# RELATION-BASED MODELLING OF POINT CLOUD FOR 3D CITY RECONSTRUCTION

YUAN LI

PhD

The Hong Kong Polytechnic University

2020

The Hong Kong Polytechnic University
Department of Land Surveying and Geo-Informatics

# Relation-based Modelling of Point Cloud for 3D City Reconstruction

Yuan LI

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

August 2019

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

______________________ (Signed)

LI, Yuan (Name of student)

# Abstract

Point cloud data obtained from different platforms, such as aerial laser scanning (ALS), mobile laser scanning (MLS) and photogrammetric point clouds generated from dense image matching (DIM), can intuitively present the three-dimensional (3D) geometric features of objects and are therefore important data sources for 3D city reconstruction. In recent years, with rising demand for the development of digital or smart cities, a great deal of research has investigated the semantic interpretation of point clouds and 3D city reconstruction from such data sources. However, most of these studies are applicable to only relatively simple urban scenes with low-rise buildings. Considering the rapid development and importance of city planning and management, modern cities, especially metropolises such as Hong Kong, have an urgent need for effective 3D city reconstruction methods.

However, the complexity of urban scenes (e.g., dense environments, various types of objects and high-rise buildings with manifold structures) in modern cities and the inevitable defects of point clouds (e.g., noise, loss of data and density anisotropy) make the automatic modelling of point clouds a challenging task. To overcome these difficulties, this research investigated the multiple relations contained in point clouds and exploited them for point cloud interpretation and 3D city reconstruction. Specifically, the multiple relations include geometric, contextual and topological relations. Geometric relations refer to local homogeneities of geometric properties, such as density isotropy, normal consistency, planarity, linearity and scattering. Contextual relations are related to neighbouring or adjacent relationships that could be associated with the specific labels assigned to the entities. Topological relations guarantee topological correctness during the generation of watertight and manifold 3D models that conform to CityGML. These multiple relations are comprehensively incorporated in the point cloud modelling process in three stages—segmentation, classification and 3D reconstruction.

Based on the assumption that ground objects can be regarded as combinations of

different simple shapes, the segmentation of point clouds partitions objects into groups of linear, planar and scatter shapes. The segmentation method first eliminates outliers with a filter that is robust to the varying point density and then generates supervoxels with adaptive sizes based on the homogeneities at the point level. Relations between the supervoxels are then derived and used to cluster adjacent supervoxels with similar geometric properties into structural components. This segmentation method interprets the point clouds on the geometric level and helps to provide essential clues for subsequent semantic interpretation. This is of particular importance when the point clouds (such as MLS data) present abundant details of the objects.

In the classification stage, structural information presenting the relations between structural components is derived at various scales. Such information can be of great help in distinguishing between objects with global or local similarities. In this research, structural, geometric and contextual information is comprehensively incorporated and encoded into a conditional random field (CRF) to make unary and pairwise inferences. High-order potentials defined upon regions independent of connection relationships are also introduced into the CRF to eliminate regional label noise. The classification finally outputs a point cloud with semantic labelling that is spatially smooth.

In the 3D reconstruction stage, points labelled as building are clustered into individual buildings and treated as inputs to produce polygonal 3D models. To avoid complex topological computation, a space-partition-and-approximation strategy is used. The building surface is first approximated by a set of planar primitives that are refined based on several geometric relation-based rules. With these planar primitives, the space occupied by the bounding box of the building is partitioned into non-overlapping convex cells based on a half binary space partition tree. The 3D space occupied by the building can be approximated by cells that are inside the building, and the interfaces between the inside and outside cells constitute the surface of the final building model. To ensure optimal selection of the inside cells, topological relations are extracted as interface facets and intersection edges, and are introduced into a global energy function, which can be solved as a linear programming problem with binary integer variables. The surface components of the building are generated from the selected cells and each is assigned a specific surface type defined in CityGML. The relationships between the surface components, e.g., adjacency, parallelism and perpendicularity, are determined based on the relationships between the cell, facet and edge complexes.

Experiments with point clouds from three representative data sources, including two MLS point clouds (in Paris and Hong Kong) and a photogrammetric point cloud (in Hong Kong), were carried out to evaluate the performances of the proposed methods in various scenarios. Nine and eleven different classes were recognised from the laser scanning point clouds with overall accuracies of 97.13% and 95.79%, respectively, indicating the effectiveness of the proposed classification method. For the photogrammetric point cloud, the classification result for a specific class, *building*, was evaluated and found to have a considerably good result, with an $F_1$-score of 82.40%. The buildings extracted from the photogrammetric data were further used to generate 3D building models in CityGML format via the proposed reconstruction method. The reconstruction results were qualitatively and quantitatively compared with the results of previous studies, and the comparisons suggested that the proposed method in this research performed best in terms of robustness and producing regular and geometrically accurate building models, with an average root-mean-square error of less than 0.9 m.

This research investigates the use of multiple relations in the pipeline of segmentation, classification and modelling of unordered point clouds for 3D city reconstruction. The developed pipeline shows promising ability to interpret point clouds and reconstruct 3D building models in complex urban scenes. In addition, it has high levels of automation and efficiency. The developed methods advance the current 3D city modelling technology from point cloud data with more automation and better performance. The final output of the 3D city models in the CityGML format can facilitate their use in various applications. The presented research and developments are significant for 3D city reconstruction and modelling, which will facilitate the construction of spatial data infrastructure for a smart city and have great potential to support applications in various domains, such as urban planning and design, urban management, and urban environmental studies.

# Publications Arising from the Thesis

**Journal Papers:**

[1] **Y. Li**, B. Wu, and X. Ge, 2019. Structural segmentation and classification of mobile laser scanning point clouds with large variations in point density. ***ISPRS Journal of Photogrammetry and Remote Sensing***, 153:151-165, doi: 10.1016/j.isprsjprs.2019.05.007. (*A Journal*)

[2] **Y. Li** and B. Wu, 2018. Analysis of rock abundance on lunar surface from orbital and descent images using automatic rock detection. ***Journal of Geophysical Research – Planets***, 123(5):1061-1088, doi: 10.1029/2017JE005496. (*A Journal*)

[3] Q. Zhu, **Y. Li,** H. Hu, and B. Wu, 2017. Robust point cloud classification based on multi-level semantic relationships for urban scenes. ***ISPRS Journal of Photogrammetry and Remote Sensing***, 129:86-102, doi: 10.1016/j.isprsjprs.2017.04.022. (*A Journal*)

[4] X. Ge, B. Wu, **Y. Li**, and H. Hu, 2019. A multi-primitive-based hierarchical optimal approach for semantic labeling of ALS point clouds. ***Remote Sensing***, 11: 1243, doi:10.3390/rs11101243. (*A Journal*)

[5] L. Xie, Q. Zhu, H. Hu, B. Wu, **Y. Li**, and Y. Zhang, 2018. Hierarchical regularization of building boundaries in noisy aerial laser scanning and photogrammetric point clouds. ***Remote Sensing***, 10: 1996, doi: 10.3390/rs10121996. (*A Journal*)

[6] B. Wu, J. Huang, **Y. Li**, Y. Wang, and J. Peng, 2018. Rock abundance and crater density in the candidate Chang'E-5 landing region on the Moon. ***Journal of Geophysical Research – Planets***, 123(12): 3256-3272, doi: 10.1029/2018JE005820. (*A Journal*)

**Conference Paper:**

[1] **Y. Li** and B. Wu, 2019. Structural segmentation of point clouds with varying density based on multi-size supervoxels. ***ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences***, IV-2/W5, 389-396, doi: 10.5194/isprs-annals-IV-2-W5-389-2019.

# Acknowledgments

The study described in this thesis was supported by grants from the National Natural Science Foundation of China, Project No. 41671426.

Pursuing a PhD in Hong Kong has been one of the most valuable experiences of my life. It is like a long journey full of challenges, and it would have been much more difficult for me to finish this journey without all the help, support and advice I have received in the past three years.

Foremost, I would like to express my deepest gratitude and respect to my supervisor, Dr Bo Wu, for his strong support and hard criticism of my work. What I have learnt from him is more than academic knowledge. His general guidance and wise advice on conducting scientific research helped me in the past years and will continue to help me in my new career.

I would like to acknowledge all of the academic and administrative staff members of LSGI for their support of this research. In particular, I want to thank Prof Zhilin Li and Dr Wei Yao for their professional advice and comments during the confirmation of my study. I also want to express my thanks to all the professors at PolyU from whom I took classes.

I wish to thank Dr Han Hu and Dr Xuming Ge, who gave me a lot of valuable advice and feedback. I also want to thank all the other groupmates in our PhotoLab for their unconditional help. Thanks to Dr Lei Ye, Mr Linfu Xie, and Mr Wai Chung Liu for the methodological discussion and advice, Ms Mst Ilme Faridatul for proofreading of this dissertation, and Ms Yiran Wang for her help in both study and life throughout my time in Hong Kong.

Finally, I am grateful to my beloved family for their constant understanding, support and pride in me. Many thanks are also given to Mr Kai Wang for all of his encouragement and companionship.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1 Introduction

## 1.1 Research Background

Three-dimensional (3D) models are the fundamental data that constitute the spatial data infrastructure for a digital city or smart city (Cocchia, 2014). 3D point clouds obtained from light detection and ranging (LiDAR) or photogrammetry through dense image matching (DIM) can intuitively show the 3D geometric features of urban objects with intensity or textural information, and they have therefore been essential data sources for 3D city reconstruction in past decades. Mature technology can produce triangulated irregular network (TIN) models (mesh models) from unordered point clouds (Früh and Zakhor, 2003). However, such TIN models of continuous surfaces cannot be used for complicated 3D analyses of smart cities due to the lack of semantic and topological information. Smart city applications require interoperable 3D models that conform to a standard format (e.g., CityGML) (Gröger and Plümer, 2012) for complicated applications. The generation of such 3D models still involves tremendous manual interference with computer-aided software systems (Hu et al., 2003). Although particular interactive tools (Arikan et al., 2013; Nan et al., 2010) have been developed for modelling individual buildings, the interactive process is still time-consuming and labour-intensive.

Recent advances in automatic 3D city reconstruction have revealed that enriching the point clouds or TIN models with semantic segmentation and then reconstructing each segment, is an effective and scalable paradigm for large-scale reconstruction (Blaha et al., 2016; Lafarge and Mallet, 2012; Poullis, 2013; Verdie et al., 2015; Xiong et al., 2015). However, the semantic segmentation or classification of point clouds is non-trivial work because of the complexity of urban scenes (Blaha et al., 2016; Zhu et al., 2017), especially when the datasets have high spatial resolution but suffer from heavy noise. A general trend is to adopt a hierarchical classification strategy that starts with the geometric segmentation of the point clouds (Gerke and Xiao, 2014; Vosselman et al., 2017; Xu et al., 2014), which provides an opportunity for incorporating multiple types of relations (Zhu et al., 2017) into the process of 3D reconstruction from point clouds.

The geometric segmentation or partition of point clouds is useful for interpreting scenes at the geometric level and provides essential clues for subsequent semantic classification and interpretation. This is especially true for terrestrial laser scanning (TLS) and mobile laser scanning (MLS) data, where the object surfaces are presented in detail because of the close scanning distances. However, because of the obvious defects of point cloud data, such as noise, missing data and density anisotropy, existing segmentation and classification methods are not resilient and thus require extensive drudgery in the form of manual quality control, especially for DIM point clouds and mobile laser scanning data (Nex and Gerke, 2014; Yokoyama et al., 2013).

The defects of point clouds, especially the loss of data, also cause problems in the reconstruction of buildings (Xiong, 2014). Missing data on the vertical surfaces of objects, mainly the façades of buildings, could be a serious problem in aerial laser scanning (ALS) data. Therefore, most of the previous work on 3D building reconstruction from ALS data only focused on the reconstruction of building rooftops (Chen et al., 2014; Hu et al., 2018b; Lafarge and Mallet, 2012; Sampath and Shan, 2010) and only produced building models in 2.5D (Zhou and Neumann, 2010, 2011). Although DIM point clouds generated from oblique images through multi-view stereo (MVS) pipelines (Furukawa and Ponce, 2010; Vu et al., 2012) provide more information on vertical surfaces than ALS data, missing data remains an issue where there are occlusions between closely distributed high-rise buildings.

In addition, the architectural styles of buildings can vary greatly by culture, location and time (Relph, 2016), which makes it impossible to fit the point clouds with a predefined model library as in the work of Kada and McKinley (2009) and Poullis and You (2009b). The complexity of buildings also increases the difficulty of topological computation, resulting in crack effects in the final results (Poullis, 2013; Poullis and You, 2009a; Xie et al., 2018) or needing extra work to automatically or manually correct the topological errors in the models (Xiong et al., 2014). Even when the topological errors are enforced to be corrected, e.g., multiple intersection lines are enforced to join at the same point, the results might violate certain geometric relations with respect to regularity, such as co-planarity, parallelism and orthogonality. A number of methods have been proposed toward to preserve the topological relations while reconstructing the building rooftops (Chen et al., 2017; Chen et al., 2014), but they cannot capture the topological relations between two roof components when there is a large height-jump.

All of the above methods only focused on the reconstruction of building rooftops; in contrast, the reconstruction of true 3D building models has not been intensively investigated. The reasons for this are two-folds. First, the points on building façades are much sparser than those on roofs or might be completely missing due to occlusion, as shown in Figure 1.1 (a). Second, at the LOD (level of detail)-2 scale, the building façades are likely to be assumed to be non-essential to facilitate the characteristics of buildings, but this is not the case for many modern buildings (as shown in Figure 1.1 (b)). A strategy to generate true 3D models of buildings with a variety of appearances is to avoid topological computations during reconstruction but to use a set of space units (e.g., 3D faces, boxes and polyhedral cells) obtained through space partitioning to model the building surface (Boulch et al., 2014; Nan and Wonka, 2017; Verdie et al., 2015). Topological relations between these 3D faces have been used to guarantee the model to be watertight (Nan and Wonka, 2017); however, the use of topological relationships as constraints has not been systematically studied.

(a) Missing data on building façades in LAS (left) and DIM (right) point clouds.

(b) Buildings with vertical characteristics

Figure 1.1 Challenges in reconstruction of true 3D building models from point clouds.

Objects, especially man-made objects, always feature various relations, including geometric, semantic and topological relations. These relations form the complexity of the actual world and help to facilitate hypotheses about the objects, when the observation data suffer from quality issues. Introducing such relations into the framework of point cloud modelling can improve the understanding of the 3D data. Using this strategy, this research investigates the use of multiple relations in the pipeline of modelling unordered point clouds to overcome the defects of the point clouds and to produce true 3D models with correct topological relationships in CityGML format so that the models can be used for various 3D GIS applications.

## 1.2 Research Objectives

The aim of this research is to provide an automatic framework to segment and classify point clouds with semantic labels, and then based on the labelled point clouds to produce 3D models that conform to CityGML (Gröger and Plümer, 2012) for further 3D GIS applications.. The objectives of this research are listed as follows.

(1) To develop a point cloud segmentation method that interprets the point clouds on the geometric level and decomposes objects into structural components with shape labels;

(2) To develop a point cloud classification method that interprets the 3D scenes at the semantic level, and that is able to process point clouds with high complexity and various defects (e.g., noise, density anisotropy, missing data and data inaccuracy) from different sources;

(3) Based on the segmented and classified point clouds, to develop an automatic and robust reconstruction method that produces true 3D models of buildings that are watertight, manifold, regularised and in CityGML format.

(4) To systematically validate the developed methods and evaluate the performances with actual point clouds from different sources in representative urban scenes (e.g., Hong Kong).

## 1.3 Innovation and Contributions of This Research

The main novelty of this research lies in the development and incorporation of multiple relations, including geometric, contextual, structural and topological relations, in point cloud modelling. These relations help to overcome the defects of the point cloud data, and to facilitate the hypotheses about the complexity of the actual world. The contributions of this research are as follows:

1) Multiple relations between different primitives and different levels are introduced to decompose objects into different structures. The density isotropy of the points is the basis of a novel noise filter that is adaptive to point clouds with varying point densities; the spatial and spectral homogeneities at the point level facilitate the generation of multi-size supervoxels, where local geometric shapes are well preserved in terms of linearity, planarity and scattering; and the contextual relations between the supervoxels guarantee that the segmentation is spatially smooth and remains physically meaningful.

2) Structural information is derived from the shape labels of the structural components during the semantic interpretation. The structural information is

captured by multiple contextual relations and presents the structural environment of a specific object and the structural differences at different scales. These relations affect on both the inference and the refinement of the classification. Hence, the developed method can be applied to point clouds obtained from different sources and can be effective in dense scenes that have multiple classes of objects, even with local or global similarities.

3) By adopting a space-partition-and-approximation strategy, the topological relations between 3D space units are extracted and used as constraints in the generation of 3D building models. The method developed in this research converts complex 3D topological computations into simple topological relations in lower dimensions. Therefore, the method is robust to buildings with various architectural styles and is able to produce watertight, manifold and regularised models, which are further converted into CityGML models for 3D GIS applications.

## 1.4 Outline of the Dissertation

This thesis consists of seven chapters. Following the introduction chapter, the remainder of this thesis is organised as follows.

Chapter 2 reviews previous work related to the first three objectives of this research: segmentation, classification and 3D building reconstruction of/from point clouds, in sequence. Short summaries of the various relations and their usages, and the previous methods used in related work, are given at the end of this chapter.

Chapter 3 describes the segmentation method developed in this research. This method starts with a noise filter that can adaptively remove noise in the point cloud based on density isotropy. Then the proposed two segmentation steps, which are the generation of supervoxels based on point relations and the generation of structural components based on supervoxel relations, are given in detail.

Chapter 4 presents the three aspects in the classification stage: extraction of discriminative features from contextual relations, training/inferencing and global energy optimisation with full-range contextual information. Structural information derived from the structural components is abstracted as contextual relations and used to establish a high-order CRF model for inference and optimisation.

Chapter 5 demonstrates the process of producing 3D CityGML building models from point clouds. First, the building bounding box is partitioned into convex 3D cells with the planar primitives extracted from the point clouds and refined based on geometric relations. Then, the topological relations between the basic 3D cells are extracted as facets and edges, and are used as constraints to select optimal cells that approximate the building surface. Finally, the surface models are converted into CityGML models according to a set of knowledge-based rules.

Chapter 6 describes the experimental evaluations using three datasets. For each dataset, a brief description of the dataset is first given. Then, the experimental results and corresponding analyses of the newly developed methods are described.

Chapter 7 concludes and discusses the work of this research. Following a summary of the achievements in this research, conclusions are drawn, and recommendations are made for future research.

Figure 1.2 shows the relationship of the chapters in this dissertation.

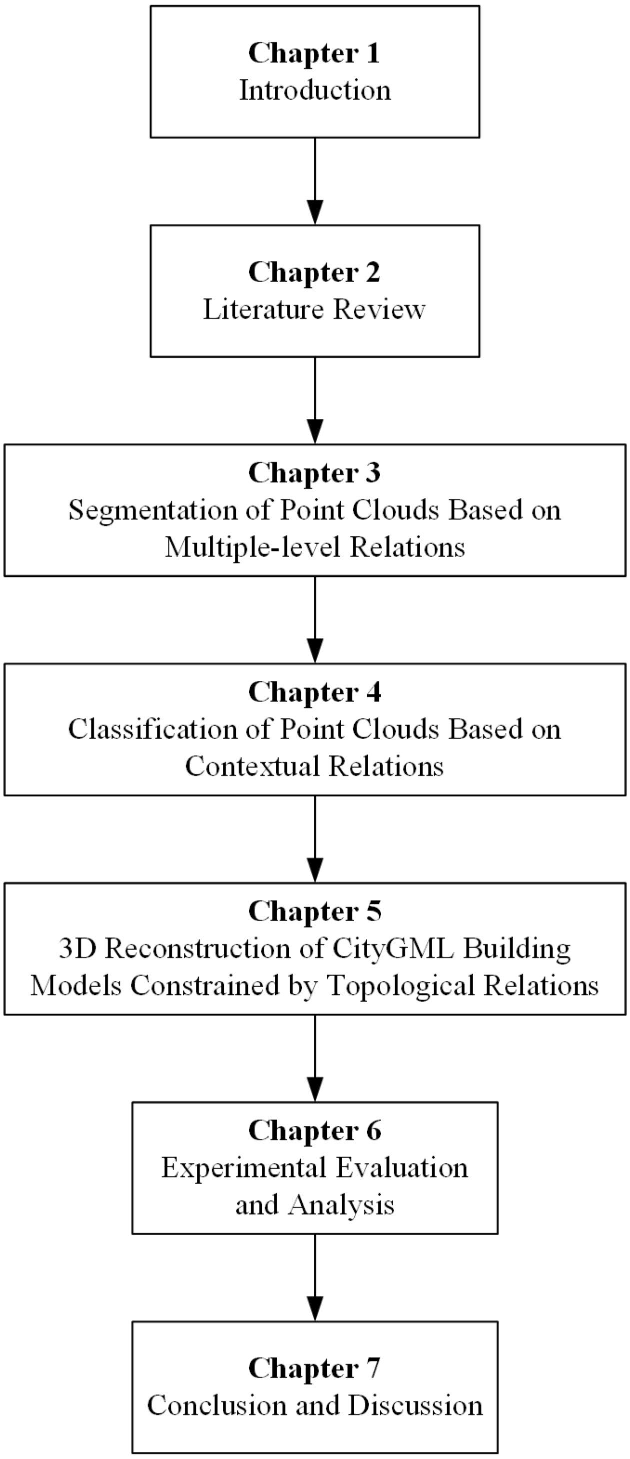



Figure 1.2 Schematic structure of this dissertation.

# Chapter 2 Literature Review

The modelling of point clouds is a broad research area. In this research, it specially involves three aspects: segmentation, classification and 3D reconstruction. The first two aspects interpret the point clouds at geometric and semantic levels, respectively, and the last aspect outputs models as a fundamental data infrastructure for smart cities. The following reviews the state-of-the-art developments in these three aspects.

## 2.1 Segmentation of Point Clouds

Geometric segmentation or partition is to understand the 3D point clouds at the geometric level. It could be a perquisite step for point cloud classification (Aijazi et al., 2013; Vosselman et al., 2017; Xu et al., 2014; Zhu et al., 2017) and 3D reconstruction of buildings (Verdie et al., 2015; Xie et al., 2017; Xiong et al., 2015). According to the purposes and strategies, there are different categories of segmentation methods that are detailed described in (Grilli et al., 2017; Nguyen and Le, 2013; Vo et al., 2015). The following presents the segmentation paradigms related to this research particularly.

### 2.1.1 Shape Detection

Random Sample Consensus (RANSAC) initially introduced by Fischler and Bolles (1981) is one of the most adopted shape detection paradigms in 3D data processing (Schnabel et al., 2007; Wang et al., 2018). It fits the input data with a set of mathematical model parameters of particular shapes, e.g., planes, cylinders, spheres, cones and tori. The RANSAC-based methods can detect the approximate shapes from the 3D point clouds, even in the presence of outliers up to 50% (Zolanvari et al., 2018). Hough Transform (Hough, 1962) is another fitting-based shape detection method, which was first introduced to detect linear features from the images. In the work of

Vosselman and Dijkman (2001), the Hough Transform was extended into 3D space for the detection of planar surfaces of buildings. Later, the 3D Hough Transform was further extended to recognize cylinders and spheres (Rabbani and Van Den Heuvel, 2005; Tarsha-Kurdi et al., 2007; Vosselman et al., 2004). These methods have shown quite outstanding performances on detecting particular shapes from buildings that do not have significant protrusions or complex exterior structures. However, as the target shapes are limited only to simple mathematic models, these fitting-based methods are able to detect free-formed shapes.

Region growing is another common method used to detect planes or other surfaces that have consistent features. This method first selects a set of seeds to initialize the features of the regions, and then iteratively includes the neighbouring entities that meet particular criteria into a region, while keeping refining the features of the regions (e.g., normal and curvature) with the newly included entities. Aiming to fit many planes, Vosselman et al. (2004) developed a surface growing method with a brute-force seed selection approach. This method was later used in the classification of ALS point clouds as the first step to generate fundamental entities to be classified (Gerke and Xiao, 2014; Vosselman et al., 2017; Xu et al., 2014). However, as surface growing can only capture planar objects, extra segmentation method, such as mean-shift (Comaniciu and Meer, 2002), was adopted to cluster the non-planar segments for better segmentation of the point clouds.

### 2.1.2 Voxelisation

Voxelisation is to partition point clouds into supervoxels (or voxels) with similar sizes and shapes. "Supervoxel" is an extension of "superpixel" (Achanta et al., 2012; Ren and Malik, 2003) from 2D to 3D. It refers to a small cluster of unorganised points generated through space partitioning, and the points in each cluster maintain the original geometries but together constitute a regular shape. Recently, supervoxels have been widely used for the interpretation of large-scale point clouds (Dong et al., 2018;

Kang and Yang, 2018; Luo et al., 2018; Zhu et al., 2017) for three reasons. First, the local homogeneity is well preserved within the supervoxels. Secondly, supervoxels provide explicit adjacent relationships rather than vague neighbouring relationships among unorganised points. And thirdly, exploiting supervoxels instead of individual points can significantly reduce the computational load and time.

Early works on voxelisation were mainly based on the distances between point. For instance, Aijazi et al. (2013) defined a supervoxel's size using radius search. Zhou et al. (2012) and Babahajiani et al. (2015) clustered points within a given radius into one supervoxel and limited the number of points to avoid bestriding object boundaries. Other properties, such as colours and normals (Lim and Suter, 2009; Papon et al., 2013), were also introduced into the generation of supervoxels to restrict the size of supervoxels.

Generally, the seeds of supervoxels were selected based on an octree structure with a fixed resolution, such as the widely adopted voxel cloud connectivity segmentation (VCCS) (Papon et al., 2013) and its extension (Zhu et al., 2017). Sizes of supervoxels generated by such methods are almost the same, resulting in supervoxels located in areas with sparse density containing insufficient points for feature extraction. This could be especially serious for TLS and MLS point clouds where the variations in point density are great. To obtain supervoxels that best present the local properties, Yang et al. (2015) generated two-scale supervoxels and merged them based on a set of prior rules. This method made the supervoxels somewhat adaptive to point density, but the knowledge-based rules could be inapplicable to other scenes or data. Instead of octree with a fixed resolution, Li and Sun (2018) selected supervoxel seeds with an adaptive octree and generated supervoxels by region growing. Lin et al. (2018) proposed a toward better boundary preserved (TBBP) method that formalised the supervoxel segmentation as a subset selection problem. Instead of the size of supervoxels, this method used the number of supervoxels as a direct constraint to select representative points, so that the final supervoxels would be free of size constraint and well preserve the object boundaries. However, the adaptability to the boundaries of this method also causes a problem that the supervoxels are quite sensitive to linear and scatter features.

### 2.1.3 Graph-based Partitioning

The graph-based partitioning converts the point cloud into a graphical model, where the nodes of the graph correspond to the points and the edges corresponds to the neighbouring relationships between the points (Nguyen and Le, 2013). The graph-based segmentation of point clouds is an extension of the paradigm in the 2D images (Felzenszwalb and Huttenlocher, 2004). But unlike the regularly organised image data structure, the point cloud is unorganised and discrete, and therefore the neighbouring relationships are generally determined by k-nearest neighbours (KNN) (Golovinskiy and Funkhouser, 2009) or by an optimal neighbourhood (Landrieu et al., 2017; Weinmann et al., 2017) defined by a minimum entropy function (Demantke et al., 2011).

The point cloud is then segmented based on the cuts of the graph. The min-cut based segmentation proposed by Golovinskiy and Funkhouser (2009) is a well-known method. In this method, two extra nodes, namely source and sink, respectively corresponding to the foreground and background, were introduced into the graph and each node corresponding to the point was connected to the source and sink by two edges. A penalty function was defined to encourage smooth segmentation where the foreground was weakly connected to the background. This method can automatically segment an object out from the point cloud by each cut, but the location of the object is required as prior knowledge.

The graph-based segmentation can also be cast into a labelling problem with multiple labels. For instance, Landrieu and Simonovsky (2018) computed four shape descriptors, namely linearity, planarity, scattering and verticality, from the optimal neighbourhood of each point and encoded them into the nodes of an underlying graphical model. The geometric segmentation was then converted into a global energy optimisation problem defined as in Guinard and Landrieu (2017), which was solved via $\ell_0$-cut (Landrieu and Obozinski, 2017). Rather than retrieving individual objects, this method decomposed objects into simple but homogeneous parts, that could be used for further semantic interpretation.

The graph-based segmentation methods can effectively interpret the geometries of complex scenes, even in the presence of high-level noise and density anisotropy. However, because of the great data volume of points, the point-wise feature computation and labelling process cannot run real time (Nguyen and Le, 2013) and it requires vast amounts of memory space to establish the fundamental graphical model. A segmentation method combining different paradigms was proposed by Dong et al. (2018). This method first divided the point cloud into multiscale planar supervoxels and points corresponding to non-planar supervoxels, as basic units. Then a hybrid region growing algorithm was exploited to merge adjacent units to generate initial planes, which were further enriched and refined by optimizing the global energy via extended $\alpha$-expansion (Delong et al., 2012). The fundamental graphical model of this method was built on the basis of initial planes, making the global energy optimisation much more efficient. However, this method only focused on the segmentation of planar surfaces and therefore inaccurate results could appear in non-planar areas.

## 2.2 Classification of Point Clouds

The purpose of classification is to interpret the point clouds at the semantic level, which generally refers to a process of assigning each point a semantic label, such as *buildings*, *cars*, *trees* and so forth. Generally, the steps constituting a classification framework can be summarised as primitive definition, feature extraction, inference and refinement. The specific steps are variable to different methods. For instance, methods adopting deep learning algorithms (LeCun et al., 2015) could be free of extracting handcrafted features and some methods do not refine the inference results obtained from the classifiers (Guo et al., 2011; Mallet et al., 2011; Weinmann et al., 2015a; Zhang et al., 2016b). The following reviews the contents related to the steps that are generally involved in a classification framework.

### 2.2.1 Primitive Definition

Individual points can be directly fed as the primitives for a point-wise classification framework. In this case, an appropriate neighbourhood needs to be defined for each point for feature extraction and the determination of adjacent relationships between points. The neighbourhood is generally an assemble of 3D points within a small local region, such as a sphere defined by fixed radius (Lee and Schenk, 2002) or *k* nearest points (Golovinskiy and Funkhouser, 2009) or nearest points within an optimal size (Hackel et al., 2016; Wang et al., 2015; Yang and Dong, 2013), a cylinder defined by fixed-radius (Chehata et al., 2009) or k nearest points projected on 2D plane (Niemeyer et al., 2014). Some other work also organised the neighbouring relationships with the aid of triangular structures such as Delaunay TIN (Sánchez-Lopera and Lerma, 2014) and Voronoi Graph (Landrieu and Simonovsky, 2018).

Based on the assumption that points have homogeneous properties are likely to belong to the same object, segment-based classification methods have been proposed. A straightforward strategy is to make the segments containing as many homogeneous points as possible, by using algorithms like RANSAC, 3D Hough transform, region growing or graph-based method (Landrieu and Simonovsky, 2018; Vosselman et al., 2017; Xiong et al., 2011). Hybrid primitives of individual points and point clusters at different scales were combined in some previous studies (Gerke and Xiao, 2014; Xu et al., 2014; Zhang et al., 2016b). Another strategy is to segment the point clouds into supervoxels. The supervoxels were either directly used to be classified (Kang and Yang, 2018; Luo et al., 2018; Zhu et al., 2017) or used to produce larger segments for the classification (Aijazi et al., 2013).

### 2.2.2 Feature Extraction

After the definition of primitives, discriminative features are then extracted from these primitives (segments of point cloud or neighbourhoods defined for individual points). 3D geometric properties related to height, normal and derived from the covariance matrix (e.g., linearity, planarity, scattering, eigenentropy and so forth) are the most considered features in a great deal of previous studies (Landrieu et al., 2017; Weinmann et al., 2015a; Weinmann et al., 2017; Yang et al., 2015). The Fast Point Feature histograms (FPFHs) descriptor (Rusu et al., 2009) is also a well-known 3D feature that encodes the local surface geometry around a point. Adjacent information has also been investigated to derive discriminative features. For instance, Zhu et al. (2017) regarded the number of adjacent supervoxels sharing consistent normal vectors as a discriminative feature to distinguish between building and vegetation. Sánchez-Lopera and Lerma (2014) defined an angular classifier that separated building points from vegetation and other small objects based on the surrounding ground points. By constructing a neighbourhood sphere centred at the centroid of a specific region, Xiong et al. (2011) defined what was above, below and adjacent to each region and appended contextual features with respect to such relations to feature vectors of regions. The contextual features derived from geometries were widely considered in classification frameworks based on graphical models, where the contextual features were encoded to the edges to capture the possibility of two connected nodes belonging to the same object (Niemeyer et al., 2014; Vosselman et al., 2017).

Beside discriminative features retrieved from geometries, features derived from other sources, such as colour and intensity (Kang and Yang, 2018), also help to distinguish between objects. Fusing point clouds with images (Cao et al., 2011; Gerke and Xiao, 2014) is a common way to obtain extra colour information for feature extraction. Mallet et al. (2008) and Mallet et al. (2011) investigated the potential of echo-based and full-waveform features derived from full-waveform data in addition to the traditional geometric features, and found that two features computed from the radiometric calibration of the full-waveform data significantly contributed to improving the accuracy of classification.

### 2.2.3 Inference

The inference is to compute the probability or conditional probability of a primitive belonging to a specific class based on the features extracted from the primitive. Generally, the inference strategies can be grouped into unsupervised and supervised ones.

Unsupervised inference does not require training data, but certain prior knowledge about the classes in the presented scenes should be needed. The prior-knowledge is generally about the value distributions of discriminative features on different classes, for eaxample, buildings should have larger elevation values than ground or clutters (Lafarge and Mallet, 2012). Based on such prior-knowledge, Lafarge and Mallet (2012), Verdie et al. (2015) and Zhu et al. (2017) defined a set of formulas to compute the probabilities corresponding to different classes. Rather than formulas, Yang et al. (2015) defined a set of rules that directly classified the primitive based on its geometric features and the geometric features of adjacent primitives. The unsupervised classification is simple and easy to implement, but it requires accurate prior knowledge of objects in the scenes and this would limit the scalability of such methods with respect to other data or scenarios.

The supervised inference involves a process of learning. It first learns and trains a mathematic model with the training examples and then uses this model to predict the conditional probability of the target data. The learning schemes can be grouped into instance-based learning, rule-based learning, probabilistic learning, max-margin learning, ensemble learning and deep learning (Weinmann et al., 2015a). The first five ones can be regarded as classic machine learning schemes, including a great number of algorithms that have been widely adopted by many point cloud classification methods, such as Gaussian maximum-likelihood (Weinmann et al., 2014), support vector machine (Mallet et al., 2011; Mallet et al., 2008; Yang and Dong, 2013), random forest (RF) (Chehata et al., 2009; Hackel et al., 2016; Landrieu et al., 2017; Niemeyer et al., 2014) and adaptive boosting (Lodha et al., 2007; Zhang et al., 2016a; Zhang et al., 2016b). Unlike the classic machine learning algorithms, deep learning can automatically learn effective features during training and therefore is free of extraction of handcrafted features. Because traditional convolutional deep learning architectures require highly regular formats of the input data, researchers converted the unorganised 3D point cloud into a sequence of 2D images (Boulch et al., 2018) or regular 3D voxels (Huang and You, 2016; Maturana and Scherer, 2015; Wu et al., 2015b). However, as the converted data cannot present the points' inherent 3D structure and loses fine details, Qi et al. (2017) proposed PointNet, which made it possible to directly implement convolution on individual points based on a key symmetric function. But according to Landrieu and Simonovsky (2018), the performance of PointNet or its extension (Engelmann et al., 2017) is limited by the size of input and fails to consider contextual information within both short- and long-range simultaneously.

### 2.2.4 Refinement

Semantic context has been widely considered and applied in the point cloud classification to refine the initial inference results. Generally, two types of methods are used to introduce semantic contextual information to the classification framework. The first is to use knowledge-based rules to determine or modify the semantic labels to make the labelling result more reasonable in terms of adjacency relationships. For instance, by using the rules defined based on building structures, Xu et al. (2014) were able to separate the roofs and walls from points that were labelled beforehand as buildings. Verdie et al. (2015) and Zhu et al. (2017) used semantic rules to modify the labels of objects according to their sizes, heights and the labels of adjacent objects.

A more general way is to capture the semantic context with the edges of a graphical model and establish a random field, such as Markov random field (MRF)

(Li, 2009) and conditional random field (CRF) (Lafferty et al., 2001). Therefore, the labelling refinement is converted into a global energy optimisation problem, which can be formulated as Equation (2-1).

$$arg\ \min_{\mathbf{y}} E(\mathbf{y}) = \sum_{i \in \boldsymbol{v}} D(\mathbf{y}_i) + \sum_{(i,j) \in \boldsymbol{e}} V(\mathbf{y}_i, \mathbf{y}_j) \tag{2-1}$$

where $E$ denotes the global energy of the graphical model, $\boldsymbol{v}$ denotes the set of nodes of the graph and $\boldsymbol{e}$ denotes the set of edges. $D(\mathbf{y}_i)$ is the unary potential describing the fidelity of labelling $\mathbf{y}$ to the observations, and $V(\mathbf{y}_i,\mathbf{y}_j)$ is the pairwise potential that penalises discontinuities in labelling $\mathbf{y}$. The purpose is to make the final labelling spatially smooth while remaining loyal to the initial inference by minimising the global energy via graph cuts (Boykov et al., 2001) or loopy belief propagation (Szeliski et al., 2008).

Lafarge and Mallet (2012) and Niemeyer et al. (2014) adopted point-level semantic contextual information and encoded it as pairwise interactions in an MRF and a CRF, respectively. To make the labelling spatially smooth, even with isolated points, Niemeyer et al. (2016) added a high-order term to the CRF, in which the high-order cliques corresponded to predefined segments and were modelled by the robust $P^n$ Potts model (Kohli and Torr, 2009). Compared to the point-based graphical model, relatively longer interactions can be captured by constructing a graph from multi-scale point clusters or supervoxels (Landrieu and Simonovsky, 2018; Lim and Suter, 2009). To introduce more contextual information, a high-order model based on supervoxels was established by Luo et al. (2018). With the label cost introduced by Delong et al. (2012), Luo et al. (2018) modelled the label redundancies within connected cliques. However, as both the pairwise and high-order interactions in their work were defined based on adjacent supervoxels, no refinement could be achieved for those isolated supervoxels. This issue can become especially serious when the supervoxels are not adaptive to the variations in point density.

## 2.3 3D Reconstruction from Point Clouds

3D reconstruction using point clouds obtained from ALS or aerial images has been an active topic in the photogrammetry, computer vision and remote sensing communities. Buildings, as the key features of urban areas, are the major study objects in related research. There is a considerable body of literature on the reconstruction of buildings, and these studies can be differentiated by several properties, such as data sources, output models, amount of automation and modelling strategies. In the following, the review of this topic is narrowed to the automatic reconstruction of polygonal building models, which is still considered a challenging task (Musialski et al., 2013; Wang et al., 2018).

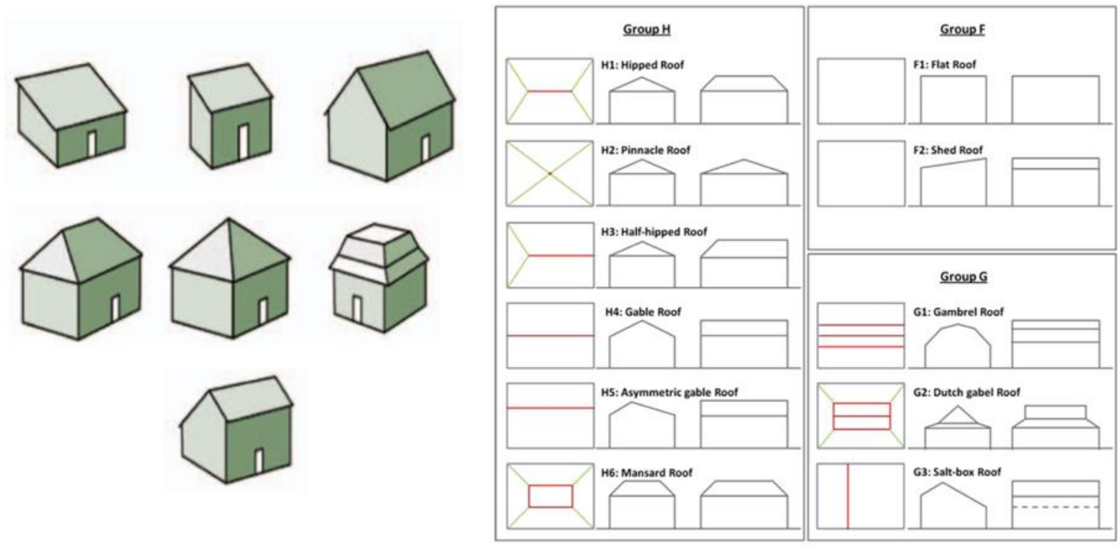
### 2.3.1 2.5D Reconstruction of Buildings

2.5D reconstruction of buildings refers to the reconstruction of building rooftops only. A large number of methods have been proposed for this purpose, and a certain level of success has been achieved in generating rooftop models with different levels of detail (LOD). In general, these methods can be grouped into model-driven, data-driven and hybrid-driven methods (Haala and Kada, 2010; Wang et al., 2018).

A. Model-Driven Methods

The model-driven methods take the building roof as a combination of a set of predefined primitives (e.g., a saddleback roof, sometimes with hip ends on one or both sides, pent, flat, tent and mansard roofs) (Huang et al., 2013; Kada, 2009; Kada and McKinley, 2009; Poullis and You, 2009b), as shown in Figure 2.1. Generally, the model-driven methods always start with the extraction and regularisation of building footprint boundaries. The regularised footprints are then decomposed into a set of 2D cells, and points corresponding to each cell are fitted by the primitives in the hypothetical model library with appropriate parameters. Finally, the building rooftops are automatically reconstructed by gluing the most fitted primitives with a set of 3D

Boolean operations (Rivers et al., 2010).




*(Poullis and You, 2009b)*     *(Huang et al., 2013)*

Figure 2.1 The basic primitive libraries defined in model-driven methods.

The boundaries of building footprints are generally extracted by conducting 2D alpha-shape (Liang et al., 1998) or ball-pivoting (Medeiros et al., 2004) algorithms on projected points, or from Delaunay triangulations (Maas and Vosselman, 1999). As the initial boundaries always have zig-zag shapes (Du et al., 2017; Poullis, 2013), simplification or regularisation of the initial boundaries is required. According to Xie et al. (2018), the boundary simplification and regularisation can be grouped into origin points using selection-based, local data fitting-based and dominant orientation-based methods. The Douglas Peucker (Douglas and Peucker, 1973) method is one of the most commonly used selection-based methods. This algorithm uses perpendicular distance as a global indicator. Points with a deviation exceeding a predefined threshold are kept and treated as the vertices constituting the final boundary. The second class of methods first detects line segments from the original boundary and then assembles them to from a closed polygon. Some researchers used the Douglas Peucker method to divide the original boundaries into a set of polylines and then further strengthened these polylines by fitting them with line segments (Jung et al., 2017; Kim et al., 2007).

In the third class of methods, the Manhattan assumption (Coughlan and Yuille, 2001) is adopted to produce simplified and regularised boundaries (Gross et al., 2005; Ma, 2005). It first detects the dominant direction of the building by, e.g., intersecting horizontal lines in 3D space (Vosselman, 1999) and rectangle fitting (Gross et al., 2005), and then enforces the rectangular shape constraints on the segments.

The regularised boundaries of building footprints are then decomposed into a set of 2D cells, which are mostly non-intersecting and quadrangular sections (Brenner and Haala, 1998; Kada, 2009; Kada and McKinley, 2009; Vosselman and Dijkman, 2001). The reconstruction of building rooftops with complex structures is then reduced to the simple subtasks of determining roof types and estimating model parameters, for which solutions already exist. For instance, Poullis et al. (2008) used the Gaussian mixture model (GMM) for elevation distribution to estimate the parameters of primitives during the determination of the best fitting ones. Kada and McKinley (2009) determined the roof types based on the normal directions of roof surfaces and estimated the parameters based on one eaves height and up to two ridge heights. The Reversible Jump Markov Chain Monte Carlo model was used by Huang et al. (2013) to determine and fit the parametric primitives to the input data. However, as the predefined primitives were not able to capture all of the roof structures, interactive manual editing was often required (Brenner and Haala, 1998; Kada and McKinley, 2009). Henn et al. (2013) proposed a model-driven method that used machine learning to determine the roof types and an enhanced RANSAC to estimate the model parameters, making the reconstruction of linear roof structures fully automatic.

The model-driven methods provide an efficient and robust solution for reconstructing building rooftops, which can be presented by a combination of specific basic primitives. However, building rooftops, especially buildings in a metropolis, always have complex and arbitrary structures, making them unique from others, which limits the scalability of model-driven methods.

B. Data-Driven Methods

Over the past two decades, many data-driven methods have been proposed with various strategies and technologies (Wang et al., 2018). A common idea in data-driven reconstruction is to extract the planar primitives, compute the pairwise intersection lines and then validate the corner points (Dorninger and Pfeifer, 2008; Maas and Vosselman, 1999; Vosselman, 1999; Vosselman and Dijkman, 2001). However, it can be non-trivial to guarantee that more than two intersection lines join at the same corner point and sometimes manual interferences are needed (Dorninger and Pfeifer, 2008). Sohn et al. (2008) proposed a method that used a binary space partitioning (BSP) tree to divide the building boundary into a set of small planes based on linear features, including step lines and intersection lines. These small planes were then used to build a plane adjacency graph and merged based on their normal directions. Jung et al. (2017) also used BSP to recover the topologies between primitives and regularised the building rooftop in a framework of minimum description length in combination with the hypothesize and test procedure. Methods adopting BSP can avoid complicated computation of intersections and produce rooftop models without topological errors, but erroneous linear features can lead to irregular shapes and corners. Sampath and Shan (2010) restored the interior corners of building rooftops using an adjacency matrix, where the adjacent relationships between planar primitives were determined based on the Voronoi diagram of the points. In this method, the flat roof planes that were not adjacent to any other roof planes, were regularised separately, which might lead to crack effects in stepped roof structures.

Many methods have been proposed specifically for the reconstruction of high-rise flat-roof buildings (Matei et al., 2008; Poullis, 2013; Poullis and You, 2009a; Zhou and Neumann, 2008). These methods generally involve the regularisation of boundaries extracted from individual flat roof primitives, which can be referred to the regularisation of building footprints. For instance, Zhou and Neumann (2008) determined the dominant directions based on the tangent directions of the original boundary and then regularised the boundary by snapping the points on the original boundary to the dominant directions. Poullis (2013) determined the dominant directions based on the local tangent directions and constructed a GMM. The original boundary was then regularised by classifying the boundary points using a graphical model. The primitive boundaries were still extracted and regularised separately, which resulted in crack effects in the final models. To reduce crack effects, the boundary between two planar components of a building was extracted as one polyline using the subgraph of the Voronoi diagram in the work of Chen et al. (2014; 2017). The topological relationships between the flat roof components were well preserved and watertight and compact models could be produced, however, the topological relationships between roofs with large height jump could not be accurately restored.

Methods have also been proposed for the reconstruction of non-linear roof structures. Generally, these methods were designed to produce watertight building models where the roofs were presented by simplified triangular meshes. Thus, triangulation processing methods, e.g., vertex decimation, vertex clustering and edge collapse (Haala and Kada, 2010; Wang et al., 2018), were often used in such methods. Lafarge and Mallet (2012) extracted planar, spherical, cylindrical and conoidal shapes from the point clouds by region growing and iterative non-linear minimisation. The primitives of different shapes were modelled as mesh-patches and intersected with adjacent ones to form the hybrid building roofs. Zhou and Neumann (2010) proposed a more general method using a 2.5D dual contouring (2.5D D-C). In this method, the points were first converted into surface and boundary Hermite data based on a 2D grid. The triangular mesh was created by computing a hyper-point in each quadtree cell and simplified by collapsing subtrees and adding quadratic error functions associated with leaf cells. Finally the mesh was closed by connecting hyper-points with surface and boundary polygons to generate a watertight mesh model. This method was extended by Zhou and Neumann (2011; 2012) to generate building mesh models with better topology control and global regularities.

The main advantage of data-driven methods is that they can reconstruct building rooftops with arbitrary and complex structures and are not restricted to a predefined primitive library. However, a specific data-driven method is always only applicable to certain types of buildings and might have drawbacks in terms of the topological correctness, sensitiveness to data quality and computation cost (Wang et al., 2018).

C. Hybrid-Driven Methods

The hybrid-driven reconstruction of building rooftops is a combination of the data-driven and model-driven methods. Such methods generally decompose the building rooftops into planar primitives and store the topological relationships between the primitives using a roof topology graph (RTG), as shown in Figure 2.2. The nodes of an RTG denote the primitives and the edges denote the adjacency relationships between the primitives associated with topologies. The advantage of the RTG is that it can encode topologies and knowledge as constraints into the procedure of determination of essential features, such as roof types, heights, ridge directions and corner points so that the rooftops can be reconstructed with more accuracies and rationalities.

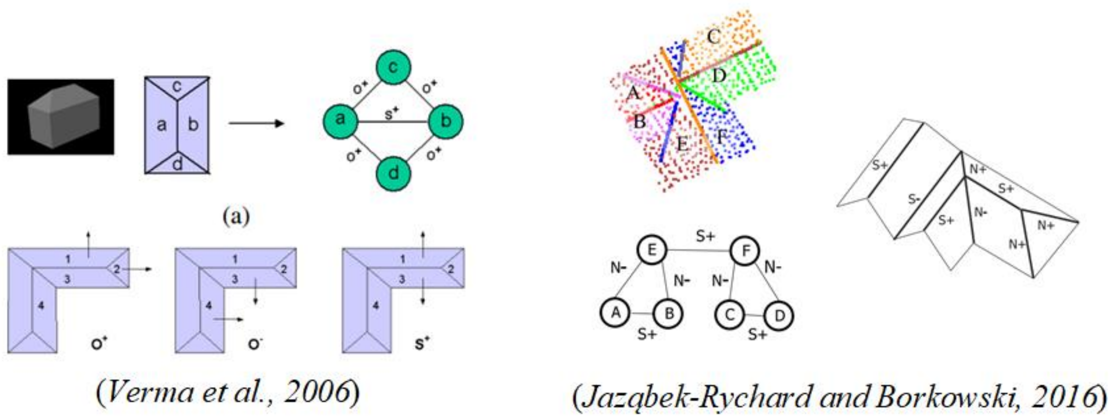



Figure 2.2 Roof topology graphs (RTGs) defined in hybrid-driven methods.

RTGs have been used to reconstruct building rooftops from images (Ameri and Fritsch, 2000), and Verma et al. (2006) introduced it into the reconstruction based on point clouds. In the work of Verma et al. (2006), an RTG was established for each building, where the edges of the RTG were categorised into three groups based on the normal directions of corresponding primitives connected by the edges. Similarly, RTGs were also established for a set of simple sub-roof structures. Thus, the simple sub-roof structures constituting a complex roof could be recognised by subgraph matching with a brute-force search. Parts of the complex RTG that could not be matched were regarded as rectilinear objects and modelled based on their regularised outlines. Elberink (2009) and Elberink and Vosselman (2009) extended the library of the sub-roof structures and proposed a target-based graph matching method that could handle both complete and incomplete data. However, over- or under-segmentation of the roof planes, loss of roof planes and a lack of basic primitive models can cause incorrect matching in these methods. To correct the topological errors in RTGs, Xiong et al. (2014; 2015) identified four types of errors, i.e., false node, missing node, false edge and missing edge, and proposed a graph edit dictionary to correct such errors, but manual editing was required during the correction. Jarząbek-Rychard and Borkowski (2016) proposed an unambiguous decomposition method for building rooftop reconstruction. In this method, the basic model library contained a set of structure-dependent soft modelling rules rather than strict geometric primitives. Therefore, this method was more flexible than previous ones and a balance could be achieved between reconstruction precision and regularity. However, the RTG-based matching was still error-prone, and the scalability was limited by the predefined library.

Instead of decomposing buildings into primitives with fixed structures, Lin et al. (2013) decomposed buildings with a hierarchy tree, where the top-level was blocks with closed surfaces, the middle level was pairs of surface patches and the bottom-level was surface primitives labelled as wall, roof and column. The buildings were decomposed progressively based on this hierarchy tree and reconstructed with hard constraints on different levels. This hierarchy presentation of buildings made this method flexible enough for the reconstruction of buildings with various styles and even with incomplete data. However, the hard constraints enforced in the procedure (e.g., about the symmetry), do not always exist in real situations. Hu et al. (2018b) also adopted a hierarchical decomposition strategy. This method first partitioned the RTG into sub-graphs, called structures, according to the convexity and concavity of the edges, and the structures were composed of a set of planar primitives. The structures were further refined according to gestalt laws, so that all of the structures could be presented as understandable sub-RTGs.

### 2.3.2 3D Reconstruction of Buildings

Unlike the reconstruction of building rooftops, the reconstruction of entire buildings as true 3D models using point clouds has not been intensively investigated. A review of the methods developed for the 3D reconstruction of buildings is presented below.

One strategy of the existing methods is to model the building surface based on information derived from a series of horizontal contours. Following the idea of generating a topological surface model based on a contour tree (Wu et al., 2015a), Wu et al. (2017) proposed a method that reconstructed 3D building models using a graph-based localised contour tree. This tree stored the topological relationships between the contours that were extracted from a normalised digital surface model (nDSM) with a constant interval. Surface models between every two contours connected in the tree were generated through weighted bipartite graph matching, and the building model was then generated by integrating all of the surface models. The interval between contours was important in this method: whereas a small interval can hinder computational efficiency, a large interval might lead to inaccuracies at the junctions between different building parts and the loss of small building structures. Instead of a constant contour interval, Yi et al. (2017) selected key contours based on the distribution histogram of boundary points in an upward direction. These key contours were regularised by decomposing them into linear primitives and fitting them with constrained line segments. This method was efficient and able to reconstruct building models in a large-scale area with full automation, but the fundamental contour processing was sensitive to noise and short segments.

A more frequently adopted strategy is to partition 3D space into basic units, e.g., 3D faces, boxes and polyhedral cells, based on the planar primitives derived from the buildings, and then to select the optimal set of basic units to approximate the buildings. Verdie et al. (2015) partitioned the enveloping space of building meshes into polyhedral cells and considered the selection of occupied cells (cells that were inside the building meshes) as a binary labelling problem, which was formulated by a graphical model and solved via the max-flow algorithm (Boykov and Kolmogorov, 2004). To reduce the complexity of accurately computing the probability of each cell being occupied, the 3D space was presented with the assistance of a set of discrete anchors (the corner points of a uniform grid), and it could easily be determined whether they were inside the building mesh or not. The adoption of discrete anchors made this method free of complex geometric computation, but the number of anchors could be extremely huge for large buildings, which might harm the efficiency of this method. To avoid this dilemma, Li et al. (2016a) partitioned the 3D space into boxes aligned to the building's dominant direction. These boxes could be easily computed to be inside, outside or intersecting with the building meshes based on the eight corners, with the total number of boxes remaining acceptable. Li et al. (2016b) revised the box selection metric to a data fitting score computed based on the six faces of each box and the corresponding supporting points. An MRF formulation was adopted to make the optimal selection of boxes in terms of fitting scores and the connections between boxes. However, it is obvious that these two methods are only applicable to scenes that satisfy the Manhattan world assumption. Nan and Wonka (2017) proposed a more general method called PolyFit. PolyFit generated a set of hypothetical faces by intersecting planes extracted by RANSAC and selected faces that best approximated the building surface using binary integer linear programming. A hard edge constraint that an edge must be connected by zero or two faces was encoded in the global energy formulation, to make the final model manifold and watertight. However, PolyFit was only intended for the reconstruction of simple polygonal surfaces, and in practical applications, this method might encounter computational bottlenecks with complex objects (Nan and Wonka, 2017).

Although the above studies have made certain achievements, their performances were hindered by their own limitations. The automatic generation of true 3D models of various buildings in large-scale scenes remains a challenging task. In addition, all of the above methods stopped at restoring the building geometries. Further efforts, such as converting the geometric models into CityGML models for GIS applications, have rarely been made.

## 2.4 Summary of Related Work

Generally, three types of relations are considered in previous research. These three types of relations and their major usages in previous studies are summarised in Table 2.1.

Table 2.1 Summary of relations used in previous studies.

<table>
<tr><th colspan="2">Relation type</th><th>Usage</th><th>References</th></tr>
<tr><td rowspan="3">Geometric relations</td><td>Geometric homogeneity in terms of basic geometric properties, e.g., coordinates and normals.</td><td>In segmentation: to decide if two points should be assigned to the same segment.</td><td>(Papon et al., 2013; Schnabel et al., 2007; Tarsha-Kurdi et al., 2007; Vosselman et al., 2004)</td></tr>
<tr><td rowspan="2">Orientation-based relations, e.g., parallelism, orthogonality, co-plnarity and symmetry</td><td>2D: Boundary regularisation.</td><td>(Xie et al., 2017; Yi et al., 2017)</td></tr>
<tr><td>3D: Refinement of building planes.</td><td>(Li et al., 2016b; Verdie et al., 2015)</td></tr>
<tr><td rowspan="3">Contextual relations</td><td>Class-oriented relations, e.g. roof is above building façade.</td><td>Used as knowledge-based rules in the refinement of classification.</td><td>(Verdie et al., 2015; Xu et al., 2014; Zhu et al., 2017)</td></tr>
<tr><td rowspan="2">Label-oriented (the labels do not necessarily have physical meanings, e.g., a series of numbers.)</td><td>In graph-based partitioning.</td><td>(Landrieu and Simonovsky, 2018)</td></tr>
<tr><td>In graph-based classification framework.</td><td>(Luo et al., 2018; Niemeyer et al., 2013, 2014; Vosselman et al., 2017; Zhu et al., 2017)</td></tr>
<tr><td colspan="2" rowspan="3">Topological relations</td><td>Building structure analysis.</td><td>(Wu et al., 2017)</td></tr>
<tr><td>In hybrid-driven reconstruction of building rooftops.</td><td>(Elberink, 2009; Elberink and Vosselman, 2009; Jarza̧bek-Rychard and Borkowski, 2016; Verma et al., 2006; Xiong et al., 2014; Xiong et al., 2015)</td></tr>
<tr><td>In 3D Reconstruction of buildings.</td><td>(Li et al., 2016a; Li et al., 2016b; Nan and Wonka, 2017; Verdie et al., 2015)</td></tr>
</table>

Figure 2.3 summarises the representative segmentation and classification methods as they are quite relevant. Segmentation is always performed before classification to obtain the primitives to be classified. However, the segments are generally input as pure geometric primitives without any physical meaning: in most existing methods, only simple contextual relations are captured by the edges of MRF

or CRF models. In the reference step, the unary features extracted from individual primitives play a decisive role; this might lead to inaccurate results, as the relations between the primitives and their environments are ignored. To introduce more relations into the essential reference step, and to enrich the relations used for the classification refinement, structural primitives should be generated and used to interpret the point clouds instead of pure geometric primitives; this is an objective of this research.

The existing building reconstruction methods, as well as their limitations, are summarised in Figure 2.4. Although many methods have been developed in the past two decades, most of them only focus on the reconstruction of building rooftops. A number of methods investigate the reconstruction of buildings as a whole, but is applicable to large-scale scenes with various and complex buildings. In addition, few efforts have been made to investigate the generation of CityGML models that could be used in GIS applications. Therefore, one of the major objectives of this research is to reconstruct true 3D CityGML models in large-scale scenes with various and complex buildings by considering the geometric and topological relations between the building structures.

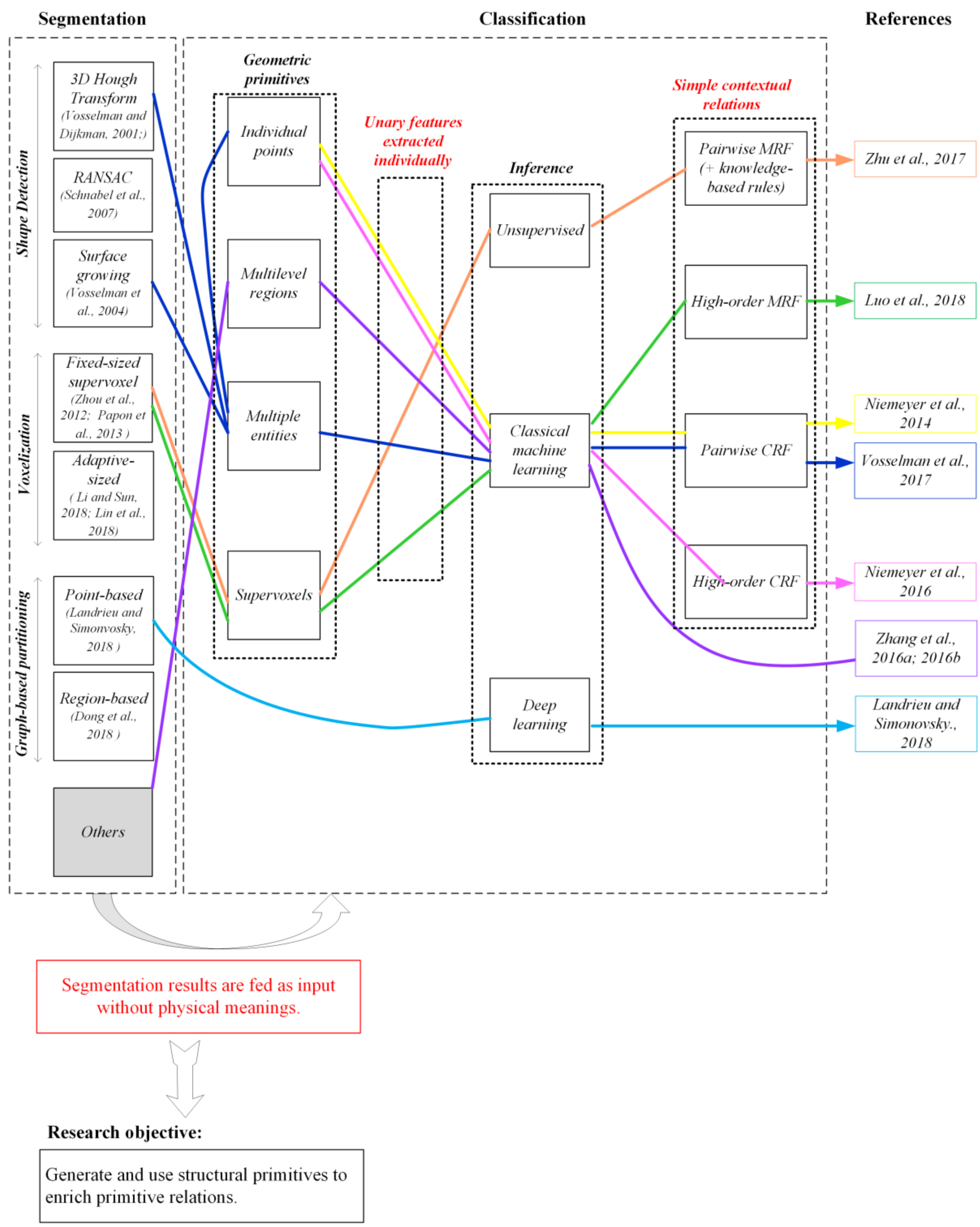



Figure 2.3 Summary of representative segmentation and classification methods. The coloured lines refer to different classification methods.

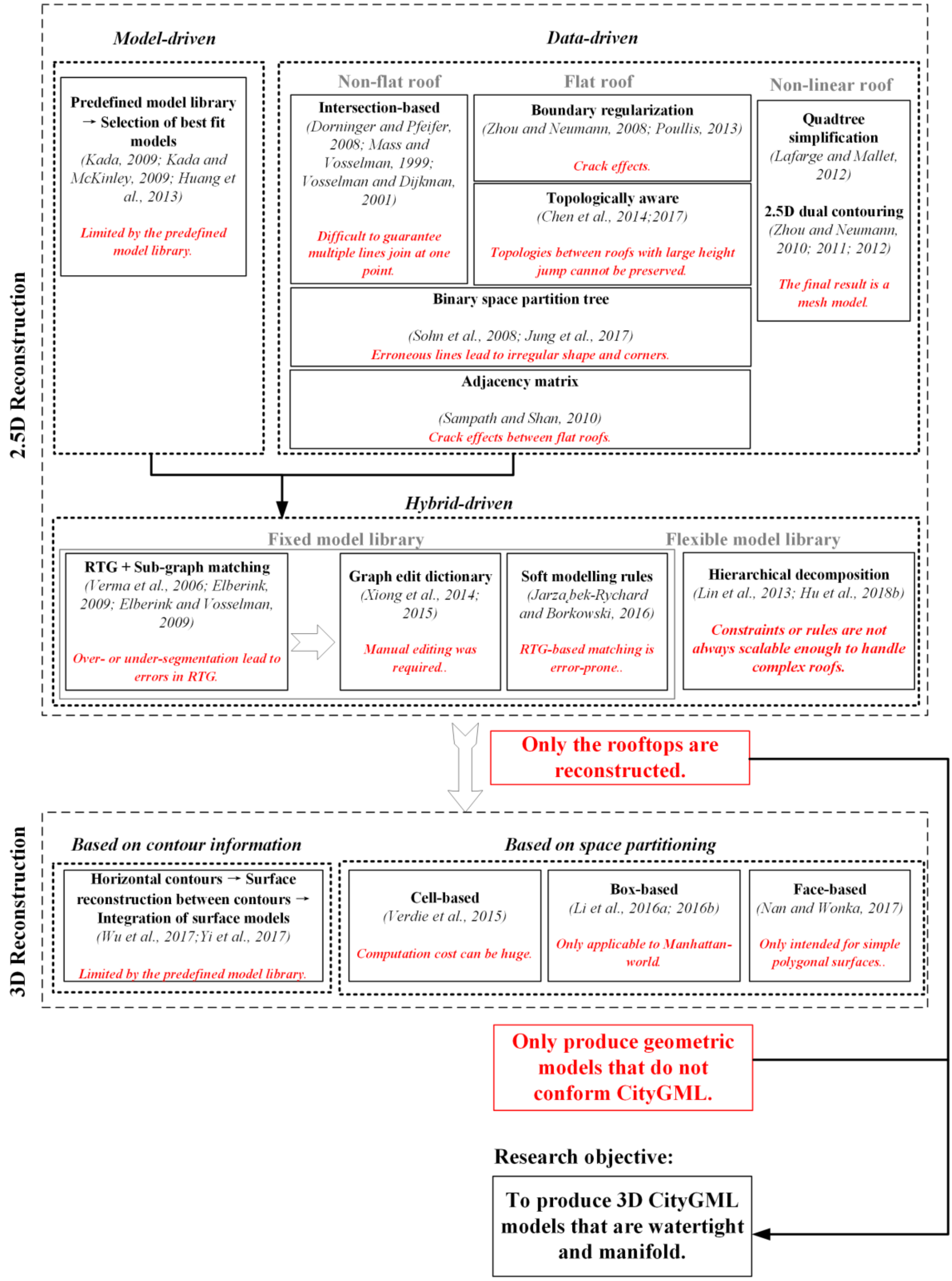



Figure 2.4 Summary of representative 3D reconstruction methods.

# Chapter 3 Segmentation of Point Clouds Based on Multiple-Level Relations

Objects are formed by various structures of different shapes (e.g., linear, planar and scatter shapes) and such structural information is essential for the recognition of objects, especially for objects presented with abundant details, such as the street facilities presented in MLS data. However, in previous studies, structural information has rarely been exploited to improve the precision of distinguishing between objects with global or local similarities, such as traffic signs and traffic lights. In addition, the defects of point clouds (e.g., the great data volume, anisotropy of point density and noise) and the complexity of scenes and objects make the decomposition of objects into physically meaningful structures remain a challenging issue. Hence, an effective segmentation method is first proposed in this research to decompose objects into different structures with physically meaningful labels. This method takes advantage of multiple-level relations, so that it is robust to data noise and the variations in point density and is capable of handling large-scale point clouds.

Below presents the developed structrual segmentation method. Section 3.1 first gives an overview of the method, where the multilevel relations are addressed. Section 3.2 describes a novel adaptive noise filter that takes effect based on anisotropy of point density. Section 3.3 presents a supervoxel segmentation algorithm that takes advantages of geometric relations at the point level. In Section 3.4, the supervoxels are further used to generate structural components based on a MRF framework, where the geometric relations and the contextual relations on the supervoxel level are encoded. Section 3.5 provides a summary of this chapter.

## 3.1 Overview of the Point Cloud Segmentation Method

The first step of the proposed segmentation method is to filter ground points using an adaptive surface filter proposed by Hu et al. (2014) to simplify long-range connections between ground objects. A novel adaptive noise filter based on the density anisotropy at the point level is subsequently implemented on the non-ground points to further refine the connections between close objects. The remaining points are then segmented through two steps. In the first step, the non-ground points are partitioned into multi-size supervoxels through a coarse-to-fine seed selection process and a supervoxel expansion process based on relations in terms of point distance, angle between normal vectors and colour similarities (if colour exists), followed by recovering the adjacency relationships between supervoxels. In the second step, three shape descriptors, corresponding to linearity, planarity and scattering, are firstly derived from supervoxels and used to establish a MRF model, where the contextual information at the supervoxel level is encoded. The labelled supervoxels are finally clustered into structural components by region growing, according to the consistency of their labels. Figure 3.1 gives an overview of the proposed segmentation method.

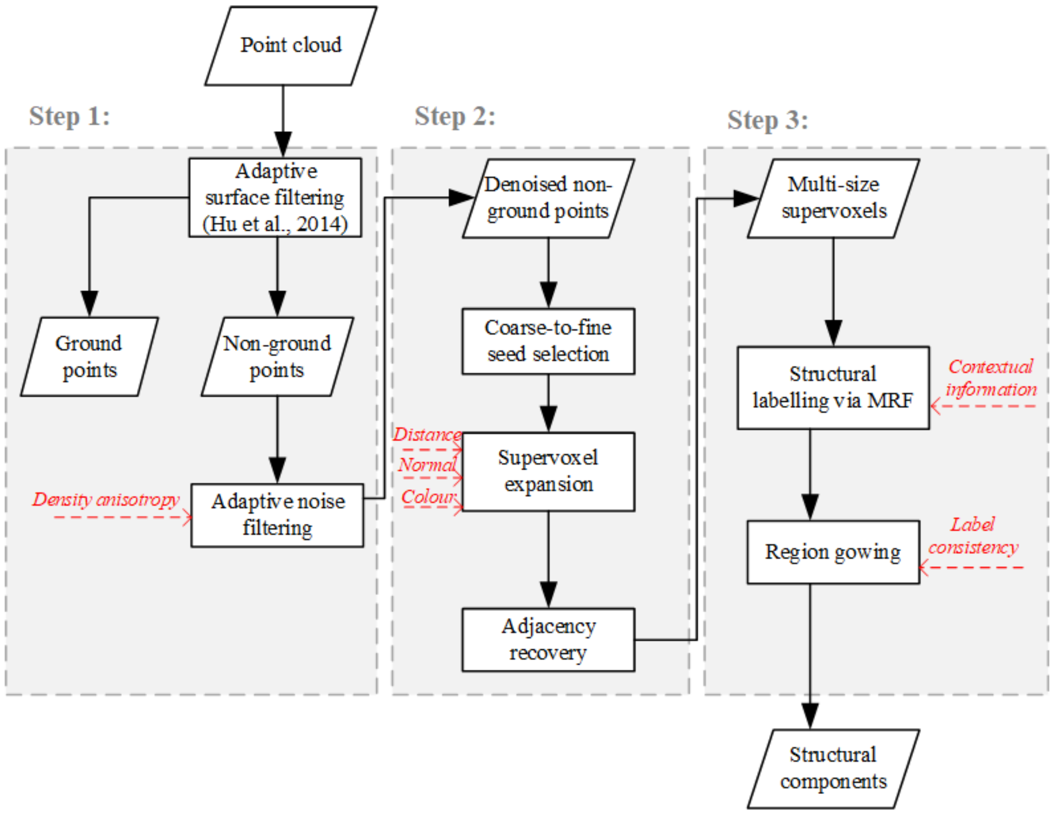


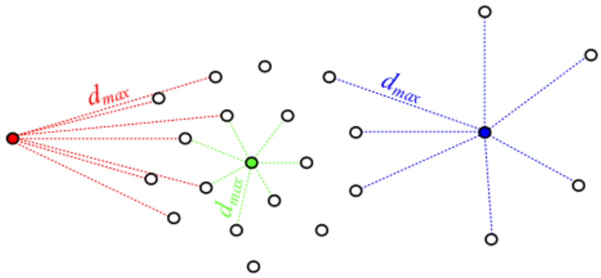

Figure 3.1 Overview of the segmentation method. The red dashed lines indicate there are relations encoded in the corresponding steps.

## 3.2 Adaptive Noise Filtering based on Density Anisotropy of Points

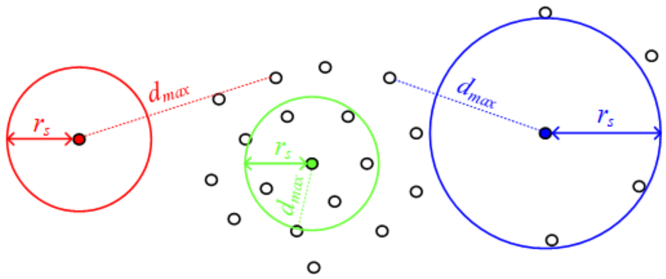
The presence of noise poses a serious problem for the interpretation of point clouds and therefore needs to be removed as a pre-processing step. However, the point density of point cloud data always varies with the changing scanning distance, which is especially obvious for MLS and TLS data, and this is likely to make points located in sparse areas be removed by some common noise filters, e.g., the Statistical Outlier Removal (SOR) filter (Rusu et al., 2008) or the Radius Outlier Removal (ROR) filter (Rusu, 2018) in Point Cloud Library (PCL) (Rusu and Cousins, 2011).

Although the point density varies, the variation is believed to be continuous on the same object surface. And based on this assumption, an adaptive noise filter is proposed in this research. In this algorithm, points with low density isotropy are considered as outliers and will be removed. The definition of density isotropy is as follows. First, find the $k$ nearest neighbours for each point $p$ in a given point set $\mathcal{P}$ (as shown in Figure 3.2 (a)) and record the maximum distance $d_{\max}$ between $p$ and its $k$ nearest neighbours. Second, for each $p \in \mathcal{P}$, compute the average value of $d_{\max}$ of its $k$ nearest neighbours, denoted as $r_s$ (as shown in Figure 3.2 (b)), and the density isotropy of $p$ is defined as Equation (3-1).

$$D_{isotropy} = r_s / d_{\max} \tag{3-1}$$




(a) Maximum distance $d_{\max}$ defined by k-NN search.




(b) Adaptive comparision radius $r_s$ defined by the average value of $d_{\max}$ of $p$'s $k$ nearest neighbours.

Figure 3.2 The computation of density isotropy. The red point refers to an outlier, the green and blue points refer to points in dense and sparse areas, respectively.

Generally, for points located in areas with even point density, $r_s$ is likely to be similar to, or even larger than, $d_{\max}$, resulting in greater density isotropy $D_{\text{isotropy}}$, even for points located in sparse areas, such as the blue point in Figure 3.2. Only points with $D_{\text{isotropy}}$ lower than a given threshold $\lambda$ are treated as outliers, as the red point shown in Figure 3.2, no matter it is located in dense or sparse areas. Theoretically, excluding heavier noise requires a larger $\lambda$.

Compared to SOR and ROR filters, this adaptive noise filter provides a good trade-off between the capability of removing outliers and preserving the density consistency of the point cloud, as shown in Figure 3.3. In the comparison, $k$ is set 20 and $\lambda$ is set 0.5 for the adaptive noise filter. The $k$ nearest neighbours is also set 20 for SOR filter and the standard deviation multiplier is set 1.0. For the ROR filter, the radius is set 0.3 and the minimum number of neighbours in radius is set 3.

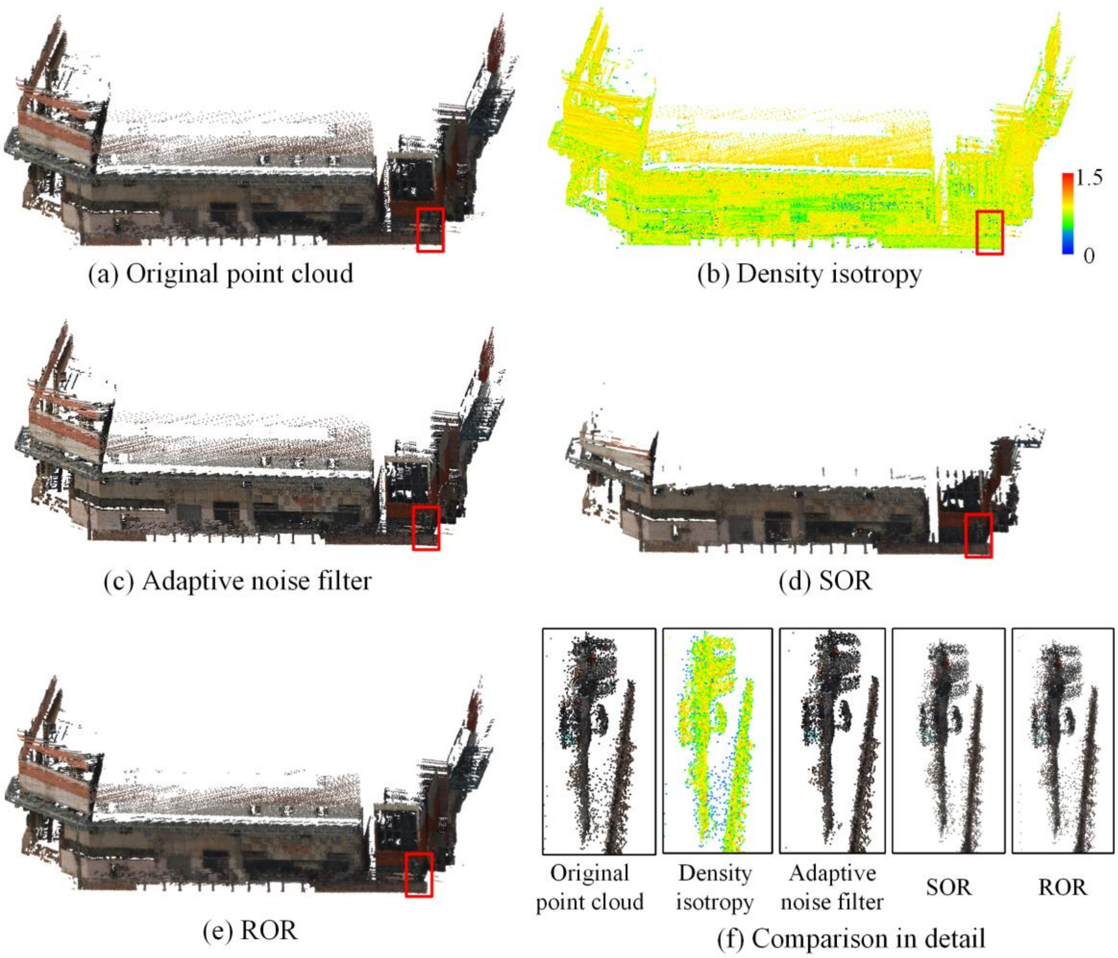



Figure 3.3 Adaptive noise filtering result of point cloud with great variations in point density and comparisons with SOR and ROR filters.

## 3.3 Generation of Supervoxels based on Point Relations

The supervoxel segmentation algorithm in this research is an extension of the VCCS (Papon et al., 2013). Instead of fixed supervoxel seed resolution as in the VCCS, the developed algorithm adopts a coarse-to-fine seed selection strategy to make the supervoxels adaptive to the varying density of the point clouds. The coarse-to-fine seed selection process constrains the sizes of the supervoxels to be neither too large nor too small. If the supervoxels are too large, the segmentation might bestride boundaries between objects, and if too small, the features derived from the supervoxels become meaningless. Then the supervoxels are generated by expanding the seeds with adaptive resolutions based on the relations at the point level. During the expansion, the adjacencies between supervoxels are also determined by the neighbouring relationships between the occupied points.

### 3.3.1 Coarse-to-Fine Seed Selection

For a point set $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, the purpose is to generate a set of supervoxels $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$, where each $V_i = \{p \mid p \in \mathcal{P}\}$ should contain sufficient points for local feature computation. Simultaneously, the size of $V_i$ should not be too small so that it can effectively capture the local geometries at an appropriate scale. In other words, there are two constraints on the size of supervoxel $V_i$ as shown in Equation (3-2).

$$\begin{cases} K_i \geqslant K_{min} \\ r_i \geqslant r_{min} \end{cases} \tag{3-2}$$

where $K_i$ is the number of points contained in $V_i$, and $r_i$ is the maximum size of the bounding box of $V_i$. $K_{min}$ and $r_{min}$ are user-defined thresholds.

First, based on the constraint $K_i \geq K_{min}$, the maximum seed resolution $r_{max}$ can be determined as the following equation.

$$r_{max} = \max\{d_{max}(p, \mathcal{N}_p^{K_{min}}) \mid p \in \mathcal{P}\} \tag{3-3}$$

where $\mathcal{N}_p^{K_{min}}$ is the $K_{min}$ nearest neighbours of $p$ and $d_{max}(p, \mathcal{N}_p^{K_{min}})$ denotes the maximum distance between p and $\mathcal{N}_p^{K_{min}}$.

Then, a coarse octree is built based on the maximum seed resolution $r_{max}$ and the coarse seeds $Seed_0$ are selected as points that are closest to the centroids of the leaf nodes in the coarse octree. The coarse seeds $Seed_0$ are then traversed individually. For each seed $s_i \in Seed_0$, the number of points occupied by corresponding octree leaf node is noted as $K_i$. If $K_i$ is larger than $4K_{min}$ ($4K_{min}$ is used here because in an octree built from point clouds, the number of points occupied by a node is generally a quarter of the number in its parent node), this leaf node will be further split into a deeper octree structure with a corresponding resolution of $r_{\max}/2^n$ ($n$ is the difference between the current octree depth and the depth of the coarse octree), as shown in Figure 3.4. The new seeds are then selected as the points closest to the centroids of all of the leaf nodes in the new octree. The traversing is repeated until all of the leaf nodes in the octree contain points less than $4K_{\min}$ or the supervoxel resolution $r_i$ becomes smaller than $r_{\min}$.

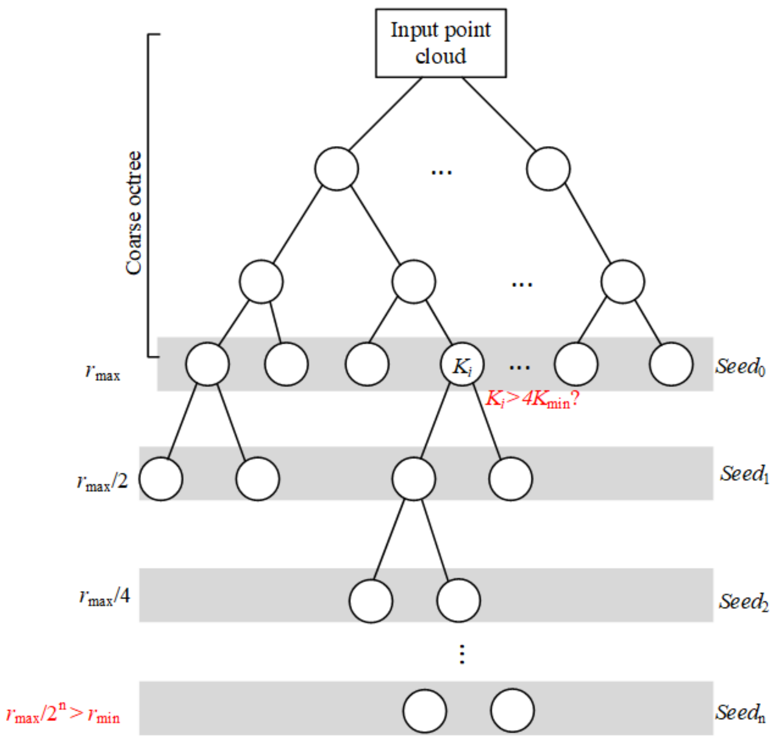



Figure 3.4 Coarse-to-fine supervoxel seed selection based on an octree structure.

### 3.3.2 Supervoxel Expansion with Point Relations

After the coarse-to-fine seed selection, a set of points is selected from $\mathcal{P}$ as the final seeds of supervoxels, denoted as $\boldsymbol{s} = \{s_1, s_2, \ldots, s_m\}$, with corresponding multiple resolutions $\boldsymbol{r} = \{r_1, r_2, \ldots, r_m\}$. Simultaneously, the initial centroids of supervoxels $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$ are determined by the seeds. The supervoxels are then expanded using a similar iterative strategy as the VCCS (Papon et al., 2013). In each iteration, the point relations are used to determine whether or not the $k$ nearest neighbours $\mathcal{N}_{p_{new}}$

of a point $p_{\mathrm{new}}$, which is newly assigned to a supervoxel $V$ in the last iteration ($p_{\mathrm{new}}$ is the seed point in the first iteration), should be assigned to supervoxel $V$. The point relations are used to measure the homogeneity $H(p, V)$ between a point $p$ and the centroid of a supervoxel $V$ in terms of coordinate, normal and colour (if colour exists), which are formulated as the following.

$$H(p,V) = h_{coord} + h_{normal} + h_{colour} \tag{3-4}$$

$$\begin{cases} h_{coord} = \dfrac{d_{Eu}(p,c)}{r} \\ h_{normal} = \dfrac{\theta(\boldsymbol{n}_p,\boldsymbol{n}_V)}{\pi} \\ h_{colour} = \dfrac{d_{Eu}(C_p,C_V)}{\sqrt{3}\,m} \end{cases} \tag{3-5}$$

where $d_{Eu}(p, c)$ is the Euclidean distance between point $p$ and the supervoxel centroid $c$, $r$ is the resolution corresponding to the seed of supervoxel $V$, $\theta(\boldsymbol{n}_p, \boldsymbol{n}_V)$ denotes the angle between the normal of $p$ and $V$, $C_p$ and $C_V$ are corresponding colour vectors with RGB channels and $m$ is the grayscale of single colour channel.

The coordinate relation constrains the final supervoxels to be compact, the normal relation helps to preserve sharp features during the expansion, and the colour relation enhances the local homogeneity in addition to geometric properties. Instead of an adjacent octree as in the VCCS, the k-NN search is adopted to determine the neighbouring relationships between points during the expansion. In each expansion iteration, for two adjacent points $p_{\mathrm{i}} \in V_{\mathrm{a}}$ and $p_{\mathrm{j}} \in V_{\mathrm{b}}$, expansion operation (reassigning $p_{\mathrm{j}}$ to $V_{\mathrm{a}}$) will be performed if $H(p_{\mathrm{j}}, V_{\mathrm{a}}) < H(p_{\mathrm{j}}, V_{\mathrm{b}})$. Note that, $p_{\mathrm{i}}$ and $p_{\mathrm{j}}$ are regarded as adjacent only when they are mutually in each other's k nearest neighbours as illustrated in Figure 3.5. If points $p_i$ and $p_j$ are adjacent and they belong to two different supervoxels, the supervoxels are also regarded as adjacent (as shonw in Figure 3.5 (b)). The pseudo-code for the supervoxel expansion is shown in Table 3.1.


(a) $p_i$ and $p_j$ are not adjacent as $p_i$ is not in the $k$ nearest neighbours of $p_j$.

(b) $p_i$ and $p_j$ are adjacent as they are mutually in each other's $k$ nearest neighbours.

Figure 3.5 Recovery of adjacency between points and supervoxels. The red and green points refer to two different supervoxels.

Table 3.1 Pseudo-code for supervoxel expansion.

**Algorithm:** Supervoxel expansion

**Input:** The point cloud $\mathcal{P}$; supervoxel seeds $\boldsymbol{s}$; seed resolutions $\boldsymbol{r}$.

**Output:** the supervoxels $\mathcal{V}$ with adjacent relationships $\boldsymbol{e}$.

Initialize: $V_i = \{\boldsymbol{s}_i\}$ $(i = 1, \ldots, |\boldsymbol{s}|)$; $\boldsymbol{e} = \varnothing$;

point $p_n$'s ($p_n \in \mathcal{P}$) adjacent neighbourhood: $\mathcal{N}_n^{\mathcal{P}} = \{p \in \mathcal{N}_{p_n}^{K_{\min}} \mid p_n \in \mathcal{N}_p^{K_{\min}}\}$;

$V_i$'s ambiguous neighbourhood $\mathcal{N}_i^{\mathcal{V}} = \{V \in \mathcal{V} \mid d_{\mathrm{Eu}}(V, V_i) < r_{\max}\}$;

the index of supervoxel occupying point $p_n \in \mathcal{P}$: $VIdx_n = -1$; and the distance between them $dV_n = \infty$;

Exchanged points $P = \varnothing$.

**for** $i = 0$ to $|\mathcal{V}| - 1$

  search nearest point $p_n \in \mathcal{P}$; $dV_n = H(p_n, V_i)$; $VIdx_n = i$.

  push $p_n$ to $P$: $P \leftarrow p_n$.

**end for**

**while** $P \neq \varnothing$

  $old\_P = P$; $P = \varnothing$

  **for** $j = 0$ to $|old\_P| - 1$

    **for** $n = 0$ to $|\mathcal{N}_j^{\mathcal{P}}| - 1$

      **for** $i = 0$ to $|\mathcal{N}_{VIdx_n}^{\mathcal{V}}| - 1$

        **if** $H(p_n, V_i) < dV_n$

          push $p_n$ to $V_i$: $V_i \leftarrow p_n$; erase $p_n$ from $V_{VIdx_n}$;

          push $p_n$ to $P$: $P \leftarrow p_n$; $dV_n = H(p_n, V_i)$; $VIdx_n = i$

        **else**

          push $e = (VIdx_n, i)$ to $\boldsymbol{e}$: $\boldsymbol{e} \leftarrow e$; if $e \notin \boldsymbol{e}$.

        **end if**

      **end for**

```
            end for
      end for
  end while
  returen 𝒱 and e
```

Figure 3.6 demonstrates the process of coarse-to-fine seed selection and the result of supervoxel expansion. Each point in the final seeds corresponds to a leaf node in the multiple-resolution octree (see Figure 3.4). The points contained by the leaf nodes constitute the initial supervoxels as shown in Figure 3.6 (a). These supervoxels might bestride boundaries between object as shown in the magnified view in Figure 3.6 (a). But from Figure 3.6 (b) it can be seen that, the sharp corners are finally well preserved by expanding supervoxels with point-level constraints. The adjacencies between supervoxels are also correctly recovered as shown in Figure 3.6 (c).

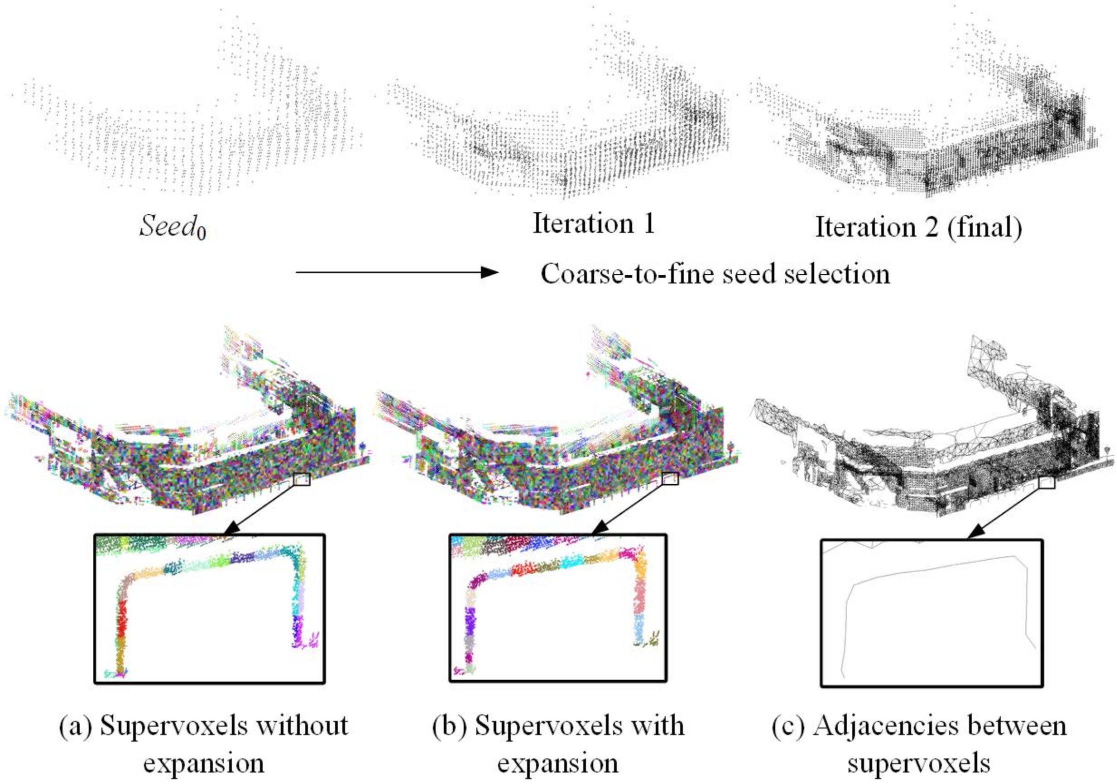



Figure 3.6 The process of coarse-to-fine seed selection and the result of supervoxel expansion.

## 3.4 Generation of Structural Components based on Supervoxel Relations

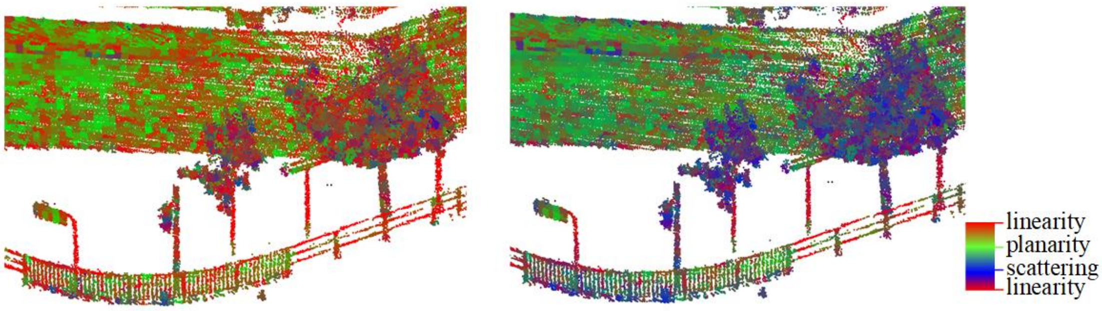
To generate structural components, the supervoxels with adaptive sizes are first classified into three structural categories, namely linearity, planarity and scattering, via an MRF framework, where the contextual relations at the supervoxel level are encoded. Then by region growing, the supervoxels with the same structural labels are clustered into structural components, of which the structural labels are inherently obtained from the corresponding supervoxels.

### 3.4.1 Structural Labelling via MRF

To label the supervoxels into different structural classes, the local geometric characteristics corresponding to linearity, planarity and scattering are measured by three local shape descriptors, which are defined based on the eigenvalues derived from the covariance matrix through the principle component analysis (Jolliffe, 2011). Different definitions of the shape descriptors are found in the work of Weinmann et al. (2014), Hackel et al. (2016) and Yang et al. (2015), but it is noted that the definition of Yang et al. (2015) has better ability of describing the scatter property of points as shown in Figure 3.7. Therefore, in this research, three shape descriptors $f_l$, $f_p$ and $f_s$ describing the linearity, planarity and scattering of points are computed as defined in the work of Yang et al. (2015) and have the following formats.

$$\begin{cases} f_l = \dfrac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}} \\ f_p = \dfrac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}} \\ f_s = \dfrac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}} \end{cases} \tag{3-6}$$

where $\lambda_1 > \lambda_2 > \lambda_3$ are the eigenvalues derived from the covariance matrix of each supervoxel.




(a) Shape descriptors measuring linearity, planarity and scattering as defined in Weinmann et al. (2014) and Hackel et al. (2016).

(b) Shape descriptors measuring linearity, planarity and scattering as defined in Yang et al. (2015).

Figure 3.7 Capabilities of differently defined shape descriptors of measuring local geometric characteristics.

The structural labelling of the supervoxels can be determined by the three shape descriptors. To make the structural labelling spatially smooth, contextual relations at the supervoxel level are introduced to an MRF framework. Let $\boldsymbol{x}$ be the set of nodes that correspond to the supervoxels and let $\boldsymbol{e}$ be the set of edges that correspond to the adjacency relationships between supervoxels. A graphical model $G(\boldsymbol{x},\boldsymbol{e})$ is therefore established. The global energy of $G(\boldsymbol{x},\boldsymbol{e})$ is formulated as Equation (3-7).

$$arg\ \min_{\mathbf{y}} E_G(\mathbf{y}) = \sum_{i \in \boldsymbol{x}} \Phi(\mathbf{y}_i) + \mu \sum_{(i,j) \in \boldsymbol{e}} \Psi(\mathbf{y}_i, \mathbf{y}_j) \tag{3-7}$$

where $\mathbf{y}$ is a labelling configuration whose value space is $\Omega$ = {*linearity*, *planarity*, *scattering*}, $i \in \boldsymbol{x}$ corresponds to a supervoxel, $(i, j) \in \boldsymbol{e}$ corresponds to the edge between two adjacent supervoxels, and $\mu$ ($\mu \in (0,1)$) is a constant parameter used to adjust the effectiveness of pairwise interactions.

The unary potentials presenting the fidelity to the local geometric characteristics are given as Equation (3-8).

$$\Phi(\mathbf{y}_i) = \begin{cases} 1 - f_l, & \text{if } \mathbf{y}_i = linearity \\ 1 - f_p, & \text{if } \mathbf{y}_i = planarity \\ 1 - f_s, & \text{if } \mathbf{y}_i = scattering \end{cases} \tag{3-8}$$

And the pairwise terms defined based on Potts model (Li, 2009) are given as

$$\Psi(\mathbf{y}_i, \mathbf{y}_j) = 1[\mathbf{y}_i \bullet \mathbf{y}_j] \tag{3-9}$$

where 1[•] is a binary function that equals to 0 if $\mathbf{y}_i = \mathbf{y}_j$, otherwise, 1[•] equals to 1.

The minimum global energy $\min_{\mathbf{y}} E(\mathbf{y})$ is approximated using graph cuts with α-expansion operations (Boykov et al., 2001) to obtain spatially smooth structural labelling **y**.

### 3.4.2 Structural Component Growing

After structural labelling, supervoxels with the same labels are merged into large segments, called structural components, by region growing (Vosselman et al., 2004). During the region growing, a supervoxel is first randomly selected as the seed of a structural component and its adjacent supervoxels are regarded as candidates. Only candidates having the same structural labels with the seed are merged into the structural components, and their adjacent supervoxels are pushed into the queue of the candidates. This process is repeated until no new candidates are found. Table 3.2 shows the pseudo-code for the structural component growing.

Table 3.2 Pseudo-code for the structural component growing.

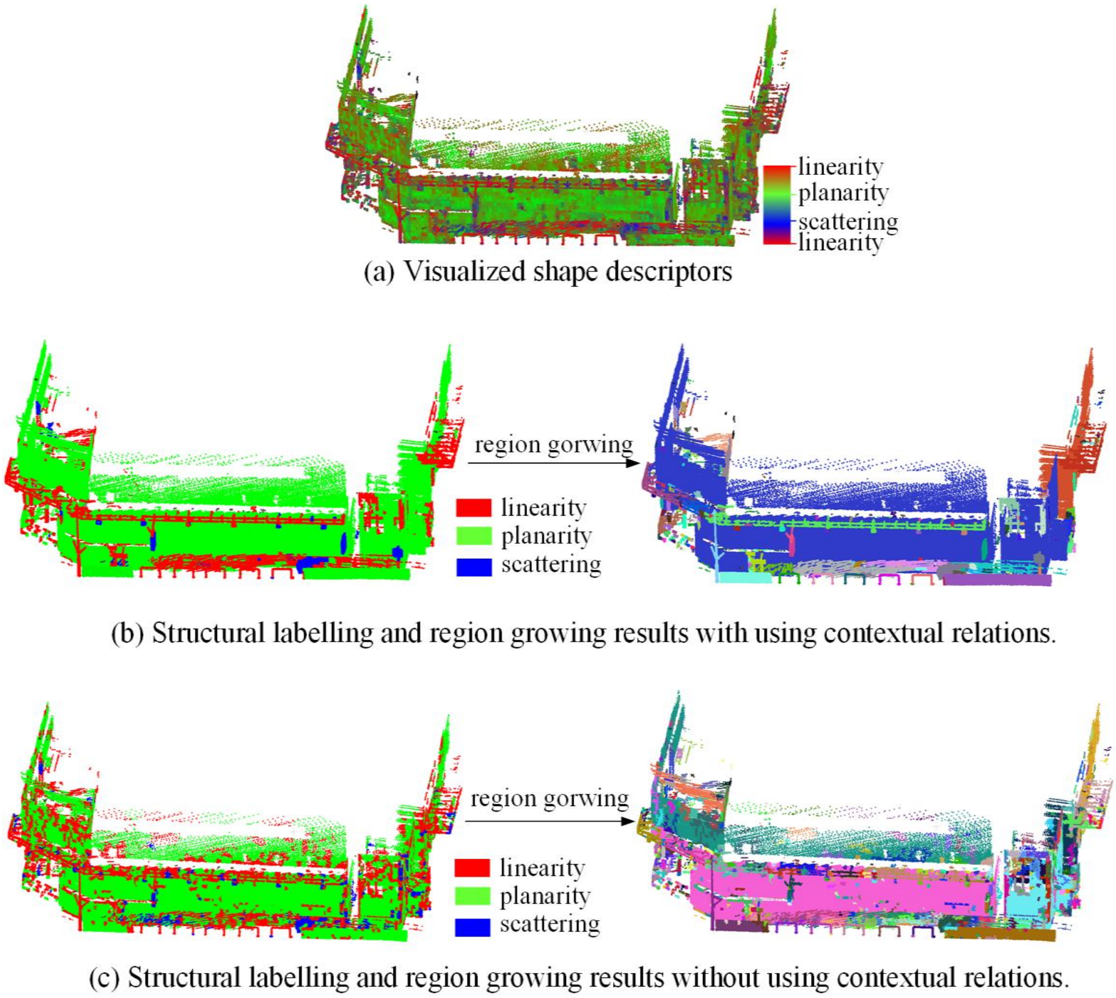
**Algorithm:** Structural component growing

**Input:** The supervoxels $\mathcal{V}$; the structural labelling $\mathbf{y}$; the adjacent relations between supervoxels $\mathbf{e}$.

**Output:** the structural components $\mathcal{S}$ with a corresponding structural labelling $\mathbf{Y}$.

Initialize: $\mathcal{S} = \varnothing$; $\mathbf{Y} = \varnothing$.
Initialize: Remaining supervoxels $\mathcal{V}' = \mathcal{V}$; remaining supervoxel labelling $\mathbf{y}' = \mathbf{y}$.
**while** $\mathcal{V}' \neq \varnothing$
  Initialize: structural component $S = \{\mathcal{V}'_0\}$; structural component label $y = \mathbf{y}'_0$.
  erase $\mathcal{V}'_0$ from $\mathcal{V}'$; erase $\mathbf{y}'_0$ from $\mathbf{y}'$.
  candidate supervoxels $C = \{V \in \mathcal{V}' \mid \exists e \in \boldsymbol{e},\ e == (S_0, \mathcal{V}'_i)\ \&\&\ \mathbf{y}'_i == y\}$.
  **while** $C \neq \varnothing$
    **for** $i = 1, \ldots, |\mathcal{V}'|$
      **if** $\exists e = (C_0, \mathcal{V}'_i) \in \boldsymbol{e}\ \&\&\ \mathbf{y}'_i == y$
        push $\mathcal{V}'_i$ to $C$: $C \leftarrow \mathcal{V}'_i$; push $C_0$ to $S$: $S \leftarrow C_0$.
        erase $C_0$ from $C$.
      **end if**
    **end for**
  **end while**
  push $S$ to $\mathcal{S}$: $\mathcal{S} \leftarrow S$; push $y$ to $\mathbf{Y}$: $\mathbf{Y} \leftarrow y$.
**end while**
returen $\mathcal{S}$ and $\mathbf{Y}$

The label of the structural component is inherently obtained from the supervoxels constituting it. With the structural labels, the sizes of structural components corresponding to different shape categories may vary significantly, and this contributes to the semantic segmentation because the discrimination is enhanced. In the final result, if two adjacent supervoxels are in two different structural components, the corresponding structural components are also regarded as adjacent. Figure 3.8 (a) and (b) show the process of generating structural components from supervoxels via MRF, where the contextual relations between supervoxels are encoded. The results are compared with the results without using the contextual relations between supervoxels (as shown in Figure 3.8 (c)). It can be seen that adopting the contextual relations can significantly improve the consistency of the structural labelling and result in a more reasonable decomposition of objects with structural components.




(a) Visualized shape descriptors

(b) Structural labelling and region growing results with using contextual relations.

(c) Structural labelling and region growing results without using contextual relations.

Figure 3.8 Generation of structural components using contextual relations between supervoxels compared to the results without using the contextual relations between supervoxels.

## 3.5 Summary and Discussion

This chapter presents the segmentation method that decomposes objects into structural components by using multiple-level relations, which include density isotropy at the point level, geometric (and radisometric) homogeneity at the point level and contextual information at the supervoxel level. These point relations are exploited to tackle all the possible issues that may hinder the segmentation, and they take effect at different stages of the segmentation and make the method more robust to noise and varying point density simultaneously compared with existing methods. The contextual information takes effect on the supervoxels via an MRF framework, resulting a spatially smooth and reasonable decomposition of the objects. Furthermore, because of changes of relations, such as the close distance between different objects, errors may be caused in the segmentation result. In fact, the adoptment of supervoxels and structural components provides a friendly way for interactive editing during manual quality control, which is inevitable in practice.

The structural components generated from this stage help to understand the objects in the point clouds at the geometric level. The structural information derived from the components can further help to interpret the scenes at the semantic level as presented in the next chapter.

# Chapter 4 Classification of Point Clouds Based on Contextual Relations

Classification (or semantic segmentation) is the process of object recognition with respect to semantic classes (e.g., buildings, trees, cars, and so forth) and it plays an essential role in 3D city modelling (Lafarge and Mallet, 2012; Verdie et al., 2015). Although many efforts have been made in this field, most of them made the inference, which is the key step of classification, only based on information derived from individual entities (e.g., points, supervoxels and planar segments) (Mallet et al., 2008; Weinmann et al., 2015a; Weinmann et al., 2017). Contextual information has been used as knowledge-based rules or as a smooth constraint in graphical models (Niemeyer et al., 2014; Vosselman et al., 2017; Zhu et al., 2017), which, however, only takes effect during the refinement of the classification.

In reality, objects can be very complex and formed by various structures of different shapes, especially for street facilities presented in MLS data with abundant details. Introducing structural information into the inference step of the classification can considerably improve the accuracy of classification. To this end, a point cloud classification method based on contextual relations between structural components is developed in this research. The contextual relations used here are specially associated with structural labels, namely *linearity*, *planarity* and *scattering*, which inherently exist in the structural components generated from the last stage. By taking advantage of such structural information, the proposed method can therefore distinguish between multiple classes of objects, even with local or global similarities.

This chapter presents the developed point cloud classification method and is organised as follows. Section 4.1 shows an overview of the proposed method, where the contextual relations are embedded in a CRF framework. Section 4.2 presents how the contextual relations are abstracted as unary and pairwise features, which are further used to train two independent RF classifiers and make inferences corresponding to the unary and pairwise potentials of the CRF, as described in Section 4.3. In Section 4.4, the global energy of the CRF is optimised with full-range contextual information. Finally, A summary of the proposed classification method and discussions are given in Section 4.5.

## 4.1 Overview of the Point Cloud Classification Method

The proposed point cloud classification method is to assign a semantic label (e.g., *building*, *trees*, *cars*, and so forth) to each structural component through a high-order CRF framework as shown in Figure 4.1. The high-order CRF is established on the basis of supervoxels and structural components generated from the segmentation stage, and high-order regions further defined on the basis of structural components. Contextual relations derived from these multiple entities, which inherently have structural labels (*linearity*, *planarity* and *scattering*), are integrated into different parts of the high-order CRF to take effect in the inference and refinement steps of the semantic labelling. Finally, the semantic labelling result is combined with the ground points obtained from the segmentation stage to generate the final classification result.

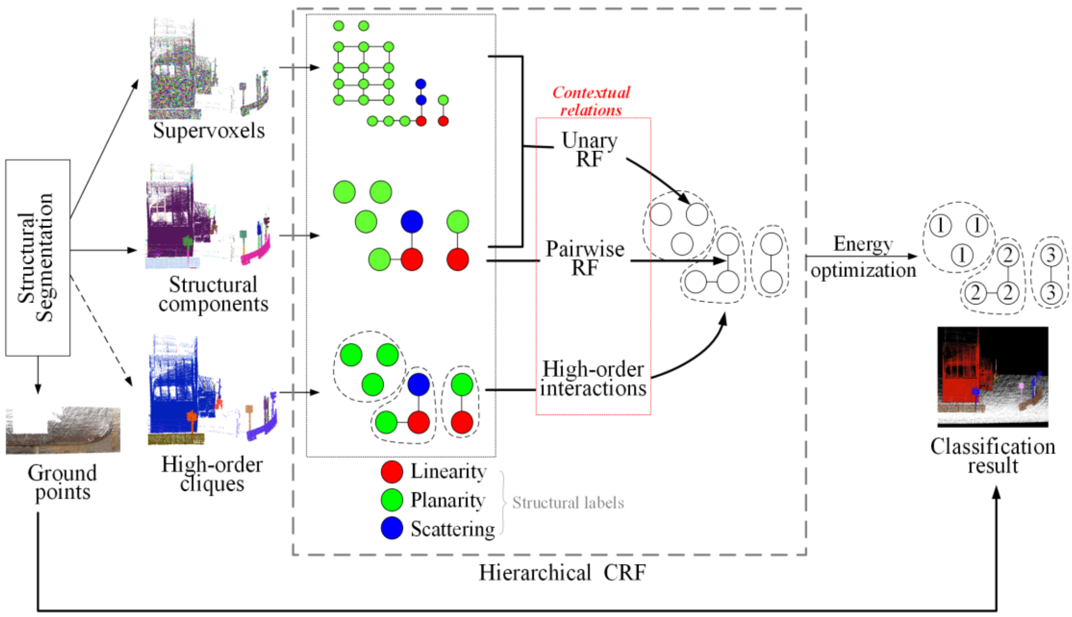



Figure 4.1 Overview of the classification method based on contextual relations embedded in a hierarchical CRF.

The high-order CRF framework is established based on an undirected graphical model $\mathcal{G}(\mathcal{S}, \mathcal{E}, \mathcal{R})$. $\mathcal{S}$ and $\mathcal{E}$ are the set of nodes and edges corresponding to the structural components and the adjacency relationships between them, respectively. $\mathcal{R}$ is the set of high-order regions generated by clustering adjacent or close structural components. The purpose of semantic labelling is to find a label configuration $\boldsymbol{f}$ that makes the global energy minimum as Equation (4-1).

$$arg\ \min_{\boldsymbol{f}} E_{\mathcal{G}}(\boldsymbol{f}) = \sum_{i \in \mathcal{S}} D(\boldsymbol{X}_i, \boldsymbol{f}_i) + \alpha \sum_{(i,j) \in \mathcal{E}} V(\boldsymbol{X}_{ij}, \boldsymbol{f}_i, \boldsymbol{f}_j) + \beta \sum_{R \in \mathcal{R}} H(\boldsymbol{f}_R) \quad (4\text{-}1)$$

where the first and second terms on the right-hand side are the unary and pairwise energies computed with the unary and pairwise RF classifiers. The contextual relations derived structural labels and the differences between supervoxels and structural components are used as important information while training the RFs. The third terms are high-order interactions, capturing the long-range contextual information in terms of label redundancies. Detailed descriptions of the training and inference with unary and pairwise RFs and global energy optimisation with full-range contextual information, are given below.

## 4.2 Discriminative Features Derived from Contextual Relations

Feature extraction is an essential step in the classification framework based on the classic machine learning algorithms. However, many previous studies mostly focused on unary features extracted from individual entities, but ignored the contextual relations between the entities. Even though some methods benefited from encoding simple contextual information as pairwise features into the MRF or CRF for the refinement of classification, the effects were limited. Therefore, in addition to the features extracted from individual entities or simple contextual relations, this research also investigates the feature extraction from contextual relations with structural labels as both unary and pairwise features, with a purpose to obtain a more accurate inference result for the classification.

### 4.2.1 Unary Features

In previous research, the unary features were generally extracted from individual entities and described the characteristics of entities independently. Extraction of such independent unary features have been extensively investigated and well-described in previous studies (Landrieu et al., 2017; Vosselman et al., 2017; Weinmann et al., 2015a; Yang et al., 2012; Zhu et al., 2017). Among such unary features, those pertaining to this work are summarised into three types, namely height, 2D and covariance, as shown in

Table 4.1, and they are extracted independently from the structural components. Besides, the structural labels of the structural components are also considered as a useful unary feature in this research.

Table 4.1 Unary features derived from individual structural components.

| ***Feature type*** | ***Description*** | ***Dimension*** |
|---|---|---|
| **Height** | Maximum and minimum heights above the ground. | 2 |
| **2D** | Area, perimeter and shape constraint (Yang et al., 2012) of the structural components projected on xy-plane. | 3 |
| **Covariance** | Features derived from the covariance matrix via principle component analysis, including linearity, planarity, scattering, length, area, volume, omni-variance, anisotropy, eigen-entropy and change of curvature (Landrieu et al., 2017; Weinmann et al., 2015a; Weinmann et al., 2017). | 10 |
| **Structural label** | Structural label of the structural component. | 1 |

In addition, this research also investigates the extraction of unary features from contextual information to distinguish between structural components having similar appearances, but belonging to different objects, such as the poles of traffic lights and traffic signs. The contextual relations here are twofold.

First, they include “above” and “below” relations derived from adjacent structural components with respect to structural labels, as shown in Figure 4.2. The “above” and “below” relations are defined based on the height features. For a structural component, if both the maximum and minimum heights are larger than those of its neighbour, it is “above” its neighbour, and if both less, it is “below” (such as the traffic sign, traffic light and tree shown in Figure 4.2). If there is no “above” or “below” neighbour (such as the building façade and its appurtenances shown in Figure 4.2), the corresponding feature is set as 0. If there is more than one neighbour, the closest one is then selected.

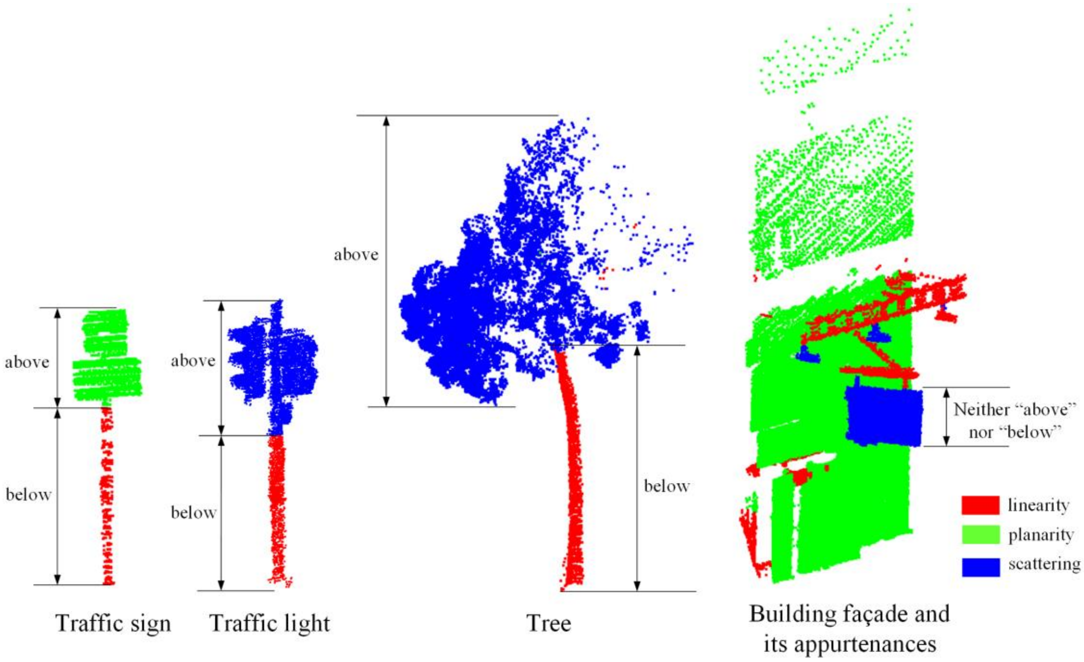



Figure 4.2 "Above" and "below" relationships between structural components.

Second, they refer to the geometric differences between structural components and their occupied supervoxels as shown in Figure 4.3. This kind of differences presents the geometric variations of structural components from local to global scales and this can be an essential clue for object recoginisation. For example, for a thick tree trunk, the local geometry might be planar if the diameter of the trunk is much larger than the supervoxel sizes, but as a whole, the trunk presents a linear shape. It is similar to the tree branches and cars, which have linear and planar shapes at local scales but scatter shapes at the global scales.

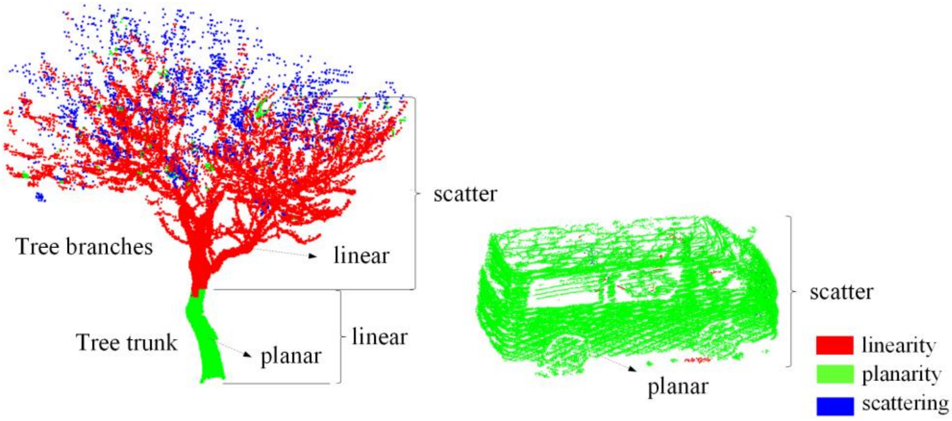



Figure 4.3 Geometric differences between structural components and their occupied supervoxels.

The detailed descriptions of the unary features derived from contextual relations are shown in Table 4.2. All of the unary features shown in Table 4.1 and Table 4.2 are normalised to the interval of [0~1] by the min-max normalisation (Ge et al., 2019). At this point, each node $i \in \mathcal{S}$ thus corresponds to a structural component with a normalised feature vector $\boldsymbol{X}_i$.

Table 4.2 Unary features derived from contextual relations.

| ***Feature type*** | ***Description*** | ***Dimension*** |
|---|---|---|
| ***Features derived from structural labels of adjacent structural components*** | | |
| **Neighbours with linear labels** | 2D shape constraint and length of above and below adjacent structural components labelled as *linearity*. | 4 |
| **Neighbours with planar labels** | 2D shape constraint and area of above and below adjacent structural components labelled as *planarity*. | 4 |
| **Neighbours with scatter labels** | Volume of above and below adjacent structural component labelled as *scattering*. | 2 |
| ***Features derived from the differences between structural components and occupied supervoxels*** | | |
| **Covariance differences** | For each structural component, 10 covariance features are also derived from its occupied supervoxels. The differences are calculated by covariance features derived from the structural component minus the average value of those derived from the supervoxels. | 10 |

### 4.2.2 Pairwise Features

Pairwise features are used here to directly describe the contextual relations between structural components. One type of pairwise features is defined on the basis of the intersecting points between adjacent structural components. The intersecting points refer to points occupied by two supervoxels that belong to different structural components as shown in Figure 4.4. Further information in terms of height, 2D and covariance are extracted from the intersecting points as pairwise features as shown in Table 4.3. Another type of pairwise features is derived from the structural labels of adjacent structural components. This type of pairwise features is abstracted as the combination of structural labels, including *linearity-planarity*, *planarity-scattering*, *scattering-linearity*, as described in Table 4.3. All of the pairwise features are normalised in the same way with the unary features and form a pairwise feature vector $\boldsymbol{X}_{ij}$ for each pair of adjacent structural components $(i,j) \in \mathcal{E}$.

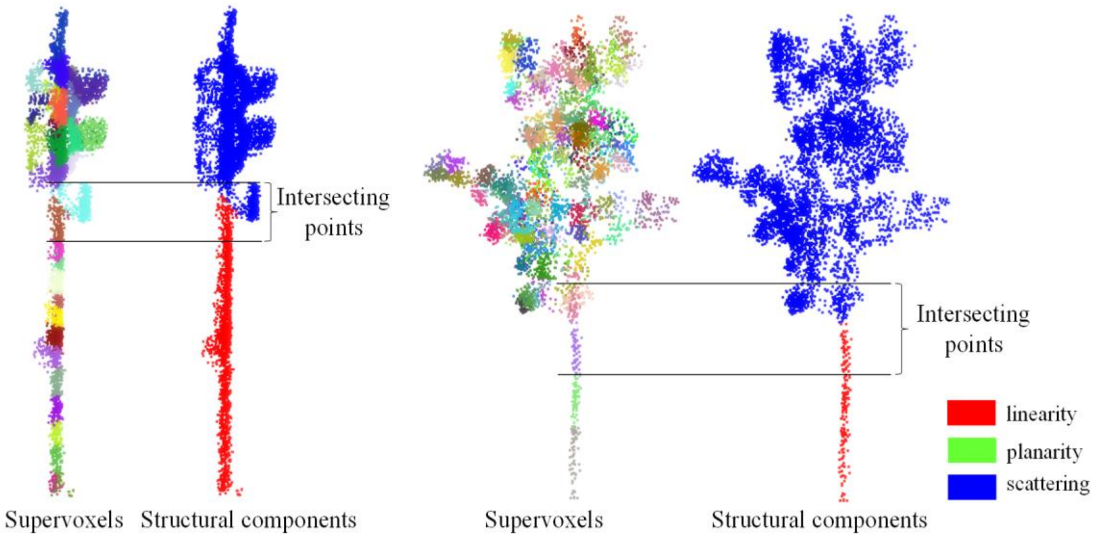



Figure 4.4 Intersecting points between adjacent structural components.

Table 4.3 Pairwise features extracted from adjacent structural components.

| ***Feature type*** | ***Description*** | ***Dimension*** |
|---|---|---|
| ***Features derived from intersecting points*** | | |
| **Height** | Maximum and minimum heights of intersecting points above the ground. | 2 |
| **2D** | Area, perimeter and shape constraint of the intersecting points projected on xy-plane. | 3 |
| **Covariance** | Covariance features (e.g., linearity, planarity, scattering, omni-variance, etc.) derived from the covariance matrix of intersecting points. | 10 |
| ***Features derived from structural labels*** | | |
| **Label combination** | Label combination types, i.e., *linearity-planarity*, *planarity-scattering*, *scattering-linearity* (the combination is unordered). | 1 |

## 4.3 Training and Inference

At this stage, two independent RF classifiers are trained using the above described unary and pairwise features, respectively. The RF is a well-known bootstrap ensemble learning algorithm consisting of a number of randomised decision trees and it is adopted in this work for the following reasons. First, it is quite suitable for multiple-class classification and it can handle many features (Gislason et al., 2006), and second, it can provide a good trade-off between classification accuracy and computational efficiency (Hackel et al., 2016; Weinmann et al., 2017).

With respect to the training of the RF classifiers, as noted by Landrieu and Simonovsky (2018), very small segments may harm the training of classifiers, therefore a minimum point number $n_{min} = 40$ (as in Landrieu and Simonovsky (2018)) is set for the selection of valid structural components as training samples. The unary RF is trained with feature vectors $\{\boldsymbol{X}_i\}$ to distinguish between multiple semantic classes, whilst the pairwise RF is only trained to learn whether the pairs of nodes connected by the edges belong to the same class based on feature vectors $\{\boldsymbol{X}_{ij}\}$. At the same time, to guarantee an even distribution of sampling during the training, prior class probabilities are defined as the quotients of the total sample sizes dividing the maximum sample size. Table 4.4 shows an example of the sample size distribution of three classes, where the maximum sample size is 3079, and the prior class probability of each class is computed by the total sample size dividing the maximum sample size. In this research, each RF contains $n_T = 500$ trees, and each tree is trained independently on a randomly selected subset of the training samples. The maximum depths of the trees are set as $\sqrt{N(\boldsymbol{X})}$, where $N(\boldsymbol{X})$ refers to the number of dimensions of feature vector $\boldsymbol{X}$.

Table 4.4 An example of the computation of prior class probabilities based on the size distribution of training samples.

| | Linear sample size | Planar sample size | Scatter sample size | Total sample size | Prior class probability |
|---|---|---|---|---|---|
| *Building* | 1735 | 1165 | 179 | **3079** | 1 |
| *Tree* | 176 | 23 | 28 | 227 | 13.56 |
| *Car* | 173 | 59 | 15 | 247 | 12.46 |

During the inference process, the feature vector $\boldsymbol{X}$ corresponding to a node or an edge of unknown class in the graphical model $\mathcal{G}$ is presented to the corresponding RF and each tree in the RF castes a vote for the most likely class. The unary energy in Equation (4-1) then is calculated based on the votes of trees in the unary RF using Equation (4-2).

$$D(\boldsymbol{X}_i, \boldsymbol{f}_i) = 1 - \exp\left(N_{\boldsymbol{f}_i}(\boldsymbol{X}_i) / n_{T_{unary}}\right) \tag{4-2}$$

where $N_{\boldsymbol{f}_i}(\boldsymbol{X}_i)$ is the number of votes for the class labelled as $\boldsymbol{f}_i$ based on the feature vector $\boldsymbol{X}_i$, $n_{T_{unary}}$ is the total number of trees in the RF.

For the pairwise issues, the RF is only used to determine whether the two connected nodes belong to the same class or not. Obviously, this designation will markedly reduce the number of target classes so that the accuracy of the predictions of the pairwise RF will be improved. Similarly, the pairwise energy in Equation (4-1) is defined as

$$V(\boldsymbol{X}_{ij}, \boldsymbol{f}_i, \boldsymbol{f}_j) = 1 - \exp\left(N_{1[\boldsymbol{f}_i \bullet \boldsymbol{f}_j]}(\boldsymbol{X}_{ij}) / n_{T_{pair}}\right) \tag{4-3}$$

where $N_{1[\boldsymbol{f}_i \bullet \boldsymbol{f}_j]}$ refers to the number of votes for $\boldsymbol{f}_i = \boldsymbol{f}_j$ (if $\boldsymbol{f}_i = \boldsymbol{f}_j$, $1[\boldsymbol{f}_i \bullet \boldsymbol{f}_j] = 0$) or $\boldsymbol{f}_i \neq \boldsymbol{f}_j$ (if $\boldsymbol{f}_i \neq \boldsymbol{f}_j$, $1[\boldsymbol{f}_i \bullet \boldsymbol{f}_j] = 1$), and $n_{T_{pair}}$ is the total number of trees in the pairwise RF.

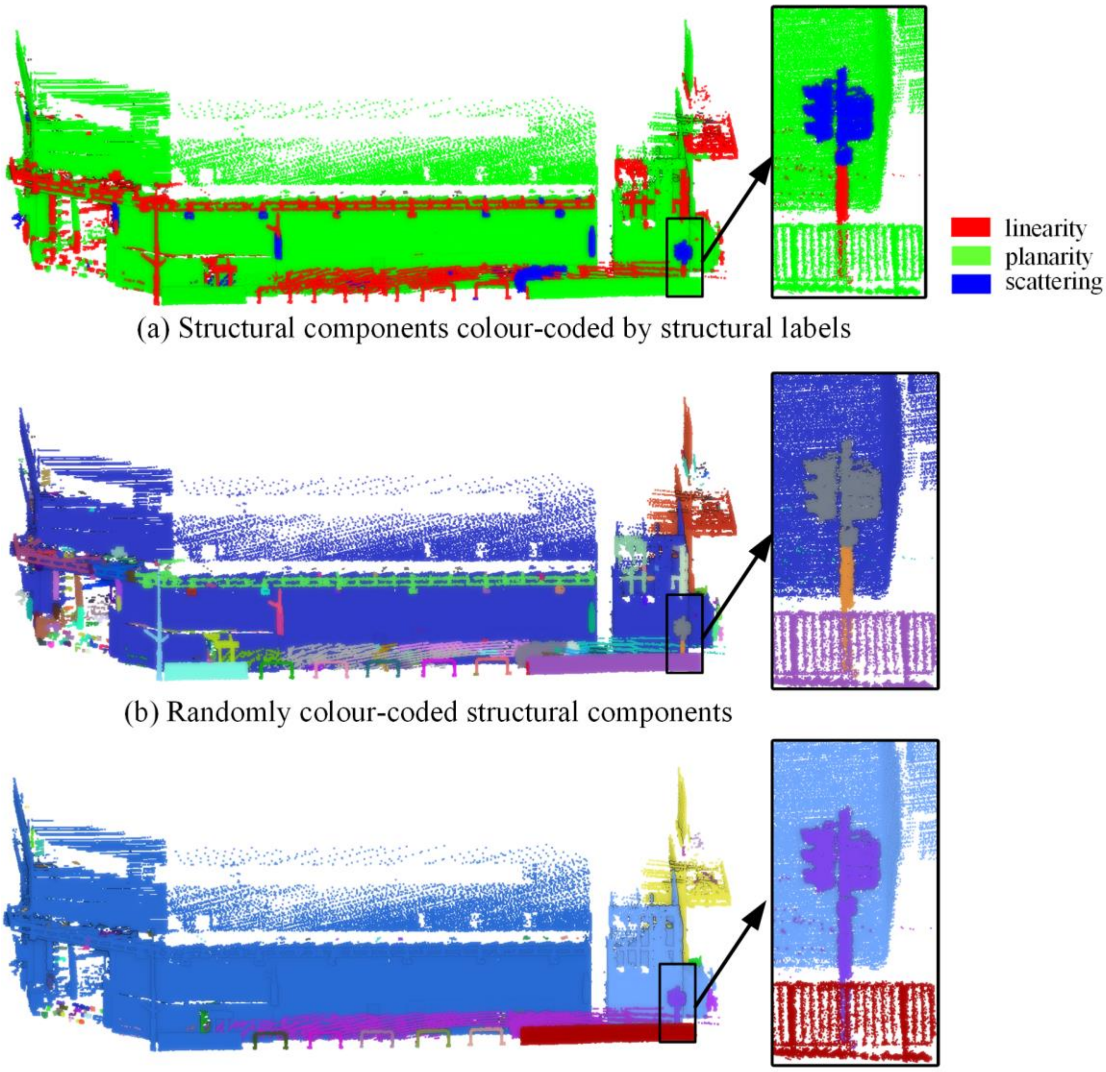
At this point, the unary and pairwise energies of the CRF model in Equation (4-1) are constructed, and these energies and the high-order energies introduced by regional label costs are further optimised as described below.

## 4.4 Global Energy Optimisation with Full-Range Contextual Information

The CRF provides a probabilistic framework for context-based classification. Relatively shorter-range contextual information has already been captured by the unary and pairwise energies of the CRF as described above. In this section, the longer-range contextual information are investigated and modelled by the high-order energies of the CRF. So that global energy optimisation of the CRF can take advantage of considering full-range contextual information.

### 4.4.1 Declaration of High-order Cliques

Although the point density of the laser scanning point may vary with respect to the changes of scanning distance, the variation on the same object surface is believed to be consistent. Therefore, it can be assumed that the length of labels within a density-consistent region should be as short as possible. During the expansion of supervoxels (see Section 3.3.2), the adjacency relationships between supervoxels are defined based on points that are mutually in each other's neighbourhood. Therefore a density-consistent region can be defined as a cluster of adjacent supervoxels. Simultaneously, as the structural components are clusters of adjacent supervoxels with the same structural labels (see Section 3.4.2), which are the subsets of a density-consistent region, it is not hard to imagine that a density-consistent region is also a cluster of adjacent structural components as shown in Figure 4.5. Furthermore, small density-consistent regions that contain fewer than 100 points are merged into the closest large regions if the shortest distance between them falls below a given threshold of 0.5 m (as shown in Figure 4.6), in order to reduce the fragmentation effect in the final semantic labelling result, which could be common on the far sides of the scanner, such as the tops of tree crowns and building facades.

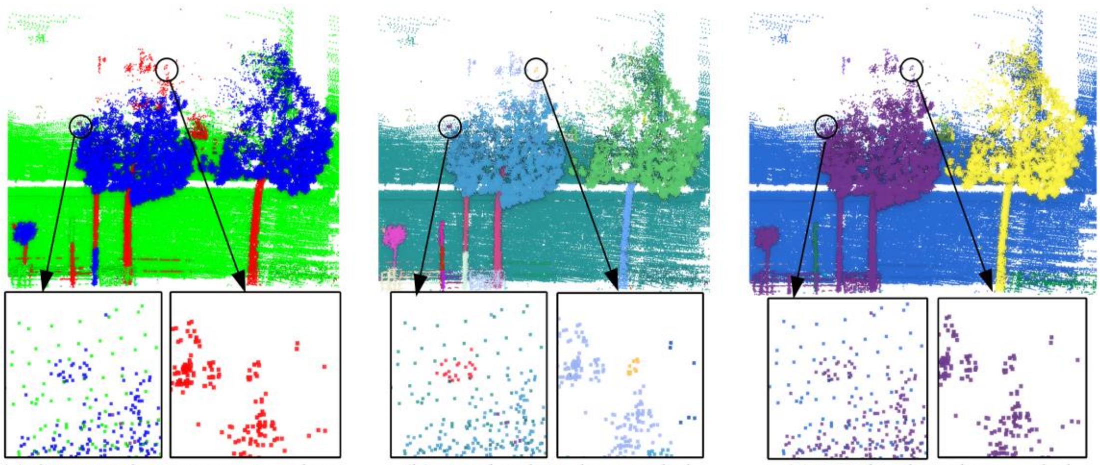



(a) Structural components colour-coded by structural labels

(b) Randomly colour-coded structural components

(c) Randomly colour-coded density-consistent regions

Figure 4.5 Density-consistent regions generated by clustering adjacent or close structural components.



(a) Structural components colour-coded by structural labels　(b) Randomly colour-coded structural components　(c) Randomly colour-coded density-consistent regions

Figure 4.6 Small density-consistent regions are merged into the closest large regions.

At this point, each density-consistent region, noted as $R$, is generated as a set of structural components, which means $R = \{S_i \mid S_i \in \mathcal{S}\}$ ($R \in \mathcal{R}$). Therefore, the high-order label cost (Delong et al., 2012), which measures the label redundancy within a density consistent region $R$, can be modelled with the high-order term $H(\boldsymbol{f}_R)$ in Equation (4-1) as follow

$$H(\boldsymbol{f}_R) = \sum_{l \in \mathcal{L}} h_l^R \cdot \delta(l, \boldsymbol{f}_R) \tag{4-4}$$

where $h_l^R$ is the per-label cost assigned to each label $l$ ($l \in \mathcal{L}$, $\mathcal{L}$ is the namespace of the semantic labelling $\boldsymbol{f}$) defined as in the work of (Luo et al., 2018), and $\delta(\cdot)$ is an indicator function formulated as

$$\delta(l, \boldsymbol{f}_R) = \begin{cases} 1, & \text{if } \exists i \in R\colon \boldsymbol{f}_i = l \\ 0, & \text{otherwise} \end{cases} \tag{4-5}$$

### 4.4.2 Energy Optimisation

The optimisation of the global energy with regional label cost can be NP-hard (Delong et al., 2012). But as the density-consistent regions are all spatially separated in this work, therefore there is no edge $(i,j) \in \mathcal{E}$ striding across a region $R$, the global energy of the CRF in Equation (4-1) can therefore be converted into the following format

$$arg\ \min_{\boldsymbol{f}} E_{\mathcal{G}}(\boldsymbol{f}) = \sum_{R\in\mathcal{R}}\left(\sum_{i\in R} D(\boldsymbol{X}_i,\boldsymbol{f}_i) + \alpha \sum_{(i,j)\in E} V(\boldsymbol{X}_{ij},\boldsymbol{f}_i,\boldsymbol{f}_j) + \beta \sum_{l\in\mathcal{L}} h_l^R \cdot \delta(l,\boldsymbol{f}_R)\right) \quad (4\text{-}6)$$

where $E \subseteq \mathcal{E}$ is the subset of edges that correspond to the adjacency relationships between structural components contained in $R$.

Equation (4-6) transforms the problem of minimising the global energy of $\mathcal{G}(\mathcal{S}, \mathcal{E}, \mathcal{R})$ with regional label costs into a problem of minimising the global energy of each subgraph $G(R, E)$ with global label costs, as shown in Equation (4-7) and Equation (4-8).

$$\min_{\boldsymbol{f}} E_{\mathcal{G}}(\boldsymbol{f}) = \sum_{G\subseteq\mathcal{G}}\left(\min_{\boldsymbol{f}'} E_G(\boldsymbol{f}')\right) \quad (4\text{-}7)$$

$$\min_{\boldsymbol{f}'} E_G(\boldsymbol{f}') = \sum_{i\in R} D(\boldsymbol{X}_i,\boldsymbol{f}_i') + \alpha \sum_{(i,j)\in E} V(\boldsymbol{X}_{ij},\boldsymbol{f}_i',\boldsymbol{f}_j') + \beta \sum_{l\in\mathcal{L}} h_l \cdot \delta(\boldsymbol{f}') \quad (4\text{-}8)$$

where $\boldsymbol{f}'$ ($\boldsymbol{f}' \subseteq \boldsymbol{f}$) is a labelling configuration for all of nodes in the subgraph $G(R, E)$. The minimum energy presented by Equation (4-8) can be approximated using the α-expansion algorithm, in which the label costs are encoded by a test-and-reject approach (Delong et al., 2012). The weighting parameter $\alpha$ is adjusted between the range of 0 ~ 1, and empirically, $\alpha$ being set between 0.3 ~ 0.5 leads to a reasonably smooth classification result. The other parameter $\beta$ controls the weight of the label costs, and it imposes more costs on the number of used categories with a larger value. According to Luo et al. (2018), 20 ~ 120 is a reasonable range for the value of $\beta$. In this research, $\beta$ is set as 20 and it can effectively refine the fragmentation effect in the classification result as shown in Figure 4.7.

After energy minimisation, a spatially smooth labelling, which assigns each structural component a semantic label, is obtained. These semantic labels are sequentially passed to the points, and the labelled points are fused with the ground points obtained from the segmentation stage to generate the final classified point cloud.

Figure 4.7 shows the classification result of two test datasets using the high-order CRF compared with the results of the unary RF predictions and the pairwise CRF (only the unary and pairwise energies are optimised). It can be seen that the unary RF prediction result has the heaviest fragmentation effect. The pairwise CRF, which considers more short-range contextual information, to some extent smoothens the semantic labelling result. And the high-order CRF is able to make the labelling spatially smooth, even for isolated segments, by taking advantage of full-range contextual information.

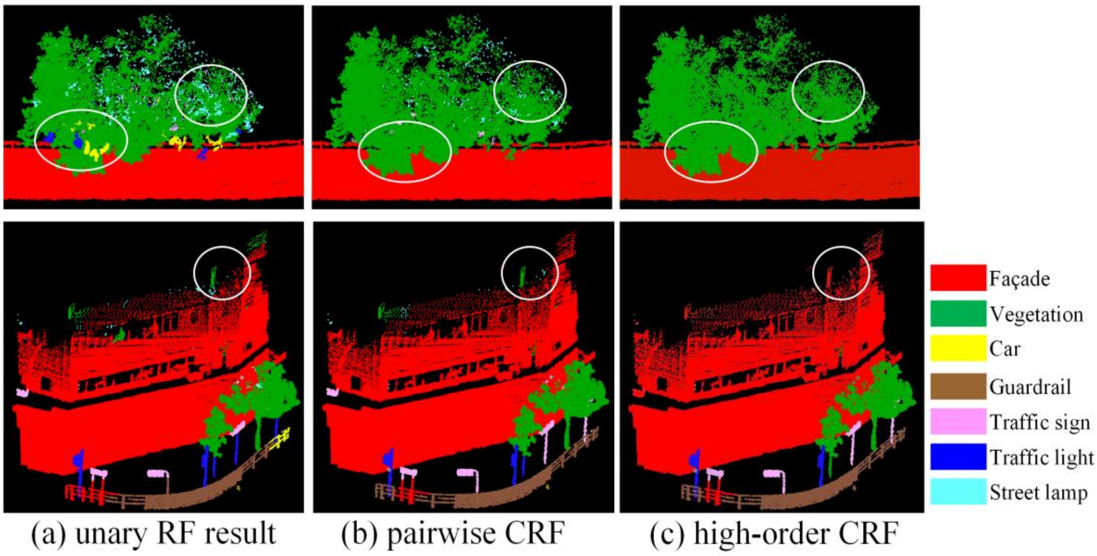



Figure 4.7 Classification result of the high-order CRF ($\alpha = 0.3$, $\beta = 20$) compared to the results of unary RF and pairwise CRF ($\alpha = 0.3$).

## 4.5 Summary and Discussion

This chapter presents the contents of the three steps involved in the classification process via energy optimisation of a high-order CRF, which is established based on the supervoxels and structural components with structural labels. Multiple levels and types of contextual relations are derived and encoded into different parts of the CRF to make the classification accurate and spatially smooth. First, contextual information derived from the structural labels of neighbours, and geometric differences between the local and global scales, together with the general 2D and 3D information, are abstracted as unary features for the semantic inference of individual structural components. Second, contextual relations between structural components are derived from their structural labels and the intersecting points, in order to be used for the pairwise inference judging whether or not two adjacent structural components belong to the same class. Long-range contextual information is modelled by the regional label costs, which are defined on the density-consistent regions and take effect or isolated fragments.

The classified point cloud with semantic labels provides an intuitive interpretation about the scene presented. However, they cannot be directly used in GIS systems or applications because of its discrete property and great data volume. In consideration of such fact, the next chapter will further explore the generation of 3D models, which conform to CigyGML, from the classified point clouds.

# Chapter 5 3D Reconstruction of CityGML Building Models Constrained by Topological Relations

After the point cloud classification, the point clusters corresponding to specific classes can be extracted according to the semantic labels assigned to the points, which provide the fundamental data for the reconstruction of 3D objects. As buildings are the key features in urban areas, this research focuses on the 3D reconstruction of building models. This issue has gained increasing attention in academic communities in the past two decades, and various methods have been developed, including model-driven methods (Huang et al., 2013; Kada and McKinley, 2009), data-driven methods (Chen et al., 2017; Poullis, 2013; Sampath and Shan, 2010; Sohn et al., 2008; Vosselman and Dijkman, 2001) and hybrid-driven methods (Elberink and Vosselman, 2009; Lin et al., 2013; Verma et al., 2006; Xiong et al., 2015). However, these methods only focus on the modelling of building rooftops and produce 2.5D building models. A recent trend that can generate true 3D building models is to partition the 3D space into a set of basic units and then use these basic units to approximate the building surfaces (Li et al., 2016a; Li et al., 2016b; Nan and Wonka, 2017; Verdie et al., 2015). However, the scalability of these methods is hindered by the lack of topological-relation constraints.

In this research, an innovative method is developed for the 3D reconstruction of building models constrained by topological relations. This method adopts a space-partition-and-approximation strategy, but unlike previous studies, the topological relations between the basic units, which can be regarded as Constructive Solid Geometries (CSGs) (Piekarski and Thomas, 2001), are extracted after the space partitioning and are then used as constraints in the approximation step to select the optimal basic units forming the CSG models of buildings. The topological-relation constraints enhance the fidelity of building models to the input point clouds, and to some extent, make the models regularised. The conversion from the CSG models to the CityGML models, a type of boundary representations, is also investigated in this research to provide a complete pipeline of 3D building reconstruction for 3D GIS applications.

This chapter is organised as follows. Section 5.1 presents an overview of the 3D building reconstruction method. Section 5.2 illustrates the generation of basic 3D cells, which are CSGs, based on a half BSP tree, where the geometric relations are used to regularise the space partition. Section 5.3 describes the extraction of topological relations between basic 3D cells as facet and edge features and the optimal selection of the basic cells that approximate the buildings. The polygonal models of buildings formed by the selected basic cells are then converted into CityGML models in Section 5.4. Finally, a summary and discussions are given in Section 5.5.

## 5.1 Overview of the 3D Reconstruction Method

Figure 5.1 gives an overview of the 3D reconstruction method based on topological relations. This method consists of three steps. In the first step, planar primitives are extracted from the building point cloud by RANSAC and refined based on geometric relation constraints. The refined planar primitives are then used to partition the 3D space occupied by the building bounding box (the root CSG) into a set of basic cells (the leaf CSGs) based on a half BSP tree. The second step selects a set of cells that best approximates the building surface, and this can be modelled by a binary labelling problem and be solved by integer linear programming (ILP) (Schrijver, 1998). Simultaneously, the topological relations between the cells are extracted as facet and edge features, which are encoded in the objective function and constraints of the ILP problem to make the modelling results more accurate and regularised. Taking advantage of CSGs that the objects are solid if all the primitive shapes are solid (Piekarski and Thomas, 2001), the polygonal models formed by the selected cells are guaranteed to be watertight. The last step analyses the surface features of the polygonal models conformed by the selected cells and decomposes the model surface into different components. These components are then recognised as specific surface types defined by CityGML based on a set of rules, to finally produce 3D building models conforming to CityGML.

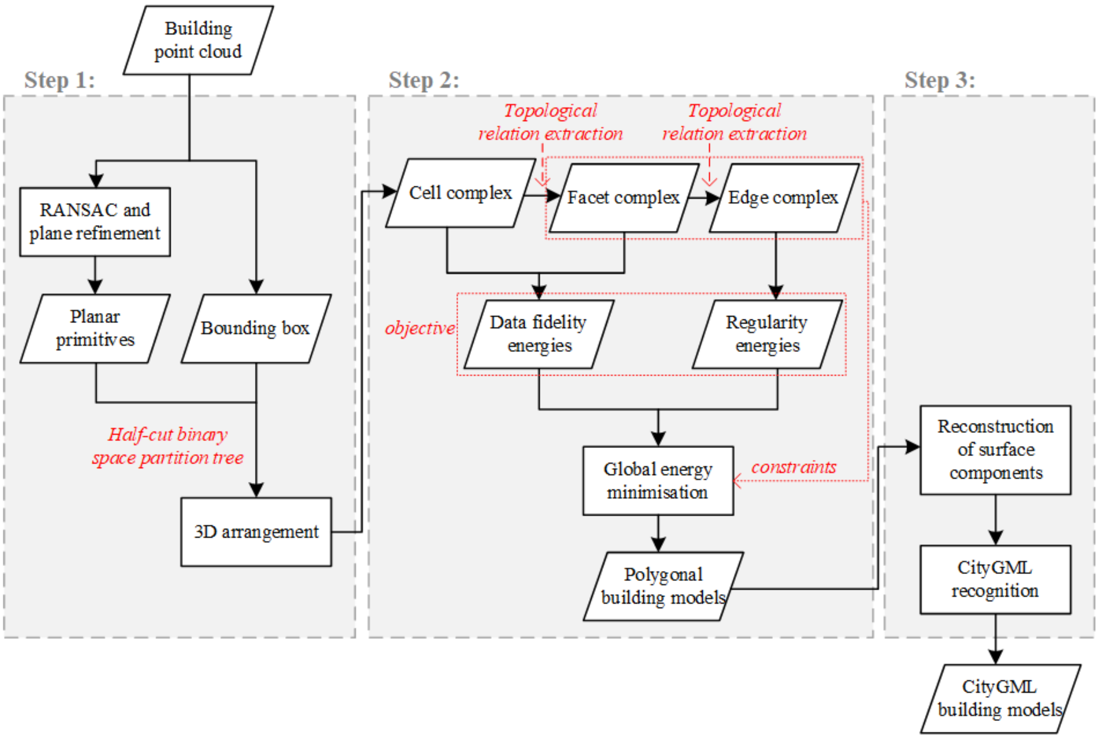



Figure 5.1 Overview of the topological-relation constrained 3D reconstruction of CityGML building models.

## 5.2 Generation of Cell Complex via 3D Arrangement

In this reconstruction method, a set of 3D cells called a cell complex is used to present the 3D space occupied by the bounding box of a building. These cells have simple and convex geometries and are compactly connected with each other. The 3D cells are presented in the CSG way, therefore, a subset of the cell complex will facilitate the watertight and manifold polygonal modelling of buildings without any intersecting computation to determine the edges and corner points. The following presents the generation of the cell complex via a 3D arrangement with a set of planar primitives extracted from building point clouds.

### 5.2.1 Extraction and Refinement of Planar Primitives with Geometric Relations

Most buildings are constituted of planar components that conform to some common regularity rules in terms of geometric relations, e.g., parallelism, orthogonality, symmetry and so forth. If a building contains curved surfaces, they can be approximated by multiple planar surfaces. In this research, a set of initial planar primitives $\mathbb{P} = \{P_1, P_2, \ldots, P_n\}$ are extracted using RANSAC (Schnabel et al., 2007) and simultaneously, the planar coefficients, including the normal vector $\boldsymbol{n}_i$ and a distance coefficient $d_i$ corresponding to each $P_i \in \mathbb{P}$, are obtained.

To enhance the regularities between the planar primitives, the initial planar primitives are refined based on a set of rules for geometric relationships, including *parallelism*, *orthogonality*, *z-symmetry* and *co-planarity*, as defined in Verdie et al. (2015). In addition, three more relations, *verticality*, *horizontality* and *xy-parallelism*, are also considered in this work. These six geometric relations are mathematically described in Table 5.1, with an angle threshold $\varepsilon$ and a Euclidean distance $d$. In Table 5.1, $\boldsymbol{n}_z$ denotes the unit vector in the z-axis and $\boldsymbol{n}^{xy}$ denotes the projection of normal $\boldsymbol{n}$ on the xy-plane.

Table 5.1 Geometric relations used to refine the building planar primitives.

| ***Relation type*** | ***Description*** |
|---|---|
| *Horizontality* | $P_i$ is vertical if $\theta(\boldsymbol{n}_i, \boldsymbol{n}_z) < \varepsilon$. |
| *Verticality* | $P_i$ is vertical if $\theta(\boldsymbol{n}_i, \boldsymbol{n}_z) > \pi/2 - \varepsilon$. |
| *Parallelism* | $P_i$ and $P_j$ are parallel if $\theta(\boldsymbol{n}_i, \boldsymbol{n}_j) < \varepsilon$. |
| *Orthogonality* | $P_i$ and $P_j$ are orthogonal if $\theta(\boldsymbol{n}_i, \boldsymbol{n}_j) > \pi/2 - \varepsilon$. |
| *Z-symmetry* | $P_i$ and $P_j$ are z-symmetric if $\lvert\theta(\boldsymbol{n}_i, \boldsymbol{n}_z) - \theta(\boldsymbol{n}_j, \boldsymbol{n}_z)\rvert < \varepsilon$. |
| *XY-Parallelism* | $P_i$ and $P_j$ are xy-parallel if $\theta(\boldsymbol{n}_i^{xy}{}_i,\ \boldsymbol{n}_j^{xy}) < \varepsilon$. |
| *Co-planarity* | For two parallel plane primitives $P_i$ and $P_j$, $P_i$ and $P_j$ are co-planar if $\lvert d_i - d_j\rvert < d$. |

As the geometric relations are mutually defined based on multiple planar primitives, there can be conflicts between the geometric relations. For example, a planar primitive $P$ can be parallel to $P_i$ and, simultaneously, orthogonal to $P_j$, where $P_i$ and $P_j$ do not meet the orthogonality condition. To handle such conflicts, a geometric relation priority rank is defined as *horizontality* = *verticality* > *parallelism* > *orthogonality* > *z-symmetry* = *xy-parallelism* > *co-planarity*. Based on this priority rank, the initial planar primitives are refined as shown in Figure 5.2, where "//", "⊥", "⇔" and "$//^{xy}$" denote the parallel, orthogonal, z-symmetric and xy-parallel relations, respectively. First, the initial planar primitives are clustered into horizontal, vertical and oblique plane clusters based on the vertical and horizontal relations. For the horizontal planes, the normal vectors are forced to be $\boldsymbol{n}_z$. The normal vectors of the vertical planes are first forced to be orthogonal to $\boldsymbol{n}_z$ and later adjusted in the xy-plane. The vertical and oblique planes are further clustered based on the parallel relation and the averaged normal is computed for each plane cluster. If the average normal is parallel, orthogonal or z-symmetric to any existing refined normal $\boldsymbol{n}'$ (the first refined normal is $\boldsymbol{n}_z$ if the horizontal plane exists), it will be adjusted according to $\boldsymbol{n}'$; otherwise, the averaged normal will be added into the set of refined normals. For the oblique planes, the xy-parallel relation is also checked with the existing refined normals. This process is repeated with an order from the vertical plane clusters to the oblique plane clusters until all of the plane clusters are refined. Finally, the refined plane clusters are further clustered based on the co-planar relation.

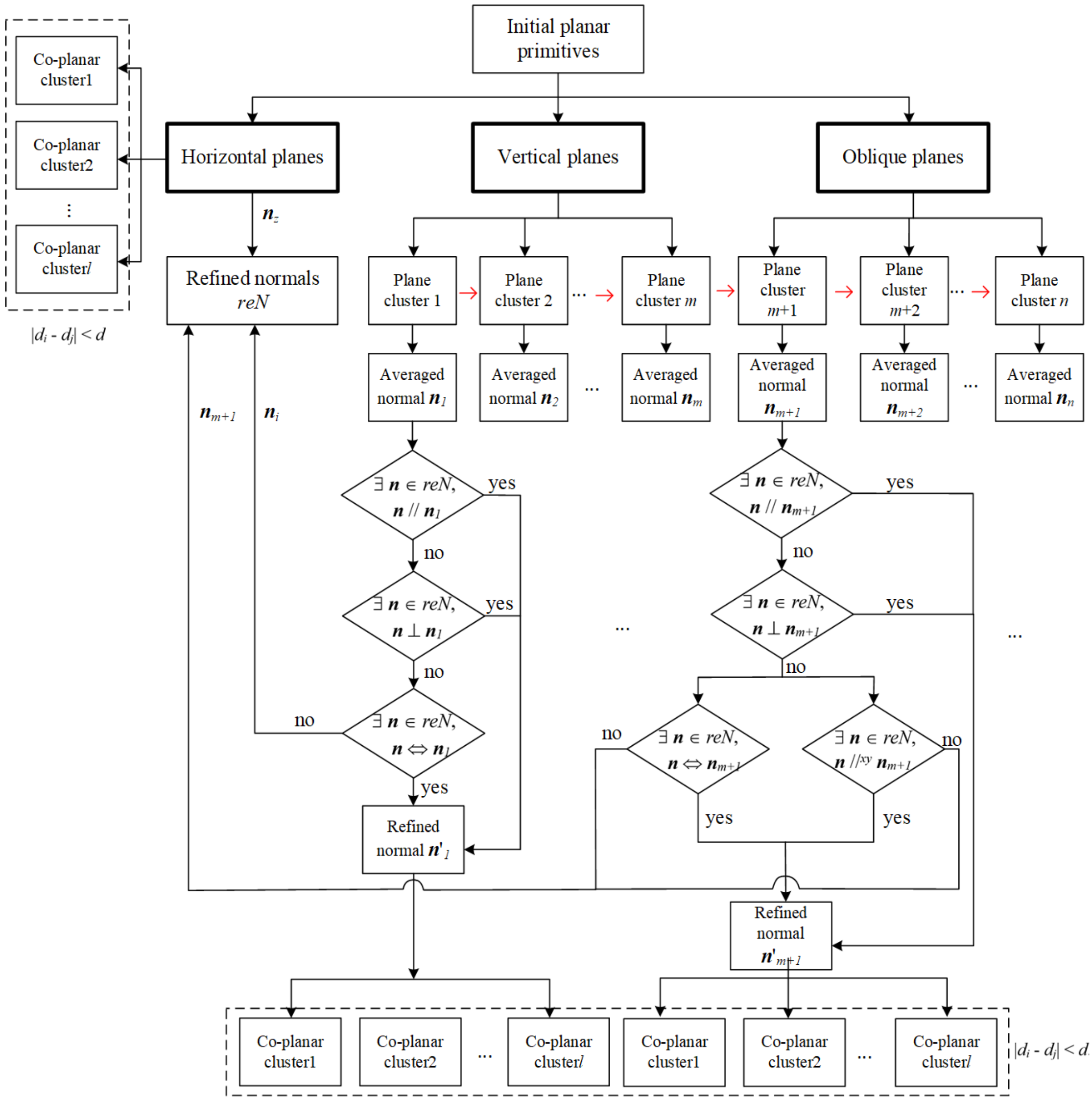



Figure 5.2 The flowchart of the refinement of planar primitives based on geometric relations.

Figure 5.3 shows the planar primitives extracted from the building point cloud and the comparison between the original normals and the refined normals of the plane clusters. It can be seen from the original normals of the plane clusters that, although they roughly meet the geometric relations, there are always small deviations between them. After the plane refinement, the geometric relations between them are stricter, which is consistent with the hypotheses about man-made objects.

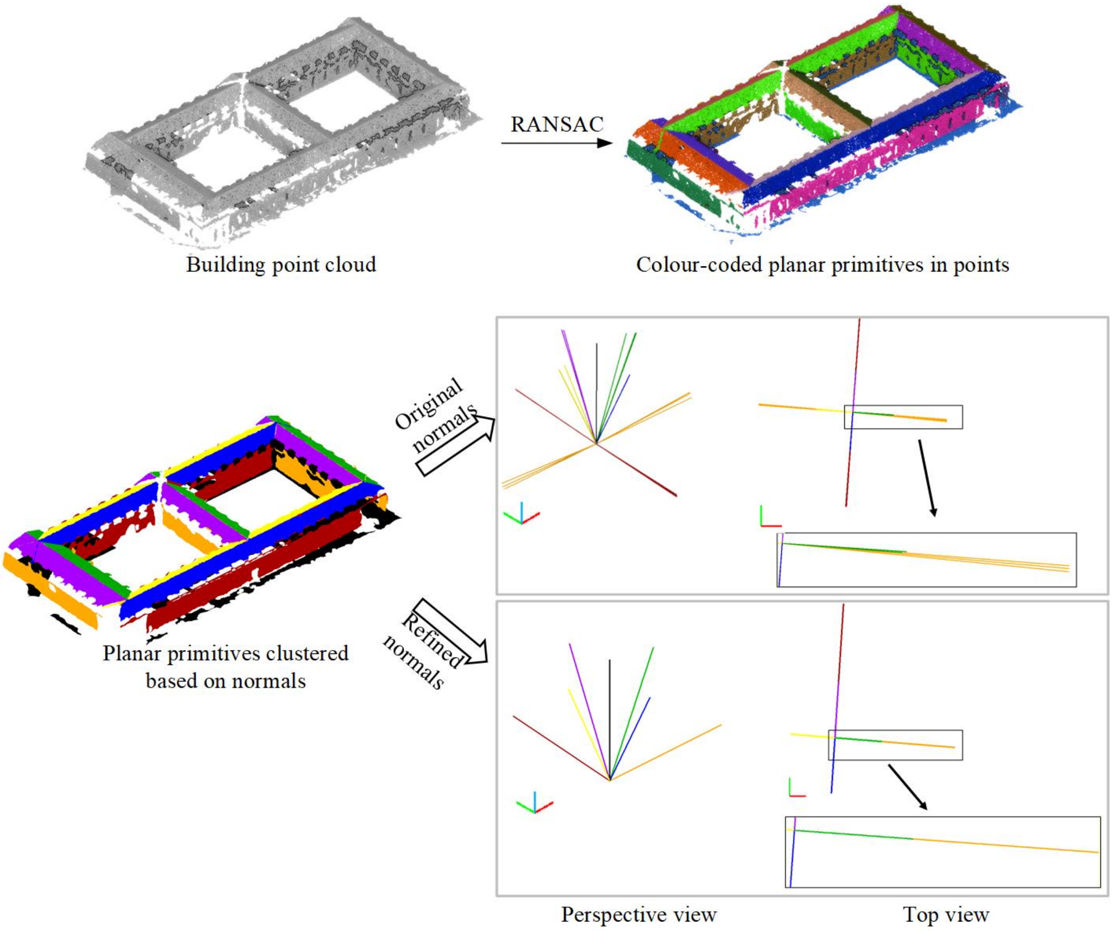



Figure 5.3 Planar primitives extracted from the building point cloud and the refinement of plane normals.

After the refinement, the points supporting each planar primitive are projected on the plane based on the refined normal, and the 2D alpha-shape (Liang et al., 1998) is used to extract its boundaries (including outer and inner boundaries) from the projected points. The planar primitives are finally presented in terms of the corresponding planar coefficients and their boundary points.

### 5.2.2 3D Arrangement Based on a Half Binary Space Partition Tree

After the extraction and refinement of the planar primitives, the bounding box of the building, which is expanded outward with a constant value, e.g., one meter, is then partitioned into a set of convex cells with 3D Boolean operations. By each partition, the parent cells are split into zero or two children cells, which can be recorded as a

BSP tree (as shown in Figure 5.4 and Figure 5.5). In the studies of Verdie et al. (2015) and Nan and Wonka (2017), the entire 3D space was partitioned with each individual plane, as shown in Figure 5.4, which is called a full BSP tree in this research. Obviously, this full BSP tree will result in a redundant cell complex and lead to a large computational cost in the propagation of the partitioning and later processing. Therefore, instead of full partition, this research uses a half BSP tree, which only partitions the parent cells occupying the planar primitive during each partition, as shown in Figure 5.5.

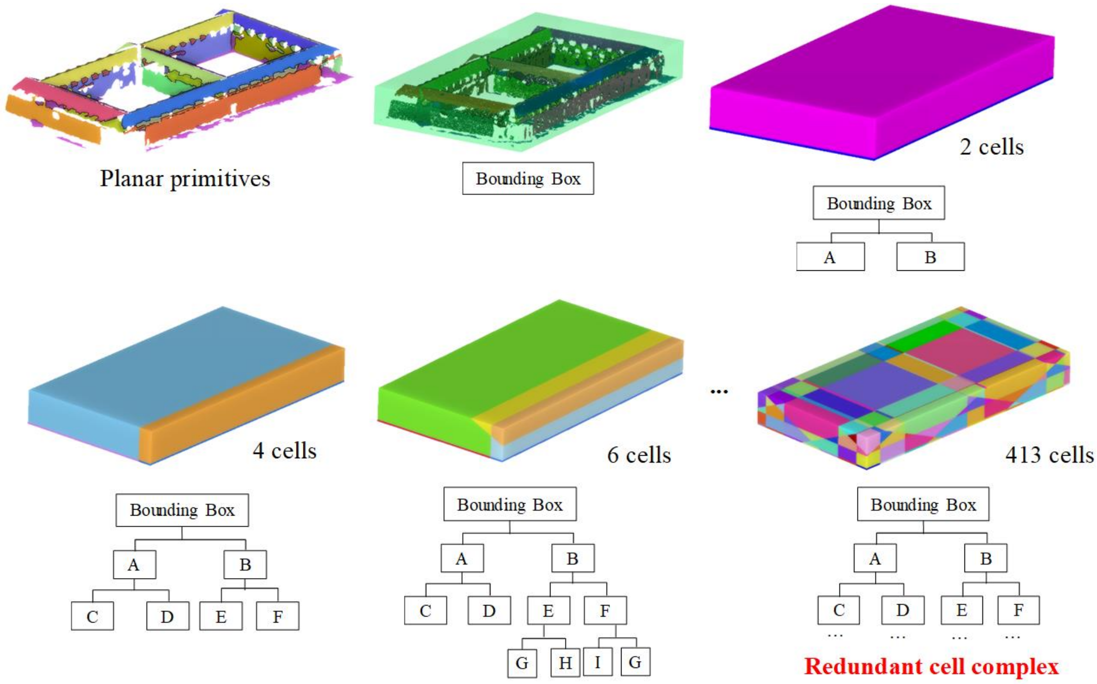



Figure 5.4 Space partition of the building bounding box with planar primitives based on the full BSP tree.

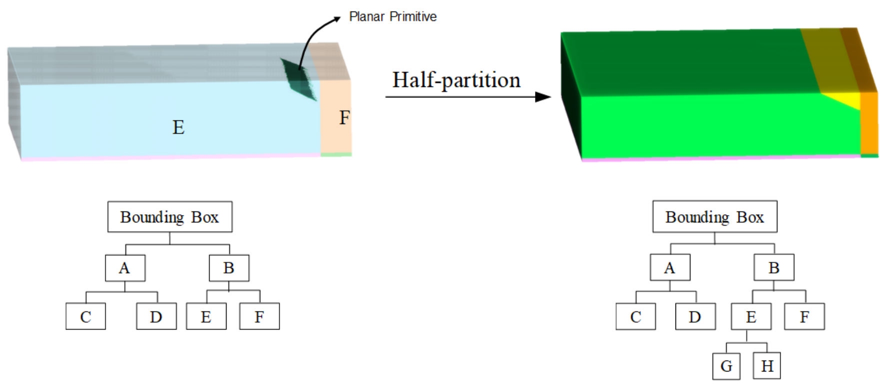



Figure 5.5 Space partition with planar primitive based on the half BSP tree.

Because the half BSP tree only partitions parent cells containing the planar primitives, the final cell complex is related to the partitioning order of the planar primitives. Thus, a partitioning order as shown in Figure 5.6 is defined in this research. First, vertical planar primitives are designed to have higher priority than horizontal or oblique ones to avoid incomplete partitioning caused by the missing data on the building façades, which is a common problem in point cloud data. Second, in the same priority class, planar primitives with larger areas are considered to have higher priority than smaller ones to make the size of the final cell complex as small as possible.

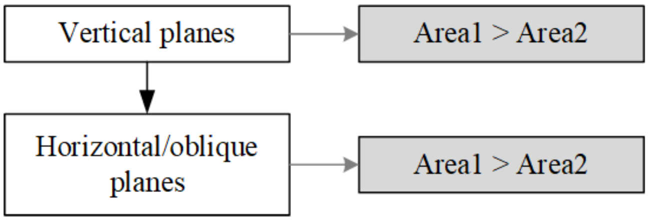



Figure 5.6 Partitioning orders of the planar primitives.

To further avoid incomplete partitioning caused by missing data for building roof structures, the planar primitives are all expanded with a constant value $\xi$ (e.g., $\xi = 3$ m) during the partitioning, as shown in Figure 5.7 (a) and (b). Although the extension of the planar primitives increases the size of the final cell complex, the half partition strategy still makes it considerably smaller than the result based on a full BSP tree (as shown in Figure 5.7 (c) and (d)).

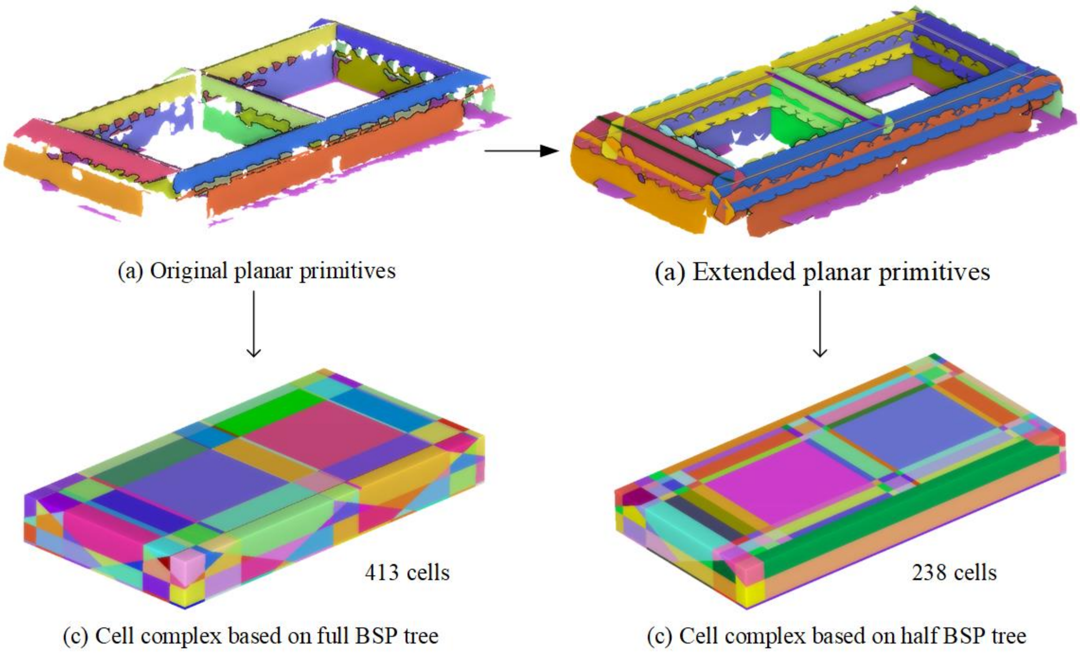



Figure 5.7 Extension of planar primitives and comparison of the final cell complexes based on full and half BSP trees.

The surfaces of small building structures are prone to being incorrectly presented by point cloud data because of occlusions, sparse point densities and the limitations of multi-view stereo (MVS) pipelines. For the DIM point clouds, another problem is that the sharp corners are always presented as smooth conversions, which can be incorrectly recognised as planar primitives with thin and long shapes. To exclude these invalid planar primitives, two thresholds $\tau$ and $\upsilon$ (e.g., $\tau$ = 2 m$^2$ and $\upsilon$ = 0.2) for the area and shape factor of the primitives are set. The shape factor is defined based on the area and perimeter (including both inner and outer boundaries) of a planar primitive with the following format

$$Shape\ factor = \frac{4\pi \cdot Area(P)}{Perimeter(P)^2} \tag{5-1}$$

where $P \in \mathbb{P}$ is a planar primitive and *Area*($P$) and *Perimeter*($P$) denote the area and perimeter of $P$, respectively. Only primitives with either areas or shape factors greater than the corresponding threshold are considered valid for the 3D space partition. Finally, the 3D arrangement is based on the valid planar primitives and the final cell complex is noted as $\mathbb{C}$.

## 5.3 Optimal Selection of Occupied Cells Constrained by Topological Relations

As the cell complex $\mathbb{C}$ is generated based on the planar primitives presenting the building surfaces, it can be assumed that there is no cell $C \in \mathbb{C}$ bestriding the building surfaces, which means that each $C \in \mathbb{C}$ is either occupied by the building (inside the building) or is empty (outside the building). Therefore, the geometric reconstruction of the building surface can be considered as the selection of occupied cells, which can be further modelled by a binary labelling problem. In solving the labelling problem, the topological relations between the cells are introduced as constraints. The topological relations include the interfaces, which are extracted as a set of 3D facets $\mathbb{F}$, and the intersection lines, which are extracted as a set of 3D edges $\mathbb{E}$, between the cells. The 3D cells, facets and edges are used together to establish the global energy function as follows:

$$arg\ \min_{\boldsymbol{l}} E(\boldsymbol{l}) = \sum_{C \in \mathbb{C}} D_{Cell}(C, \boldsymbol{l}_C) + \gamma \sum_{F \in \mathbb{E}} D_{Facet}(F, \boldsymbol{l}_F) + \eta \sum_{E \in \mathbb{E}} D_{Edge}(E, \boldsymbol{l}_E) \quad (5\text{-}2)$$

where $\boldsymbol{l}$ is the binary labelling configuration that assigns each cell, facet and edge a label $\boldsymbol{l}_C$, $\boldsymbol{l}_F$ *and* $\boldsymbol{l}_E$, respectively ( $\boldsymbol{l}_C, \boldsymbol{l}_F, \boldsymbol{l}_E \in \{0, 1\}$; 1 denotes that the cell/facet/edge is selected, and 0 denotes that it is not); $\gamma$ and $\eta$ are two parameters that control the weights of the topological constraints.

The optimal selection of occupied cells involves finding a labelling $\boldsymbol{l}$ that minimises the global energy $E(\boldsymbol{l})$, and $\boldsymbol{l}_C$, $\boldsymbol{l}_F$ and $\boldsymbol{l}_E$ meet the constraints on the topological relations between the corresponding elements. Thus, the binary labelling is converted into an ILP problem, where Equation (5-2) is the objective function and the topological relations between the elements formulate the constraints of the ILP problem.

### 5.3.1 Extraction of Topological Relations as 3D Facets and Edges

A. Extraction of 3D Facets

The 3D facets refer to the interfaces between the 3D cells (as shown in Figure 5.8). As the 3D cells are generated by partitioning the 3D space with planar primitives, it can be imagined that each 3D facet is connected to at most two cells and at least one cell (only facets on the surfaces of the building bounding box are connected to single cells). A facet that is connected to a pair of cells $C_i$ and $C_j$ ($C_i$, $C_j \in \mathbb{C}$) is denoted as $F_{ij}$ and that connected to a single cell $C_k$ ($C_k \in \mathbb{C}$) is denoted as $F_k$.

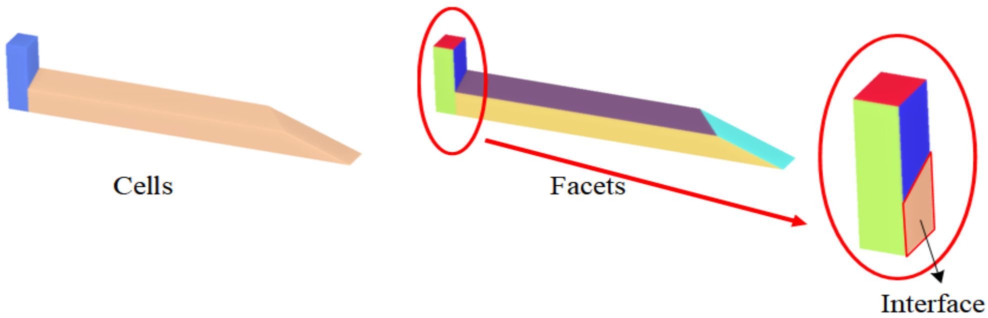
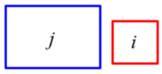
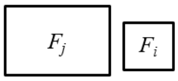
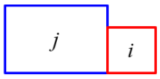
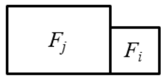
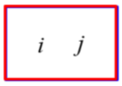
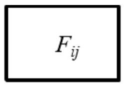
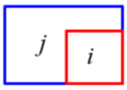
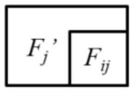
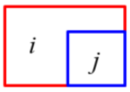
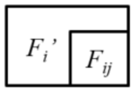
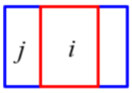
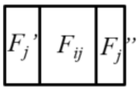
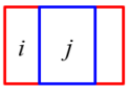
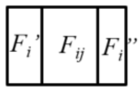
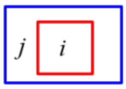
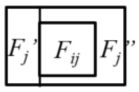
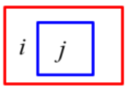
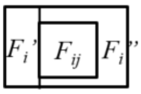
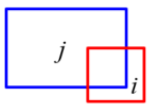
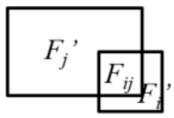



Figure 5.8 The interface between two cells.

Take the set of facets connected with pairs of cells as $\mathbb{F}^2$ and the set of cells connected with single cells as $\mathbb{F}^1$. To extract these two types of facets, an iterative extraction and update process is designed based on 2D topological relations as follows. For a cell $C_i \in \mathbb{C}$, the 3D faces constituting $C_i$ are directly obtained from its geometric data structure. For each face $F_i$ obtained from $C_i$, if there exists a facet $F_j$ ($F_j \in \mathbb{F}^1$ corresponds to cell $C_j$) that is co-planar with $F_i$, $F_i$ and $F_j$ are rotated onto their co-plane to generate two 2D polygons $Poly_i$ and $Poly_j$. Based on the topological relation between $Poly_i$ and $Poly_j$, a set of new 2D polygons is generated with 2D Boolean operations. These new polygons are then re-rotated to the 3D space and output as new 3D facets. A set of operations is defined as shown in Table 5.2 (where the red polygon denotes $Poly_i$ and the blue one denotes $Poly_j$) to determine whether the new 3D facets are added to $\mathbb{F}^1$ or to $\mathbb{F}^2$.

Table 5.2 2D topological relations between facet projections and the corresponding outputs and operations.

| ***Topological relation*** | | ***Output*** | ***Operations*** |
|---|---|---|---|
|  | *Separated* |  | Add $F_i$ to $\mathbb{F}^1$. |
|  | *Connecting* |  | Add $F_i$ to $\mathbb{F}^1$. |
|  | *Full-overlapping* |  | Remove $F_j$ from $\mathbb{F}^1$; add $F_{ij}$ to $\mathbb{F}^2$. |
|  | *Part-overlapping1* |  | Remove $F_j$ from $\mathbb{F}^1$; add $F_{ij}$ to $\mathbb{F}^2$; add $F_j'$ to $\mathbb{F}^1$. |
|  | *Part-overlapped1* |  | Remove $F_j$ from $\mathbb{F}^1$; add $F_{ij}$ to $\mathbb{F}^2$; add $F_i'$ to $\mathbb{F}^1$. |
|  | *Part-overlapping2* |  | Remove $F_j$ from $\mathbb{F}^1$; add $F_{ij}$ to $\mathbb{F}^2$; add $F_j'$, $F_j''$ to $\mathbb{F}^1$. |
|  | *Part-overlapped2* |  | Remove $F_j$ from $\mathbb{F}^1$; add $F_{ij}$ to $\mathbb{F}^2$; add $F_i'$, $F_i''$ to $\mathbb{F}^1$. |
|  | *Contained* |  | Remove $F_j$ from $\mathbb{F}^1$; add $F_{ij}$ to $\mathbb{F}^2$; add $F_j'$, $F_j''$ to $\mathbb{F}^1$. |
|  | *Containing* |  | Remove $F_j$ from $\mathbb{F}^1$; add $F_{ij}$ to $\mathbb{F}^2$; add $F_i'$, $F_i''$ to $\mathbb{F}^1$. |
|  | *Intersecting* |  | Remove $F_j$ from $\mathbb{F}^1$; add $F_{ij}$ to $\mathbb{F}^2$; add $F_i'$ to $\mathbb{F}^1$; and $F_j'$ to $\mathbb{F}^1$. |

This iteration continues until all of the cells in $\mathbb{C}$ have been checked. Finally, the 3D facet set $\mathbb{F}$ is obtained by merging $\mathbb{F}^1$ and $\mathbb{F}^2$. The pseudo-code for the extraction of 3D facets is shown in Table 5.3. As each facet in $\mathbb{F}^2$ records the indexes of two connected cells, the adjacency relationships between the 3D cells are therefore also obtained.

Table 5.3 Pseudo-code for the extraction of 3D facets from 3D cell complex.

**Algorithm:** Extaction of 3D facets from 3D cell complex

**Input:** Cell complex $\mathbb{C}$

**Output:** Facet complex $\mathbb{F}$

*Note: $F_{ij(k)}$ refers to the $k^{th}$ element in the 3D facet set $\mathbb{F}$, and ij means this facet is connected with the cells $C_i$ and $C_j$ ($C_i$, $C_j \in \mathbb{C}$).*

Initialize: $\mathbb{F} = \varnothing$;<br>
        Set of facets connected with single cells $\mathbb{F}^1 = \varnothing$;<br>
        Set of facets connected with pairs of cells $\mathbb{F}^2 = \varnothing$;<br>
**for** $i$: $C_i \in \mathbb{C}$<br>
    Get cell faces $\mathbb{F}_i$ = GetCellFaces($C_i$).<br>
    **for** $k$: $F_{i(k)} \in \mathbb{F}_i$<br>
        **for** $n$: $F_{(n)} \in \mathbb{F}^1$<br>
            Cell index $j$ = GetCellIndexofFacet($F_{(n)}$);<br>
            **if** $F_{i(k)}$ is co-planar with $F_{(n)}$<br>
                Plane normal $\boldsymbol{n}$ = GetFacetNormal($F_{(n)}$);<br>
                Rotation matrix $R$ = RotateZ-axisTo($\boldsymbol{n}$);<br>
                Polygon $Poly_i$ = Rotate3DFacet($F_{i(k)}$, $R$);<br>
                Polygon $Poly_j$ = Rotate3DFacet($F_{(n)}$, $R$);<br>
                2D topological relation $re$ = 2DTopologicalRelation($Poly_i$, $Poly_j$);<br>
                **if** *re* == *Seperated* or *re* == *Connecting*<br>
                    Push $F_{i(k)}$ to $\mathbb{F}^1$: $\mathbb{F}^1 \leftarrow F_{i(k)}$;<br>
                **else**<br>
                    New polygons *newPolys* = GenerateNewPolygons($Poly_i$, $Poly_j$);<br>
                    *// The newPolys is a vector, where the first element always refers to*<br>
                    *// the overlapping part of $Poly_i$, and $Poly_j$. Elements corresponding to*<br>
                    *// $Poly_i$ rank in front of those corresponding to $Poly_j$.*<br>
                    New 3D facets *newFacets* = Rotate3DFacets(*newPolys*, $R^{-1}$);<br>
                    Erase $F_{(n)}$ from $\mathbb{F}^1$;<br>
                    Note $newPolys_{(0)}$ as $F_{ij}$; push $F_{ij}$ to $\mathbb{F}^2$: $\mathbb{F}^2 \leftarrow F_{ij}$;<br>
                    **In case:** *re* == *Full-overlapping*<br>
                        Continue;<br>
                    **In case:** *re* == *Part-overlapping1*<br>
                        Note $newpolys_{(1)}$ as $F_j$; push $F_j$ to $\mathbb{F}^1$: $\mathbb{F}^1 \leftarrow F_j$;<br>
                    **In case:** *re* == *Part-overlapped1*<br>
                        Note $newpolys_{(1)}$ as $F_i$; push $F_i$ to $\mathbb{F}^1$: $\mathbb{F}^1 \leftarrow F_i$;

**In case:** *re* == *Part-overlapping2* or *re* == *Contained*
Note *newpolys*$_{(1)}$ as $F_j$; push $F_j$ to $\mathbb{F}^1$: $\mathbb{F}^1 \leftarrow F_j$;
Note *newpolys*$_{(2)}$ as $F_j$; push $F_j$ to $\mathbb{F}^1$: $\mathbb{F}^1 \leftarrow F_j$;
**In case:** *re* == *Part-overlapped2* or *re* == *Containing*
Note *newpolys*$_{(1)}$ as $F_i$; push $F_i$ to $\mathbb{F}^1$: $\mathbb{F}^1 \leftarrow F_i$;
Note *newpolys*$_{(2)}$ as $F_i$; push $F_i$ to $\mathbb{F}^1$: $\mathbb{F}^1 \leftarrow F_i$;
**In case:** *re* == Intersecting
Note *newpolys*$_{(1)}$ as $F_i$; push $F_i$ to $\mathbb{F}^1$: $\mathbb{F}^1 \leftarrow F_i$;
Note *newpolys*$_{(2)}$ as $F_j$; push $F_j$ to $\mathbb{F}^1$: $\mathbb{F}^1 \leftarrow F_j$;
**end if**
**end if**
**end for**
**end for**
**end for**
$\mathbb{F} = \mathbb{F}^2$;
Insert facets in $\mathbb{F}^1$ to $\mathbb{F}$: $\mathbb{F} \Leftarrow \mathbb{F}^1$;
**returen** $\mathbb{F}$

B. Extraction of 3D Edges

With the 3D facets, the 3D edges are then extracted from the intersecting lines between the facets. Theoretically, one intersecting line can correspond to multiple facets as shown in Figure 5.9 (a). In this research, the 3D edges are only defined based on the intersecting line segments between pairs of facets as shown in Figure 5.9 (b), for the following two reasons. First, it is easy to handle the misalignment between facets in a pairwise situation. Second, because the surface of the final polygonal model must be formed by pairwise-connected facets, constraints on the model surface can be easily encoded into the pairwise edges.

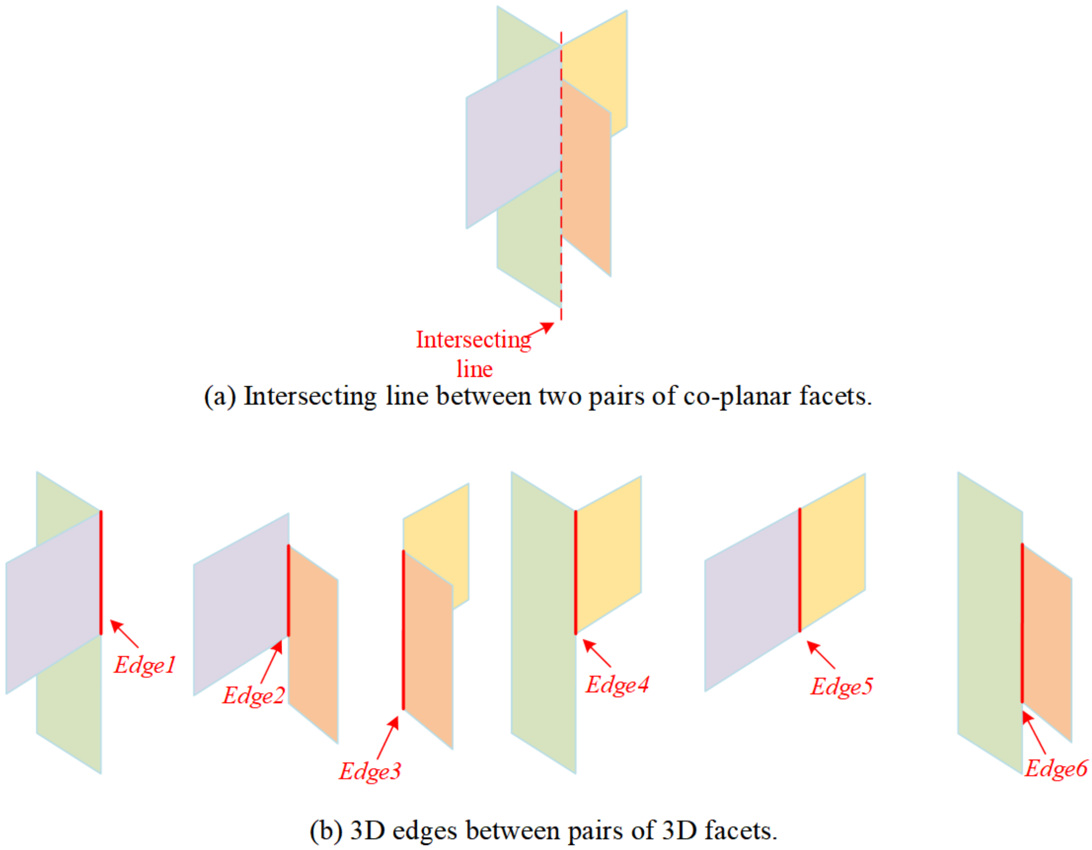
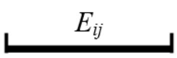
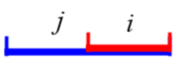
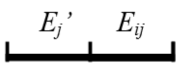
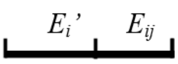
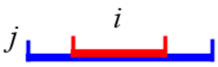
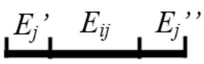
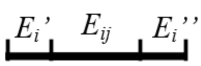
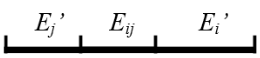



Figure 5.9 Intersecting line and 3D edges between 3D facets.

The 3D edges between the pairs of facets are extracted with a similar strategy to that for the extraction of facets. First, two edge sets $\mathbb{E}^1$ and $\mathbb{E}^2$ are initialised to store the edges connected by individual facets and pairs of facets. Note that in no situation is an edge in the complex connected to a single facet. Thus, $\mathbb{E}^1$ is only a temporary set that stores the candidate edges. For a facet $F_i \in \mathbb{F}$, the edges constituting the boundary of $F_i$ are obtained from its geometric data structure. For each boundary edge $E_i$, if there is an edge $E_j \in \mathbb{E}^1$ ($j$ refers to the index of the facet, to which the edge is connected) that is co-linear with $E_i$, the topological relation between $E_i$ and $E_j$ is computed. Based on their topological relation, new edges are generated and are further added into $\mathbb{E}^1$ or $\mathbb{E}^2$ as described in Table 5.4, where the red line segment refers to $E_i$ and the blue one refers to $E_j$. Note that, for an edge $E_{ij} \in \mathbb{E}^2$, $i$ and $j$ refer to the indices of the two connected facets.

Table 5.4 1D topological relations between the co-linear 3D edges and the corresponding outputs and operations.

| ***Topological relation*** | | ***Output*** | ***Operations*** |
|---|---|---|---|
|   | *Separated* |   | Add $E_i$ to $\mathbb{E}^1$. |
|   | *Connecting* |   | Add $E_i$ to $\mathbb{E}^1$. |
|   | *Full-overlapping* |   | Remove $E_j$ from $\mathbb{E}^1$; add $E_{ij}$ to $\mathbb{E}^2$. |
|   | *Part-overlapping* |   | Remove $E_j$ from $\mathbb{E}^1$; add $E_{ij}$ to $\mathbb{E}^2$; add $E_j$' to $\mathbb{E}^1$. |
|   | *Part-overlapped1* |   | Remove $E_j$ from $\mathbb{E}^1$; add $E_{ij}$ to $\mathbb{E}^2$; add $E_i$' to $\mathbb{E}^1$. |
|   | *Contained* |   | Remove $E_j$ from $\mathbb{E}^1$; add $E_{ij}$ to $\mathbb{E}^2$; add $E_j$', $E_j$'' to $\mathbb{E}^1$. |
|   | *Containing* |   | Remove $E_j$ from $\mathbb{E}^1$; add $E_{ij}$ to $\mathbb{E}^2$; add $E_i$', $E_i$'' to $\mathbb{E}^1$. |
|   | *Intersecting* |   | Remove $E_j$ from $\mathbb{E}^1$; add $E_{ij}$ to $\mathbb{E}^2$; add $E_i$' to $\mathbb{E}^1$; and $E_j$' to $\mathbb{E}^1$. |

Because only facets that belong to the same cell or adjacent cells can produce edges, the owner–member relationships between the cells and facets and the adjacency relationships between corresponding cells are considered during the iteration to speed up the extraction. The pseudo-code for the extraction of 3D edges is shown in Table 5.5. Figure 5.10 shows the extracted 3D facet and edge complexes of two example buildings.

Table 5.5 Pseudo-code for the extraction of pairwise 3D edges from 3D facet complex.

**Algorithm:** Extaction of 3D pairwise edges from 3D facet complex

**Input:** Facet complex $\mathbb{F}$; Cell complex $\mathbb{C}$.

**Output:** Edge complex $\mathbb{E}$

*Note: $E_{ij(k)}$ refers to the $k^{th}$ element in the 3D edge set $\mathbb{E}$, and ij means this edge is connected to the $i^{th}$ and $j^{th}$ facets $F_i$, $F_j$ ($F_i$, $F_j \in \mathbb{F}$).*

Initialize: $\mathbb{E} = \varnothing$;<br>
    Set of facets connected with single cells $\mathbb{E}^1 = \varnothing$;<br>
    Set of facets connected with pairs of cells $\mathbb{E}^2 = \varnothing$;<br>
**for** $i$: $F_i \in \mathbb{F}$<br>
  Boundary edges $\mathbb{E}_i$ = GetBoundaryEdges($F_i$);<br>
  Pair of cells $\{C_{i1}, C_{i2}\}$ = GetConnectedCells($F_i$);<br>
  **for** $k$: $E_{i(k)} \in \mathbb{E}_i$<br>
    **for** $n$: $E_{(n)} \in \mathbb{E}^1$<br>
      Facet index $j$ = GetFacetIndexofEdge($E_{(n)}$);<br>
      Pair of cells $\{C_{j1}, C_{j2}\}$ = GetConnectedCells($F_j$);<br>
      **if** $C_{im} == C_{jn}$ or $C_{im}$ and $C_{jn}$ are neighbors ($m, n \in \{1,2\}$)<br>
        **if** $E_{i(k)}$ is co-linear with $E_{(n)}$<br>
          Line direction ***dir*** = GetEdgeDirection($E_{(n)}$);<br>
          Rotation matrix $R$ = RotateX-axisTo(***dir***);<br>
          Rotated edge $E'_i$ = Rotate3DEdge($E_{i(k)}$, $R$);<br>
          Rotated edge $E'_j$ = Rotate3DEdge($E_{(n)}$, $R$);<br>
          1D topological relation *re* = 1DTopologicalRelation($E'_i$, $E'_j$);<br>
          **if** *re* == *Seperated* or *re* == *Connecting*<br>
            Push $E_{i(k)}$ to $\mathbb{E}^1$: $\mathbb{E}^1 \leftarrow E_{i(k)}$;<br>
          **else**<br>
            New edges *newEdges* = GenerateNewEdges($E'_i$, $E'_j$);<br>
            // *The newEdges is a vector, where the first element always refers*<br>
            // to *the overlapping part of $E'_i$,and $E'_j$. Elements corresponding*<br>
            // to *$E'_i$ rank in front of those corresponding to $E'_j$.*<br>
            New 3D edges *new3DEdges* = Rotate3DEdges(*newEdges*, $R^{-1}$);<br>
            Erase $E_{(n)}$ from $\mathbb{E}^1$;<br>
            Note *new3DEdges* $_{(0)}$ as $E_{ij}$; push $E_{ij}$ to $\mathbb{E}^2$: $\mathbb{E}^2 \leftarrow E_{ij}$;<br>
            **In case:** *re* == *Full-overlapping*<br>
              Continue;

**In case:** $re$ == *Part-overlapping*
Note $new3DEdges_{(1)}$ as $E_j$; push $E_j$ to $\mathbb{E}^1$: $\mathbb{E}^1 \leftarrow E_j$;
**In case:** $re$ == *Part-overlapped*
Note $new3DEdges_{(1)}$ as $E_i$; push $E_i$ to $\mathbb{E}^1$: $\mathbb{E}^1 \leftarrow E_i$;
**In case:** $re$ == *Contained*
Note $new3DEdges_{(1)}$ as $E_j$; push $E_j$ to $\mathbb{E}^1$: $\mathbb{E}^1 \leftarrow E_j$;
Note $new3DEdges_{(2)}$ as $E_j$; push $E_j$ to $\mathbb{E}^1$: $\mathbb{E}^1 \leftarrow E_j$;
**In case:** $re$ == *Containing*
Note $new3DEdges_{(1)}$ as $E_i$; push $E_i$ to $\mathbb{E}^1$: $\mathbb{E}^1 \leftarrow E_i$;
Note $new3DEdges_{(2)}$ as $E_i$; push $E_i$ to $\mathbb{E}^1$: $\mathbb{E}^1 \leftarrow E_i$;
**In case:** $re$ == Intersecting
Note $new3DEdges_{(1)}$ as $E_i$; push $E_i$ to $\mathbb{E}^1$: $\mathbb{E}^1 \leftarrow E_i$;
Note $new3DEdges_{(2)}$ as $E_j$; push $E_j$ to $\mathbb{E}^1$: $\mathbb{E}^1 \leftarrow E_j$;
**end if**
**end if**
**else**
Continue;
**end if**
**end for**
**end for**
**end for**
$\mathbb{E} = \mathbb{E}^2$;
**returen** $\mathbb{E}$

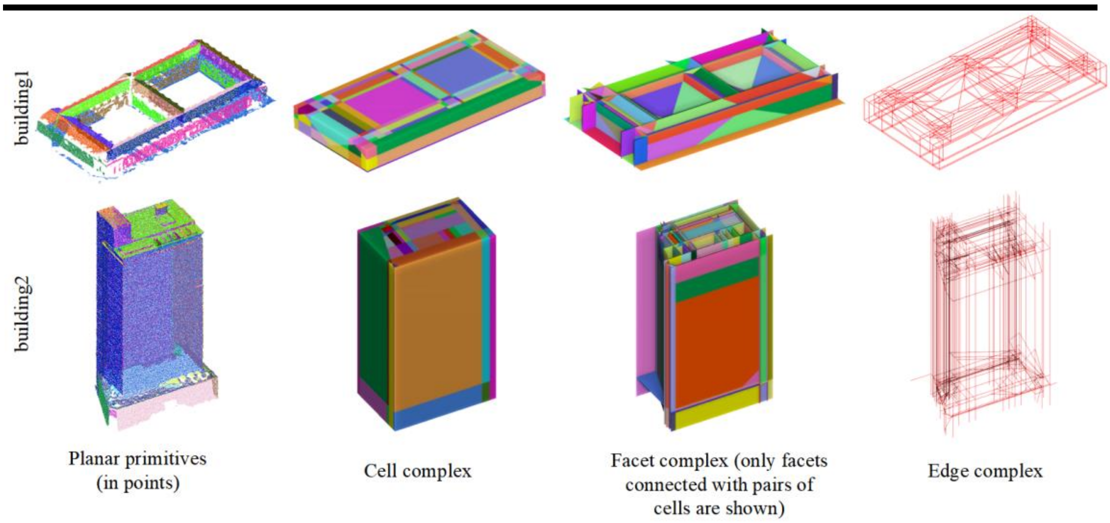



Figure 5.10 3D facet and edge complexes extracted from the cell complexes of two example buildings.

### 5.3.2 Topological-Relation Constrained Selection of Cells

A. Cell Energies

In this research, the cells inside the building are defined as occupied, and those outside the building are defined as empty. As the cell complex $\mathbb{C}$ is generated by partitioning the 3D space based on planar primitives, all of the cells in $\mathbb{C}$ are supposed to be convex and not to bestride the building surface. Therefore, whether a cell $C \in \mathbb{C}$ is occupied or empty can be determined by checking whether its centroid is inside or outside the building. Figure 5.11 is a 2D diagrammatic sketch of the determination of occupied and empty cells, based on the number of intersection points that the rays starting from the cell centroids have with the building surfaces. In theory, for a point inside the building, a ray starting from this point and extending in any upper or horizontal direction must have an odd number of intersection points with the building surface (as shown in Figure 5.11 (b)). For a point outside the building, the number of intersection points will be even (as shown in Figure 5.11 (c)).

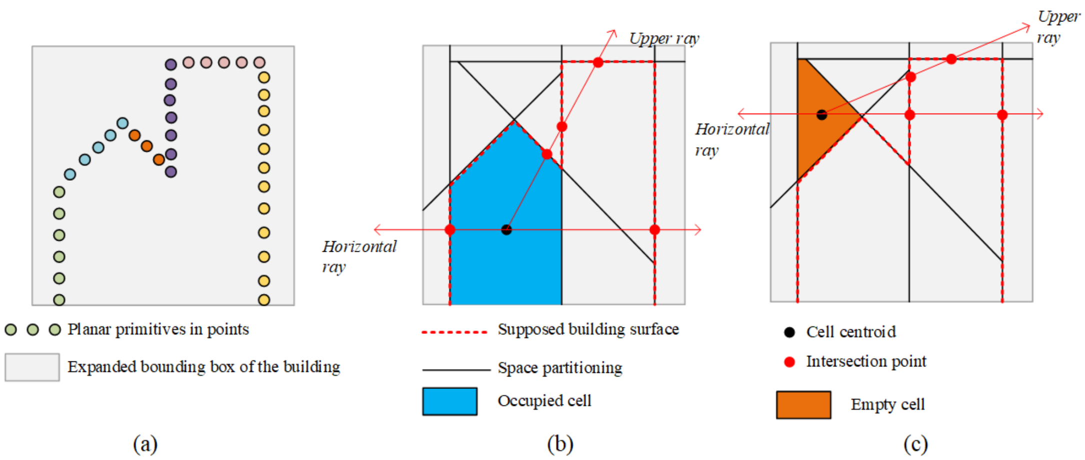



Figure 5.11 Occupied and empty cells illustrated in 2D.

As the exact building surface is not available at this point, in this research, the planar primitives with boundaries (see Section 5.2.1) are used to roughly present the building surface. Because there can be holes and crack effects between the planar primitives because of the missing data, using a ray in a single direction to determine whether the cells are occupied or empty might lead to inaccuracies. Therefore, for each cell $C \in \mathbb{C}$, a number of rays starting from its centroid are generated with an angle interval of δ (e.g., δ = 30°) to determine whether its centroid is inside or outside the building. The probability of $C$ being occupied can be formulated as

$$p_C = \frac{N_{odd}(rays)}{N(rays)} \tag{5-3}$$

where $N(rays)$ is the total number of rays drawn from $C$'s centroid, and $N_{ood}(rays)$ is the number of rays that have odd numbers of intersection points with the planar primitives.

Cells with high probabilities of being occupied should be selected to form the polygonal building model. Therefore, the cell energy in Equation (5-2) is given as

$$D_{Cell}(C, \boldsymbol{l}_C) = \frac{|\boldsymbol{l}_C - p_C|}{N(\mathbb{C})} \tag{5-4}$$

where $N(\mathbb{C})$ is the number of cells in $\mathbb{C}$, $\boldsymbol{l}_C \in \{0, 1\}$ is the binary label assigned to $C$ and $\boldsymbol{l}_C = 1$ denotes that $C$ is occupied and 0 denotes that it is empty.

B. Facet Energies

If a 3D facet $F \in \mathbb{F}$ is supported by points in the building point cloud, it is defined as occupied, otherwise, it is empty, as shown in Figure 5.12 (b) and (c). The points with perpendicular distances to facet $F$ smaller than $d$ ($d$ is the same as the distance threshold used to determine the co-planar relation; see Section 5.2.1) are regarded as supporting points of $F$.

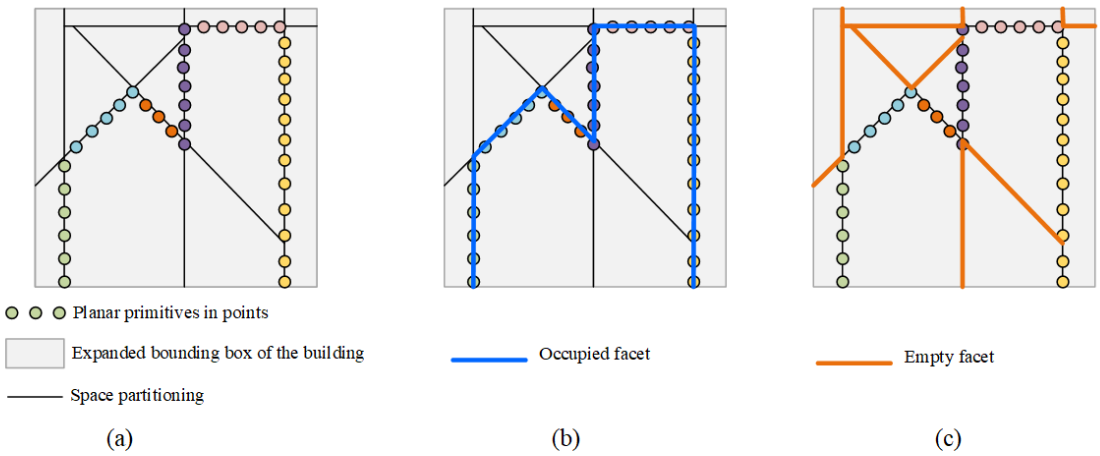



Figure 5.12 Occupied and empty facets illustrated in 2D.

With the supporting points of facet $F$, the coverage ratio is calculated as described in Nan and Wonka (2017). This coverage ratio is defined as the probability of $F$ being occupied and has the following format:

$$p_F = \frac{Area(Polygon(\mathcal{P}_F))}{Area(F)} \tag{5-5}$$

where $\mathcal{P}_F$ is the set of supporting points of $F$ and $Polygon(\mathcal{P}_F)$ is the polygon extracted from $\mathcal{P}_F$ by alpha-shape.

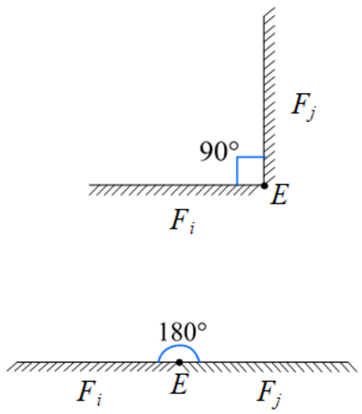
Facets with high probabilities are supposed to be visible in the final polygonal model of the building. The facet energy in Equation (5-2) can be formulated as

$$D_{Facet}(F, \boldsymbol{l}_F) = \frac{|\boldsymbol{l}_F - p_F|}{N(\mathbb{F})} \tag{5-6}$$

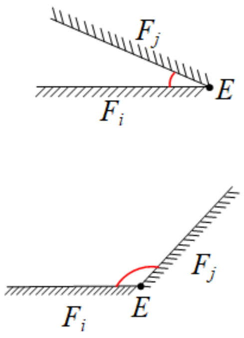
where $N(\mathbb{F})$ is the number of facets in $\mathbb{F}$, $\boldsymbol{l}_F \in \{0, 1\}$ is the binary label assigned to $F$ and $\boldsymbol{l}_F = 1$ denotes that $F$ is occupied and 0 denotes that it is empty.

C. Edge Energies

Unlike the cell and facet energies, which measure the fidelities of the modelled surfaces to the point clouds, the edge energies are used to introduce regularity constraints on the buildings into the optimal selection. Because man-made objects are generally believed to conform to planarity and orthogonality, in this research, the flat and right angles between pairs of facets are favoured, while other angles are penalised as shown in Figure 5.13.




(a) Favoured edge angles between facets.




(b) Penalised edge angles between facets.

Figure 5.13 Different types of edge angles between pairs of facets.

For each edge $E \in \mathbb{E}$, the edge angle between the corresponding connected facets is computed, and a regularity value of 0 is set to $E$ if the angle is 0° or 180°, and 1 otherwise. The edge energy in Equation (5-2) is thus given as Equation (5-7).

$$D_{Edge}(E, \boldsymbol{l}_E) = \frac{\boldsymbol{l}_E \cdot A(E)}{N(\mathbb{E})} \tag{5-7}$$

where $N(\mathbb{E})$ denotes the number of edges in $\mathbb{E}$ and $A(E)$ is the regularity value set to an edge $E \in \mathbb{E}$.

D. Energy Optimisation

With the cell, facet and edge energies defined as above, the global energy in Equation (5-2) is presented as a sectional continuous function. To simplify the ILP problem, the continuous occupied probabilities of the cells and facets are binarilised into {0, 1} based on the two user-defined thresholds $\varepsilon_C$ and $\varepsilon_F$ (e.g., $\varepsilon_C = 0.5$ and $\varepsilon_F = 0.3$), as shown in Equations (5-8) and (5-9).

$$\begin{cases} p'_C = 1, & \text{if } p_C \geqslant \varepsilon_C \\ p'_C = 0, & \text{otherwise} \end{cases} \tag{5-8}$$

$$\begin{cases} p'_F = 1, & \text{if } p_F \geqslant \varepsilon_F \\ p'_F = 0, & \text{otherwise} \end{cases} \tag{5-9}$$

The global energy in Equation (5-2) can therefore be turned into the quadratic format as follows:

$$arg\ \min_{l} E(\boldsymbol{l}) = \sum_{C \in \mathbb{C}} \frac{(\boldsymbol{l}_C - p'_C)^2}{N(\mathbb{C})} + \gamma \sum_{F \in \mathbb{F}} \frac{(\boldsymbol{l}_F - p'_F)^2}{N(\mathbb{F})} + \eta \sum_{E \in \mathbb{E}} \frac{\boldsymbol{l}_E \cdot A(E)}{N(\mathbb{E})} \quad (5\text{-}10)$$

where $\gamma$ controls the weight of the facet energy. Because both the cell energy and the facet energy measure the fidelity of the modelling to the point clouds, $\gamma$ is set as 1 in this research so that they play equal roles during the energy optimization. $\eta$ controls the weight of the edge energy, and it exposes more regularity with a larger value. Theoretically, to make a balance between the fidelity and the regularity of the modelling, the value $\eta$ should be at least set as 2. But considering that only a small portion of the edges correspond to irregular angles (not right or flat angles), $\eta$ needs a larger value to modify the irregularities. In this research, $\eta$ is set as 5.

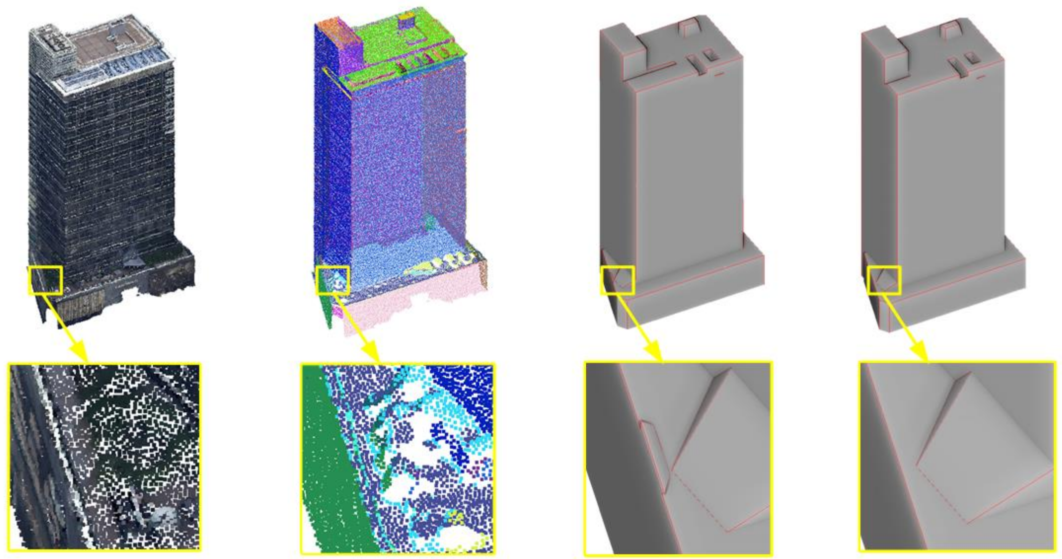
Simultaneously, because of the topological relationships between the cells, facets and edges, the binary variables $\boldsymbol{l}_C$, $\boldsymbol{l}_F$ and $\boldsymbol{l}_E$ ($C \in \mathbb{C}$, $F \in \mathbb{F}$ and $E \in \mathbb{E}$) also meet the following constraints

$$\begin{cases} \text{①}: \boldsymbol{l}_{F_j} = (\boldsymbol{l}_{C_{i1}} - \boldsymbol{l}_{C_{i2}})^2 \\ \text{②}: \boldsymbol{l}_{E_k} = \boldsymbol{l}_{F_{j1}} \cdot \boldsymbol{l}_{F_{j2}} \end{cases} \quad (5\text{-}11)$$

The first constraint means that a facet $F \in \mathbb{F}$ is visible ($\boldsymbol{l}_F = 1$) when only one of its connected cells is selected ($\boldsymbol{l}_C = 1$); otherwise, it is invisible ($\boldsymbol{l}_F = 0$). The second constraint means that an edge $E \in \mathbb{E}$ is valid ($\boldsymbol{l}_E = 1$) when both of its connected facets are visible ($\boldsymbol{l}_F = 1$); otherwise, it is invalid ($\boldsymbol{l}_E = 0$).

With the objective function in Equation (5-10) and the constraints in Equation (5-11), the ILP problem can be solved by the Gurobi solver (Gurobi, 2015). Finally, the building model is formed by the selected cells with visible facets and valid edges labelled as 1, as shown in Figure 5.14.

By comparing Figure 5.14 (c) with (d), it can be seen that small protrusions, which are caused by clutter or inaccuracies in the point clouds, can be removed with the edge regularity terms, making the modelling results more consistent with the hypotheses about the geometries of man-made objects.



(a) Original building point cloud.  (b) Planar primitives in colored points.  (c) Optimal selection result with $\eta = 0$.  (d) Optimal selection result with $\eta = 5$.

Figure 5.14 Geometric models of a building formed by occupied cells selected with different parameter values.

## 5.4 Generation of CityGML Models

### 5.4.1 Identification of CityGML Surface Types

To produce building models that can be used for urban applications, the geometric models generated through the above steps are further converted into formats that conform to CityGML (Gröger and Plümer, 2012). CityGML defines five building surface types: *GroundSurface*, *WallSurface*, *RoofSurface*, *OuterCeilingSurface*, and *OuterFloorSurface*, as shown in Figure 5.15.

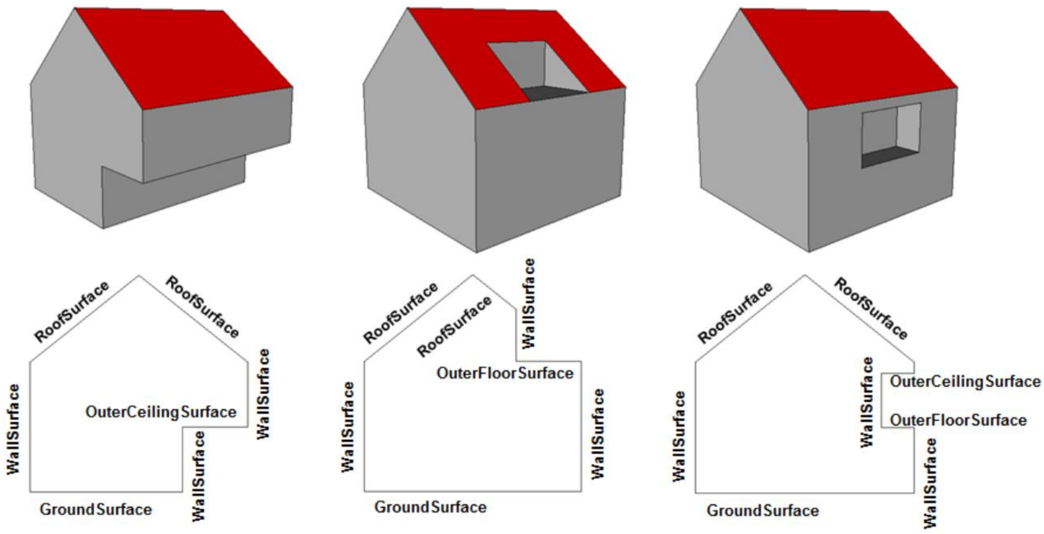



Figure 5.15 Building surface types defined by CityGML (Gröger and Plümer, 2012).

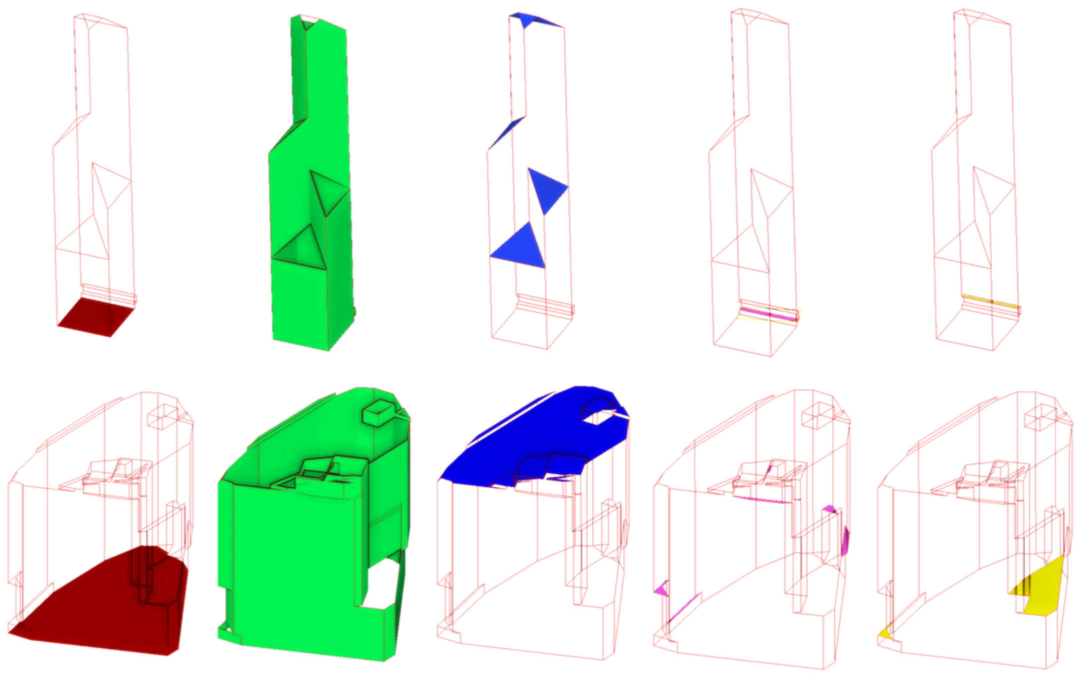
As shown in Figure 5.15, the surface normals are very important for the recognition of building surface types. The normal vector of each selected (labelled as 1) facet $F_{ij} \in \mathbb{F}$ is determined with the following two steps (note that if a facet $F_{ij}$ is labelled as 1, there must be only one cell $C_i$ or $C_j$ that is also labelled as 1 according to the first constraint in Equation (5-11)). First, an initial normal vector $\boldsymbol{n}'$ is computed based on three non-collinear vertexes of $F_{ij}$ chosen in anti-clockwise order. Second, the centroids of $F_{ij}$ and its connected cell $C_i$ or $C_j$, which is also labelled as 1, are joined to form a vector $\boldsymbol{d}$. If $\boldsymbol{n}' \cdot \boldsymbol{d} > 0$, $F_{ij}$'s final normal vector $\boldsymbol{n} = \boldsymbol{n}'$, and $\boldsymbol{n} = -\boldsymbol{n}'$ otherwise. Note that if $\boldsymbol{n} = -\boldsymbol{n}'$, the vertexes of $F_{ij}$ will be restored in the inverted order.

The facets with modified normals are then clustered into surface components based on their adjacent and co-planar relationships, which are already recorded in the edge complex $\mathbb{E}$ (refer to Section 5.4.2 for the specific clustering process). Each of the surface components of the building is assigned a CityGML surface type as mentioned above, based on the rules described in Table 5.6. Note that the angle and distance thresholds $\varepsilon$ and $d$ in Table 5.6 are the same as those in Table 5.1, $\overline{z}$ refers to the elevation of the centroid of the surface components and $z_{min}$ and $z_{max}$ denote the minimum and maximum elevation values of the buildings, respectively.

Table 5.6 Knowledge-based rules for the determination of the building surface type.

| ***Normal direction*** | | ***Elevation information*** | ***Surface type*** |
|---|---|---|---|
| $\lvert\theta(\boldsymbol{n}', \boldsymbol{n}_z)\rvert \geq \pi/2 - \varepsilon$ | | - | *WallSurface* |
| $\varepsilon < \lvert\theta(\boldsymbol{n}', \boldsymbol{n}_z)\rvert < \pi/2 - \varepsilon$ | $\boldsymbol{n}', \boldsymbol{n}_z > 0$ | - | *RoofSurface* |
| | $\boldsymbol{n}', \boldsymbol{n}_z < 0$ | - | *WallSurface* |
| $\lvert\theta(\boldsymbol{n}', \boldsymbol{n}_z)\rvert \leq \varepsilon$ | $\boldsymbol{n}', \boldsymbol{n}_z > 0$ | $\overline{z} < z_{min} + (z_{max} - z_{min})/3$ and $\overline{z} < 10$ m | *OuterFloorSurface* |
| | | *otherwise* | *RoofSurface* |
| | $\boldsymbol{n}', \boldsymbol{n}_z < 0$ | $\overline{z} < z_{min} + d$ | *GroundSurface* |
| | | *otherwise* | *OuterCeilingSurface* |

Figure 5.16 shows two example buildings with simple and complex structures, for which all five surface types (shown in different colours) defined by CityGML are recognised based on the rules.



*GroundSurface* *WallSurface* *RoofSurface* *OuterCeilingSurface* *OuterFloorSurface*

Figure 5.16 Surface type recognition of two example buildings.

### 5.4.2 Recovery of Relationships between Building Surfaces

According to the second constraint in Equation (5-11), for each $E_{ij} \in \mathbb{E}$ that is labelled as 1, the two corresponding facets $F_i$ and $F_j$ must be also labelled as 1. Therefore, the relationships between the surface components are recovered based on the edge complex $\mathbb{E}$, and the selected facets are clustered as follows.

For each $E_{ij}$ ($E_{ij} \in \mathbb{E}$ and $\boldsymbol{l}_{E_{ij}} = 1$), if $A(E_{ij}) < \varepsilon$ ($\varepsilon$ is the angle threshold and is the same as in Table 5.1 and Table 5.6), the two corresponding facets $F_i$ and $F_j$ belong to the same surface component. Otherwise, $F_i$ and $F_j$ belong to two different surface components, the two corresponding surface components are considered adjacent and $E_{ij}$ contributes to the intersection edge between the two surface components.

After the clustering, each surface component in the CityGML model corresponds to a set of facets $\{F \in \mathbb{F} \mid \boldsymbol{l}_F = 1\}$, and each pair of adjacent surface components corresponds to a set of edges $\{E \in \mathbb{E} \mid \boldsymbol{l}_E = 1\}$. The normal of each surface component is determined by averaging the normals of its corresponding facets, and then the angles between the adjacent surface components are computed. The two angles of 0° (± $\varepsilon$) and 90° (± $\varepsilon$) are recorded to present the parallel and perpendicular relationships, respectively.

Table 5.7 shows an example of the recovered relationships, i.e., *adjacent*, *parallel* and *perpendicular*, between the surface components of a building model. The visualised result is shown in Figure 5.17, where the target surfaces are shown in red and the queried surfaces are in yellow.

Table 5.7 Relationships between the surface components of an example building.

| ***SurfaceID1*** | ***SurfaceID2*** | ***Adjacent*** | ***Parallel*** | ***Perpendicular*** | ***Angle*** |
|---|---|---|---|---|---|
| Surface0 | Surface1 | 1 | 0 | 1 | 90.0 |
| Surface0 | Surface2 | 0 | 0 | 1 | 90.0 |
| Surface0 | Surface3 | 1 | 0 | 0 | 135.0 |
| Surface0 | Surface4 | 0 | 1 | 0 | 0.0 |
| … | … | … | … | … | … |
| Surface1 | Surface0 | 1 | 0 | 1 | 90.0 |
| Surface1 | Surface2 | 1 | 0 | 0 | 135.6 |
| Surface1 | Surface3 | 0 | 0 | 0 | 45.0 |
| … | … | … | … | … | … |

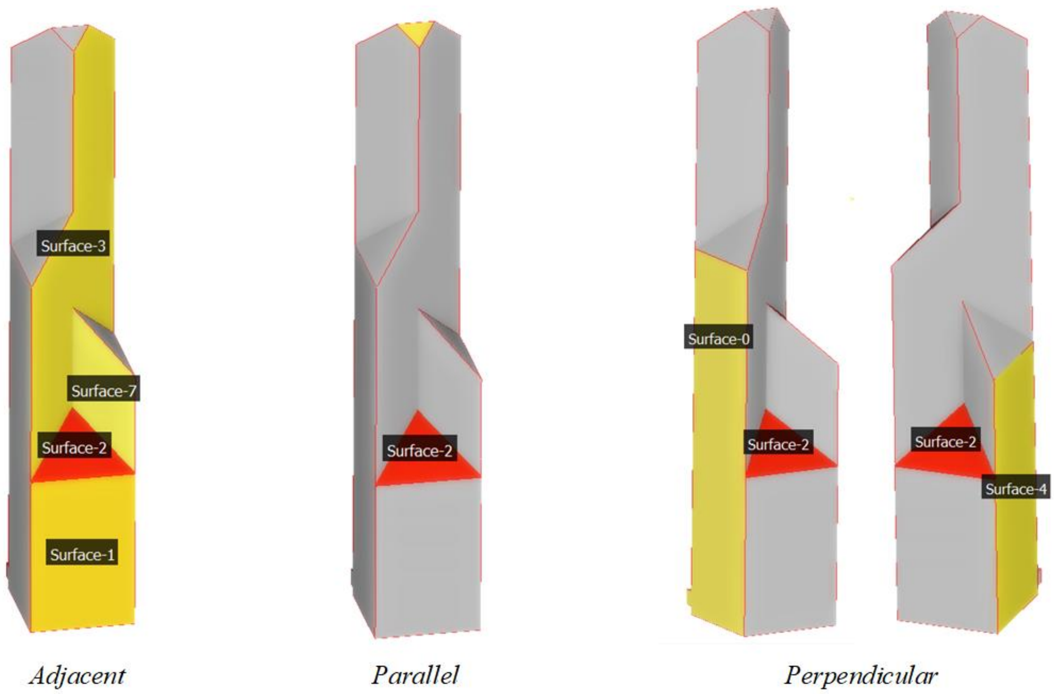



Figure 5.17 Topological and geometric relationships between the surface components of an example building.

## 5.5 Summary and Discussion

This chapter presents the reconstruction of the true 3D building models that conform to CityGML from points known to belong to the *building* class. The geometric models of the buildings are first generated through a space-partition-and-approximation strategy. Multiple relations are introduced into this procedure, including the geometric relationships between the planes for the refinement of building planar primitives, the 2D and 1D topological relationships for the extraction of facet and edge elements, and the topological relationships between the cells encoded in the ILP functions for the optimal selection of occupied cells. All of these relationships are used comprehensively to constrain the geometries of the building models to be watertight, manifold and regularised. The geometric models of buildings with refined surface normals are then converted into CityGML format with specified surface types. Topological relationships between the building surfaces are also determined and recorded for applications related to spatial query.

The geometric and CityGML modelling results of several example buildings are shown in this chapter to emphasise the features of the developed method. In the next chapter, 3D reconstruction results of large-scale buildings are examined using systematic experimental evaluations, where the segmentation and classification results of the point clouds are also demonstrated and evaluated.

# Chapter 6 Experimental Evaluation and Analysis

This chapter systematically evaluates the developed methods described above using three datasets. The first dataset is a benchmark MLS point cloud *Paris-rue-Cassette* (Vallet et al., 2015), the second is a point cloud obtained from the mobile mapping system (MMS) and presents a scene in Sham Shui Po, Hong Kong, and the third is a DIM point cloud acquired in Central, Hong Kong, generated from oblique images via the MVS pipeline. For quantitative evaluation, the following metrics are defined.

$$overall\ accuracy = \frac{total\ number\ of\ matched\ points}{total\ number\ of\ points} \tag{6-1}$$

$$precision = \frac{number\ of\ matched\ points\ (objects)}{number\ of\ detected\ points\ (objects)} \tag{6-2}$$

$$recall = \frac{number\ of\ matched\ points\ (objects)}{number\ of\ points\ (objects)\ in\ ground\ truth} \tag{6-3}$$

$$F_1\ score = \left(\frac{precision^{-1} + recall^{-1}}{2}\right)^{-1} \tag{6-4}$$

where *precision* measures the correctness of the segmentation/classification, *recall* measures the completeness, and $F_1$*-score* is a comprehensive metric of correctness and completeness.

For the quantitative evaluation of the geometric accuracy of models generated from the building point clouds, the root-mean-square error (RSME) formulated as in Equation (6-6) is used.

$$RSME = \sqrt{\frac{\sum_{p \in \mathcal{P}_B} \| p - \mathcal{M}_B \|^2}{N(\mathcal{P}_B)}} \tag{6-5}$$

where $\| p - \mathcal{M}_B \|$ denotes the Euclidean distance between a point $p$ and the geometric model $\mathcal{M}_B$ of a building $B$, and $N(\mathcal{P}_B)$ is the number of points in the building point cloud $\mathcal{P}_B$.

The remainder of this chapter is organised as follows. The experimental results of the *Paris-rue-Cassette* and the *Hong Kong (Sham Shui Po)* datasets are described and evaluated in Sections 6.1 and 6.2, respectively. As these two datasets are ground-obtained point clouds, the buildings therein are highly incomplete; therefore, the developed 3D building reconstruction method is not tested on these two datasets. Detailed results of the 3D building reconstruction with (mostly) complete buildings extracted from the classification results of the DIM point cloud, the *Hong Kong (Central)* dataset, are given and evaluated in Section 6.3.

## 6.1 Experiment Using the *Paris-rue-Cassette* Dataset

### 6.1.1 Data Description

The *Paris-rue-Cassette* dataset (Vallet et al., 2015) was acquired in Paris with a Stereopolis II MLS system (Paparoditis et al., 2012) in January 2013. This dataset consists of 300 million points covering 10 km of streets and a 1 km$^2$ square in the 6$^{th}$ district of Paris, as shown in Figure 6.1, where the two black boxes give detailed views of the square and a street scene. In this dataset, 12 million points covering a street of about 200 m (as shown in the yellow rectangle in Figure 6.1) have point-wise ground-truth labels in terms of both segments and classes. Therefore, in the following, the experimental evaluations and analyses of this dataset are made based on the part with ground-truth, as did in Weinmann et al. (2015b) and Hackel et al. (2016).

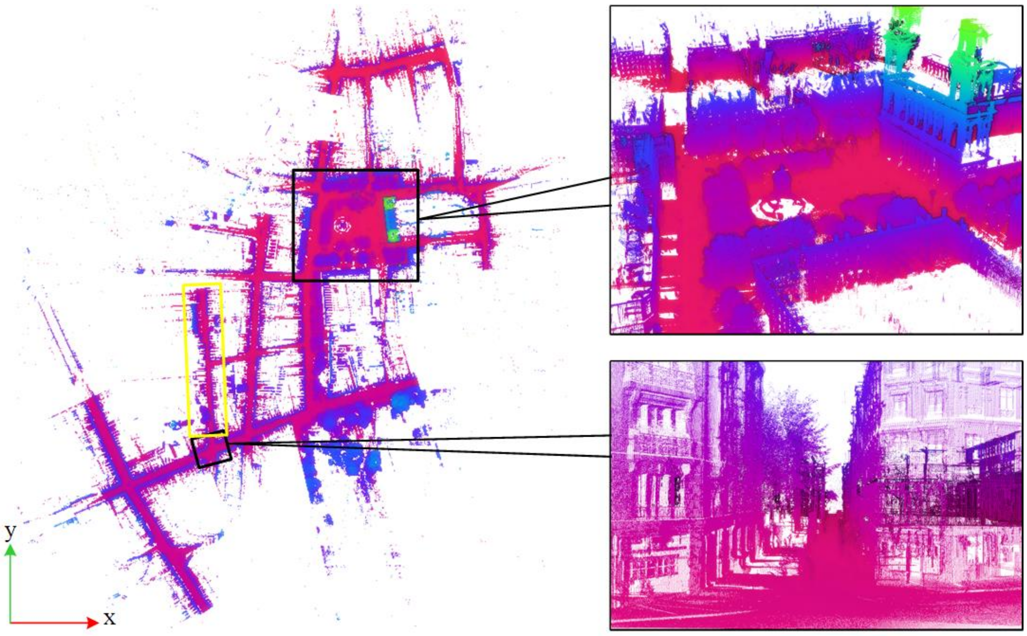



Figure 6.1 Overview of the *Paris-rue-Cassette* dataset (colour-coded by height).

Before the segmentation of the point cloud, which is the first step of the point cloud modelling framework proposed in this work, an adaptive surface filter (Hu et al., 2014) is used to separate the non-ground points from the ground points, aiming to reduce the long interactions between different ground objects. The ground and non-ground points extracted from the *Paris-rue-Cassette* dataset are shown in Figure 6.2.

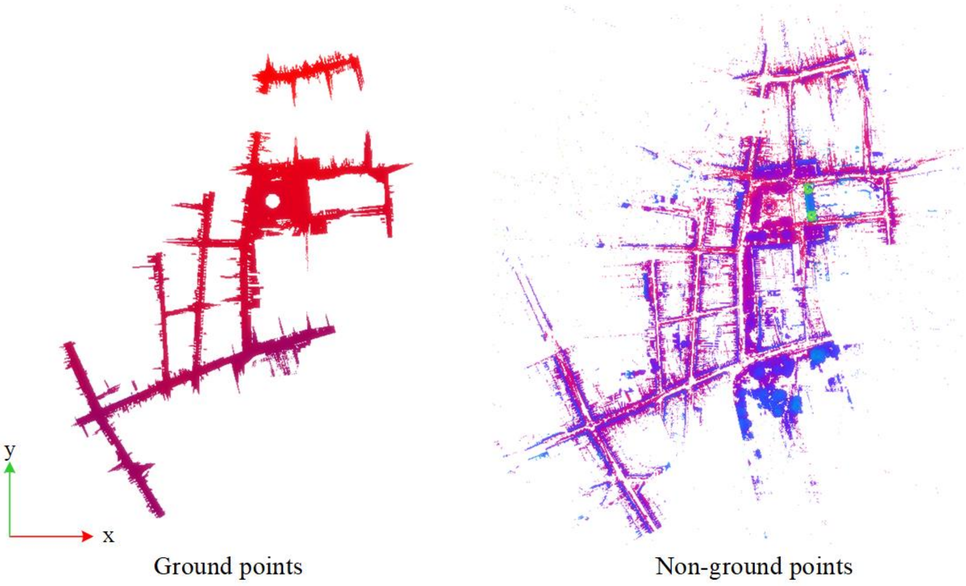
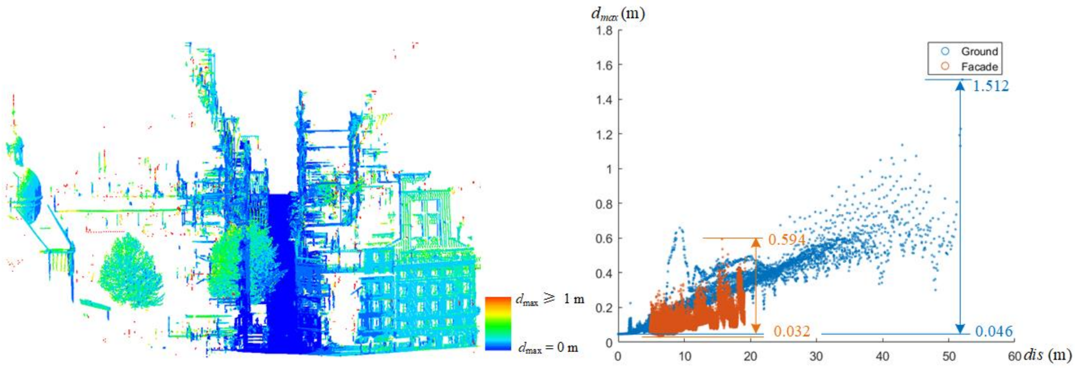



Figure 6.2 Ground and non-ground points extracted from the *Paris-rue-Cassette* dataset.

For the MLS point cloud, the point density can vary dramatically with the scanning distance changing from the near ranges to the far ranges. The variations in point density of the *Paris-rue-Cassette* dataset are quantitatively evaluated on the basis of $d_{max}$ — the maximum distance between a point and its $k$ ($k$ = 20) nearest neighbours, as shown in Figure 6.3 (a). Note that $d_{max}$ is inversely related to point density. Figure 6.3 (b) shows the statistical results of the $d_{max}$ of the ground and façade points versus their scanning distances *dis*. It can be seen that with the scanning distance increasing, $d_{max}$ increases (i.e., point density decreases) with a considerable rate for both ground and façade points.




(a) Non-ground points color-coded by $d_{max}$.

(b) $d_{max}$ of the ground and façade points versus their scanning distances *dis*.

Figure 6.3 Variations in point density of the *Paris-rue-Cassette* dataset.

### 6.1.2 Segmentation Result

Figure 6.4 (a) ~ (c) shows the intermediate productions and the final structural labelling result of the *Paris-rue-Cassette* dataset. It can be found that supervoxels in areas with low point densities, e.g., tree crowns and tops of buildings, have larger sizes than those in areas with high point densities. This guarantees that effective local shape descriptors can be derived from the supervoxels for the structural labelling. To quantitatively evaluate the accuracy of the structural labelling result, the non-ground points in the test area (the yellow box in Figure 6.1) are manually labelled into three structural categories, i.e., *linearity*, *planarity* and *scattering*, as shown in Figure 6.4 (d). The result shows that the total overall labelling accuracy of the *Paris-rue-Cassette* dataset is 93.32%.

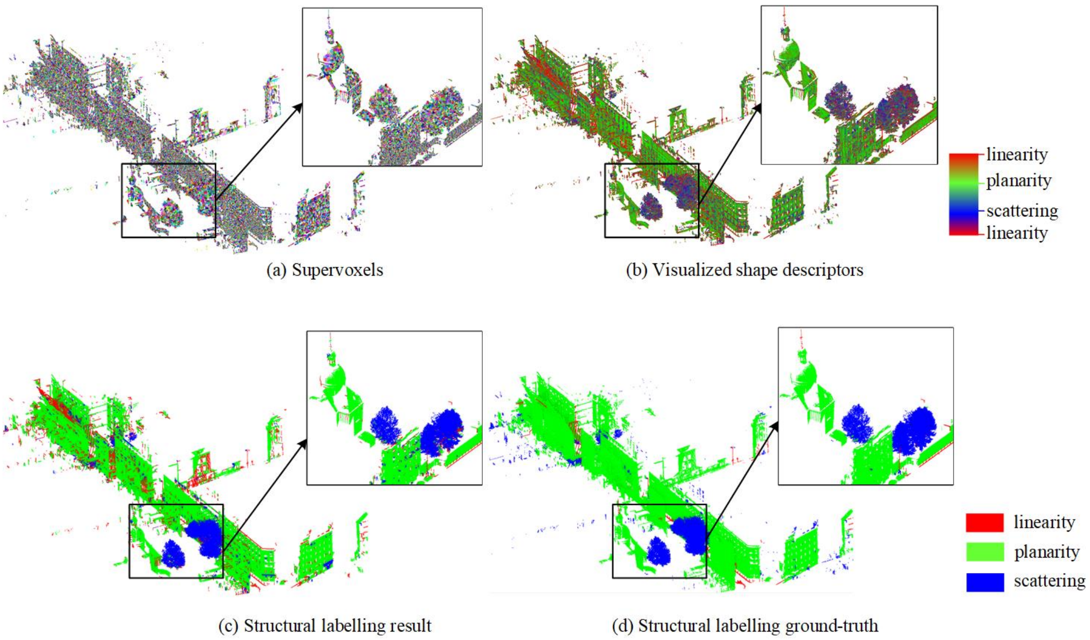
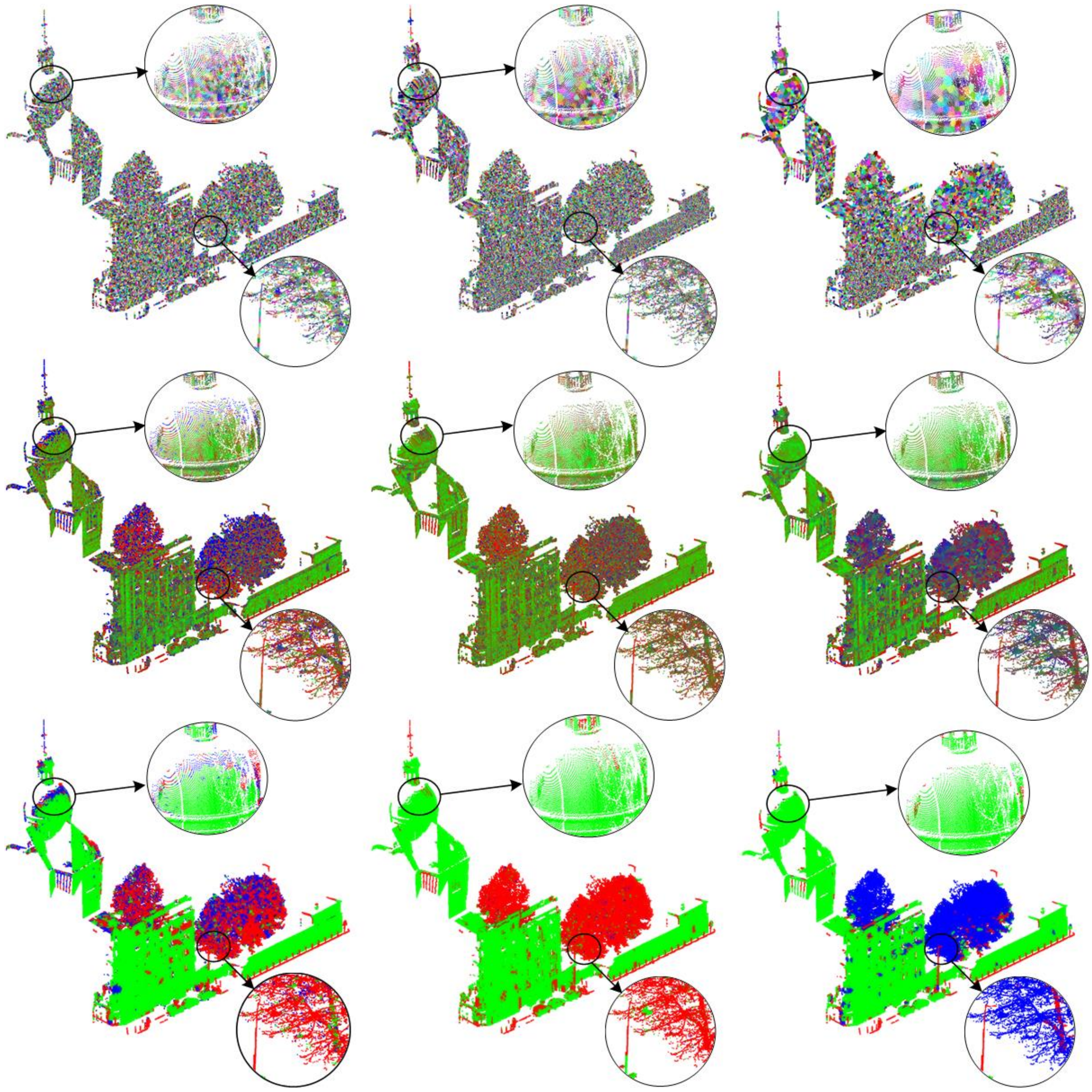



Figure 6.4 Multi-size supervoxels and structural labelling result of the *Paris-rue-Cassette* dataset.

The structural labelling result is compared with the results based on supervoxels generated by two other methods, which are the VCCS (Papon et al., 2013) and TBBP (Lin et al., 2018). The VCCS is a well-known method that produces supervoxels with fixed sizes, resulting in that supervoxels in areas with low point densities contain insufficient points for the computation of meaningful local features, as shown in Figure 6.5 (a). The TBBP is one of the state-of-the-art methods that produces supervoxels with adaptive sizes, and it can well preserve the boundaries of objects. However, the adaptivity of this method tends to make it susceptible to data quality and present scatter objects as linear ones, as shown in Figure 6.5 (b). The fixed resolution of the VCCS, the expected resolution of the TBBP, and the minimum resolution of the proposed method are all set as 0.3 m. The overall accuracies of the results based on VCCS and TBBP are 91.87% and 87.94%, respectively, considerably lower than the result of the method proposed in this research (93.32%).



(a) VCCS (Papon et al., 2013) (b) TBBP (Lin et al., 2018) (c) Proposed method

Figure 6.5 Structural labelling result of the *Paris-rue-Cassette* dataset compared with the results based on VCCS and TBBP.

Based on the structural labelling result, total 3442 structural components of the *Paris-rue-Cassette* dataset were generated as shown in Figure 6.6 (a) and the ground-truth of the segmentation is shown in Figure 6.6 (b). A structural component $S^{SC}$ and a segment in the ground-truth $S^{GT}$ are defined as matched if

$$\frac{|S^{GT}|}{|S^{GT} \cup S^{SC}|} > m \quad \text{and} \quad \frac{|S^{SC}|}{|S^{GT} \cup S^{SC}|} > m \tag{6-6}$$

where $|\cdot|$ denotes the cardinal (number of points) of a set, and $m = 0.5$ results in a 1-to-1 matching result.

Based on Equation (6-6), 255 structural components were matched with the segments in the ground-truth (matched objects are coloured in yellow in Figure 6.6 (c), and those mismatched are in red), leading to a recall value of 60.7% and a low precision value of 7.4%. This might be because $m$ = 0.5 is a very strict constraint that penalises over-segmentation too much (Vallet et al., 2015), while "over-segmentation", e.g., to separate wall lamps from the building façade (as shown in the top row of Figure 6.6 (d)), is one of the purposes of the generation of structural components. In addition, fragmental point sets (as shown in the bottom row in Figure 6.6 (d)) could be easily determined as structural components, while they belong to one segment with the building façade in the ground-truth. The fragmentation of the point cloud is extremely serious, leading the number of structural components to be far greater than the number of segments in the ground-truth.

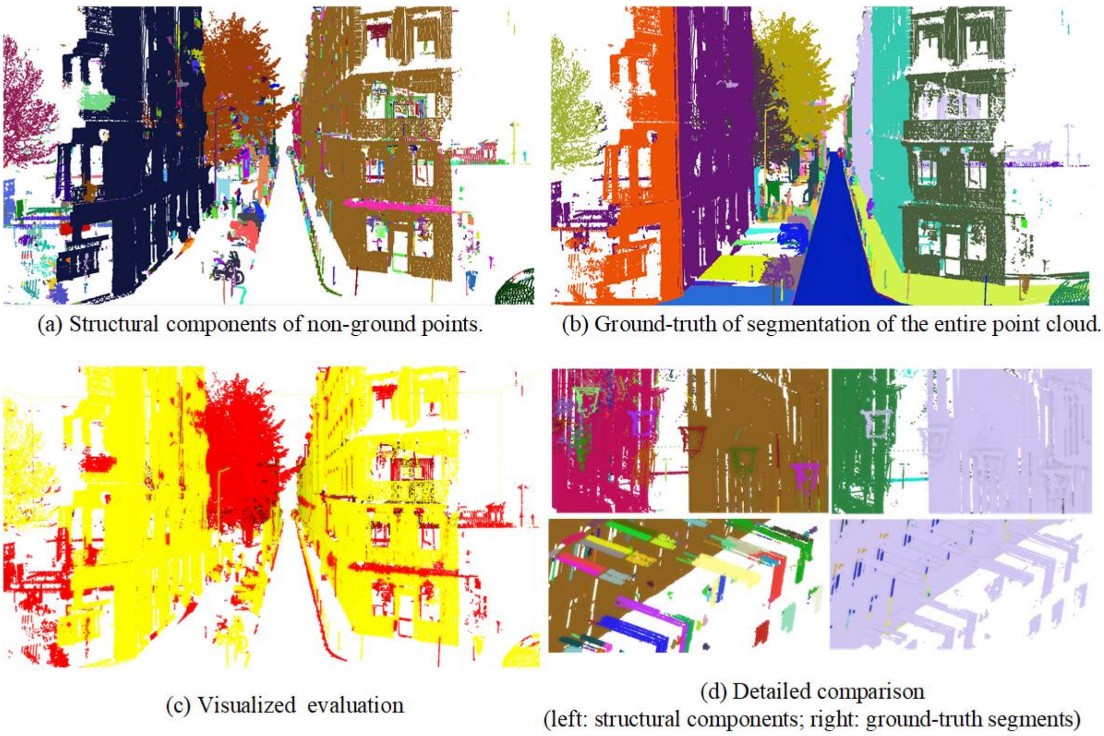



Figure 6.6 Structural components of the *Paris-rue-Cassette* dataset and comparison with the ground-truth.

### 6.1.3 Classification Result

With respect to semantic objects, nine categories (*façade*, *ground*, *vegetation*, *car*, *motorcycle*, *pedestrian*, *traffic sign*, *street lamp* and *bollard*) of objects appearing in this street are considered. To train the RF classifiers, a number of objects corresponding to the above categories were firstly manually detected as training samples from areas beyond the yellow rectangle in Figure 6.1. Each category contains training samples with different point densities (as shown in Figure 6.7), and the distribution of structural components containing more than $n_{min}$ = 40 points generated from these samples is shown in Table 6.1.

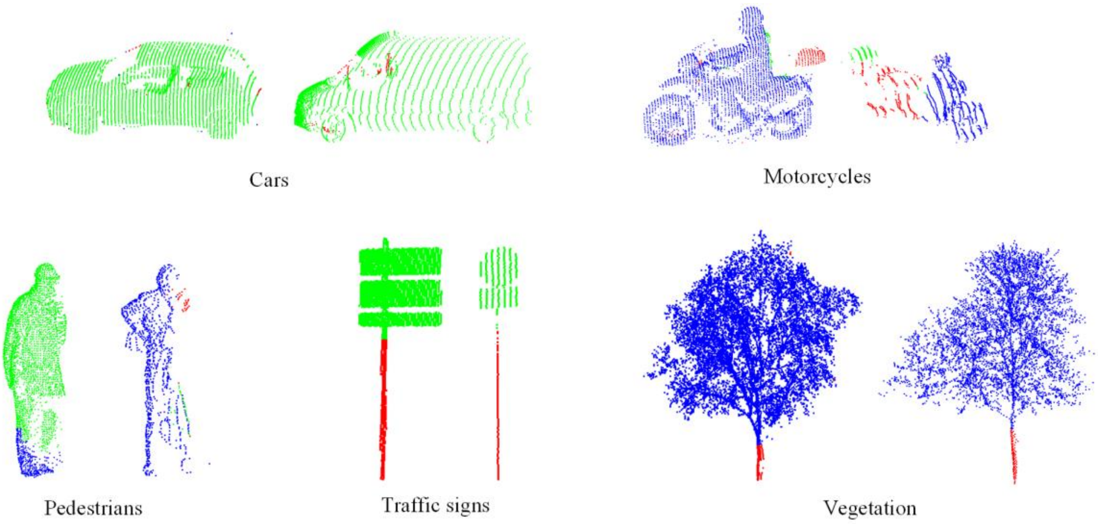



Figure 6.7 Examples of manually selected training samples with different point densities. Red, green and blue refer to linear, planar and scatter structural components, respectively.

Table 6.1 Number distribution of manually selected samples and valid structural components generated from these samples. (*Ground* refers to small, isolated ground patches).

| | ***Sample size*** | ***Linear structural component size*** | ***Planar structural component size*** | ***Scatter structural component size*** |
|---|---|---|---|---|
| *Facade* | 8 | 1735 | 1165 | 179 |
| *Ground* | 20 | 29 | 13 | 0 |
| *Vegetation* | 30 | 176 | 23 | 28 |
| *Car* | 35 | 173 | 59 | 15 |
| *Motorcycle* | 35 | 99 | 42 | 35 |
| *Pedestrian* | 30 | 22 | 22 | 11 |
| *Traffic sign* | 26 | 30 | 26 | 1 |
| *Street lamp* | 15 | 18 | 17 | 10 |
| *Bollard* | 56 | 56 | 0 | 0 |

Figure 6.8 shows the semantic classification and visualised evaluation of the testing area in the *Paris-rue-Cassette* dataset. Figure 6.8 (a) and (b) show the ground-truth from two different views (grey in the ground-truth refers to unidentified points), (c) and (d) are the classification results of the developed method, (e) and (f) compare the classification results with the ground-truth, where yellow refers to correctly classified points, red refers to misclassified points and blue refers to points unidentified in the ground-truth.

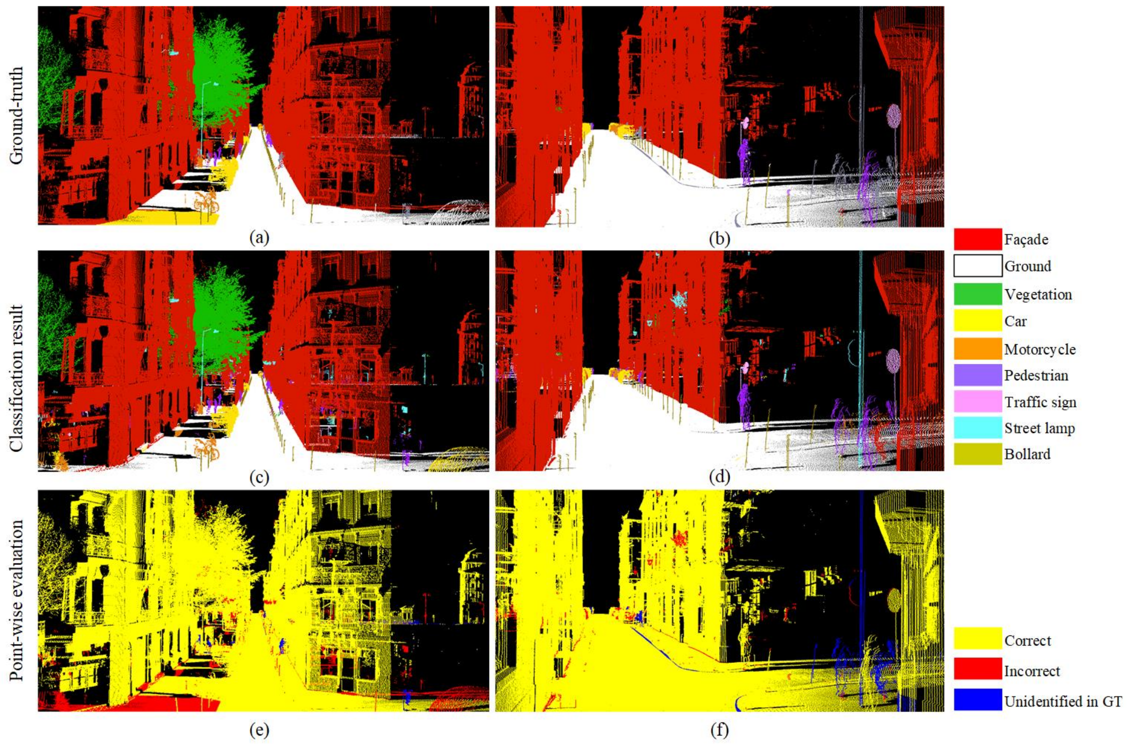



Figure 6.8 Semantic classification and visualised evaluation results of the testing area in the *Paris-rue-Cassette* dataset.

The quantitative evaluation results were compared with those of the studies of Weinmann et al. (2015b) and Hackel et al. (2016), as shown in Table 6.2 ~ 6.5. Note that, the unweighted mean $F_1$-scores shown in Table 6.5 are computed based on the $F_1$-scores of the in common categories in this research and previous studies. In terms of class-wise metrics, the proposed method yields the best results for most categories. Especially for *vegetation*, the proposed method significantly increases the precision and $F_1$-score by 35.81% and 23.09%, respectively, compared with the best results of the previous studies. The proposed method also shows the best performance in extracting pedestrians, with a high recall value of 93.36% and the highest precision and $F_1$-scores of 38.52% and 54.54%, respectively. With respect to the categories of *façade* and *ground*, which constitute more than 90% of the *Paris-rue-Cassette* dataset, extremely good results, with all three class-wise metrics higher than 96%, are derived from this research, whilst *façade* has the highest $F_1$-score of 98.60%. The recall values of *car* and *motorcycle* are relatively lower than the best results from previous studies.

This might be caused by that around 12% of the points labelled as *car* in the ground-truth (as shown in the left-bottom of Figure 6.8 (a)) are labelled as ground by the proposed method (see the result shown in Figure 6.8 (c) and the visualised comparison in Figure 6.8 (e)), for which it is believed that the result of the proposed method is more reasonable than the ground-truth.

Table 6.2 Recall (%) of the classification result of the *Paris-rue-Cassette* dataset compared with previous studies.

| | ***Façade*** | ***Ground*** | ***Vegetation*** | ***Car*** | ***Motor-cycle*** | ***Pedestrian*** | ***Traffic sign*** | ***Street lamp*** | ***Bollard*** |
|---|---|---|---|---|---|---|---|---|---|
| (Weinmann et al., 2015b) | 87.21 | 96.46 | 86.02 | 61.12 | 82.85 | 82.25 | 76.57 | - | - |
| (Hackel et al., 2016) | 94.21 | 98.22 | 84.78 | **93.07** | **97.58** | **96.87** | 89.63 | - | - |
| Proposed method | **97.95** | **99.45** | **89.59** | 76.56 | 87.86 | 93.36 | **96.16** | 92.74 | 82.64 |

Table 6.3 Precision (%) of the classification result of the *Paris-rue-Cassette* dataset compared with previous studies.

| | ***Façade*** | ***Ground*** | ***Vegetation*** | ***Car*** | ***Motor-cycle*** | ***Pedestrian*** | ***Traffic sign*** | ***Street lamp*** | ***Bollard*** |
|---|---|---|---|---|---|---|---|---|---|
| (Weinmann et al., 2015b) | 99.28 | **99.24** | 25.66 | 67.67 | 17.74 | 14.95 | 9.24 | - | - |
| (Hackel et al., 2016) | **99.64** | 98.71 | 56.62 | 86.08 | **51.99** | 18.99 | **24.88** | - | - |
| Proposed method | 99.25 | 96.89 | **92.43** | **93.07** | 42.46 | **38.52** | 14.63 | 8.36 | 89.63 |

Table 6.4 $F_1$-score (%) of the classification result of the *Paris-rue-Cassette* dataset compared with previous studies.

| | ***Façade*** | ***Ground*** | ***Vegetation*** | ***Car*** | ***Motor-cycle*** | ***Pedestrian*** | ***Traffic sign*** | ***Street lamp*** | ***Bollard*** |
|---|---|---|---|---|---|---|---|---|---|
| (Weinmann et al., 2015b) | 92.85 | 97.83 | 39.53 | 64.23 | 29.23 | 16.61 | 25.01 | - | - |
| (Hackel et al., 2016) | 96.85 | **98.47** | 67.90 | **89.43** | **67.84** | 39.60 | **31.34** | - | - |
| Proposed method | **98.60** | 98.15 | **90.99** | 84.01 | 57.25 | **54.54** | 25.40 | 15.34 | 85.99 |

Table 6.5 Overall accuracy and mean $F_1$-score of the classification result of the *Paris-rue-Cassette* dataset compared with previous studies.

| | (Weinmann et al., 2015b) | (Hackel et al., 2016) | Proposed method |
|---|---|---|---|
| Overall accuracy (%) | 89.60 | 95.74 | **97.13** |
| Mean $F_1$-score (%) | 52.18 | 70.20 | **72.70** |

Another reason for the low recall values might be that the proposed method, which takes advantage of structural information, is likely to fail to identify fragmented cars and motorcycles (as shown in Figure 6.9 (a)), where the geometric structures and relationships are destroyed. Similarly, fragments of *façade* can be incorrectly identified as other objects, e.g., *traffic sign*, as shown in Figure 6.9 (b), which results in relatively lower precision values of the corresponding categories.

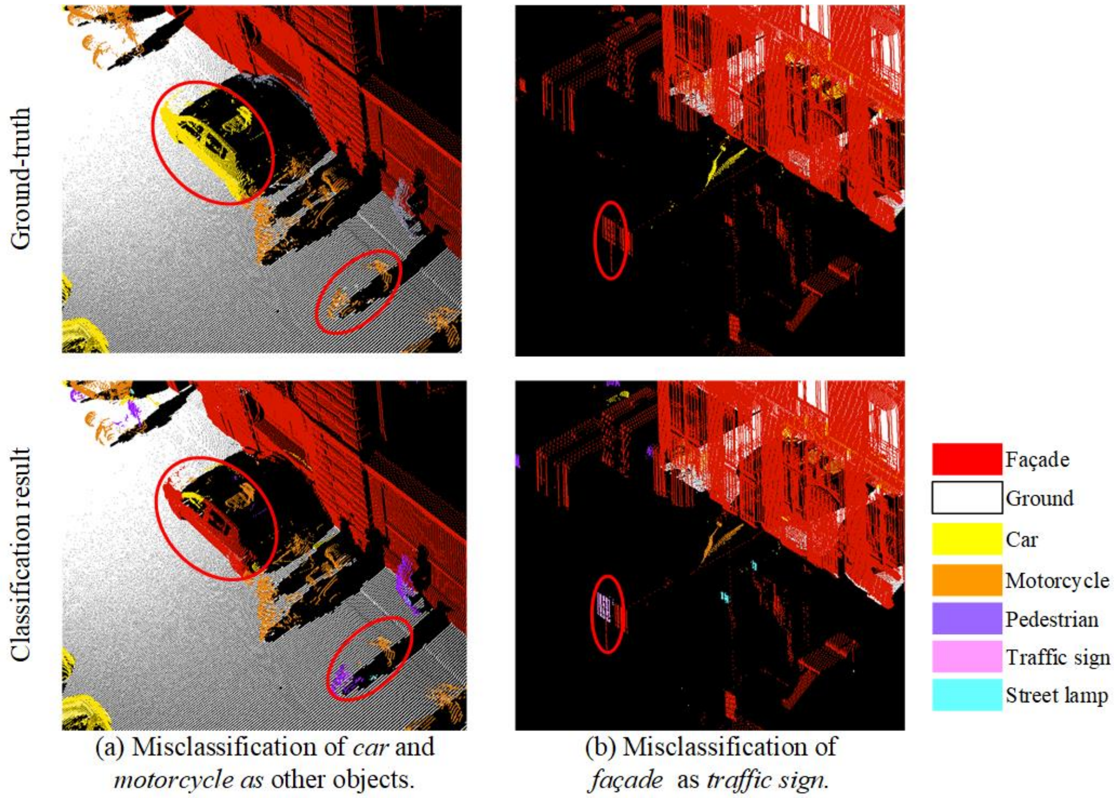



Figure 6.9 Misclassifications (in the red circles) of *car*, *motorcycle* and *façade* caused by fragmentation of the point cloud.

Two additional categories — *street lamp* and *bollard* — are considered in this research. Although the *street lamp* has low precision value for the same reason with traffic sign (see Figure 6.8 (b), (d), (f) and Figure 6.9 (b)), it yields a high recall value of 92.74%. Furthermore, the high recall values of *traffic sign*, *street lamp* and *bollard* indicate the proposed method is able to distinguish between various pole-like objects. According to the global metrics shown in Table 6.5, the proposed method yields the highest overall accuracy of 97.13% and the highest mean $F_1$-score of 72.70%, indicating the effectiveness of the proposed method.

## 6.2 Experiment Using the *Hong Kong (Sham Shui Po)* Dataset

### 6.2.1 Data Description

The *Hong Kong (Sham Shui Po)* dataset was acquired by an UltraCam Mustang MMS, which was equipped with a multi-beam rotating light detection and ranging

(LiDAR) system and a high-resolution camera. The LiDAR point cloud consists of more than 115 million points and covers approximately 2 km of streets in Sham Shui Po, Hong Kong, as shown in Figure 6.10, where the two magnified views in the middle show the details of two crossroads with their corresponding image views shown on the right. This dataset features with heavy noise and large variations in point density and various classes of objects, including the building façades and multiple street facilities.

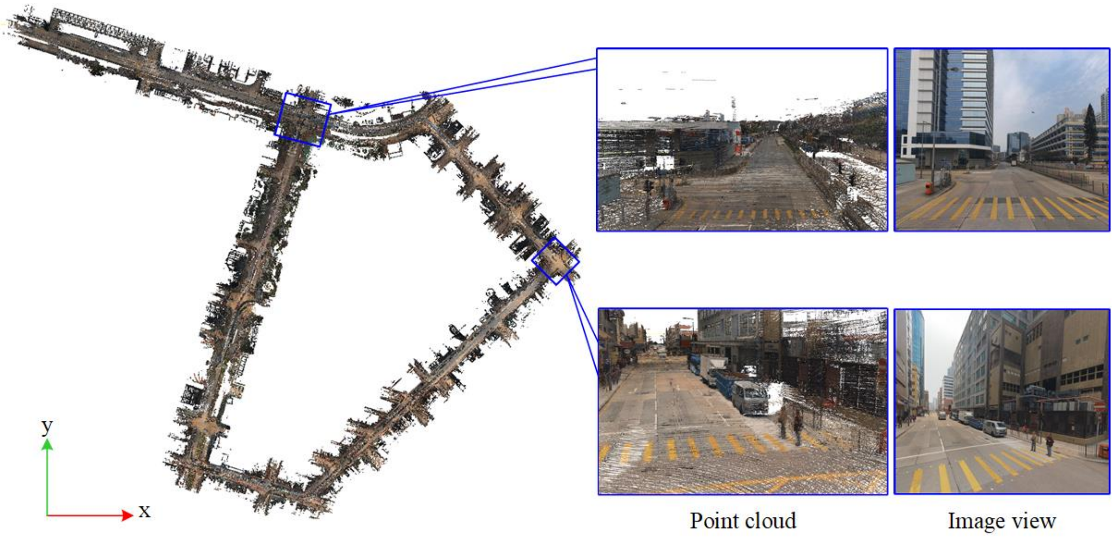
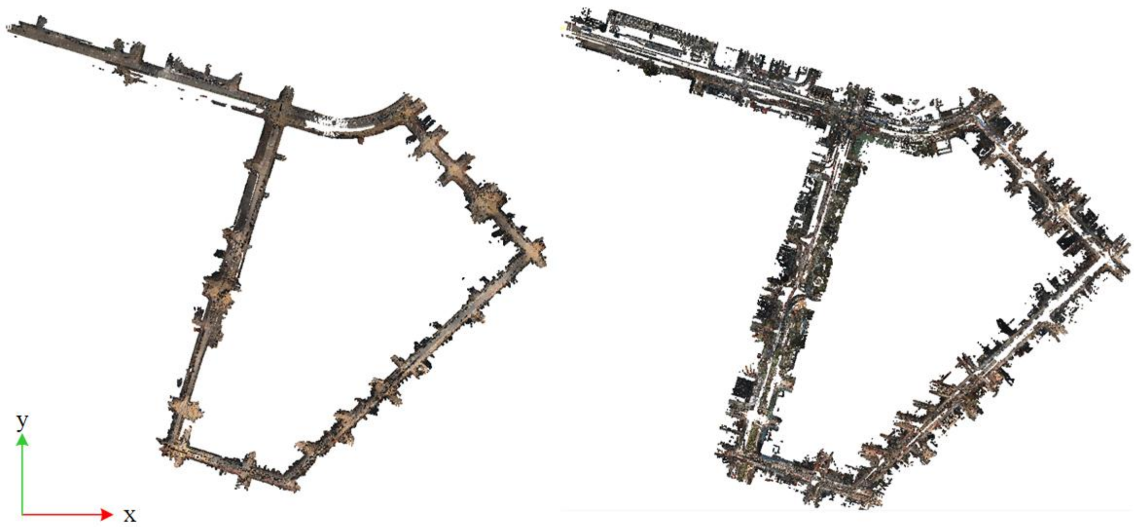


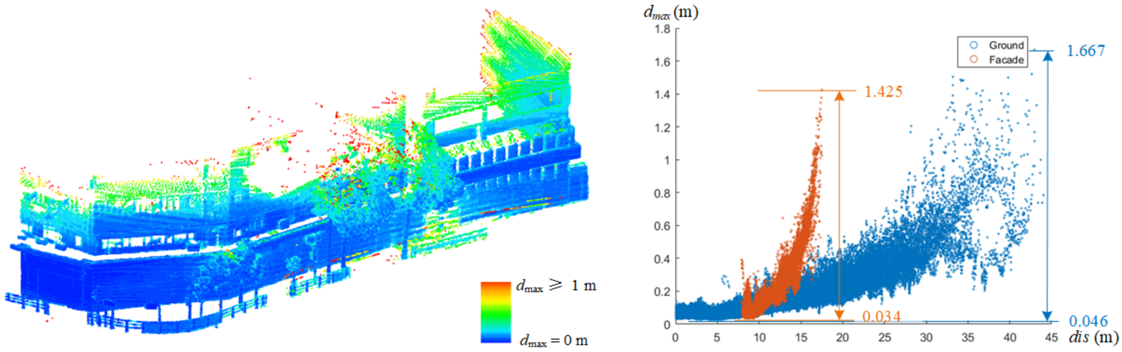

Figure 6.10 Overview of the *Hong Kong (Sham Shui Po)* dataset.

The surface filtering result of the *Hong Kong (Sham Shui Po)* dataset is shown in Figure 6.11, and the variations in point density of this dataset (shown in Figure 6.12) are estimated in the same way with the Paris-rue-Cassette dataset (see Section 6.1.1). According to Figure 6.12 (b), the $d_{max}$ values of the façade points increase from 0.034 m to 1.425 m along with their scanning distances (*dis*) increasing to about 20 m away from the scanner. Figure 6.12 (b) suggests that the point densities of the façade points are about 40 times in close ranges of those in far ranges. The density variations of the ground points are less significant compared with the façade points, but the variations are still notable.




Ground points Non-ground points

Figure 6.11 Ground and non-ground points extracted from the *Hong Kong (Sham Shui Po)* dataset.




(a) Non-ground points color-coded by $d_{max}$.

(b) $d_{max}$ of the ground and façade points versus their scanning distances *dis*.

Figure 6.12 Variations in point density of the *Hong Kong (Sham Shui Po)* dataset.

### 6.2.2 Segmentation Result

The multiple-size supervoxels generated from the *Hong Kong (Sham Shui Po)* dataset and the corresponding structural labelling result are shown in Figure 6.13 (a) ~ (c), and the overall labelling accuracy is 96.9% according to the manually labelled ground-truth (as shown in Figure 6.13 (d)). The structural labelling result of this dataset is also compared with the results based on supervoxels generated by VCCS (Papon et al., 2013) and TBBP (Lin et al., 2018), which have overall labelling accuracies of 95.9% and 85.3%, respectively.

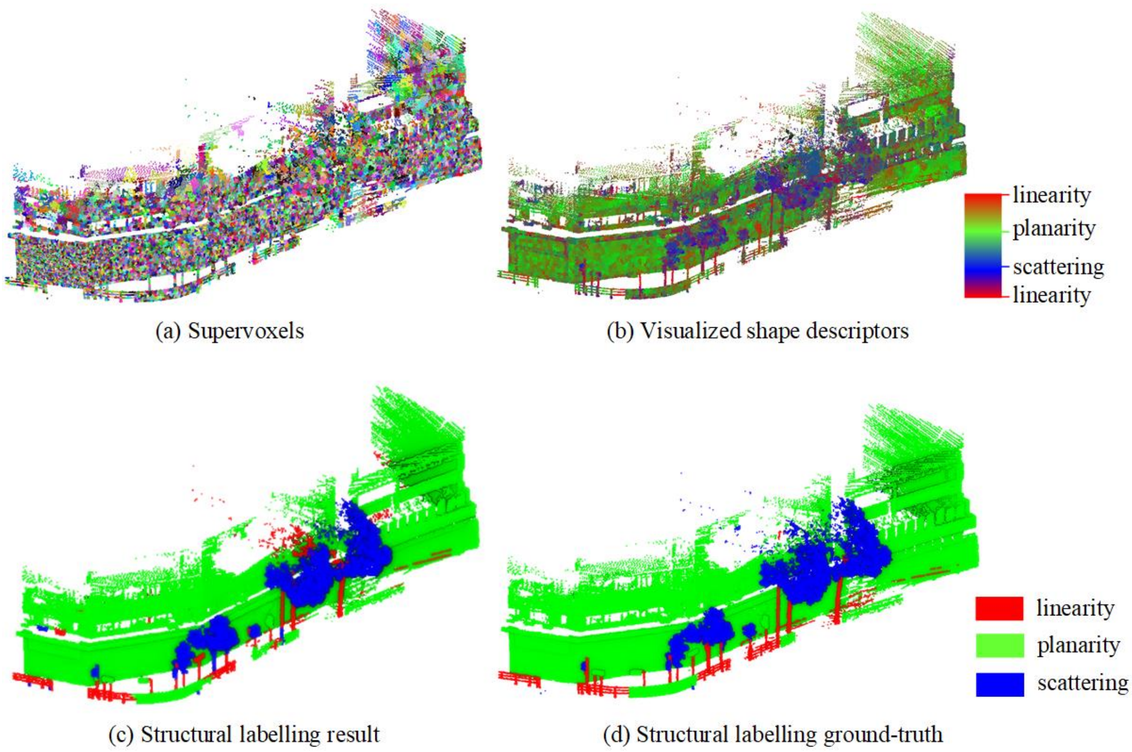
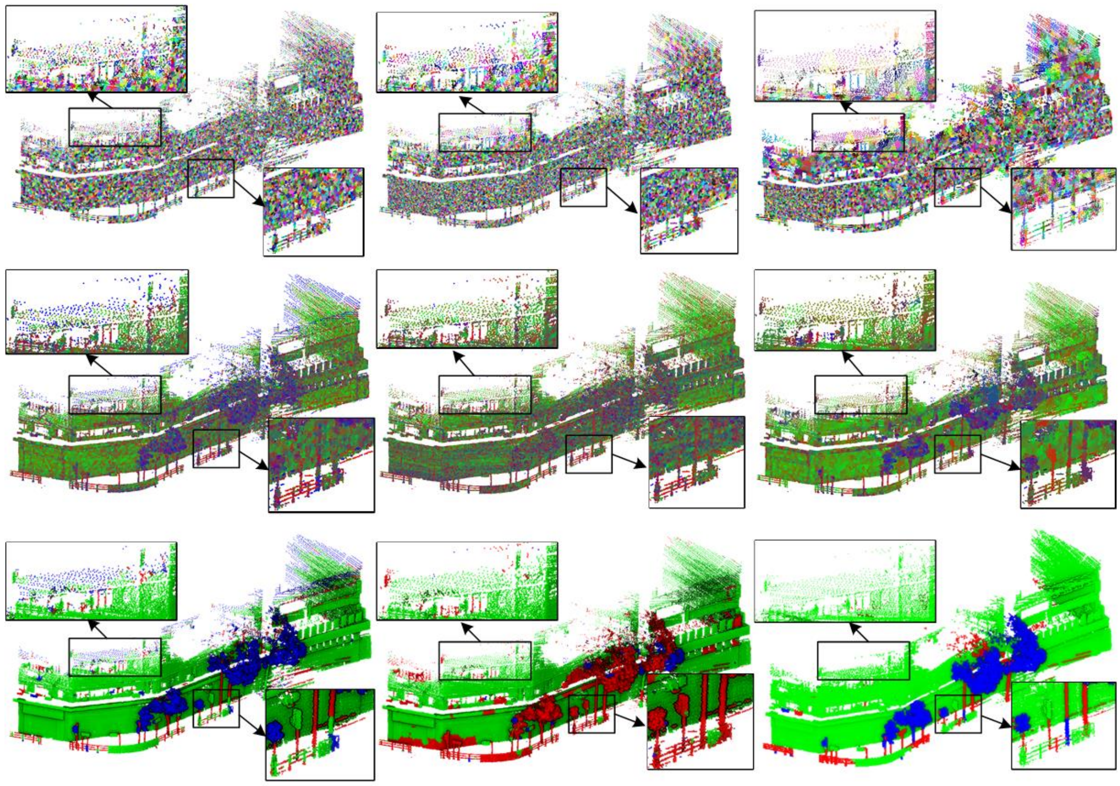



Figure 6.13 Multi-size supervoxels and structural labelling result of the *Hong Kong (Sham Shui Po)* dataset.

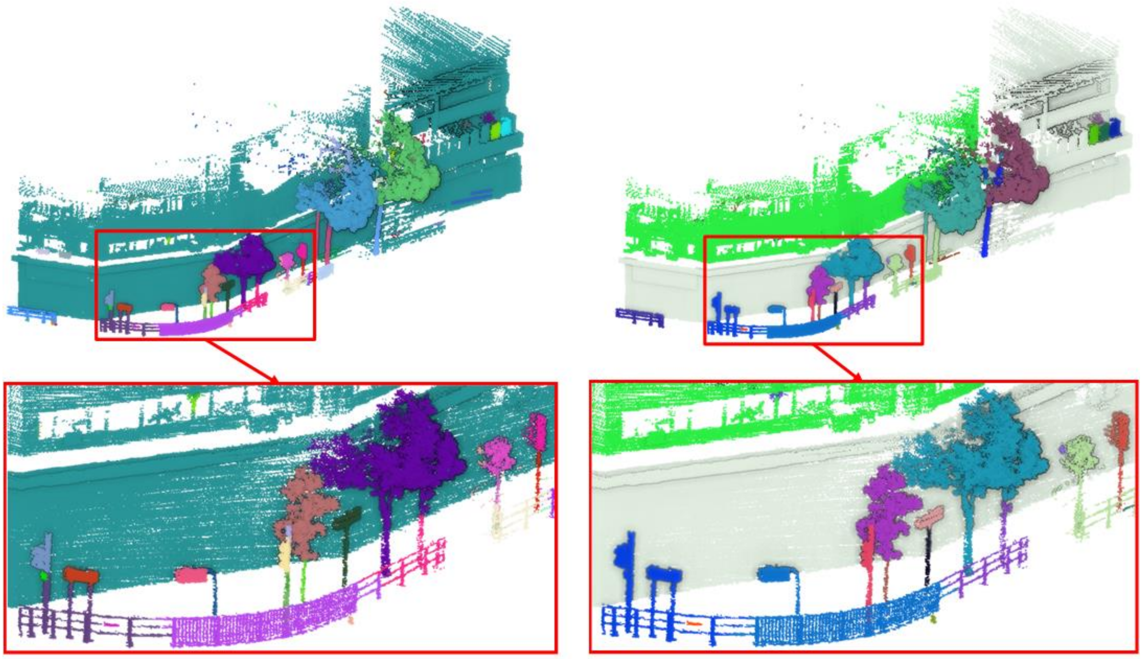
The visualised comparisons with the results based on VCCS and TBBT are shown in Figure 6.14. Similar to the *Paris-rue-Cassette* dataset, the local structural features derived from fixed-sized supervoxels generated by VCCS are erroneous in sparse areas because of the insufficiency of points (as shown in the magnified views of the façade tops in Figure 6.14 (a)). And the TBBT once again shows its weakness at presenting scatter shapes, which are mainly the tree crowns as shown in Figure 6.14 (b). It therefore can be concluded that, compared with VCCS and TBBT, the proposed method performs best for the structural labelling purpose.



(a) VCCS (Papon et al., 2013) (b) TBBP (Lin et al., 2018) (c) Proposed method

Figure 6.14 Structural labelling result of the *Hong Kong (Sham Shui Po)* dataset compared with the results based on VCCS and TBBP.

The structural components of the *Hong Kong (Sham Shui Po)* dataset are also compared with the results of another graph partition method — the $\ell_0$-cut (Landrieu and Obozinski, 2017), which is performed on the same multi-size supervoxels. From Figure 6.15 it can be seen that the structural components are generally consistent with the result of $\ell_0$-cut, while the proposed method deals better with small objects as shown in the magnified views in Figure 6.15. In addition, the $\ell_0$-cut partitions the supervoxels into a set of segments without meaningful labels, but only a series of numbers. On the contrary, the structural components generated by the proposed method inherently have labels indicating their local geometric characteristics.



(a) Structural components (b) $\ell_0$-cut result

Figure 6.15 Structural components generated from the *Hong Kong (Sham Shui Po)* dataset and the segmentation result of $\ell_0$-cut.

### 6.2.3 Classification Result

In the *Hong Kong (Sham Shui Po)* dataset, more categories are considered, including *façade*, *ground*, *vegetation*, *car*, *pedestrian*, *guardrail*, *traffic sign*, *traffic light*, *street lamp*, *others* and *mussy points*. *Others* refers to some less common objects, e.g., fire hydrants, post-boxes and garbage bins, and *mussy points* refers to scanning artefact caused by fast-moving vehicles or pedestrians.

The classification result of the entire *Hong Kong (Sham Shui Po)* dataset is shown in Figure 6.16. The ground-truth semantic labels of points in the training area covering a ~380 m street and in the testing area covering ~660 m streets were manually labelled. The testing area includes a piece of urban arterial road and a relatively narrow street, as shown in Figure 6.16.

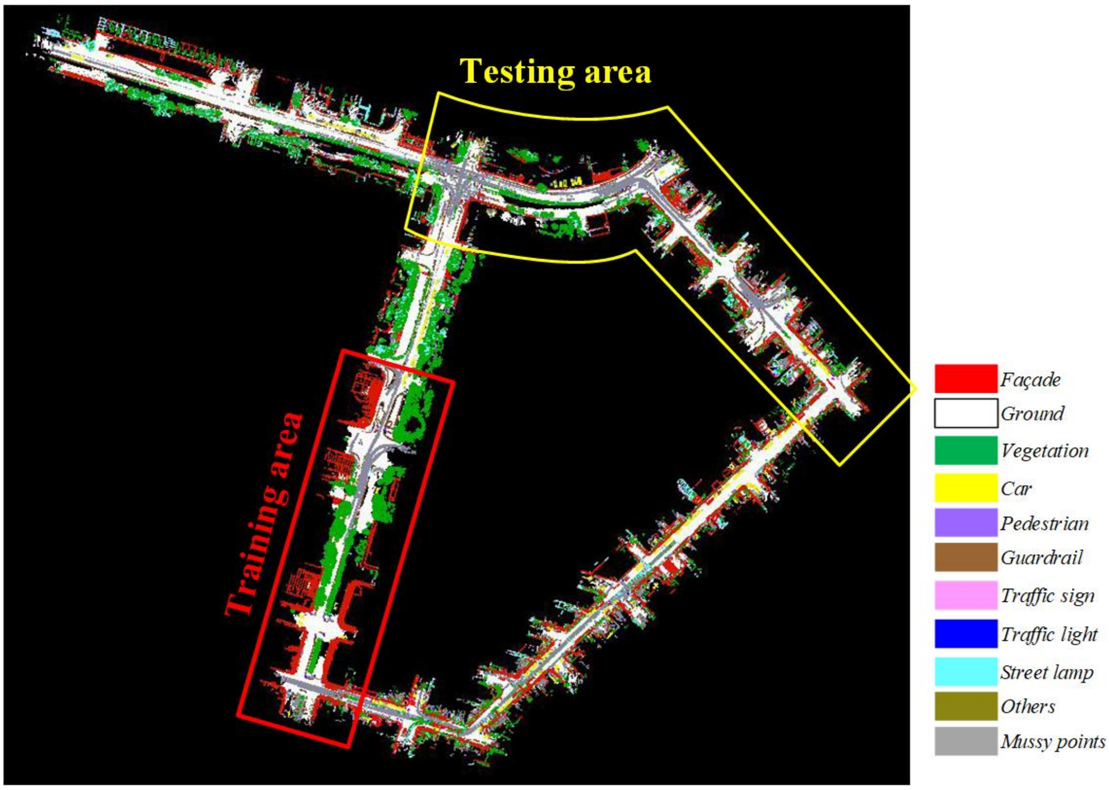



Figure 6.16 Overview of the classification result of the *Hong Kong (Sham Shui Po)* dataset.

The qualitative evaluation results in the testing area are shown in Figure 6.17, where (a) and (b) are the ground-truth, (c) and (d) are the classification results from two different views, and (e) and (f) show the comparisons between the classification results and the ground-truth. According to Figure 6.17, misclassifications occur mainly with fragmented objects that have very sparse densities and connected objects with similar shapes (e.g., the guardrail connected to the green belt in the middle of the road as shown in the black circles in Figure 6.17 (a), (c) and (e)). Close proximity between objects and unexpected appurtenances (e.g., a sign on the pole of a street lamp) may also cause partial misclassifications of the objects, as illustrated by the black boxes in Figure 6.17 (b), (d) and (f). These errors in classification were probably propagated from errors in the segmentation, because the objects are too close to be correctly decomposed. In practice, such misclassifications can be interactively corrected at the supervoxel level in an efficient way. In general, the proposed method showed considerably good performance in distinguishing multiple objects, even objects that are locally or globally similar, e.g., the traffic sign, traffic light and street lamp.

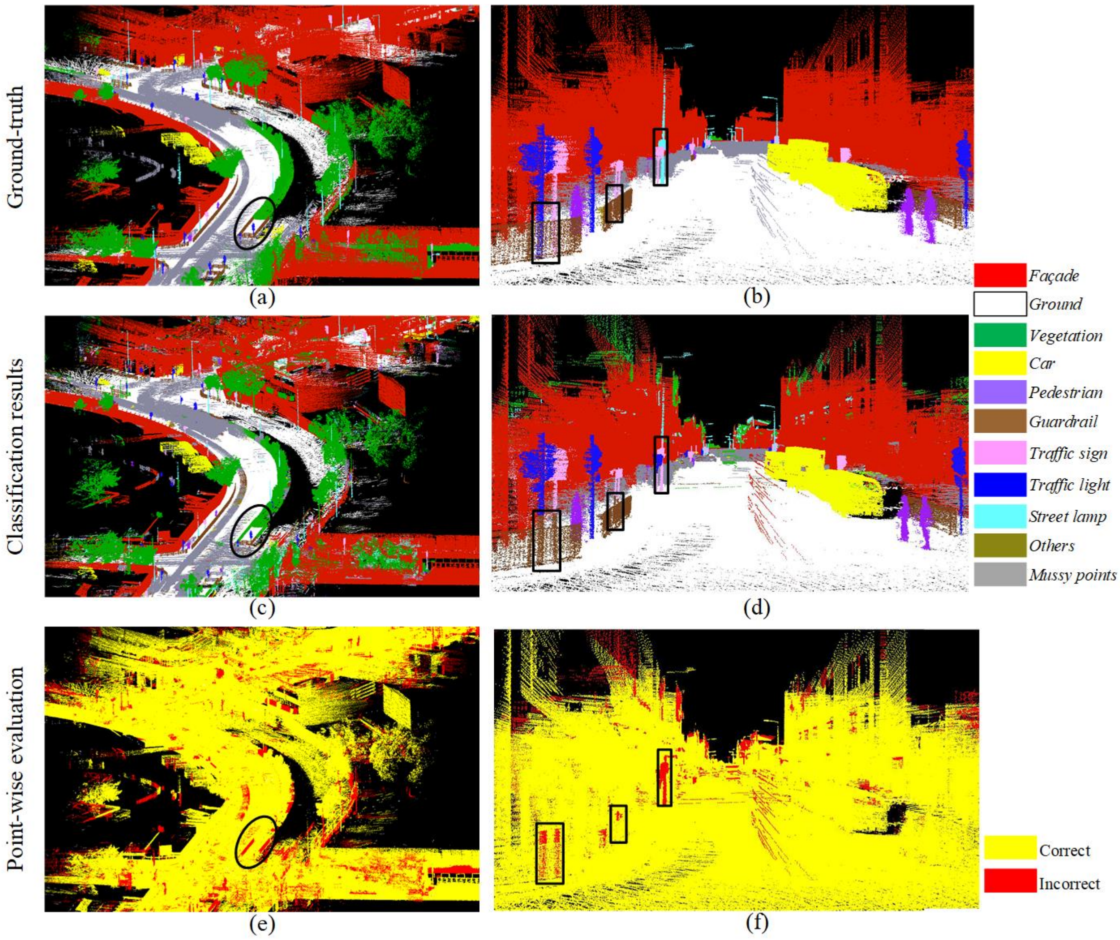



Figure 6.17 Semantic classification and visualised evaluation results of the testing area in the *Hong Kong (Sham Shui Po)* dataset.

To verify the effectiveness of the structural information and high-order contextual information, three comparative trials are designed for quantitative comparison. The first trail (SV + RF) uses the height, 2D and covariance features extracted from individual supervoxels to train a single RF classifier, and no structural or contextual information is used. The second trial (SC + RF) uses all features described in Section 4.2.1 extracted from structural components to train a single RF, and no contextual information is introduced. The third trail (SC + pairwise CRF) introduces pairwise contextual information on the basis of the second trail. The results of these comparative trials are quantitatively compared with the results of the proposed method (SC + high-order CRF), as shown in Table 6.6 ~ 6.9. The pairwise weight factor α is set 0.3 in both the third trial and the proposed method, and the high-order weight factor β is 20.

Table 6.6 Recall (%) of the classification result of the *Hong Kong (Sham Shui Po)* dataset compared with comparative trials.

| | *Façade* | *Ground* | *Vegetation* | *Car* | *Pedestrian* | *Guardrail* | *Traffic sign* | *Traffic light* | *Street lamp* | *Mussy points* | *Others* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SV + RF | 89.12 | 99.76 | 33.73 | 75.53 | 44.21 | 35.01 | 66.97 | 74.95 | 37.77 | 60.64 | **35.99** |
| SC + RF | 94.87 | **99.78** | 80.86 | 81.85 | 54.63 | 76.09 | **71.46** | 85.48 | 52.29 | 88.62 | 35.11 |
| SC + pairwise CRF | 95.90 | 99.77 | 83.65 | 83.61 | 60.96 | **78.18** | 68.86 | 87.15 | **54.38** | 95.67 | 26.81 |
| SC + high-order CRF | **96.06** | 99.75 | **84.02** | **83.73** | **61.14** | **78.18** | 68.92 | **87.24** | 54.27 | **96.14** | 26.49 |

Table 6.7 Precision (%) of the classification result of the *Hong Kong (Sham Shui Po)* dataset compared with comparative trials.

| | *Façade* | *Ground* | *Vegetation* | *Car* | *Pedestrian* | *Guardrail* | *Traffic sign* | *Traffic light* | *Street lamp* | *Mussy points* | *Others* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SV + RF | 79.02 | 99.32 | 65.51 | 84.63 | 45.98 | 84.16 | 22.24 | **86.31** | 41.72 | 40.46 | 25.94 |
| SC + RF | **96.82** | 99.68 | 82.11 | 91.98 | 51.94 | 83.89 | 55.19 | 76.04 | 36.30 | 79.43 | 34.34 |
| SC + pairwise CRF | 96.32 | 99.72 | **85.67** | **94.52** | 61.78 | **86.34** | 73.55 | 79.36 | 43.69 | **87.08** | 38.42 |
| SC + high-order CRF | 96.30 | **99.74** | 85.61 | 93.92 | **63.46** | 86.17 | **75.45** | 79.87 | **51.76** | 87.30 | **43.38** |

Table 6.8 $F_1$-score (%) of the classification result of the *Hong Kong (Sham Shui Po)* dataset compared with comparative trials.

| | *Façade* | *Ground* | *Vegetation* | *Car* | *Pedestrian* | *Guardrail* | *Traffic sign* | *Traffic light* | *Street lamp* | *Mussy points* | *Others* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SV + RF | 83.77 | 99.54 | 44.53 | 79.82 | 45.08 | 49.45 | 33.39 | 80.23 | 39.65 | 48.54 | 30.15 |
| SC + RF | 95.84 | 99.73 | 81.48 | 86.62 | 53.25 | 79.80 | 62.28 | 80.48 | 42.85 | 83.77 | **34.72** |
| SC + pairwise CRF | 96.11 | **99.74** | 84.65 | **88.73** | 61.37 | **82.06** | 71.13 | 83.07 | 48.45 | 91.17 | 31.58 |
| SC + high-order CRF | **96.18** | **99.74** | **84.81** | 88.53 | **62.28** | 81.98 | **72.04** | **83.39** | **52.99** | **91.51** | 32.89 |

Table 6.9 Overall accuracy (%) and mean F1-score (%) of the classification result of the *Hong Kong (Sham Shui Po)* dataset compared with comparative trials.

| | SV + RF | SC + RF | SC + pairwise RF | SC + high-order CRF |
|---|---|---|---|---|
| Overall accuracy | 86.47 | 93.33 | 95.68 | **95.79** |
| Mean $F_1$-score | 57.65 | 72.80 | 76.19 | **76.94** |

Table 6.6 shows that 10 of 11 categories have increased recall values with the introduction of structural information, and the increases are more than 10% for more than half of the categories. For *vegetation* and *guardrail* especially, the increases exceed 40%. A notable increase in precision values can also be found in Table 6.7; half of the categories have increases in precision of greater than 5%, and the precision of two categories (*traffic sign* and *mussy points*) are increased by more than 30%. Generally, the introduction of structural information can significantly increase both the completeness and correctness of classification, as indicated by the $F_1$-scores in Table 6.8, in which all categories have increased $F_1$-scores, and the $F_1$-scores in four categories (*vegetation*, *guardrail*, *traffic sign* and *mussy points*) are increased by 30% or more.

The introduction of pairwise contextual information could further improve the results of the second trial, as suggested by Table 6.6 ~ 6.9, and this is in agreement with many previous studies (Lim and Suter, 2009; Niemeyer et al., 2014; Zhu et al., 2017). After the high-order interactions being introduced, although for some categories the recall or precision values do not significantly improve or even slightly decline, a global increase is suggested by the highest overall accuracy and mean $F_1$-score of 95.79% and 76.94%, respectively. In general, $F_1$-scores increase for more than half of the categories. The greatest improvement can be found for the *street lamp*, for which the precision is increased by 8% and the $F_1$-score is increased by 4.5%. Most small, isolated components appearing among tree crowns or at façade edges are likely to be misclassified, as demonstrated in Figure 6.18 (c); this could not be corrected by pairwise interactions, but high-order interactions were able to take effect on such isolated fragments (as shown in Figure 6.18 (d)).

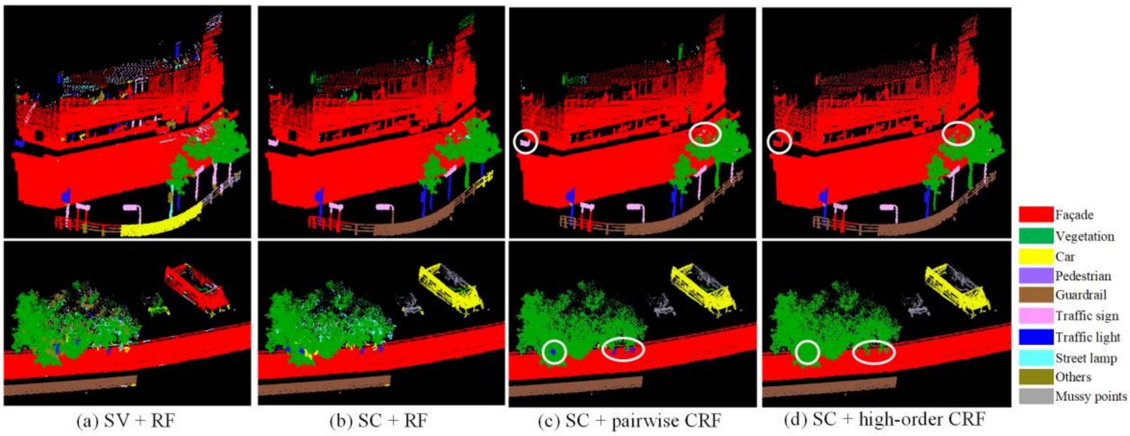



Figure 6.18 Classification results of two regions in different comparative trials (a) ~ (c) and the proposed classification framework (d). White circles in (c) and (d) highlight the misclassifications corrected by the regional label costs.

## 6.3 Experiment Using the *Hong Kong (Central)* Dataset

### 6.3.1 Data Description

The DIM point cloud acquired in Central, Hong Kong covers an area of 770 m × 900 m as shown in Figure 6.19. This dataset was generated using the software ContextCapture by Bentley (http://www.acute3d.com/contextcapture/) using oblique images, and it was resampled with a uniform spatial distance of 0.2 m, resulting more than 120 million points. As illustrated in Figure 6.19, most of this area is densely covered by high-rise buildings, including many landmark buildings (the buildings highlighted in yellow boxes in Figure 6.19), and only small parts of this area are covered by vegetation and other objects, such as cars and fences. The tallest building in this area, the International Finance Centre (IFC) building (Phase-2), has its top part missing (as shown in the red box in Figure 6.19), possibly because of low flight height and the lack of overlap between the oblique images.

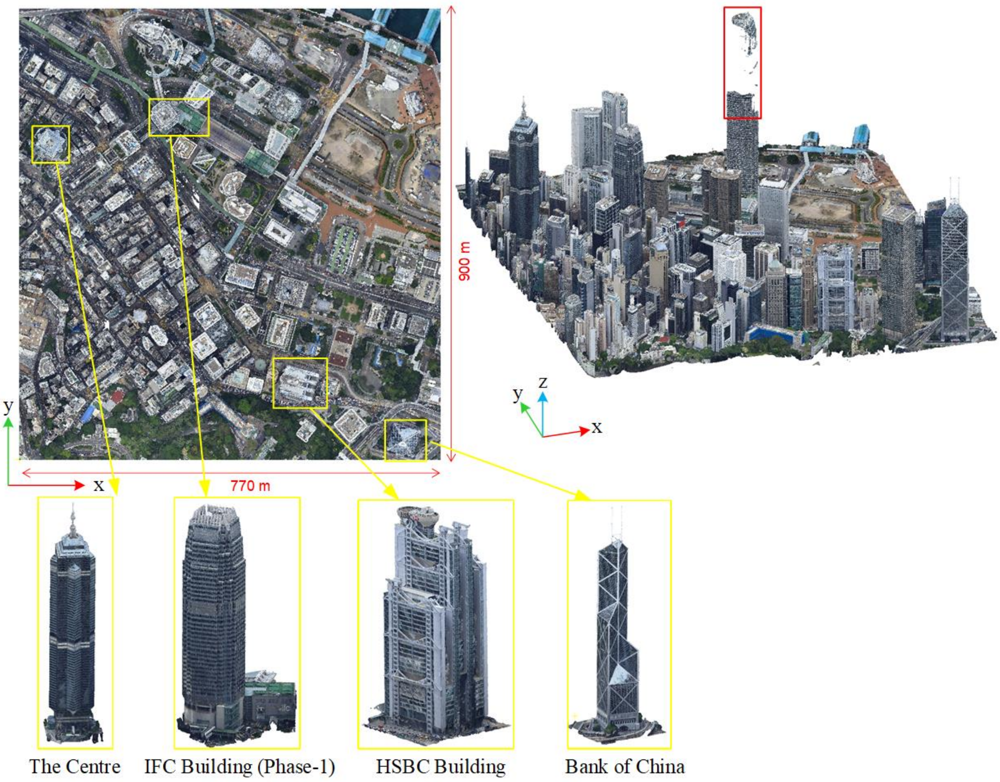



Figure 6.19 Overview of the *Hong Kong (Central)* dataset.

The ground and non-ground points of the *Hong Kong (Central)* dataset extracted by adaptive surface filtering are shown in Figure 6.20. As this point cloud is resampled with a uniform spatial distance, the ground and non-ground points all have even point densities.

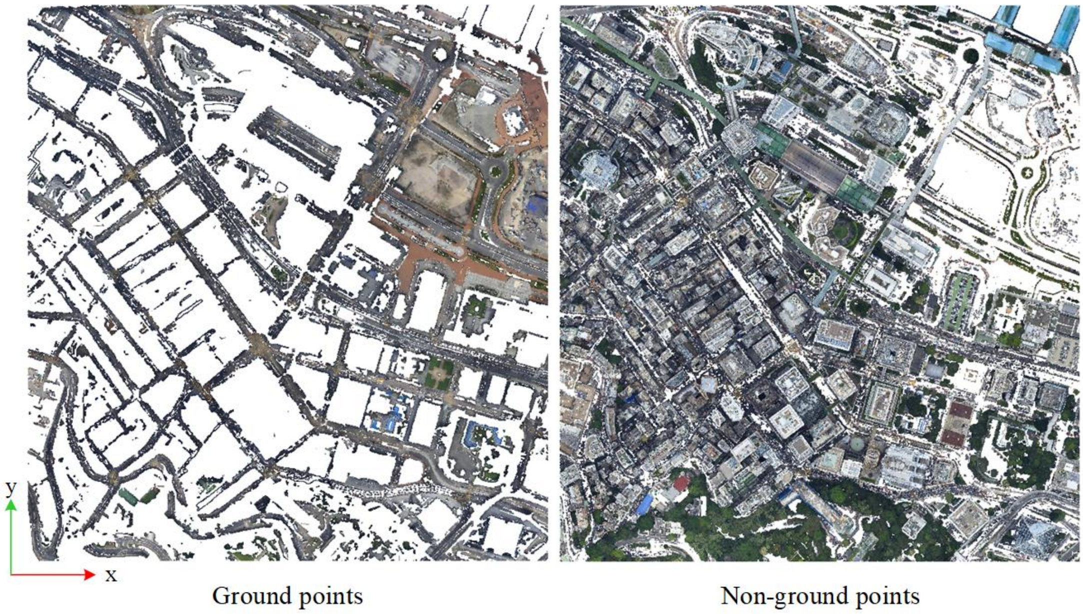



Figure 6.20 Ground and non-ground points extracted from the *Hong Kong (Central)* dataset.

### 6.3.2 Segmentation and Classification Results

In the *Hong Kong (Central)* dataset, the buildings are densely located (as shown in Figure 6.21 (a)) and sometimes have connecting structures between them (e.g., footbridges as shown in Figure 6.21 (b)). Vegetation and junctions between different objects are mostly presented by the DIM point cloud as smooth surfaces (as shown in Figure 6.21 (c)). Therefore, an over-segmentation of planar objects (as shown in Figure 6.21 (d) – (f)) was preferred for this dataset to obtain sufficient training examples for the classification. The extra constraint that $\theta(\boldsymbol{n}_{SV1}, \boldsymbol{n}_{SV2}) < 30°$ was adopted during the growing of structural components, which were labelled as *planarity*.

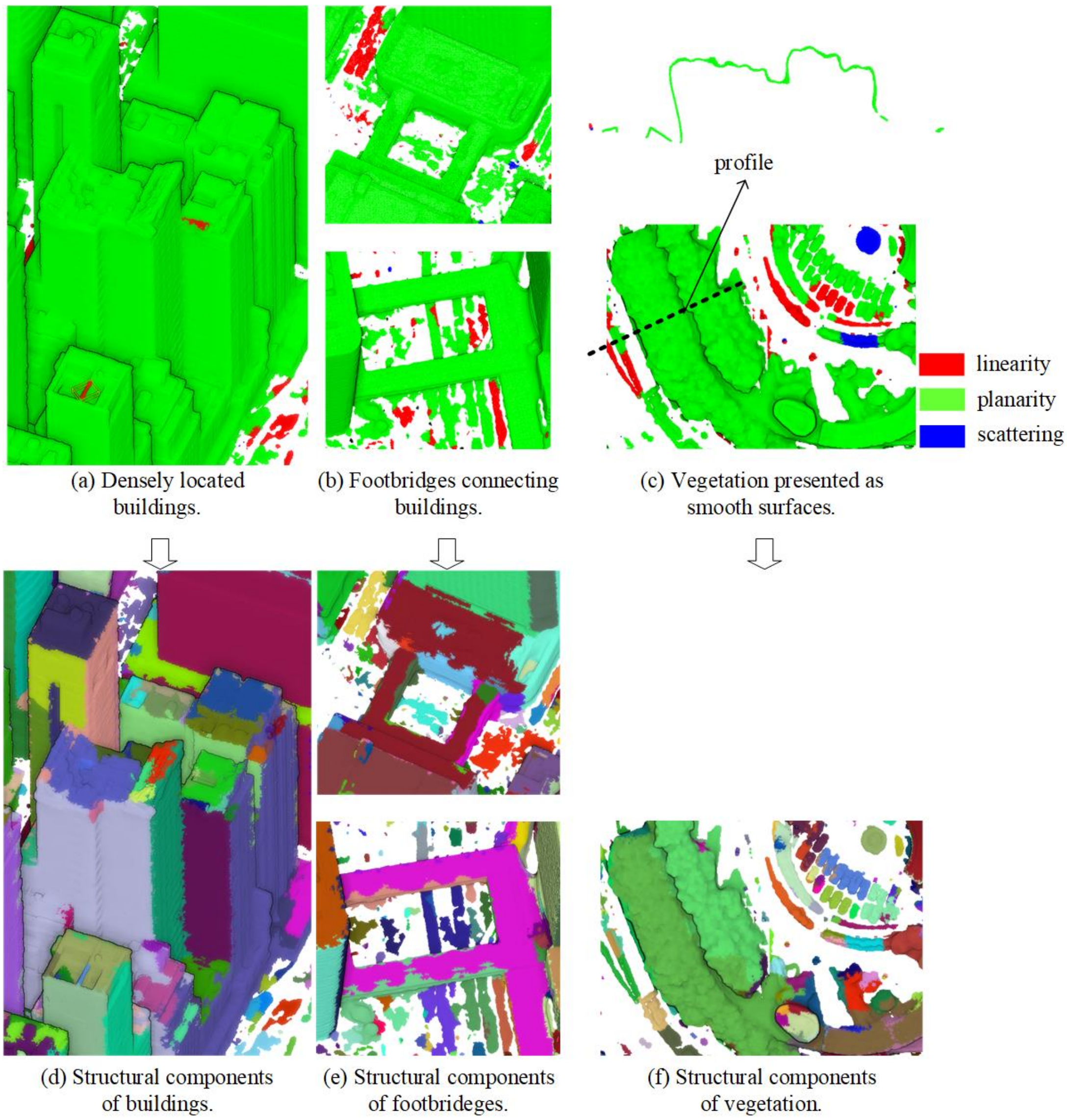



Figure 6.21 Characteristics of the *Hong Kong (Central)* dataset and structural components generated by over-segmentation.

For classification, the *Hong Kong (Central)* dataset was split into two areas: the testing area covering 2/3 of the entire area (as shown in Figure 6.22 (a)) and the training area covering the rest of the area (as shown in Figure 6.22 (b)). To train the RF classifiers, objects in the training area were manually labelled into four categories: including *building*, *vegetations*, *ground and others*. Footbridges were also grouped into the category of *building* according to the ground-truth obtained from a manually labelled ALS point cloud covering the same area, and *others* refers to small objects, such as cars and fences.

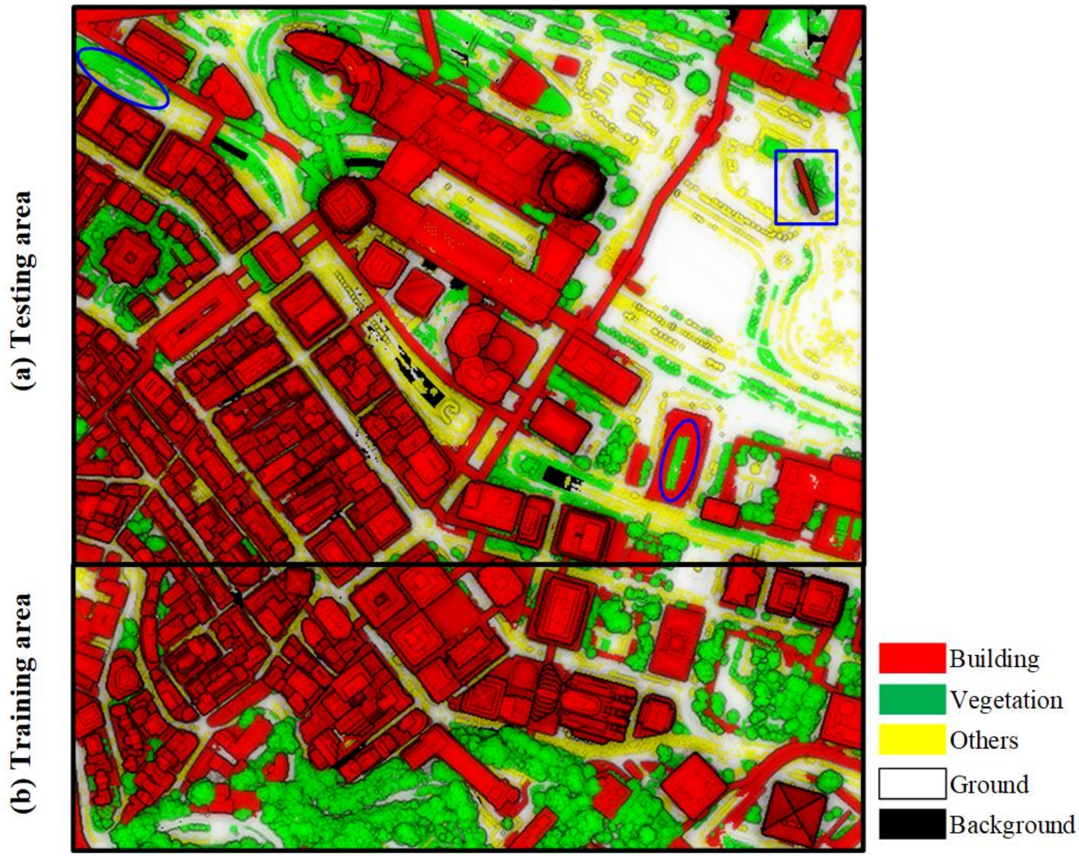



Figure 6.22 The training and testing areas in Central, Hong Kong, and the classification result of the *Hong Kong (Central)* dataset in 2D view.

Figure 6.23 shows the classification result in the testing area of the *Hong Kong (Central)* dataset in 3D view. From Figure 6.22 and Figure 6.23, it can be ascertained that most of the buildings (including footbridges) were correctly detected by the proposed method. There were no other objects with areas larger than 50 $m^2$ incorrectly detected as buildings, except a Ferris wheel, as shown in the blue boxes in Figure 6.22 (a) and Figure 6.23, perhaps due to the lack of training samples corresponding to the Ferris wheel. The main vegetation areas were also found by the proposed method, although some vegetation areas on the roofs of buildings or in the middle of roads (as shown in the blue circles in Figure 6.22 (a)) were incorrectly detected, perhaps due to the unexpected fluctuation in the surfaces of buildings or densely located cars.

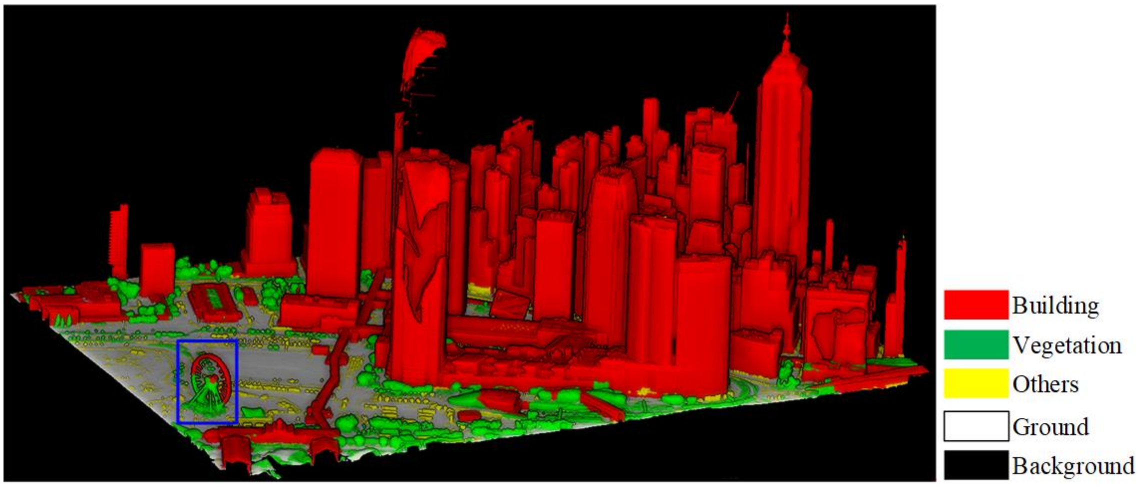



Figure 6.23 Classification result in the testing area of the *Hong Kong (Central)* dataset in 3D view.

As the classified point cloud is used to generate 3D building models, which is the final purpose of this research, the *building* class in was further quantitatively analysed by comparing it with the ground-truth obtained from a manually labelled ALS point cloud. Because most buildings in the ALS point cloud lack points on their façades, the comparison was conducted in 2D by projecting the points labelled as *building* into a grid of 0.85 m/pixel (which is about the resolution of the ALS point cloud), as shown in Figure 6.24. The true positive (TP), false positive (FP) and false negative (FN) detections are highlighted in yellow, red and blue, respectively. The quantitative evaluations of the classification results are shown in Table 6.10, in which *recall* = TP / (TP + FN), *precision* = TP / (TP + FP) and the $F_1$*-score* is computed as in Equation (6-4).

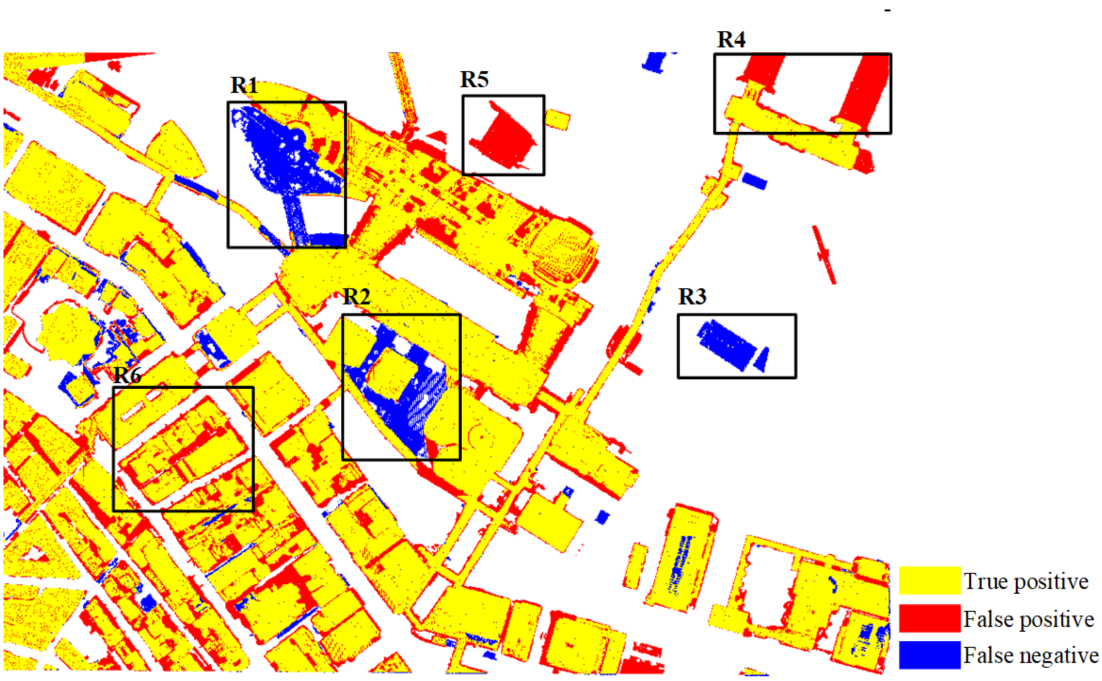



Figure 6.24 Visualised evaluation of the classification result in the testing area of the *Hong Kong (Central)* dataset.

Table 6.10 The evaluations (in percentage) of the classification result in the testing area of the *Hong Kong (Central)* dataset.

| ***Recall*** | ***Precision*** | ***$F_1$-score*** |
|---|---|---|
| 90.61 | 75.56 | 82.40 |

The recall value of 90.61% indicates the high completeness of the detected buildings. The main false negative detections appeared in the regions R1 ~ 3 as shown in Figure 6.24. Figure 6.25 (a) and (b) demonstrate the detailed comparisons in regions R1 and R2. The false negative detections in the DIM point cloud are mainly caused by points of low elevation, which are first recognised as ground points by the surface filter (Hu et al., 2014), with the corresponding points in the ALS labelled as *building*. Labelling these points as *ground* instead of *building* is more reasonable, because many cars (as shown in the black circle in Figure 6.25 (a)) are found above these points. Another reason for large false detections is that the images from which the DIM point cloud was generated and the ALS point cloud were acquired at different times, and changes might have taken place in the area. As shown in Figure 6.25 (c), the buildings detected in the ALS are missing in the DIM point cloud.

Figure 6.25 (d) shows the classification result in region R4 in detail. The buildings detected in the DIM point cloud are unclassified in the ALS, resulting in false positive detections in the comparison. Points appearing in the DIM but missing in the ALS can also lead to false positive detections, as shown in Figure 6.25 (e) and (f). The missing points are either caused by changes of objects (see Figure 6.25 (e)) or occlusions between high-rise buildings (see Figure 6.25 (f)). This latter reason is universal for nearly all of the buildings in the ALS data, resulting in a relatively lower precision value of 75.56%.

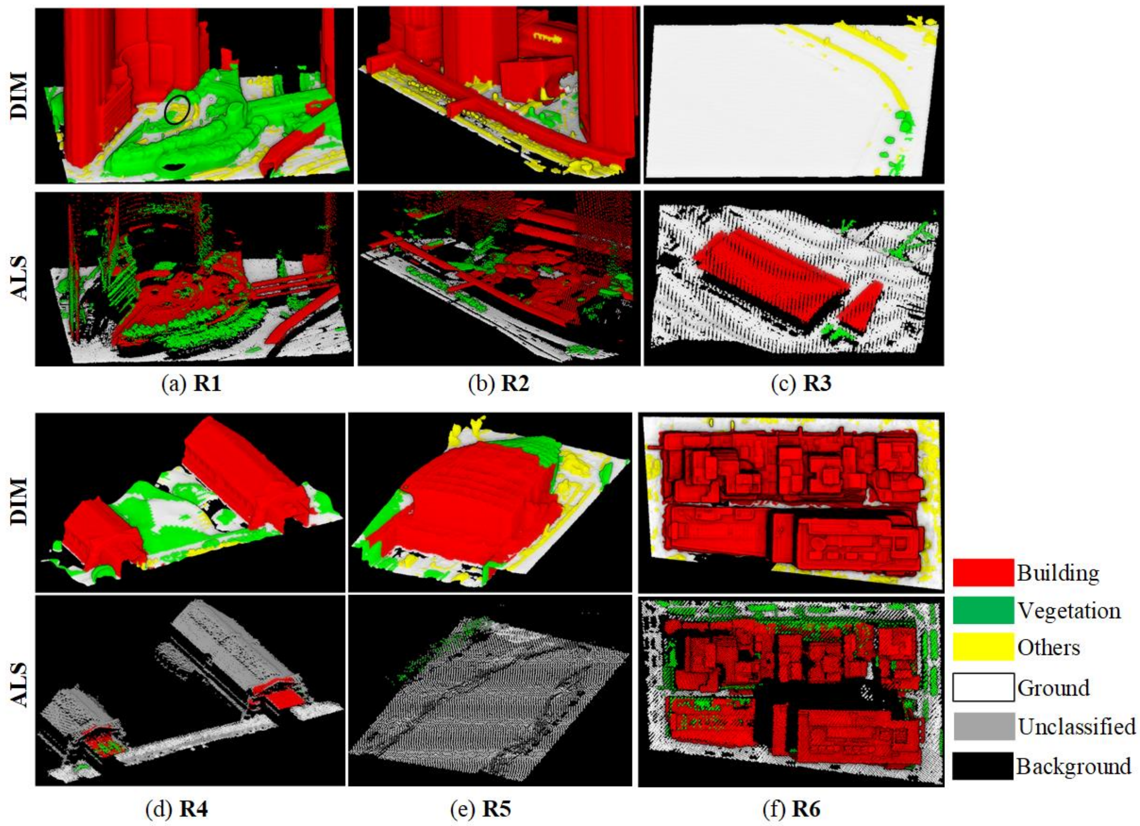
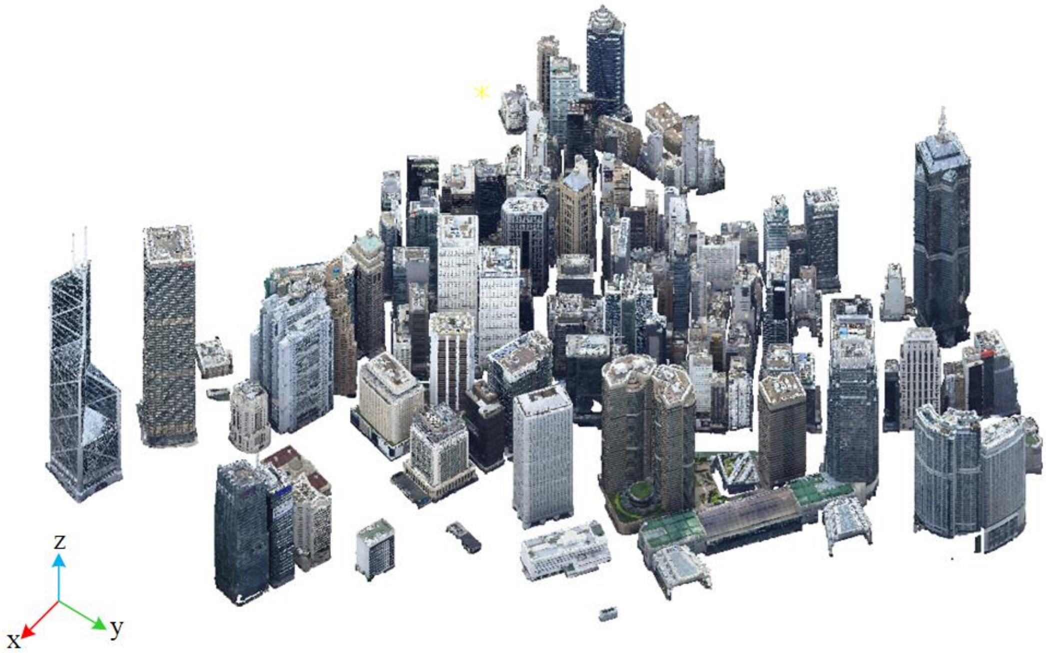



Figure 6.25 Detailed comparisons of the classified DIM (the *Hong Kong (Central)* dataset) and the ALS point clouds.

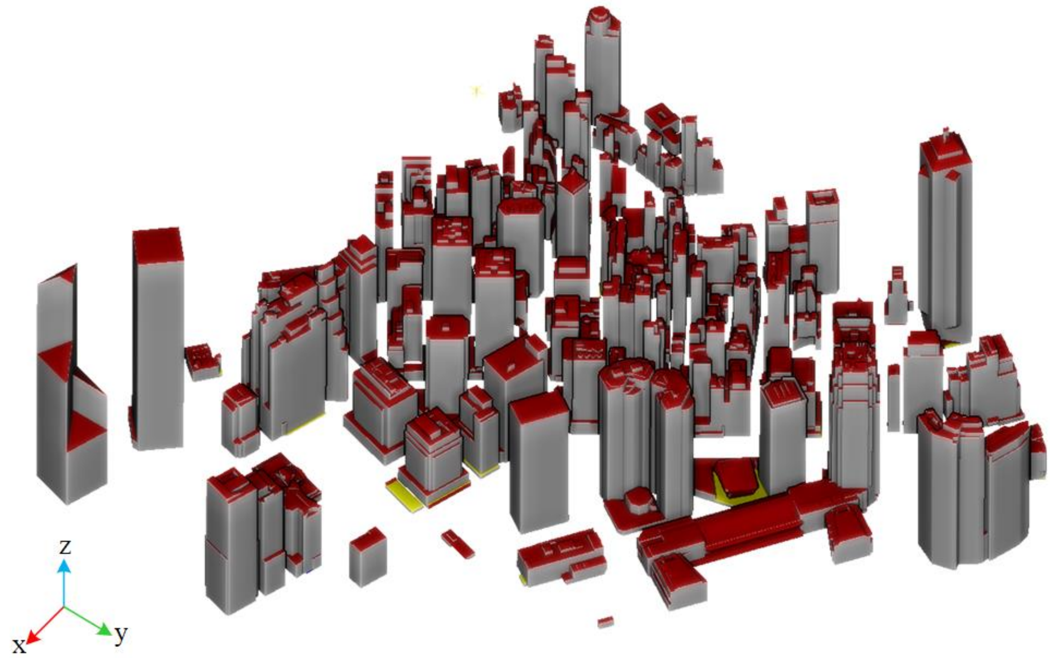
Misclassifications also exist at small or parts of objects, such as complex building structures with low elevations and fluctuant surfaces being classified as *vegetation* or *others*. However, in general, the classification result of the *Hong Kong (Central)* dataset by the proposed method is consistent with the manual labelling result of the ALS point cloud. Considering the inherent differences between the DIM and ALS point clouds, the comprehensive evaluation metric, the $F_1$-score, was of 82.40%, which suggests that the result for building detection is quite good, and it can be further applied in the reconstruction of 3D building models.

### 6.3.3 3D Building Reconstruction Results

After classification, points labelled as *building* were extracted as building point clusters, and in total, 105 buildings with complete structures (as shown in Figure 6.26 (a)) were selected by manual checking and refinement (footbridges and incomplete buildings, such as the IFC building (Phase-2) shown in Figure 6.19, were manually excluded). Figure 6.26 (b) shows the reconstructed 3D building models of the 105 buildings in the CityGML format, where red, grey and yellow indicate roof, wall and outer-floor surfaces, respectively.




(a) Point clouds of 105 buildings in the Central, Hong Kong.




(b) 3D building models in CityGML format.

Figure 6.26 Overview of the 3D building reconstruction result of the *Hong Kong (Central)* dataset. Red, grey and yellow indicate roof, wall and outer-floor surfaces, respectively.

Figure 6.27 shows the detailed reconstruction results for four challenging types of buildings: buildings with complex structures, buildings with missing data, buildings with curved surfaces and buildings with true 3D structures.

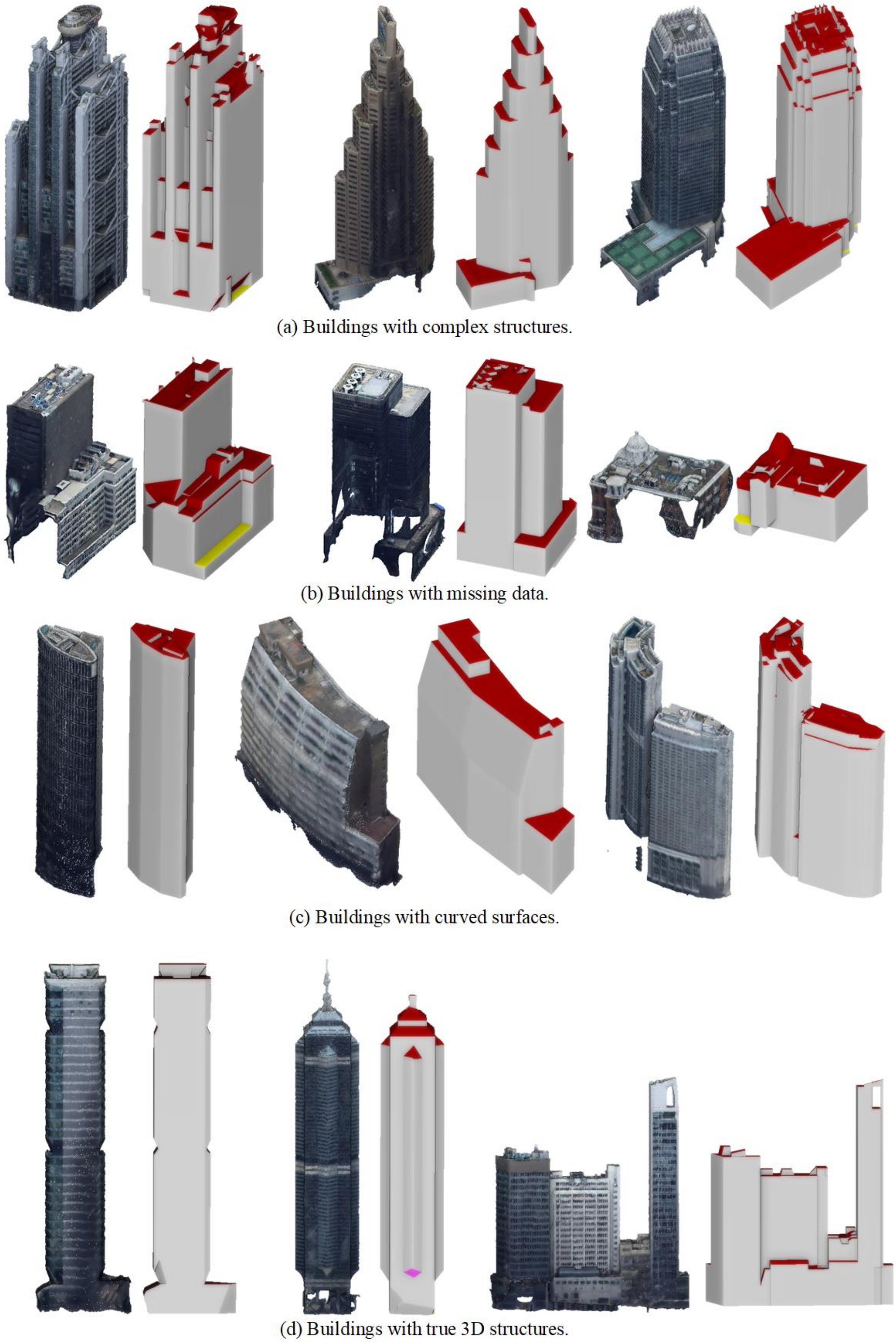



Figure 6.27 Reconstruction results for four challenging types of buildings in CityGML format. Red, grey, yellow and pink indicate roof, wall, outer-floor and outer-ceiling surfaces, respectively.

Taking Figure 6.26 and Figure 6.27 together, it can be qualitatively concluded that the proposed reconstruction method had outstanding performance for modelling buildings with various architectural styles. Even with the complexity of buildings and poor-quality input data, the proposed method produced satisfactory building models.

Figure 6.28 and Figure 6.29 show the quantitative evaluations of the reconstruction results of several buildings, which are also compared with the results of PolyFit (Nan and Wonka, 2017) and 2.5D (D-C) (Zhou and Neumann, 2010). For buildings with simple structures (as shown in Figure 6.28), both PolyFit and the proposed method can produce polygonal models with high regularity, whereas 2.5D (D-C) produced models in the triangular mesh format and with irregular boundaries (see the magnified views showing the details of the reconstructed building rooftops in Figure 6.28). The proposed method had the smallest *RSME* values, 0.67 m and 0.58 m, for two simple buildings, indicating higher modelling accuracy than PolyFit and 2.5D (D-C). The main inaccuracies of the models generated by the proposed method occurred at linear and small structures on the building roofs, which were eliminated during the extraction of the planar primitives of the buildings. Points carved into the building, which were generated by mismatching during the MVS pipeline, could also cause inaccuracies on the building façades (such as *Building*-1 shown in Figure 6.28); this was consistent with the results of PolyFit and 2.5D (D-C).

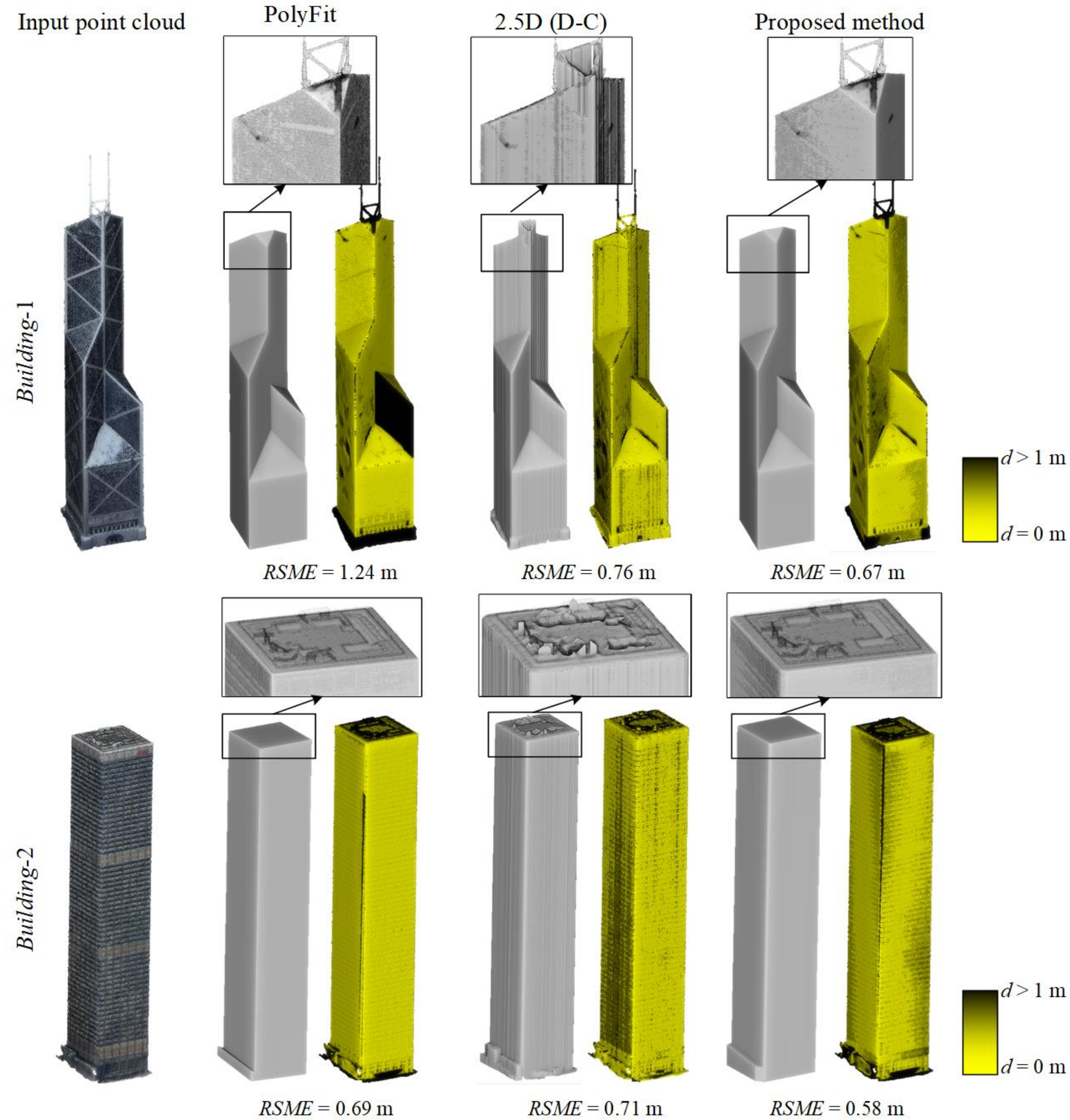



Figure 6.28 Evaluations of the reconstruction results of two simple buildings and comparison with the results of PolyFit and 2.5D (D-C). The details of the reconstructed building rooftops are shown in the black rectangles, where the input point clouds are overlapped on the building models.

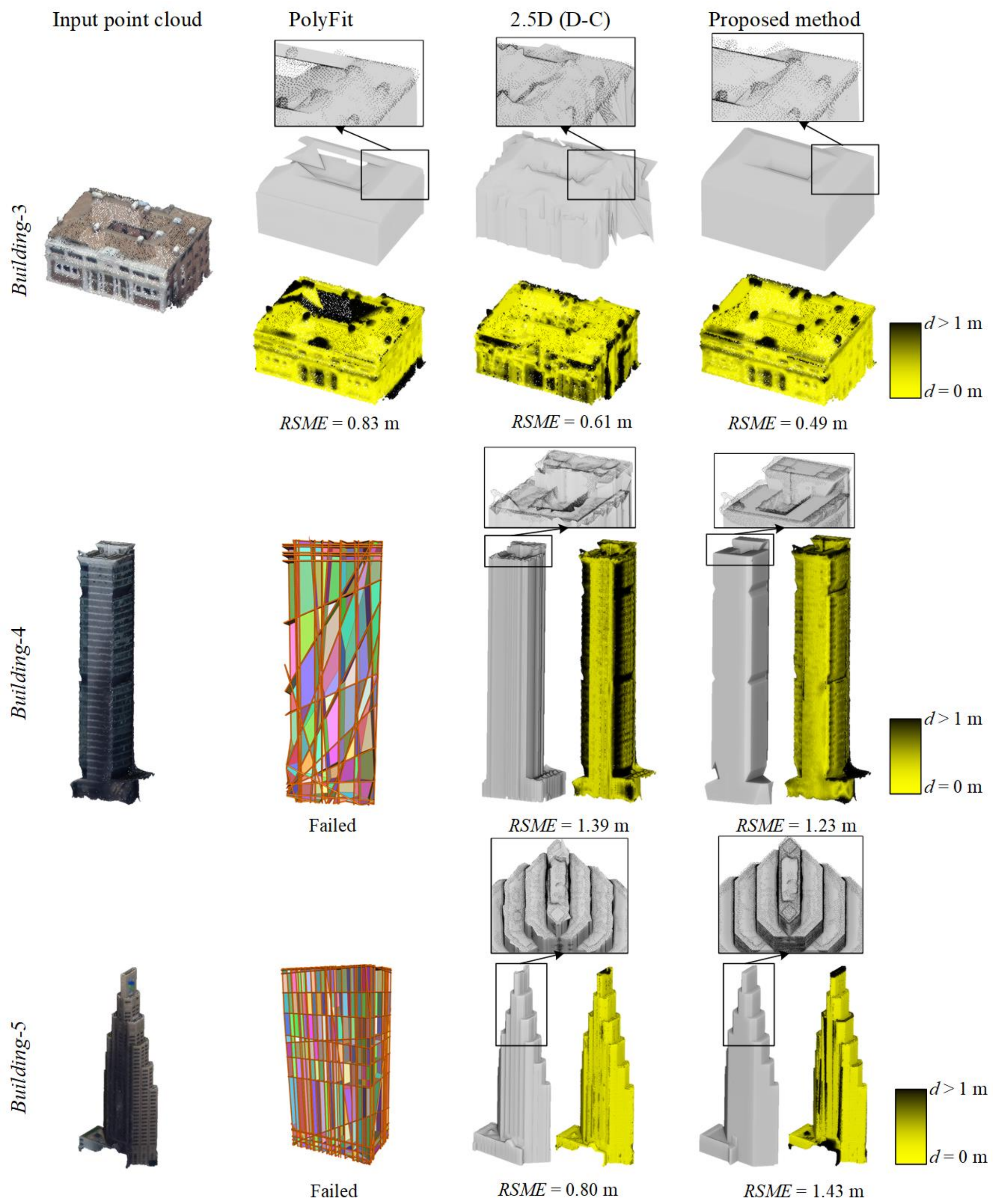



Figure 6.29 Evaluations of the reconstruction results of three complex buildings and comparison with the results of PolyFit and 2.5D (D-C). The details of the reconstructed building rooftops are shown in the black rectangles, where the input point clouds are overlapped on the building models.

Although PolyFit showed competitive performance in reconstructing simple buildings with high regularity, it tended to generate extremely inaccurate models for buildings with complex structures (such as *Building*-3 shown in Figure 6.29) or directly failed to output the building models (such as *Building*-4 and 5 shown in Figure 6.29). In contrast, both the proposed method and 2.5D (D-C) showed high robustness in the reconstruction of buildings with various complex architectural styles. However, again, 2.5D (D-C) only produced triangular mesh models with highly irregular boundaries, mostly with relatively lower geometric accuracy compared with the proposed method (as indicated by the *RSME* measurements of *Building*-1~4 shown in Figure 6.28 and Figure 6.29). With respect to the modelling of small structures on the building rooftops, 2.5D (D-C) had better performance than the proposed method (as illustrated by *Building*-3 in Figure 6.29), because of the failure to present such small structures with planar primitives. However, 2.5D (D-C) only focuses on the reconstruction of non-vertical structures (e.g., building rooftops), whereas the proposed method considers the structures of entire buildings and has better performance in modelling building façades (as illustrated by *Building*-4 in Figure 6.29).

In general, both the proposed method and 2.5D (D-C) have higher robustness than PolyFit, and the proposed method can generate polygonal models with high regularity and true 3D structures, which is beyond the capability of 2.5D (D-C). The proposed method can also convert the geometric models into CityGML formats for further urban applications.

However, compared with manually generated 3D models, the 3D building models generated by the proposed approach may be unsatisfactory with respect to details. This may be because the 3D cells forming the geometries of the building models are generated based on 3D planes, and planes cannot present details perfectly. In fact, this is a common issue with reconstruction methods based on planar segments. The 3D modelling results are also limited by the quality of the input point clouds, which also present details in an undesirable way. In general, the proposed method has outstanding performance considering the limited quality of the input data.

# Chapter 7 Conclusions and Discussion

This dissertation presents a complete framework of point cloud modelling. Three innovative methods – segmentation, classification and 3D building reconstruction – and systematic experimental analyses are described in detail in the previous chapters. This chapter summarises the achievements, draws conclusions from this research and then makes recommendations for future research.

## 7.1 Summary of the Research Work

In this research, three mutually connected methods based on multiple relations were developed for modelling point clouds that may have serious defects (e.g., varying point density, noise and missing data).

First, a segmentation method of point clouds based on multi-level relations was introduced. During segmentation, supervoxels with adaptive sizes were generated, and the local shape descriptors were derived from the supervoxels for structural labelling, providing essential clues for further decomposition of objects into different components with meaningful labels.

After the segmentation, structural components were classified based on the contextual relations between them. The contextual relations presenting the structural information were encoded in a CRF framework and affected both the training/inference and the refinement stages of the classification. High-order contextual information was also introduced into the CRF for refining regional label redundancies.

With the classified point clouds, building points are easily extracted and clustered. For each individual building, a true 3D model in the CityGML format can be automatically generated by the developed 3D reconstruction method, which benefits from the use of topological-relation constraints. This method adopts a space-partition-and-approximation strategy and abstracts the topological relations between the space elements as constraints of an ILP problem, for which mature solutions have already been found.

Based on the proposed methods, a package of tools for point cloud modelling was developed in C++ using Microsoft Visual Studio 2017. This tool package consists of four modules: a basic module for data management and three functional modules corresponding to the three proposed methods. During the development of the point cloud modelling tools, several open source libraries, including PCL (Rusu and Cousins, 2011), OpenCV (Bradski and Kaehler, 2000), CGAL (Fabri and Teillaud, 2011) and Gurobi solver (Gurobi, 2015), were used to solve some specific problems, as shown in Figure 7.1.

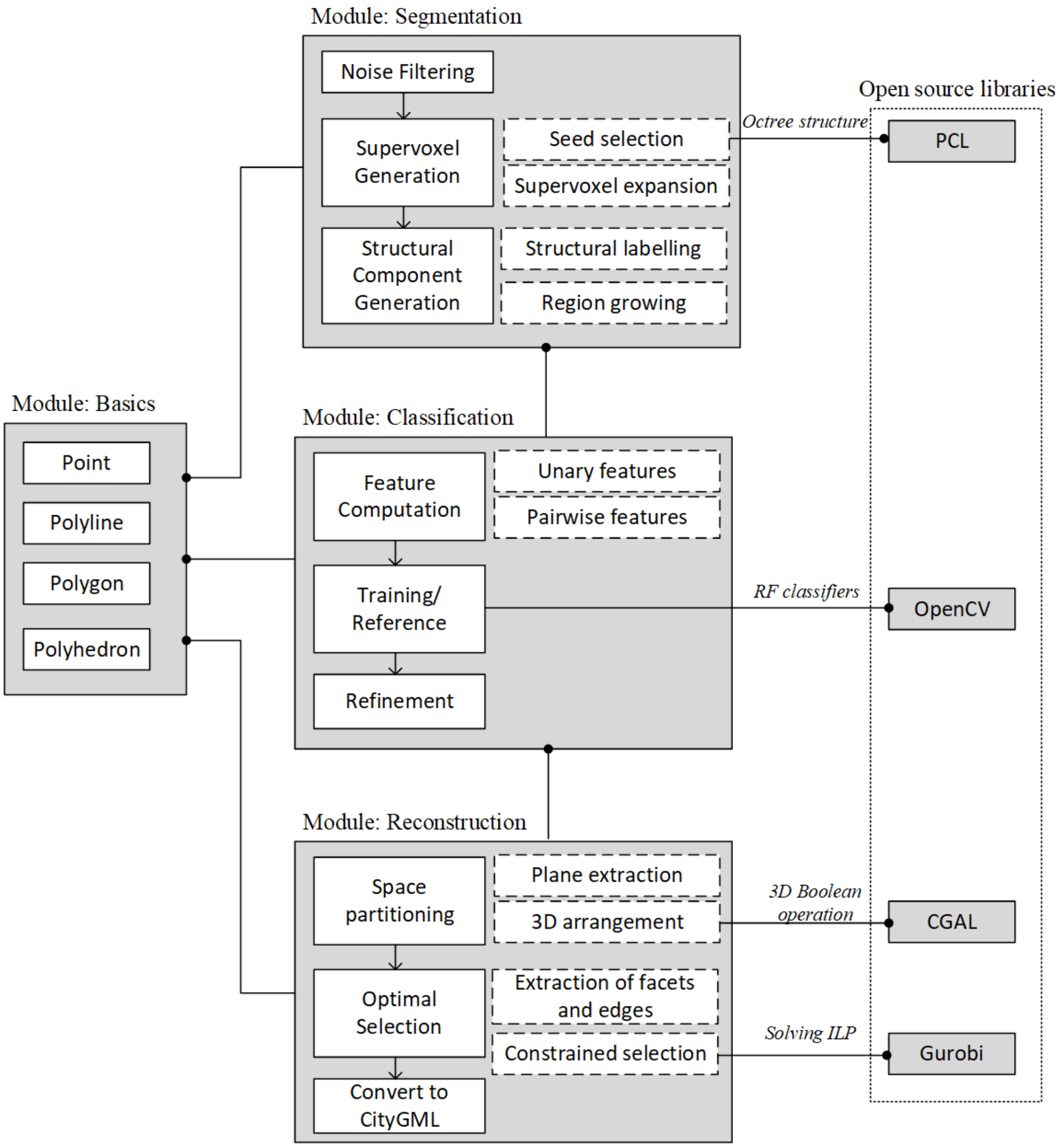
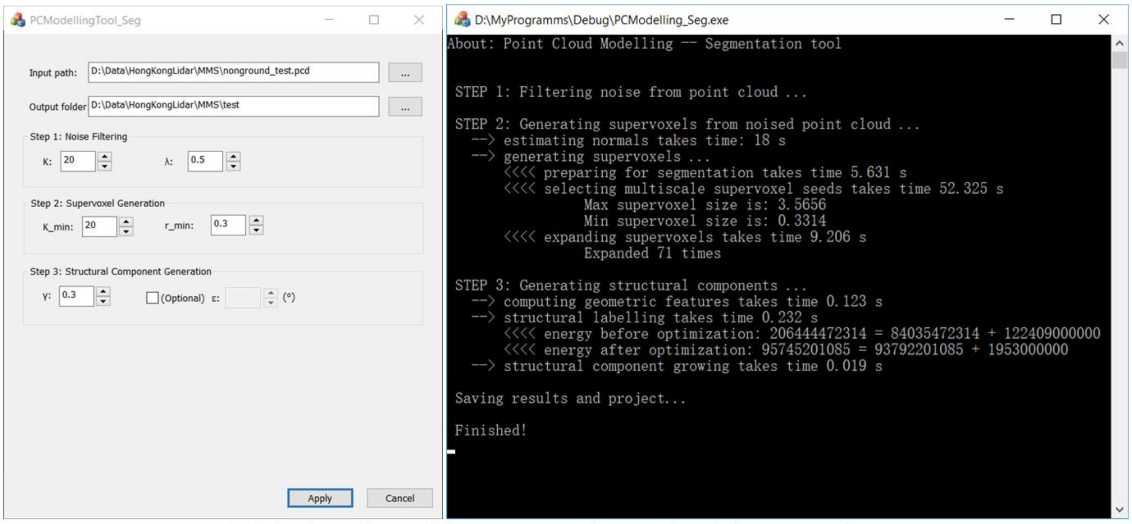
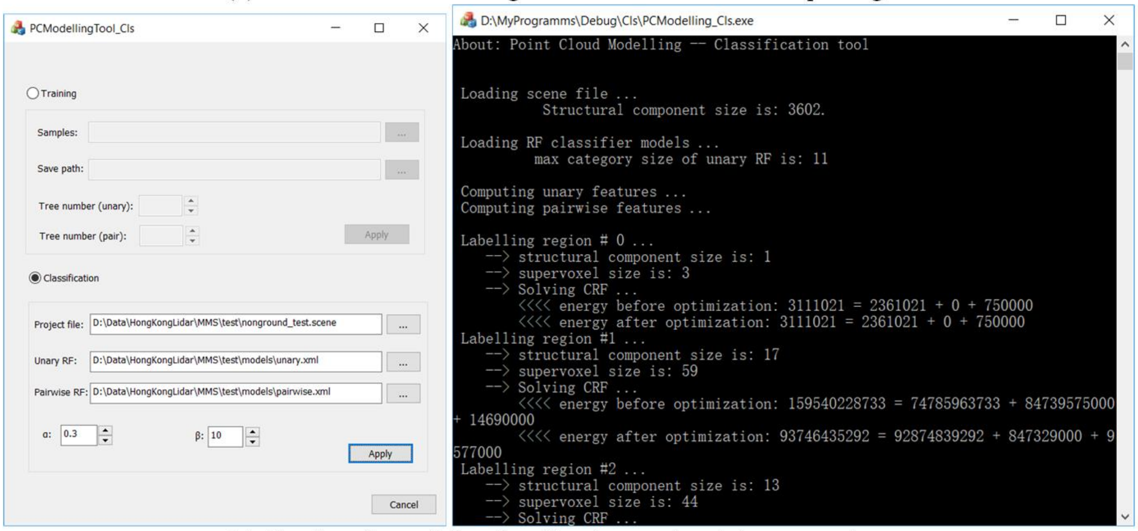



Figure 7.1 Module design of the point cloud modelling tools developed in this research.

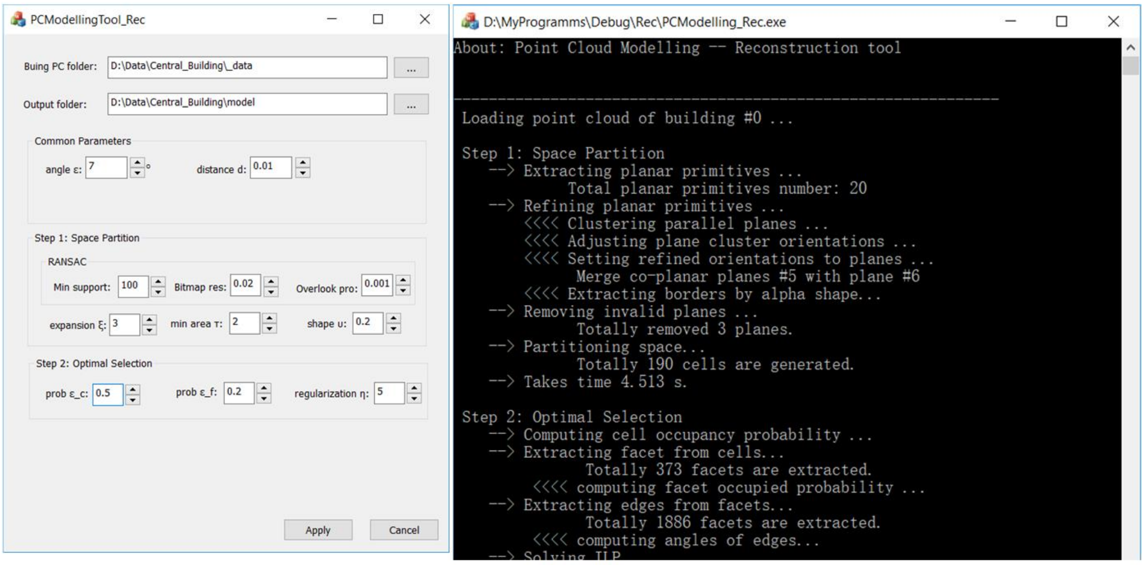
The interfaces of the developed tools are shown in Figure 7-2, and each of the tools is accompanied by a console to show the processing log.




(a) The interface of the segmentation tool and the output log.




(b) The interface of the classification tool and the output log.




(c) The interface of the reconstruction tool and the output log.

Figure 7.2 Interfaces of the developed tools and their output logs.

With these developed tools, three systematic experiments were carried out to investigate the performances of the proposed methods using datasets acquired from different sources, including an MLS point cloud (in Paris), a point cloud obtained from MMS (in Sham Shui Po, Hong Kong) and a DIM point cloud (in Central, Hong Kong). Qualitative and quantitative evaluations were made based on the experimental results of these three datasets, and the results were compared with previous methods.

## 7.2 Conclusions

The main novelty of this study lies in the introduction and combination of multiple relations, including geometric relations, structural relations and topological relations, during the process of object decomposition, recognition and 3D building reconstruction. The following summarises the effectiveness of the multiple relations during each stage of the modelling of point clouds, and the results of the experiments.

(1) Density anisotropy is important during noise filtering, especially for ground-obtained point clouds, in which there are large variations in the point densities. For such point clouds, multiple-level relations can help to obtain a better segmentation of objects in an efficient way. The multi-size supervoxels generated based on the point relations can significantly reduce the computational cost in the further segmentation step, while well preserving the local characteristics of the data. In addition, the relations at the supervoxel level make the final segmentation spatially smooth and reasonable. The segmentation produces both supervoxels and structural components, and this enables a flexible way at different scales for interactively modifying the errors in the segmentation, which may have influences on the following classification and reconstruction.

(2) Objects are formed by structures of different shapes, and such structural information is essential for the recognition of objects, especially objects presented in detail. Capturing structural information as contextual relations

and abstracting them as discriminative features at the inference stage of the classification, can be more effective than applying them only during the refinement stage of the classification. With structural information, multiple classes of objects, even with local or global similarities, will be correctly recognised.

(3) By adopting a space-partition-and-approximation strategy, complex topological computations or modifications can be avoided, and the topological relations can be easily obtained from the space units obtained from the space partition. This strategy also provides an opportunity to use the topological relations to constrain the geometric model to be topologically correct. Extra constraints in terms of regularity can also be attached to the topological relations so that the final building model can be consistent with some architectural design principles.

(4) In the experiments, the ground-obtained datasets feature greatly varying point density and noise. The quantitative evaluation and comparison with previous studies and comparative trials indicated that the proposed method performed best in the identification of multiple categories. The results also suggested that the introduction of structural information could significantly improve the completeness and accuracy of classification. The effectiveness of the 3D building reconstruction method was verified using the DIM point cloud, from which relatively more complete buildings could be extracted. Compared with two typical previous methods, the proposed method showed outstanding performance and high robustness for the reconstruction of various buildings, even those have serious data missing issues, which are probably caused by occlusions.

(5) In general, the segmentation, classification and building modelling pipeline proposed in this research provides a practicable solution for 3D city reconstruction with high automation and efficiency. The adoption of

supervoxels instead of points can significantly reduce the computation load of the graph-based segmentation, thus making the segmentation more efficient. The segmentation also provides a more flexible way to obtain the training samples for classification. Because each of the structural components is designed only to correspond to one object, users can easily click on the structural component and choose an appropriate class label for it during the production of training samples. With respect to the 3D reconstruction of buildings, although some manual interventions were involved to remove extremely incomplete buildings and footbridges in the experiments in this research, the reconstruction of individual buildings is totally automatic and does not require reduplicative parameter tuning. In addition, the half space partition strategy designed in this research significantly reduces the size of the 3D cell complex, while preserving the completeness of the space partition, thus contributing to the efficiency of the subsequent optimal selection of occupied cells.

## 7.3 Discussion and Future Works

Following the conclusions drawn from the proposed methods and the experimental analyses, recommendations for future research are presented below.

**(1) Point cloud segmentation and classification**

As the initial inference in the classification is largely based on structural information about objects, any changes, such as close distance and unexpected appurtenances, may lead to misclassifications. Fragmentation and missing data that break down the point density consistency may also make the high-order interactions useless during the refinement stage of the classification. Although the DIM point cloud suffers much less from fragmentation and missing data, the smooth planarised presentation of objects and inaccuracy of data are serious defects that may limit the extraction of structural information.

These challenges are commonly faced by many segmentation and classification methods that use only the geometric properties of the point cloud. In future research, new methods will be investigated to solve these problems based on this research, such as exploiting the properties of full-waveform LiDAR or fusing spectral images with the point clouds.

With the development of convolutional networks, e.g., the 3D convolutional neural networks (Huang and You, 2016), graph convolutional networks (Landrieu and Simonovsky, 2018; Qi et al., 2017) and interpolated convolutional networks (Mao et al., 2019), deep learning becomes more and more popular in the classification of point cloud data. In fact, the framework developed in this research allows the flexible selection of classifiers for the inference, including the graph convolution networks. In future research, the performance of the proposed framework will be further investigated with the state-of-the-art deep learning algorithms.

**(2) 3D reconstruction of buildings**

Although mature plane detection algorithms such as RANSAC can produce satisfactory results for the majority of buildings, small structures presented by a very few points are likely to be omitted. In addition to planar structures, buildings can also consist of linear structures or curved surfaces. Although large non-planar components can be approximated by multiple planes, the approximation as a whole tends to have less regularity and accuracy.

Occlusion caused by densely located objects or complex terrain relief, is another problem for the generation of accurate 3D models with fine details. Although the proposed 3D reconstruction approach is able to produce complete building models from point clouds that have serious data missing issues, details cannot be recovered because they are not captured in the images due to occlusions. The most effective way to solve this problem is to introduce more data from different sources, such as the mobile and backpack mapping systems, which are flexible enough to mapping zones that are blind to the aerial images.

In the future, efforts will be made to investigate the generation of more accurate models with higher LODs using data from multiple sources. In addition to the polygonal models generated by the method proposed in this research, the non-planar structures will be presented as triangular meshes, and the final building models can therefore output hybrid presentations as in the study of Hu et al. (2018a). Efforts will also be made to investigate the benchmark of 3D building models, to find the most appropriate way to evaluate the 3D building models, with respect to not only geometry, but also semantics.

**(3) Enrich the model components in CityGML**

As the models will have more details and be presented in a hybrid format, a new automatic method needs to be developed to automatically convert the new models into CityGML format. In addition to the geometries, the enrichment of the topological, and semantical components in CityGML will also be investigated in the future, so that the models can be used to support various 3D GIS analysis functions.

Furthermore, the semantic and topological information is investigated only for independent buildings in this research. In fact, in urban areas, especially in metropolises like Hong Kong, buildings are often connected by, e.g., footbridges, and this is an important part of the spatial infrastructure for smart city. Therefore, further efforts will be made to analyze the connectivity rules between complex building shapes and investigate the methods to embed them in the CityGML models.

**(4) Generation of virtual reality models**

Besides the generation of building models with higher LODs and detailed semantic information, texture mapping is another essential step for the generation of virtual reality city models (Buyukdemircioglu et al., 2018; Lee and Yang, 2019). 3D models generated from photogrammetric point clouds by dense image matching inherently take advantage of the projection relationships from the 2D images to the

3D space. Co-registration and fusion technologies of LiDAR and image data (Poliyapram et al., 2019) also make it possible to achieve texture mapping for 3D models generated from LiDAR point clouds.

However, generation of virtual reality models is still facing many challenges, especially for building models with complex structure and high level of details, such as the selection of optimal images for texture mapping with respect to both visual angles and occlusions, and the uniform of tones between different images. Based on the modelling result of this research, further study will be conducted in the future to study the efficient and effective methods for seamless, consistent and photorealistic texture mapping.

# References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 2274-2282.

Aijazi, A.K., Checchin, P., Trassoudaine, L., 2013. Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation. Remote Sensing 5, 1624-1650.

Ameri, B., Fritsch, D., 2000. Automatic 3D building reconstruction using plane-roof structures. American Society for Photogrammetry and Remote Sensing (ASPRS) Washington DC, 47.

Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., Maierhofer, S., 2013. O-snap: Optimization-based snapping for modeling architecture. ACM Transactions on Graphics (TOG) 32, 6.

Babahajiani, P., Fan, L., Kamarainen, J., Gabbouj, M., 2015. Automated super-voxel based features classification of urban environments by integrating 3D point cloud and image content, Signal and Image Processing Applications (ICSIPA), 2015 IEEE International Conference on. IEEE, pp. 372-377.

Blaha, M., Vogel, C., Richard, A., Wegner, J.D., Pock, T., Schindler, K., 2016. Large-scale semantic 3d reconstruction: an adaptive multi-resolution model for multi-class volumetric labeling, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3176-3184.

Boulch, A., de La Gorce, M., Marlet, R., 2014. Piecewise‐planar 3D reconstruction with edge and corner regularization, Computer Graphics Forum. Wiley Online Library, pp. 55-64.

Boulch, A., Guerry, J., Le Saux, B., Audebert, N., 2018. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. Computer & Graphics, 71: 189-198.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1124-1137.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 1222-1239.

Bradski, G., Kaehler, A., 2000. OpenCV. Dr. Dobb's journal of software tools, 3.

Brenner, C., Haala, N., 1998. Fast production of virtual reality city models. International Archives of Photogrammetry and Remote Sensing 32, 77-84.

Buyukdemircioglu, M., Kocaman, S., Isikdag, U., 2018. Semi-automatic 3D city model generation from large-format aerial images. ISPRS International Journal of Geo-Information 7(9): 339-359.

Cao, Y., Zhao, H., Li, N., Wei, H., 2011. Land-cover classification by airborne LIDAR data fused with aerial optical images, Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM), 2011 International Workshop on. IEEE, pp. 1-6.

Chehata, N., Guo, L., Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 38, W8.

Chen, D., Wang, R., Peethambaran, J., 2017. Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. IEEE Transactions on Geoscience and Remote Sensing 55, 7032-7052.

Chen, D., Zhang, L., Mathiopoulos, P.T., Huang, X., 2014. A methodology for automated segmentation and reconstruction of urban 3-D buildings from ALS point clouds. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7, 4199-4217.

Cocchia, A., 2014. Smart and digital city: A systematic literature review, Smart city. Springer, Cham, pp. 13-43.

Comaniciu, D., Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 603-619.

Coughlan, J.M., Yuille, A.L., 2001. The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference, Advances in Neural Information Processing Systems, pp. 845-851.

Delong, A., Osokin, A., Isack, H.N., Boykov, Y., 2012. Fast approximate energy minimization with label costs. International Journal of Computer Vision 96, 1-27.

Demantke, J., Mallet, C., David, N., Vallet, B., 2011. Dimensionality based scale selection in 3D lidar point clouds. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38, W12.

Dong, Z., Yang, B., Hu, P., Scherer, S., 2018. An efficient global energy optimization approach for robust 3D plane segmentation of point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 137, 112-133.

Dorninger, P., Pfeifer, N., 2008. A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. Sensors 8, 7323-7343.

Douglas, D.H., Peucker, T.K., 1973. 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 10, 112-122.

Du, S., Zhang, Y., Zou, Z., Xu, S., He, X., Chen, S., 2017. Automatic building extraction from LiDAR data fusion of point and grid-based features. ISPRS Journal of Photogrammetry and Remote Sensing 130, 294-307.

Elberink, S.O., 2009. Target graph matching for building reconstruction. Proc. Lserscanning, 9: 49-54.

Elberink, S.O., Vosselman, G., 2009. Building reconstruction by target based graph matching on incomplete laser data: analysis and limitations. Sensors 9, 6101.

Engelmann, F., Kontogianni, T., Hermans, A., Leibe, B., 2017. Exploring spatial context for 3d semantic segmentation of point clouds, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 716-724.

Fabri, A., Teillaud, M., 2011. Cgal-the computational geometry algorithms library, 10e Colloque National en Calcul des Structures, p. 6.

Felzenszwalb, P.F., Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. International Journal of Computer Vision 59, 167-181.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381-395.

Früh, C., Zakhor, A., 2003. Constructing 3D City Models by Merging Aerial and Ground Views. IEEE Computer Graphics and Applications 23, 52-61.

Furukawa, Y., Ponce, J., 2010. Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1362-1376.

Ge, X., Wu, B., Li, Y., Hu, H., 2019. A multi-primitive-based hierarchical optimal approach for semantic labeling of ALS point clouds. Remote Sensing, 11(10): 1243.

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

Gislason, P.O., Benediktsson, J.A., Sveinsson, J.R., 2006. Random forests for land cover classification. Pattern Recognition Letters 27, 294-300.

Golovinskiy, A., Funkhouser, T., 2009. Min-cut based segmentation of point clouds, 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. IEEE, pp. 39-46.

Grilli, E., Menna, F., Remondino, F., 2017. A review of point clouds segmentation and classification algorithms. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 42, 339.

Gröger, G., Plümer, L., 2012. CityGML–Interoperable semantic 3D city models.

ISPRS Journal of Photogrammetry and Remote Sensing 71, 12-33.

Gross, H., Thoennessen, U., Hansen, W.v., 2005. 3D modeling of urban structures. International Archives of Photogrammetry and Remote Sensing 36, W24.

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

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

Gurobi, 2015. Gurobi optimization < https://www.gurobi.com/ >.

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

Hackel, T., Wegner, J.D., Schindler, K., 2016. Fasst Semantic Segmentation of 3D point clouds with strongly varying density. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences 3.

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

Hough, P.V., 1962. Method and means for recognizing complex patterns. Google Patents.

Hu, H., Ding, Y., Zhu, Q., Wu, B., Lin, H., Du, Z., Zhang, Y., Zhang, Y., 2014. An adaptive surface filter for airborne laser scanning point clouds by means of regularization and bending energy. ISPRS Journal of Photogrammetry and Remote Sensing 92, 98-111.

Hu, J., You, S., Neumann, U., 2003. Approaches to large-scale urban modeling. IEEE Computer Graphics and Applications 23, 62-69.

Hu, P., Dong, Z., Yuan, P., Liang, F., Yang, B., 2018a. Reconstruction of 3D models from point clouds with hybrid presentation. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 42.

Hu, P., Yang, B., Dong, Z., Yuan, P., Huang, R., Fan, H., Sun, X., 2018b. Towards Reconstructing 3D Buildings from ALS Data Based on Gestalt Laws. Remote Sensing 10, 1127.

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

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

Jarząbek-Rychard, M., Borkowski, A., 2016. 3D building reconstruction from ALS data using unambiguous decomposition into elementary structures. ISPRS Journal of Photogrammetry and Remote Sensing 118, 1-12.

Jolliffe, I., 2011. Principal component analysis, International encyclopedia of statistical science. Springer, pp. 1094-1096.

Jung, J., Jwa, Y., Sohn, G., 2017. Implicit regularization for reconstructing 3D building rooftop models using airborne LiDAR data. Sensors 17, 621-648.

Kada, M., 2009. The 3D Berlin project, Photogrammetric Week, pp. 331-340.

Kada, M., McKinley, L., 2009. 3D building reconstruction from LiDAR based on a cell decomposition approach. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 38, W4.

Kang, Z., Yang, J., 2018. A probabilistic graphical model for the classification of mobile LiDAR point clouds. ISPRS Journal of Photogrammetry and Remote Sensing.

Kim, C., Habib, A., Mrstik, P., 2007. New approach for planar patch segmentation using airborne laser data. Proceedings of the American Society for Photogrammetry and Remote Sensing (ASPRS) 2007.

Kohli, P., Torr, P.H., 2009. Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision 82, 302-324.

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

Lafferty, J., McCallum, A., Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Landrieu, L., Obozinski, G., 2017. Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. SIAM Journal on Imaging Sciences 10, 1724-1766.

Landrieu, L., Raguet, H., Vallet, B., Mallet, C., Weinmann, M., 2017. A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 132, 102-118.

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

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436.

Lee, I., Schenk, T., 2002. Perceptual organization of 3D surface points. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences 34, 193-198.

Lee, J., Yang, B., 2019. Developing an optimized texture mapping for photorealistic 3D buildings. Transactions in GIS 23(1): 1-21.

Li, M., Nan, L., Liu, S., 2016a. Fitting boxes to Manhattan scenes using linear integer programming. International Journal of Digital Earth 9, 806-817.

Li, M., Sun, C., 2018. Refinement of LiDAR point clouds using a super voxel based approach. ISPRS Journal of Photogrammetry and Remote Sensing 143, 213-221.

Li, M., Wonka, P., Nan, L., 2016b. Manhattan-world urban reconstruction from point clouds, European Conference on Computer Vision. Springer, pp. 54-69.

Li, S.Z., 2009. Markov random field modeling in image analysis. Springer Science & Business Media.

Liang, J., Edelsbrunner, H., Fu, P., Sudhakar, P.V., Subramaniam, S., 1998. Analytical shape computation of macromolecules: I. Molecular area and volume through alpha shape. Proteins: Structure, Function, and Bioinformatics 33, 1-17.

Lim, E.H., Suter, D., 2009. 3D terrestrial LIDAR classifications with super-voxels and multi-scale conditional random fields. Computer-Aided Design 41, 701-710.

Lin, H., Gao, J., Zhou, Y., Lu, G., Ye, M., Zhang, C., Liu, L., Yang, R., 2013. Semantic decomposition and reconstruction of residential scenes from LiDAR data. ACM Transactions on Graphics (TOG) 32, 66-76.

Lin, Y., Wang, C., Zhai, D., Li, W., Li, J., 2018. Toward better boundary preserved supervoxel segmentation for 3D point clouds. ISPRS Journal of Photogrammetry and Remote Sensing.

Lodha, S.K., Fitzpatrick, D.M., Helmbold, D.P., 2007. Aerial lidar data classification using adaboost, 3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on. IEEE, pp. 435-442.

Luo, H., Wang, C., Wen, C., Chen, Z., Zai, D., Yu, Y., Li, J., 2018. Semantic Labeling of Mobile LiDAR Point Clouds via Active Learning and Higher Order MRF. IEEE Transactions on Geoscience and Remote Sensing.

Ma, R., 2005. Building model reconstruction from LiDAR data and aerial photographs. The Ohio State University.

Maas, H. G., Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. ISPRS Journal of Photogrammetry and Remote Sensing 54, 153-163.

Mallet, C., Bretar, F., Roux, M., Soergel, U., Heipke, C., 2011. Relevance assessment of full-waveform lidar data for urban area classification. ISPRS Journal of Photogrammetry and Remote Sensing 66, S71-S84.

Mallet, C., Bretar, F., Soergel, U., 2008. Analysis of full-waveform lidar data for classification of urban areas. Photogrammetrie Fernerkundung GeoInformation (PFG) 5, 337-349.

Mao, J., Wang, X., Li, H., 2019. Interpolated convolutional networks for 3d point cloud understanding. Proceedings of the IEEE International Conference on Computer Vision, pp. 1578-1587.

Matei, B.C., Sawhney, H.S., Samarasekera, S., Kim, J., Kumar, R., 2008. Building segmentation for densely built urban regions using aerial LIDAR data, 2008 IEEE Conference on Computor Vision and Pattern Recognition. IEEE, Anchorage, AK, USA, pp. 1-8.

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

Medeiros, E., Velho, L., Lopes, W., 2004. Restricted BPA: applying ball-pivoting on the plane, Proceedings. 17th Brazilian Symposium on Computer Graphics and Image Processing. IEEE, Curitiba, Brazil, Brazil, pp. 372-379.

Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., Van Gool, L., Purgathofer, W., 2013. A survey of urban reconstruction, Computer graphics forum. Wiley Online Library, pp. 146-177.

Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., Chen, B., 2010. Smartboxes for interactive urban reconstruction, ACM Transactions on Graphics (TOG). ACM, p. 93.

Nan, L., Wonka, P., 2017. Polyfit: Polygonal surface reconstruction from point clouds, Proceedings of the IEEE International Conference on Computer Vision, pp. 2353-2361.

Nex, F., Gerke, M., 2014. Photogrammetric DSM denoising. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 40, 231.

Nguyen, A., Le, B., 2013. 3D point cloud segmentation: A survey, 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM). IEEE, pp. 225-230.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2013. Classification of urban LiDAR data using conditional random field and random forests, Urban Remote Sensing Event (JURSE), 2013 Joint. IEEE, pp. 139-142.

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

Niemeyer, J., Rottensteiner, F., Sörgel, U., Heipke, C., 2016. Hierarchical higher order crf for the classification of airborne lidar point clouds in urban areas. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives 41 (2016) 41, 655-662.

Paparoditis, N., Papelard, J. P., Cannelle, B., Devaux, A., Soheilian, B., David, N., Houzay, E., 2012. Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology. Revue Française de Photogrammétrie et de Télédétection 200, 69-79.

Papon, J., Abramov, A., Schoeler, M., Worgotter, F., 2013. Voxel cloud connectivity segmentation-supervoxels for point clouds, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2027-2034.

Piekarski, W., Thomas, B.H., 2001. Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer, Proceedings Fifth International Symposium on Wearable Computers. IEEE, pp. 31-38.

Poliyapram, V., Wang, W., Nakamura, R., 2019. A point-wise LiDAR and image multimodal fusion network (PMNet) for aerial point cloud 3D semantic segmentation. Remote Sensing 11(24): 2961-2985.

Poullis, C., 2013. A framework for automatic modeling from point cloud data. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 2563-2575.

Poullis, C., You, S., 2009a. Automatic reconstruction of cities from remote sensor data, 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2775-2782.

Poullis, C., You, S., 2009b. Photorealistic large-scale urban city model reconstruction. IEEE Transactions on Visualization and Computer Graphics 15, 654-669.

Poullis, C., You, S., Neumann, U., 2008. Rapid creation of large-scale photorealistic virtual environments, 2008 IEEE Virtual Reality Conference. IEEE, Reno, NE, USA, pp. 153-160.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. Computer Vision and Pattern Recognition (CVPR), IEEE 1, 4.

Rabbani, T., Van Den Heuvel, F., 2005. Efficient hough transform for automatic detection of cylinders in point clouds. ISPRS Wg Iii/3, Iii/4 3, 60-65.

Relph, E., 2016. The Modern Urban Landscape (Routledge Revivals). Routledge.

Ren, X., Malik, J., 2003. Learning a classification model for segmentation, null. IEEE, p. 10.

Rivers, A., Durand, F., Igarashi, T., 2010. 3D modeling with silhouettes. Acm.

Rusu, R.B., 2018. PCL: Radius outlier removal < http://docs.pointclouds.org/1.8.1/classpcl_1_1_radius_outlier_removal.html >

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration, 2009 IEEE International Conference on Robotics and Automation. IEEE, pp. 3212-3217.

Rusu, R.B., Cousins, S., 2011. Point cloud library (pcl), 2011 IEEE International Conference on Robotics and Automation, pp. 1-4.

Rusu, R.B., Marton, Z.,C., Blodow, N., et al., 2008. Towards 3D point cloud based object maps for household environments. Robotics and Autonomous Systems 56(11): 927-941.

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

Sánchez-Lopera, J., Lerma, J.L., 2014. Classification of lidar bare-earth points, buildings, vegetation, and small objects based on region growing and angular classifier. International Journal of Remote Sensing 35, 6955-6972.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for point‐cloud shape detection, Computer Graphics Forum. Wiley Online Library, pp. 214-226.

Schrijver, A., 1998. Theory of linear and integer programming. John Wiley & Sons.

Sohn, G., Huang, X., Tao, V., 2008. Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data. Photogrammetric Engineering & Remote Sensing 74, 1425-1438.

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C., 2008. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 1068-1080.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., 2007. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data, ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, pp. 407-412.

Vallet, B., Brédif, M., Serna, A., Marcotegui, B., Paparoditis, N., 2015. TerraMobilita/iQmulus urban point cloud analysis benchmark. Computers & Graphics 49, 126-133.

Verdie, Y., Lafarge, F., Alliez, P., 2015. LOD generation for urban scenes. Association for Computing Machinery (ACM) 30.

Verma, V., Kumar, R., Hsu, S., 2006. 3D building detection and modeling from aerial LIDAR data, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. IEEE, pp. 2213-2220.

Vo, A. V., Truong-Hong, L., Laefer, D.F., Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. ISPRS Journal of Photogrammetry and Remote Sensing 104, 88-100.

Vosselman, G., 1999. Building reconstruction using planar faces in very high density height data. International Archives of Photogrammetry and Remote Sensing 32, 87-94.

Vosselman, G., Coenen, M., Rottensteiner, F., 2017. Contextual segment-based classification of airborne laser scanner data. ISPRS Journal of Photogrammetry and Remote Sensing 128, 354-371.

Vosselman, G., Dijkman, S., 2001. 3D building model reconstruction from point clouds and ground plans. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences 34, 37-44.

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

Vu, H. H., Labatut, P., Pons, J.-P., Keriven, R., 2012. High accuracy and visibility-consistent dense multiview stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 889-901.

Wang, H., Wang, C., Luo, H., Li, P., Chen, Y., Li, J., 2015. 3-D point cloud object detection based on supervoxel neighborhood with Hough forest framework. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 8, 1570-1581.

Wang, R., Peethambaran, J., Chen, D., 2018. LiDAR point clouds to 3-D urban models: A review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 11, 606-627.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015a. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing 105, 286-304.

Weinmann, M., Jutzi, B., Mallet, C., 2014. Semantic 3D scene interpretation: a framework combining optimal neighborhood size selection with relevant features. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2, 181.

Weinmann, M., Jutzi, B., Mallet, C., 2017. Geometric features and their relevance for 3d point cloud classification. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 4, 157.

Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C., 2015b. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. Computers & Graphics 49, 47-57.

Wu, B., Yu, B., Wu, Q., Yao, S., Zhao, F., Mao, W., Wu, J., 2017. A graph-based approach for 3D building model reconstruction from airborne LiDAR point clouds. Remote Sensing 9, 92-108.

Wu, Q., Liu, H., Wang, S., Yu, B., Beck, R., Hinkel, K., 2015a. A localized contour tree method for deriving geometric and topological properties of complex surface depressions based on high-resolution topographical data. International Journal of Geographical Information Science 29, 2041-2060.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015b. 3d shapenets: A deep representation for volumetric shapes, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912-1920.

Xie, L., Hu, H., Zhu, Q., Wu, B., Zhang, Y., 2017. Hierarchical Regularization of Polygons for Photogrammetric Point Clouds of Oblique Images. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 42.

Xie, L., Zhu, Q., Hu, H., Wu, B., Li, Y., Zhang, Y., Zhong, R., 2018. Hierarchical Regularization of Building Boundaries in Noisy Aerial Laser Scanning and Photogrammetric Point Clouds. Remote Sensing 10, 1996.

Xiong, B., 2014. Reconstructing and correcting 3D building models using roof topology graphs. University of Twente, Faculty of Geo-Information Science and Earth Observation (ITC).

Xiong, B., Elberink, S.O., Vosselman, G., 2014. A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. ISPRS Journal of

Photogrammetry and Remote Sensing 93, 227-242.

Xiong, B., Jancosek, M., Elberink, S.O., Vosselman, G., 2015. Flexible building primitives for 3D building modeling. ISPRS Journal of Photogrammetry and Remote Sensing 101, 275-290.

Xiong, X., Munoz, D., Bagnell, J.A., Hebert, M., 2011. 3-D scene analysis via sequenced predictions over points and regions, 2011 IEEE International Conference on Robotics and Automation. IEEE, pp. 2609-2616.

Xu, S., Vosselman, G., Elberink, S.O., 2014. Multiple-entity based classification of airborne laser scanning data in urban areas. ISPRS Journal of Photogrammetry and Remote Sensing 88, 1-15.

Yang, B., Dong, Z., 2013. A shape-based segmentation method for mobile laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 81, 19-30.

Yang, B., Dong, Z., Zhao, G., Dai, W., 2015. Hierarchical extraction of urban objects from mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing 99, 45-57.

Yang, B., Wei, Z., Li, Q., Li, J., 2012. Automated extraction of street-scene objects from mobile lidar point clouds. International Journal of Remote Sensing 33, 5839-5861.

Yi, C., Zhang, Y., Wu, Q., Xu, Y., Remil, O., Wei, M., Wang, J., 2017. Urban building reconstruction from raw LiDAR point data. Computer-Aided Design 93, 1-14.

Yokoyama, H., Date, H., Kanai, S., Takeda, H., 2013. Detection and classification of pole-like objects from mobile laser scanning data of urban environments. International Journal of CAD/CAM 13.

Zhang, Z., Zhang, L., Tong, X., Guo, B., Zhang, L., Xing, X., 2016a. Discriminative-dictionary-learning-based multilevel point-cluster features for ALS point-cloud classification. IEEE Transactions on Geoscience and Remote Sensing 54, 7309-7322.

Zhang, Z., Zhang, L., Tong, X., Mathiopoulos, P.T., Guo, B., Huang, X., Wang, Z., Wang, Y., 2016b. A multilevel point-cluster-based discriminative feature for ALS point cloud classification. IEEE Transactions on Geoscience and Remote Sensing 54, 3309-3321.

Zhou, Q. Y., Neumann, U., 2008. Fast and extensible building modeling from airborne LiDAR data, GIS'08 Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, Irvine, California, USA.

Zhou, Q. Y., Neumann, U., 2010. 2.5 d dual contouring: A robust approach to creating building models from aerial lidar point clouds, European Conference on Computer Vision. Springer, pp. 115-128.

Zhou, Q. Y., Neumann, U., 2011. 2.5 D building modeling with topology control, CVPR 2011. IEEE, pp. 2489-2496.

Zhou, Q. Y., Neumann, U., 2012. 2.5D building modeling by discovering global regularities, 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Providence, RI, USA, pp. 326-333.

Zhou, Y., Yu, Y., Lu, G., Du, S., 2012. Super-segments based classification of 3D urban street scenes. International Journal of Advanced Robotic Systems 9, 248.

Zhu, Q., Li, Y., Hu, H., Wu, B., 2017. Robust point cloud classification based on multi-level semantic relationships for urban scenes. ISPRS Journal of Photogrammetry and Remote Sensing 129, 86-102.

Zolanvari, S.I., Laefer, D.F., Natanzi, A.S., 2018. Three-dimensional building façade segmentation and opening area detection from point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 143, 134-149.