer learning to the new dataset by fine-tuning without many iterations. There is a big shift between the validation and training losses in all learning curves because validation loss oscillates around 0.2 when the training loss is low and close to zero. This is the over-fitting problem which might be due to the random selection of validation sets from the training set and using a low number of training dataset in fine-tuning these very deep networks. To overcome this problem, we should select validation sets from different areas without overlap and increase the number of training sets, which needs a great effort without having any available ground truth. Although the over-fitting problem limits the generalization of the models for test areas, we will show in Section 4.5.2 that these fine-tuned models are still usable to obtain promising results for roof type classification. Figure 4.12 (e) demonstrates that the deeper network (ResNet-152) gives more robust answers than the shallower networks and its validation loss fluctuates in the last iterations when its training loss is reduced to zero. Whereas, the validation losses of VGGNet 16/19 have periodical trends and their training losses have small and large fluctuations after converging to zero. The reason behind the fluctuation difference between ResNet and VGGNet models could be their methods for decreasing the learning rate. While VGGNet uses the “step” function to decrease the learning rate in each 5000 iterations, ResNet employs the “multi-step” function.

4.5.1.2 Training SVM on the DeCAFs

In this approach, the DeCAFs are extracted from FC6 and FC7 of the pretrained VGG 16/19, and FC1000 of ResNet 50/101/152 for all training and test patches. The feature vectors for VGGNet and ResNet have 4096 and 1000 dimensions, respectively, for each patch. The SVM classifier with a RBF kernel is then applied to the feature vectors to classify the roofs. In the training process, 5 – *fold* cross-validation is employed on a grid search, which is initialized by the parameters,

$C = 0.1, 1, 10, 100, 1000$ and $\gamma = 1e-2, 1e-3, 1e-4, 1e-5$ to determine the best C and γ . This search processes results in the optimum combination of RBF parameters of $C = 10$ and $\gamma = 0.001$.

4.5.1.3 Training SVM and RF on the RGB Features

In these conventional approaches, the RGB feature vector for each patch is generated by vectorizing each channel and concatenating them. The final feature vector has 9408 dimensions. Based on our experiments, the optimum parameters for the SVM model on the RGB features were $C = 10$ and $\gamma = 0.0001$. Before applying the SVM on the RGB features, the RGB features are normalized based on mean and standard deviation values of each feature vector. The RF estimator fits a number of decision tree classifiers on various sub-samples of the dataset and averaging is used to improve the predictive accuracy and control over-fitting [166]. The sub-sample size is always the same as the original input sample size, but the samples are drawn with replacements [166]. The main parameter to adjust in RF is the number of estimators, which is the number of trees in the forest. In order to adjust the number of estimators of the RF in the training process, a grid search is initialized by the parameters, $n_{estimators} : [33, 50, 80, 101, 200, 301, 500, 600, 750 \text{ and } 1000]$ and a 5 - fold cross validation is then applied to it to obtain the one with the highest accuracy. In our experiments, 1000 estimators were selected.

4.5.2 Discussion of Results

In this section, the fine-tuned models and the trained SVM model are applied to the test dataset. In order to quantitatively evaluate the classification performance of the models, their results are compared to the ground truth [167] using standard measures, such as accuracy, completeness, correctness and $F_{1-score}$ (*Quality*):

$$\begin{aligned}
 Acc. &= \frac{TP}{TP + TN + FN + FP}, & Compl. &= \frac{TP}{TP + FN}, \\
 Corr. &= \frac{TP}{TP + FP}, & F_{1-score} &= \frac{2 \times TP}{2 \times TP + FN + FP}
 \end{aligned} \tag{4.10}$$

In Equation 4.10, TP (True Positive) is the number of patches which belongs to the same class in the both test data and ground truth, FP (False Positive) is the number of patches from different classes which are classified wrongly as current test class, FN (False Negative) is the number of patches which are classified wrongly to the incorrect classes, and TN (True Negative) is the number of patches which do not belong to the same class in the test data and ground truth. Completeness (*Compl.*) measure represents the number of patches which are labeled as belonging to a roof type and indeed belong to that roof type. *Compl.* cannot represent how many patches from one roof type are not labeled correctly. Correctness (*Corr.*)

4.5. Experimental Results and Discussion

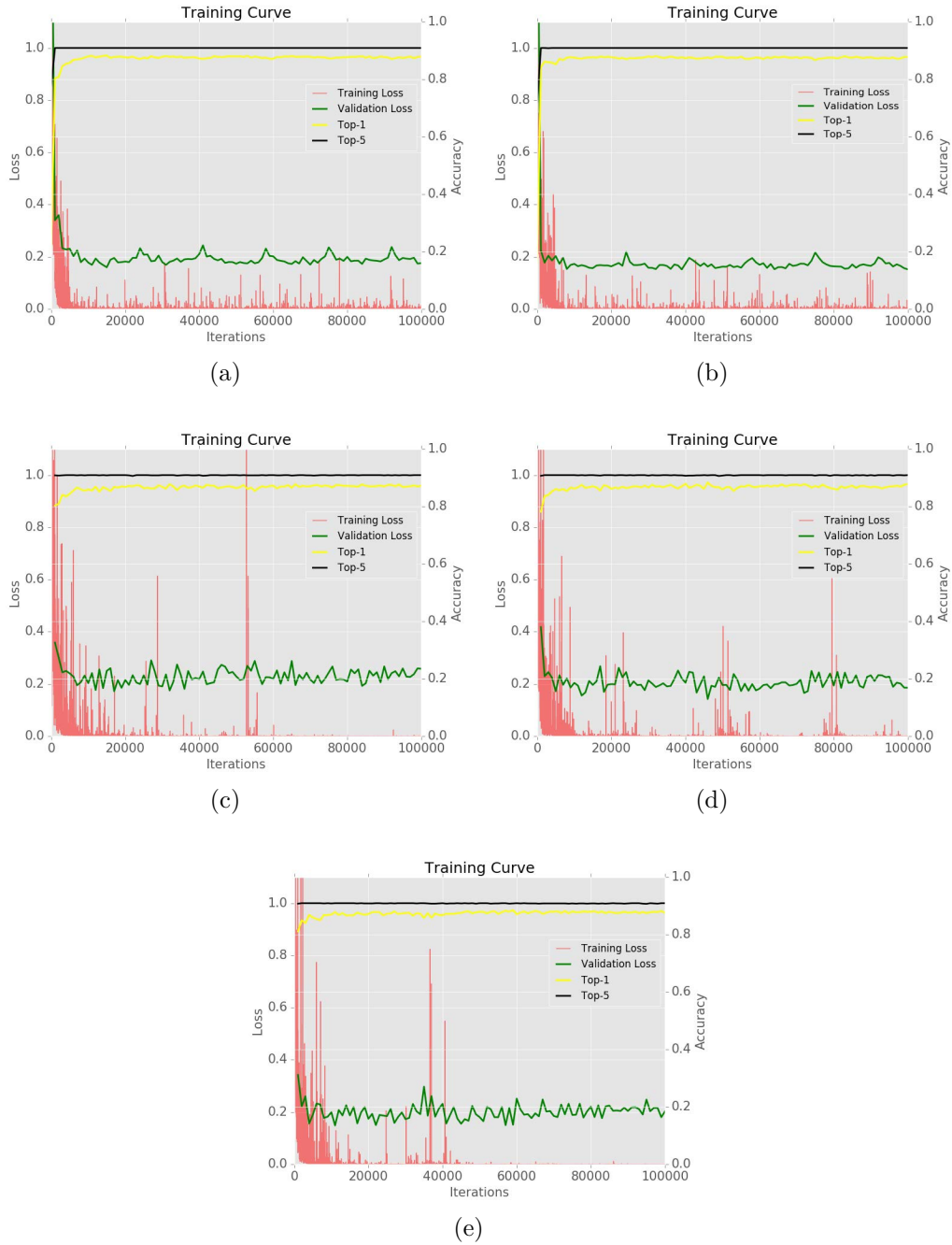


Figure 4.12: Fine-tuning curve including training loss, validation loss, and top-1 and top-5 accuracy (a) VGGNet-16, (b) VGGNet-19, (c) ResNet-50, (d) ResNet-101 and (e) ResNet-152

4. Roof Type Classification

represents the number of patches of one roof type which are labeled correctly. It cannot show how many patches from other roof type classes are labeled incorrectly as belonging to that roof type. To consider the effectiveness of these measures, they are combined into two measures, namely, $F_{1-score}$ and Accuracy ($Acc.$). Tables 4.4 and 4.5 show the results based on three measures of $Comp.$, $Corr.$ and $F_{1-score}$ for each class of the test dataset and for all classes.

Table 4.4 shows the performance of fine-tuned-based classification. According to this table, fine-tuned VGGNet-16 has its highest performance for the flat and gable roofs. It performed moderately for the half-hip, hip and pyramid roofs, and almost failed in the classification of the mansard roofs. Fine-tuned VGGNet-19 and ResNet 50/101/152 performed well for flat, gable and half-hip roofs. They have a lower classification performance for hip and pyramid roofs and almost failed for mansard roofs. The differences between the performances of the fine-tuned models for different roof types are derived from the number of patches which are used from each class for training. As shown in Table 4.1, flat and gable roofs have the highest number of patches in comparison to other roof types, which allows them to transfer more information about their features to the fine-tuned model. According to Table 4.4, the fine-tuned VGGNet-16 and VGGNet-19 performed similarly considering different evaluation measures. In addition, comparing the performance of the fine-tuned ResNets for all classes shows that ResNet-50 and ResNet-101 perform similarly, whereas ResNet-152 (with the $Comp.$ of 89.06%, the $Corr.$ of 89% and the $F_{1-score}$ of 88.99%) outperforms the other ResNets for all roof type classes. Altogether, ResNets outperform VGGNets by 6% in $Comp.$, 3% in $Corr.$ and 5% in $F_{1-score}$.

Table 4.5 represents the classification results based on the application of SVM to different DeCAFs. According to the results, similar to the fine-tuned-based approaches, SVM performs well for the flat and gable roofs, moderately for the half-hip and hip, and poorly for pyramid and mansard roofs. As has been already mentioned, this is due to the different number of samples used for training. The classification performance of SVM using two different FC layers of the VGGNets, namely FC6 and FC7, are shown and compared in Table 4.5. These layers form a higher level of abstraction and contain richer semantic information than the earlier layers in the networks. According to the results for VGGNet 16/19, since FC6 is located at a lower level than FC7 and, therefore, presents features with a lower level of abstraction, SVM could better adapt to our classification problem using FC6 features. Consequently, the results are slightly better than using FC7. By comparing the performances of SVM with VGGNet-16 and VGGNet-19 features, it can be concluded that using a deeper network, such as VGGNet-19 with three more convolutional layers than VGGNet-16, can improve the classification performance to some degree (about 2%). However, VGGNet-16 performs better for the pyramid roof class.

4.5. Experimental Results and Discussion

Fine-tuned VGGNet-16

Images	TP	FN	FP	TN	Compl.	Corr.	F_1 -score
flat	971	187	82	2890	84%	92%	88%
gable	1580	180	242	2128	90%	87%	88%
half-hip	743	227	108	3052	77%	87%	81%
hip	128	55	90	3857	70%	59%	64%
pyramid	22	10	39	4059	69%	36%	47%
mansard	6	21	119	3984	22%	5%	8%
All classes	3450	680	680	19970	83.54%	86.23%	84.63%

Fine-tuned VGGNet-19

Images	TP	FN	FP	TN	Compl.	Corr.	F_1 -score
flat	944	214	97	2875	82%	91%	86%
gable	1561	199	226	2144	89%	87%	88%
half-hip	781	189	151	3009	81%	84%	82%
hip	125	58	82	3865	68%	60%	64%
pyramid	21	11	25	4073	66%	46%	54%
mansard	6	21	111	3992	22%	5%	8%
All classes	3438	692	692	19958	83.24%	85.40%	84.18%

Fine-tuned ResNet-50

Images	TP	FN	FP	TN	Compl.	Corr.	F_1 -score
flat	1062	96	139	1770	92%	88%	90%
gable	1594	166	214	609	91%	88%	90%
half-hip	775	195	93	2216	80%	89%	84%
hip	122	61	62	3799	67%	66%	66%
pyramid	21	11	23	4053	66%	48%	55%
mansard	4	23	21	4073	15%	16%	15%
All classes	3578	552	552	16520	86.63%	86.75%	86.60%

Fine-tuned ResNet-101

Images	TP	FN	FP	TN	Compl.	Corr.	F_1 -score
flat	1061	97	143	2829	92%	88%	90%
gable	1587	173	191	2179	90%	89%	90%
half-hip	769	201	109	3051	79%	88%	83%
hip	122	61	56	3891	67%	69%	68%
pyramid	19	13	26	4072	59%	42%	49%
mansard	7	20	40	4063	26%	15%	19%
All classes	3565	565	565	20085	86.32%	86.78%	86.47%

Fine-tuned ResNet-152

Images	TP	FN	FP	TN	Compl.	Corr.	F_1 -score
flat	1066	92	76	1830	92%	93%	93%
gable	1646	114	190	534	94%	90%	92%
half-hip	817	153	100	2234	84%	89%	87%
hip	126	57	53	3768	69%	70%	70%
pyramid	18	14	13	4067	56%	58%	57%
mansard	5	22	20	4078	19%	20%	19%
All classes	3678	452	452	16520	89.06%	89.00%	88.99%

Table 4.4: The results of fine-tuned CNN models

4. Roof Type Classification

VGGNet-16 features + SVM								
Images	FC	TP	FN	FP	TN	Compl.	Corr.	$F_{1-score}$
flat	FC7	979	179	175	2797	85%	85%	85%
	FC6	1002	156	175	2797	87%	85%	86%
gable	FC7	1241	519	307	2063	71%	80%	75%
	FC6	1290	470	323	2047	73%	80%	76%
half-hip	FC7	645	323	419	2743	67%	61%	63%
	FC6	643	325	403	2759	66%	61%	64%
hip	FC7	81	102	188	3759	44%	30%	36%
	FC6	74	109	139	3808	40%	35%	37%
pyramid	FC7	2	32	40	4056	6%	5%	5%
	FC6	8	26	36	4060	24%	18%	21%
mansard	FC7	2	25	51	4052	7%	4%	5%
	FC6	3	25	35	4068	7%	5%	6%
All classes	FC7	2950	1180	1180	19470	71.43%	73.56%	72.26%
	FC6	3019	1111	1111	19539	73.10%	74.08%	73.49%

VGGNet-19 features + SVM								
Images	FC	TP	FN	FP	TN	Compl.	Corr.	$F_{1-score}$
flat	FC7	967	191	153	2819	84%	86%	74%
	FC6	987	171	169	2803	85%	85%	85%
gable	FC7	1333	427	344	2026	76%	79%	85%
	FC6	1347	413	316	2054	76%	81%	79%
half-hip	FC7	613	355	398	2764	63%	61%	62%
	FC6	657	311	373	2789	68%	64%	66%
hip	FC7	78	105	162	3785	43%	33%	37%
	FC6	82	101	117	3830	45%	41%	43%
pyramid	FC7	1	33	33	4063	3%	3%	3%
	FC6	1	33	34	4062	3%	3%	2%
mansard	FC7	2	25	46	4057	7%	4%	5%
	FC6	3	24	44	4059	7%	6%	8%
All classes	FC7	2994	1180	1180	19514	72.49%	73.78%	73.07%
	FC6	3077	1053	1053	19597	74.50%	75.30%	74.85%

ResNet 50/101/152 features + SVM								
Images	FC1000	TP	FN	FP	TN	Compl.	Corr.	$F_{1-score}$
flat	resnet50	994	164	183	2789	86%	84%	85%
	resnet101	1000	158	273	2699	86%	79%	82%
	resnet152	1004	154	183	2789	87%	85%	86%
gable	resnet50	1325	435	313	2057	75%	81%	78%
	resnet101	1273	487	259	2111	72%	83%	73%
	resnet152	1342	418	316	2054	76%	81%	79%
half-hip	resnet50	621	347	406	2756	64%	60%	62%
	resnet101	653	315	386	2776	67%	63%	65%
	resnet152	682	286	394	2768	70%	63%	67%
hip	resnet50	80	103	156	3791	44%	34%	38%
	resnet101	93	90	115	3832	51%	45%	48%
	resnet152	77	106	87	3860	42%	47%	44%
pyramid	resnet50	8	26	27	4069	24%	23%	23%
	resnet101	15	19	38	4058	44%	28%	34%
	resnet152	11	23	16	4080	32%	41%	36%
mansard	resnet50	0	27	17	4086	0%	0%	0%
	resnet101	3	24	22	4081	11%	12%	12%
	resnet152	1	26	17	4086	4%	6%	4%
All classes	resnet50	3028	1102	1102	19548	73.32%	74.01%	73.58%
	resnet101	3037	1093	1093	19557	73.54%	74.46%	73.74%
	resnet152	3117	1013	1013	19637	75.47%	75.52%	75.41%

Table 4.5: Results of SVM on the DeCAFs

RGB features+RBF-SVM classifier							
Images	TP	FN	FP	TN	Compl.	Corr.	$F_{1-score}$
flat	743	415	417	2555	64%	64%	64%
gable	1058	702	500	1870	60%	68%	64%
half-hip	474	494	458	2704	49%	51%	50%
hip	62	121	241	3706	34%	20%	26%
pyramid	0	34	146	3950	0%	0%	0%
mansard	1	26	30	4073	4%	3%	3%
All classes	2338	1792	1792	18858	56.61%	59.75%	58%
RGB features+RF classifier							
Images	TP	FN	FP	TN	Compl.	Corr.	$F_{1-score}$
flat	840	318	194	2778	73%	81%	77%
gable	1345	415	573	1797	76%	70%	73%
half-hip	431	537	306	2856	45%	58%	51%
hip	53	130	209	3738	29%	20%	24%
pyramid	1	33	123	3973	3%	0%	1%
mansard	5	22	50	4053	19%	9%	12%
All classes	2675	1455	1455	19195	64.77%	67.33%	65.65%

Table 4.6: Conventional classification approaches

Among ResNet models, the deepest one (ResNet-152), with the *Compl.* of 75.47%, the *Corr.* of 75.52% and the $F_{1-score}$ of 75.41%, performs slightly better than the others. However, the SVM classification using ResNet features for the pyramid and mansard roofs performs poorly in comparison to the SVM on VGGNet features, and SVM on the ResNets features performs slightly better for pyramid roofs. ResNet-152 features, for example, classified the pyramid roofs with the *Compl.* of 32%, the *Corr.* of 41% and the $F_{1-score}$ of 36%, whereas SVM on FC6 features of VGGNet-16 resulted in a classification with the *Compl.* 24%, the *Corr.* of 18% and the $F_{1-score}$ of 21%. ResNet models generally outperform the VGGNet models, according to Tables 4.4 and 4.5, for the approach based on fine-tuning and the approach using the DeCAFs. In addition, the ResNet models are less sensitive to the small number of training samples, such as for the mansard roofs, compared to the VGGNet models. Moreover, it can be concluded that the approach based on fine-tuning has a better performance than the ones based on DeCAFs when the number of training samples is limited. Altogether, according to the results, the fine-tuned ResNet-152 with the *Compl.* of 89.06% and the *Corr.* of 89% and with the $F_{1-score}$ of 88.99% on the test data outperformed all the other fine-tuned models and the DeCAFs.

Table 4.6 shows the performances of conventional classifiers, such as RF and SVM, on the RGB features. As the results show, their performance is dependent on the number of training samples. Thus, they performed well for the flat and gable roofs, performed moderately for the half-hip roofs and almost failed in the classification of the other roof types. In comparison to the DeCAFs, SVM is more

4. Roof Type Classification

Roof type classification methods	Compl.	Corr.	F_1 -score	Acc.
SVM with RGB features	56.61%	59.75%	58%	56.61%
RF with RGB features	64.77%	67.33%	65.65%	64.77%
SVM with VGGNet-19 (FC6) features	74.50%	75.30%	74.85%	74.50%
SVM with ResNet-152 features	75.47%	75.52%	75.41%	75.47%
fine-tuned VGGNet-16	83.54%	86.23%	84.63%	83.54%
fine-tuned ResNet-152	89.06%	89.00%	88.99%	89.06%

Table 4.7: Comparison of $Compl.$, $Corr.$, F_1 -score, and $Acc.$ for the best results of DeCAFs, the best results of fine-tuned CNN models and conventional models, such as RF and SVM, with RGB features

	TP		FN		FP		Compl.		Corr.		F_1 -score		Total Acc.	
	DSM	Image	DSM	Image	DSM	Image	DSM	Image	DSM	Image	DSM	Image	DSM	Image
flat	638	1066	520	92	135	76	55.09%	92.06%	82.54%	93.35%	49.34%	86.39%	84.14%	95.93%
Non-flat	2837	2896	135	76	520	92	95.46%	97.44%	84.51%	96.92%	81.24%	94.52%		

Table 4.8: Classification of flat and non-flat roofs based on the DSM and image-based method (ResNet-152)

sensitive to the training sample number using RGB features. According to Table 4.6, the RF classifier on the RGB features performs about 7% better than SVM classifier on the RGB features for all classes.

Table 4.7 compares the performances of the CNN-based and conventional approaches for CNN-based approaches; the result of each model which has the best performance in Tables 4.4 and 4.5 is used for comparison. The approach based on fine-tuning generally performed better than the approach based on DeCAFs. Moreover, using ResNet-152 features for the SVM classifier resulted in a higher performance in comparison to the VGGNet features. Table 4.7 also shows that deep learning-based approaches perform better than the conventional methods, such as SVM and RF, on RGB features.

As discussed in Section 4.4, extracted mean and Gaussian curvature features from DSM can be used to classify the flat and non-flat roofs. The DSM patches were generated in the same way as the test patches for the image-based classification to compare the results of DSM-based classification with image-based classification. The mean and Gaussian curvature images were then provided for each patch. The patches were classified into flat and non-flat roofs based on the sign of mean and Gaussian curvatures using Equation 4.9. The results of the DSM-based classification are then compared with those of image-based classification for the flat and non-flat classes as presented in Table 4.8. The results of the fine-tuned ResNet-152 are reported for the image-based classification. As shown in Table 4.8, ResNet-152 outperforms the DSM-based classification for just discriminating flat from non-flat roofs.

4.6 Summary and Conclusion

In this chapter, the potential of the DSM of satellite imagery to define a roof building library is evaluated. A simple roof type classification for two categories (flat and non-flat) is employed based on surface curvatures of the DSM. Although this method can obtain the accuracy of 84 % which is better than some of the methods presented in Table 4.7, it is limited to two types of roof. Image-based classification used two pretrained VGGNet and ResNet to classify the roof types other than non-flat roofs, such as gable, half-hip, hip, pyramid and mansard roofs. Two types of pretrained CNNs, which are RBF-SVM with VGGNet and ResNet features of fully connected layers and the fine-tuning ResNet 50/101/152, are evaluated. ResNet-152 with an accuracy of 89.06 % obtains the best result among the different models. Figure 4.13 shows the final results of the classification with fine-tuned ResNet-152, which is finally used for the 3D reconstruction in the next chapter.

4. Roof Type Classification

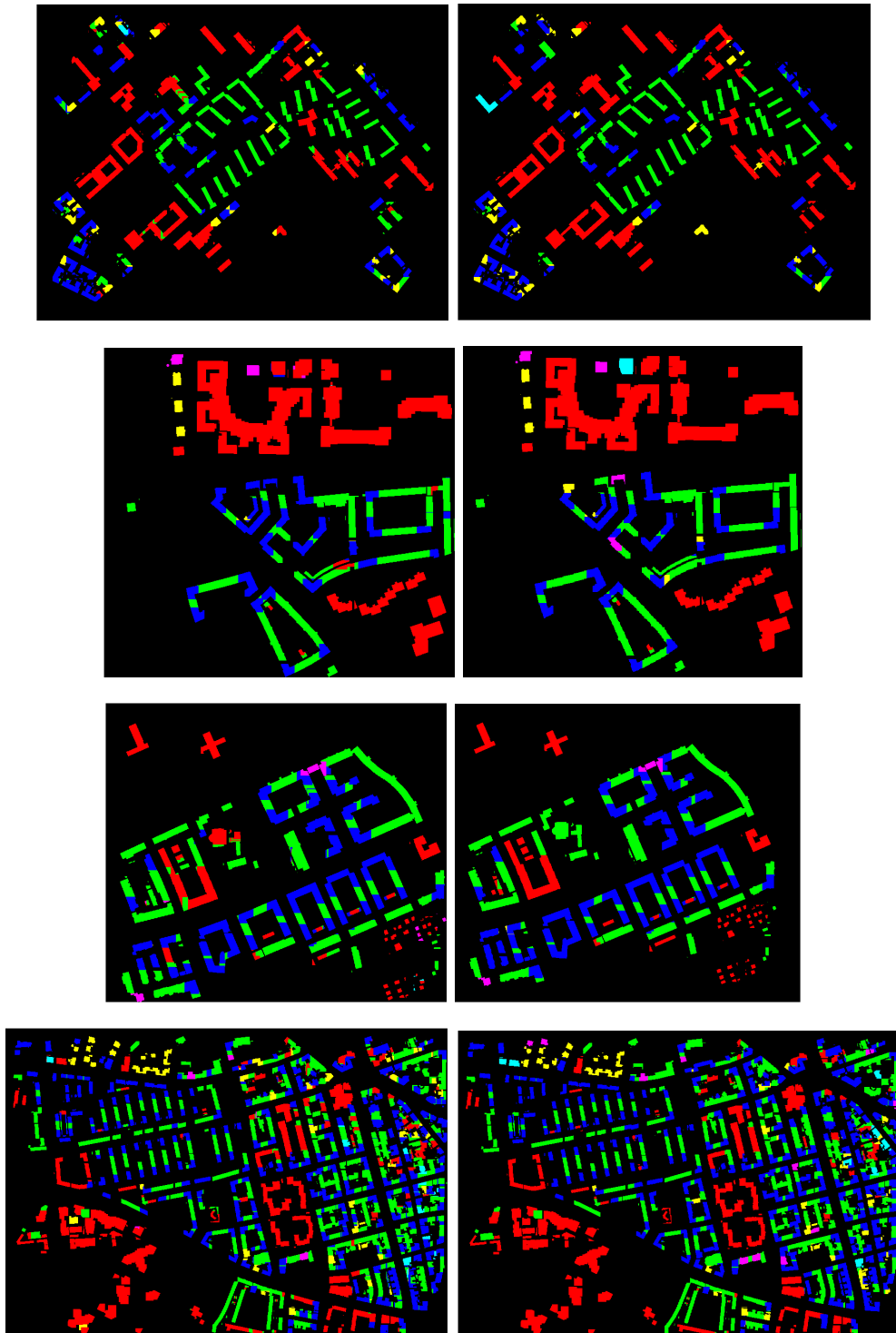


Figure 4.13: Illustration of classified roof types (green: gable, blue: half-hip, red: flat, yellow: hip, light blue: mansard, pink: pyramid roofs, (left) the results of four different areas, (right) their related ground truths

3D Building Model Reconstruction

In this dissertation, the concept of 3D building reconstruction from satellite data is to create a watertight parameterized 3D model of a building so that the geometrical parameters, such as the slope and volume of the model, can be estimated. The proposed reconstruction method is based on some data-driven steps which have been presented in the previous chapters. The data-driven steps for reconstructing 3D building models from the DSM of satellite imagery are as follows: Extracting the parameterized building outlines (Chapter 2), decomposing them into rectangular shapes (Chapter 3) and selecting the roof type for each part of the building block (Chapter 4). After all these data-driven steps, the buildings are then reconstructed by assembling simple rectangular structures, which is shown in this Chapter.

In this chapter, a hybrid 3D building model reconstruction is detailed where the model is defined as a geometrical shape with some parameters which are initialized by the preknowledge obtained from data-driven steps (Section 5.1). Using the building mask and the nDSM, this initialization is then improved in (Section 5.2 and Section 5.3). The improvement of the initial parameters consists of the improvement of rectangles by shifting and orienting them to coincide with the building mask boundary and decomposing rectangles by detecting height discontinuities and roof types. Moreover, the roof type classification results obtained from the previous chapter are improved by combining them in each rectangle based on a predefined set of combination rules and then by fusing them with the nDSM in a Bayesian framework. In the end, a modified parameter optimization based on the brute-force search is used separately in 2D and 3D to find the final model as the best fitting model to the nDSM among all possible models (Section 5.4). Figure 5.1 depicts the workflow of the proposed multistage hybrid method for 3D building model reconstruction.

5. 3D Building Model Reconstruction

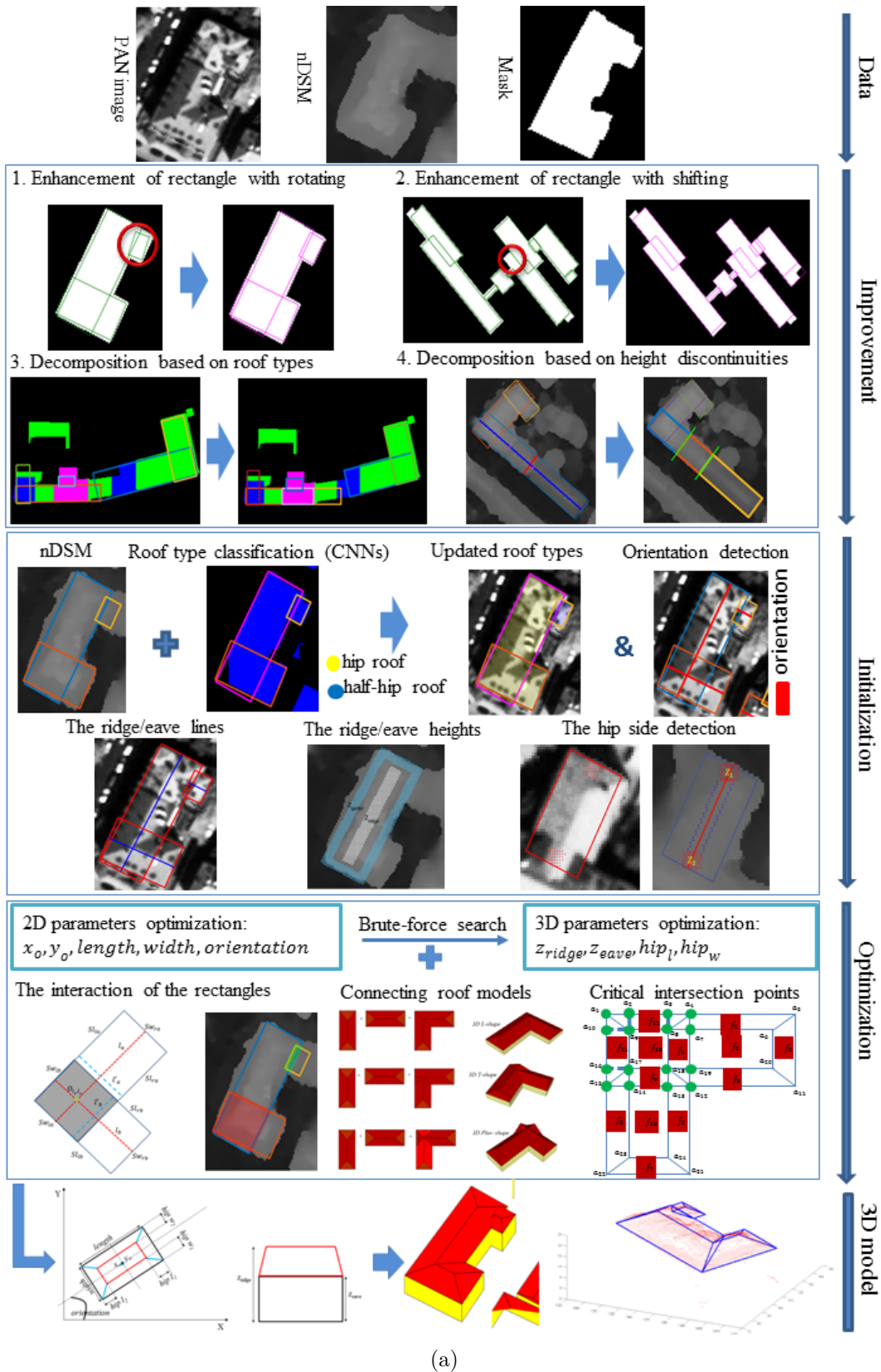


Figure 5.1: Workflow of the proposed method.

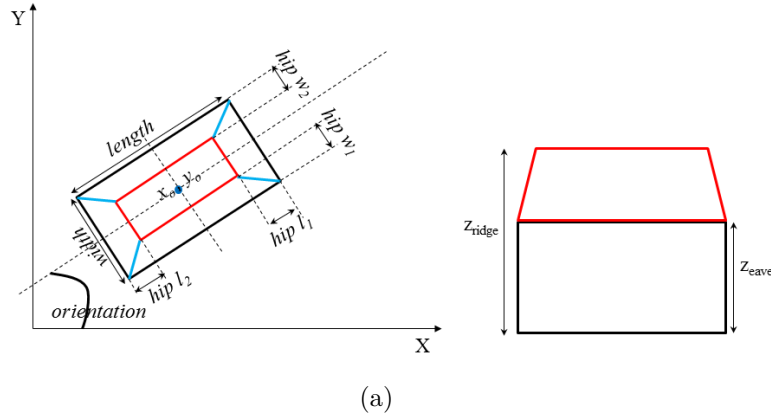


Figure 5.2: Geometrical parameters of the roof models

5.1 Geometrical Structure of the Roof Model

In Chapter 3, building blocks were decomposed into a number of simple rectangles. In Chapter 4, a library of the primitives, including flat, gable, half-hip, hip, pyramid and mansard roofs, that are visible in DSMs and satellite images was then defined, which is necessary for model-driven methods.

The primitive library for the proposed hybrid method is categorized into two groups: Single- and multiplane roofs, as can be seen in Figure 5.3. In this dissertation, the planar roofs and rectangular footprints are used due to not only their simplicity and small number of parameters, but also their basic forms, which can cover most building shapes in urban areas. The geometrical parameters of the primitives in the library are defined as:

$$\psi \in \Psi; \Psi = \{\mathcal{P}, \mathcal{C}, \mathcal{S}\}, \quad (5.1)$$

where the parameter Ψ contains the position parameters $\mathcal{P} = \{x_o, y_o, orientation\}$, the contour parameters $\mathcal{C} = \{length, width\}$, and the shape parameters \mathcal{S} including ridge/eave (gutter) height and the longitudinal and latitudinal hip distances, which are Z_{ridge} , Z_{eave} , $hipl_1$, $hipl_2$, $hipw_1$ and $hipw_2$, respectively. Figure 5.2 illustrates the geometrical parameters of a roof model. The roof components, such as vertices, edges and facets, and their relationships are determined from the geometrical parameters of the primitives [101].

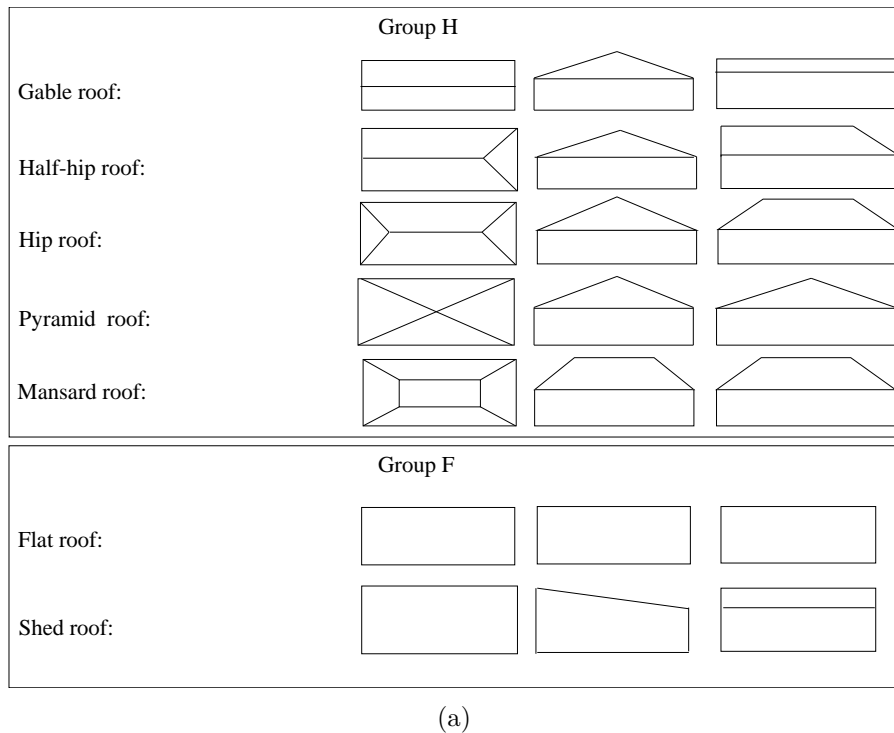


Figure 5.3: Groups of single and multiplane roof models

5.2 Enhancement of 2D Roof Model Rectangles

In Chapters 3 and 4, building roofs were decomposed and classified into several rectangles. Some of the rectangles have errors from the decomposition and selection steps which need to be improved. The improvement consists of two parts: 2D enhancement in the rectangle parameters and detection of the height discontinuities in two main orientations of each rectangle.

5.2.1 Enhancement of the Rectangle Parameters

The rectangle sides obtained from the extraction, decomposition and selection steps do not always fit the boundaries of building masks perfectly, as they suffer from rotation and shift errors, as shown in Figures 5.5 (a) and (c). To correct the rotation error, each rectangle is converted into a binary image and dilated with a disk-shaped structure element of two-pixel size to overlap the boundary of the building mask. A Hough transform is then used to find co-linear points and the orientation of their corresponding line (θ_1). The difference between θ_1 and the orientation of the rectangle side (θ_2) gives the degree which the rectangle should be rotated to fit the building mask:

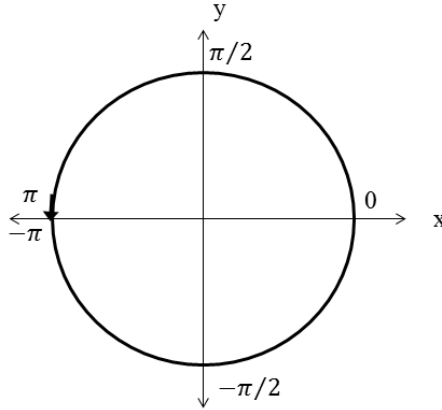


Figure 5.4: The computational domain of four-quadrant inverse tangent of elements of X and Y

$$\Delta\theta = \theta_1 - \theta_2 = \text{atan2}\left(\frac{\Delta Y_1}{\Delta X_1}\right) - \text{atan2}\left(\frac{\Delta Y_2}{\Delta X_2}\right) \quad (5.2)$$

In this equation, ΔX and ΔY are the differences between the coordinates of the Hough line and the rectangle side, and $\text{atan2}()$ is the four-quadrant inverse tangent which returns values in $[-\pi, \pi]$, as shown in Figure 5.4. The corrected rectangle of Figure 5.5 (a) is illustrated in Figure 5.5 (b).

Several lines orthogonal to the width and length of the rectangle are considered, so-called profile lines, to calculate the shifts between rectangle sides and boundaries of the building mask. Figure 5.6 shows the building mask, the corresponding rectangles and the profile lines. As the figure illustrates, the extent of the profile lines which cover a small area outside the boundary intersect with the boundary of the building mask. The distance between the intersection points and the rectangle sides is then used to measure the shift error. In the next step, each side of the rectangle is shifted to fit into the boundaries of the building mask. Figure 5.7 shows the parameters of the profile lines. In this figure,

- ds stands for the distance between two successive profiles,
- rs corresponds to the sampling resolution of the building mask on the profile, and
- l_{ext} corresponds to half of the rectangle's length and width for orthogonal profile lines.

Figure 5.8 illustrates a rectangle (magenta lines) after the shift correction (blue lines).

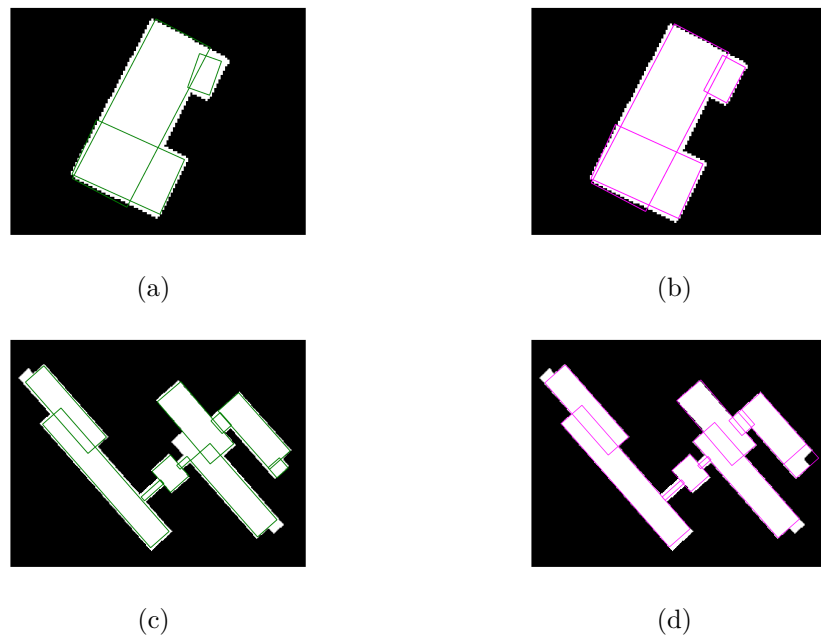


Figure 5.5: (a) The decomposed rectangles, (b) the decomposed rectangles after imposing the rotation, (c) the decomposed rectangles, and (d) the decomposed rectangle after imposing the shift

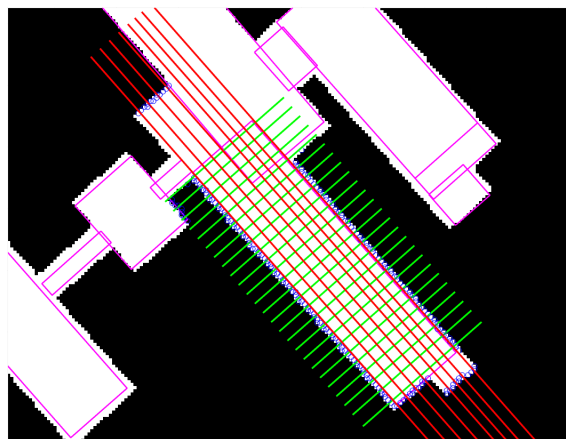


Figure 5.6: Profiles acquisition; red lines are the measuring slices perpendicular to the width and green lines are the measuring slices perpendicular to the length of the rectangle. Blue points are intersection points of the profile and building mask

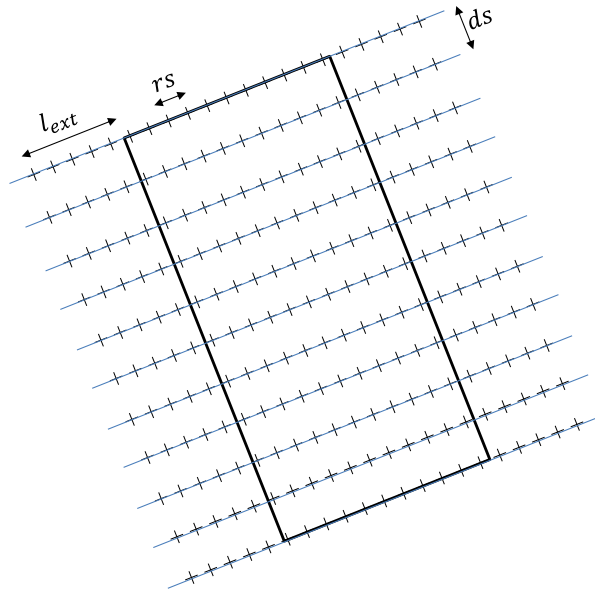


Figure 5.7: The profiles acquisition of the rectangle and their parameters: ds , rs , and l_{ext} . Plus shape points stand for pixels and blue lines show the line profiles.

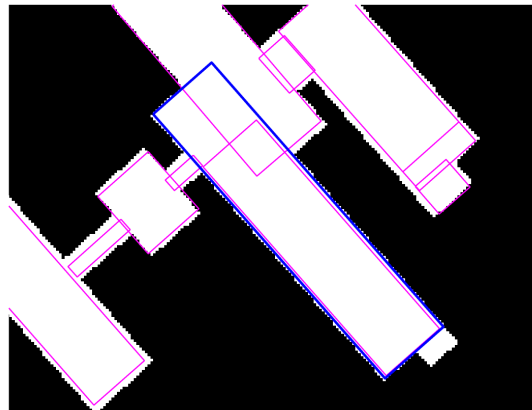


Figure 5.8: The correction of the rectangle after imposing the shift: Blue rectangle

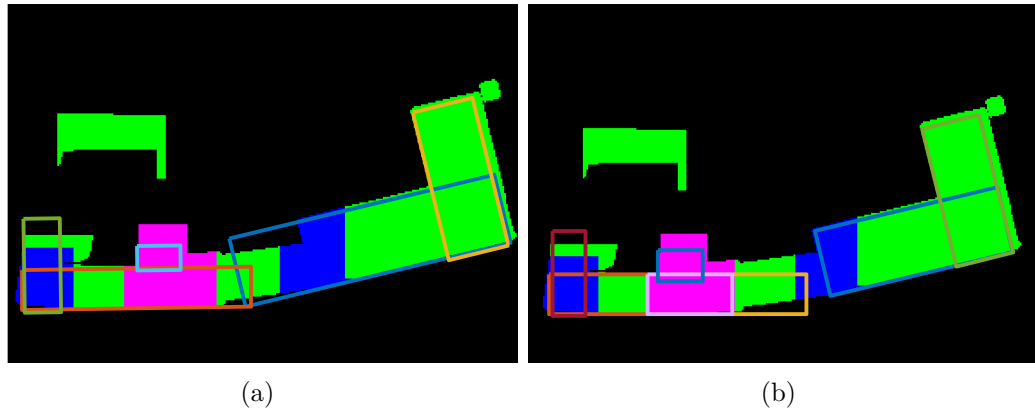


Figure 5.9: Decomposition of a rectangle after imposing the rules of roof type combination: The pyramid roof is shown in magenta (a) before decomposition and (b) after decomposition

5.2.2 Decomposition of the Rectangles based on the Roof Types

The rectangles can be improved by decomposing them using some rules defined based on roof types combinations. The roof types are obtained from the classification in Chapter 4. Since the roof types are classified based on a patch-based deep learning method, there are different roof types within the rectangles. In the case of having more than one roof type, the combination of roof types in one rectangle is verified. Combination of gable with half-hip and hip roof is meaningful, which results in half-hip or hip roofs. A rectangle is decomposed where the flat roof neighbors one of the sloped roofs. Pyramid and mansard roofs are also decomposed where they neighbor other sloped or flat roof types. In Figure 5.9, for instance, a pyramid roof is separated from other roof types by decomposing rectangles where a pyramid is changed to another roof type.

5.2.3 Detection of the Height Discontinuities

The rectangles can be improved by detecting height discontinuities. The height discontinuity points are detected by grouping the pixels on the profiles of the middle lines. Since the basic shape of the model is rectangular, the height discontinuities are measured on two profile lines which are parallel (for flat and sloped roofs) and perpendicular (for only flat roofs) to the longest line of the rectangle. Each pixel on these two lines has different height values (Z) obtained from the nDSM. The gradient (G) is then computed as the profile derivative based on the predefined resolution rs (here rs is selected as one pixel):

$$\frac{\Delta Z}{\Delta x}(i) = \frac{Z(i + rs) - Z(i)}{rs} = \frac{G(i)}{rs}. \quad (5.3)$$

Pixels with gradients (height differences) less than a threshold (threshold selected to be one meter) are then clustered in one group. Each group should contain more than four pixels, otherwise it is not considered in computing height discontinuities. If the difference between the height averages of two sequential groups is more than a threshold (threshold selected to be 2.5 meter), there is a height discontinuity point between them. The number of groups with the property mentioned previously presents the number of jumps. The points with large gradients where the groups are separated are detected as the height discontinuity points. The points detected on the rectangle sides are then excluded. Finally, the rectangles are split at the positions of the discontinuity points. Figure 5.10 shows the result of height discontinuity detection for a building.

5.3 Selection and Initialization of Roof Model Parameters

After improving the rectangles, the model fitting algorithm requires some initialization. The main orientation of the roof model is detected during the initialization procedure and the classification results from Chapter 4 are improved based on the height information from the nDSM. The other initial roof model parameters, such as length, width, rectangle center, height of the eave, height of the ridge and hip distances, are then calculated.

5.3.1 Detection of Building Roof Model Orientation

In order to determine the main orientation of the rectangles, which can be parallel to the length or width of the rectangle, three profiles are computed, one in the middle of the rectangles and two with an offset from the middle for each orientation.

The maximum Z values of the three profiles for each orientation are then computed. The maximum value of each orientation is replaced by the mean of Z values of the neighboring pixels within a window size of $width_{rectangle}/4$ to reduce the noise effect of the DSM. After that, a threshold is defined according to the minimum value of these maximum values. The number of points (N_{Points}) which have Z values higher than this min value is then determined for each orientation. The orientation of the line with the maximum N_{Points} is selected as the main orientation of the building. This line is also used to detect the building ridge lines in the sloped roofs.

The height points of neighboring rectangles can affect the determination of the main orientation of each rectangle for complex buildings with several parts and orientations, where the rectangles overlap each other. Dealing with this issue, the height values of half of the overlapping pixels related to neighboring rectangles are lowered to the height of the rectangle's border. The main orientation of the building is then

5. 3D Building Model Reconstruction

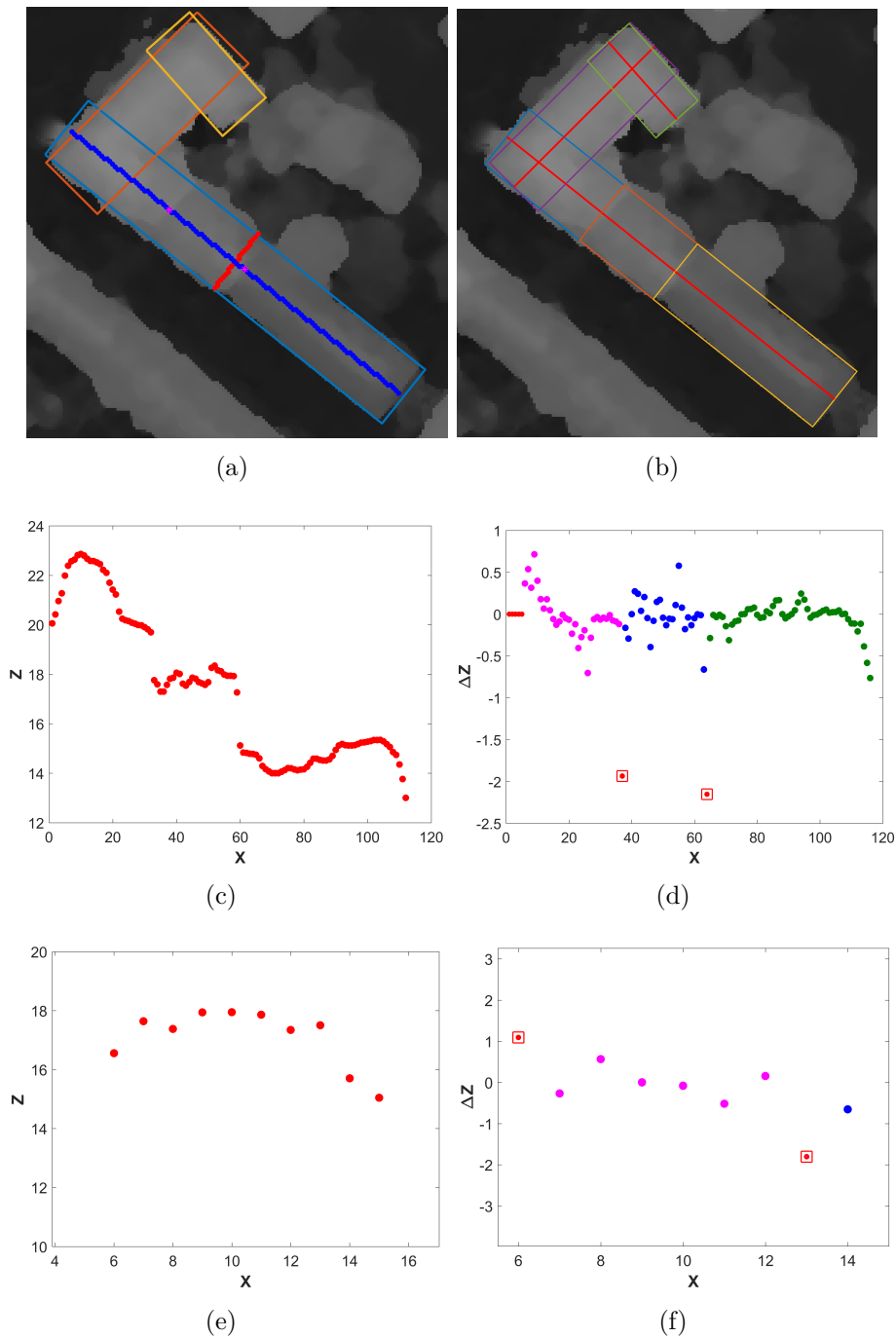


Figure 5.10: Height discontinuities detection: (a) Longitudinal profile line (blue) and latitudinal profile line (red) drawn on the nDSM; (b) decomposition of the blue rectangle in (a) based on height discontinuity points (two magenta points in (a)); (c) the 3D point clouds projected in the 2D plane passing the longitudinal profile (red points); and (d) the height differences between the 3D point clouds along the longitudinal profile ΔZ ; three groups of points in magenta, blue and green. The height discontinuity points are shown in red dot squares. (e) The 3D point clouds drawn in the 2D plane passing the latitudinal profile (red points). (f) The height differences (the gradients) between the 3D point clouds along the latitudinal profile ΔZ ; two groups of points in magenta and blue. The height discontinuity points are shown in red dot squares.

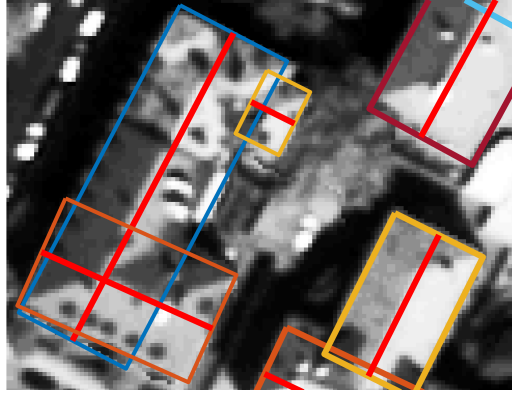


Figure 5.11: Roof model orientation detection

computed according to what has been explained above. Figure 5.11 shows the result of roof model orientation detection for a *U – shape* building.

5.3.2 Improvement of Roof Type Classification Results

Before fitting a 3D model to each rectangle, its corresponding roof type should be determined, since different roof types might have been assigned to each rectangle through the classification step. Only one roof type is selected for each rectangle from all roof types that have been assigned to it based on a set of rules to improve the roof type classification results. To improve the classification results within each rectangle, firstly, a rectangle is divided into three parts. A roof type is then assigned to each part based on the majority of pixels in each class related to each part of the rectangle. Next, a set of rules based on the combination of these roof types is determined as follows:

- {half-hip,gable,half-hip} \longrightarrow {hip},
- {half-hip,half-hip,hip} \longrightarrow {hip},
- {hip,half-hip,half-hip} \longrightarrow {hip},
- {hip,half-hip,hip} \longrightarrow {hip},
- {gable,half-hip,half-hip} \longrightarrow {half-hip},
- {gable,gable,half-hip} \longrightarrow {half-hip},
- {half-hip,gable,gable} \longrightarrow {half-hip},
- {half-hip,half-hip,gable} \longrightarrow {half-hip},
- {gable,gable,gable} \longrightarrow {gable},

5. 3D Building Model Reconstruction

	Flat	Gable	Half-hip	Hip	Pyramid	Mansard
Flat	0.97	0.027	0.006	0	0	0
Gable	0.028	0.908	0.0028	0.066	0	0
Half-hip	0	0.094	0.81	0.0915	0	0.0045
Hip	0.006	0.056	0.087	0.85	0	0
Pyramid	0	0.144	0.36	0.12	0.38	0
Mansard	0.51	0	0.215	0.275	0	0

Table 5.1: The confusion matrix of the classification for all four areas (Chapter 4)

		Probability					
Results of classification		flat	gable	half-hip	hip	pyramid	mansard
	flat	0.75	0.166	0.083	0	0	0
	gable	0.083	0.67	0.083	0.16	0	0
	half-hip	0	0.133	0.417	0.417	0.033	0
	hip	0.05	0.116	0.417	0.417	0	0
	pyramid	0	0.166	0.166	0.166	0.5	0
	Mansard	0.25	0	0.166	0.166	0	0.417

Table 5.2: prior probabilities of roof models

- {half-hip, half-hip, half-hip} \rightarrow {half-hip},
- {hip, hip, hip} \rightarrow {hip},
- {gable, hip, gable} \rightarrow {hip}, and {gable},
- {gable, half-hip, gable} \rightarrow {gable},

Algorithm 1 in Appendix D shows how the roof types classification results are updated according to these predefined rules.

After using the rules mentioned, each rectangle has one type of roof. A deep learning method was only applied on the patches of images without using the nDSM to classify the roof types in Chapter 4. The size of each patch is fixed and cannot cover a whole rectangle, especially in the case of long rectangles. Therefore, one rectangle can have either more than one roof type or only one incorrect roof type. A rectangle, for instance, is classified as a half-hip roof in the Figure 5.12, while it is a hip roof in reality. To solve this problem and to modify the classification results, the height information from the nDSM is used to make a final decision on the roof

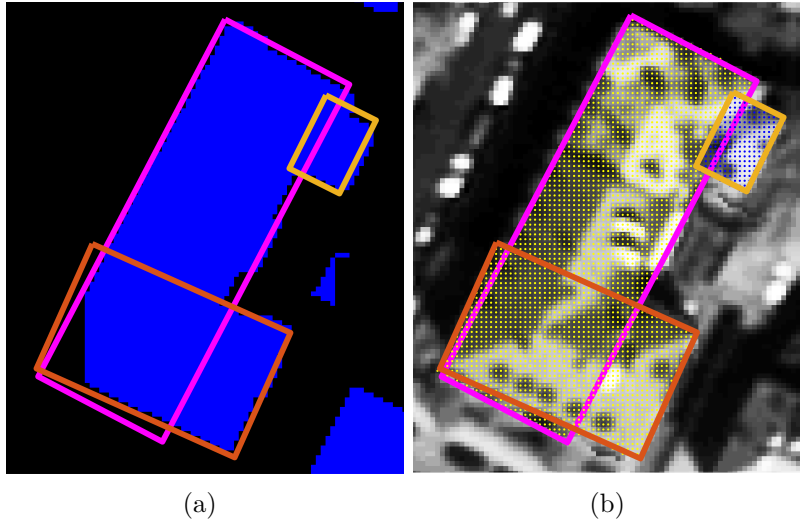


Figure 5.12: Roof type refinement: (a) The result of the classification before refinement, (b) roof types after refinement; blue and yellow colors stand for half-hip and hip roofs, respectively

type. Consequently, the roof models with different prior probabilities obtained from deep learning-based classification are fitted to the nDSM of each rectangle to achieve data information about roof models. The final roof types are classified using prior probabilities and the data information of roof models in a Bayesian formula.

Table 5.1 shows the confusion matrix of the classification for all four areas which are used in this dissertation. Each row of the table stands for the true positive and false negative ratios of one roof type. The true positive ratio is the percentage of the patches which are classified correctly for one roof type and the false negative ratio is the percentage of the misclassified patches of that roof type. Considering the false negative of a roof type gives the opportunity for refinement of the classification results. If some patches of the flat roof, for instance, are classified as gable and half-hip roofs (false negative), the prior probabilities are computed for them. According to Table 5.1, prior probabilities of each class which is determined by the user should sum to one (Table 5.2). In this table, the roof which is chosen from the classification takes the highest probability among the other roof types. In the first row, for instance, the flat roof takes the highest probability, but the gable and half-hip roofs (as false negative roof types) also take a portion in the final decision of roof type selection. Table 5.1 only conducts the user to give probabilities to each type of roof and is useful to create Table 5.2. Since this table is obtained from the classification on the small areas, it is not perfectly reliable to present the prior information about the roof types and to use for fusing with the nDSM. The probability of the flat roofs, for instance, is 97%, which is very high, and it cannot allow nDSM information to participate in the final decision about roof type. Therefore, the handcrafted probability is calculated for each roof type as a prior

probability (Table 5.2) according to the statistical information obtained from the roof type classification results (Table 5.1). The prior probability of each roof type is the proportion of six, which is the number of roof types. The summation of each row is equal to one. In the first row of Table 5.2, for instance, the probabilities of the flat, gable and half-hip roofs are determined as $\frac{4.5}{6} = 0.75$, $\frac{1}{6} = 0.166$ and $\frac{0.5}{6} = 0.083$, respectively.

Based on this predefined probability, a Bayesian formulation is used to select the best roof type. The Bayesian approach is known to be robust and useful for parameter estimation problems. A set of models $M = \{M_1, M_2, \dots, M_m\}$ for presenting data D is given. Each model is presented by the prior probability $P(M_i)$ according to the probabilities shown in Table 5.2. $P(D|M_i)$ is the likelihood which is the probability of the observing data D knowing the model M_i . Bayes' rule states:

$$P(M_i|D) = \frac{P(D|M_i)P(M_i)}{P(D)} \propto P(D|M_i)P(M_i) \quad (5.4)$$

where $P(M_i|D)$ is the posterior probability for each roof type. $P(D)$ is the normalizing constant which does not depend on the model M , therefore, it is not considered in calculating $P(M_i|D)$ and an unnormalized density is preferred [120]. The best model \hat{M}_i with high posterior probability is then chosen from the entire set of possible solutions M .

$$\hat{M}_i = \arg \max_{M_i \in M} (P(M_i|D)) = \arg \max_{M_i \in M} P(D|M_i) \cdot P(M_i) \quad (5.5)$$

Let us consider D_i as the partial data of rectangle i . $P(D|M_i)$ is also shown as likelihood $\mathcal{L}(D_i|M_i)$ which is given by:

$$\mathcal{L}(D_i|M_i) \propto \exp\left(-\sqrt{\frac{1}{N} \sum_j \rho(od_j^2)}\right) \quad (5.6)$$

where N is the number of inner nDSM pixels of the rectangle and $\rho(od_j^2)$ is the Huber loss [168], which is computed by:

$$\rho(od_j^2) = \begin{cases} \frac{1}{2}od_j^2 & \text{for } |od_j| < T \\ T(|od_j| - \frac{1}{2}T), & \text{otherwise} \end{cases} \quad (5.7)$$

In this equation, od_j^2 is the shortest orthogonal distance (Euclidean distance) from point p_j to the surface of the 3D model defined by configuration M_i , and T is the threshold of this error, which is computed by fitting a plane to the part of the nDSM of the building and calculating the percentage of the points which have the closest distance to the surface of the 3D model. This threshold is equal to 1 m for sloped and flat roofs. Among the model set of M , the model with the maximum $P(M_i|D)$ is selected for further processing.

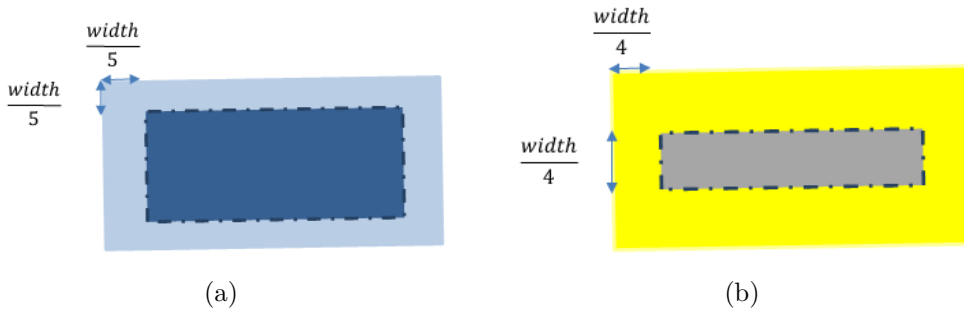


Figure 5.13: (a) Buffer of the roof eave height estimation. (b) Buffer of the roof ridge line height estimation

5.3.3 Initialization of Roof Model Parameters

As explained in Section 5.1, the number of roof model parameters for initialization depends on the roof model type. The most complex roof type in the library is the mansard roof, which has eleven parameters. The number of parameters is variable from a flat roof with four parameters to a mansard roof with eleven parameters. These parameters are either equal to zero or changed during the optimization, depending on the roof type. This section shows how the roof parameters are initialized. The position of the rectangle center (x_o and y_o) is calculated by averaging the four vertices of the rectangle. The *length* and *width* of rectangles are replaced by each other regarding orientation of the roof model. The orientation of the building which is similar to the orientation of the ridge line is calculated according to Equation 5.2.

The ridge line and eave heights are computed based on averaging the pixels within two buffers, each with a different width (Figure 5.13) around the main middle line (gray color) and rectangle border (light blue color). In the following, the principal of the roof height estimation, which is inspired by [169], is explained in detail.

Estimating the height of the roof eave: To estimate the height of the eave, an area close to the eave of the roof is selected by defining a buffer. The buffer width is equal to one-fifth of the rectangle width. The pixels of the nDSM within the buffer are then selected and are ranked by increasing heights. The average of ranked pixels in the range of $[0.3 \times N_{eave}, 0.7 \times N_{eave}]$ estimates the height of the eave where N_{eave} is the number of buffer pixels. Since the height of the boundary is changed gradually and there is no sharp building boundary in the nDSM with a specific height, selecting this range is safe enough to avoid the height of the ground and the height of the ridge line in the sloped roofs.

Estimating the height of the roof ridge line: The heights of ridge lines are estimated similar to the eave heights. Thus, an area close to the main middle line of the rectangle is selected by defining a buffer. The buffer width is equal to one-fourth of the rectangle width. The pixels of nDSM which belong to the buffer are then selected and ranked by increasing heights. The average of ranked pixels in the range of $[0.85 \times N_{ridge}, 0.95 \times N_{ridge}]$ estimates the height of the ridge line, where N_{ridge} is the number of pixels within the buffer. This range is admissible focusing on the highest pixels while avoiding the noisy pixels on the ridge lines and the pixels from the eave.

The initialization of latitudinal and longitudinal hip distances, such as $hipw_1$, $hipw_2$, $hipl_1$, and $hipl_2$, in Figure 5.2 is calculated based on the following geometrical rules:

- The longitudinal hip distance for a pyramid roof is equal to half of the minimum length of the rectangle,
- the longitudinal hip distance for a half-hip, hip and mansard roof are equal to the quarter of minimum length of the rectangle,
- the latitudinal hip distance for a pyramid roof, half-hip and hip is equal to half the width of the rectangle, and
- the latitudinal hip distance for a mansard roof is equal to quarter of the width of the rectangle.

These rules hold for Munich as most of the roofs are sloped ones.

5.4 3D Building Model Reconstruction

In this section, the complete parametric roof model is reconstructed. Firstly, the initial parameter obtained from the previous section is computed precisely by an optimization method to discover the best fitting model to the nDSM. In the end, the vertices of each rectangle intersect with each other according to some predefined rules.

5.4.1 Optimization

The optimization is performed on the primary model obtained from the initial parameters. The goal is improving the primary model to have the best fit to the height data and boundary of the building. There are two categories of initial parameters in 2D, such as $\mathcal{P} = \{x_o, y_o, orientation\}$ and $\mathcal{C} = \{length, width\}$, and in 3D, such as Z_{eave} , Z_{ridge} , $hipw_1$, $hipw_2$, $hipl_1$ and $hipl_2$. The optimization is performed for each

Parameters	Ranges	Unit	Step size
x_o	[-3, 3]	pixel	1
y_o	[-3, 3]	pixel	1
Z_{ridge}	[-2.4, 2.4]	meter	0.2
Z_{eave}	[-2.4, 2.4]	meter	0.2
$length$	[-5, 5]	pixel	1
$width$	[-5, 5]	pixel	1
$Orientation$	$[-\Delta\theta, +\Delta\theta]$	degree	1
$hipl_1$	$[-1, 0.35] \times initial(hipl_1)$	pixel	0.16
$hipl_2$	$[-1, 0.35] \times initial(hipl_1)$	pixel	0.16
$hipd_1$	$width/2$	pixel	0
$hipd_2$	$width/2$	pixel	0

Table 5.3: Predefined parameters and their corresponding ranges, units and step sizes

category independently. Several models are generated by changing the parameters in the predefined ranges. Ten models can be generated, for example, for changing the orientation in $-10, -8, \dots, 8, 10$ degrees.

The 2D and 3D optimization is based on a brute-force algorithm which discovers the best combination among all combination of the parameters. The range of refinement of the model parameters is limited, since the initialization is acceptable in most of the buildings. On the other hand, using the nDSM for model fitting results in defining the range of parameters in discrete numbers. Searching the optimum parameters in the discrete ranges using the complex cost function might be stopped by being trapping in local minima. Searching for the best among all possible combinations motivated us to use the brute-force search algorithm to find the global optimum of parameters.

Table 5.3 represents the range of each parameter and their corresponding step sizes. These ranges are obtained based on empirical trials and the quality of initial parameters. The same ranges of parameters have been defined for all four areas. The parameters of longitudinal hip distances $hipl_{1\&2}$ and $hipd_{1\&2}$ are changed depending on the parameters of $length$ and $width$, while the other parameters are changed independently, as shown in Table 5.3.

ranges of parameters [htbp]

The 2D optimization improves the 2D parameters so that they have the best fit to the boundary of the building. The 2D improvement in Section 5.2 was for building rectangles in case their sides and orientation were far from the boundary of the building mask. However, the interaction between the rectangles in one building

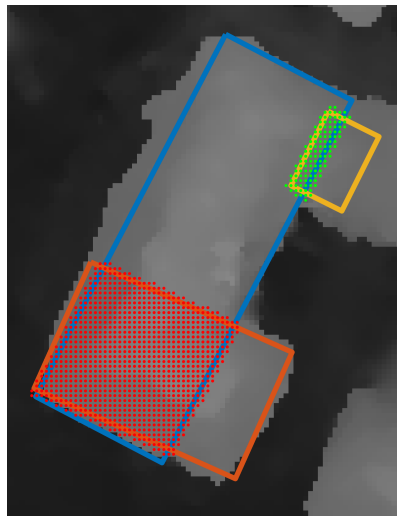


Figure 5.14: Overlapping (red dots) and neighboring (green dots) parts of rectangles in one building block

block has not been considered. Therefore, 2D optimization with a tight range of the 2D parameters is performed to find a combination among all those of 2D parameters so that its model not only has a minimum distance to the boundary, but is also oriented in relation to neighboring models. The best 2D models are then used as the input of the 3D optimization. A brute-force search is used to select the best 3D model among all the models which are generated by changing the 3D parameters in their related ranges. The ridge and eave heights of neighboring models for one block of a building then affect the selection of the final 3D model.

The metric of 2D optimization is the PoLiS distance [136] between the boundary of the building (obtained from the RANSAC fitting line to the boundary of building mask and canny edges of the PAN image) and model vertices. The metric of 3D optimization is the normal distance between the 3D points and the roof model surface, which is calculated by the Equations 5.6 and 5.7.

5.4.1.1 Finding the Interaction of the Rectangles in One Building Block

The neighboring and overlapping rectangles are found to verify the interaction between models (rectangles) of a building block. In order to distinguish the two types of overlaps, the overlapping rectangles share a significant area which can be represented by an intersection roof model type, such as *L-shape* or *T-shape*. The neighboring rectangles, however, share a small area which is not significant and cannot present a connecting roof model type. Figure 5.14 shows the overlapping and neighboring rectangles. To find the overlap of two overlapping rectangles:

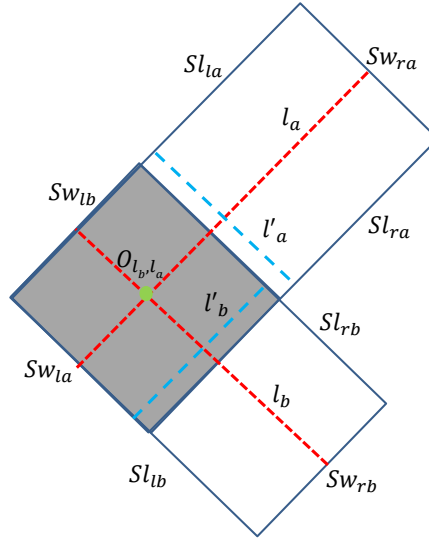


Figure 5.15: The parameters of the rectangle to find the overlapping sides which is shown in gray

1. find the middle lines of two rectangles and their intersection. In Figure 5.15, two rectangles a and b , where the middle lines are l_a, l'_a, l_b and l'_b , are presented. The intersection between the middle lines are notated by O ,
2. find the left and right sides of the rectangles parallel and perpendicular to l_a, l'_a, l_b and l'_b which are notated with $Sl_{ra}, Sl_{la}, Sw_{ra}$ and Sw_{la} for the rectangle a and notated by $Sl_{rb}, Sl_{lb}, Sw_{rb}$ and Sw_{lb} for the rectangle b ,
3. find the overlap of the rectangles by intersecting their building mask ($mask_{overlap}$)
4. if O_{l_a, l_b} and $mask_{overlap}(O_{l_a, l_b})$ exist, then accept rectangles have overlap,
5. find the overlapping side of rectangles, if the overlap exists. Exemplarily, for rectangle a and middle lines l_a , if $distance(Sw_{la}, O_{l_a, l_b}) < distance(Sw_{ra}, O_{l_a, l_b})$, then the overlapping side is located on the left side of the rectangle notated,
6. verify the fifth item for rectangle b and other middle lines to find the overlapping part.

Figure 5.15 shows the example of finding overlapping parts of two rectangles according to the items mentioned above.

In order to discover the neighboring rectangles, the fourth item is rejected as two rectangles have only a small overlap (shown in green dots in Figure 5.14) and intersection of the ridge lines is outside the overlapping part. Those sides of rectangles which overlap are detected to find the overlapping side.

5.4.1.2 2D Optimization

Generating all possible models for one rectangle through exhaustive search and selection of the best one among them are represented by the algorithms 2 and 3 in Appendix D, respectively.

Let the vector of parameters be $\Psi_{1 \times 11} = \{x_o, y_o, z_{eave}, z_{ridge}, length, width, orientation, hipl_1, hipl_2, hipd_1, hipd_2\}$, and the goal of 2D optimization step is finding the best combination of the parameters $\{x_o, y_o, length, width, orientation\}$. Algorithm 2 illustrates optimization through exhaustive search in which all possible 2D building models are generated by changing the initial 2D parameters of $\Psi_{1 \times 11}$ ($\{x_o, y_o, length, width, orientation\}$) within the predefined ranges as shown in Table 5.3. The parameter combinations are then derived and their model costs are calculated based on the PoLiS distance between the model vertices and the reference building boundaries. The best 2D building model with the minimum cost is then selected by Algorithm 3 for each rectangle. For the building blocks composing of more than one rectangle, the rectangles are ordered based on their lengths. After selecting the best model of the first rectangle, the best model for the second rectangle is selected by investigating its neighborhood relations with the other rectangles based on the method explained in Section 5.4.1.1. If the second rectangle, for example, is adjacent to the first rectangle, the differences between the orientation of the first rectangle and the orientations of all possible models for the second rectangle are calculated. If these differences are within the angle threshold of $[0, 90, 180, 270, 360] \pm (thr_\theta = 15)$, the 2D model with minimum cost and the minimum orientation difference is selected as the best 2D model for the second rectangle; otherwise, the model with the minimum cost value is selected as the final model for the second rectangle. This method is used for selecting the best models for all rectangles in the building block.

5.4.1.3 3D Optimization

The result of 2D optimization is used as input for 3D optimization. In the vector Ψ_{3D} , 3D parameters are variable and are optimized in combination with 2D parameters. The 3D parameters consist of $\{z_{eave}, z_{ridge\ line}, hipl_1\}$.

The initial value of the longitudinal and latitudinal hip distances are as follows:

- The initial value for the hip roof is $hipl_2 = hipl_1 = length/3$, $hipl_2 = 0$ and $hipl_1 = length/3$ for half hip roofs, and $hipl_2 = hipl_1 = 0$ for the gable roof.
- $hipd_1$ and $hipd_2$ are equal to the half the rectangle width for gable, half-hip and hip roofs.
- The initial value of longitudinal hip distance for mansard roofs is $hipl_2 = hipl_1 = length/3$ and $hipd_2 = hipd_1 = width/3$ for latitudinal hip distance.

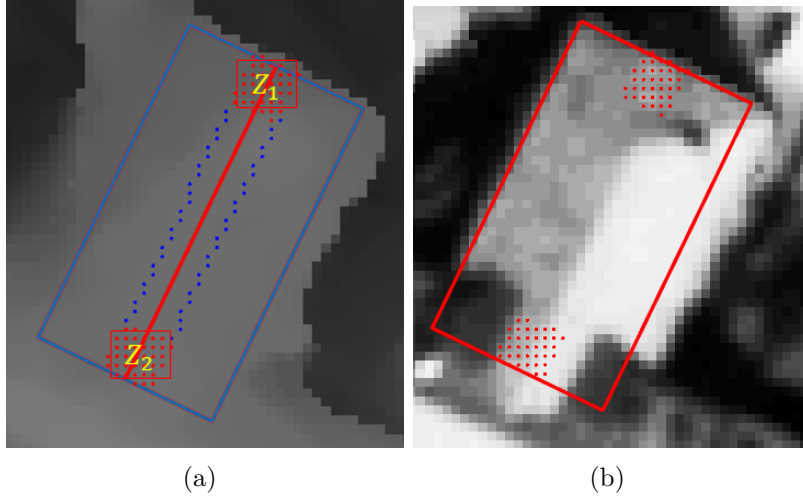


Figure 5.16: Buffer of the hip side determination, (a) projected on the DSM, (b) projected on the image

- The initial value for a pyramid roof is $hipl_2 = hipl_1 = length/2$ and $hipd_2 = hipd_1 = width/2$.

The other important parameter for fitting the 3D model to the nDSM is the determination of the hip part in the half-hip roof. Since the roof type classification only determines the roof types without considering their geometrical structures, the hip part of half-hip roof is an unknown parameter. This parameter is determined based on the comparison of the height computed in the small buffer at the end of the ridge line (shown by the red points in Figure 5.16). Therefore, the side with smaller height value is selected as a hip_{side} .

$$hip_{side}(rectangle, nDSM) = \begin{cases} [0, 1], & \text{if } Z_2 \geq Z_1; \\ [1, 0], & \text{if } Z_1 \geq Z_2. \end{cases}$$

The pseudocodes of finding the best 3D parameters are shown in the Algorithm 4 and Algorithm 5 in Appendix D, respectively. Algorithm 4 shows the optimization algorithm based on an exhaustive search which is used for generating all possible 3D building models by changing the initial value of the 3D parameters ($\Psi_{3D} = \{z_{eave}, z_{ridge}, hipl_1\}$) in ranges defined in Table 5.3. The cost values of each 3D model are calculated based on the RMSE of orthogonal Huber distances between the 3D point clouds obtained from nDSM and the 3D models. Algorithm 5 selects the best 3D models among all possible 3D building models generated by the previous algorithm.

Similar to the 2D optimization, in a building block containing multiple rectangles, the relationships between the neighboring rectangles affect the model selection.

By contrast, in the selection of the best 3D building model, the height differences between ridge and eave lines of the neighboring rectangles are considered instead of orientation. The best 3D model of the first rectangle is initially selected based on the minimum cost value. If the second model, for example, is neighbor to the first model, the differences between the ridge/eave height values of all possible 3D models of the second rectangle and best 3D model of the first rectangle are calculated. If these differences are in a height threshold ($thr_h = 1\text{ m}$) and in a cost threshold ($thr_{cost} = 1.5\text{ m}$), the model with the minimum cost value and minimum height differences is selected as the final model for the second rectangle; otherwise, the model with the minimum cost value is selected. This method is performed for all rectangles in a building block.

5.4.2 Intersection and Reconstruction

As Figure 5.2 shows, the optimum parameters of the model are used to compute the vertices of the 3D model. The models for the parts created are intersected with each other to reconstruct the 3D building model for one building block including several parts. Different connecting roof types, such as *L-shape*, *T-shape* and *Plus-shape*, can be further created. Depending on the type of interaction between the models (the degree of overlap), the roof type of neighboring models, and the minimum cost value, the connecting roof type is chosen among the connecting roof types mentioned previously. Neighboring models which have significant overlap create different connecting roof models. Figure 5.17 shows the different models and their related connecting roof models. The intersection points of two basic mansard models which overlap each other are computed to create several connecting roof models, as Figure 5.18 shows. For example, $\{a_{5,6,9,1,5}, a_{5,11,20,6,5}, a_{20,17,9,6,20}, a_{20,11,12,18,20}\}$ are important intersection points for generating an *L-shape*. Mansard roof models, as a complete roof type including eleven parameters, are extensible to other roof models defined in the library with fewer parameters. After creating different connecting roof models, the one with the least cost value (Equation 5.7) is selected. If the cost values are close, *L-shape* is selected which is simpler by fewer intersection points.

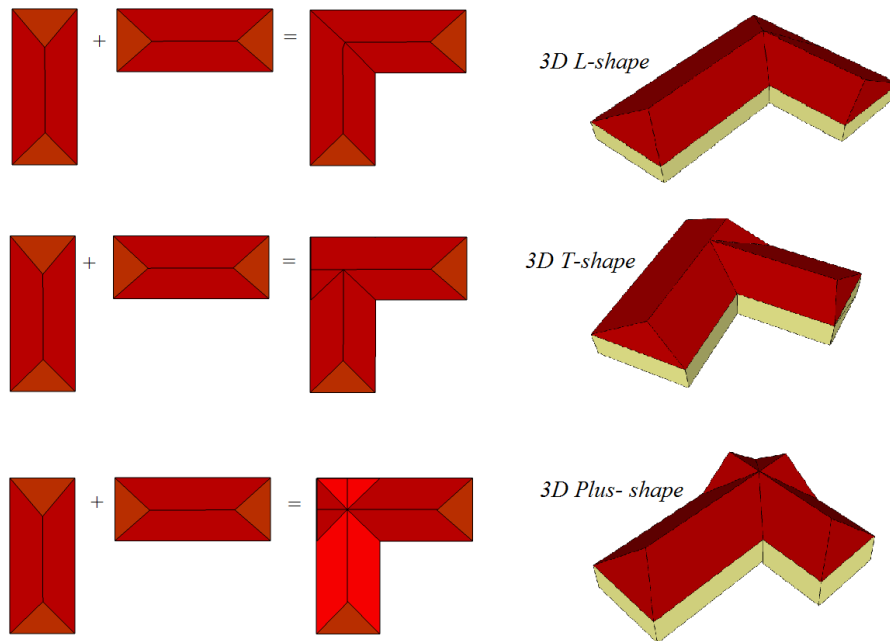


Figure 5.17: Connecting roof types

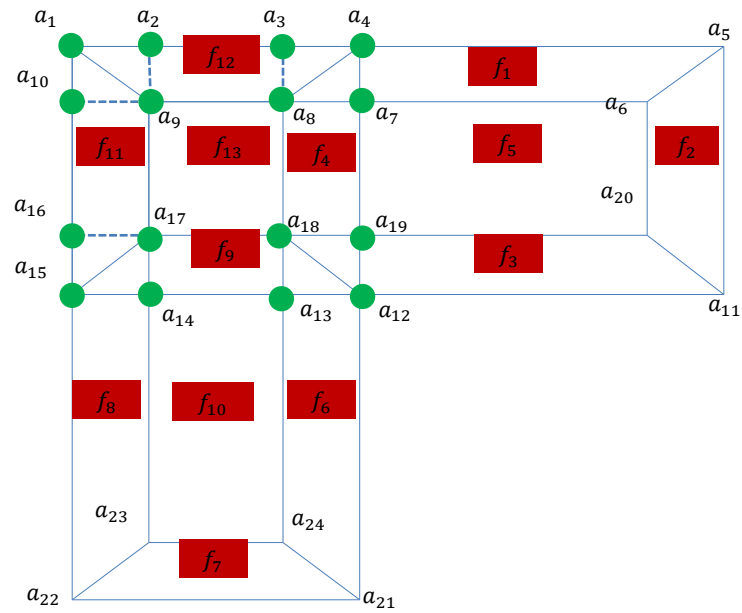
5.5 Result and Discussion

In this section, we provide qualitative and quantitative evaluations of the resulting 3D building models of four different areas in Munich city using our proposed approach together with the evaluations of the data-driven steps, including outline extraction and decomposition, height discontinuity detection, roof type classification and ridge line orientation identification.

5.5.1 Qualitative Evaluation

Visualizations of 3D building models and their proximities to the building structures in reality are major criteria for evaluating 3D model reconstruction approaches in technical applications. Figures 5.19, 5.21, 5.23 and 5.25 show the results of the data-driven steps of our proposed approach, where the results of the outline extraction and decomposition, height discontinuity detection, roof type classification and ridge line orientation identification are illustrated in the sub-figures (a)–(d), respectively. In addition, Figures 5.20, 5.22, 5.24 and 5.26 represent the reconstructed 3D building models. As shown in these figures, the proposed algorithm can reconstruct the high-rise buildings (more than 3 m height) and the buildings with average area larger than 75 m² (300 pixels with a resolution of 0.5 m). The low-rise and small structures, such as garages, were discarded due to the thresholds used in the building detection step.

5. 3D Building Model Reconstruction



<i>L-shape</i>		<i>T-shape</i>		<i>Plus-shape</i>	
$M_1: f_{1,2,3,4,13,12,5}$ $M_2: f_{13,6,7,8,11,9}$		$M_1: f_{1,2,3,4,5}$ $M_2: f_{6,7,8,10,9,11,13,12}$		$M_1: f_{12,1,2,3,9,11,5,4,13}$ $M_2: f_{7,10,9,13,12,8,11,6,4}$	
$a_{5,6,9,1,5}$ $a_{5,11,20,6,5}$ $a_{20,17,9,6,20}$ $a_{20,11,12,18,20}$	$a_{12,21,24,18,12}$ $a_{8,24,23,9,8}$ $a_{9,23,22,1,9}$ $a_{24,21,22,23,24}$	$a_{4,5,6,7,8,4}$ $a_{5,11,20,6,5}$ $a_{11,12,18,19,20,11}$ $a_{6,20,19,18,8,6}$	$a_{12,21,24,18,12}$ $a_{21,22,23,24,21}$ $a_{1,9,23,22,1}$ $a_{1,4,8,9,1}$ $a_{8,24,23,9,8}$	$a_{5,6,10,1,5}$ $a_{6,20,16,10,6}$ $a_{20,11,15,16,20}$ $a_{5,11,20,6,5}$	$a_{4,21,24,3,4}$ $a_{3,24,23,2,3}$ $a_{23,22,1,2,23}$ $a_{24,21,22,23,24}$
X		$a_{4,5,6,7,8,4}$ $a_{5,11,20,6,5}$ $a_{11,12,18,19,20,11}$ $a_{6,20,19,18,8,6}$	$a_{23,22,1,2,23}$ $a_{24,3,4,21,24}$ $a_{2,3,24,23,2}$ $a_{23,24,21,22,23}$	X	
		$a_{5,6,9,1,5}$ $a_{5,11,20,6,5}$ $a_{20,11,15,17,20}$ $a_{9,17,15,1,9}$ $a_{9,17,20,6,9}$	$a_{22,15,17,23,22}$ $a_{18,24,23,17,18}$ $a_{24,21,22,23,24}$ $a_{18,12,21,24,18}$		
		$a_{5,6,10,1,5}$ $a_{5,11,20,6,5}$ $a_{20,11,15,16,20}$ $a_{6,20,16,10,6}$	$a_{22,15,17,23,22}$ $a_{18,24,23,17,18}$ $a_{24,21,22,23,24}$ $a_{18,12,21,24,18}$		

Figure 5.18: Possible connecting of roof types by different combinations of the roof vertices and intersection points (green points) of the two mansard roofs

A total number of 208 buildings, including single, compound and complex buildings with inner yards, multiple orientation or multiple roof types were reconstructed. According to the results, the 3D building model reconstructions were satisfactory, except for a few buildings.

Figures 5.27 and 5.28 illustrate examples of the reconstructed building models. More precisely, the height discontinuities were detected correctly in B0, B3 and B8. Moreover, the 2D quadrilateral footprints demonstrate the successful intersection of neighboring rectangles to fit the building shapes. Moreover, B7 shows the strength of our approach in reconstructing buildings with multiple roof types, such as the half-hip and pyramid roofs in this example. Further examples of buildings with multiple roof types can be seen in B5 and B8 with flat-slopped and hip-mansard roofs, respectively. In addition, B4 shows that our proposed approach can intersect both perpendicular and parallel rectangles. In addition to its strengths, our proposed approach suffers from a few drawbacks, which we investigate and discuss in the following section.

5.5.1.1 Challenges and Difficulties

There are some buildings in which the height discontinuities are under- or over-detected due to the poor quality of the DSM data. The over-detection of height discontinuities could cause the rectangles to be over-split, as depicted by a green ellipse in Figure 5.29. In order to overcome this drawback, one can fuse the nDSM data and the image information.

Another failure case was that a number of building footprints were poorly positioned (exemplified in Figure 5.30), which could be due to the limited performance of the 2D optimization algorithm, the existence of objects, such as trees and cars, neighboring the buildings, or the jagged building outlines caused by the orthorectification of the PAN images.

Our approach assumes rectilinear forms as the basic building shapes similar to the typical forms of the building footprints in urban areas. However, there are buildings with nonrectilinear shapes, such as the one depicted by a green ellipse in Figure 5.30. These buildings can be detected by calculating the rectangularity degrees of 2D footprints as the ratio between the inner area of the rectangle and the area of the MBR and minimum bounding circle [30]. If this ratio is close to one for MBR, the rectangle model is considered, and if it is close to one for the minimum bounding circle, the circular model is considered for further 2D optimizations. Moreover, some remarkable roof types, such as conical, shell, semi-ellipsoid, elliptic and spherical, which have been not predefined in the library are reconstructed based on existing roof types in the library. In Figure 5.31, for example, the semi-ellipsoid-shaped roof was reconstructed by some flat roof shapes at different height levels, since the

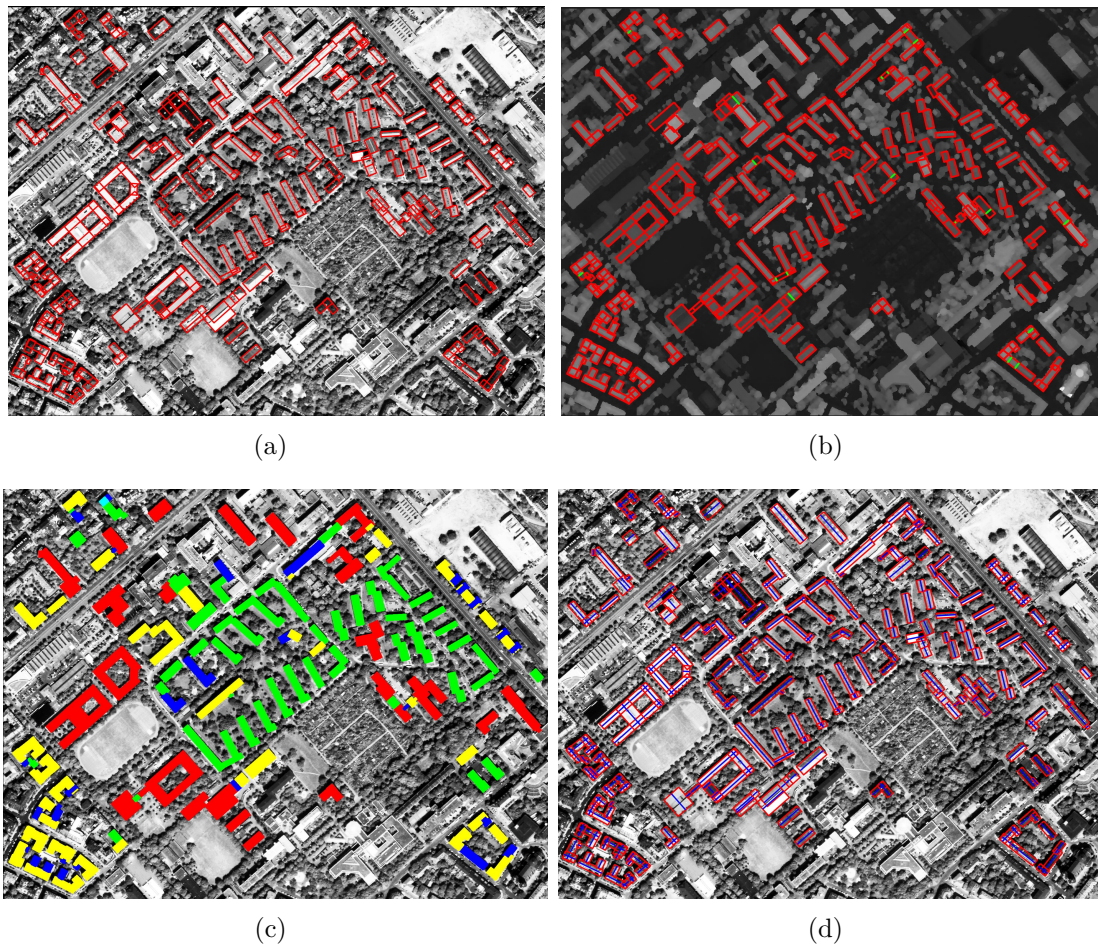
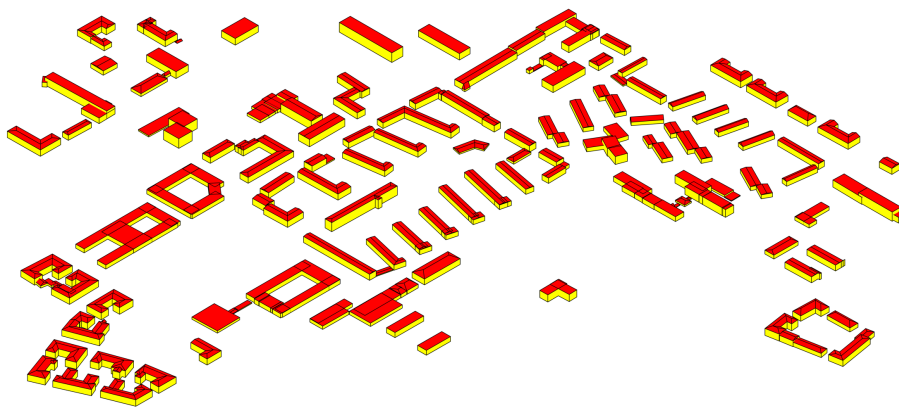


Figure 5.19: Illustration of the data-driven steps of 3D building model reconstruction for the first area (Area-1): (a) Decomposed 2D building outline into rectilinear shape, (b) detected height discontinuities (green lines), (c) classified roof types (red: flat, green: gable, blue: half-hip, yellow: hip, magenta: pyramid, cyan: mansard) and (d) extracted ridge lines (blue lines).



(a)



(b)

Figure 5.20: Illustration of 3D building reconstruction for Area-1: (a) 3D building models on the PAN image, (b) 3D building models

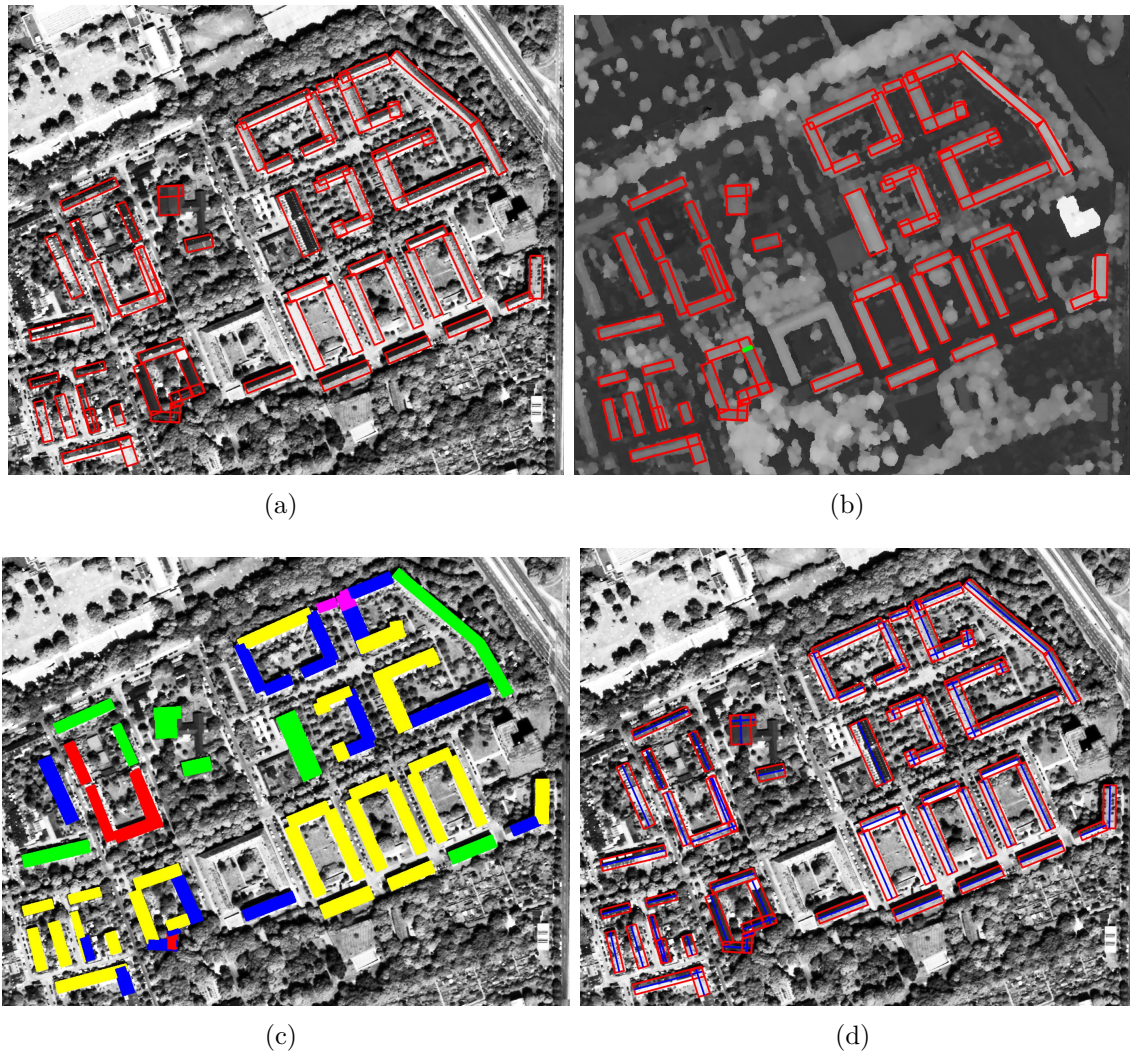
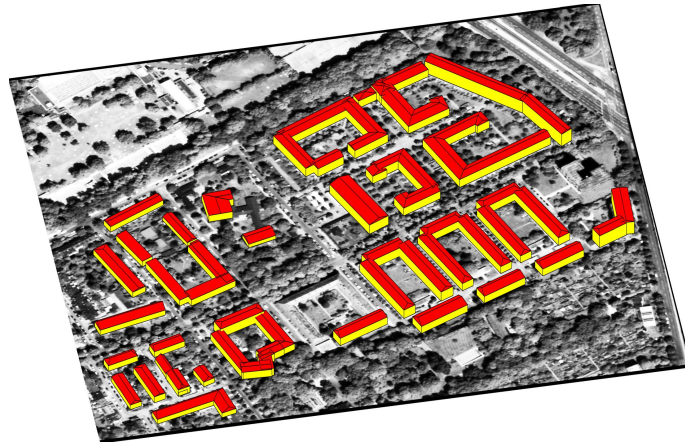
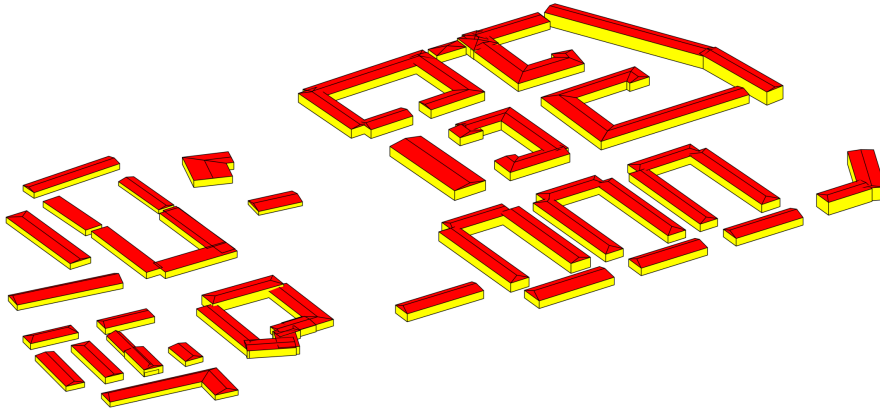


Figure 5.21: Illustration of the data-driven steps of 3D building model reconstruction for the second area (Area-2): (a) Decomposed 2D building outline into rectilinear shape, (b) detected height discontinuities (green lines), (c) classified roof types (red: flat, green: gable, blue: half-hip, yellow: hip, magenta: pyramid, cyan: mansard), (d) extracted ridge lines (blue lines).



(a)



(b)

Figure 5.22: Illustration of 3D building reconstruction for Area-2: (a) 3D building models on the PAN image, (b) 3D building models

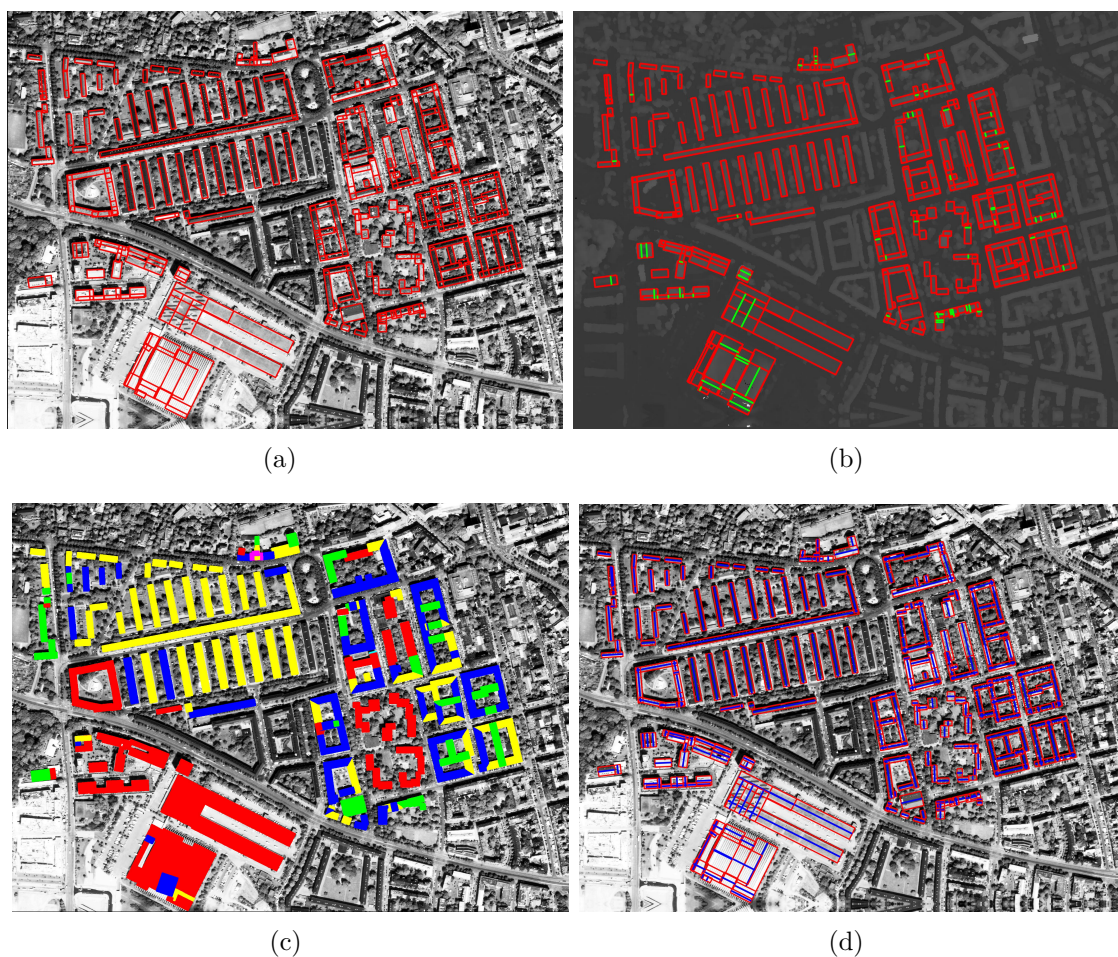
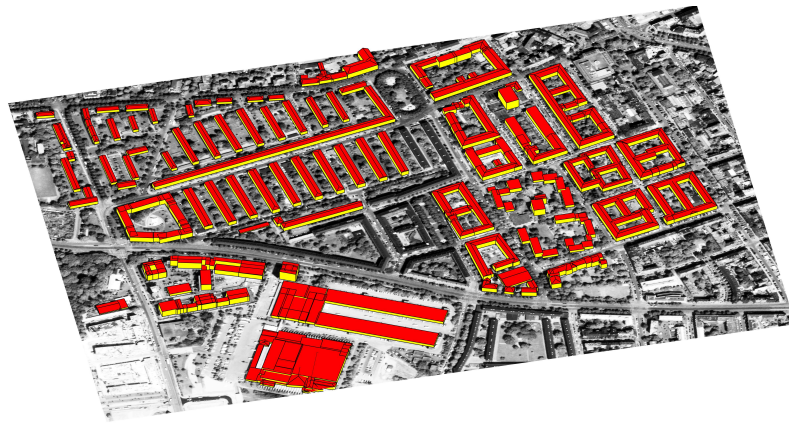
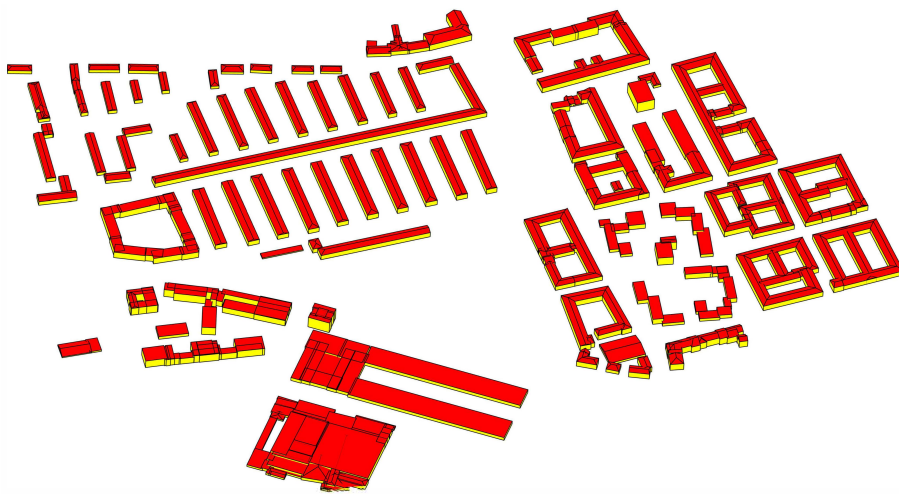


Figure 5.23: Illustration of the data-driven steps of 3D building model reconstruction for the third area (Area-3): (a) Decomposed 2D building outline into rectilinear shape, (b) detected height discontinuities (green lines), (c) classified roof types (red: flat, green: gable, blue: half-hip, yellow: hip, magenta: pyramid, cyan: mansard), and (d) extracted ridge lines (blue lines)



(a)



(b)

Figure 5.24: Illustration of 3D building reconstruction for Area-3: (a) 3D building models on the PAN image, (b) 3D building models

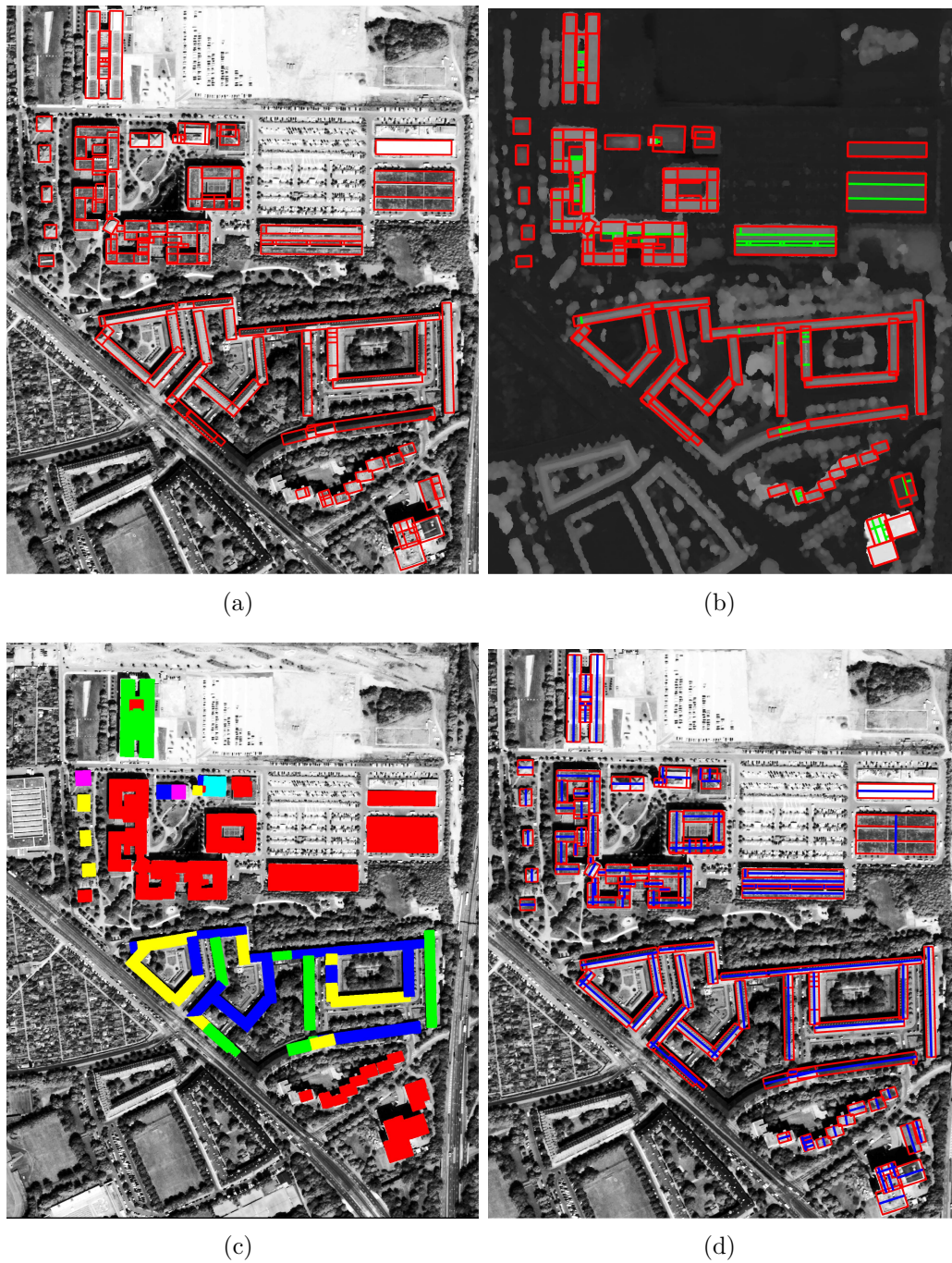
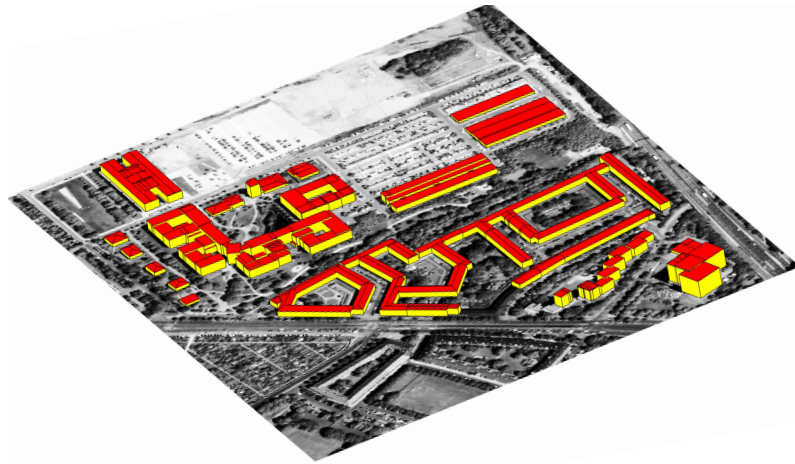
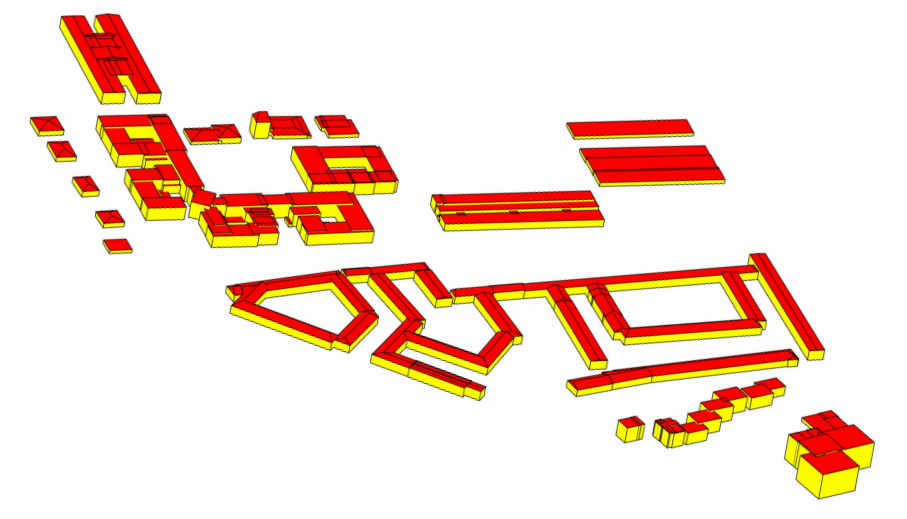


Figure 5.25: Illustration of the data-driven steps of 3D building model reconstruction for the fourth area (Area-4): (a) Decomposed 2D building outline into rectilinear shape, (b) detected height discontinuities (green lines), (c) classified roof types (red: flat, green: gable, blue: half-hip, yellow: hip, magenta: pyramid, cyan: mansard), (d) extracted ridge lines (blue lines).



(a)



(b)

Figure 5.26: Illustration of 3D building reconstruction for Area-4: (a) 3D building models on the PAN image, (b) 3D building models

5. 3D Building Model Reconstruction

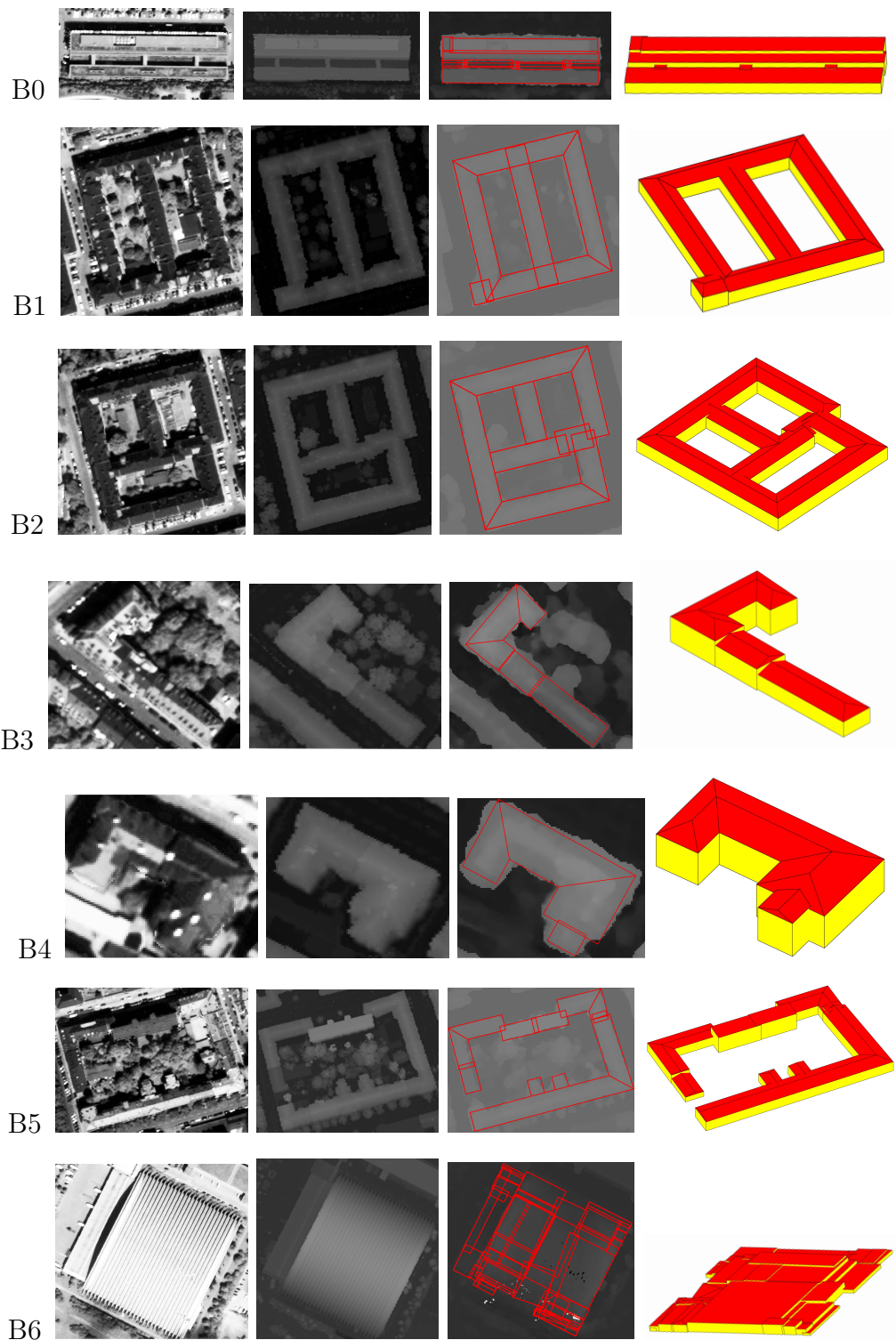


Figure 5.27: Different examples of 3D building models. PAN image (1st column), LiDAR data (2nd column), rectangles (3rd column) and 3D building model (4th column)

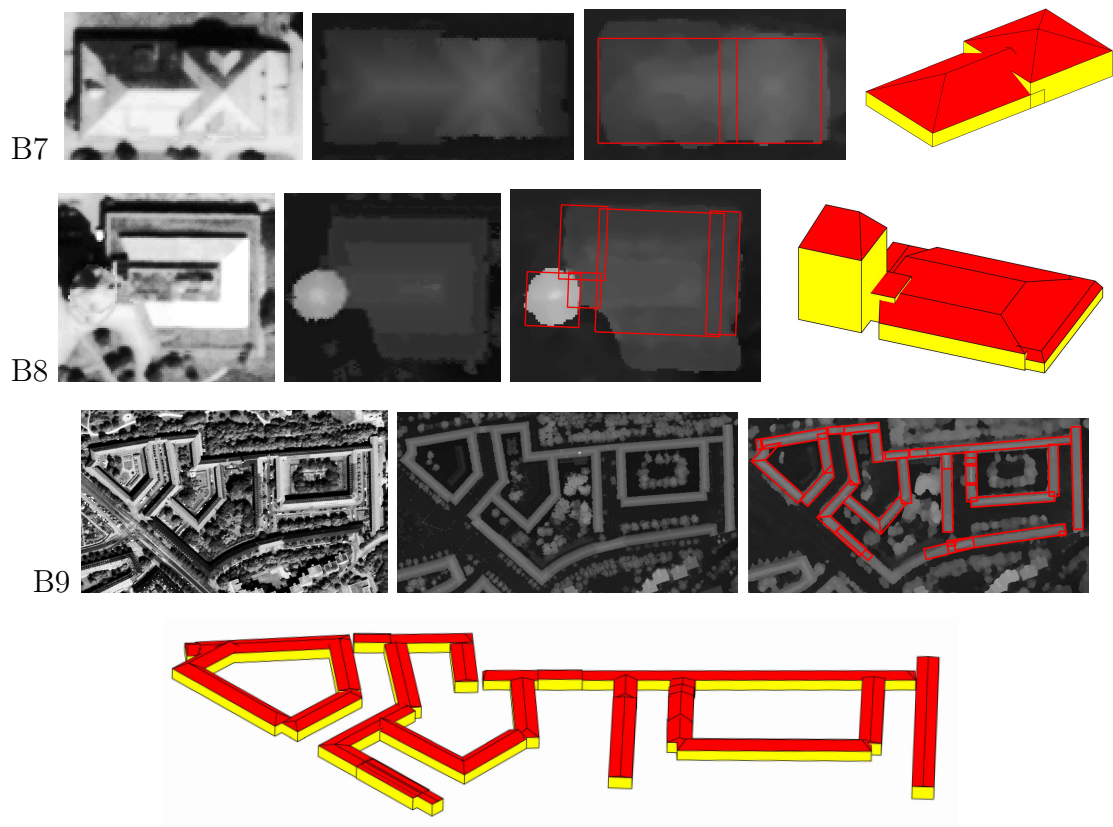


Figure 5.28: Different examples of 3D building models. PAN image (1st column), LiDAR data (2nd column), rectangles (3rd column) and 3D building model (4th column)

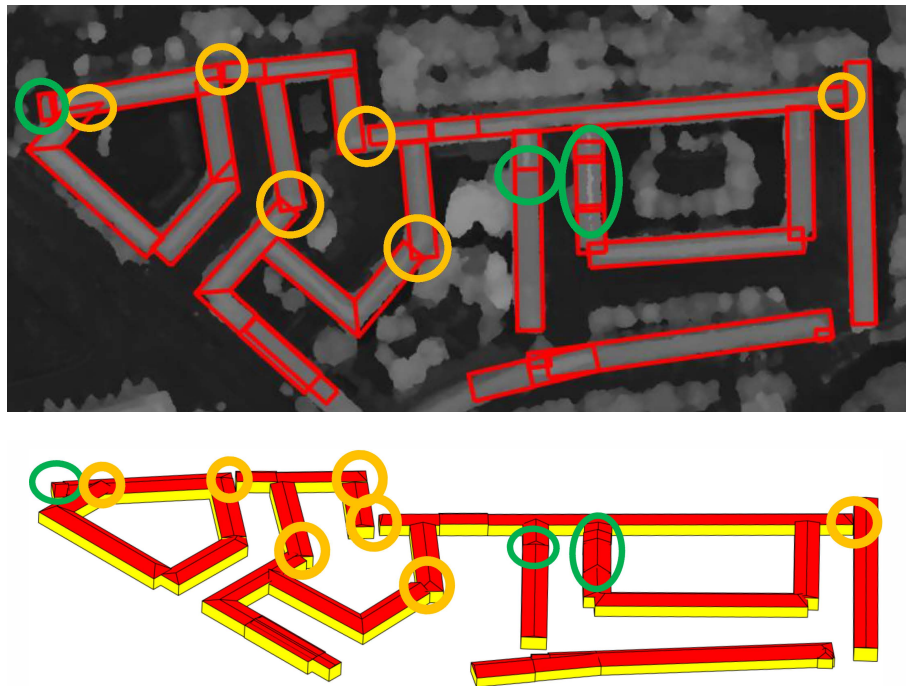


Figure 5.29: Over-detection of height discontinuities highlighted with green ellipses; Up: the error locations on the nDSM, bottom: the error locations on the 3D model

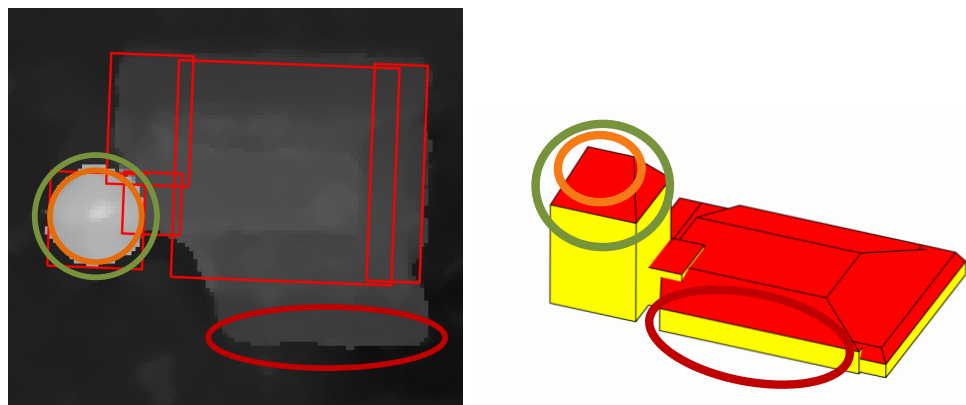


Figure 5.30: Inaccurate building footprint positions (red ellipse). Circular footprint (green circle) and conical roof shape (orange circle); left side: the error locations on the nDSM, right side: the error locations on the 3D model

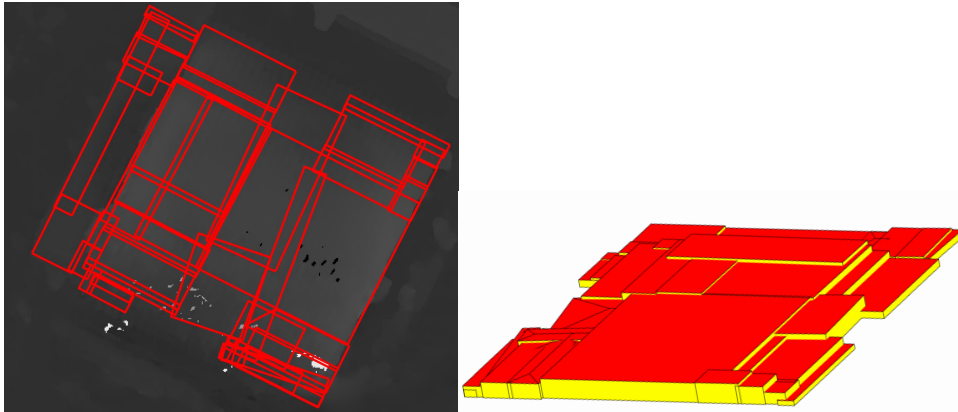


Figure 5.31: Semi-ellipsoid roof types; left side: the error locations on the nDSM, right side: the error locations on the 3D model

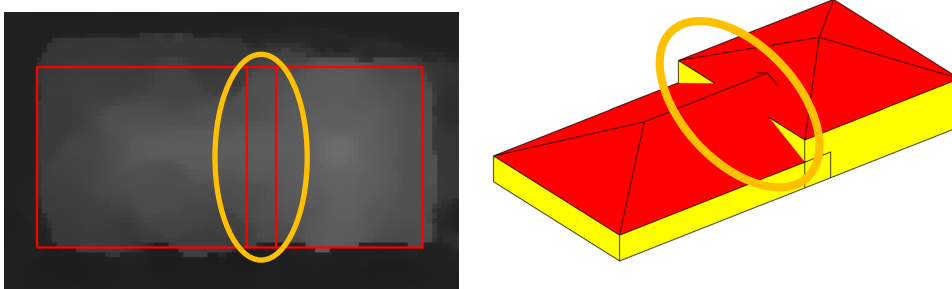


Figure 5.32: Deficiencies of intersection and merging; left side: the error locations on the nDSM, right side: the error locations on the 3D model

heights change gradually. Furthermore, the cylindrical tower with a conical roof in Figure 5.30 (orange circle) was reconstructed as a cubic structure with a hip roof. These limitations could be overcome by including structures with curved shapes in the building model libraries.

Our proposed approach fails to intersect and merge the neighboring rectangles when their overlap is not large enough, for example, the area which is depicted by an orange circle in Figure 5.29. Moreover, the intersection can suffer deficiencies if the intersection rule is not defined for different roof types. The intersection of the pyramid and half-hip roofs, for example, failed in forming a *T-shape* structure in Figure 5.32. These types of failures could be remedied by defining appropriate intersection rules. In addition, the poor quality of DSM data can cause intersection failures, such as that depicted by an orange ellipsoid in Figure 5.33, where the *L-shape* structure could not be reconstructed.

After fitting 3D models to each rectangle, the models are assembled to create the full model of each building block. There are some cases where the assembling process

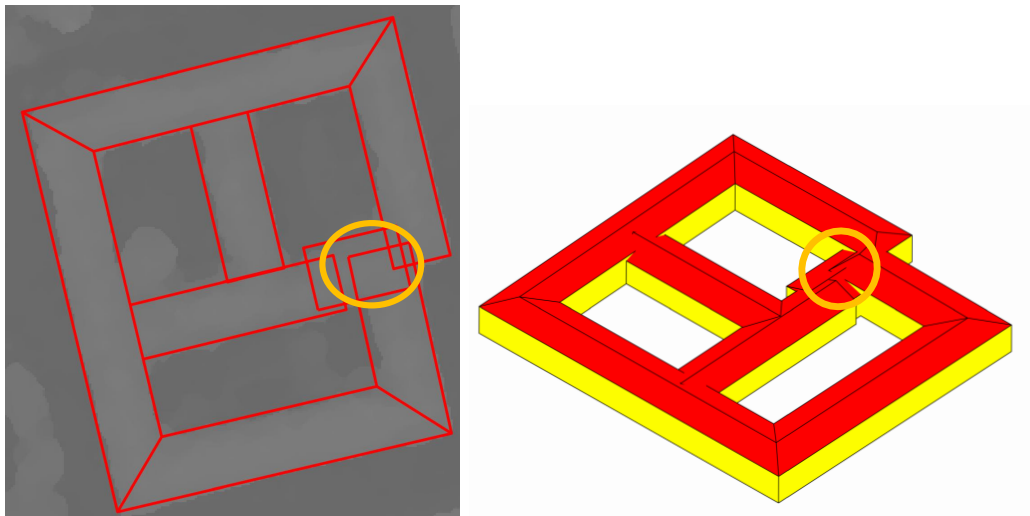


Figure 5.33: Deficiencies of intersection and merging; left side: the error locations on the nDSM, right side: the error locations on the right side

introduces artifacts to the final model. In Figure 5.29, for example, assembling the sequence of rectangles depicted by a green ellipsoid resulted in artifacts, as the ridge lines could not be aligned appropriately. This misalignment could be due to the lack of 2D alignment rule for aligning the neighboring rectangles during the reconstruction process and missing the initial ridge lines by splitting the rectangles. These artifacts can be reduced by applying some postprocessing to the final building models.

According to the results, some parts of buildings could be missed due to information loss during different steps of the reconstruction procedure, such as building detection or building decomposition into rectangles. In Figure 5.34, for example, a rectangle is missing, which was probably removed in the decomposition and selection steps, in the area depicted by a gray ellipsoid. Another failure case because of decomposition is shown in Figure 5.31, where the building is divided into too many rectangles, while the building seems to be covered only by one rectangle.

5.5.1.2 Comparing Height Maps and Profiles to LiDAR Data

In this section, we qualitatively evaluate the reconstructed 3D building models by comparing the height map (nDSM) of the building models (Figure 5.35) to the nDSM of LiDAR and satellite data. Consequently, a profile line is computed for an area on the model-nDSM, LiDAR-nDSM and satellite-nDSM data, as illustrated in Figure 5.36. As the height profiles in Figure 5.36 demonstrate, the model-nDSM is closer to the LiDAR-nDSM compared to the satellite-nDSM. Moreover, there are shifts of about 1 m between the model- and LiDAR-nDSMs in some profile peaks, which are acceptable as they are within the standard deviation (σ) of the elevation

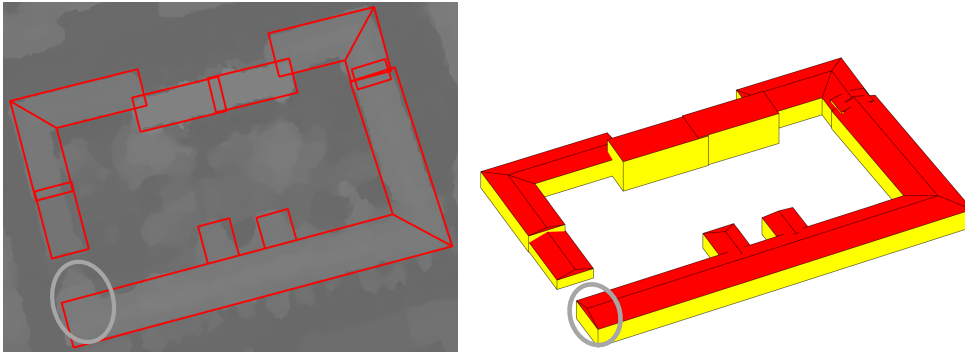


Figure 5.34: Missing the building part; left side: the error locations on the nDSM, right side: the error locations on the right side

errors of the satellite-nDSM. In Figure 5.36 (a), the LiDAR-nDSM profile shows the asymmetric roofs, while the building roofs are symmetric in reality. Thus, the LiDAR-nDSM is not perfectly accurate as reference data.

Considering the LiDAR-nDSM as the reference data, we compute the height difference between the model-nDSM and the LiDAR-nDSM of the areas studied and represent them as error maps and histograms in Figure 5.37. The buildings depicted in black were not considered for the 3D model reconstruction as they are located on image borders. According to the error maps and histograms in Figure 5.37, most of the errors alternate between -2 and 2 meters, which represents 2σ of the mean value (μ) with a 95% confidence interval, as the error histograms show. The areas with relatively large errors usually belong to the complex buildings, such as the ones in the third and fourth rows of Figure 5.37. Moreover, the areas with an error larger than 3 meters are depicted in dark blue, which usually belongs to non-building areas that were reconstructed as part of buildings. This occurs when reliable 2D footprints are not available or an inappropriate roof type is assigned to the buildings during roof type classification. Additionally, the building areas which could not be reconstructed by the 3D modeling approach are depicted in dark red. These missing parts belong to the buildings with undefined roof shapes or the structures on the rooftops. In Figure 5.37, the error histograms follow normal distribution, where their means and standard deviations are depicted in the figures. The long tails of the distribution could be due to the existence of outliers around the building outlines and fitting inappropriate models to the buildings.

5.5.2 Quantitative Evaluation

The quality of the reconstructed 3D models depends on the performances of the data-driven steps. Therefore, in this section, we compare the results of each data-driven step with the ground truth data. Furthermore, we evaluate the quality of the resulting 3D building models by comparing them to nDSM-based LiDAR data

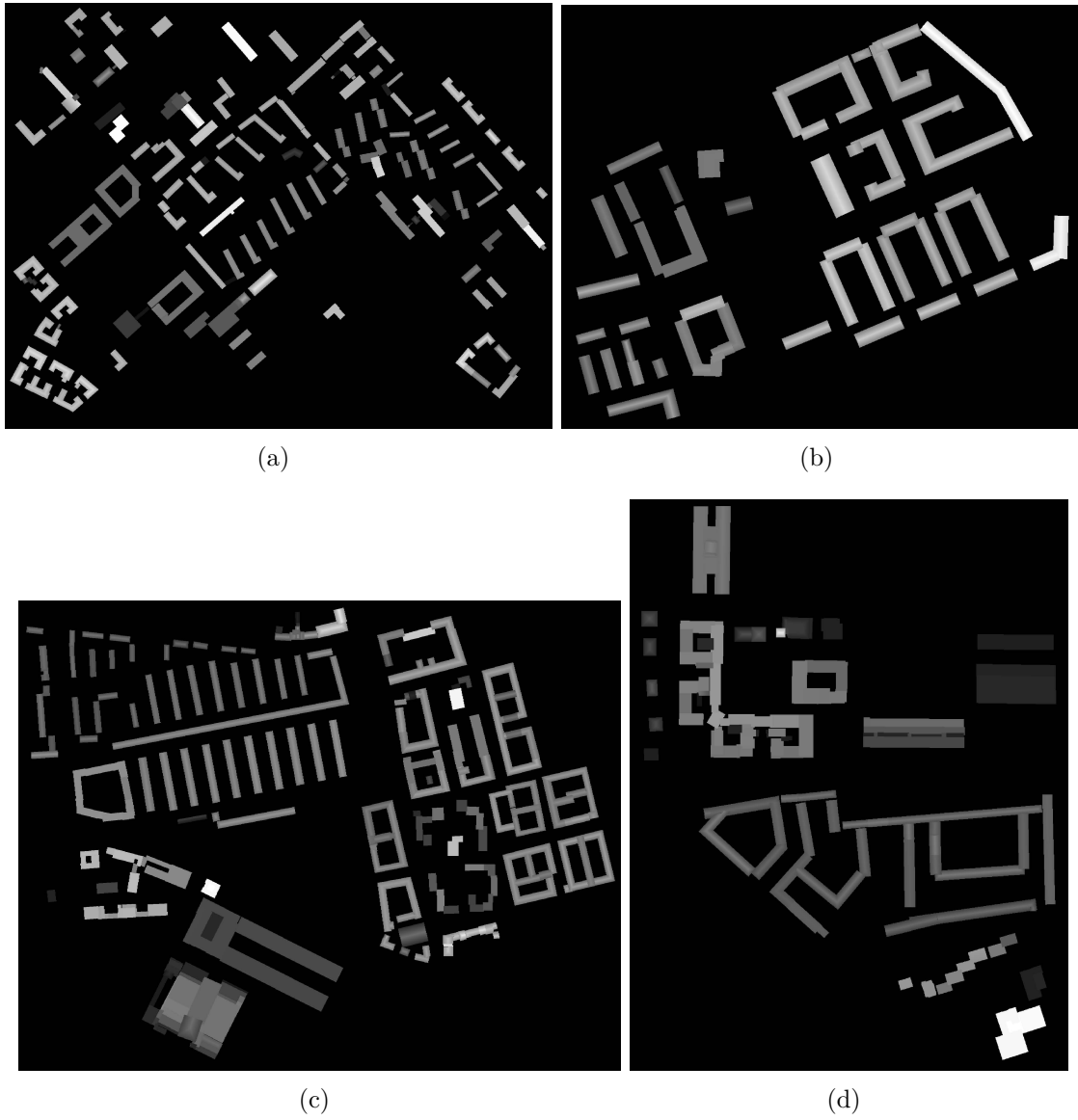
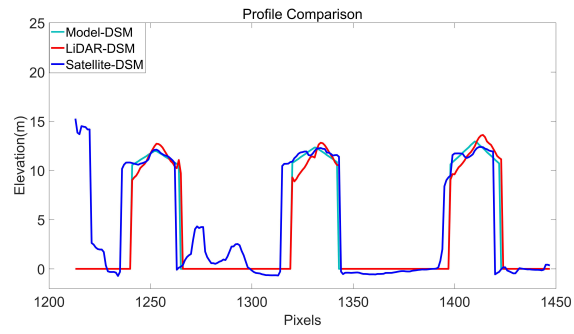
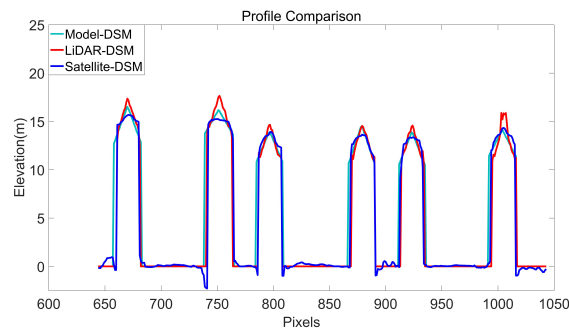


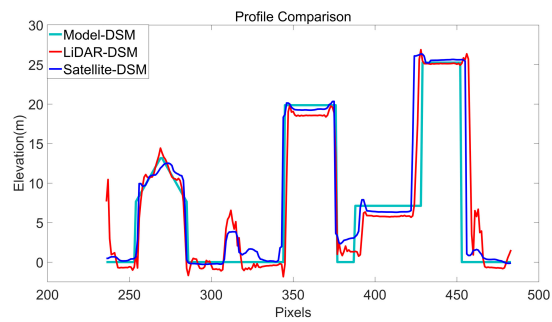
Figure 5.35: The model-nDSM for four areas of Munich



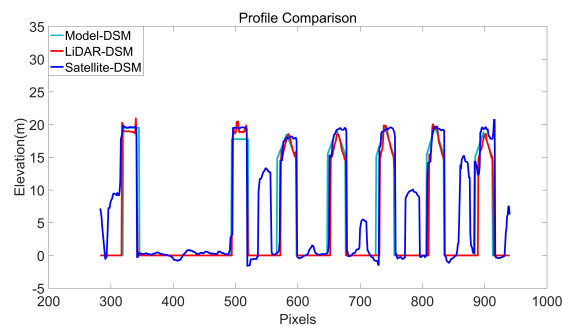
(a)



(b)



(c)



(d)

Figure 5.36: The comparison of the three profiles of the model-nDSM, the LiDAR-nDSM and the satellite-nDSM for four case studies of Munich.

5. 3D Building Model Reconstruction

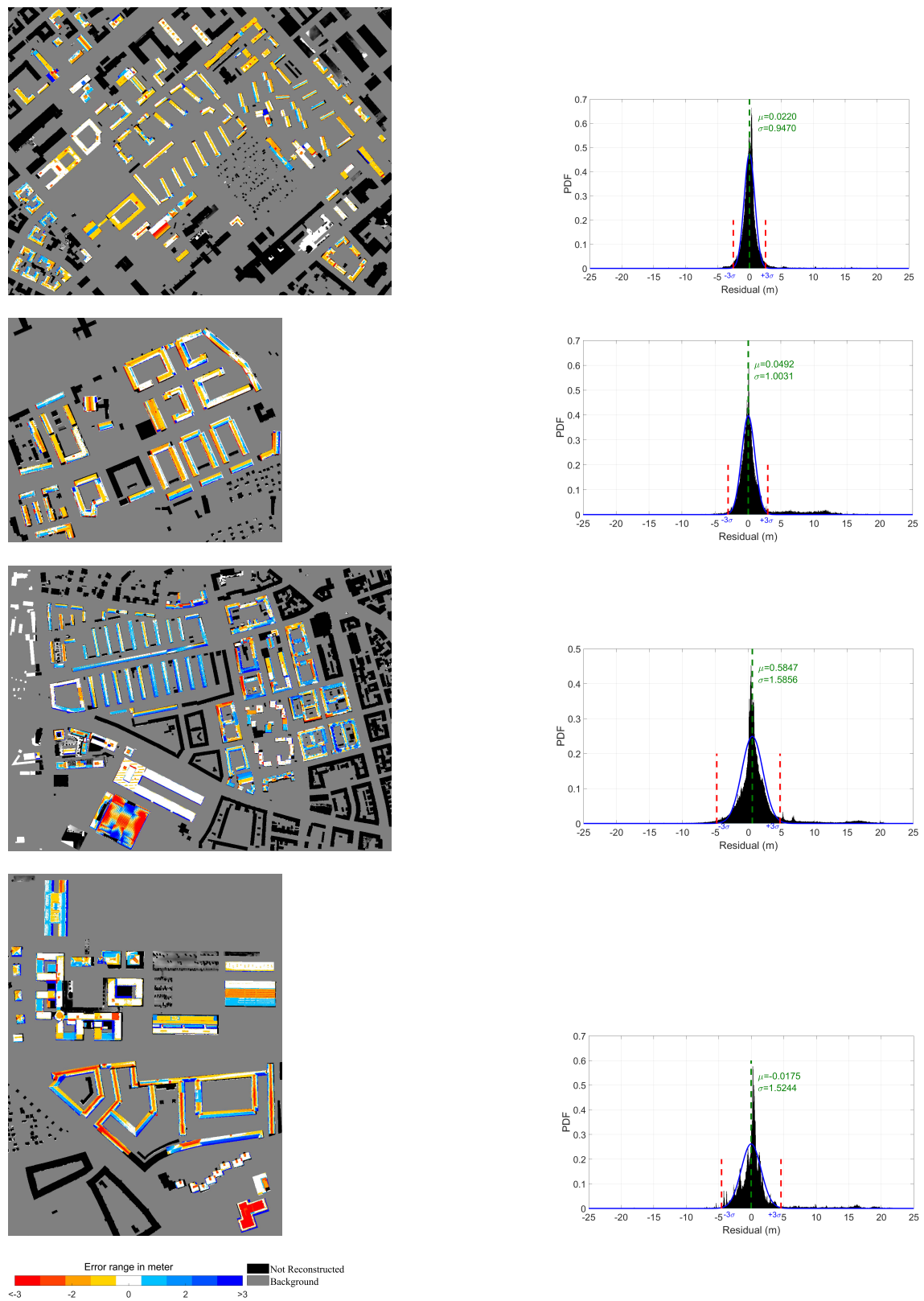


Figure 5.37: The errors maps (left) and the histograms of errors with the corresponding theoretical normal for four areas of Munich (right)

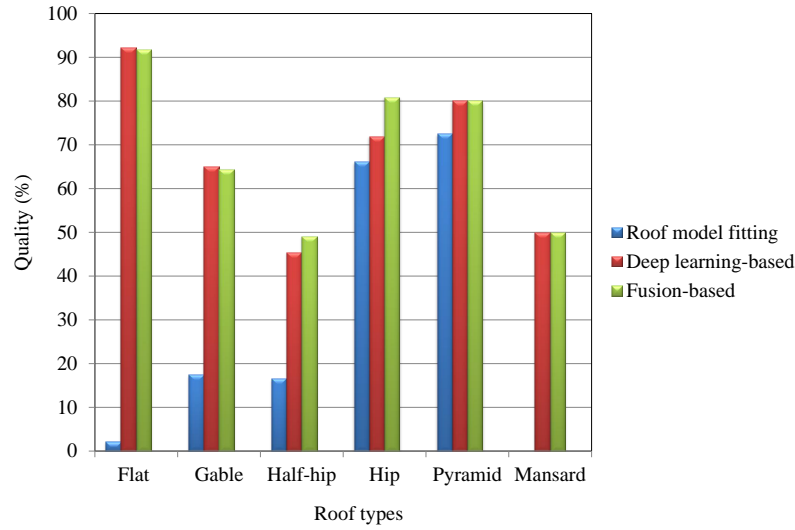


Figure 5.38: Comparing the performance of three roof type classification strategies, including roof model fitting to nDSM data of the buildings, applying a deep learning method to MS images and the fusion of the two other strategies. The results are the average of the four areas in Munich.

as ground truth.

5.5.2.1 Roof Type Classification

Figure 5.38 compares the precision of the assigned roof types to the building models based on three different roof type classification strategies, including roof model fitting to nDSM data of the buildings, applying a deep learning method to PS images and the fusion of the strategies mentioned previously. As the figure shows, roof model fitting to nDSM data obtained the poorest classification results. In addition, the two other strategies performed comparatively. According to the results, the fusion-based strategy outperformed using only the deep learning method for the hip and half-hip roofs, which indicates the added value by the nDSM data.

5.5.2.2 Ridge Orientation Detection

In this section, the performance of our proposed approach in ridge line orientation detection is evaluated. The ridge line detection depends on the roof types, for example, the orientation of the middle lines of the sloped roofs (e.g. gable, half-hip, hip, mansard) which could be parallel (PA) or perpendicular (PP) to the roof's longest line were detected. Moreover, ridge points (PO) were detected for the pyramid roofs with no orientation, and no ridge (WR) line was detected for the flat roofs. Therefore, the orientation of the flat and pyramid roofs were calculated based on the orientations of their virtual rectangle's middle lines (PA or PP). The quality of these orientations is evaluated based on the final roof model orientations. As shown

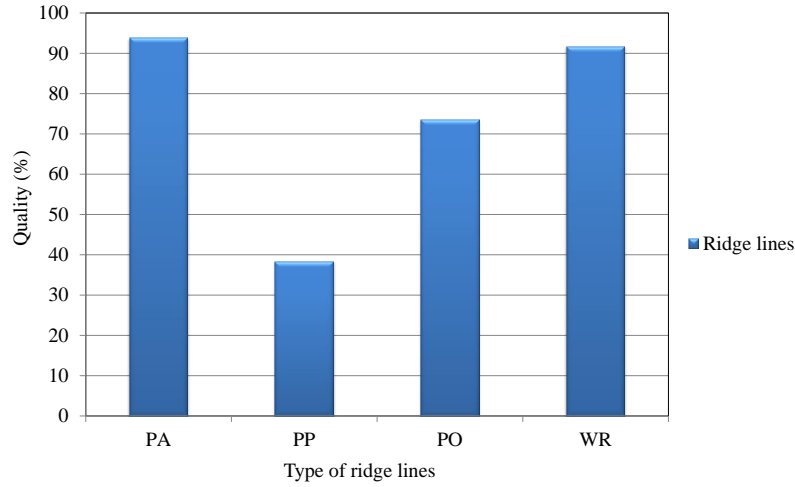


Figure 5.39: Evaluation of the ridge line orientation detection averaged over the four areas of Munich, where PA and PP denote the middle line parallel and perpendicular to the roof’s longest lines, relatively, PO denotes the ridge points for pyramid roofs and WR indicates the orientations of the flat roof buildings which have no ridge lines.

in Figure 5.39, the proposed method detected the PA and WR orientations better than the PP and PO ones. It is worth mentioning that the evaluation of PO and PP could be affected by the possible errors introduced by the manual annotation of the ground truth for the flat and pyramid roofs.

5.5.2.3 3D Building Models

In order to evaluate the resulting 3D building models quantitatively, we consider their nDSMs and compare them to the LiDAR-nDSM as the reference data. We conducted statistical analysis of the height differences (ΔH) between model-nDSM, satellite-nDSM and LiDAR-nDSM for the evaluation, where the equations used are represented in Table 5.4. According to the table, the estimated mean error ($\hat{\mu}$) indicates the mean height difference of all images which could be considered as a systematic error. Furthermore, the estimated standard deviation ($\hat{\sigma}$) demonstrates the smoothness of the model-nDSM. In addition, we computed the RMSE to measure the closeness of the model-nDSM to the reference one [170]. We assumed in our analysis that the errors follow the normal distribution; however, there are outliers (e.g. at the building boundaries) which need to be discarded for more reliable evaluations. Consequently, we eliminated the errors out of $[\mu - 3\sigma, \mu + 3\sigma]$ through an iterative procedure which calculate μ and σ in each iteration until no outlier remains in the data [171]. In addition to the statistical measures mentioned previously, we computed the normalized median absolute deviation (NMAD), which is proportional to the median of absolute difference between the errors and median error (please

Checkpoints before removing the outliers (N)	
Checkpoints after removing the outliers (N^*)	
Error	$\Delta h = h_{model/satellite} - h_{LiDAR}$
Mean Error after removing outliers	$\hat{\mu} = \frac{1}{N^*} \sum_{i=1}^n \Delta h_i$
Standard Deviation after removing outliers	$\hat{\sigma} = \sqrt{\frac{1}{(N^*-1)} \sum_{i=1}^{N^*} (\Delta h_i - \hat{\mu})^2}$
Root Mean Square Error	$RM\hat{S}E = \sqrt{\frac{1}{N^*} \sum_{i=1}^{N^*} \Delta h_i^2}$
Normalized Median Absolute Error	$NMAD = 1.4826 \cdot median_j(\Delta h_j - median(\Delta h))$

Table 5.4: Equations of accuracy measures

	Satellite	N	N^*	$\hat{\mu}(m)$	$\hat{\sigma}(m)$	$RM\hat{S}E(m)$	$NMAD(m)$
Area 1	WordView-2	367683	339870	0.022	0.9471	0.9473	0.7570
Area 2		171908	152100	0.0492	1.0031	1.0043	0.8768
Area 3		780152	703631	0.5847	1.5856	1.69	1.2173
Area 4		276954	251056	-0.0175	1.5244	1.5245	1.1830

Table 5.5: Accuracy measures (in meter) based on the differences between model-DSM and reference LiDAR-DSM

see Table 5.4) and is known to be robust in the presence of outliers and the errors with non-normal distribution [172]. The results of the statistical analysis for the model-nDSM and satellite-nDSM are presented in Tables 5.5 and 5.6, respectively. According to the tables, the quality of the model-nDSM of Area-1 is higher than the other areas and its quality is similar to the that of its corresponding satellite-nDSM. This could be due to the accurate building outlines obtained from the 2D CM providing appropriate initialization for the 2D optimization step.

The $RM\hat{S}E$ and NMAD of the model-nDSM of Area-3 and Area-4 are more than the other areas. In addition, these quality measures of model-nDSM are more than the satellite-nDSM ones, which might be due to the complexity of the buildings shapes (especially B9 in Figure 5.28). The maximum differences between quality measures (NMADs and $RM\hat{S}E$ s) of the model-nDSM and the satellite-nDSM do not exceed one meter, which are acceptable in the accuracies of satellite-nDSM, LiDAR-nDSM and model fitting.

The ridge lines and eave heights are compared with reference data obtained from the Department of Urban Planning and Building (DUPB) of Munich to evaluate the 3D building models. Tables 5.7 and 5.8 show this evaluation by $|\Delta H_{ridge}|$ and $|\Delta H_{eave}|$. In these tables, ‘‘Building Numbers’’ are shown as a-b, in which ‘‘a’’ corresponds to the building block number and ‘‘b’’ corresponds to the building number. The number of each building block has been plotted in Figures 5.40 and 5.41.

5. 3D Building Model Reconstruction

	Satellite	N	N^*	$\hat{\mu}(m)$	$\hat{\sigma}(m)$	$RMSE(m)$	$NMAD(m)$
Area 1	WorldView-2	367683	339241	0.0209	0.9444	0.9446	0.7419
Area 2		171908	156938	0.1751	0.8828	0.9	0.8263
Area 3		780238	727246	0.5776	1.3078	1.4297	1.0424
Area 4		276954	243819	0.3307	0.9507	1.0066	0.6146

Table 5.6: Accuracy measures (in meter) based on the differences between satellite-nDSM and reference LiDAR-nDSM

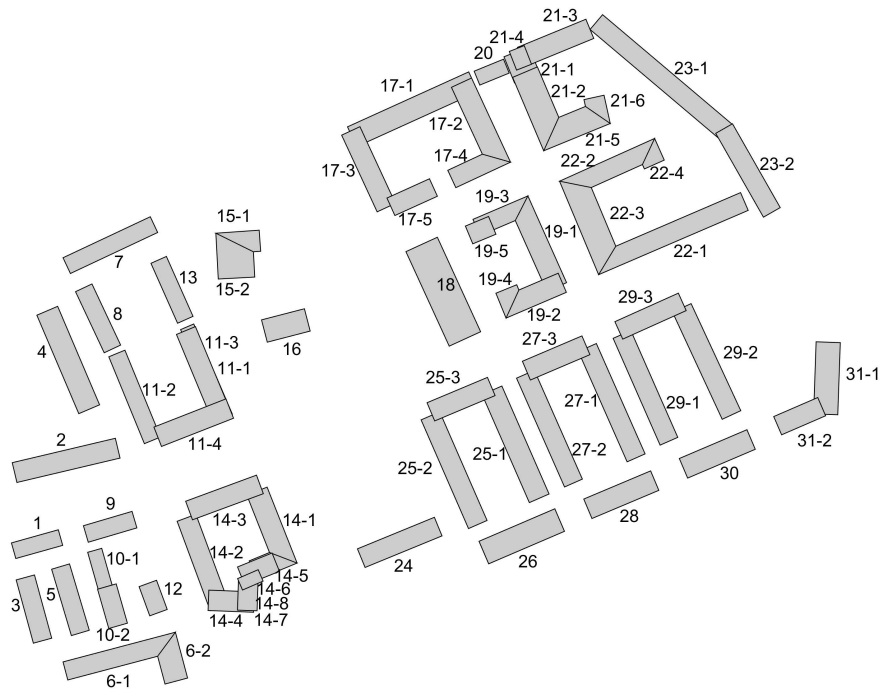


Figure 5.40: The building blocks and their numbers for Area-2

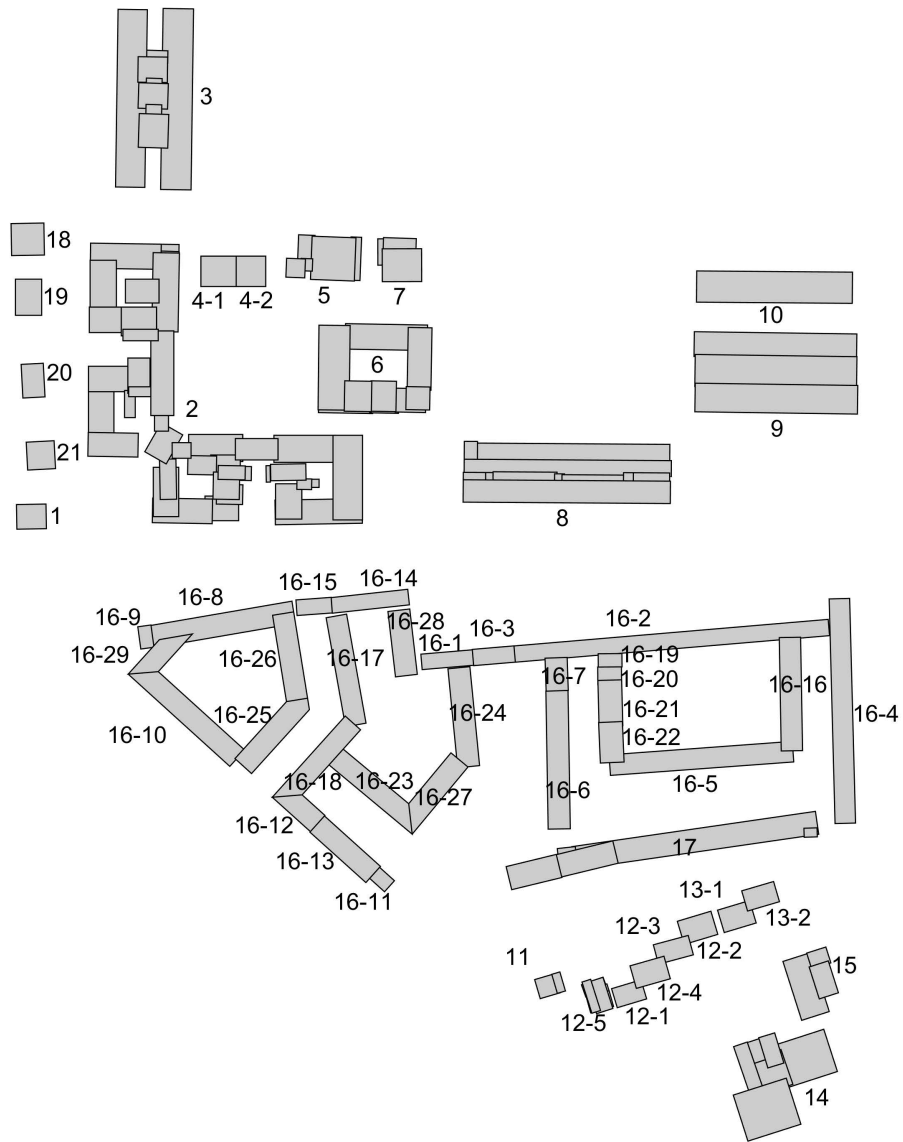


Figure 5.41: The building blocks and their numbers for Area-4

5. 3D Building Model Reconstruction

	Building Number	Ridge line			Eave heights		
		Model (m)	Reference (m)	$ \Delta H_{ridge} $ (m)	Model (m)	Reference (m)	$ \Delta H_{eave} $ (m)
1	17	14.0312	15	0.97	11.285	11	0.285
2	18	17.4579	19	1.5421	13.3837	15	1.6163
3	19	14.424	15	0.576	12.5237	11	1.524
4	20	14.22	15	0.78	10.86	11	0.14
5	21	14.0771	15.5	1.423	12.2066	11.9	0.31
6	22	15.2811	15.6	0.319	11.8707	11.5	0.3707
7	23	20.7604	19.97	0.79	17.3528	16.5	0.8528
8	24	15.8697	16.2	0.33	11.0331	12.3	1.267
9	25	16.2102	17.4	1.19	13.582	13.6	0.018
10	26	16.3967	16.8	0.403	10.5358	12.5	1.96
11	27	14.6644	15	0.3356	10.4881	10.9	0.412
12	28	16.27	16.8	0.53	10.4106	12.5	2.09
13	29	13.9435	14.7	0.7565	10.4634	10.7	0.2366
14	30	15.6579	16.8	1.1421	10.5092	12.5	1.99
μ				0.7919	0.9337		
σ				0.4102	0.7742		
RMSE				0.8851	1.1952		
NMAD				0.5479	0.8205		

Table 5.7: The difference between ridge/eave lines of the building model and DUPB-based reference data for Area-2

The existence of complex buildings with poor quality nDSM in Area-4 caused poorer extraction of ridge and eave lines compared to the another area.

5.6 Summary and Conclusion

In this chapter, our proposed novel multistage hybrid 3D building model reconstruction approach, which performs based on the nDSM of the WorldView-2 satellite, has been explained. The data-driven part of the proposed approach consists of some data-driven steps, such as 1) mask refinement, 2) building outline extraction, 3) building outline decomposition and 4) roof type classification. Moreover, auxiliary data, such as orthorectified PAN images and PS satellite images, were used to overcome the poor quality of the DSM. In the model-driven part, firstly, a library of six roof types, including flat, gable, half-hip, hip, pyramid and mansard roofs, was designed. Next, the geometrical relations were defined based on the eleven parameters for each roof type in the library. From these parameters, the number of 2D ones which parameterize the rectangles derived from the decomposition step is fixed, whereas the number of 3D ones varies according to the roof types. These parameters were initialized using the preknowledge obtained from the data-driven part. The preknowledge was further improved by detecting the height discontinuities, classifying the roof types using nDSM data, and defining a set of constraints based on roof shapes and geometrical structures. The combination of 2D and 3D parameters generates the initial 3D model. A discrete search space was defined

5.6. Summary and Conclusion

	Building Number	Ridge line			Eave heights		
		Model (m)	Reference (m)	$ \Delta H_{ridge} $ (m)	Model (m)	Reference (m)	$ \Delta H_{eave} $ (m)
1	4-1	13.6193	13.6	0.0193	5.41	5.3	0.11
2	4-2	16.41	17.8	1.39	10.0689	8.25	1.8189
3	12-1	-	-	-	20.777	21.3	0.523
4	12-2	-	-	-	26.777	26.7	0.077
5	12-3	-	-	-	25.4316	26.7	1.2684
6	12-4	-	-	-	22.9066	24	1.0934
7	12-5	-	-	-	26.2303	26.7	0.4697
8	13-1	-	-	-	20.8302	20.5	0.3302
9	13-2	-	-	-	15.7442	15.9	0.156
10	16-1	20.7855	20.7	0.1302	12.9192	14	1.081
11	16-2	20.1455	21.7	1.5545	13.5116	15.2	1.6884
12	16-3	20.9384	20.7	0.2384	14.7994	14	0.7994
13	16-5	20.6747	22	1.3253	13.1306	13.5	0.3694
14	16-6	16.7505	21	4.2495	15.1597	14.5	0.6597
15	16-7	20.6842	21	0.3158	13.7642	14.5	0.7358
16	16-8	17.4533	20.7	3.2467	11.6737	13.6	1.9263
17	16-9	18.7044	20.7	1.9956	11.6737	13.6	1.9263
18	16-10	21.2904	20.7	0.5904	13.5565	13.6	0.0435
19	16-11	16.6668	20.4	3.7332	15.9165	13.2	2.7165
20	16-12	16.5861	20.4	3.814	14.7764	18.1	3.3236
21	16-13	16.5864	20.4	3.8136	15.8916	13.2	2.6916
22	16-14	19.5136	21.4	1.8864	13.4226	14.4	0.9774
23	16-16	17.2514	21.8	4.549	15.3826	15.1	0.2826
24	16-17	17.46	21.4	3.94	15.17	14.4	0.77
25	16-18	16.5864	21.4	4.8136	14.7764	14.4	0.3764
26	16-21	21.6229	22	0.3771	13.8356	13.5	0.3356
27	16-24	18.0491	20.7	2.651	14.4457	14	0.4457
28	16-25	17.4533	20.1	2.6467	13.2158	13.2	0.016
29	16-26	17.4533	20.7	3.2467	15.0637	13.6	1.4637
30	16-28	17.1301	21	3.8699	15.2008	14.5	0.7008
31	17	18.1492	17	1.1492	11.6674	11	0.6674
32	18	14.6804	14.3	0.3804	8.4767	8.1	0.3767
33	19	13.33	14.5	1.17	7.4581	8	0.542
34	20	15.6958	17.3	1.6042	9.5981	8.6	0.9981
35	21	13.2968	14.4	1.1032	7.8932	8.25	0.357
		μ		2.1359			0.9176
		σ		1.5368			0.8202
		RMSE		2.6152			1.223
		NMAD		2.039			0.4999

Table 5.8: The difference between ridge/eave lines of the building model and DUPB-based reference data for Area-4

based on a domain in which the initial parameters alter in the specified range to generate different models to search for more reliable 3D models. The cost function was defined based on the minimum distances of the approximated 3D building model into the 3D point clouds and the reference building outline. Subsequently, all possible 3D building models were searched for each single building, and the one with a minimum cost value was selected as the final building model. The neighborhood relationships, such as orientation and ridge/eave heights, for a building block consisting of several buildings, in addition to the cost value mentioned previously, were also considered in the final decision of which 3D model to choose. After selecting 3D building models for all parts of the building block, the intersection and merging processes were carried out to reconstruct the 3D building block model. Finally, the results were evaluated quantitatively and qualitatively. Approximately 208 buildings in four areas of Munich have been reconstructed. The height profiles showed strong improvement in the ridge and eave lines in comparison to the LiDAR DSM and satellite DSM. Most of the roof types and ridge line orientations were detected correctly. The quantitative evaluations showed that ridge/eave heights were detected at the level of the uncertainty of the 3D point clouds (approx. 1 m); however, better results were obtained for the eave heights. More precise evaluation could not be performed due to the non-availability of sufficient reference data from the DUPB. Furthermore, the results represented that a small number of buildings were failed because their roof types had not been defined in the library and their reconstruction was carried out based on the existing roof types. In addition, the intersection of the neighboring rectangles in some building blocks, due to having very complex shapes, could not perform well according to the predefined rules. The orthorectification of the PAN images resulted in jagged building boundaries, especially for the area far from the nadir point and for the high-rise buildings. This drawback resulted in inaccurate reference building outlines for the 2D optimization step, which, in turn, reduced the accuracy of the 3D building models. Despite all these limitations, most of the buildings were reconstructed successfully and their generalization was satisfactory.

Summary, Conclusions, and Future Work

In this chapter, we summarize this dissertation and conclude the points discussed, followed by potential future work.

6.1 Summary

In this dissertation, a novel automatic 3D building model reconstruction approach was proposed which reconstructs building models in LoD2 based on the DSM data generated from the VHR stereo imagery of the WorldView-2 satellite. In order to overcome the drawbacks of the DSM data (e.g. blurred building outlines), a multistage hybrid method was proposed in which the DSM data is used together with its corresponding orthorectified PAN and PS data of multispectral (MS) WorldView-2 imagery.

In order to deal with the blurred building boundaries in the DSM-based building masks specified, gradient features were extracted from the corresponding PAN images using the SIFT method. Based on the extracted features, the image pixels were then classified as a building or non-building by means of an SVM classifier. Experimental results showed that this classification refined the building boundaries to a large degree. In the next step, a new heuristic data-driven approach based on the RANSAC method was proposed to extract parameterized building boundaries from the refined building masks through fitting lines to the building boundary points. The resulting building boundaries were irregular due to the presence of noise and artifacts; therefore, the line segments needed to be regularized. Since the regularization process requires the definition of the main orientations of the buildings, a novel orientation extraction method was proposed which can extract even multiple main orientations for the buildings with complex structures. Using the buildings' main orientations, the line segments were regularized by a LS adjustment method

which aligned them either parallel or perpendicular to the main orientations. After the regularization, the line segments were connected and intersected to their closest neighbors based on some predefined rules to create building polygons. Experimental results demonstrated that the building polygons extracted were similar to the original building shapes and the proposed approaches worked well for complex building outlines as well. The shape of the resulting building polygons was usually complex and, thus, needed to be simplified. Consequently, a rule-based generalization was applied to the building polygons which discarded short line segments. A new building decomposition approach was then proposed to decompose the generalized building polygons into a number of basic rectangular shapes. The rectangles generated overlapped for some complex buildings so that they were not representative enough for 3D building model reconstruction. Therefore, a novel heuristic approach was proposed to select the combination of relevant rectangles which cover the whole building footprint with minimum overlaps and represent the main parts of the building. Experimental results indicated that the proposed generalization and decomposition approaches performed well, even in the case of complex buildings with multiple orientations.

In the next step, an approach based on deep learning was employed to classify building roofs into six different categories, including flat, gable, half-hip, hip, pyramid and mansard roofs. The library of these roof types was generated through analyzing the height profiles of the decomposed building outlines in the DSM data. Training and test image patch datasets were generated from PS images with high spectral and spatial resolution in which the geometrical structure of different roofs is visible and recognizable to classify the roof types and to evaluate the results. The PS images were tiled into smaller patches for the classification. Therefore, a semi-automatic method was proposed based on a building mask skeleton and morphological operator. The image patches were then labeled manually and used for fine-tuning a pretrained CNN model, which was later used for the classification of the roofs. The classification results showed that this approach outperformed the conventional classification methods, such as SVMs. The results were then used as preknowledge in the next steps and updated later by fitting their models to the DSM data.

In the subsequent step, a new approach was proposed for creating watertight parameterized 3D building models (LoD2) by utilizing the nDSM of satellite imagery and the results of the previous steps (e.g. geometrical parameters of the roofs). More precisely, the decomposed building rectangles were improved by computing height discontinuities from the nDSM data. In addition, the roof type classifications were updated for each rectangle according to a set of roof combination rules and the nDSM data. For this purpose, the most probable roof types were fitted to the nDSM of the rectangles and their similarities to the 3D building points were calculated. The similarities together with the prior probabilities were then served as inputs to a Bayesian formula to determine the final roof type for each rectangle.

For the next step, which is the model-driven part of the pipeline, the geometrical relationships between 11 parameters of the roofs in the roof library were predefined and initialized by the results of the previous steps. The roof model parameters were categorized into the 2D and 3D ones. While 2D parameters derived from the building rectangles were fixed, the 3D parameters varied according to the roof types. By concatenating all parameters, initial 3D building models were generated which were optimized through an exhaustive search for the parameter values in defined discrete search spaces. In this process, the models' 2D parameters were evaluated by comparing the model to the building boundaries (obtained from building masks and PAN image) using the PoLiS metric. The 3D model parameters were evaluated by comparing the model to the 3D point clouds by means of the Euclidean Huber distance. After that, regarding the building blocks, the interactions between the rectangles were considered based on their overlaps, which led to reliable building roof models. Finally, the reconstructed models for all its rectangles were assembled through an intersection and merging process to reconstruct watertight models of each building block.

In order to evaluate the proposed approach, the 3D building models of four different areas in Munich were reconstructed and the results were evaluated qualitatively and quantitatively. According to the results, the model reconstructions of most of the buildings were successful; however, the reconstructed models failed in representing the original building models in a few cases, which could be due to the poor quality of the DSM data, decomposition errors, limitations of the roof model library, the presence of jagged building outlines in the PAN images, or limitations of the rules in the intersection and merging processes.

6.2 Conclusions

This dissertation proves that the DSM of VHR satellite imagery combined with the spatial and spectral information from spectral datasets, such as PAN and PS, can be used for large-scale 3D building model reconstructions. In addition, the proposed multistage approach with different sets of rules can handle noisy 3D point clouds and avoid many possible imperfections. It is worth mentioning that the incomplete set of rules may not reliably and precisely reconstruct all possible 3D building models. Furthermore, the proposed approach can achieve a high level of automation by reducing the number of primitive roof model types required and performing automatic parameter initializations. Only a few threshold parameters are needed to be manually initialized according to the resolution of the data as well as some knowledge about the region of interests and empirical investigations.

Various parts of the proposed approach have been evaluated qualitatively and quantitatively. According to the results, the proposed mask refinement method improved the DSM-based building mask specified by about 8% in area accuracy. Since

this method is based on supervised classification, a diverse set of training features should be considered by selecting them from the buildings with various roof types and orientations. In addition, applying the building outline extraction together with line segment regularization, adjustment, intersection and connection to the refined building masks could increase the precision of the outlines by about 70% compared to the building mask before refinement. These outlines outperform the ones extracted by the DP method (especially for the buildings with fine details). Moreover, the proposed approach can consider multiple orientation for complex buildings, which allows one to achieve better building extractions than the MBR-based approaches, such as iMBR and CMBR. Additionally, the proposed generalization method allows one to discard irrelevant vertices while preserving the building shapes. In addition, the decomposition method introduced has been shown to decompose building outlines into rectangles better than the state-of-the-art methods, as its resulting rectangles are large enough and informative. Moreover, using the information from the previous steps, such as multiple main orientations, in the presence of over-segmented building outlines can help one to avoid an over-splitting of the building polygons. The proposed method also considers the overlap of rectangles, which can facilitate the 3D building model reconstruction.

The method based on deep learning in the roof type classification step could outperform the conventional approaches by about 33%. In addition, the method used for generating training datasets based on the skeleton points of the building masks can result in overlapping patches. This could cause misclassifications, as each patch could contain multiple roof types while it is assigned only to one label. Furthermore, defining the library with six roof types speeded up the classification and reconstruction processes to a large degree, although other roof types, such as shell, canonical, ellipsoid and elliptic, could exist in the scenes, especially in older cities such as Munich. The roof types assigned were updated by fitting their model to the DSM data, which resulted in small improvements in half-hip and hip roofs. Furthermore, the results showed that the orientations of most of the ridge lines were defined correctly.

The proposed 3D building model reconstruction generally allows the reconstruction on buildings higher than 3 m and larger than 75 m² (300 pixels in the images with 0.5 m resolution). The evaluation of areas with 208 buildings demonstrated that the reconstruction of most of the buildings were satisfactory. Comparing the results to the reference LiDAR data indicated that the RMSE and NMAD were smaller than 2 m, which is acceptable according to the 3σ rule. Moreover, there were a few buildings for which the reconstructed models suffered from some deficiencies. The buildings that were partially lost in the detection step, for example, could not be recovered and fully reconstructed throughout the processing chain. In addition, the occasional poor performance of the mask refinement caused artifacts in the building outlines.

6.3 Future Work

The DSM-based building masks in this dissertation were refined by applying SVM to the SIFT features. Future studies could consider extracting image features by deep learning methods (e.g. CNNs) instead of the engineered feature extraction techniques. Furthermore, future studies involving roof type classification may input DSM data together with PS images and building rectangles to a CNN model to avoid the roof type updating step. In addition to dealing with the limited number of samples for training the CNN-based roof classifier, one could utilize a shallower network or generate synthetic training samples using a generative adversarial network (GAN).

Additionally, the roof type classification might be improved by applying a B-Spline surface approximation to the 3D point clouds within the decomposed building polygons. The resulting B-Spline knots, e.g., ridge lines endpoints, might be served as input to the CNN together with the PAN images. B-Spline can also be used to model very complex roof shapes which are not rectilinear, in particular, for historical buildings with curve shapes. Moreover, the ridge lines and their orientations can also be extracted by CNNs, which avoids the limitations of the symmetry constraint used in this dissertation.

An iterative robust estimation method could be applied to increase the robustness of model fitting in the presence of noisy DSM. A 3D GAN could also be used in 3D building reconstruction based on deep learning to reconstruct a 3D building model from DSM data and PS images. This network is based on 3D voxel space, which can subsequently be converted to vector data by fitting planes to the 3D voxelized models and extracting lines and corners from the voxels to reconstruct LoD2 models. In addition, a GAN could be designed to work with vector data and generating parameterized 3D building models.

A future preprocessing step to optimize the spaceborne DSM based on high level DSM (e.g. from LiDAR data) using a GAN- or segmentation-based method could reduce artifacts and increase the sharpness and shapes of the DSMs before starting the whole workflow of building reconstruction.

In this appendix, we describe datasets which have been used for our experiments.

A.1 DSM from Stereo Satellite Imagery

Digital Surface Models exhibit the height values of aboveground objects such as building and trees, while DTM refers to the height values of the ground surface without any man-made and natural elevated objects [170]. The nDSM exhibits the heights of objects above the ground, which can be simply described as subtraction of the DTM from DSM.

In this dissertation, the DTM is generated from DSM based on an advanced morphological method. In this method, instead of the standard method which uses dilation and erosion morphological operators, low rank median and high rank median filters are applied, respectively. The DTM is the by-product of the building mask generation procedure, which is explained in Chapter 1 [19].

In this dissertation, the DSM is generated by the photogrammetric technique from stereo satellite imagery based on the matching process between two or more stereo images. The DSM generation from stereo satellite imagery is not the focus of this thesis, therefore, only a brief introduction to satellite sensor, which is used in this dissertation and DSM generation procedure, is represented. A lot of different methods have been developed for DSM generation in the last decades [173, 174] and, in the last few years, new methods and algorithms coming from the computer vision side [175] have obtained very interesting results for DSM generation from spaceborne stereo data [33].

Stereo spaceborne image acquisition: There are two possibilities to obtain spaceborne stereo data: Along track stereo images are acquired at the same time and the same orbit either using more than one sensor looking at the earth or rotating the sensors in the along track direction; across track stereo images, which

are obtained from either two or more adjacent orbits or the same orbit just using a charged-couple device (CCD) line combination, by looking at the same area from two or more orbits positions [176]. Most of the stereo satellite data are obtained by along track stereoscopy, since image-matching density and, therefore, the quality of the DSM is reduced by the radiometric changes and shadow properties during two acquisition dates from two orbits. For this reason, the new generation of the VHR satellite data is used with along track stereoscopy technology. Worldview-2 is one of these VHR satellite sensors and its data is used in this dissertation. It is the Digital Globe’s third satellite which was launched on October 8th, 2009 [170, 177]. It delivers 0.5 meter ground sample distance (GSD) PAN images and 2-meter resolution 8-band MS images (red, green, blue, near-infrared 1, coastal, yellow, red, edge and near infrared 2). Using along track stereoscopy technology, the Worldview-2 sensor can acquire two stereo pairs in a large area on the same day and same orbit with an overlap in the test area, leading to four views with stereo angles between 12° and 24° [32, 170].

DSM generation method: The total DSM generation correction includes improvement of the rational polynomial coefficients (RPC) of the stereo scene blocks, dense stereo matching, spatial intersection and outlier removal, and interpolation. Stereo image matching is the core of the DSM generation. Reconstructing the details of the small structure and sharp object boundaries from VHR stereo images requires the use of dense stereo image matching that is not based on the correlation of image window. In this dissertation, stereo satellite DSM are computed by semi-global matching (SGM) method, which performs pixel-wise matching. The SGM-based DSM generation consists of three main steps: Matching cost computation, cost aggregation and disparity map computation. Dense stereo matching is performed using SGM on the stereo pair with known epipolar geometry. In order to generate an epipolar stereo pair, firstly, the tie points coordinates with sub-pixel accuracy are approximated by the pyramid-based local LS matching. These tie points are used to align the columns of the stereo pair. A quasi-epipolar stereo pair with epipoles corresponding to the aligned columns is then generated. The matching procedure of the SGM is based on the minimization of the global energy function for all disparities, which are the local shifts between the stereo pairs. This energy function is defined as:

$$E(Dm) = \sum_p \left(Cm(p, Dm_p) + \sum_{q \in N_p} \begin{cases} 0, & \|Dm_p - Dm_q\| = 0 \\ p_1, & \|Dm_p - Dm_q\| = 1 \\ p_2, & \|Dm_p - Dm_q\| > 1 \end{cases} \right) \quad (\text{A.1})$$

This equation includes two terms of the pixel-wise matching cost and regularization. Cm is the matching cost term between the image pixels for each pixel

location p and possible disparity Dm_p . Both mutual information as well as the census cost function are used. The regularization term of E penalizes disparity changes in the neighborhood N_p at each position p . The penalty p_1 is added for all disparity changes equal to one pixel. A fixed cost p_2 is added at larger discontinuities (disparity change > 1 pixel). This term (i.e regularization term) favors similar disparities for neighboring pixels but also allows large jumps at discontinuities.

The SGM performs a semi-global optimization by aggregating costs from 16 directions to reconstruct a near optimal image Dm . Occluded and mismatched regions are removed using a bidirectional consistency check and remaining small regions are deleted as outlines. For more details about outlier detection and SGM procedure, please refer to [33, 175, 178].

It is necessary to establish an RPC correction for DSM generation before dense stereo matching by SGM. The RPC is the transformation between object space and image space in order to georeference the image. These coefficients are originally obtained from the orbit and altitude information from the sensor. The original RPC image has a low absolute accuracy, which is lower than the ground resolution of approximately 2.5 m. The sub-pixel ground control points (GCPs) are required to refine RPCs. Such very accurate GCPs are derived from DGPS ground survey or high-resolution orthoimages and DEMs. These highly accurate GCPs are not available for the large-scale and near real-time application, such as disaster monitoring. Therefore, the image-based GCPs' extraction and correction method are used [179, 180]. For VHR satellite image, such as WorldView-2, RPCs are refined by incorporating some prior constraints into the adjustment model [170].

The 3D point cloud is generated from an SGM disparity map by the spatial intersection. They are then aligned with the shuttle radar topography mission (SRTM) DSM using a 3D affine transformation. Subsequently, the aligned point cloud and georeferenced image are used in the image resection for the final estimation of the affine RPC correction parameters [179]. The 3D points cloud is then generated using the RPC correction parameters and spatial intersection of the complete disparity map. The irregular point cloud is converted to the regularly spaced grid by the interpolation to facilitate further application. Orthoimages with the defined datum and projection are created by the orthorectification of the near nadir image with the DSM generated and the affine-corrected RPCs. The complete process is described in [33].

Although the SGM method is robust in illumination changes and accurate on the boundaries of an object, some holes, due mainly to occlusion, texture loss and a low contrast region, might still occur. These holes can be filled by either SRTM data or spectral data using interpolation methods [181, 36]. Remaining holes which cannot be filled by DSM and SRTM DSM are filled using B-spline interpolation, which results in the gradual height changes in the building boundaries.

All of the processes of the DSM generation have been carried out using DLR in-house XDibias software modules. Figures A.1 illustrate datasets for Area-3 (DSM,

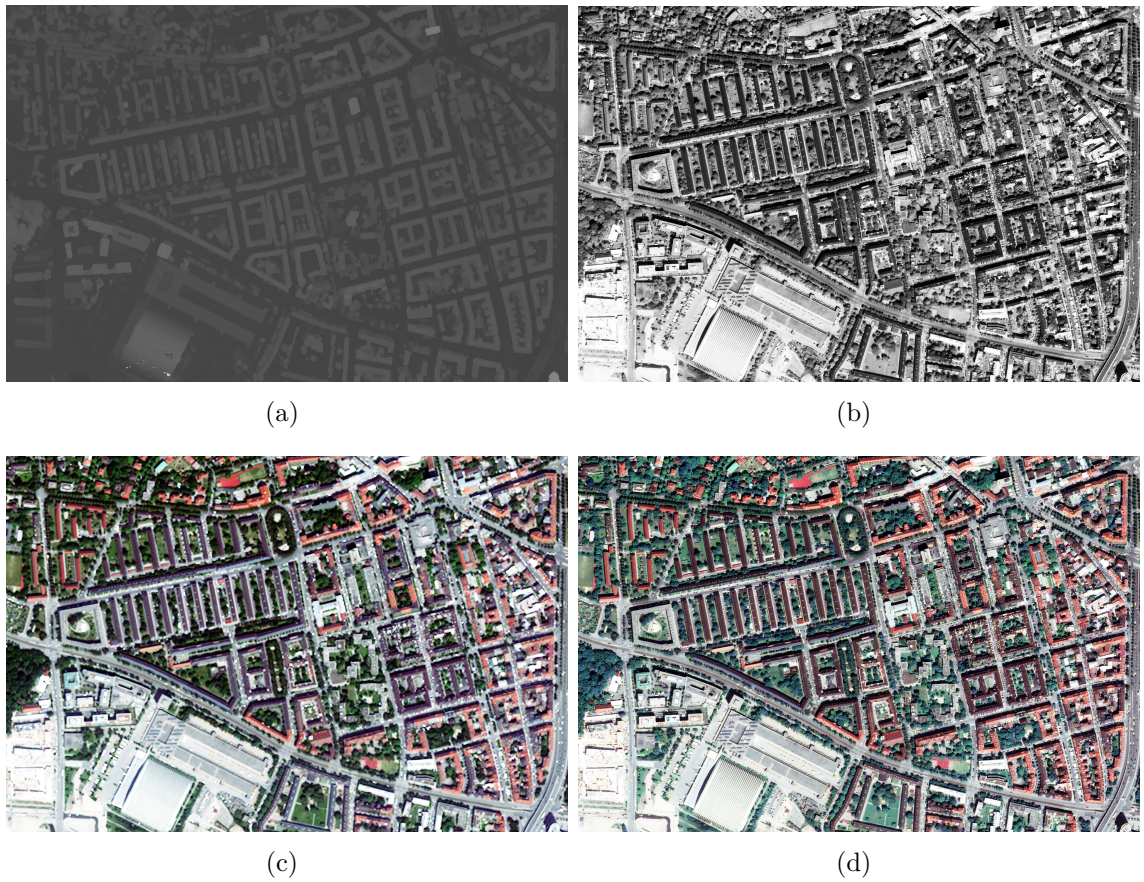


Figure A.1: Datasets (a) DSM of satellite imagery, (b) PAN image, (c) MS image, (D) PS image

PAN, MS and PS images) which are applied in this dissertation.



Figure A.2: LiDAR based DSM

A.2 DSM based LiDAR Data

The LiDAR data is captured by airborne LiDAR which detects the terrain surface from the aircraft. The data is provided by the “State Office for Digitization, Broadband and Surveying” of Munich. The resolution of the data is $5 \text{ points}/\text{m}^2$. The DSM of LiDAR data is used for the evaluation of the 3D building reconstruction results. In order to generate the DSM, the first pulse of LiDAR data including the ground and objects aboveground, such as trees and buildings, which covers 9.9km^2 area of Munich is rasterized at a resolution of 0.5 m after coordinate system transformation. After that, B-Spline interpolation is used to generate the DSM. The resolution of the DSM is 0.5, which is the same as the resolution of the stereo satellite data based DSM. Figure A.2 shows the DSM generated from 3D point clouds of LiDAR data, for example, of Area-3.

B

Tables of State-of-the-Art

In this appendix, Tables [B.1](#), [B.2](#), and [B.3](#) show the summary of different building modeling approaches according to the strategy of data processing, data sources, representation, level of details, scale and automation level. Different methods based on data processing strategy which use satellite imagery are shown in green and for LiDAR data in blue. In these tables, the satellite-based research works are shown by green and the outstanding research works which used LiDAR data are shown by blue.

B. Tables of State-of-the-Art

Authors	Representation	Level of details	Data sources	The Scale	Level of automation
(Mass and Vosselman, 1999)	CSG	LOD2	LiDAR data	48 buildings	Automatic
(Suveg and Vosselman, 2004)	CSG	LOD2	Aerial images/GIS maps	61 buildings	Automatic
(Poullis and You ,2009)	CSG	LOD2	3D point cloud of LiDAR data	Large scale	Semi-automatic (manual editing for complex building)
(Lafarge et al., 2010)	CSG	LOD2	DSM from satellite images	Three dense urban areas of Amiens (on the 500 quadrilateral object)	Automatic with constraints
(Hammoudi and Dornaika, 2011)	B-rep	LOD2	Aerial images	A few buildings in 0.173 km ²	Automatic for polyhedral model
(Huang et al., 2013)	CSG	LOD2	nDSM from LiDAR data	96 buildings	Automatic with constraints and pre-defined probability matrix
(Nguatem et.al., 2013)	B-rep	LOD2	3D point cloud from LiDAR or multi-view aerial images/cadastral maps	Middle scale (152 quadrilateral footprint)	Automatic
(Henn and et al., 2013)	CSG	LOD2	Sparse LiDAR data/2D ground plan	A part of Cologne with 1050 buildings	Automatic with constraints
(Zheng and Weng, 2015)	CSG	LOD2	LiDAR DSM/GIS maps	Two city blocks (Indianapolis in the US) with 268 buildings	Automatic

Table B.1: A summary of model-driven based building modeling approaches according to data sources, representation, level of details, scale and level of automation

(Brenner, 2000)	B-rep	LOD2	LiDAR DSM/ground plan	6465 buildings for four cities	Automatic
(Ameri and Fritch,2000)	B-rep	LOD2	DSM from matching of aerial images/aerial image	14 buildings	Automatic
(Vosselman and Dijkman, 2001)	B-rep	LOD2	Dense LiDAR data /2D ground plan	106 buildings	Automatic
(Jibrini, 2002)	B-rep	LOD2	Very high-resolution aerial stereo pair/cadastral map	A few complex building	Automatic
(Rottensteiner and Briese, 2003)	B.rep	LOD2	LIDAR DSM/aerial images	14 very complex blocks of buildings	Automatic
(Flamanc et al., 2003)	B-rep	LOD2	Aerial images/ cadastral map	5000 buildings	Semi-automatic
(Verma et al., 2006)	B-rep	LOD2	LiDAR data	248 buildings	Automatic
(Durupt and Taillandier, 2006)	B-rep	LOD2	DSM from multi-view stereo images	620 buildings of Amiens city	Automatic
(Tarsha-Kurdi et al., 2007)	CSG	LOD2	LiDAR DSM	62 buildings	Automatic
(Dorninger and Pfeifer, 2008)	B-rep	LOD2	Dense LiDAR data	Large scale	Automatic
(Milde et al., 2008)	B-rep	LOD2	LiDAR data	13 buildings	Automatic and semi-automatic
(Oude Elberink and Vosselman, 2009)	CSG	LOD2	Very dense LiDAR data	four areas from two cities in the Netherlands contain 728 buildings	Semi-automatic
(Sampath and Shan ,2010)	B-rep	LOD2	Dense LiDAR data	15 complex buildings	Automatic
(Zhou and Neumann, 2010)	Mesh	LOD2	LiDAR data	1879	Automatic
(Rau and Lin, 2011)	Mesh	LOD2	LiDAR data/2D ground plans	1573 building	Automatic
(Kada and Wichmann, 2012)	CSD	LOD2	LiDAR data	A few complex buildings	Semi-automatic
(Lafarge and Mallet, 2012)	Mesh	LOD2	LiDAR data	Large scale	Automatic
(Sohn et al., 2012)	CSG	LOD2	LiDAR data	21 buildings	Automatic
(Sun and Salvaggio, 2013)	B_rep	LOD1	LiDAR data	A few buildings	Automatic
(Peara and Mass, 2014)	CSG	LOD2	LiDAR data/digital aerial images	107 buildings from three areas of ISPRS benchmark	Automatic
(Xiong, et al., 2014)	CSG	LOD2	Very dense LiDAR data	4500 buildings of Enschede city	Automatic
(Orthuber and Avbelj, 2015)	Mesh	LOD2	Dense LiDAR	29 buildings	Automatic
(Duan and Lafarge, 2016)	Mesh	LOD1	Satellite images	Large scale	Automatic

Table B.2: A summary of data-driven based building modeling approaches according to data sources, representation, level of details, scale and level of automation

B. Tables of State-of-the-Art

(Haala and Brenner, 1999)	CSG	LOD2	LiDAR DSM/2D ground plan	A small area	Automatic
(Kada and McKinley, 2009)	B-rep	LOD2	LiDAR/2D ground plan	East Berlin with 244,000 buildings and Cologne with 280,000 buildings	Semi-automatic
(Sirmacek and et al., 2012)	B-rep	LOD2	DSM of satellite imagery/orthorectified panchromatic image	A few buildings of an area of Munich	Automatic
(Lin et al., 2013)	CSG	LOD2	Ground-based LiDAR data	53 buildings	Semi-automatic
(Arefi and Reinartz, 2013)	B-rep	LOD	DSM of satellite imagery/orthorectified panchromatic image	Seven buildings of an area of Munich	Automatic
(Kwak and et al., 2013)	B-rep	LOD1	LiDAR data/aerial image	32 buildings	Automatic
(Wang et al., 2015)	CSG	LOD2	LiDAR point cloud/multispectral aerial imagery.	107 buildings for three areas from ISPRS benchmark	Automatic
(Zheng et al., 2017)	B-rep	LOD2	LiDAR DSM/GIS map	519 buildings	Automatic

Table B.3: A summary of hybrid-based building modeling approaches according to data sources, representation, level of details, scale and level of automation

C

Support Vector Machine (SVM)

In this appendix, we explain the SVM classifier that was used in Chapter 2 section 2.1.3.

The SVM is an automatic or semi-automatic classification task in which the training data and test data should be split. Each instance in the training data contains a class label and several features. The goal of the SVM is to produce a model (based on the training data) which predicts the class label of the test. It seeks to find the optimum hyperplane between two classes that has maximum distance from the support vector points. The SVM classifier is a binary classification in which the class label can only take two values (-1, +1). There are more than two classes in the real world. Therefore, the SVM classifier is extended to reduce a multiclass problem into multiple binary classification problems. In our application, we used the LIBSVM classifier developed by Chang *et al.* [182]. The LIBSVM classifier uses the *one-versus-all* method for multi-class classification. It builds a binary-class problem for each label, so that instances associated with that label are in one class and the rest are in another class. The classifier with the highest output function assigns the class.

In the next session, we explain the mathematical concepts of the binary SVM classifier.

C.1 Binary SVM Classifier

Suppose we have a set of training data points $\mathbb{D} = \{(\vec{x}_i, y_i)\}$, where $x_i \in \mathbb{R}^d$, $i = 1, \dots, N$ which are labeled based on the class label $y_i \in \{-1, +1\}$. The binary SVM classifier should find a linear discriminative hyperplane in the vector space. A hyperplane is defined by a normal vector \vec{w} named *weight vector* and is perpendicular to the hyperplane. An intercept term b is specified so that one can choose among all the hyperplanes that are perpendicular to the normal vector. To find the efficient hyperplane, a kernel function is used to map the support vector

points into the high-dimensional feature space. Each point on the hyperplane satisfies $\omega^T \phi(\vec{x}_i) = -b$. $\phi(\vec{x}_i)$ is a transform that is related to a kernel function by the equation $k(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$. Kernel function can be selected as linear: $k(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j$, polynomial: $k(\vec{x}_i, \vec{x}_j) = (\gamma \vec{x}_i^T \vec{x}_j + r)^d$, $\gamma > 0$, Radial Basis Function (RBF): $k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2)$, or sigmoid: $k(\vec{x}_i, \vec{x}_j) = \tanh(\gamma \vec{x}_i^T \vec{x}_j + r)$ in which r , γ and d are kernel parameters. In our experiment, we use the RBF kernel function *et al.* [182]. Consequently, the decision function is defined as following relationship, which is recognized as *primal* function of classification.

$$f(\vec{x}_i) = \text{sign}(\omega^T \phi(\vec{x}_i) + b), \quad (\text{C.1})$$

where *sign* determines the positive and negative label of \vec{x}_i . The SVMs select b and \vec{w} as the computed hyperplane maximizes the geometrical margin between the classes. The geometric margin is equal to $\frac{2}{\|\omega\|}$ if it is defined by two parallel lines (Equation (C.2)) and a unit normal vector $\omega/\|\omega\|$.

$$\begin{aligned} \omega^T \phi(\vec{x}_i) - b &= 1, \\ \omega^T \phi(\vec{x}_i) - b &= -1. \end{aligned} \quad (\text{C.2})$$

In order to maximize the geometric margin, the weight vector $\|\omega\|$ should be minimized. Since using $\|\omega\|$ makes the optimization problem infeasible, minimization of the quadratic form of $\frac{1}{2}\|\omega\|^2$ is suggested.

$$\arg \min_{\omega, b} \frac{1}{2} \|\omega\|^2, \quad (\text{C.3})$$

subject to:

$$y_i(\omega^T \phi(\vec{x}_i) - b) \geq 1, \quad (\text{C.4})$$

The method of *Lagrange multipliers* is applied to solve this constrained problem. Therefore, the Lagrangian dual is defined:

$$\max_{\alpha \geq 0} (\min_{\omega, b} \{ \frac{1}{2} \|\omega\|^2 - \sum_i \alpha_i [y_i(\omega^T \phi(x_i) - b) - 1] \}) \quad (\text{C.5})$$

where ω and b are the *primal variables* and α_i is dual variables. Equation C.5 is convex and when α is fixed, the derivatives regarding ω , b vanishes and the following constraints are derived.

$$\omega = \sum_i \alpha_i y_i \phi(x_i), \quad (\text{C.6})$$

$$\sum_i \alpha_i y_i = 0. \quad (\text{C.7})$$

By substituting Equation C.6 into Equation C.5, the dual problem can be written as:

$$\max_{\alpha \geq 0} \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{\phi}(x_i, x_j) \right\}, \quad (\text{C.8})$$

where $\bar{\phi}(x_i, x_j)$ is a kernel function.

C.2 Soft-margin

In the real world, data are not linearly separable into the classes because of outliers and noisy measurements. Therefore, some points – outliers or noisy examples – are inside or on the wrong side of the margin. To handle this case without changing the family of decision functions, *soft-margin* is introduced to find a hyperplane that discriminates classes by paying a cost for each misclassified example, which depends on how far it is from meeting the margin requirement. In *soft-margin*, the slack variable ξ_i is introduced as a non-zero variable, which allows \vec{x}_i to not meet the margin requirement at a cost proportional to the value of ξ_i . The optimization problem of the *soft-margin* method is then a trade-off between a large margin and less cost to pay that is formalized as the following relationship.

$$\min_{\omega, \xi, b} \left\{ \frac{1}{2} \|\omega\|^2 + C \sum_i \xi_i \right\}, \quad (\text{C.9})$$

subject to:

$$y_i (\omega^T \phi(x_i) - b) \geq 1 - \xi_i, \quad (\text{C.10})$$

where C is a regularization term, which provides a way to control over-fitting: If C is large, the method is converted to *hard-margin* in which misclassification affect on the classifier performance, whereas the small value of C allows to admit misclassification in the training data. The dual form of the soft margin problem is:

$$\min_{\alpha_i} \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \right\}, \quad 0 \leq \alpha_i \leq C \text{ and } \sum_i \alpha_i y_i = 0. \quad (\text{C.11})$$

Instead of $\phi(x_i)^T \phi(x_j)$, a different kernel function can be used, as mentioned in section C.1. In the case of using an RBF kernel, parameters of C and γ are selected by the *k-fold* cross-validation method [183].

D

Algorithms

In this appendix, the pseudocodes of Chapter 5 are presented by the following algorithms.

D.1 Improvement of Roof Type Classification Results

Algorithm 1 shows the rules for improvement of the roof type classification results (Section 5.3.2).

D.2 2D Optimization

Using Algorithm 2, all possible 2D models for one rectangle are generated by the exhaustive search and then the best one is selected by Algorithm 3 (Section 5.4.1.2).

D.3 3D Optimization

After finding the best 2D building models, the Algorithm 4 searches all combinations of 3D parameters, which results in different 3D building models and then the best 3D building model is selected among all the possible models by Algorithm 5 (Section 5.4.1.3).

Data: I_{class} : the classified images based on the roof types
Result: f_{rt} : final roof type
Initialization :
 num_{labels} : the number of roof types labels for each rectangle;
 $LBL \mid_{i=1}^{num_{labels}}$: the classification results for each rectangle;
 Ind_{LBL_i} : the pixel indices related to each label for each rectangle;
 $L_{lib_j} \mid_{j=1}^{n=6}$: roof types labels in the pre-defined library. $j \in [1(\text{flat}), 2(\text{gable}), 3(\text{half-hip}), 4(\text{hip}), 5(\text{pyramid}), 6(\text{mansard})]$.;
for $p \leftarrow 1$ **to** n **do**
 | $Ind_{LBL_p} \leftarrow find(I_{class} == L_{lib_p});$
end
 $[index_{class}] \leftarrow \max(length(Ind_{LBL_i} \mid_{i=1}^{num_{labels}}));$
if $LBL_i \mid_{i=1}$ **then**
 | $f_{rt} \leftarrow LBL_i;$
 | **else if** $LBL_i \mid_{i \geq 2}$ **then**
 | $f_{rt} \leftarrow$ set LBL_i according to the predefined rules (Section 5.3.2);
 | **end**
 | **else**
 | $f_{rt} \leftarrow index_{class}$
 | **end**
end

Algorithm 1: Determination of the roof type for each rectangle according to the classification results

Data: Initial 2D parameters:

$$\Psi_{2D} = \{x_o, y_o, [], [], length, width, orientation, [], [], [], []\},$$

nDSM, rectangle, building mask

Result: All possible models $M(\Psi_{2D})$ and their corresponding cost values $Cost(\Psi_{2D})$

Initialization :

Ψ_{range} : tolerance range of each parameter $\psi \in \Psi_{2D}$

$PR_{N_{range} \times 11}$: The tolerance ranges matrix of parameters (N_{range} is a length of Ψ_{range}) ;

$Index = [1, 2, 5, 6, 7]$: indices of Ψ_{2D} ;

$cont \leftarrow true$, $ms(\Psi_{2D})$, $M(\Psi_{2D})$ and $MT(\Psi_{range})$ are empty;

$n \leftarrow 1$, $N_{Perm} \leftarrow 1$;

$Perm = \mathbf{1}_{i=1:maxPL}$, $maxPL \leftarrow 11$;

while ($cont$) **do**

for $j \leftarrow 1$ **to** $length(Index)$ **do**

$ms_{1 \times 11}(Index(j)) \leftarrow PR_{N_{range} \times 11}(Perm(Index(j)), Index(j))$

$MT_{N_{Perm} \times 11}(\Psi_{range}) \leftarrow ms$;

$M_{N_{Perm} \times 11}(\Psi_{2D}) = \Psi_{2D} + MT_{N_{Perm} \times 11}(\Psi_{range})$

$Cost_{N_{Perm} \times 11}(\Psi_{2D}) = PoLiS(M_{N_{Perm} \times 11}(\Psi_{2D}))$

$Perm(Index(1)) \leftarrow Perm(Index(1)) + 1$;

$N_{Perm} \leftarrow N_{Perm} + 1$;

if $Perm(Index(1)) > length(PR(:, Index(1)))$ **then**

$n \leftarrow 2$;

$carry \leftarrow true$;

while ($carry$) **do**

if $n > length(Index)$ **then**

$carry \leftarrow false$;

$cont \leftarrow false$;

else

$Perm(Index(n)) \leftarrow Perm(Index(n)) + 1$;

if $Perm(Index(n)) > length(PR(:, Index(n)))$ **then**

$n \leftarrow n + 1$;

if $n > length(Index)$ **then**

\perp break

else

$carry \leftarrow false$;

$Perm(Index(1) : Index(n) - 1) \leftarrow 1$;

Algorithm 2: Generating all the possible 2D roof models (rectangles) according to permutations of the 2D roof model parameters in their related ranges

Data: u : the rectangle index, (u, j) : an index of $M_{N_{Perm} \times 11}(\Psi_{2D})$ or $Cost_{N_{Perm} \times 11}(\Psi_{2D})$ for rectangle u , $Cost_{N_{Perm} \times 11}^{(u,j)}(\Psi_{2D})$, $M_{N_{Perm} \times 11}^{(u,j)}(\Psi_{2D})$, $IndRect$: indices of the overlapping and neighboring rectangles

Result: $M_{best}(\Psi_{2D})$ and $Cost_{best}(\Psi_{2D})$: the best 2D model

Initialization :

num_{rect} : the number of rectangle

thr_θ : angle threshold between neighboring rectangles.

for $u \leftarrow 1$ **to** num_{rect} **do**

if $u == 1$ **then**

$[i1, i2] \leftarrow \min(Cost_{N_{Perm} \times 11}^1(\Psi_{2D}));$

$M_{best}^u(\Psi_{2D}) \leftarrow M_{1 \times 11}^{(1,i2)}(\Psi_{2D});$

$Cost_{best}^u(\Psi_{2D}) \leftarrow i1;$

else if $u > 1$ **then**

if $IndRect(u, u + 1)$ **then**

$ind_{Perm} \leftarrow \text{find}(|M_{N_{Perm} \times 11}^{u+1}(\Psi_{2D}(7)) - M_{best}^u(\Psi_{2D}(7))| < thr_\theta);$

if $ind_{Perm} \neq \text{empty}$ **then**

$M_{best}^{u+1}(\Psi_{2D}) \leftarrow M_{1 \times 11}^{(u+1, ind_{Perm})}(\Psi_{2D});$

$Cost_{best}^{u+1}(\Psi_{2D}) \leftarrow Cost_{1 \times 11}^{(u+1, ind_{Perm})}(\Psi_{2D});$

else

$[i1, i2] \leftarrow \min(Cost_{N_{Perm} \times 11}^{u+1}(\Psi_{2D}));$

$M_{best}^{u+1}(\Psi_{2D}) \leftarrow M_{1 \times 11}^{(u+1, i2)}(\Psi_{2D});$

$Cost_{best}^{u+1}(\Psi_{2D}) \leftarrow i1;$

end

end

end

end

end

Algorithm 3: Selection of the best 2D model among all possible models

Data: Initial 3D parameters:

$$\Psi_{3D} = \{[\], [\], z_{eave}, z_{ridge\ line}, [\], [\], [\], hipl_1, hipl_2, hipd_1, hipd_2\},$$

$$M_{best}(\Psi_{2D}), f_{rt} \text{ nDSM, rectangle, building mask.}$$

Result: All possible models $M(\Psi_{3D})$ and their corresponding cost values $Cost(\Psi_{3D})$

Initialization :

Ψ_{range} : tolerance range of each parameter $\psi \in \Psi_{3D}$;

$PR_{N_{range} \times 11}$: the tolerance ranges matrix of parameters (N_{range} is the length of Ψ_{range});

if $f_{rt} == [2, 3, 4, 5, 6]$ **then**

 | $Index \leftarrow [3, 4, 8]$;

 | **else if** $f_{rt} == [1]$ & $z_{eave} \neq z_{ridge\ line}$ **then**

 | $Index \leftarrow [3, 4]$

 | **end**

 | **else if** $f_{rt} == [1]$ **then**

 | $Index \leftarrow [3]$

 | **end**

end

$cont \leftarrow true$, $M(\Psi_{3D})$ and $MT(\Psi_{range})$ are empty;

$ms(\Psi_{3D})$, $n \leftarrow 1$, $N_{perm} \leftarrow 1$;

$Perm = \mathbf{1}_{i=1:maxPL}$, $maxPL \leftarrow 11$;

```

while (cont) do
  for  $j \leftarrow 1$  to  $\text{length}(\text{Index})$  do
    |  $ms_{1 \times 11}(\text{Index}(j)) \leftarrow PR_{N_{\text{range}} \times 11}(\text{Perm}(\text{Index}(j)), \text{Index}(j))$ 
  end
   $MT_{N_{\text{Perm}} \times 11}(\Psi_{\text{range}}) \leftarrow ms;$ 
   $M_{N_{\text{perm}} \times 11}(\Psi_{3D+2D}) = \Psi_{3D} + MT_{N_{\text{Perm}} \times 11}(\Psi_{\text{range}}) + M_{\text{best}}(\Psi_{2D})$ 
   $Cost_{N_{\text{Perm}} \times 11}(\Psi_{3D+2D}) = \rho(\text{od}(nDSM, M_{N_{\text{perm}} \times 11}(\Psi_{3D+2D})))$ 
   $\text{Perm}(\text{Index}(1)) \leftarrow \text{Perm}(\text{Index}(1)) + 1;$ 
   $N_{\text{Perm}} \leftarrow N_{\text{Perm}} + 1;$ 
  if  $\text{Perm}(\text{Index}(1)) > \text{length}(PR(:, \text{Index}(1)))$  then
    |  $n \leftarrow 2;$ 
    |  $\text{carry} \leftarrow \text{true};$ 
    | if  $\text{Perm}(\text{Index}(1)) > \text{length}(PR(:, \text{Index}(1)))$  then
      | |  $n \leftarrow 2;$ 
      | |  $\text{carry} \leftarrow \text{true};$ 
      | | while (carry) do
        | | | if  $n > \text{length}(\text{Index})$  then
          | | | |  $\text{carry} \leftarrow \text{false};$ 
          | | | |  $\text{cont} \leftarrow \text{false};$ 
        | | | else
          | | | |  $\text{Perm}(\text{Index}(n)) \leftarrow \text{Perm}(\text{Index}(n)) + 1;$ 
          | | | | if  $\text{Perm}(\text{Index}(n)) > \text{length}(PR(:, \text{Index}(n)))$  then
            | | | | |  $n \leftarrow n + 1;$ 
            | | | | | if  $n > \text{length}(\text{Index})$  then
              | | | | | | break
            | | | | | end
          | | | | else
            | | | | |  $\text{carry} \leftarrow \text{false};$ 
            | | | | |  $\text{Perm}(\text{Index}(1) : \text{Index}(n) - 1) \leftarrow 1;$ 
          | | | | end
        | | | end
      | | end
    | end
  end
end

```

Algorithm 4: Generating all possible 3D roof models according to the permutation of roof model parameters in their related ranges

Data: u : the rectangle index, (u, j) : an index of $M_{N_{Perm} \times 11}(\Psi_{2D+3D})$ or $Cost_{N_{Perm} \times 11}(\Psi_{2D+3D})$ for rectangle u , $Cost_{N_{Perm} \times 11}^{(u,j)}(\Psi_{2D+3D})$, $M_{N_{Perm} \times 11}^{(u,j)}(\Psi_{2D+3D})$, $IndRect$: indices of the overlapping and neighboring rectangles

Result: $M_{best}(\Psi_{2D+3D})$ and $Cost_{best}(\Psi_{2D+3D})$: the best 2D+3D model

Initialization :

num_{rect} : the number of rectangle

thr_h : angle threshold between neighboring rectangles.

thr_c : cost value threshold

for $u \leftarrow 1$ **to** num_{rect} **do**

if $u == 1$ **then**

$[i1, i2] \leftarrow \min(Cost_{N_{Perm} \times 11}^1(\Psi_{2D+3D}))$;

$M_{best}^u(\Psi_{2D+3D}) \leftarrow M_{1 \times 11}^{(1,i2)}(\Psi_{2D+3D})$;

$Cost_{best}^u(\Psi_{2D+3D}) \leftarrow i1$;

else if $u > 1$ **then**

if $IndRect(u, u + 1)$ **then**

$ind1_{perm} \leftarrow$

$find(|M_{N_{Perm} \times 11}^{u+1}(\Psi_{2D+3D}(3)) - M_{best}^u(\Psi_{2D+3D}(3))| < thr_h)$;

$ind2_{perm} \leftarrow$

$find(|M_{N_{Perm} \times 11}^{u+1}(\Psi_{2D+3D}(4)) - M_{best}^u(\Psi_{2D+3D}(4))| < thr_h)$;

$ind3_{perm} \leftarrow find(Cost_{N_{Perm} \times 11}^{u+1}(\Psi_{2D+3D}) < c_{thr})$;

if $ind1_{perm} \wedge ind2_{perm} \wedge ind3_{perm} \neq empty$ **then**

$[i1, i2] \leftarrow \min(Cost_{size(ind3_{perm}) \times 11}^{(u+1, ind3_{perm})}(\Psi_{2D+3D}))$

$M_{best}^{u+1}(\Psi_{2D+3D}) \leftarrow M_{1 \times 11}^{(u+1, ind3_{perm}(i2))}(\Psi_{2D+3D})$;

$Cost_{best}^{u+1}(\Psi_{2D+3D}) \leftarrow Cost_{1 \times 11}^{u+1, ind3_{perm}(i2)}(\Psi_{2D+3D})$;

else

$[i1, i2] \leftarrow \min(Cost_{N_{Perm} \times 11}^{u+1}(\Psi_{2D+3D}))$;

$M_{best}^{u+1}(\Psi_{2D+3D}) \leftarrow M_{1 \times 11}^{(u+1, i2)}(\Psi_{2D+3D})$;

$Cost_{best}^{u+1}(\Psi_{2D+3D}) \leftarrow i1$;

end

end

end

end

end

Algorithm 5: Selection of the best 3D model among all possible models

Acronyms

AD.....	Area Difference
B-rep	Boundary representation
CCD.....	Charged-Couple Device
CH.....	Convex Hull
CityGML	City Geography Markup Language
CM	Cadastral Map
CNNs.....	Convolutional Neural Networks
CMBR.....	Combined Minimum Bounding Rectangle
Compl	Completeness
Conv.....	Convolution
Corr	Correctness
CSG	Constructive Solid Geometry
CT.....	Contour Trueness
DeCAF	Deep Convolutional Activation Feature
DEM.....	Digital Elevation Model
CLF	Compass Line Filter
DoG	Difference of Gaussians
DP	Douglas-Peucker
DTM.....	Digital Terrain Model
DSM.....	Digital Surface Model
EM	Expectation Maximization

Acronyms

EO	Earth Observation
F-distribution	Fisher-distribution
FCN	Fully Convolutional Neural Networks
FN	False Negative
FP	False Positive
FC	Fully Connected
GCPs	Ground Control Points
GH	Gauss Helmert
GIS	Geographical Information System
GMM	Gaussian Mixture Model
GM	Gauss Markov
GSD	Ground Sample Distance
IEEE	Institute of Electrical and Electronics Engineers
iMBR	iterative MBR
InHist	Intensity Histogram
LOD	Level of Details
LiDAR	Light Detection and Ranging
LS	Least-Squares
MAP	Maximum A Posterior
MBR	Minimum Bounding Rectangle
MDL	Minimum Description Length
ML	Machine Learning
MRF	Markov Random Field
MS	Multi-Spectral
MSAC	M-Estimator Sample Consensus
MVS	Multi-View Stereo
nDSM	normalized Digital Surface Model
NDVI	Normalized Differenced Vegetation Index
PAN	Panchromatic
PA	Parallel

PCA	Principal Component Analysis
PoLiS	Polygons and Line Segment
PO	Point
PP	Perpendicular
PS	Pan-Sharpened
RF	Random Forest
RANSAC	Random Sampling Consensus
RBF	Radial Basis Function
RJMCMC	Reversible Jump Markov Chain Monte Carlo
rMBR	recursive Minimum Bounding Rectangle
RMSE	Root Mean Square Error
RPC	Rational Polynomial Coefficients
RTG	Roof Topology Graph
RTrees	Random Trees
SD	Symmetric Difference
SGM	Semi-Global Matching
SGD	Stochastic Gradient Descent
SIFT	Scale-Invariant Feature Transform
SIFT-Color	Scale-Invariant Feature Transform for RGB Color Channels
SIS	Sequential Importance Sampling
SMC	Sequential Monte Carlo
SMS	Split-Merge-Shape
SNR	Signal to Noise Ratio
SRTM	Shuttle Radar Topography Mission
SURF	Speeded Up Robust Features
SV	Soft Voting
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TIN	Triangulation Irregular Network
TN	True Negative

Acronyms

TP	True Positive
UN.....	United Nation
VR.....	Virtual Reality
VHR.....	Very High Resolution
WLD.....	Weber Local Descriptor
WR	Without Ridge line

List of Symbols

- \mathbf{A} Partial derivative of observation equations with respect to unknown parameters (design matrix); Jacobian matrix
- a_1 The first component of normal vector to line equation F_1
- a_2 The second component of normal vector to line equation F_1
- a_{11} The x^2 coefficient of quadratic surface
- a_{22} The y^2 coefficient of quadratic surface
- a_{12} The xy coefficient of quadratic surface
- a_{10} The x coefficient of the quadratic surface
- a_{01} The y Coefficient of the quadratic surface
- a_{00} Intercept coefficient of the quadratic surface
- $ac(.)$ Activation function
- AD Area Difference
- $Area$ The number of pixels inside building mask or the area of the rectangular polygon
- \mathbf{B} Partial derivative of observation equations with respect to observations
- b Interception term in an SVM method
- \mathbf{b} Biases vector in CNNs
- \mathbf{b} Intercept vector of line equations
- b An intercept of line equation in the vector of \mathbf{b}
- BI Binary image
- c_i A class of main orientations

C	Regularization parameter of an SVM method
\mathcal{C}	Contour parameters
$Cm(\cdot)$	Matching cost term
CT	Contour Trueness
$Cost(\cdot)$	Cost function of Model
$Cost_{best}(\cdot)$	The cost function of the best models
\mathbb{D}	Set of training data in a supervised machine learning algorithm
$\vec{d}_p(\mathbf{V1}, \mathbf{V2})$	Directed PoLiS distance (or PoLiS distance) between two polygons of $\mathbf{V1}$ and $\mathbf{V2}$
D	Data
Dm_p	Disparity of pixel p
Dm_q	Disparity of pixel q
Dm	Near optimal image in SGM method
ds	Distance between two successive profiles
$exp(\cdot)$	Exponential function
$E(\cdot)$	Energy function
ev_n	Eigen vector in last column of \mathbf{V} corresponding to λ_n
\mathbf{e}_j	Edge of building polygon
$\mathbf{F}(\cdot)$	A vector of target functions
\mathbf{F}_1	Line equation (target function in GH model)
\mathbf{F}_2	Constraint equation to normalize the normal vectors of the line segments (target function in GH model)
f	Decision function in an SVM method
f_{rt}	final roof type
$F(x)$	The residual mapping in pre-trained ResNet
$q(x, y)$	A local parameterization of surface S in a neighborhood of point p
$q_x(p)$	The derivative of quadratic surface with respect to x
$q_y(p)$	The derivative of quadratic surface with respect to y
$q_{xx}(p)$	The second derivative of quadratic surface with respect to x
$q_{yy}(p)$	The second derivative of quadratic surface with respect to y

GE	Generalized building outlines
$G(i)$	Gradients (height differences) between pixels i and $i + rs$
$H(x)$	The mapping function in pre-trained ResNet
\mathcal{H}	The mean curvature
$hipl$	Longitudinal hip distances
$hipw$	Latitudinal hip distances
hip_{side}	The Function of determining the hip side
h_{LiDAR}	Height of DSM based LiDAR
h_{model}	Height of DSM based model
$h_{satellite}$	Height of DSM based satellite
I	Image
I_{class}	the classified images based on the roof types
Ind_{LBL_i}	the pixel indices related for each rectangle
$Index$	Indices of Ψ_{2D}
$IndRect$	indices of the overlapping and neighboring rectangles
J	Dimensionality of vector $V1$
\mathbf{J}	Jacobian Matrix
\mathbf{k}	Correlation vector
K	Dimensionality of vector $V2$
\mathcal{K}	Gaussian curvature
k_1	Minimum curvature
k_2	Maximum curvature
k	Kernel function
l_{ij}	Length of the line segment j which belongs to the orientation class c_i
$\mathbf{l}_1, \mathbf{l}_2, \text{ and } \mathbf{l}_3$	Line segments
l_n	small line segment
l_{n-1}	The predecessor of line segment l_n
l_{n+1}	The successor of line segment l_n
\mathbf{L}	Observations in LS

List of Symbols

\mathbf{l}	Observation vector of line points coordinates (x, y)
\log	Logarithm function
l_{ext}	The length and width of profiles out of rectangle
l_a	The longest middle line of rectangle a
l'_a	The shortest middle line of rectangle a
l_b	The longest middle line of rectangle b
l'_b	The shortest middle line of rectangle b
LBL	the classification results for each rectangle
L_{lib_j}	roof types labels in the pre-defined library
m	Dimensionality of a point or a vector
$mask_{overlap}$	The building mask in overlapping parts of two neighboring rectangles
M	The set of models
M_{best}	The best 2D models
M_i	A model in the set of models M
MO	Number of main orientations
$MT(\Psi_{range})$	The permutation matrix of the parameter ranges
\hat{M}_i	The best model with high posterior probability
n	Dimensionality of a point or a vector
n_x	Unit normal component to the line in x direction
n_y	Unit normal component to the line in y direction
n_i	Unit normal to a line
num_{labels}	the number of roof types labels for each rectangle
n_p	The unit normal to S at point p
$n_{estimators}$	Number of estimators in RF classifier
N	Number of pixels or points in point set X or Bin
N^*	Number of checkpoints after removing the outliers
N_G	Number of building outline vertices after generalization
N_O	Number of building outline vertices before generalization
N_{eave}	The number of buffer pixels around eave line

$N_{ridgeline}$	The number of buffer pixels around ridge line
N_{range}	The length of Ψ_{range}
N_{Perm}	The number of permutation of parameters ranges
num_{rect}	The number of rectangle
od_j	Orthogonal distances from points p_j to the surface of the 3D model
OR	Original building outlines
$orientation$	The orientation of roof model primitives
O_{l_b, l_a}	Intersection between middle lines l_a and l_b
P	Points (x_p, y_p) which are not on the line
$p_{(c_i)}$	A probability of every existing main orientation class c_i
$Patch_i(.)$	The i^{th} Patch
$P(.)$	Probability function
\mathcal{P}	Position parameters
$P(M_i)$	Prior probability of model i in the set M
$P(D M_i)$	likelihood function: The probability of data (D) given the model (M_i)
$P(M_i D)$	Posterior probability: The probability of model (M_i) given the data (D)
$P(D)$	The probability of data (D)
$PR_{N_{range} \times 11}$	The tolerance ranges matrix of parameters
p_1	The penalty for one pixel disparity changes
p_2	The penalty for more than one pixel disparity changes
$Perm$	Probability function
QP	Quality Parameters
QR	QR Decomposition of matrix A
r	Distance vector of points P to the line
$\ r\ $	A magnitude of distance from points P to the line
\mathbb{R}^m	A m -dimensional real coordinate space
$R_{(.)}$	Rectangle
rs	Sampling resolution of the building mask on the profile

List of Symbols

SD	Symmetric Difference
σ^2	Variance
σ	Standard deviation
S	Smooth surface
\mathcal{S}	Shape parameters
SW_{ra}	The latitudinal right side of rectangle a
SW_{la}	The latitudinal left side of rectangle a
Sl_{ra}	The longitudinal right side of rectangle a
Sl_{la}	The longitudinal left side of rectangle a
SW_{rb}	The latitudinal right side of rectangle b
SW_{lb}	The latitudinal left side of rectangle b
Sl_{rb}	The longitudinal right side of rectangle b
Sl_{lb}	The longitudinal left side of rectangle b
SE	Set of structuring elements
se	A structuring element in a structuring element set SE
T	Transpose operator
TQ	Total Quality
t	iteration
\mathcal{T}	The classification results based on Mean Curvature and Gaussian Curvature
T	Threshold of the errors in Huber loss $\rho(od_j^2)$
thr	Threshold
$\mathbf{U}_{m \times m}$	An orthogonal matrix where its columns are the eigenvectors of $\mathbf{A}\mathbf{A}^T$
U	Set of linear coefficients
u_{il}	Linear coefficient corresponding to the points x_i and x_l
\mathbf{v}	Residuals of the observations
$\mathbf{V}_{n \times n}$	An orthogonal matrix where its columns are the eigenvectors of $\mathbf{A}^T \mathbf{A}$
$\mathbf{V1}$	Vector of polygon vertices
$\mathbf{V2}$	Vector of polygon vertices
$\mathbf{v1}$	Vertices vector of $V1$

$\mathbf{v2}$	Vertices vector of $V2$
V_N	The number of building polygon vertices
\mathbf{V}_t	Velocity of t^{th} iteration
\mathbf{w}	Vector of misclosure
\mathbf{W}	Weights
\mathbf{W}_t	Updated weights
w	Weight vector in an SVM method
\mathbf{X}	Unknown parameters
\mathbf{X}^0	Approximate value of unknown parameters
$\hat{\mathbf{x}}$	The difference between the estimated values of the unknown vector and its initial values \mathbf{X}^0
$\hat{\mathbf{X}}$	Estimated unknown parameters
x_p	x-coordinate of points P
x	Observation in line equation F_1
x_o	x-coordinate of position center of the roof model primitives
X	Set of m -dimensional points
x_i	A m -dimensional point in a point set X
y	Label vector assigned to point set X through a classification task
y_i	The label class i
y_p	y-coordinate of points P
y	Observation in line equation F_1
y_o	y-coordinate of position centers of the roof model primitives
\mathbf{z}	Entry of softmax function
Z_{ridge}	The height of ridge lines
$Z_{eave/gutter}$	The height of eave lines
Z	The height value
\oplus	Dilation
\ominus	Erosion
\circ	Opening
\bullet	Closing

List of Symbols

∂	Partial derivative
Σ	An $m \times n$ diagonal matrix with the square roots of the eigenvalues (λ) of $\mathbf{A}^T \mathbf{A}$
$\ \cdot\ _2$	L^2 norm
$\ \cdot\ _1$	L^1 norm
$ \cdot $	Absolute value
$\mathbf{1}$	Indicator function
Δ	The differences between two variables
Θ^l	Parameters of the last layer of CNNs
Θ^{l-1}	Parameters of the previous layer of CNNs
$\mathcal{L}(\cdot)$	log-likelihood function
\mathcal{L}	log loss function
$\nabla(\cdot)$	Gradient
ν	The momentum which is the coefficient of weight from previous iteration in CNNs
τ	The learning rate for weighting the negative gradient in CNNs
α	Dirichlet parameter
η_t	The scheduling rate to reduce learning rate at the end of specific iterations in CNNs
β	An identity function in pre-trained ResNet
λ	Lagrange multiplier
λ_1	Minimum Eigenvalue
λ_2	Maximum Eigenvalue
λ_n	Minimum singular value of matrix Σ
$\phi(\cdot)$	A Kernel approximation function
$\bar{\phi}(\cdot)$	A Kernel approximation function
$\Phi(\cdot)$	A monotonically decreasing function
γ	A Kernel parameter
$\Gamma(\cdot)$	A Gamma function
ξ_i	Slack variable corresponding to point x_i in an SVM method
ω	Weight vector in an SVM method

ϵ	Threshold of sign matrix
$\epsilon_{\mathcal{H}}$	Threshold of Mean Curvature
$\epsilon_{\mathcal{K}}$	Threshold of Gaussian Curvature
ψ	A geometrical parameter of the primitives in the library which is in set of Ψ
Ψ	Geometrical parameters set of the roof model primitives
Ψ_{range}	Vector of range of parameters
$\rho(\cdot)$	Huber loss
μ	Mean value
$\hat{\mu}$	The estimated mean value
$\hat{\sigma}$	The estimated standard deviation
θ_1	Orientation of the extracted line using Hough transform
θ_2	Orientation of the rectangle side
ΔX	The differences between the X coordinate of the Hough line and the rectangle side
ΔY	The differences between the Y coordinate of the Hough line and the rectangle side
ΔZ	Height difference
Δh	Error or difference between two DSMs
$RM\hat{S}E$	The estimated Root mean square errors
$NMAD$	Normalized median absolute errors
$ \Delta H_{ridge} $	Height differences between ridge lines of 3D building model and reference data from DUPB
$ \Delta H_{eave} $	Height differences between eave lines of 3D building model and reference data from DUPB

List of Figures

1.1	Workflow of the multistage hybrid 3D building model reconstruction.	6
1.2	(a) Exemplary image covering an area of $2500m \times 2000m$ in central Munich acquired by the Worldview-2 satellite, (b) classification of the image using the proposed approach by Krauss <i>et al.</i> [19]	11
1.3	Parametric gable model (coordinate origin (X_θ, Y_θ) , orientation (θ) , slope of roof plane (α) , length (L) , width (W) , heights of gutter/eave $(H_{gutter/eave})$ and ridge lines (H_{ridge}))	21
2.1	Overview of our proposed building outlines extraction method.	32
2.2	(a) DSM-based building mask. (b) Building mask after refinement.	33
2.3	Illustration of building and non-building pixels as positive and negative pixel labels, respectively	35
2.4	(a) Original PAN image and (b) its refined mask, bright and dark green areas, together with the areas removed from the mask, red areas.	37
2.5	Line segment extraction process.	38
2.6	(a) Extracted line segments for an example building. (b) Three main orientations of the building depicted by magenta, green and blue colors. (c) Assignment of the line segments to the building's main orientations. (d) Splitting the building into several CHs. (e) Extracted line segments after regularization and adjustment. (f) Outline of the building after intersecting and connecting the line segments.	40
2.7	Process chain of the line segment regularization, where CH denotes a convex hull and MO indicates the number of main orientation classes c_i in each CH.	41
2.8	Fitting a model to a set of points using LS adjustment.	46

2.9	Each sub-figure shows a condition and its corresponding constraint for connecting the line segments and the result of the line segment connection. (a) Join parallel lines with a perpendicular distance larger than five pixels. (b) Join parallel lines with a perpendicular distance smaller than or equal to five pixels with the assumption, $l_2 > l_1, l_3$. (c) Join intersecting lines.	48
2.10	Orthorectified PAN images used in our experiments.	49
2.11	Refined masks (bright and dark green areas) together with the areas removed from the masks specified (red areas). (a) I_1 , (b) I_5 , (c) I_2 , (d) I_4	51
2.12	The Quality measure of the given and refined masks of the images $I_1 - I_{10}$ using SVM classification. Regarding training, 30 % of the pixels within the building boundaries, indicated by the masks, is sampled from images: (a) I_1 and I_2 , (b) I_1, I_2 and I_3 , (c) I_4, I_6 and I_8 , (d) I_1, I_3, I_4 and I_8	52
2.13	Extracted outlines of the buildings within the images (a) I_2 , (b) I_7 , (c) I_3 , (d) I_6 and (e) I_8 using our proposed approach.	54
2.14	Extracted outlines of the buildings within the images (a) I_4 , (b) I_1 , (c) I_5 , (d) I_9 and (e) I_{10} using our proposed method.	55
2.15	Failure case examples of the proposed approach.	56
2.16	Comparison of the building outlines based on our proposed line segment extraction method (red) and the DP algorithm (green), after the regularization step. The yellow lines depict the building footprints.	57
2.17	Illustrations of the PoLiS distance of the extracted (blue) building polygons to the reference (red) ones. (a) PoLiS = 5.19, (b) PoLiS = 1.60.	58
2.18	(a) PoLiS distance between the extracted and the reference building outlines when only the DSM-based masks specified (the red bars), refined masks (the green bars), and regularization and line segment adjustment (the blue bars) are used. (b) Degree of improvement to the building outlines extracted using the DSM-based masks specified when the refinement step (the green bars), and regularization and line segment adjustment steps (the blue bars) are employed.	59
3.1	Generalization and decomposition process	62
3.2	Illustration of common line segments and wrongly selected common line segments	64
3.3	Buffer zone around each line which has been introduced as a common line; (a) buffer's bins around a common line; (b) illustration of building mask points inside the buffer's bins	64
3.4	Aggregated footprint	65

3.5	Simplification rules. (a) Unidirected lines with small offset l_n , (b) extrusion, (c) approximately parallel lines and (d) corners	66
3.6	(a) Outline simplification red: Before simplification green: After simplification (b) rectified building outline green: before rectification blue: after rectification	67
3.7	(a) Footprint (b) aggregated footprint (c) red: before simplification green: after simplification	67
3.8	Direction of movement based on a footprint-based building mask . . .	69
3.9	Rectangle creation using a moving line segment (red: building outline, blue: moving line segment, yellow: buffer of line segment)	69
3.10	Rectangle-based decomposition. (a) Rectangle extraction based on building polygon line segments. (b) Illustration of rectangles (blue: redundant rectangles, red: relevant rectangles)	69
3.11	Selected rectangles	71
3.12	Rectangle selection. (a) Selected rectangles after first verification, (b) selected rectangles after second verification (red rectangles), (c) final result of selection after using (a) and (b) verifications	71
3.13	Illustration of building footprint generalization, decomposition and selection: (a) the generalized footprint, (b) the decomposed footprint into rectangular shapes (c) and the relevant rectangles selected . . .	72
3.14	Illustration of generalized extracted building outlines, (a) I_1 , (b) I_3 , (c) I_2 , (d) I_{10} , (e) I_8 , (f) I_6 , (g) I_7 , (h) I_4 , (i) I_9 , (j) I_5	73
3.15	Illustration of decomposed building outlines into rectangular shapes, (a) I_1 , (b) I_3 , (c) I_2 , (d) I_{10} , (e) I_8 , (f) I_6 , (g) I_7 , (h) I_4 , (i) I_9 , (j) I_5	74
3.16	Illustration of the relevant rectangles selected, (a) I_1 , (b) I_3 , (c) I_2 , (d) I_{10} , (e) I_8 , (f) I_6 , (g) I_7 , (h) I_4 , (i) I_9 , (j) I_5	75
3.17	Comparison of the PoLiS metric between building outlines and ground truth; green: before generalization, blue: after generalization	77
4.1	Roof model library (a) flat, (b) gable, (c) half-hip, (d) hip, (e), pyramid, and (f) mansard roof models	82
4.2	Illustration of different roof models with their PS image patch, DSM, latitudinal profile and longitudinal profile (Basic roof types)	83
4.3	Illustration of different roof models with their PS image patch, DSM, latitudinal profile and longitudinal profile (Combinatorial roof types)	84
4.4	Illustration of different roof models with their PS image patch, DSM, longitudinal and latitudinal profiles profile (Less frequent roof types)	85
4.5	Illustration of 3D profile, 2D longitudinal profile and 2D latitudinal profile of the satellite's DSM and LiDAR's DSM data for the various roof types	86

4.6	(a) PS image of a building roof, (b) building mask and overlapping skeleton points, (c) projected skeleton points on the PS image of a building roof, (d) extracting the patches for one skeleton point in a size of 56×56 pixels, (e) resize roof patch to the proper size of pretrained convolutional neural networks (CNNs) (224×224 pixels) .	88
4.7	The library of roof patches including six roof types	88
4.8	The network structure of the VGG 16 and VGG 19	93
4.9	(a) Plain net and (b) Residual net	94
4.10	Residual block (up and left), bottleneck layer (bottom and left), ResNet architectures (right)	95
4.11	Example of a drawback of image-based classification, (a) PS image, (b) image-based classification, (red: flat roof, green: gable roof, and blue: half-hip roof), c) DSM-based roof type classification, (gray: flat roof, white: non-flat roof)	97
4.12	Fine-tuning curve including training loss, validation loss, and top-1 and top-5 accuracy (a) VGGNet-16, (b) VGGNet-19, (c) ResNet-50, (d) ResNet-101 and (e) ResNet-152	103
4.13	Illustration of classified roof types (green: gable, blue: half-hip, red: flat, yellow: hip, light blue: mansard, pink: pyramid roofs, (left) the results of four different areas, (right) their related ground truths . . .	110
5.1	Workflow of the proposed method.	112
5.2	Geometrical parameters of the roof models	113
5.3	Groups of single and multiplane roof models	114
5.4	The computational domain of four-quadrant inverse tangent of elements of X and Y	115
5.5	(a) The decomposed rectangles, (b) the decomposed rectangles after imposing the rotation, (c) the decomposed rectangles, and (d) the decomposed rectangle after imposing the shift	116
5.6	Profiles acquisition; red lines are the measuring slices perpendicular to the width and green lines are the measuring slices perpendicular to the length of the rectangle. Blue points are intersection points of the profile and building mask	116
5.7	The profiles acquisition of the rectangle and their parameters: ds , rs , and l_{ext} . Plus shape points stand for pixels and blue lines show the line profiles.	117
5.8	The correction of the rectangle after imposing the shift: Blue rectangle	117
5.9	Decomposition of a rectangle after imposing the rules of roof type combination: The pyramid roof is shown in magenta (a) before decomposition and (b) after decomposition	118

5.10	Height discontinuities detection: (a) Longitudinal profile line (blue) and latitudinal profile line (red) drawn on the nDSM; (b) decomposition of the blue rectangle in (a) based on height discontinuity points (two magenta points in (a)); (c) the 3D point clouds projected in the 2D plane passing the longitudinal profile (red points); and (d) the height differences between the 3D point clouds along the longitudinal profile ΔZ ; three groups of points in magenta, blue and green. The height discontinuity points are shown in red dot squares. (e) The 3D point clouds drawn in the 2D plane passing the latitudinal profile (red points). (f) The height differences (the gradients) between the 3D point clouds along the latitudinal profile ΔZ ; two groups of points in magenta and blue. The height discontinuity points are shown in red dot squares.	120
5.11	Roof model orientation detection	121
5.12	Roof type refinement: (a) The result of the classification before refinement, (b) roof types after refinement; blue and yellow colors stand for half-hip and hip roofs, respectively	123
5.13	(a) Buffer of the roof eave height estimation. (b) Buffer of the roof ridge line height estimation	125
5.14	Overlapping (red dots) and neighboring (green dots) parts of rectangles in one building block	128
5.15	The parameters of the rectangle to find the overlapping sides which is shown in gray	129
5.16	Buffer of the hip side determination, (a) projected on the DSM, (b) projected on the image	131
5.17	Connecting roof types	133
5.18	Possible connecting of roof types by different combinations of the roof vertices and intersection points (green points) of the two mansard roofs	134
5.19	Illustration of the data-driven steps of 3D building model reconstruction for the first area (Area-1): (a) Decomposed 2D building outline into rectilinear shape, (b) detected height discontinuities (green lines), (c) classified roof types (red: flat, green: gable, blue: half-hip, yellow: hip, magenta: pyramid, cyan: mansard) and (d) extracted ridge lines (blue lines).	136
5.20	Illustration of 3D building reconstruction for Area-1: (a) 3D building models on the PAN image, (b) 3D building models	137
5.21	Illustration of the data-driven steps of 3D building model reconstruction for the second area (Area-2): (a) Decomposed 2D building outline into rectilinear shape, (b) detected height discontinuities (green lines), (c) classified roof types (red: flat, green: gable, blue: half-hip, yellow: hip, magenta: pyramid, cyan: mansard), (d) extracted ridge lines (blue lines).	138

5.22	Illustration of 3D building reconstruction for Area-2: (a) 3D building models on the PAN image, (b) 3D building models	139
5.23	Illustration of the data-driven steps of 3D building model reconstruction for the third area (Area-3): (a) Decomposed 2D building outline into rectilinear shape, (b) detected height discontinuities (green lines), (c) classified roof types (red: flat, green: gable, blue: half-hip, yellow: hip, magenta: pyramid, cyan: mansard), and (d) extracted ridge lines (blue lines)	140
5.24	Illustration of 3D building reconstruction for Area-3: (a) 3D building models on the PAN image, (b) 3D building models	141
5.25	Illustration of the data-driven steps of 3D building model reconstruction for the fourth area (Area-4): (a) Decomposed 2D building outline into rectilinear shape, (b) detected height discontinuities (green lines), (c) classified roof types (red: flat, green: gable, blue: half-hip, yellow: hip, magenta: pyramid, cyan: mansard), (d) extracted ridge lines (blue lines).	142
5.26	Illustration of 3D building reconstruction for Area-4: (a) 3D building models on the PAN image, (b) 3D building models	143
5.27	Different examples of 3D building models. PAN image (1st column), LiDAR data (2nd column), rectangles (3rd column) and 3D building model (4th column)	144
5.28	Different examples of 3D building models. PAN image (1st column), LiDAR data (2nd column), rectangles (3rd column) and 3D building model (4th column)	145
5.29	Over-detection of height discontinuities highlighted with green ellipses; Up: the error locations on the nDSM, bottom: the error locations on the 3D model	146
5.30	Inaccurate building footprint positions (red ellipse). Circular footprint (green circle) and conical roof shape (orange circle); left side: the error locations on the nDSM, right side: the error locations on the 3D model	146
5.31	Semi-ellipsoid roof types; left side: the error locations on the nDSM, right side: the error locations on the 3D model	147
5.32	Deficiencies of intersection and merging; left side: the error locations on the nDSM, right side: the error locations on the right side	147
5.33	Deficiencies of intersection and merging; left side: the error locations on the nDSM, right side: the error locations on the right side	148
5.34	Missing the building part; left side: the error locations on the nDSM, right side: the error locations on the right side	149
5.35	The model-nDSM for four areas of Munich	150
5.36	The comparison of the three profiles of the model-nDSM, the LiDAR-nDSM and the satellite-nDSM for four case studies of Munich.	151

5.37	The errors maps (left) and the histograms of errors with the corresponding theoretical normal for four areas of Munich (right)	152
5.38	Comparing the performance of three roof type classification strategies, including roof model fitting to nDSM data of the buildings, applying a deep learning method to MS images and the fusion of the two other strategies. The results are the average of the four areas in Munich. . .	153
5.39	Evaluation of the ridge line orientation detection averaged over the four areas of Munich, where PA and PP denote the middle line parallel and perpendicular to the roof's longest lines, relatively, PO denotes the ridge points for pyramid roofs and WR indicates the orientations of the flat roof buildings which have no ridge lines.	154
5.40	The building blocks and their numbers for Area-2	156
5.41	The building blocks and their numbers for Area-4	157
A.1	Datasets (a) DSM of satellite imagery, (b) PAN image, (c) MS image, (D) PS image	170
A.2	LiDAR based DSM	171

List of Tables

2.1	Specifications of the images used in our experiments.	50
2.2	Parameters Used in Our Approach	50
2.3	Comparing the PoLiS distance of the building outlines extracted using our method and DP	60
3.1	Quality assessment of the generalized building footprint	79
3.2	Quality assessment of extracted building outline for ten areas (I_1 to I_{10})	80
4.1	The distribution of the training and test sets used in the experiment	89
4.2	The model hyper-parameters of VGGNet-16 and 19 and of ResNet 50/101/152	96
4.3	Surface type determined based on the surface curvature sign	99
4.4	The results of fine-tuned CNN models	105
4.5	Results of SVM on the DeCAFs	106
4.6	Conventional classification approaches	107
4.7	Comparison of <i>Compl.</i> , <i>Corr.</i> , $F_{1-score}$, and <i>Acc.</i> for the best results of DeCAFs, the best results of fine-tuned CNN models and conventional models, such as RF and SVM, with RGB features	108
4.8	Classification of flat and non-flat roofs based on the DSM and image-based method (ResNet-152)	108
5.1	The confusion matrix of the classification for all four areas (Chapter 4)	122
5.2	prior probabilities of roof models	122
5.3	Predefined parameters and their corresponding ranges, units and step sizes	127
5.4	Equations of accuracy measures	155
5.5	Accuracy measures (in meter) based on the differences between model-DSM and reference LiDAR-DSM	155

5.6	Accuracy measures (in meter) based on the differences between satellite-nDSM and reference LiDAR-nDSM	156
5.7	The difference between ridge/eave lines of the building model and DUPB-based reference data for Area-2	158
5.8	The difference between ridge/eave lines of the building model and DUPB-based reference data for Area-4	159
B.1	A summary of model-driven based building modeling approaches according to data sources, representation, level of details, scale and level of automation	174
B.2	A summary of data-driven based building modeling approaches according to data sources, representation, level of details, scale and level of automation	175
B.3	A summary of hybrid-based building modeling approaches according to data sources, representation, level of details, scale and level of automation	176

Bibliography

- [1] M. Roupé, *Development and implementations of virtual reality for decision-making in urban planning and building design*. PhD thesis, Chalmers University of technology, Gothenburg, Sweden, 2013.
- [2] U. N. department of economic and social affairs, 16 May 2018. <https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects>.
- [3] G. Wölfle, R. Hoppe, and F. Landstorfer, “Radio network planning with ray optical propagation models for urban, indoor, and hybrid scenarios,” in *Proc. IEEE Wireless*, pp. 1–8, September 1999.
- [4] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool, “3D urban scene modeling integrating recognition and reconstruction,” *International Journal of Computer Vision*, vol. 78, pp. 121–141, July 2008.
- [5] J. Lee and S. Zlatanova, “A 3D data model and topological analyses for emergency response in urban areas,” in *Geospatial information technology for emergency response* (J. L. S. Zlatanova, ed.), pp. 159–184, CRC Press, 2008.
- [6] F. Lafarge, *Modèles stochastiques pour la reconstruction tridimensionnelle d’environnements urbains*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2007.
- [7] E. Tunc, F. Karsli, and E. Ayhan, “3D city reconstruction by different technologies to manage and reorganize the current situation,” in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, pp. 443–448, July 2004.
- [8] G. S. N. Perera and H.-G. Maas, “Cycle graph analysis for 3D roof structure modelling: Concepts and performance,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, pp. 213–226, July 2014.

- [9] I. Tomljenovic, B. Höfle, D. Tiede, and T. Blaschke, “Building extraction from airborne laser scanning data: an analysis of the state of the art,” *Remote sensing*, vol. 7, pp. 3826–3862, April 2015.
- [10] U. Weidner and W. Förstner, “Towards automatic building extraction from high-resolution digital elevation models,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 50, pp. 38–49, August 1995.
- [11] F. Rottensteiner, J. Trinder, S. Clode, and K. Kubik, “Automated delineation of roof planes from lidar data,” in *ISPRS Workshop Laser scanning 2005*, vol. 36, pp. 221–226, ISPRS Workshop groups WG III/3, III/4, V/3, 2005.
- [12] H. Arefi and P. Reinartz, “Building reconstruction using dsm and orthorectified images,” *Remote Sensing*, vol. 5, no. 4, pp. 1681–1703, 2013.
- [13] L. Yong and W. Huayi, “Adaptive building edge detection by combining lidar data and aerial images,” in *the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, pp. 197–202, July 2008.
- [14] J.-R. Małgorzata and H.-G. Maas, “Geometric refinement of als-data derived building models using monoscopic aerial images,” *Remote sensing, MDPI*, vol. 282, pp. 1–16, March 2017.
- [15] E. Kwak, *Automatic 3D building model generation by integrating LiDAR and aerial images using a hybrid approach*. PhD thesis, Department of Geomatics Engineering, Calgary, Alberta, 2013.
- [16] F. Remondino, M. G. Spera, E. Nocerino, and F. Nex, “State of the art in high density image matching,” *The Photogrammetric Record*, vol. 29, pp. 144–166, June 2014.
- [17] L. Yan, L. Fei, C. Chen, Z. Ye, and R. Zhu, “A multi-view dense image matching method for high-resolution aerial imagery based on a graph network,” *Remote Sensing*, vol. 8, p. 799, September 2016.
- [18] K. Bittner, F. Adam, S. Cui, M. Körner, and P. Reinartz, “Building footprint extraction from vhr remote sensing images combined with normalized dsms using fused fully convolutional networks,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 8, pp. 2615–2629, 2018.
- [19] T. Krauss, B. Sirmacek, H. Arefi, and P. Reinartz, “Fusing stereo and multi-spectral data from WorldView-2 for urban modeling,” in *Proc. SPIE Defense, Security and Sensing*, vol. 8390, pp. 1–15, June 2012.

-
- [20] N. Haala and M. Kada, “An update on automatic 3d building reconstruction,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, pp. 570–580, November 2010.
- [21] Y.-H. Tseng and S. Wang, “Semi-automated building extraction based on csg model-image fitting,” *Photogrammetric Engineering and Remote Sensing*, vol. 69, pp. 171–180, February 2003.
- [22] S. Wang and Y.-H. Tseng, “Model-based building reconstruction from topographic maps and lidar data,” in *Asian Conf. Remote Sensing*, November 2007.
- [23] S. O. Elberink and G. Vosselman, “Building reconstruction by target based graph matching on incomplete laser data: Analysis and limitations,” *Sensors*, vol. 9, no. 8, pp. 6101–6118, 2009.
- [24] M. Buyukdemircioglu, S. Kocaman, and U. Isikdag, “Semi-automatic 3d city model generation from large-format aerial images,” *ISPRS International Journal of Geo-Information*, vol. 7, no. 9, p. 339, 2018.
- [25] C. Brenner, “Modeling 3d objects using weak csg primitives,” in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, pp. 1085–1090, October 2004.
- [26] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, “Structural approach for building reconstruction from a single dsm,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 32, pp. 135–147, November 2010.
- [27] F. Lafarge and C. Mallet, “Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation,” *International journal of computer vision*, vol. 99, pp. 69–85, February 2012.
- [28] B. Xiong, M. Jancosek, S. O. Elberink, and G. Vosselman, “Flexible building primitives for 3d building modeling,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 101, pp. 275–290, March 2015.
- [29] F. Tarsha-Kurdi, T. Landes, P. Grussenmeyer, and M. Koehl, “Model-driven and data-driven approaches using lidar data: Analysis and comparison,” in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 3/W49A)*, pp. 87–92, September 2007.
- [30] Y. Zheng, Q. Weng, and Y. Zheng, “A hybrid approach for three-dimensional building reconstruction in indianapolis from lidar data,” *Remote Sensing*, vol. 9, p. 310, March 2017.

- [31] T. Partovi, R. Bahmanyar, T. Krauß, and P. Reinartz, “Building roof component extraction from panchromatic satellite images using a clustering-based method,” in *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3, pp. 247–252, September 2014.
- [32] B. Sirmacek, H. Taubenböck, P. Reinartz, and M. Ehlers, “Performance evaluation for 3-D city model generation of six different DSMs from air- and spaceborne sensors,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 1, pp. 59–70, 2012.
- [33] P. d’Angelo, M. Lehner, T. Krauss, D. Hoja, and P. Reinartz, “Towards automated DEM generation from high resolution stereo satellite images,” vol. 37, pp. 1137–1342, July 2008.
- [34] D. Hoja, P. Reinartz, and M. Lehner, “DSM generation from high resolution satellite imagery using additional information contained in existing DSM,” vol. 36, May 2005.
- [35] D. Grigillo, M. Kosmatin Fras, and D. Petrovič, “Automated building extraction from ikonos images in suburban areas,” *International journal of remote sensing*, vol. 33, pp. 5149–5170, July 2012.
- [36] Z. G. Bafghi, J. Tian, P. d’Angelo, and P. Reinartz, “A new algorithm for void filling in a DSM from stereo satellite images in urban areas,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, pp. 55–61, June 2016.
- [37] T. Partovi, R. Bahmanyar, T. Krauß, and P. Reinartz, “Building outline extraction using a heuristic approach based on generalization of line segments,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, pp. 933–947, October 2017.
- [38] T. Partovi, F. Fraundorfer, S. Azimi, D. Marmanis, and P. Reinartz, “Roof type selection based on patch-based classification using deep learning for high resolution satellite imagery,” vol. 42, June 2017.
- [39] T. Partovi, F. Fraundorfer, R. Bahmanyar, H. Huang, and P. Reinartz, “Automatic 3-d building model reconstruction from very high resolution stereo satellite imagery,” *Remote Sensing*, vol. 11, July 2019.
- [40] G. Zhou and X. Zhou, “Seamless fusion of lidar and aerial imagery for building extraction,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 7393–7407, April 2014.

-
- [41] K. Karantzas and N. Paragios, "Large-scale building reconstruction through information fusion and 3-d priors," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2283–2296, 2010.
- [42] V. Verma, R. Kumar, and S. Hsu, "3D building detection and modeling from aerial lidar data," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2213–2220, IEEE, June 2006.
- [43] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds," *IEEE Transactions on geoscience and remote sensing*, vol. 48, pp. 1554–1567, November 2010.
- [44] S. Sun and C. Salvaggio, "Aerial 3D building detection and modeling from airborne lidar point clouds," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, pp. 1440–1449, May 2013.
- [45] D. Bulatov, F. Rottensteiner, and K. Schulz, "Context-based urban terrain reconstruction from images and videos," in *Proceedings of the XXII ISPRS Congress of the International Society for Photogrammetry and Remote Sensing ISPRS Annals, Melbourne, Australia*, vol. 25, p. 3, Aug-Sep 2012.
- [46] P. Dorninger and N. Pfeifer, "A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds," *Sensors*, vol. 8, no. 11, pp. 7323–7343, 2008.
- [47] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Conditional random fields for lidar point cloud classification in complex urban areas," in *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, vol. I-3, pp. 263–268, June 2012.
- [48] M. Gerke and J. Xiao, "Fusion of airborne laserscanning point clouds and images for supervised and unsupervised scene classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 78–92, December 2014.
- [49] M. Gerke and J. Xiao, "Supervised and unsupervised MRF based 3D scene classification in multiple view airborne oblique images," vol. 2, pp. 25–30, November 2013.
- [50] T. Hermosilla, L. A. Ruiz, J. A. Recio, and J. Estornell, "Evaluation of automatic building detection approaches combining high resolution images and lidar data," *Remote Sensing*, vol. 3, pp. 1188–1210, June 2011.
- [51] F. Rottensteiner, J. Trinder, S. Clode, and K. Kubik, "Using the dempster-shafer method for the fusion of lidar data and multi-spectral images for building detection," *Information fusion*, vol. 6, pp. 283–300, December 2005.

- [52] A. P. Dempster, “A generalization of bayesian inference,” *Journal of the Royal Statistical Society, Series B*, vol. 30, no. 2, pp. 205–247, 1968.
- [53] G. Shafer, *A mathematical theory of evidence*, vol. 42. Princeton university press, 1976.
- [54] K. Bittner, S. Cui, and P. Reinartz, “Building extraction from remote sensing data using fully convolutional networks,” vol. 42, June 2017.
- [55] G. Vosselman and H. Maas, *Airborne and Terrestrial Laser Scanning*. Whittles Publishing, 2010.
- [56] B. C. Matei, H. S. Sawhney, S. Samarasekera, J. Kim, and R. Kumar, “Building segmentation for densely built urban regions using aerial LiDAR data,” in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, June 2008.
- [57] J. Park, I. Lee, Y. Choi, *et al.*, “Automatic extraction of large complex buildings using lidar data and digital maps,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVI, p. 148–154, Citeseer, July 2006.
- [58] M. Kada and A. Wichmann, “Sub-surface growing and boundary generalization for 3D building reconstruction,” in *Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Melbourne, Australia*, vol. 25, pp. 233–238, September 2012.
- [59] V. Hough and C. Paul, “Method and means for recognizing complex patterns,” December 1962. US Patent 3,069,654.
- [60] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, pp. 381–395, June 1981.
- [61] M. Gerke, C. Heipke, and B.-M. Straub, “Building extraction from aerial imagery using a generic scene model and invariant geometric moments,” in *Remote Sensing and Data Fusion over Urban Areas, IEEE/ISPRS Joint Workshop 2001*, pp. 85–89, IEEE, February 2001.
- [62] M. Dutter, “Generalization of building footprints derived from high resolution remote sensing data,” 2007.
- [63] H. Arefi, J. Engels, M. Hahn, and H. Mayer, “Approximation of building boundaries,” in *Urban Data Management Systems (UDMS) Workshop, Stuttgart*, pp. 37–46, CRC Press, 2007.

-
- [64] J. Avbelj, *Fusion of Hyperspectral Images and Digital Surface Models for Urban Object Extraction*. PhD thesis, Technical University of Munich, 2015.
- [65] M. Brédif, O. Tournaire, B. Vallet, and N. Champion, “Extracting polygonal building footprints from Digital Surface Models: A fully-automatic global optimization framework,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 77, pp. 57 – 65, March 2013.
- [66] P. J. Green, “Reversible Jump Markov Chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, vol. 82, pp. 711–732, December 1995.
- [67] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, pp. 112–122, December 1973.
- [68] G. Vosselman, “Building reconstruction using planar faces in very high density height data,” in *International Archives of Photogrammetry and Remote Sensing*, vol. 32, pp. 87–94, International Society for Photogrammetry and remote sensing, 1999.
- [69] A. Sampath and J. Shan, “Building boundary tracing and regularization from airborne LiDAR point clouds,” *Photogrammetric Engineering & Remote Sensing*, vol. 73, pp. 805–812, July 2007.
- [70] R. A. Jarvis, “On the identification of the convex hull of a finite set of points in the plane,” *Information processing letters*, vol. 2, pp. 18–21, March 1973.
- [71] E. Orthuber and J. Avbelj, “3D building reconstruction from lidar point clouds by adaptive dual contouring,” in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, p. 157, Copernicus GmbH, March 2015.
- [72] Q.-Y. Zhou and U. Neumann, “Fast and extensible building modeling from airborne LiDAR data,” in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, no. 7, pp. 1–7, ACM, November 2008.
- [73] M. Morgan and A. Habib, “Interpolation of LiDAR data and automatic building extraction,” in *ACSM-ASPRS Annual conference proceedings*, pp. 432–441, April 2002.
- [74] M. Sester and H. Neidhart, “Reconstruction of building ground plans from laser scanner data,” in *11th AGILE International Conference on Geographic Information Science in University of Girona, Spain*, pp. 1–11, 2008.

- [75] R. Guercke and M. Sester, “Building footprint simplification based on Hough transform and least squares adjustment,” in *Proceedings of the 14th Workshop of the ICA commission on Generalisation and Multiple Representation. Paris, France*, vol. 30, July 2011.
- [76] L.-C. Chen, T.-A. Teo, Y.-C. Shao, Y.-C. Lai, and J.-Y. Rau, “Fusion of lidar data and optical imagery for building modeling,” in *International Archives of Photogrammetry and Remote Sensing*, vol. 35, pp. 732–737, January 2004.
- [77] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, pp. 679–698, November 1986.
- [78] A. Alharthy and Bethel, “Building extraction and reconstruction from lidar data,” in *Proceedings of ASPRS annual conference*, vol. XXXIII, pp. 18–26, 2002.
- [79] J.-Y. Rau and B.-C. Lin, “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*, vol. 66, pp. S13–S27, December 2011.
- [80] G. Sohn and I. Dowman, “Data fusion of high-resolution satellite imagery and lidar data for automatic building extraction,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, pp. 43–63, May 2007.
- [81] N. Haala, C. Brenner, and K. heinrich Anders, “3D urban GIS from laser altimeter and 2D map data,” in *International Archives of Photogrammetry and Remote Sensing*, CiteSeer, 1998.
- [82] C. Lin and R. Nevatia, “Building detection and description from a single intensity image,” *Computer vision and image understanding*, vol. 72, pp. 101–121, November 1998.
- [83] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon, “Automatic line matching and 3D reconstruction of buildings from multiple views,” in *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, vol. 32, pp. 69–80, June 1999.
- [84] W. Zhang, H. Wang, Y. Chen, K. Yan, and M. Chen, “3D building roof modeling by optimizing primitive’s parameters using constraints from LiDAR data and aerial imagery,” *Remote Sensing*, vol. 6, pp. 8107–8133, August 2014.
- [85] C. Brenner, “Towards fully automatic generation of city models,” in *International Archives of Photogrammetry and Remote Sensing*, vol. 33, pp. 84–92, International Society for Photogrammetry and remote sensing, July 2000.

-
- [86] I. Suveg and G. Vosselman, "Reconstruction of 3D building models from aerial images and maps," *ISPRS Journal of Photogrammetry and remote sensing*, vol. 58, pp. 202–224, January 2004.
- [87] G. Vosselman and Dijkman, "3D building model reconstruction from point clouds and ground plans," in *International archives of photogrammetry remote sensing and spatial information sciences*, vol. 34, pp. 37–44, NATURAL RESOURCES CANADA, October 2001.
- [88] A. Habib, E. Kwak, and M. Al-Durgham, "Model-based automatic 3D building model generation by integrating lidar and aerial images," in *Archives of Photogrammetry, Cartography and Remote Sensing*, vol. 22, pp. 187–200, 2011.
- [89] M. Awrangjeb, C. Zhang, and C. S. Fraser, "Automatic extraction of building roofs using LIDAR data and multispectral imagery," *ISPRS journal of photogrammetry and remote sensing*, vol. 83, pp. 1–18, September 2013.
- [90] T. Partovi, H. Huang, T. Krauß, H. Mayer, and P. Reinartz, "Statistical building roof reconstruction from WorldView-2 stereo imagery," in *the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, pp. 161–167, Copernicus GmbH, March 2015.
- [91] L. Zebedin, A. Klaus, B. Gruber-Geymayer, and K. Karner, "Towards 3D map generation from digital aerial images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, pp. 413–427, August 2006.
- [92] G. S. N. Perera, *Automated generation of 3D building models from dense point clouds and aerial photos*. PhD thesis, Technische Universität Dresden, Institute of Photogrammetry and Remote Sensing, 2015.
- [93] X. Zhuo, F. Fraundorfer, F. Kurz, and P. Reinartz, "Optimization of OpenStreetMap building footprints based on semantic information of oblique UAV images," *Remote Sensing*, vol. 10, p. 624, April 2018.
- [94] M. Kada and L. McKinley, "3D building reconstruction from LiDAR based on a cell decomposition approach," in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, p. W4, Citeseer, September 2009.
- [95] A. Henn, G. Gröger, V. Stroh, and L. Plümer, "Model driven reconstruction of roofs from sparse lidar point clouds," *ISPRS Journal of photogrammetry and remote sensing*, vol. 76, pp. 17–29, December 2013.
- [96] Y. Zheng and Q. Weng, "Model-driven reconstruction of 3-d buildings using lidar data," *IEEE geoscience and remote sensing letters*, vol. 12, pp. 1541–1545, March 2015.

- [97] B. Vallet, M. Pierrot-Deseilligny, D. Boldo, and M. Brédif, “Building footprint database improvement for 3D reconstruction: A split and merge approach and its evaluation,” *ISPRS journal of photogrammetry and remote sensing*, vol. 66, pp. 732–742, July 2011.
- [98] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, “Automatic building extraction from dems using an object approach and application to the 3D-city modeling,” *ISPRS Journal of photogrammetry and remote sensing*, vol. 63, no. 3, pp. 365–381, 2008.
- [99] M. Ortner, X. Descombes, and J. Zerubia, “Building outline extraction from digital elevation models using marked point processes,” *International Journal of Computer Vision*, vol. 72, pp. 107–132, October 2007.
- [100] C. Poullis and S. You, “Photorealistic large-scale urban city model reconstruction,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 654–669, August 2008.
- [101] H. Huang, C. Brenner, and M. Sester, “A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data,” *ISPRS Journal of photogrammetry and remote sensing*, vol. 79, pp. 29–43, May 2013.
- [102] W. Nguatem, M. Drauschke, and H. Mayer, “Roof reconstruction from point clouds using importance sampling,” in *Ann. Photogrammetry, Remote Sens. Spat. Inf. Sci. II-3 W*, vol. 3, pp. 73–78, October 2013.
- [103] H.-G. Maas and G. Vosselman, “Two algorithms for extracting building models from raw laser altimetry data,” *ISPRS Journal of photogrammetry and remote sensing*, vol. 54, pp. 153–163, July 1999.
- [104] R. Schnabel, R. Wahl, and R. Klein, “Efficient ransac for point-cloud shape detection,” in *Computer graphics forum*, vol. 26, pp. 214–226, Blackwell Publishing, June 2007.
- [105] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, “Segmentation of point clouds using smoothness constraint,” in *International archives of photogrammetry, remote sensing and spatial information sciences, ISPRS commission V symposium*, vol. 36, pp. 248–253, September 2006.
- [106] B. Ameri and D. Fritsch, “Automatic 3D building reconstruction using plane-roof structures,” in *Annual ASPRS Conference, Washington DC, USA*, p. 22–26, May 2000.
- [107] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer, “Extended ransac algorithm for automatic detection of building roof planes from lidar data,” *The Photogrammetric Journal of Finland*, vol. 21, no. 1, pp. 97–109, 2008.

-
- [108] J. Jung, Y. Jwa, and G. Sohn, “Implicit regularization for reconstructing 3D building rooftop models using airborne lidar data,” *Sensors*, vol. 17, p. 621, March 2017.
- [109] H. Lin, J. Gao, Y. Zhou, G. Lu, M. Ye, C. Zhang, L. Liu, and R. Yang, “Semantic decomposition and reconstruction of residential scenes from lidar data,” *ACM Transactions on Graphics (TOG)*, vol. 32, pp. 66:1–10:66, July 2013.
- [110] H. Wang, W. Zhang, Y. Chen, M. Chen, and K. Yan, “Semantic decomposition and reconstruction of compound buildings with symmetric roofs from LiDAR data and aerial imagery,” *Remote Sensing*, vol. 7, pp. 13945–13974, October 2015.
- [111] F. Taillandier, “Automatic building reconstruction from cadastral maps and aerial images,” in *International Archives of Photogrammetry and Remote Sensing*, vol. 36, p. W24, August 2005.
- [112] J. Overby, L. Bodum, E. Kjems, and P. Iisoe, “Automatic 3D building reconstruction from airborne laser scanning and cadastral data using hough transform,” in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, pp. 296–301, January 2004.
- [113] G. Sohn, X. Huang, and V. Tao, “Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data,” *Photogrammetric Engineering & Remote Sensing*, vol. 74, pp. 1425–1438, November 2008.
- [114] H. Jibrini, M. Pierrot-Deseilligny, N. Paparoditis, and H. Maitre, “Automatic building reconstruction from very high resolution aerial stereopairs using cadastral ground plans,” *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 33, pp. 213–219, January 2000.
- [115] J. Milde, Y. Zhang, C. Brenner, L. Plümer, and M. Sester, “Building reconstruction using a structural description based on a formal grammar,” in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, pp. 227–232, July 2008.
- [116] S. J. Oude Elberink, *Acquisition of 3D topography: automated 3D road and building reconstruction using airborne laser scanner data and topographic maps*. PhD thesis, University of Twente, April 2010.
- [117] F. Rottensteiner and C. Briese, “Automatic generation of building models from lidar data and the integration of aerial images,” in *ISPRS, Dresden*, vol. XXXIV, October 2003.

- [118] E. Schwalbe, H.-G. Maas, and F. Seidel, “3D building model generation from airborne laser scanner data using 2D GIS data and orthogonal point cloud projections,” in *Proceedings of ISPRS WG III/3, III/4*, vol. 3, pp. 12–14, Citeseer, September 2005.
- [119] Q.-Y. Zhou and U. Neumann, “2.5 d dual contouring: A robust approach to creating building models from aerial lidar point clouds,” in *European conference on computer vision*, pp. 115–128, Springer, 2010.
- [120] F. Taillandier and R. Deriche, “Automatic buildings reconstruction from aerial images: a generic bayesian framework,” in *Proceedings of the XXth ISPRS Congress, Istanbul, Turkey*, pp. 12–23, July 2004.
- [121] M. Durupt and F. Taillandier, “Automatic building reconstruction from a digital elevation model and cadastral data: an operational approach,” in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, pp. 142–147, Citeseer, July 2006.
- [122] B. Xiong, S. O. Elberink, and G. Vosselman, “A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds,” *ISPRS Journal of photogrammetry and remote sensing*, vol. 93, pp. 227–242, July 2014.
- [123] F. Rottensteiner, “Consistent estimation of building parameters considering geometric regularities by soft constraints,” in *In International Archives of Photogrammetry; Remote Sensing and Spatial Information Sciences: Bonn, Germany*, vol. 36, 2006.
- [124] L. Duan and F. Lafarge, “Towards large-scale city reconstruction from satellites,” in *European Conference on Computer Vision*, pp. 89–104, Springer, October 2016.
- [125] P. Martino and B. J. Atli, “A new approach for the morphological segmentation of high-resolution satellite imagery,” *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, vol. 39, pp. 309–320, February 2001.
- [126] D. G. Lowe, “Object recognition from local scale-invariant features,” in *The proceedings of the seventh IEEE international conference on Computer vision*, vol. 2, pp. 1150–1157, Ieee, September 1999.
- [127] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, pp. 91–110, November 2004.
- [128] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, “WLD: A robust local image descriptor,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1705–1720, September 2010.

-
- [129] P. d' Angelo, "Automatic orientation of large multitemporal satellite image blocks," in *International Symposium on Satellite Mapping Technology and Application*, vol. 38, pp. 1–7, September 2013.
- [130] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB (Second Edition)*. Gatesmark Publishing, 2009.
- [131] W. Niemeier, *Ausgleichsrechnung: Statistische Auswertemethoden*. Walter de Gruyter, September 2008.
- [132] W. Gander, "Algorithms for the qr decomposition," pp. 1–27, April 1980.
- [133] W. Gander, "The singular value decomposition," pp. 1–18, December 2008.
- [134] "Fitting lines, rectangles and squares in the plane." <http://people.inf.ethz.ch/arbenz/MatlabKurs/node85.html>. Accessed: 2008-09-24.
- [135] M. Sester and H. Neidhart, "Reconstruction of building ground plans from laser scanner data," in *Proc. International Conference on Geographic Information Science (AGILE)*, pp. 1–11, 2008.
- [136] J. Avbelj, R. Müller, and R. Bamler, "A metric for polygon comparison and building extraction evaluation," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, pp. 170–174, July 2015.
- [137] J.-H. Haunert and A. Wolff, "Optimal and topologically safe simplification of building footprints," in *Proceedings of the 18th sigspatial international conference on advances in geographic information systems*, pp. 192–201, ACM GIS, November 2010.
- [138] M. Kada and F. Luo, "Generalisation of building ground plans using half-spaces," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, 2006.
- [139] T. Commandeur, "Footprint decomposition combined with point cloud segmentation for producing valid 3D models," March 2012.
- [140] K. Buchin, W. Meulemans, and B. Speckmann, "A new method for subdivision simplification with applications to urban-area generalization," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 261–270, ACM, November 2011.
- [141] M. Peter, N. Haala, and D. Fritsch, "Preserving ground plan and facade lines for 3D building generalization," in *the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS), Beijing, China*, no. B2, pp. 481–486, July 2008.

- [142] M. Sester, “Optimization approaches for generalization and data abstraction,” *International Journal of Geographical Information Science*, vol. 19, no. 8-9, pp. 871–897, 2005.
- [143] X. Zhang, Y. Fu, A. Zang, L. Sigal, and G. Agam, “Learning classifiers from synthetic data using a multichannel autoencoder,” in *arXiv preprint arXiv:1503.03163*, pp. 1–11, 2015.
- [144] F. Alidoost and H. Arefi, “Knowledge based 3D building model recognition using convolutional neural networks from lidar and aerial imageries,” in *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 41, pp. 833–840, June 2016.
- [145] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, “Deep learning earth observation classification using ImageNet pretrained networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, pp. 105–109, January 2016.
- [146] H. Li, H. Lu, Z. Lin, X. Shen, and B. Price, “LCNN: low-level feature embedded CNN for salient object detection,” *arXiv preprint arXiv:1508.03928*, pp. 1–9, August 2015.
- [147] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 6232–6251, July 2016.
- [148] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, M. Hasan, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, “The history began from alexnet: A comprehensive survey on deep learning approaches,” *ArXiv*, vol. abs/1803.01164, pp. 1–38, 2018.
- [149] C. neural network, “Convolutional neural network — Wikipedia, the free encyclopedia,” 2013. en.wikipedia.org/wiki/Convolutional_neural_network.
- [150] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv*, vol. 1408.5093, June 2014.
- [151] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. of the NIPS’12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol. 1, pp. 1097–1105, December 2012.
- [152] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv*, vol. 1409.1556, September 2014.

-
- [153] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, June 2015.
- [154] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, June 2016.
- [155] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, “Deep learning for remote sensing image classification: A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, pp. 1–17, May 2018.
- [156] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, “Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery,” *Remote Sensing*, vol. 7, pp. 14680–14707, November 2015.
- [157] D. Gardner and D. Nichols, “Multi-label classification of satellite images with deep learning,” pp. 1–7, 2017.
- [158] G. J. Scott, M. R. England, W. A. Starns, R. A. Marcum, and C. H. Davis, “Training deep convolutional neural networks for land-cover classification of high-resolution imagery,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, pp. 549–553, February 2017.
- [159] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “DeCAF: A deep convolutional activation feature for generic visual recognition,” in *Proceedings of the 31st International Conference on Machine Learning* (E. P. Xing and T. Jebara, eds.), vol. 32 of *Proceedings of Machine Learning Research*, (Beijing, China), pp. 647–655, PMLR, 22–24 Jun 2014.
- [160] S. Valero, J. Chanussot, and P. Gueguen, “Classification of basic roof types based on vhr optical data and digital elevation model,” in *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, vol. 4, pp. IV–149, IEEE, July 2008.
- [161] H. Arefi, *From LIDAR Point Clouds to 3D Building Models*. PhD thesis, Institute for Applied Computer Science - Bundeswehr University Munich, 2009.
- [162] P. J. Besl and R. C. Jain, “Segmentation through variable-order surface fitting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 167–192, March 1988.
- [163] C. Brenner, *Dreidimensionale Gebäuderekonstruktion aus digitalen Oberflächenmodellen und Grundrissen*. PhD thesis, Bauingenieur- und Vermessungswesen der Universität Stuttgart, 2000.

- [164] O. Ruiz, S. Arroyave, and D. Acosta, “Fitting of analytic surfaces to noisy point clouds,” *American Journal of Computational Mathematics, Volume 3, Issue 1A*, pp 18-26, vol. 3, pp. 18–26, April 2013.
- [165] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, pp. 211–252, December 2015.
- [166] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [167] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios, “Building detection in very high resolution multispectral data with deep learning features,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1873–1876, IEEE, July 2015.
- [168] P. J. Huber *et al.*, “Robust estimation of a location parameter,” *The annals of mathematical statistics*, vol. 35, pp. 73–101, March 1964.
- [169] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, *A parametric model for automatic 3D building reconstruction from high resolution satellite images*. PhD thesis, INRIA, Sophia Antipolis, Cedex France, 2006.
- [170] J. Tian, *3D change detection from high and very high resolution satellite stereo imagery*. PhD thesis, Universität Osnabrück, Fachbereichs Mathematik / Informatik, October 2013.
- [171] R. Lehmann, “3 σ -rule for outlier detection from the viewpoint of geodetic adjustment,” *Journal of Surveying Engineering*, vol. 139, pp. 157–165, November 2013.
- [172] J. Höhle and M. Höhle, “Accuracy assessment of digital elevation models by means of robust statistical methods,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, pp. 398–406, July 2009.
- [173] L. Zhang, *Automatic digital surface model (DSM) generation from linear array images*. PhD thesis, ETH Zurich, 2005.
- [174] P. Reinartz, R. Müller, M. Lehner, and M. Schroeder, “Accuracy analysis for DSM and ortho-images derived from SPOT HRS stereo data using direct georeferencing,” *ISPRS journal of photogrammetry and remote sensing*, vol. 60, pp. 160–169, May 2006.

-
- [175] H. Hirschmuller, “Stereo processing by semi-global matching and mutual information,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, pp. 328–341, February 2008.
- [176] P. Michalis and I. Dowman, “A model for along track stereo sensors using rigorous orbit mechanics,” in *Proc. of International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, pp. 303–309, January 2005.
- [177] “Satellite imaging corporation.” <https://www.satimagingcorp.com/satellite-sensors/worldview-2/>. Accessed: 2012-11-21.
- [178] P. d’Angelo, “Image matching and outlier removal for large scale DSM generation,” in *Proc. of the Geomatics Conference and Symposium of Commission I, ISPRS Convergence in Geomatics (CGC & ISPRS), Calgary, AB, Canada*, pp. 1–5, Citeseer, June 2010.
- [179] P. d’Angelo and P. Reinartz, “DSM based orientation of large stereo satellite image blocks,” in *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 39, pp. 209–214, July 2012.
- [180] P. d’Angelo, A. Uttenthaler, S. Carl, F. Barner, and P. Reinartz, “Automatic generation of high quality DSM based on IRS-P5 Cartosat-1 stereo data,” in *ESA Living Planet Symposium, Bergen, Special Publication SP*, June 2010.
- [181] G. Grohman, G. Kroenung, and J. Strebeck, “Filling SRTM voids: The delta surface fill method,” *Photogrammetric Engineering and Remote Sensing*, vol. 72, no. 3, pp. 213–216, 2006.
- [182] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, April 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [183] R. Bahmanyar, *Conception and Assessment of Semantic Feature Descriptors for Earth Observation Images*. PhD thesis, Technische Universität München, 2016.