Technical University of Munich
Department of Civil, Geo, and Environmental Engineering
Photogrammetry and Remote Sensing

Change detection of construction sites based on 3D point clouds

Rong Huang

Dissertation

2021

TECHNISCHE UNIVERSITÄT MÜNCHEN
Ingenieurfakultät Bau Geo Umwelt
Photogrammetrie und Fernerkundung

Change detection of construction sites based on 3D point clouds

Rong Huang

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. André Borrmann
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. Uwe Stilla
	2. Univ.-Prof. Dr.-Ing. Helmut Mayer

Die Dissertation wurde am 19.05.2021 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 14.07.2021 angenommen.

Abstract

Automatic monitoring of construction processing has drawn attention in the fields of Architecture-Engineering-Construction, and Facilities-Management industry increasingly. Point clouds, acquired via laser scanning or stereo matching of images, have been considered the most appropriate data source in monitoring the lifecycle of construction projects due to accurate and detailed 3D information provided. However, inconsistent coordinate systems, lack of topological and semantic information, complex construction scenarios pose great challenges for using point clouds in construction monitoring and change detection.

This research aims to develop novel methods and techniques involving point cloud registration, semantic segmentation, and change detection to obtain and present spatial and temporal changes in structural components of buildings of the site during the construction process. Hereby, the work provides contributions on three major aspects: (i) global matching for point cloud registration, (ii) point embedding for semantic segmentation of point clouds, and (iii) information fusion for change detection.

To achieve robust and efficient registration of point clouds, we developed novel registration methods that utilized global features and their attributes in the frequency domain. The low-frequency components of the global signals are matched to achieve robust and efficient registration of point clouds. To interpret 3D scenes robustly and embed 3D points in discriminative feature space, we focus on the engineering of point correlations, the involvement of attention mechanism, and the improvement of receptive fields of points. The point correlations can be either considered using the manifold-learning-based method in which the correlation in feature space and spatial space are both involved or can be embedded by building global relations between points using deep learning techniques.. The attention mechanism is also involved in improving the discriminate features and suppressing interference. The improvement of the receptive field can be achieved by building multi-scale neighborhoods during feature learning. In the change detection task, both the geometric and semantic changes are considered. Geometric changes are obtained by the occupancy conflicts of point clouds. By fusing the geometric and semantic information using the Dempster-Shafer theory, we can visualize the changes in the construction sites.

The methods for the proposed co-registration and segmentation were evaluated by experiments with different open benchmark datasets and for the change detection with an TUM-PF-own datasets. For the co-registration (WHU-TLS), an average of translation errors about 40 cm and an average of rotation errors about 0,1 degree can be achieved. For the segmentation, an overall accuracy of about 85% (ISPRS Vaihingen ALS) and about 54% (TUM photogrammetric point cloud) can be obtained. As for the change detection, an overall accuracy of about 75% can be finally achieved.

Kurzfassung

Die automatische Überwachung von Bauprozessen hat in den Bereichen Architektur, Ingenieurwesen, Bauwesen und Facility Management zunehmend an Bedeutung gewonnen. Punktwolken, die durch Laserscanning oder automatische Bildzuordnung erfasst werden, gelten als die am besten geeignete Datenquelle für die Überwachung des Lebenszyklus von Bauprojekten, da sie genaue und detaillierte 3D-Informationen liefern. Inkonsistente Koordinatensysteme, fehlende topologische und semantische Informationen und komplexe Bauszenarien stellen jedoch eine große Herausforderung für die Verwendung von Punktwolken bei der Bauüberwachung und der Erkennung von Veränderungen dar.

Ziel dieser Forschungsarbeit ist es, neue Methoden und Techniken zu entwickeln, die Koregistrierung von Punktwolken, semantische Segmentierung und Änderungsdetektion umfassen, um räumliche und zeitliche Veränderungen der strukturellen Gebäudekomponenten einer Baustelle während des Bauprozesses zu erfassen und darzustellen. Dabei liefert die Arbeit Beiträge zu drei Hauptaspekten: (i) Globales Matching für die Registrierung von Punktwolken, (ii) Merkmalseinbettung für die semantische Segmentierung von Punktwolken und (iii) Informationsfusion für die Änderungsdetektion.

Um eine robuste und effiziente Registrierung von Punktwolken zu erreichen, wurden neue Methoden entwickelt, die globale Merkmale und ihre Attribute im Frequenzbereich nutzen. Die niederfrequenten Komponenten der globalen Signale werden genutzt, um eine robuste und effiziente Registrierung von Punktwolken zu erreichen. Um 3D-Szenen robust zu interpretieren und 3D-Punkte in einen diskriminierenden Merkmalsraum einzubetten, konzentriert sich die Arbeit auf die Entwicklung von Punktkorrelationen, die Einbeziehung von Aufmerksamkeitsmechanismen und die Verbesserung des rezeptiven Felds von Punkten. Die Punktkorrelationen können entweder durch nichtlineare Dimensionsreduktion bestimmt werden, bei der sowohl die Korrelation im Merkmalsraum als auch Nachbarschaftsbeziehungen einbezogen werden oder durch globale Beziehungen zwischen Punkten durch Deep-Learning-Methoden eingebettet werden. Der Einsatz eines sogenannten Aufmerksamkeitsmechanismuses dient der Verbesserung der Unterscheidungsmerkmale und unterdrückt Störungen. Die Verbesserung des rezeptiven Feldes kann durch den Aufbau von Nachbarschaften in mehreren, verschiedenen Distanzen während des Lernprozesses erreicht werden. Bei der Änderungsdetektion werden sowohl die geometrischen als auch die semantischen Veränderungen berücksichtigt. Geometrische Änderungen werden durch Belegungskonflikte von Punktwolken ermittelt. Durch die Fusion von geometrischen und semantischen Informationen unter Verwendung der Dempster-Shafer-Theorie können Änderungen auf Baustellen visualisiert werden.

Die Methoden für die vorgeschlagene Koregistrieung und Seemntierung wurden durch Experimente mit verschiedenen offenen Benchmarkdatensätzen und die Änderungsdetektion mit TUM-PF-eigenen Datensätzen evaluiert. Im Mittel konnten für die Koregistrierung (WHU-TLS) ein Translationsfehler von ca. 30 cm und einen Rotationsfehler von ca. 0,1 Grad, für die Segmentierung Gesamtgenauigkeiten von ca. 85% (ISPRS Vaihingen ALS) und ca. 54% (TUM photogrammetrische Punktwolke) und für Änderungsdetektion ca 75% erreicht werden.

Contents

List of Abbreviations

Abbreviation	Description	Page
AEC/FM	Architecture, Engineering, Construction, and Facilities Management	19
AHN	Actueel Hoogtebestand Nederland	89
BA	Bundle Adjustment	47
BIM	Building Information Model	19
CRA	Channel Relation-aware Attention Module	76
CRF	Conditional Random Fields	25
DALES	Dayton Annotated LiDAR Earth Scan	89
DFE	Directional Feature Encoding	76
DFT	Discrete Fourier Transform	33
DOF	Degree of Freedom	23
DPE	Deep Point Embedding	67
DSM	Digital Surface Model	26
DST	Dempster-Shafer Theory	29
EFE	Elevation Feature Encoding	76
FMT	Fourier-Mellin Transformation	35
4PCS	4-Points Congruent Set	23
FPFH	Fast Point Feature Histogram	24
GGO	Global Graph-based Optimization	69
GRA	Global Relation-aware Attention Module	76
GraNet	Global Relation-aware Attentional Neural Network	67
GRPC	Robust Global Registration with 3D Phase Correlation	53
HDL	Hierarchical Deep Feature Learning	69
HMMR	Hierarchical Merging-based Multiview Registration	103
ICP	Iterative Closest Point	23
IFT	Inverse Fourier Transform	34
JME	Joint Manifold-based Embedding	69
K4PCS	Keypoint-based 4PCS	23
KNN	K-Nearest Neighborhood	36
LASDU	Large-scale Aerial LiDAR Dataset for Semantic Labeling in Dense Urban Areas	89
LDA	Linear Discriminant Analysis	69
LiDAR	Light Detection and Ranging	19
LML	Local Manifold Learning	37
LoSDA	Local Spatial Discrepancy Attention Convolution Module	76
LPP	Locality Preserving Projections	38
LPT	Log-Polar Transformation	36
LRF	Local Reference Frame	24
MDF	Multi-scale Deep Features	113
MLCE	Multi-scale Local Context Embedding	67
MRF	Markov Random Field	25
MVS	Multi-View Stereo	46
NDT	Normal Distribution Transform	24
OBCD-M	Occupancy-Based Change Detection Using Multi-Stereo Vision	83

Abbreviation	Description	Page
OGF	Original Geometric Features	111
PBPC	Projection-Based Registration with 2D Phase Correlation	49
PCA	Principal Component Analysis	69
PLADE	Plane-based Descriptor	104
RF	Random Forest	25
RoPS	Rotational Projection Statistics	24
SACD	Semantic-Aided Change Detection	83
SDE	Spatial Distribution Encoding	76
SDF	Single-scale Deep Features	113
SE	Spectral Embedding	116
S8N	Stacked 8-Neighborhood Search	77
SfM	Structure From Motion	46
SHOT	Signature of Histogram of Orientations	24
SIFT	Scale-Invariant Feature Transform	24
SK4PCS	Semantic Keypoint-based 4PCS	23
SPG	Superpoint Graph	26
SRA	Spatial Relation-aware Attention Module	76
SVD	Singular Value Decomposition	35
SVM	Support Vector Machines	25
TLS	Terrestrial Laser Scanning	19
V4PCS	Voxel-based 4-Plane Congruent Sets	101
V-4PCS	Volumetric 4PCS	23

List of Figures

List of Tables

1 Introduction

1.1 Motivation

In the fields of Architecture-Engineering-Construction and Facilities-Management (AEC/FM), the management of the life-cycle of a construction project is of great importance. Activity management of the construction project usually requires a forward flow of design intent of the project and a feedback flow of project state information. As a process providing feedback information, progress monitoring is one of the core tasks in the management of a construction project [Turkan et al., 2012]. The need for automatic and accurate progress monitoring of construction projects has increased in recent decades. In the past decade, advances in computer vision have boosted many crucial tasks like building modeling [Arayici, 2007; Xu et al., 2018a; Xu & Stilla, 2019], progress tracking [Turkan et al., 2012; Kim et al., 2013; Bosché et al., 2014], scene recognition [Xu et al., 2017], and quality control [Shih & Wang, 2004; Gordon & Akinci, 2005; Arayici, 2007] towards automatic, intelligent, and integrated processing pipelines. Here, Building Information Model (BIM) is a representative platform that is increasingly widely used for the design of construction projects and the monitoring of the construction progress, which can provide a solution for accurate progress monitoring [Tang et al., 2010; Tuttas et al., 2017]. However, there are still gaps between the current development of progress monitoring and the final goal of progress monitoring with high automation and digitization. Challenges also occur in many specific tasks, such as integration of multiple attributes, alignment of time frames, and standardization of data formats and processing interfaces. For tackling these challenges, progress monitoring has been widely studied in the fields of AEC/FM. However, traditional ways of implementing progress monitoring depend on visual inspections, extensive manual records, and data analysis. The conventional methods rely heavily on personal skills, inspector experiences, and surveying techniques [Bosché, 2010]. The traditional surveying-based methods for construction monitoring are labor-intensive, error-prone, time-consuming, and lacking continuity. To address these problems, many researchers have studied various techniques for automatic construction monitoring, such as 2D imaging based site analysis [Haas et al., 1984; Abeid et al., 2003; Ibrahim et al., 2009; Chi et al., 2009; Wu et al., 2010], photogrammetry based site mapping [El-Omari & Moselhi, 2008; Golparvar-Fard et al., 2009, 2015; Braun et al., 2015; Tuttas et al., 2015, 2017], and terrestrial laser scanning (TLS) based mapping and scene analysis [Stone & Cheok, 2001; El-Omari & Moselhi, 2008; Lee et al., 2013; Kim et al., 2013; Bosché et al., 2014; Xu et al., 2017; Xu et al., 2018b]. Among the aforementioned technologies, point clouds acquired using light detection and ranging (LiDAR) and stereo vision techniques have been considered the most appropriate technique to capture 3D information of construction projects with high accuracy and efficiency [Xu & Stilla, 2021]. By using LiDAR techniques, distances from the sensor to nearby surfaces can be observed with millimeter to centimeter accuracy at speeds of thousands of point measurements per second [Tang et al., 2010]. While benefiting from the stereo vision techniques, images captured by digital cameras can not only cover regions of interest but also be processed to generate point clouds with accurate positions and rich 3D details. Besides, colors and textures also serve as additional information to aid further modelling. Laser scanning and stereo vision techniques show their advances

in providing fast, accurate, comprehensive, and detailed 3D as-built information for the sensed construction scene.

In Fig. 1.1, Tuttas et al. [2017] illustrated the general procedure for construction progress monitoring using point clouds. The construction project's schedule can be updated by comparing the as-built information from the sensed data with as-planned information provided by 4D-BIM. Thus, decisions can be made for further construction activities. For a fair comparison with as-planned BIM, as-built information from the sensed technologies has to be organized at the object level for fulfilling the requirement of progress monitoring purposes. However, the raw 3D points do not naturally contain object-oriented information. Further processing of the measured 3D data is required for achieving the task of progress monitoring.

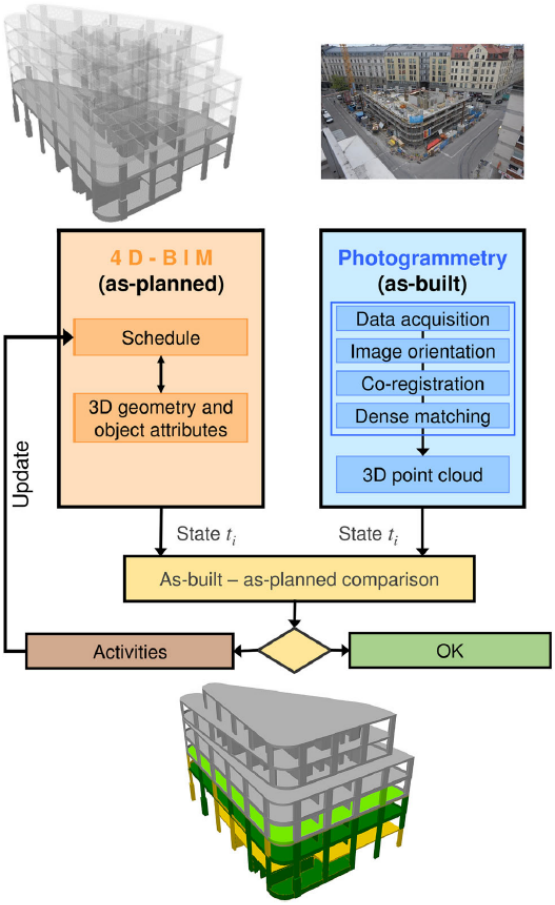

Figure 1.1: Illustration of the general procedure for construction progress monitoring using a 4D BIM and a photogrammetric point cloud [Tuttas et al., 2017].

Remarkable researches have been reported in developing processes and algorithms for processing 3D point cloud data in the field of progress monitoring. Turkan et al. [2012] developed an automated recognition system that combines 3D object recognition technology with schedule information into a combined 4D objection-oriented progress tracking system, which helped update the design plan of the construction progress. Kim et al. [2013] proposed a system by matching a-built data with an as-planned model and revising the as-built status. Bosché et al. [2014] monitored the construction of a utility corridor in a university engineering building with a Scan-versus-BIM object recognition framework. Tuttas et al. [2017] performed continuous monitoring

by acquiring and subsequently comparing co-registered point clouds. Xu et al. [2018a] provided a robust solution for reconstructing scaffolds from a photogrammetric point cloud of a construction site using a novel 3D feature descriptor and a projection-based segmentation method. Puri & Turkan [2020] developed a semi-automated methodology for monitoring bridge construction projects by comparing project as-built data from 3D point clouds and 4D project design model. The process involved the registration of virtual point cloud from 3D model and as-built point cloud and also segmentation and object recognition based on the scanned data.

From the aforementioned researches, it can be seen that three major tasks should be addressed (see Fig. 1.2) in the progress monitoring: (1) registration; (2) semantic segmentation or object recognition; (3) change detection. To be specific, point clouds acquired at different epochs should be aligned by point cloud registration for a fair comparison between different states of a construction project or comparison between as-built and as-planned data. Meanwhile, semantic segmentation should be conducted to fill semantic gaps between raw point clouds and the required as-built model. Most importantly, comparison between different states of construction process should be implemented by change detection of point clouds to detect and present changes of construction scenes.

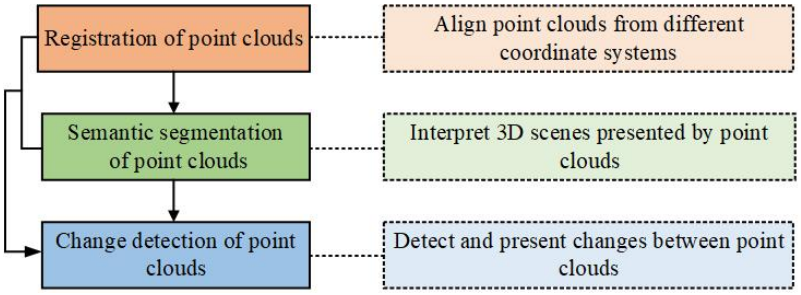

Figure 1.2: Three-step workflow for change detection from point clouds.

However, to achieve automatic progress monitoring and change detection, several essential problems must be addressed (see Fig. 1.3). First, point clouds acquired during the construction process and BIMs at different states should be aligned in the same coordinate system to enable the comparison. Generally, due to temporal changes and incompleteness caused by occlusions in complex urban environments, it leads to difficulties in setting reliable and sufficient ground control points and it is usually challenging to find correspondences completely automatically. It makes the registration of point clouds a challenging task. Second, there are uneven densities resulting from different viewing distances of scanners and noise and outliers of point clouds, which makes the extracted features ineffectual and leads to difficulties in interpreting the complex construction scene. In addition, in construction scenes, there exists not only various categories of building objects but also some temporal objects belonging to other categories, which makes construction scenes of high complexity. Third, the progress of a construction project is indicated by changes between different states of the construction. The comparison is usually conducted based on the segmentation and recognition results of building objects. However, as mentioned before, for either laser scanning or stereo vision technique, they all have limits in observations, such as occlusions. This factor should also be considered when detect and present changes in a construction scene. Considering the aforementioned problems, the research questions can be summarized as follows:

I To what extent of robustness, accuracy, and efficiency could a marker-free alignment of point clouds measured at different time epochs with temporal changes and incompleteness of data achieve?

II What are the necessary aspects that must be considered when learning features to ensure an accurate interpretation of complex scenarios using point clouds with uneven point densities, intense noise, and outliers?

III To what extent of automation could be achieved for change detection? Is it sufficient enough to detect changes by purely comparing segmentation results of building objects?

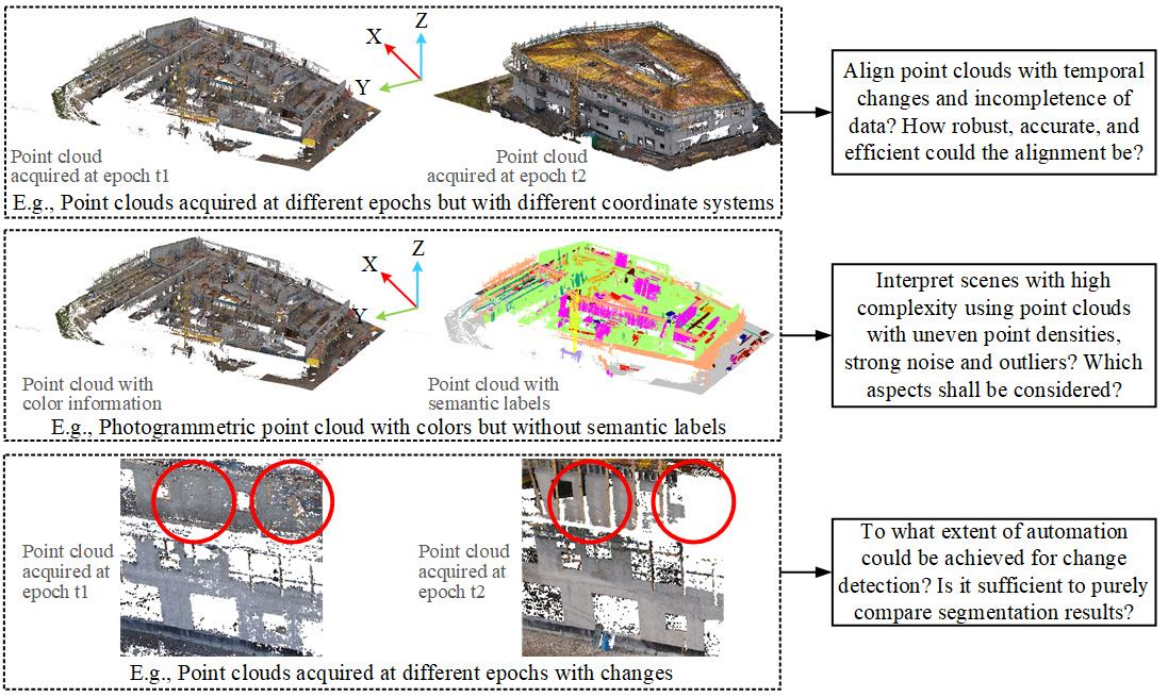

Figure 1.3: Problems for construction monitoring and change detection.

For finding answers to the aforementioned research questions, before introducing our objectives and contributions, we first provide a detailed survey and review of related work in the following section.

1.2 State of the art

Numerous researchers have studied change detection using point clouds. According to the three-step workflow mentioned in the last section, we will provide detailed reviews and discussions concerning methods and algorithms relating to registration, semantic segmentation, and change detection in the following section.

1.2.1 Registration of point clouds

Point cloud registration has long been a challenging task in the field of photogrammetry and computer vision. The objective of registration is to estimate a rigid transformation that aligns multiple individual but related point clouds into a unified coordinate system [Hebel et al., 2009; Xu et al., 2019a]. These point clouds might be acquired from different viewpoints, at different times, using different platforms, or via multimodal sensors. The registration between them should be done in an automatic and marker-free manner. Numerous studies have been intensively reported to solve mark-less point cloud registration, with two major steps always involved, including the

estimation of correspondences and the calculation of transformation parameters. Here, matching correct correspondences is the key to the success of registration. In the following, we will review methods using the matching of correspondences, which can be grouped into three fundamental classes conforming to the principles that they used: geometric constraint-based methods, feature description-based methods, and global information-based methods.

Geometric constraint-based registration

For geometric constraint-based approaches, a geometric constraint is formed by points or primitives as an indicator for retrieving and matching correspondences. For example, Iterative Closest Point (ICP) searched for associated points based on minimizing point-by-point distances between the various point clouds [Besl & McKay, 1992; Habib et al., 2010; Al-Durgham & Habib, 2013] for point cloud registration. Its variants, such as Geometric Primitive ICP [Bae & Lichti, 2008], geometric features + ICP [Habib et al., 2005, 2010; Gressin et al., 2013], Go-ICP [Yang et al., 2013], are also representative approaches, which utilize geometric constraints by minimizing distances between corresponding elements. However, for the ICP-based methods, proper initial transformation estimation is needed to avoid incorrect local optimum. Apart from ICP-based methods, many methods follow a different registration scheme, in which specially designed combinations of points or primitives matter to the identification of corresponding points. This specially designed combination of points or primitives can create a constraint when searching for candidate pairs of points, significantly increasing the efficiency compared with a random matching test. 4-point congruent systems (4PCS) and its variants such as Super4PCS [Mellado et al., 2014], keypoint-based 4PCS (K4PCS) [Theiler et al., 2014], and semantic keypoint-based 4PCS (SK4PCS) [Ge, 2017] are representative approaches following this strategy. In this type of method, corresponding sets of congruent points are identified by utilizing the constraint of intersection ratios and selecting candidates for finding correspondences. In affine transformation, intersection ratios of four points congruent sets consisting of two pairs of points are invariant. Thus, by filtering out all four point-sets follow intersection ratios from a given four point-sets in the target point cloud, we can reduce the number of candidates in the source point cloud. Compared with feature description-based registration, the geometric constraint-based methods have higher robustness to occlusions and unequal densities since the geometric constraint can be built on a larger scale than the features extracted from a local context. Similarly, instead of points, using the combination of different kinds of primitives, for example, two pairs of planes [Chen et al., 2019], is also a compelling choice. The use of geometric primitives like planes can upgrade the robustness of the geometric features, as they constraint degree of freedom (DoF) and are less sensitive to uneven point densities and outliers [Xu et al., 2019a]. For example, measured distances between points in the point-based 4PCS methods are more sensitive to noise than the primitive-based one. The volumetric 4PCS (V-4PCS) [Huang et al., 2017] is also a method under the framework of 4PCS, which extended the surface expression to volumetric ones and shows a promising improvement in computational efficiency.

Feature description-based registration

For feature description-based registration approaches, the corresponding pairs between point clouds are identified through retrieving features with the most substantial similarity. An appropriate feature description plays an essential role in this retrieving process, usually implemented by feature descriptors. Various feature descriptors have been demonstrated in many studies that are useful in feature retrieving and matching. An eligible feature descriptor should have two core characteristics, namely, high descriptiveness and rotation-invariance. High descriptiveness ensures a discriminative description of geometric features for non-corresponding points and sub-

stantial similarity between features of corresponding points. Rotation-invariance guarantees the robustness of the generated features, which the rigid transformation between point clouds should not influence. Renowned examples of feature descriptors include scale-invariant feature transform (SIFT) [Flitton et al., 2010], fast point feature histogram (FPFH) [Rusu et al., 2009], rotational projection statistics (RoPS) [Guo et al., 2013] and signature of histogram of orientations (SHOT) [Tombari et al., 2010]. However, the performance of descriptors (i.e., SIFT) highly depends on the saliency of input points selected by keypoint detectors like Harris 3D. The detection of key points will highly influence the performance of both candidate selection and feature extraction. Furthermore, the basic principle for achieving rotation in-variance mainly counts on the pose normalization. For instance, SIFT achieves rotation-invariance in feature extraction by orienting the local reference frame (LRF) axis to the dominant orientation of gradients. However, the orientation of LRF is easy to be influenced by noise and outliers. An alternative is to obtain the local geometry statistics, which are easy to implement and fast to compute. However, the critical problem is that this kind of feature may encounter low descriptiveness. Additionally, features can also be extracted from geometric primitives that clustered from points, such as lines [Habib et al., 2005; Hebel & Stilla, 2010, 2012; Ge & Hu, 2020], curves [Yang & Zang, 2014], planes [Xiao et al., 2013], surfaces [Ge & Wunderlich, 2016]. Thus, the accuracy of extracting these geometric candidates for registration, such as key points or primitives, is an important factor that influences the registration results. Besides, artifacts may also be brought in when extracting geometric primitives.

Global information-based registration

In the aforementioned registration categories, local information is mainly utilized and generated based on 3D points themselves or clusters of primitives. Registration can also make use of global features derived from the entire point clouds. For instance, in the normal distribution transform (NDT) method [Biber & Straßer, 2003], points were transformed into a normal distribution, the natural distribution of which forced alignment between point clouds [Magnusson et al., 2007]. The distribution of point densities is another global indicator for alignment. In some representative methods, coherent point-drift [Myronenko & Song, 2010] and kernel affinity correlation [Tsin & Kanade, 2004] were applied on the density for finding correspondences. In a recent work of [Dong et al., 2018], global features were used for the fast orientation of multi-scan unordered point clouds. In our previous work [Huang et al., 2019], 3D point clouds of a highly complicated scenario were projected into 1D histograms and 2D images for achieving registration in low-dimensional spaces. These projected histograms and images were also a global expression of original point clouds.

Generally, for both geometric constraint-based and feature description-based registration, they follows the strategy of finding local correspondence. Compared with local-correspondence-based approaches, the global information-based methods can avoid the establishing of local feature descriptions and thus be more robust to noise and outliers and less sensitive to irrelevant changes of details. However, large overlap are usually required for global-information-based methods. Otherwise, the approaches based on global features may make a significant difference.

1.2.2 Semantic segmentation of point clouds

For parsing semantic information of the 3D scene from point clouds, one practical solution is semantic labeling. The primary goal of semantic labeling of point clouds is to annotate every point in the point cloud with a label of semantic meaning, in accordance with geometric or radiometric information provided based on the point itself and its neighborhood. This can be achieved via the classification of acquired points. Due to the development of deep learning techniques, many new advances have also brought new solutions for semantic segmentation of point clouds. Thus,

semantic segmentation approaches can be divided into two groups based on the rule that whether deep learning techniques are applied: traditional methods and deep learning-based methods.

Point cloud classification with traditional methods

To achieve the semantic labeling of points, supervised classification is typically implemented [Vosselman et al., 2017; Li et al., 2019a]. It comprises two main steps: generation of distinctive features and classification of 3D points with corresponding features with a classifier. For the extraction of features, the local context of each point is conventionally defined by its neighboring points and presented by various handcrafted mathematical expressions based on spatial or spectral attributes of these points. For the training process, the mathematical expressions of the selected representative samples are integrated into a feature vector and fed into a classifier along with the corresponding labels. In previous work, to create discriminative features, studies have exploited both point geometry and inherent attributes. Many contextual features extracted from spatial distributions and directions of points have shown their effectiveness in the classification task [Yang et al., 2017a], such as eigenvalue based features from covariance matrix of point coordinates [Chehata et al., 2009; Weinmann et al., 2015a,c], waveform-based features from transformation [Jutzi & Gross, 2010; Zhang et al., 2011], 2D projected patterns [Zhao et al., 2018], elevation values and height differences [Maas, 1999; Gorgens et al., 2017; Sun et al., 2018], and orientations of points from normal vectors [Rabbani et al., 2006]. However, designing good handcrafted features is a critical and challenging task, which requires a good understanding of the scanned objects and highly depends on empirical tests [Xu et al., 2019b].

Classifier refers to a mathematical function or transformation implemented by algorithms or strategies, which projects input features to a category. A well-designed classifier should maximum the discrimination of features of various semantics. Regarding point cloud classification, a considerable amount of classifiers have been introduced and tested, including AdaBoost [Chan & Paelinckx, 2008], support vector machines (SVM) [Mallet et al., 2011], composite kernel SVM [Ghamisi & Höfle, 2017], and random forest (RF) [Chehata et al., 2009], Hough forest [Yu et al., 2016], 3D Markov Random Field (MRF) [Yin & Collins, 2007], and conditional random fields (CRF) [Niemeyer et al., 2014; Weinmann et al., 2015b; Yao et al., 2017; Vosselman et al., 2017; Li et al., 2019b]. For supervised classification, a classifier is trained using the generated features and the corresponding labels so that the classifier's parameters can be optimized for inferring categories of points from input features. Then, the trained classifier can be used to predict labels of points from other test areas. However, these interactions are not controllable. There is still heterogeneity in the classification results in some cases, especially in low-density areas and borders of urban objects. Therefore, contextual information is typically considered to improve the spatial smoothness of the classification results. Furthermore, to encode the spatial dependencies between 3D points, a graph structure is usually constructed to model the adjacency relationship. Numerous optimization strategies are based on specific classic graphical models such as MRF [Munoz et al., 2009; Lu & Rasmussen, 2012; Kang & Yang, 2018]. In Landrieu et al. [2017], instead of fixing on some standard graphical models, a general mathematical optimization framework with more versatile solutions for spatial smoothing is proposed.

Point cloud classification with deep learning techniques

Compared with classification approaches using handcrafted features, classification methods using learned features can automatically discover the feature representations needed for classification from raw points, requiring less prior knowledge and avoiding sophisticated feature design. Feature learning is usually achieved via dictionary learning or deep learning. At the moment, deep-learning-based methods for point cloud classification are getting increasingly popular recently,

which provides end-to-end solutions that decrease the efforts in feature design and improve abilities to learn high-level feature representation for the classification task. For a neural network-based method, its layer structure and parameters can implicitly express the spatial interactions between 3D points, facilitating the feature representations. Tennumerous neural networks have achieved remarkable performance in a wide range of applications. Examples include VoxNet [Maturana & Scherer, 2015], MultiViewCNN [Su et al., 2015], PointNet [Qi et al., 2017a], PointNet++ [Qi et al., 2017b], PointCNN [Li et al., 2018], PointSIFT [Jiang et al., 2018], superpoint graph (SPG) [Landrieu & Simonovsky, 2018], RandLA-Net [Hu et al., 2020], and many more.

Generally, deep learning methods for point cloud classification can be categorized into five major types: projection-based methods, voxel-based methods, point-based methods, graph-based methods, and attention-based methods.

Projection-based methods

The core idea of projection-based methods is to project points from the 3D Euclidean space to 2D planes or manifold space so that the projected data can utilize CNN approaches designed for 2D data. Values (e.g., grayscales, intensities, or densities) of the projected data will represent either geometry (e.g., elevations) or attributes (e.g., RGB colors) of original 3D points. The most commonly used 2D projected representation is imagery. The 2D rendered image derived from virtual cameras of different viewing positions [Su et al., 2015] is an example. Through the use of multiple 2D rendered images, the 3D geometry of an object can be delineated. The rendered 2D images of an object are then applied to a 2D CNN network for object classification. According to reported studies, projection-based methods have demonstrated success in various classification applications using large-scale LiDAR point clouds. In Yang et al. [2017b], geometric features and full-waveform attributes were generated from local neighboring points of each point and assigned with $x-$ and $y-$ coordinates to a pixel in a 2D image. The generated 2D images encapsulated either geometric and radiometric information and were then fed into 2D CNNs. With the predicted labels of 2D pixels and a back-projection, the labeling of 3D points could be achieved. In Yang et al. [2018], based on the previous strategy of [Yang et al., 2017b], a multi-scale CNN and 2D images of features extracted from neighborhoods of various scales were developed, which obtained a better performance of classification. In Boulch et al. [2018], snapshot images with pixels, in which RGB colors and depths were encoded, were applied to a 2D fully convolutional network. Once the labels of 2D pixels were obtained, they were back-projected to the original 3D space to fulfill 3D point classification eventually. In Zhao et al. [2018], 2D images of multi-scale contextual information, including features of height, intensity, and roughness, were attained to represent the original 3D LiDAR points. Then, a multi-scale CNN was applied to conduct a classification of 3D points using these 2D images. Apart from 2D images, a digital surface model (DSM), as 2.5D data, is also adopted to represent 3D points of ALS data, since points in ALS data always reveal an even distribution in horizontal directions and lack of vertical distribution. In Chen et al. [2017], the DSM generated from 3D points was utilized as one input, fed into a two-stream deep neural network. Generally speaking, the neural network design based on the projection method is directly inherited from the existing 2D CNN solutions, and there is almost no need to adjust the network structure. However, these methods are deficient in presenting information from the depth direction and inevitably cause errors in rendering and interpolation.

Voxel-based methods

Voxel-based approaches structure the 3D space into regular voxel grids and project discrete 3D points into these voxels. Then, 3D points will be represented by the spatial occupancy of voxels so that a 3D convolution with a cubic template can be applied. VoxNet [Maturana & Scherer, 2015] is one early example, which directly transformed points to 3D voxels assigned with occupancy and implemented a 3D CNN to predict class labels of 3D objects. In Engelcke et al. [2017], as

an improvement, point clouds were voxelized into grid structures, then the voting procedure was introduced in a 3D CNNs. Similar to grey values of 2D pixels, occupancy is the most commonly used attribute that could be assigned to 3D voxels. However, there are also some other strategies to represent points with voxels. For instance, in Wang & Posner [2015], values of voxels were encoded by attributes generated from spatial positions of all points within this voxel. Besides, irregular-shaped 3D grid structures (e.g., voxels may have different sizes or cuboid shaps) can also be used for organizing 3D points. For example, the octree structure was introduced to CNNs in Wang et al. [2017], wherein normal vectors of points in each leaf node were averaged as voxel values and then fed into CNN. Similarly, Kd-trees were also utilized to structure discrete 3D points [Klokov & Lempitsky, 2017]. Voxel structures can also combine with pixels. For example, in Qin et al. [2019], both voxels and pixels were used as inputs in the proposed VPNet for semantic labeling of ALS data. The generation of voxels provided contextual information from the local area. On the contrary, auxiliary structures can be utilized to assist voxels as well. For instance, in Zhou & Tuzel [2018], unified features of every voxel were extracted via an additional feature encoding layer based on the region proposal network. These features would tackle the sparsity of 3D points. In Qi et al. [2016], two aforementioned schemes (i.e., voxel-based convolution and feature encoding layers) were combined into a multi-orientation volumetric CNN. The features of each orientation were generated with a shared network, and results of an image-based CNN were also integrated. In Su et al. [2018], sparse bilateral convolutional layers directly operated on a collection of points represented as a sparse set of samples in a high-dimensional lattice. The efficiency can be maintained using indexing structures to apply convolutions only on occupied parts. Choy et al. [2019] introduced a 4D CNN for spatio-temporal perception, and a hybrid kernel was proposed to deal with the exponential increase of parameters in using a 4D hypercube. However, similar to the shortcomings of projection-based methods, the voxelization process, either regular-shaped or irregular-shaped ones, definitely lead to a loss of spatial information since this is a sampling process. In this case, aliasing is inevitable due to the setting of the voxel resolution. Moreover, points of different categories may be rasterized into a voxel with the same label, which adds ambiguity and decreases the accuracy. The 3D structure requires considerably larger memory consumption and computational cost than 2D images, which hinders applying and developing this method.

Point-based methods

Point-based methods use discrete points as input to networks. As a milestone, the emergence of PointNet [Qi et al., 2017a] started the trend of directly using discrete points in deep neural networks. PointNet and its variants [Qi et al., 2017b, 2018] showed remarkable performance on popular benchmarks with either indoor [Armeni et al., 2016; Dai et al., 2017] or outdoor [Geiger et al., 2012; Hackel et al., 2017; Zhang et al., 2019b] applications. One of the key innovations of PointNet is to consider the unstructured and disordered characteristics of 3D points through a transformation and align points into the same orientation frame. Thus, it can establish an end-to-end framework in which 3D points can be classified without preprocessing. Moreover, for each point, local and global features were considered and learned in PointNet. In Yousefhussien et al. [2018], the multi-scale frame was developed to embed PointNet, for achieving a classification of large-scale ALS point clouds. In Li et al. [2018], PointCNN with a new network structure was proposed to learn an X transformation of points to deal with permutations and centralization. In Jiang et al. [2018], PointSIFT encoded 3D information of point orientations, then the module was embedded in the multi-scale frame of PointNet++. In Thomas et al. [2019], KPConv used a newly designed point convolution, which adapted kernel points to local geometry, in which weights for kernel positions are defined by linear correlation. In Boulch [2020], ConvPoint replaced the discrete kernels by continuous ones. Zhang et al. [2019a] proposed an efficient convolutional operator which utilized statistics from concentric spherical shells to define local representative

features. The RandLA-Net method proposed an efficient and lightweight network architecture that involved a random point sampling strategy and a novel local feature aggregation module [Hu et al., 2020]. In Li et al. [2020a], a geometry-attentional network was designed, which improved PointSIFT by embedding dense hierarchical structure and elevation-attention module. In this work, low-level geometric vectors were introduced to induce the learning of high-level local pattern representation, which increased the discrimination of geometric awareness of features. In Li et al. [2020b], a density-aware convolution module was introduced to directly work on 3D point sets and deal with uneven density distribution of 3D point clouds. Additionally, a context encoding module was designed to regularize the global semantic context. In Wen et al. [2020], a directionally constrained fully convolutional neural network utilized a novel directionally constrained point convolution module to encode local context in an orientation-aware way by considering projected 2D receptive fields. Moreover, point-based methods are also used as an encoder for extracting deep features, which can be integrated with other optimization algorithms. For instance, in Huang et al. [2020b], PointNet++ with hierarchical data augmentation was proposed to learn deep features of points and then optimized by a manifold-based feature embedding.

Graph-based methods

Rather than directly using discrete points as input, points, as well as the contexts, can be structured by a graph. A graphical model could naturally represent the spatial space of 3D scenes [Landrieu et al., 2017]. The graph-structured data is then fed into a newly designed network. GraphCNN is an encouraging instance, which has shown promising results on different applications [Simonovsky & Komodakis, 2017; Landrieu & Simonovsky, 2018; Wang et al., 2018]. In the graph-structured data, the edges between the points are created for generating the topology of the graph.

Attention-based methods

Recently, the attention mechanism is becoming increasingly popular, as it can provide scores of importance for parameters. The attention helps in improving the discriminate features and suppressing interference. In Fu et al. [2019], a dual attention network was proposed to integrate local features with their global dependencies adaptively. The feature aggregation is achieved by a weighted sum of the features at all positions. Meanwhile, channel attention was also applied to learn the interdependencies between feature channels. As for semantic segmentation of point clouds, in Feng et al. [2020], a point-wise spatial relation module was introduced to learn the dependencies of all points. In Li et al. [2020a], elevation-based attentions were learnt as an activation map for the deep features used for classification. Meanwhile, some work utilizes the attention mechanism in scope for feature aggregation. In Hu et al. [2020], attention-based aggregation is applied to the local scope to integrate the neighboring features. However, most of the aforementioned attention modules only utilized relations as the weights to aggregate features. The global structural information could be further explored based on the relations.

Compared with classic classification methods using handcrafted features, deep-learning-based-methods show their advances in learning high-level feature representation and improving the discrimativeness of extracted features in the classification task. Among deep-learning-based methods, point-based approaches shows their strength due to the capabilities of dealing with geometric nature of point clouds directly. Although many state-of-the-art deep learning-based methods has achieved remarkable performance in many applications, there are still some aspects that could be further exploited. First, the local geometric characteristic of points can be further investigated using deep learning methods. Second, long-range relations may provide additional and rich information provided in large receptive fields. Third, the optimization of the deep embedded feature space could be exploited.

1.2.3 Change detection using point clouds

Change detection is to identify differences in the state of an object or phenomenon by observing it at different time points [Singh, 1989]. Generally, the change detection approaches can be categories into three classes concerning the representation type of the output for changes: (1) point-based changes; (2) voxel- or occupancy grid-based changes; (3) segment/object-based changes.

Point-based change detection

The most direct way of detecting changes between 3D data is a point-to-point comparison, which is also denoted as surface difference. Basgall et al. [2014] obtained changes by directly calculate differencing between LiDAR and stereophotogrammetric point clouds using the CloudCompare software. The changes of single buildings were detected by visual inspection. Kang et al. [2013] used the Hausdorff distance to calculate point-to-point distances to avoid local density variation issues for detecting changes between point clouds. Xu et al. [2015] utilized a point-to-plane surface difference map by merging and comparing two datasets. The changes were detected by applying context rules to the difference map. This method required heavy prior knowledge about the scene. Du et al. [2016] proposed an automatic method to detect building changes in urban areas using aerial images and LiDAR data which was coregistered using the ICP algorithm. In this method, height difference and grey-scale similarity were utilized as indicates for changes. Besides, the graph-cuts method was applied to further optimize the detected changes by considering contextual information. This method applied some thresholds for the detecting task, and the thresholds were set based on prior knowledge of the scene. In general, point-to-point distance is sensitive to point densities. Additionally, in point-based change detection, the high-level semantics are not considered, and the limitations of observation are also not tackled.

Voxel-based change detection

For voxel-based change detection methods, point clouds are structured to 3D grids or octree-based voxels, in which the changes can be conducted by comparing different occupancy states between point clouds. Pagac et al. [1998] utilized occupancy grids for constructing and maintaining a map of an autonomous vehicle's environment for the purpose of navigation, using evidential reasoning. The sensor readings are fused into the map using the Dempster-Shafer theory (DST). Wolf & Sukhatme [2004] applied a similar method for SLAM in a dynamic environment. The occupancy state is defined as free, unknown, and occupied. The changes can be obtained by comparing different states of occupancy. However, the aforementioned methods only studied occupancy modelling in 2D grids. Hebel et al. [2013] applied Dempster-Shafer theory (DST) to determine conflicting evidence along the laser pulse propagation path. Occupancy grids are utilized for tracing the propagation path in 3D space for each measurement. Meanwhile, additional attributes are taken to define the types of changes into account by considering different characteristics of man-made objects and vegetation. Xiao et al. [2015] further combined the occupancy-based change detection with a point-to-triangle distance-based method to conduct direct consistency evaluation on points. This combined method tackled irregular point densities and occlusion problems. Gehrung et al. [2016] proposed a framework for the volumetric modeling and visualization of large-scale urban environments. However, the usual raycasting-based methods bring artifacts caused by the traversal of negative discretized space. Gehrung et al. [2018] utilized knowledge about planar surfaces to prevent the type of artifacts. In Gehrung et al. [2019], a Delta Octree was utilized for encoding changes between epochs, which improved the processing efficiency of detecting changes. Although the voxel-based change detection methods consider tackled the occlusion problems, while the changes of high-level semantics are not considered.

Segment-based change detection

Segment-based methods utilize segmentation techniques to extract clusters, and these clusters serve as basic units for detecting changes. The post-classification method is regarded as a popular method since it transforms the direct geometric or spectral comparison to label changes, which tends to be more robust toward disturbances induced by acquisition conditions (such as season, luminance differences), on the other hand, it is able to provide a type change matrix. However, in most cases, the change detection results of this method highly depend on the classification or object detection results, which subsequently requires careful sample collection and feature design. Vosselman et al. [2004] detected and updated building changes in a 2D map using laser scanning data. After segmentation and filtering bare earth points, the object points were classified as buildings or vegetation based on surface roughness, segment size, height, color, and first-last pulse difference. The building segments were compared with the building objects on 2D maps for change detection. Aijazi et al. [2013] classified point clouds into two main object classes: permanent and temporary. Different natural or human-made changes occurring in the urban landscape over this period are detected and analyzed using cognitive functions of similarity, and the resulting 3D cartography is progressively modified and updated accordingly. Schachtschneider et al. [2017] assessed the behavior of each segment in the scene from temporal objects from the global occupancy grid based on the segmentation results of point clouds. A region-growing algorithm achieved the segmentation of the point clouds.

For point-based change detection, methods rely highly on strong prior knowledge on the scene, which limits the generalization of the methods. In addition, point-based change detection is comparatively less robust to change of details, i.e., point densities or illuminance. As for voxel-based change detection, although in some occupancy-based change detection methods the occlusions are addressed when detected changes and the different characteristics of objects are considered, the work on semantic changes of complex urban scenes with high occlusions is quite limited. Segment-based change detection can present high-level changes but also highly rely on the preprocessing results. The improvement of the semantic segmentation process and the consideration of missing information could be further investigated.

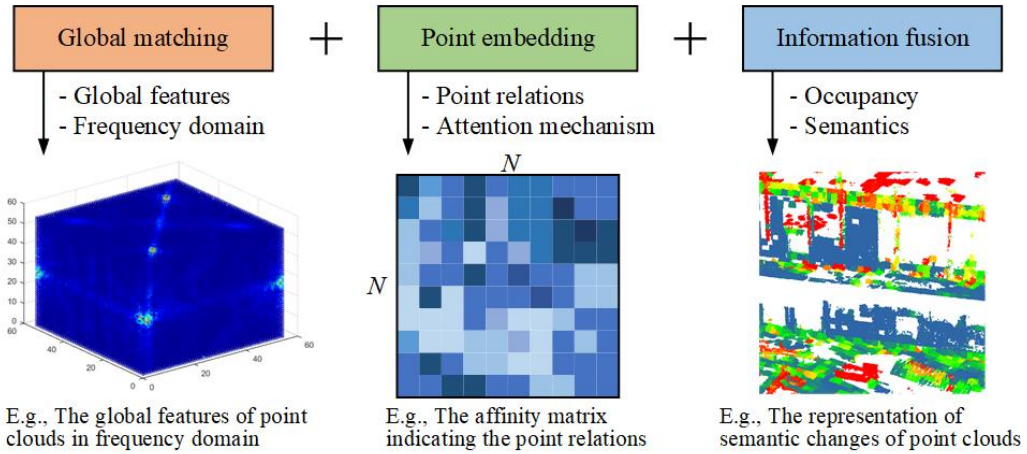

Figure 1.4: The proposed solutions for the research questions, including three aspects: (1) global matching achieved by utilizing global features and attributes in frequency domain; (2) point embedding considering point relations and the attention mechanism; (3) information fusion by fusing the geometric occupancy and semantic information.

1.3 Objectives and contributions

In this thesis, we present a framework for detecting changes from construction sites and urban scenes using 3D point clouds, with a sequence of novel algorithms and methods in the fields of registration, semantic segmentation, and change detection. We aim to develop robust methods and techniques to acquire and present changes of structural components of buildings in the construction scene during the construction process. Additionally, the specific concerns of the three tasks will be addressed, and the pros and cons of the proposed methods will be discussed. The possible further solutions will be provided as possible future work.

To provide answers for the research questions in Section 1.1, we proposed solutions (see Fig. 1.4) that focus on aspects of global matching, point embedding, and information fusion, which can help us solve the task of detecting changes in construction sites. With these solutions, a wide range of algorithms and methods were developed. In Fig. 1.5, we illustrate an overview of these methods, containing former works from other researches and our proposed methods and their developing routes.

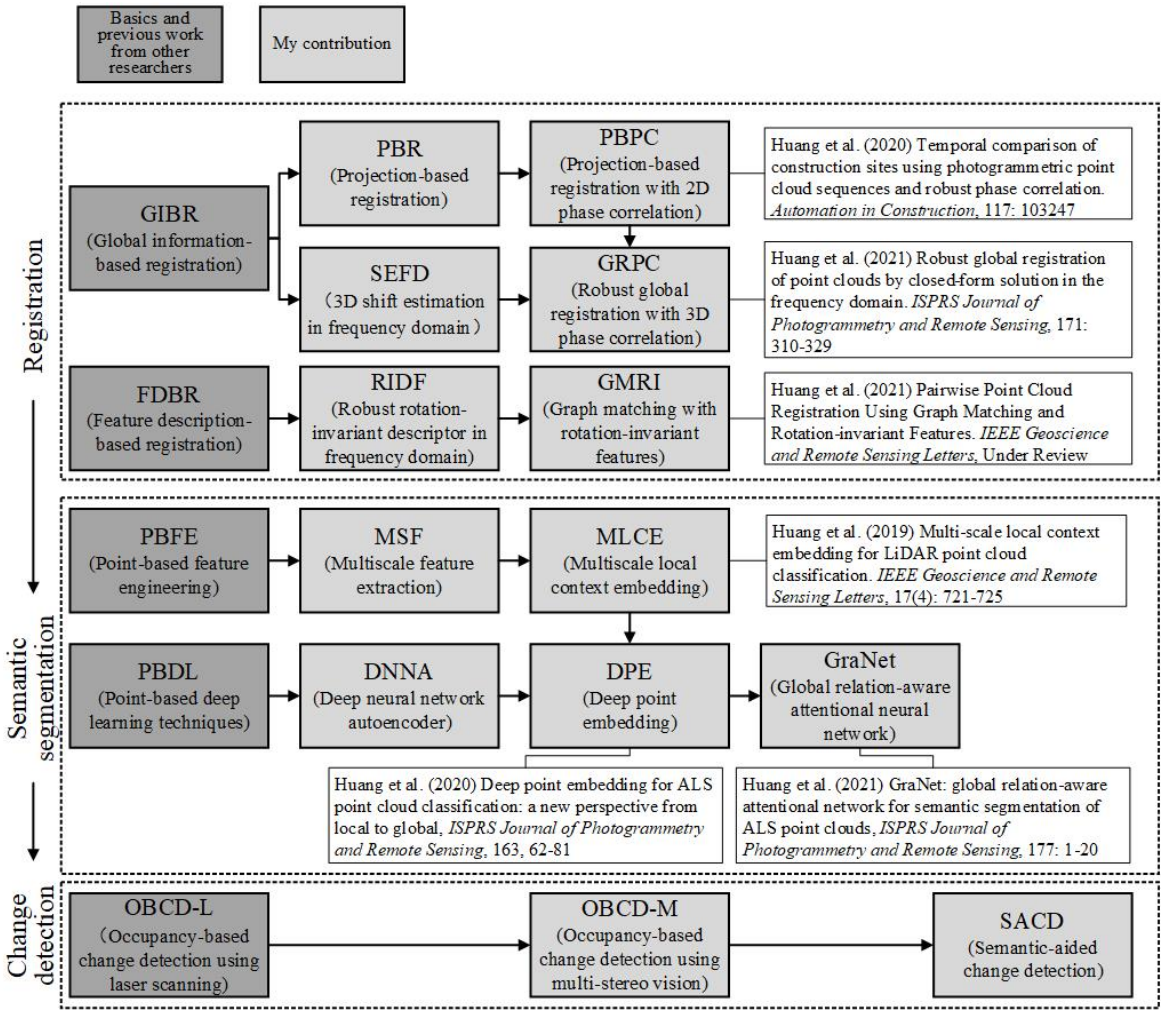

Figure 1.5: A diagram of algorithms and methods with involved publications of solved tasks, core strategies, and their relations.

1.4 Structure and organization

This thesis is organized as follows: Chapter 2 presents the theoretical basics of Fourier-based image registration, manifold learning-based dimensionality reduction, point-based deep learning operations, the voxel-based structure of point clouds, and occupancy-based change detection. Chapters 3-5 describe the core parts of this thesis, namely the methods for solving the aforementioned research questions. Chapter 6 presents the experiments, including the datasets and the evaluation metrics. Chapter 7 presents the experimental design, results, and analysis on the results. Chapter 8 presents the discussion on the three main tasks involved in the progress monitoring and change detection of construction sites using point clouds. Chapter 9 finalizes this thesis by presenting the conclusions and providing outlooks for future works.

2 Basics

In this chapter, we introduce specific techniques and algorithms about the Fourier-based image registration, manifold learning-based dimensionality reduction, point-based deep learning techniques for semantic segmentation, the voxel-based data structure, and the occupancy-based change detection. Our methods are developed based on similar concepts of these techniques. To be specific, for the task of point cloud registration, our methods are developed based on 3D extension of Fourier-based image registration and provide our solution for robust phase correlation. As for semantic segmentation, manifold learning-based methods are used for dimensionality reduction, which is vital in feature engineering. Our method improves the local manifold learning-based methods by considering spatial constraints and provides a solution for large-scale data. Deep learning techniques are also popular and advanced techniques on semantic segmentation. Our methods tend to make an improvement based on the concept of point-based deep learning techniques by considering long-range relations and involving an attention mechanism. Raycasting-based change detection provides a solution for determining changes considering the unseen space. Our method aims at considering the unseen space but with a different type of point clouds and makes an improvement by considering semantic information.

2.1 Fourier-based image registration

In this section, we will introduce image registration using Fourier-based method. The principle of phase correlation, different solutions for robust shift estimation based on phase correlation, and the application to image registration will be explained in the following sections.

2.1.1 The principle of phase correlation

Before introducing the robust shift estimation based on phase correlation, a short introduction to phase correlation will be given. Compared with some other commonly used correlation-based methods, phase correlation tends to be more accurate and efficient. The general idea of phase correlation is that any translation between two relevant images in the spatial domain can be represented as a phase shift in the frequency domain. Assume that two images are related to each other by shifts in the column and row direction denoted as x_0 and y_0, respectively, and the relation can be presented by:

$$s(x, y) = r(x - x_0, y - y_0), \tag{2.1}$$

where $s(x, y)$ and $r(x, y)$ represent the two image in spatial domain. Then, a discrete Fourier transform (DFT) can be conducted on these two images to transform them into the frequency domain. Afterward, the relation between the Fourier transforms can be written as:

$$S(u, v) = R(u, v)e^{-i(ux_0+vy_0)}, \tag{2.2}$$

in which $S(u, v)$ and $R(u, v)$ are the corresponding Fourier transforms of $s(x, y)$ and $r(x, y)$. The normalized cross-power spectrum can be represented as:

$$Q(u, v) = \frac{S(u, v)R^*(u, v)}{|S(u, v)R^*(u, v)|} = e^{-i(ux_0+vy_0)}, \tag{2.3}$$

in which R^* is the complex conjugate of R, and the magnitude of Q is normalized to 1. The inverse Fourier transform (IFT) of $Q(u, v)$ is a Dirac delta function centered on (x_0, y_0). Thus, the translation can be estimated by finding the peak coordinates of this function, as shown in Fig. 2.1. However, this solution can only provide results in integer pixel, which is not precise enough. To get subpixel estimation, one way is to get precise determination of the main peak location of the IFT of the normalized cross-power spectrum, but this method is sensitive to noise.

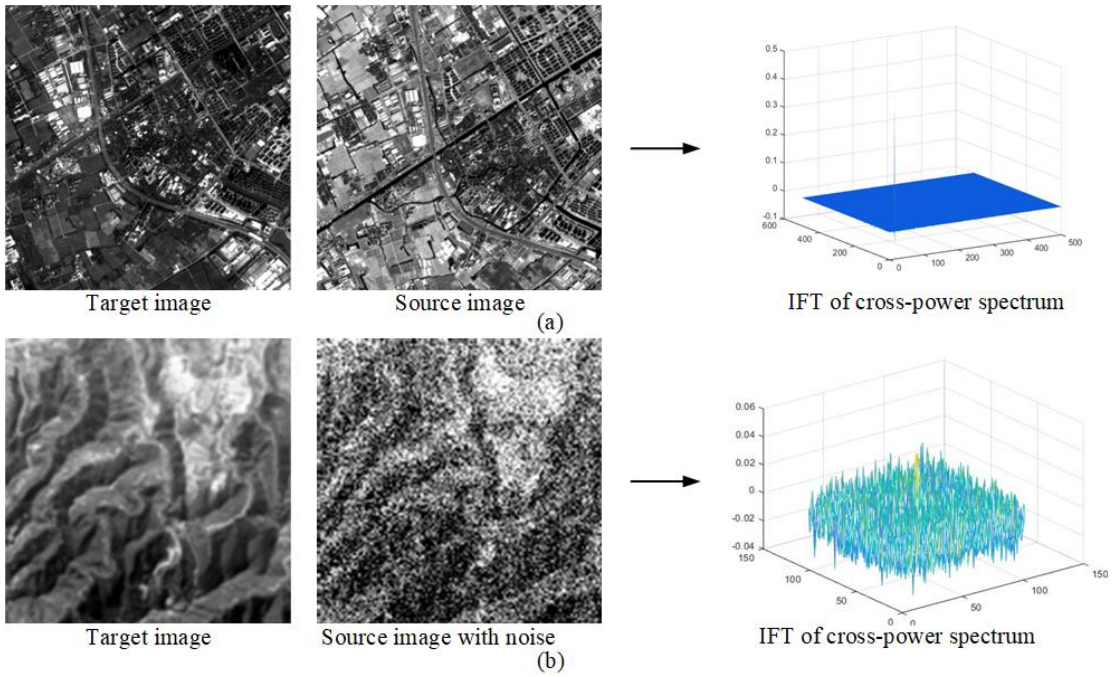

Figure 2.1: Illustration of the phase correlation. a) Phase correlation with image pair, b) phase correlation of image pair with noise.

2.1.2 Robust shift estimation

To obtain robust and precise shift estimation, two major solutions are usually applied (see Fig. 2.2). One is Stone's solution [Stone & Cheok, 2001], in which the phase shift angle is treated as a linear function of the shift parameters. By employing plane-fitting, the shifts can be estimated with the removal of high-frequency components. Fig. 2.2 illustrates the workflow of Stone's solution. Since the phase different angle is a linear function of the shift parameters, and it is defined by:

$$\angle Q(u, v) = ux_0 + vy_0. \tag{2.4}$$

We try to solve the linear function for the 2D plane for estimating the shifts.

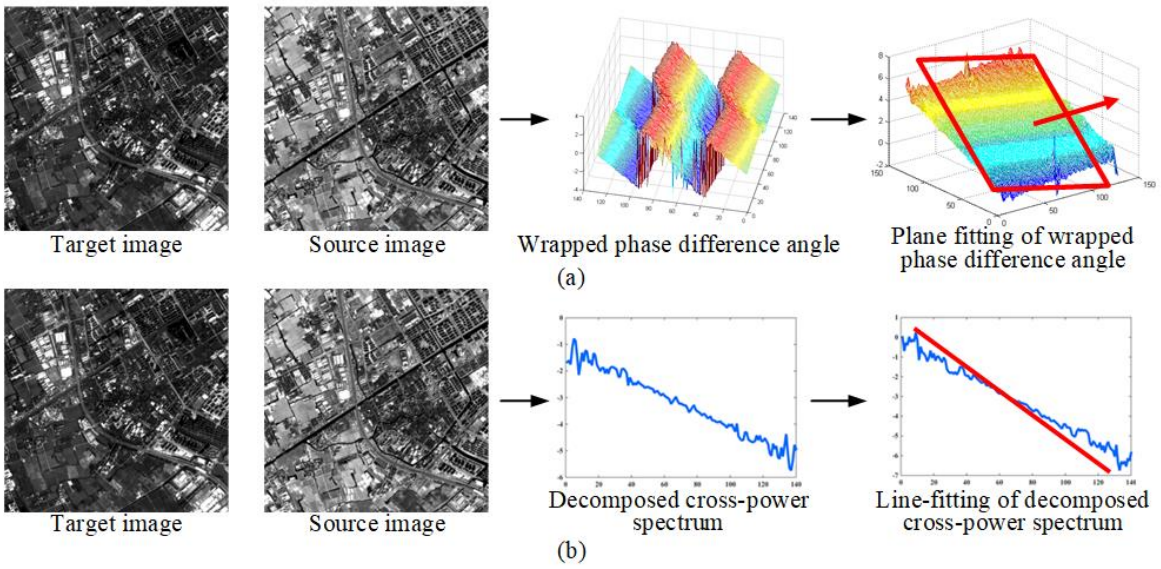

Figure 2.2: Illustration of two solutions for robust phase correlation. a) The procedures of Stone's solution, b) the procedures of Hoge's solution.

The other one is Hoge's solution [Hoge, 2003]. The idea is to represent the normalized cross-power spectrum by two rank-one signals. The normalized cross-power spectrum can be separated as:

$$Q(u,v) = e^{-i(ux_0+vy_0)} = e^{-iux_0}e^{-ivy_0} = Q_{x0}(u)Q_{y0}(v). \tag{2.5}$$

Thus, the problem of the shift estimation can be simplified by investigating the rank-one signals. First, the rank-one signals can be approximated by the singular value decomposition (SVD) method. By utilizing the decomposed signals, the problem of 2D phase unwrapping and high-frequency components eliminating can also be simplified into finding a 1D solution, which is much more robust and also avoids the ill-posed problems when noise is intense. However, the aforementioned robust phase correlation only considers shift estimation. In real application, the registration of images should also involves the estimation of rotation and scaling. Based on the characteristics of Fourier transform, the robust phase correlation can be extended to a full registration strategy with rotation, scaling, and translation using Fourier-Mellin transform (FMT).

2.1.3 Fourier-Mellin transform

Let $f(x,y)$ and $g(x,y)$ represent two relevant images in spatial domain, and these two images are related to each other by the rotation θ_0, scaling ψ, and the translation $\mathbf{t} = [t_x, t_y]$. The relation can be written as:

$$f(x,y) = g(\psi(x\cos\theta_0 + y\sin\theta_0) - t_x, \psi(-x\sin\theta_0 + y\cos\theta_0) - t_y). \tag{2.6}$$

The two images can be transformed into the frequency domain using a DFT algorithm. Let $F(u,v)$ and $G(u,v)$ be the Fourier transforms corresponding to images $f(x,y)$ and $g(x,y)$. Thus, the relation of the Fourier transforms of the corresponding images can be expressed by:

$$F(u,v) = \frac{1}{\psi^2}G(\frac{1}{\psi}(u\cos(\theta_0) + v\sin(\theta_0)), \frac{1}{\psi}(-u\sin(\theta_0) + v\cos(\theta_0)))e^{-j\theta_0(ut_x+vt_y)}. \tag{2.7}$$

In terms of the magnitude, this relation can be simplified to:

$$|F(u,v)| = \frac{1}{\psi^2}|G(\frac{1}{\psi}(u\cos(\theta_0) + v\sin(\theta_0)), \frac{1}{\psi}(-u\sin(\theta_0) + v\cos(\theta_0)))|. \tag{2.8}$$

We can see that the translation only influences the phase information, and the rotation and scaling are decoupled with the translation. To recover the rotation and scaling simultaneously, the Fourier magnitude spectrum is further transformed into a log-polar space by log-polar transformation (LPT), in which two Fourier magnitude spectrum are related to each other by a linear shift:

$$|F(r,\theta)| = \frac{1}{\psi^2}|G(\frac{r}{\psi}, \theta + \theta_0)|, \tag{2.9}$$

$$|F(\log r,\theta)| = \frac{1}{\psi^2}|G(\log r - \log\psi, \theta + \theta_0)|. \tag{2.10}$$

Thus, by finding the shift (x_0, y_0) between these two spectra in the log-polar space, the rotation and scaling can be estimated:

$$\psi = e^{x_0}, \theta_0 = y_0. \tag{2.11}$$

By applying FMT, the estimation of rotation and scaling of 2D images is transformed into a shift estimation problem.

2.2 Manifold learning-based dimensionality reduction

Manifold learning is a popular method which is widely used for dimensionality reduction and has shown promising results in the many fields, such as hyperspectral classification [Hong et al., 2017] and image reconstruction [Zhu et al., 2018]. Here, we present the basic concept of local manifold learning (LML) methods. LML methods aim to find the underlying local manifold structure that lies in a complex high-dimensional space and embed it in a lower dimensional space. Generally, LML methods comprise the following three steps: neighborhood recovery, computation of affinity weight, and calculation of embedding, that are illustrated in Fig. 2.3. Given $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathcal{R}^{N\times D}$, which denotes N samples with D dimensions, LML methods aim to embed these samples into a low-dimensional space represented by $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N] \in \mathcal{R}^{d\times N}(d << D)$. The calculation of the embedding can be formulated as:

$$\begin{aligned}\hat{\mathbf{Y}} &= \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{n} \sum_{j\in\phi_i} (\mathbf{y}_i - \mathbf{y}_j)^2 W_{ij} \right\}, s.t. \mathbf{Y}\mathbf{P}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \\ &= \arg\min_{\mathbf{Y}} \left\{ \mathbf{Y}\mathbf{L}\mathbf{Y}^{\mathrm{T}} \right\}, s.t.\ \mathbf{Y}\mathbf{P}\mathbf{Y}^{\mathrm{T}} = \mathbf{I},\end{aligned} \tag{2.12}$$

where $\mathbf{W}$ represents the affinity matrix, wherein W_{ij} represents the affinity weight of the ith and jth samples, and $j \in \phi_i$, where ϕ_i is a set of neighbors of the ith (the neighbors are usually defined by k-nearest neighborhood (KNN)). $\mathbf{L}$ is the Laplacian matrix, which is defined by $\mathbf{L} = \mathbf{W} - \mathbf{D}$, and $D_{ii} = \sum_j W_{ij}$. $\mathbf{P}$ is used to formulate the constraint defined by different LML methods.

2.2.1 Locally linear embedding

A popular example of LML methods is locally linear embedding (LLE) [Roweis & Saul, 2000]. LLE assumes a globally nonlinear but locally linear embedding; therefore, each point and its

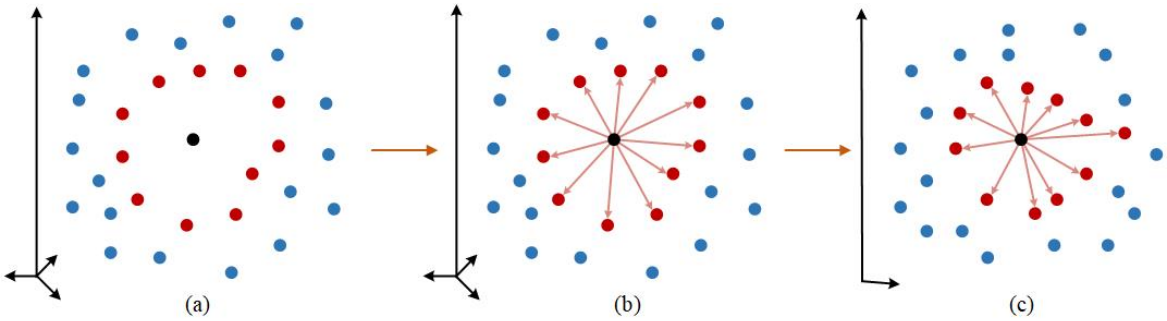

Figure 2.3: Illustration of LML methods. a) Neighborhood recovery, b) computation of affinity weight, c) calculation of embedding.

neighboring points can be embedded into a low-dimensional space by multiplying a linear reconstruction matrix. In this case, the reconstruction matrix can be obtained by minimizing the residual summation of squares for reconstructing each point from its neighboring points as:

$$\hat{\mathbf{R}} = \arg\min_{\mathbf{R}} \left\{ \sum_{i=1}^{n} ||\mathbf{x}_i - \sum_{j \in \phi_i} R_{ij}\mathbf{x}_j||_2^2 \right\} \text{s.t.} \sum_{j \in \phi_i} R_{ij} = 1, \tag{2.13}$$

where $\mathbf{R} \in \mathcal{R}^{N \times N}$ denotes the reconstruction matrix, and R_{ij} represents the reconstruction weight between the ith and jth samples.

Given the reconstruction matrix, we aim to find the d-dimensional coordinates by minimizing the following embedding cost function:

$$\hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{n} ||\mathbf{y}_i - \sum_{j \in \phi_i} R_{ij}\mathbf{y}_j||_2^2 \right\}, \tag{2.14}$$

which is subject to the following constraints:

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{N} \mathbf{y}_i &= 0 \\ \frac{1}{N}\sum_{i=1}^{N} \mathbf{y}_i\mathbf{y}_i^{\mathrm{T}} &= \mathbf{I}. \end{aligned} \tag{2.15}$$

Alternatively, the embedded coordinates can be calculated in a graph-embedding manner, and consequently rewritten as an optimization equation:

$$\begin{aligned} \hat{\mathbf{Y}} &= \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{n} ||\mathbf{y}_i - \sum_{j \in \phi_i} R_{ij}\mathbf{y}_j||_2^2 \right\}, s.t. \mathbf{Y}\mathbf{P}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \\ &= \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{n} \sum_{j \in \phi_i} ||\mathbf{y}_i - \mathbf{y}_j||_2^2 W_{ij} \right\}, s.t. \mathbf{Y}\mathbf{P}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \\ &= \arg\min_{\mathbf{Y}} \left\{ \mathbf{Y}\mathbf{L}\mathbf{Y}^{\mathrm{T}} \right\}, s.t. \mathbf{Y}\mathbf{P}\mathbf{Y}^{\mathrm{T}} = \mathbf{I}, \end{aligned} \tag{2.16}$$

where affinity weight matrix $\mathbf{W}$ can be derived from reconstruction weight $\mathbf{R}$ as:

$$W_{ij} = \begin{cases} R_{ij} + R_{ji} - R_{ij}R_{ji}, & \text{if } j \in \phi_i \\ 0, & \text{otherwise.} \end{cases} \tag{2.17}$$

Furthermore, Laplacian matrix L can be computed by

$$\begin{aligned}\mathbf{L} &= \mathbf{D} - \mathbf{W} \\ &= (\mathbf{I} - \mathbf{R})^{\mathrm{T}}(\mathbf{I} - \mathbf{R}).\end{aligned} \tag{2.18}$$

In the case of LLE, the constraint is assumed to be $\mathbf{P} = \mathbf{I}$. LLE is a feasible method that is capable of fully considering the local properties of the data. Consequently, the local manifold structure can be well learned and represented.

2.2.2 Locality preserving projections

In this part, we will review another widely-used local manifold learning (LML) method: locality preserving projections (LPP). LPP embeds the high-dimensional data or feature into a low-dimensional subspace in which the topological structure of high-dimensional data is locally preserved. Given a set of data samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{L \times N}$ with L dimension by N pixels, LPP linearly learns a mapping $\mathbf{A}$ to find the corresponding low-dimensional embedding $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N] \in \mathbb{R}^{D \times N}$ ($D << L$). This process can be modeled in the following:

$$\sum (\mathbf{y}_i - \mathbf{y}_j)^2 \mathbf{W}_{ij}, \tag{2.19}$$

where $\mathbf{W}_{ij}$, which is an adjunct matrix, can be defined as:

$$\mathbf{W}_{ij} = \begin{cases} \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2 / \sigma^2) \\ 0 \end{cases}, \tag{2.20}$$

where σ denotes the standard deviation of Gaussian kernel function.

To enhance the model's interpretability and transferability, Eq. 2.19 can be approximately modeled in a linearized way. Suppose $\mathbf{y}^T = \mathbf{a}^T \mathbf{X}$ where $\mathbf{a}$ is a linear projection, thus Eq. 2.19 can be simplified as follows:

$$\begin{aligned}\sum (\mathbf{y}_i - \mathbf{y}_j)^2 \mathbf{W}_{ij} &= \sum (\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j)^2 \mathbf{W}_{ij} \\ &= \sum \mathbf{a}^T \mathbf{x}_i \mathbf{D}_{ii} \mathbf{x}_i^T \mathbf{a} - \sum \mathbf{a}^T \mathbf{x}_i \mathbf{W}_{ij} \mathbf{x}_j^T \mathbf{a} \\ &= \mathbf{a}^T \mathbf{X}(\mathbf{D} - \mathbf{W})\mathbf{X}^T \mathbf{a} = \mathbf{a}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{a},\end{aligned} \tag{2.21}$$

where D is a diagonal matrix and $D_{ij} = \sum_j \mathbf{W}_{ij}$. $\mathbf{L}$ is the Laplacian matrix computed by $\mathbf{L} = \mathbf{D} - \mathbf{W}$. To avoid the trival solution, a necessary constraint formulated by $\mathbf{y}^T \mathbf{D} \mathbf{y} = 1$ is forced in the process of solving Eq. 2.21, and its linearized version can be written as:

$$\mathbf{a}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{a} = 1. \tag{2.22}$$

Accordingly, the variable $\mathbf{a}$ can be estimated by minimizing the following objective function:

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \left(\mathbf{a}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{a} \right). \tag{2.23}$$

The solution of Eq. 2.23 can be equivalently obtained by solving the following generalized eigenvalues decomposition problem:

$$\mathbf{X} \mathbf{L} \mathbf{X}^T = \lambda \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{a}. \tag{2.24}$$

2.3 Point-based deep learning operations

In this section, we will introduce two deep-learning based methods which directly work on 3D points for classification and semantic segmentation. The first one is PointNet [Qi et al., 2017a], and the other one is PointNet++ [Qi et al., 2017b].

2.3.1 PointNet

We first present the basic idea of PointNet. PointNet proposes a solution for spatial encoding considering the raw nature of 3D points, thereby enabling the processing of 3D points in a direct manner.

Given a sequence of unordered points $\mathbf{X} = \{x_1, x_2, ..., x_n\}$, a function α is defined to map a set of points to a feature vector:

$$f(\mathbf{X}) = \alpha(max_{i-1}^{n}(\beta(x_i))), \tag{2.25}$$

where α and β usually represent multi-layer perceptron (MLP) network. The function f is invariant to the input permutations. β can be treated as a spatial encoding for a point.

Compared to other previous proposed deep-learning-based methods, the network of this method is capable of operating on 3D point clouds directly, and it is invariant to the permutations of input points. The process of classifying points into specific categories is greatly simplified into an end-to-end method without preprocessing steps such as rasterization. The key aspect of PointNet is learning global features from a set of points. However, it lacks the capability of capturing local context at different scales.

2.3.2 PointNet++

As an improvement of PointNet, PointNet++ was derived from PointNet by adding hierarchical strategy to enhance the capability of encapsulating local information and the efficacy of feature expression in urban scenes with high complexity. The aim of PointNet++ is to learn multilevel local contextual information progressively. Typically, as illustrated in Fig. 2.4, first, the original inputs are grouped into a set overlapped sub-pointsets. Then local features are learned from these local neighborhoods to produce low-level local features. Subsequently, these local features in the lower level can be utilized for higher-level feature learning with the same procedure. These grouping and learning procedures are repeated until the global features of all input points are obtained. While dealing with the segmentation task, the features for all the original points are extracted. In this case, a hierarchical propagation strategy is adopted by propagating the features extracted from subsampled points back to the original ones.

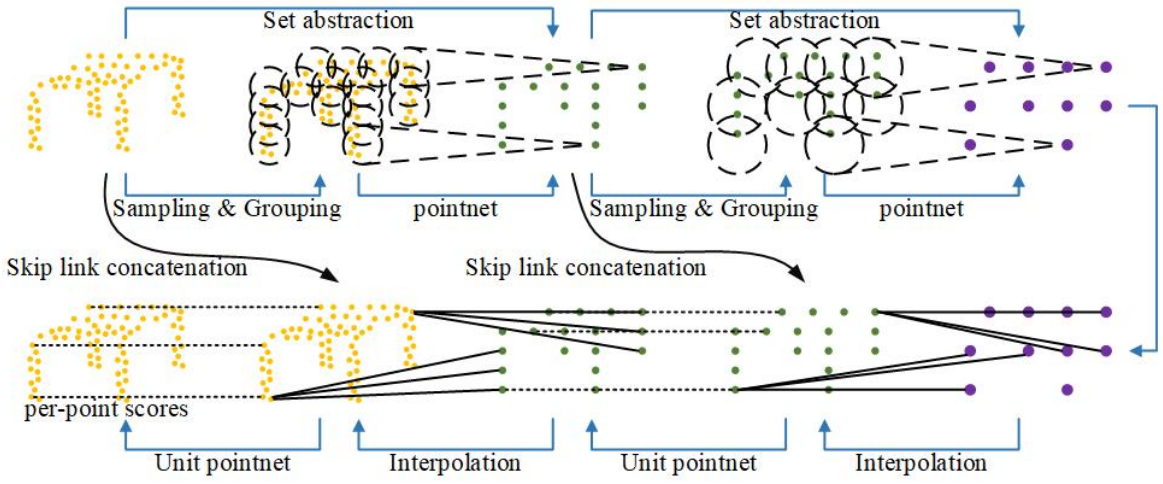

Figure 2.4: Illustration of the hierarchical architecture of PointNet++.

2.4 Voxel-based structure of point clouds

Voxelization refers to a method of structuring point clouds and coverting point clouds to voxel grids with points [Xu et al., 2021]. There are two major methods for voxelization of 3D space: octree-based voxelization and grid-based voxelization.

2.4.1 Octree-based voxelization

In Vo et al. [2015], the input point cloud can be organized with an octree strucutre, which is a tree data structure in which each internel node has eight children. It means that for each node the space is subdivided into eight subspaces. A illustration of the octree structure is presented in Fig. 2.5. By using the octree structure, unordered and unstructured 3D points in a point cloud can be organized in a strucuted way, which improves the efficiency of searching and traversing occupied space.

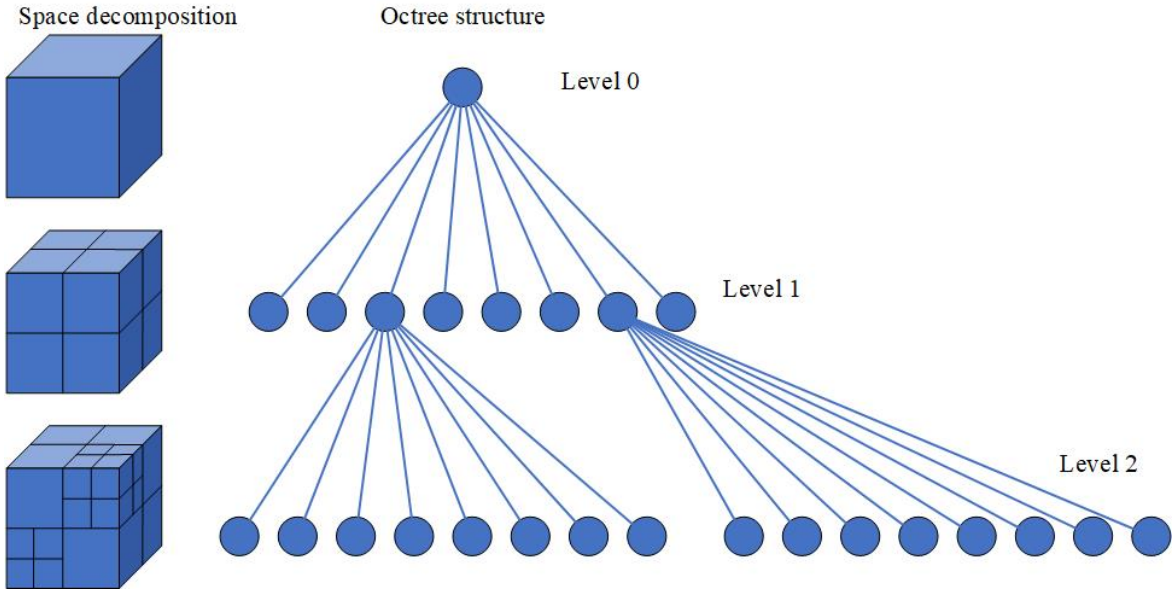

Figure 2.5: Illustration of the octree structure.

2.4.2 Grid-based voxelization

The other voxelziation method is to organize point clouds using 3D grid. Fig. 2.6 illustrates the voxelization and binarization of a given 3D point cloud. A voxelization process is presented to transform the unstructured and unordered points to a regularly resampled discrete 3D grid. Differing from the voxelization step in the other previous work, in which only the point cloud is voxelized, instead, the 3D space covering the entire point cloud is voxelized and resampled. The centers of all these voxels will be utilized to represent the point cloud and serve as basic input elements for the further process. Then, a binarization process is conducted on the resampled 3D grid, in which binary values (i.e., zero or one) are annotated to each voxel. The binary values actually denotes the occupancy of each voxel. It means that if points whose number is above a threshold fall into a voxel, the voxel will be marked as value one. Conversely, if a voxel contains limited number of points, it will be annotated with value zero. The threshold is actually set to filter out some isolated points. The thresholds are identified according to point densities of point clouds. In case that the bounding box of the point cloud is not a cubic, a zero-padding should be done, ensuring three dimensions are of the same sizes. The position and assigned labels of voxels

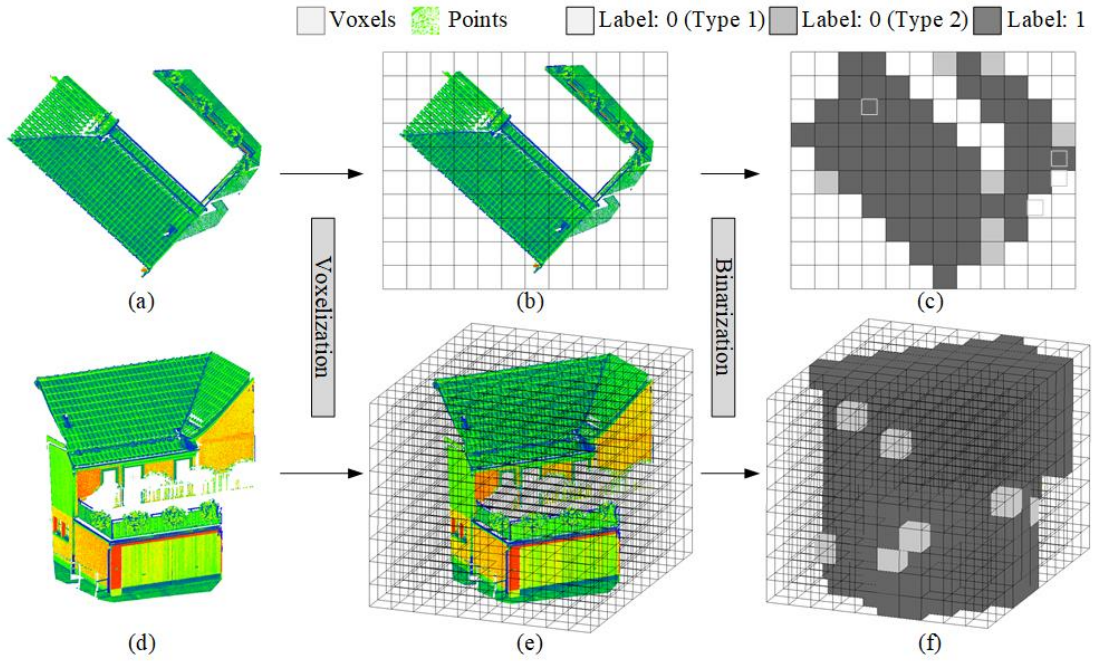

Figure 2.6: Voxelization and binarization of point clouds. a) Original point clouds from top view and d) from oblique view, b) voxelized point clouds from top view and e) from oblique view, c) binarized voxels from top view and f) from oblique view. It should be noted that Type 1 denotes the empty voxels annotated with value zero and Type 2 denotes the voxels with limited numbers of points and annotated with value zero.

will be used as inputs for further steps. In this way, an unevenly distributed point cloud can be resampled to a cubic grid, which represents the spatial distribution of point clouds and whose basic elements represent the corresponding point occupancy.

2.5 Occupancy-based change detection

Occupancy-based change detection allows for the handling of occlusions and changes implicitly, in which changes can be identified by conflicts of occupied space and empty space along the direction of laser beam [Hebel et al., 2013]. Here, we present the basic concept of occupancy-based change detection methods using LiDAR data.

2.5.1 Processing stages of LiDAR data analysis

In occupancy-based change detection, some prerequisites should be full-filled for data analysis, including the access to the component's raw measurements, i.e., the range data, the scanning geometry, and the IMU/GNSS trajectory. In general, the data acquisition and preprocessing can be distinguished as two different stages: (I) the creation of the reference database; (II) change detection based on current measurements.

In stage (I), the creation of reference data consists of the combination of multiple measurements and the optimization of raw data, including the correction of positioning errors. Hebel & Stilla [2012] presented the method for object-based analysis and registration of ALS measurements. In the object-based analysis, by computing the local principal components and applying a combination of a region growing method and RANSAC plane fitting, planar shapes can be seg-

mented and separated from clutter objects, i.e., bushes or trees. The segmentation results were latter used as support for the change detection process.

In stage (II), three major steps are conducted for the new LiDAR measurements, including (i) categorization, (ii) alignment, and (iii) comparison to the reference database. For step (a), a fast segmentation method is applied by scanline segmentation and grouping of line segments [Hebel & Stilla, 2008]. Then, the segmented planar objects can be used for alignment of the current data and the reference one [Hebel & Stilla, 2010]. Step (iii) is the most important part for change detection. We will explain it in details in further sections.

2.5.2 Generation of the database

For each measurement, it can be expressed as following considering the navigation information:

$$\mathbf{p} = \mathbf{s} + R_N R_B R_S \mathbf{r}, \tag{2.26}$$

where $\mathbf{p}$ denotes the laser point, $\mathbf{s}$ is the 3D position of the laser scanner, R_N is the rotation matrix describes the orientation provided by IMU, R_B denotes the relative orientation of IMU and laser scanner, R_S describes the scanning geometry, and $\mathbf{r}$ denotes the distance measurement provided by laser scanner. The equation can be simplified as:

$$\mathbf{p} = \mathbf{s} + \mathbf{r}, \tag{2.27}$$

where $\mathbf{r}$ considers the distance measurement and orientation information. Different from classic occupancy-based methods, Hebel et al. [2013] evaluated the occupancy of 3D space directly on the exact positions of the measured 3D points. The gird structure is only used for searching. Specifically, in stage (I), the information about the proximity of laser beams and points is assigned to a 3D grid that covers the entire scene. In stage (II), the grid structure allows for evaluating whether the current measurement confirm or contradict previous information.

In stage (I), for each laser pulse, the origin and the measured range are stored in an indexed list $\mathbf{L}$. If there are multiple echoes for a single laser pulse, there will be multiple entries in the indexed list $\mathbf{L}$. As shown in Fig. 2.7, 3D grids are filled with indices of $\mathbf{L}$. The index will be assigned to the cell according to the 3D position of the laser point. $\mathbf{V}_P$ illustrates the an index-based rasterization, while $\mathbf{V}_R$ is utilized for storing all the indices of a laser beam that go through grid cells, which is achieved by Bresenham's algorithms [Bresenham, 1965]. Thus, we could get 3D grids with laser indices stored in each cell.

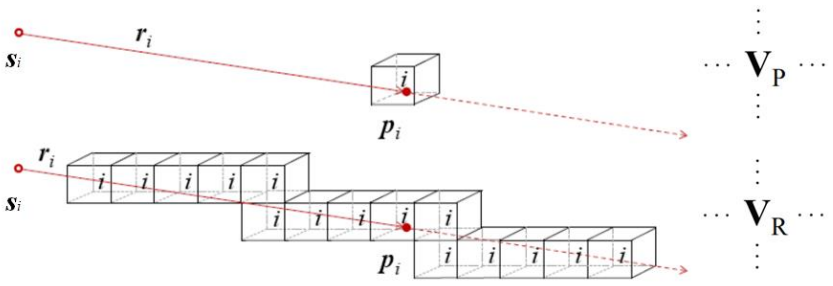

Figure 2.7: Filling of the voxel grids $\mathbf{V}_P$ and $\mathbf{V}_R$. i: index; $\mathbf{s}_i$: sensor position; $\mathbf{p}_i$: laser point; $\mathbf{r}_i$: range measurement [Hebel et al., 2011].

2.5.3 Occupancy modeling from a single measurement

Dempster-shafer theory (DST) is commonly used for data fusion. It has the advantages of assessing conflicting information implicitly and allows for combining evidence commutatively and associatively, which is essential for dealing with objects scanned from different perspectives and datasets acquired at different epoches. The state of occupancy can be represented by a universal set $U = \{empty, occupied\}$. The power set 2^U of U is given as the set $\{\emptyset, \{empty\}, \{occupied\}, U\}$, which contains all possible states. According to DST, a belief mass within range $[0, 1]$ is assigned to each element in the power set. In addition, the empty set $\emptyset$ has zero mass, and the the masses of the remaining set are summed to one:

$$m : 2^U \rightarrow [0, 1], m(\emptyset) = 0, \sum_{A \in 2^U} m(A) = 1. \tag{2.28}$$

An assignment that obeys the aforementioned rules is called as "basic belief assignment". DST makes use of the mass assignment and sets a range for each state of occupancy that contains the classic probability. The mass of each state, $m(\{empty\})$, $m(\{occupied\})$, and $m(U)$ are abbreviated as e, o, and u respectively. If u is equal to one, the occupancy at given position is totally unknown, which distinguishes the lack of information from uncertainty.

With the assumption of a straight-lines propagation of laser pulse, for a single laser range measurement $\mathbf{p} = \mathbf{s} + \mathbf{r}$, the space between the laser source and the reflecting 3D point should be empty, and the occupancy of space behind the reflecting point should be unknown. For the other space, the occupancy remains unknown. However, a laser ray is not a perfect line and a single laser point can not be presented as a pinpoint. Thus, a spatially extended distribution is assumed to present the laser beam considering the physical properties of the laser pulse propagation, errors in the georefencing and the alignment of the data, and the smallest observable and discriminable structure size. Thus, as illustrated in Fig. 2.8, on the basis of the distance d_x and d_y, the

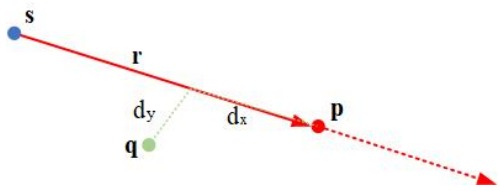

Figure 2.8: Longitudinal and transverse distances of $\mathbf{q}$ to $\mathbf{p}$ [Hebel, 2012].

assignment of belief mass to an arbitrary position $\mathbf{q}$ can be defined as:

$$\begin{aligned} e &= (1 - \frac{1}{1 + e^{-\lambda d_x - c}}) \cdot e^{-\kappa d_y^2}, \\ o &= (\frac{1}{1 + e^{-\lambda d_x - c}} + \frac{1}{1 + e^{-\lambda d_x + c}}) \cdot e^{-\kappa d_y^2}, \\ u &= 1 - e - o, \end{aligned} \tag{2.29}$$

where $d_x = (\mathbf{q} - \mathbf{p}) \cdot \mathbf{r}$ denotes the longitudinal distance and $d_y = \|(\mathbf{q} - \mathbf{p}) \times \mathbf{r}\|$ denotes the transverse distance. The parameters (λ, c, κ) depict the fuzziness of laser points. The parameters should represent the physical characteristics of the laser range measurements in the observation. The first factors of e and o are composed of sigmoid functions, among which one is used to describe the free space in front of $\mathbf{p}$ and the other one is for the unknown space behind the point. The longitudinal influence is controlled by the parameter c and λ. The second factor of e and o describes a Gaussian function, which is controlled by the parameter κ influence the transverse

extent. Fig.2.9 shows the ideal graph of nonzero belief masses around laser point $\mathbf{p}$, where the parameters λ, c, and κ are set as 12, 5, and 8, respectively.

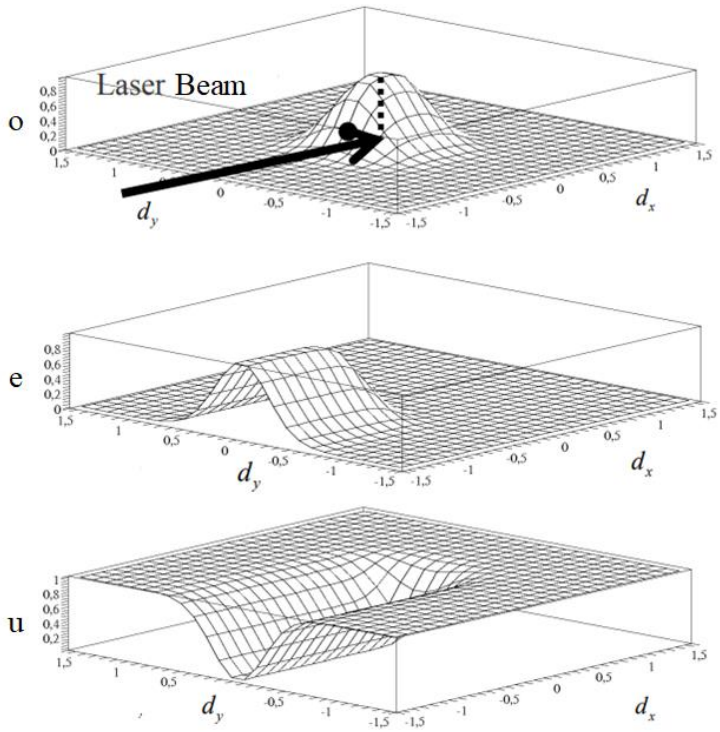

Figure 2.9: Comparison of belief masses (occupied, empty, and unknown) [Hebel, 2012].

2.5.4 Combination of evidence from different measurement

In the former section, only a single laser measurement $\mathbf{p} = \mathbf{s} + \mathbf{r}$ and its influence on mass assignment are considered for position $\mathbf{q}$. However, in practical cases, two or more laser beams are usually observed in the neighborhood of position $\mathbf{q}$. Thus, for position $\mathbf{q}$, here, we assume that two independent measurements, including $\mathbf{p}_1 = \mathbf{s}_1 + \mathbf{r}_1$ and $\mathbf{p}_2 = \mathbf{s}_2 + \mathbf{r}_2$, are obtained. Here, the DST is applied to combine multiple measurements. The amount of conflict between the two mass sets is:

$$C = e_1 \cdot o_2 + o_1 \cdot e_2. \tag{2.30}$$

Based on the combination rule of the DST, the conflict evidence is ignored and the joint mass can be calculated as:

$$\begin{aligned} e &= \frac{e_1 \cdot e_2 + e_1 \cdot u_2 + u_1 \cdot e_2}{1 - C} \\ o &= \frac{o_1 \cdot o_2 + o_1 \cdot u_2 + u_1 \cdot o_2}{1 - C}, \\ u &= \frac{u_1 \cdot u_2}{1 - C} \end{aligned} \tag{2.31}$$

The aforementioned operation can be written as $m = m_1 \oplus m_2$. Since the operation is commutative and associative, it allows for the combination of an arbitrary number of belief assignments. When the data amount is large, LiDAR point cloud can be divided into small blocks and stored separately in different files to avoid memory problem.

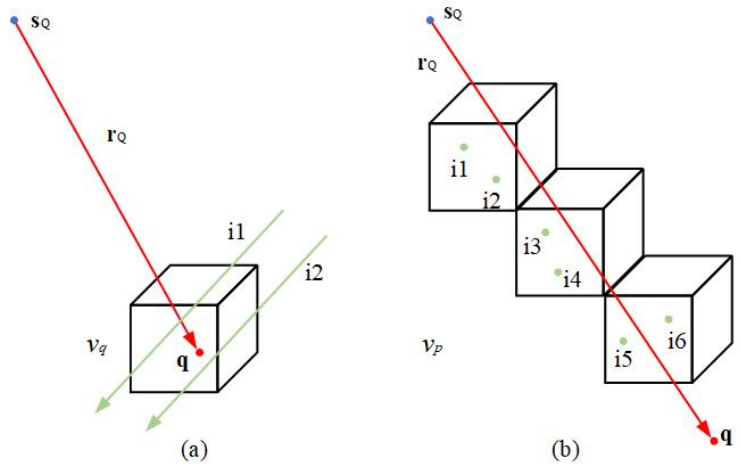

Figure 2.10: Conflicts between reference data (green) and the current measurement $\mathbf{q} = \mathbf{s}_Q + \mathbf{r}_Q$ (red) : a) empty space at $\mathbf{q}$ and b) occupied space along $\mathbf{r}_Q$ [Hebel, 2012].

2.5.5 Change detection

To detect whether new measurements confirms or contradicts the mass assignments that are obtained based on former measurements, the conflict types are defined. The first type of conflicts happen when the laser point lies in the region of empty space (Fig. 2.10a). Here, let I_q be the set of indices which are assigned to the cell $v_q in \mathbf{V}_R$. Thus, the joint mass m_1 can be obtained by taking old measurements $\mathbf{p}_i \in \mathbf{L}$ where $i \in I_q$:

$$m_1 = \oplus_{i \in I_q} m_i. \tag{2.32}$$

Meanwhile, the mass assignment m_2 can be given as:

$$e = u = 0, o = 1. \tag{2.33}$$

The conflicts between m_1 and m_2 can be obtained in the same way as it is achieved in Eq. 2.30.

The other types of conflicts occur when the laser beam traverses occupied space in front of laser point $\mathbf{q}$ (Fig. 2.10b). Here, the Bresenham's line drawing algorithm is applied to identify grid cells $\mathbf{V}_P$ through which the laser beam $(\mathbf{s}_Q, \mathbf{r}_Q)$ passes. Let v_p denote the subset of $\mathbf{V}_P$ and I_p be the set of indices assigned to cells V_p. These indices denote the laser point in the reference data that are influenced by the current laser measurement. Thus, for each position p_i with $i \in I_p$, the mass set m_2 can be updated as:

$$m_2 \leftarrow m_2 \oplus m_i, \forall i \in I_p \tag{2.34}$$

Then, the measure of conflict can be obtained as $C = m_2(\{empty\})$.

2.5.6 Consideration of additional attributes

In the former sections, the different characteristics of continuous surfaces and clutter are not considered in the model described in Eq. 2.29. However, the vegetation penetration of laser pulse when scanning vegetation may leads to self-induced conflicts of mass assignment since multiple echoes are dealt with independently. Considering this aspect, one possible solution is to increase

the degree of ignorance and the dimensions of occupied space when representing the fuzziness of vegetation. Here, two weighting factors f_e and f_o are induced:

$$\begin{aligned} e &= f_e(1 - \frac{1}{1+e^{-\lambda d_x - c}}) \cdot e^{-\kappa d_y^2}, \\ o &= f_o(\frac{1}{1+e^{-\lambda d_x - c}} + \frac{1}{1+e^{-\lambda d_x + c}}) \cdot e^{-\kappa d_y^2}, \\ u &= 1 - e - o. \end{aligned} \tag{2.35}$$

Here, classification results are utilized to separate vegetation with continuous surfaces. In addition, for the continuous surfaces, the model can be refined by considering the orientation of the surfaces. In case a continuous surface is detected, the distribution should be spread out along the detected surface. Thus, the longitudinal and traverse distances can be modified using the local normal vector:

$$\begin{aligned} \bar{d}_x &= (\mathbf{q} - \mathbf{p}) \cdot \mathbf{n}_0, \\ \bar{d}_y &= \|(\mathbf{q} - \mathbf{p}) \times \mathbf{n}_0\|. \end{aligned} \tag{2.36}$$

In the extended model, if the distance between point $\mathbf{q}$ and laser point $\mathbf{p}$ is far away, the belief mass is still modeled using (d_x, d_y) and (λ, c, κ) using Eq. 2.29. If the distance is small, the belief mass can be modified by using $(\bar{d}_x, \bar{d}_y)$ and (λ, c, κ), where the parameter κ has a substantially smaller value.

2.5.7 Multi-view stereo vision

In this section, we introduce the process of generating point clouds using images, in which two major steps are involved, including a structure from motion (SfM) process and a multi-view stereo (MVS) process. First, the image sequence of each acquisition date is processed by a SfM process (see Fig. 2.11), which involves the process of recovering relation camera poses and the sparse 3D points from unordered image sequences. Based on the camera poses recovered in the SfM process, dense point clouds can be generated based on the oriented images using the MVS process.

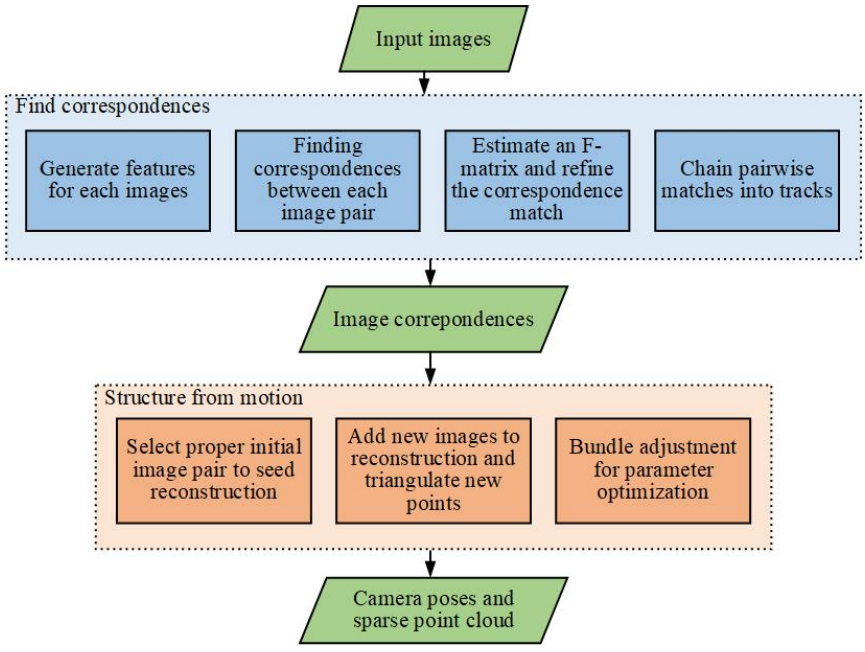

Figure 2.11: The process of SfM.

In the SfM process, the key part of the SfM process is to find the corresponding points between images. SIFT descriptor is utilized to generate features from images, and kd-tree is used to calculate the Euclidean distances between features of keypoints and to define corresponding points for matching. The **F** matrix which contains both exterior orientation parameters and initial camera parameters can be estimated using a RANSAC process, in which outliers can be excluded during the process. Following the same procedure, chain pairwise matches can be generated and formed tracks. The exterior orientation parameters, including rotation **R** and translation **t** can be generated by decomposing the essential matrix **E**, which can be obtained using **F** matrix:

$$\begin{aligned} \mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 &= 0 \\ \mathbf{E} &= \mathbf{K}_1^T \mathbf{F} \mathbf{K}_2 \end{aligned}, \tag{2.37}$$

where **K** denotes the matrix formed by camera intrinsics.

However, the exterior parameters estimated by the aforementioned process is not precise enough, Bundle Adjustment (BA) will be utilized to further optimize the estimated parameters by minimizing the reprojection errors (see Fig. 2.12). The BA process aims at minimizing the function:

$$J = \sum_i \sum_j (\tilde{\mathbf{x}}_i^j - \mathbf{K}[\mathbf{R}_i|\mathbf{t}_i]\mathbf{X}^j)^2, \tag{2.38}$$

where $\tilde{\mathbf{x}}_i^j$ denotes the image coordinates and $\mathbf{X}^j$ denotes the object coordinates.

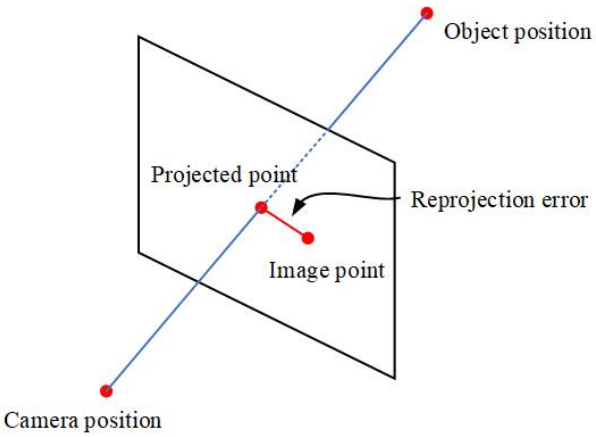

Figure 2.12: Illustration of reprojection error.

The exterior orientation of all images can be determined in a common coordinate system using the SfM process. However, we can only obtain sparse point clouds since reconstructed points are generated based on the points matched with features. Thus, the MVS process is utilized to further reconstruct dense point clouds from the images sequence, in which almost all the pixels in images are matched with others and reconstructed as 3D points. In the MVS process, first, every image is selected as master image once. For all master images all potential search images are selected based on the criteria just mentioned. To avoid that an image receives two (or more) search images which are close together, i.e. whose distance is below the minimum baseline length, only the image having the lowest sum of the angle deviation criteria is selected. For all remaining k search images, Semi Global Matching [Hirschmuller, 2007] is performed using LibTSGM [Rothermel et al., 2012], and k disparity maps are calculated. The disparity maps are fused based on a scheme which follows the one presented in, where the final value for the distance

D from the camera center to the 3D point and its stand deviation are determined by a least square adjustment. The final 3D point coordinates are then calculated by:

$$\mathbf{X}_o = \mathbf{R}(\mathbf{n} \cdot D) + \mathbf{X}_c. \tag{2.39}$$

With rotation matrix $\mathbf{R}$, unit vector $\mathbf{n}$ from the perspective center to the pixel and camera position $\mathbf{X}_c$. Additionally, the precision of the coordinates can be determined based on the standard deviation of parameters estimated in bundle adjustment. To determine the best point, the point with the highest number of rays is determined. If there are points with the same number of rays, the point with the smallest standard deviation will be chosen.

In case that disparity maps may change widely in quality and resolution, Kuhn et al. [2017] developed a feature based on Total Variation that allows pixel-wise classification of disparities into different error classes.

3 Robust registration of 3D point clouds

Point cloud registration is invariably an essential and challenging task in the fields of photogrammetry and computer vision to align multiple point clouds to a united reference frame. In this chapter, we report two point cloud registration methods [Huang et al., 2020a,c]. To be specific, the first method is a three-step projection-based methods which includes a projection from the 3D space to a 2D plane, a projection from the 3D space to a 1D histogram, and phase correlation between the images and the histograms. While, the second one is a fully 3D registration solution which is achieved by converting the estimation of rotation, scaling, and translation in the spatial domain to a problem of correlating low-frequency components in the frequency domain. Fig. 3.1, we show the key methods corresponds to the ones we have shown in our research frame in Section 1.3.

Figure 3.1: The proposed methods for registration that included in this chapter.

3.1 Projection-based point cloud registration with 2D phase correlation (PBPC)

The PBPC method consists of three essential steps: the decoupling of 3D transformation via projection-based dimensionality reduction, the estimation of horizontal transformation (i.e., rotation, scaling, and translation), and the estimation of vertical translation. In Fig. 3.2, the workflow is illustrated, showing the core steps of involved methods and sample results. It should be noted that the target point cloud provides the reference coordinate system, and the source point cloud is the point cloud need to be registered to the reference. A detailed explanation of each step will be given in the following.

In the first step, the projection-based dimensionality reduction is used to decouple the 3D transformation into the 2D horizontal transformation and 1D vertical translation. In the following step, the 3D transformation estimation can be solved by solving the 2D matching between images and the 1D matching between histograms separately. Specifically, 3D point clouds are projected to the principal projection plane vertically, generating 2D images with intensities of pixels indicating the number of projected points. Correspondingly, 1D histograms are generated by projecting all the points in point clouds to the z-axis horizontally. Afterward, in order to match the 2D images for estimating horizontal transformation, we use Fourier-Mellin Transformation (FMT) for decoupling rotation, scaling, and translation between images. In this way, both the estimation

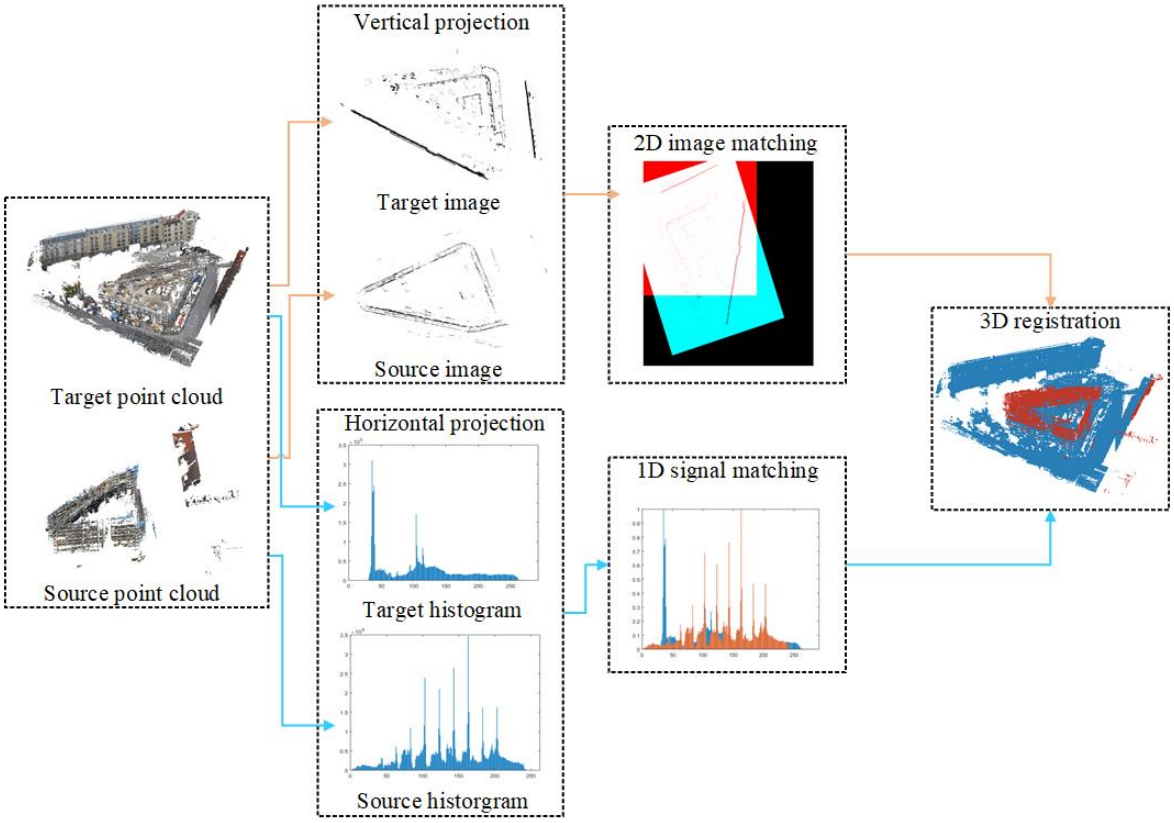

Figure 3.2: Workflow of the PBPC method.

of rotation and scaling and the estimation of translation are converted into problems of solving the shifts. The shifts can be then well estimated by a robust 2D phase correlation method. Similarly, 1D histograms are also registered by a 1D robust phase correlation method. Combining these estimated transformations using 2D images and 1D histograms to 3D transformation, point clouds acquired at different times can be finally registered into a united coordinate system. In the following sections, the essential algorithms and detailed steps of the workflow will be elaborated on.

3.1.1 Decoupling of 3D transformation

The decoupling of 3D transformation is actually to decompose 3D transformation between point clouds into a 2D transformation (i.e., rotation, scaling, and translation) between images and a vertical translation between histograms, which composed of three steps: (1) identifying the principal projection plane and axis; (2) projecting 3D points to the principal projection plane to generate 2D images; (3) projecting 3D points perpendicularly to the principal axis to obtain 1D histograms.

Identification of the principal plane and axis

The purpose of identifying the principal projection plane is to find the ground surface or floors with a horizontal orientation so that points in a point cloud can be projected to this plane. This is because for conducting the decoupling of horizontal and vertical transformations, we need a common reference plane and a reference axis. Benefiting from the characteristics of common artificial buildings, all the walls of a building are constructed along the vertical direction. This indicates that the majority of walls in a point cloud are perpendicular to a principal projection

plane in parallel to the ground surface or floors, and in the meantime, the principal axis (i.e., the normal vector) that is perpendicular to this principal plane will be in parallel to these walls as well. Thus, this principal plane and the corresponding principal axis should be identified in advance to the projection of points.

The approach of detecting and extracting a principal plane is achieved by the plane-fitting algorithm based on RANSAC [Schnabel et al., 2007], which is the most popular used method for shape fitting. Since the ground surface is normally the plane having the largest coverage area in the construction site. For some special cases that even the ground surface is occluded, we will first extract the plane of a facade and then define a plane perpendicular to this facade as the principal plane. In other words, we need to detect and extract one major planar surface (i.e., ground or facade) from the point cloud and then define the principal plane. Specifically, given points $\mathbf{P} = \{p_1, p_2, ..., p_n\}$ of a point cloud and the mathematical model of the plane $T(\theta)$ with a set of parameters $\theta \in \Theta$, RANSAC pursuits to estimate the parameter value $\theta^* \in \Theta$ which has the largest number of inliers in $\mathbf{P}$. RANSAC is an iterative strategy, and in each round of iterations (e.g., the round i) the generic process is executed with following sequences: First, select random sample $\mathbf{P}_m \in P$ with n_m points, where n_m is the minimal number of points for estimating θ_i. For the case of the plane, at least three points are mandatory to define a plane in 3D space. Then, estimate the value of θ_i using only $\mathbf{P}_m$. Afterward, evaluate the cost $C(\theta_i)$ of fitting entire dataset $\mathbf{P}$ to the estimated model $T(\theta_i)$:

$$C(\theta_i) = \sum_{\mathbf{p} \in \mathbf{P}} f_\rho(T(\theta_i), \mathbf{p}), \tag{3.1}$$

where f_ρ is the function to measure the cost for each p. The iteration process is repeated for a given number of iterations. The parameters θ^* make the model $T(\theta^*)$ earning the minimum cost are chosen as the optimized parameters for the model, and normally it means the model with the largest number of inliers $\mathbf{P}_{in} \in \mathbf{P}$. The inliers $\mathbf{P}_{in}$ are identified by a given threshold of ρ. To be specific, a point has an error e to the model $T(\theta)$, which is small than ρ, the function f_ρ returns zero value and otherwise a positive penalty T:

$$f_\rho(e) = \begin{cases} 0 & \text{if } e^2 < \rho^2 \\ T^2 & \text{else} \end{cases}. \tag{3.2}$$

If not only one plane is extracted, for the extracted planes, we will select the largest one with a relatively smaller residual as the ground. Once the principal plane is found, its normal vector, which is perpendicular to the principal plane, will serve as the principal axis for the horizontal projection. In Fig. 3.3, we give an illustration of the identified principal plane and axis.

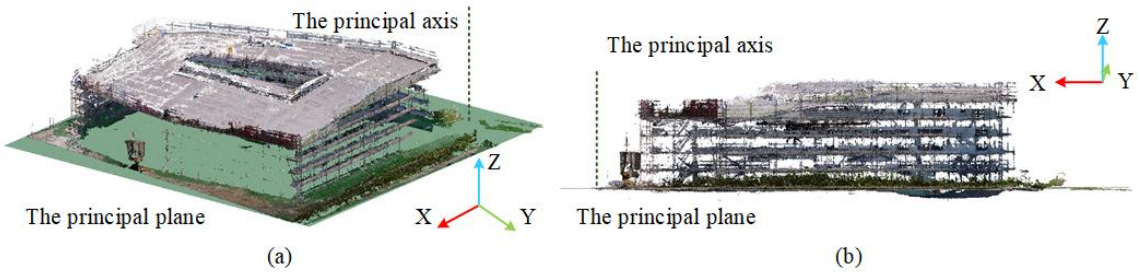

Figure 3.3: Illustration of the identified principal plane and axis. a) The identified principal plane and axis from isometric view and b) from side view.

Vertical projection

The vertical projection of the point cloud is a transformation from the 3D points into a 2D image by projecting all the points to the principal plane along the vertical direction. Subsequently, the 2D projection-based image is generated, in which the grey value of each pixel is calculated by counting the number of points projected into the area denoted by the bounding box of this pixel. For the vertical structure, we assume that the points belong to the vertical structure fall into the aligned pixels or neighboring pixels, which indicates that the bright lines in 2D projection refer to the vertical structure in 3D point clouds. Thus, we apply a filter to eliminate the artifacts caused by isolated points in 3D point clouds in the projection image by setting a threshold to the grey values of the pixels. The threshold depends on the pixel size we set for vertical projection and the noise level of point clouds.

Horizontal projection

As for generating horizontally projected 1D histograms, we adopt a similar strategy, in which the 3D points are projected to the principal axis and the height range is divided into a sequence of bins and the value for each bin also represents the number of points falling into the height range represented by each bin. Here, for the generated histogram, the peaks represent the horizontal structures since the horizontal structures create a high number of points owing to similar heights. Correspondingly, a filter is applied to eliminate some noise caused by irrelevant information and retain the main structures of the as-built buildings for further matching.

3.1.2 2D image matching with FMT and phase correlation

This section describes the Fourier-based method we apply for the 2D image matching. In the problem of 2D image matching, rotation, scaling, and translation need to be settled. In PBPC, the estimation of rotation and scaling is separated with the estimation of translation and is also transferred into a shift estimation problem using FMT. Additionally, if rotation and scaling are settled, the estimation of translation parameters is also a shift estimation problem. These two shift estimation problems are then solved by a robust phase correlation method.

Decoupling of horizontal rotation, scaling and translation with FMT

As introduced in Section 2.1, by applying FMT, the estimation of rotation and scaling of 2D projection images is transformed into a shift estimation problem. Besides, translation estimation is also converted to a shift estimation problem. An illustration of the processing of the projected images using FMT is given in Fig. 3.4. We solve the shift estimation problem with a robust phase correlation method, which will be presented in the following section.

Shift estimation with robust phase correlation

Inspired by the Hoge's solution, the cross-power spectrum can be decomposed to two rank-one signals. Thus, the problem of the shift estimation can be simplified by investigating the rank-one signals. Firstly, the rank-one signals can be approximated by the SVD method. By utilizing the decomposed signals, the problem of 2D phase unwrapping and high-frequency components eliminating can also be simplified into finding a 1D solution, which is much more robust and also avoids the ill-posed problems when noise is intense. In order to estimate the slope of the fitted line of the decomposed and unwrapped signal, a RANSAC algorithm is adopted, in which a subset with a minimized required size of randomly sampled data is utilized to fit the model. The workflow of the robust phase correlation is illustrated in Fig 3.5.

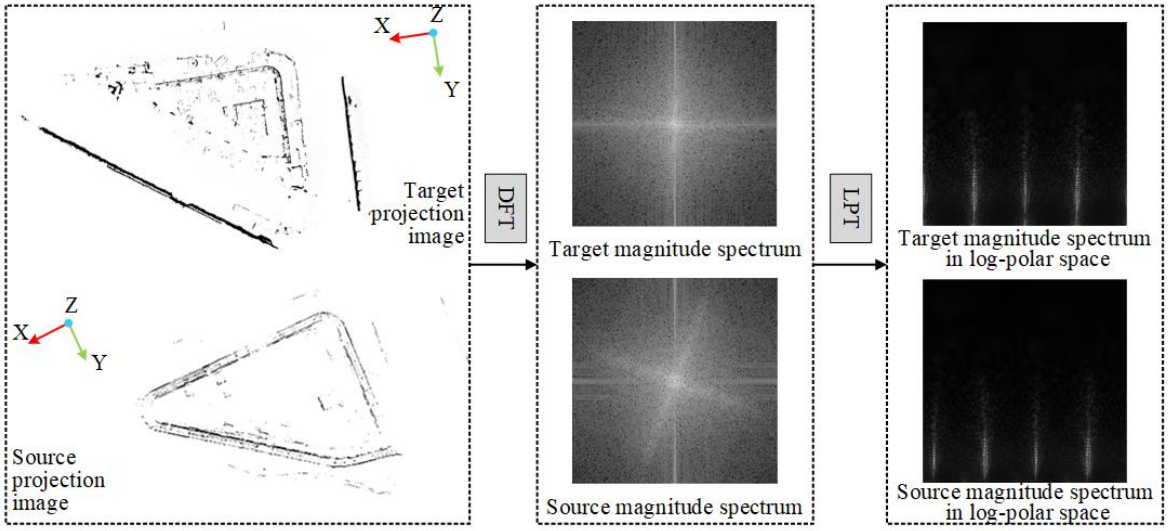

Figure 3.4: Illustration of FMT for projected images.

3.1.3 Vertical signal matching using 1D phase correlation

As for the 1D histogram matching for the vertical translation estimation, it can be achieved in a similar way to the 2D image matching. Firstly, the 1D histogram in the spatial domain can be regarded as 1D signals and transformed into the frequency domain as:

$$F(u) = G(u)e^{-i(ux_0)}. \tag{3.3}$$

Similarly, the normalized cross-power spectrum of these 1D signals can be expressed as:

$$Q(u) = \frac{F(u)G^*(u)}{|F(u)G^*(u)|} = e^{-iux_0}, \tag{3.4}$$

and the phase difference angle can be written as:

$$\angle Q(u) = ux_0. \tag{3.5}$$

Thus, the shift can be estimated by fitting the slope of the 1D line by RANSAC. As shown in Fig. 3.6, before the line fitting, 1D unwrapping needs to be conducted to solve the problem of phase-wrapping. Afterward, the aforementioned RANSAC algorithm is leveraged to fit the low-frequency components of the unwrapped phase angle for the translation estimation.

The complete process can be summarized as follows: (1) identify the principal projection plane by plane-fitting using RANSAC; (2) project the point clouds to the principal projection plane and obtain the 2D projection-based images; (3) project the point clouds to the principal axis and obtain the 1D histogram; (4) decouple the horizontal rotation, scaling and translation of the 2D images using FMT; (5) Estimate horizontal rotation and scaling using a robust phase correlation algorithm, and a similar step is conducted to estimate horizontal translation; (6) estimate vertical translation with 1D histograms using 1D phase correlation; (7) register the point clouds with estimated transformation parameters.

3.2 Robust global point cloud registration with 3D phase correlation (GRPC)

For estimating 3D transformation between two coordinate frames, traditional point cloud registration methods rely heavily on matching correspondences via local geometric features. They

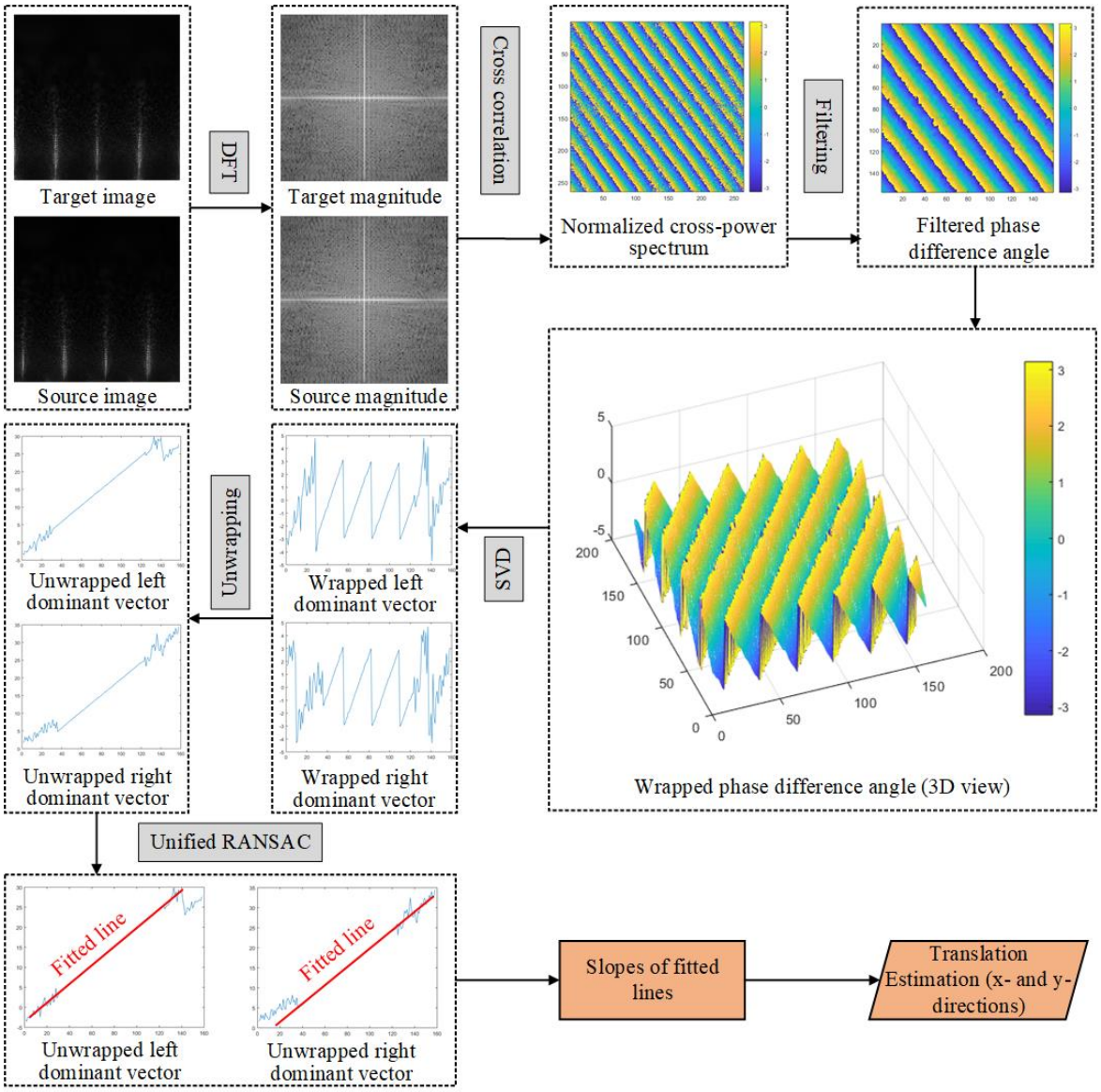

Figure 3.5: Shift estimation using a robust phase correlation method.

firstly extract key points or feature points from both source and target data and then conduct the matching of corresponding points with features for estimating the transformation between different coordinate frames. Unlike conventional methods, GRPC is a new global information-based registration strategy following a principle which estimates 3D transformation in the frequency domain robustly. Following principle of GRPC, the entire point clouds are converted into 3D signals and regards them as global features. Then, the transformation between coordinate frames is achieved via the phase correlation in the frequency domain. Comparing with using local features of key points, the use of global features can increase the robustness. By transforming 3D points to 3D signals in the frequency domain, we can separate and eliminate high-frequency parts representing noise and outliers, so that the matching of features could be more reliable. By using a novel robust and accurate phase correlation, the feature matching can be addressed by the optimization with a closed-form expression.

In Fig. 3.7, we illustrate a comparison of workflows using GRPC and conventional feature description-based ones. Specifically, the principle of GRPC for estimation of 3D transformation

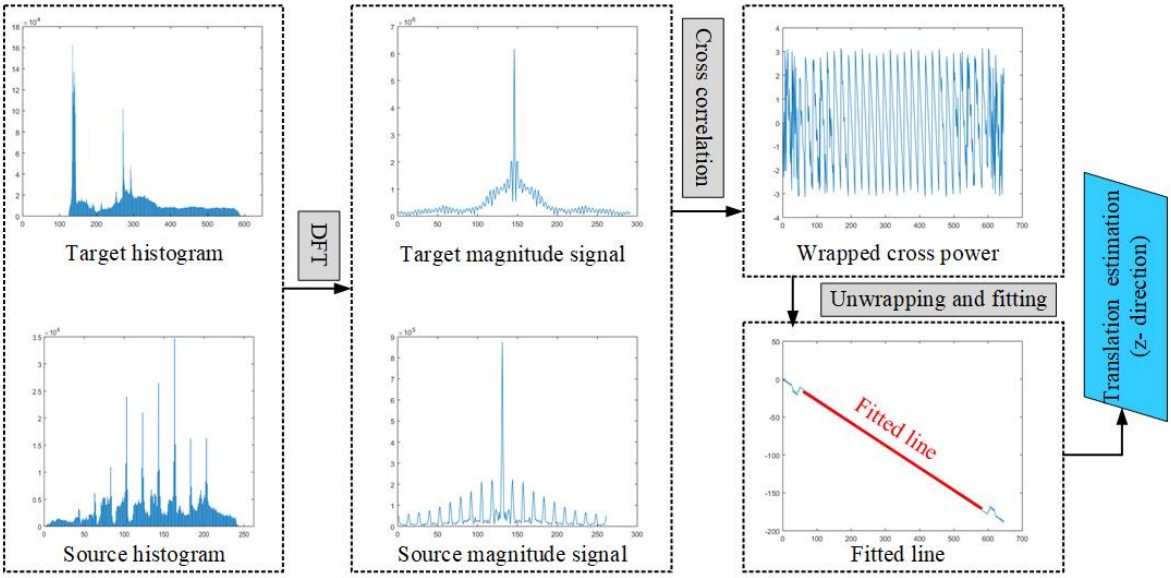

Figure 3.6: Estimation of vertical translation with 1D phase correlation.

mainly comprises three principal aspects, including the transformation from the spatial to the frequency domain, decoupling of rotation, scaling, and translation, and robust and accurate shift estimation.

3.2.1 Transformation from the spatial domain to the frequency domain

First, the point clouds are voxelized using the grid-based voxelization method in Section 2.4.2. The transformation from the spatial domain to the frequency domain is to create discrete 3D signals from unstructured and unordered points, which could be further used for the phase correlation. The transformation includes the voxelization and binarization of 3D points and 3D Fourier transformation of voxelized 3D data.

3D Fourier transform of voxelized 3D data

In the voxelization step, the original point clouds have been transformed to regularly sampled discrete 3D signals. Assume that two signals are correlated to each other by shifts in the spatial domain denoted as $\mathbf{t}_s \in \mathcal{R}^{n\times 1}$, where n is the dimensions of the data. The correlation between the two signals can be expressed as:

$$s(\mathbf{x}) = r(\mathbf{x} - \mathbf{t}_s), \tag{3.6}$$

where $s(\mathbf{x})$ and $r(\mathbf{x})$ represent two signals in the spatial domain. Afterward, a fast discrete Fourier transform (FFT) can be conducted on these two signals to transform them from the spatial domain to the frequency domain:

$$\begin{cases} S(\mathbf{k}) = FFT(s(\mathbf{x})) \\ R(\mathbf{k}) = FFT(r(\mathbf{x})) \end{cases}, \tag{3.7}$$

where $S(\mathbf{k})$ and $R(\mathbf{k})$ are the corresponding Fourier transforms of $s(\mathbf{x})$ and $r(\mathbf{x})$. Here, we use lowercase letters to represent the spatial domain, while uppercase letters denote the frequency domain. If we carry out a phase correlation between $S(\mathbf{k})$ and $R(\mathbf{k})$, the relation between these two signals can be written as:

$$S(\mathbf{k}) = R(\mathbf{k})e^{-i2\pi(\mathbf{k}\mathbf{t}_s)}. \tag{3.8}$$

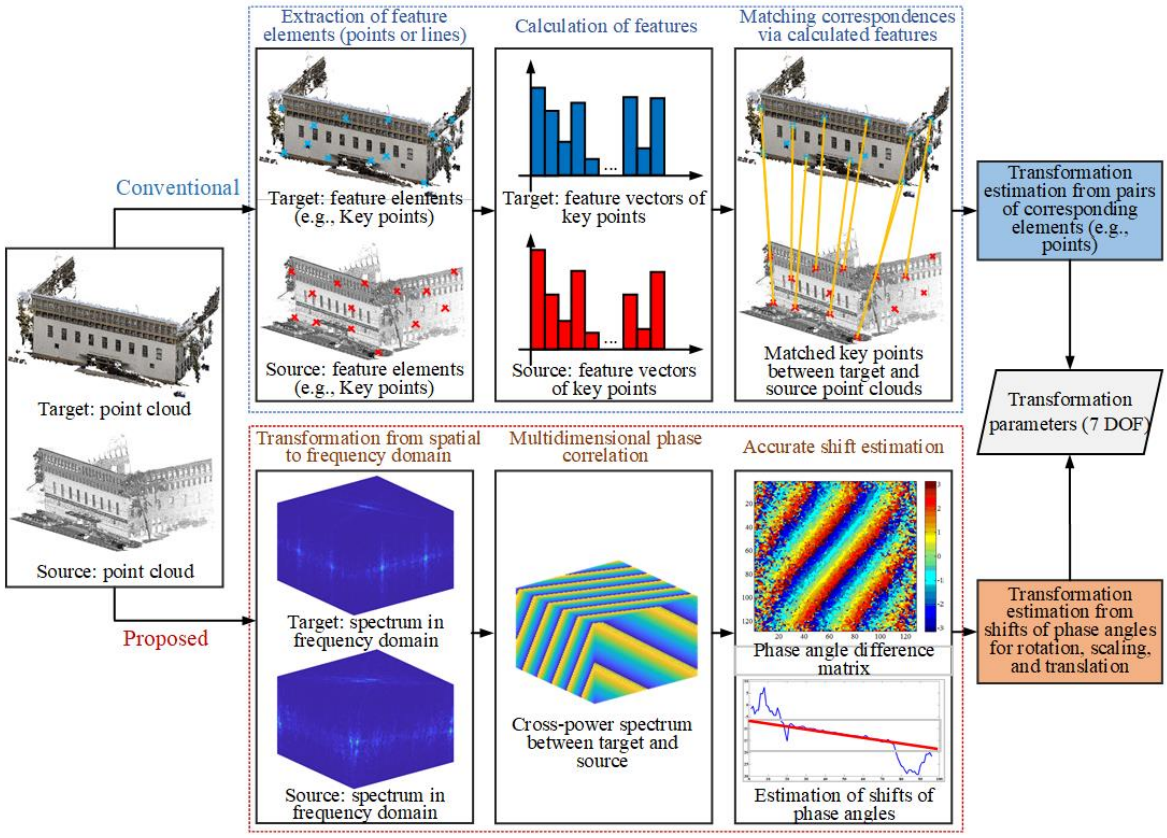

Figure 3.7: Workflow comparison of GRPC and conventional feature description-based strategy. Blue denotes the target point cloud. Red denotes the source point cloud.

The normalized cross-power spectrum can be calculated as:

$$Q(\mathbf{k}) = \frac{S(\mathbf{k})R^*(\mathbf{k})}{|S(\mathbf{k})R^*(\mathbf{k})|} = e^{-i2\pi(\mathbf{k}\mathbf{t}_s)}, \tag{3.9}$$

where R^* represents the complex conjugate of R. The magnitude of Q is 1 after the normalization. From this equation, we can find that the translation $\mathbf{t}_s$ can be solved by exploiting the correlation between the signals. At this point, we have converted the estimation of translation in the spatial domain to an addressable problem in the frequency domain. This is a commonly used strategy in dense image matching in many previous works. However, when it comes to the point cloud registration, the problem is more complex, because the transformation between coordinate frames is an ill-posed problem of seven DoFs [Bellekens et al., 2015]. For solving the ill-posed estimation problem of transformation, the transformation has to be decoupled and converted into a shift estimation task.

3.2.2 Decoupling of rotation, scaling and translation

The proposed strategy is to obtain the transformation by decomposing the transformation to several sub-problems, which can quickly solve shift estimation by phase correlation methods. First of all, we present the method used to decouple the transformation parameters, namely rotations, scaling, and translation. Before introducing details of the method, we present some basic concepts and notations used in the method. Assuming that two 3D voxel data can be presented as $f(\mathbf{x})$ and $h(\mathbf{x})$ which differ by rotations, translation, and scaling, the relation between the two 3D data can be expressed as:

$$h(\mathbf{x}) = sf(g(\alpha, \beta, \gamma)\mathbf{x} - \mathbf{t_s}), \tag{3.10}$$

in which $g(\alpha, \beta, \gamma)$ denotes rotations, s represents scaling factor, and $\mathbf{t_s} = [t_x, t_y, t_z]$ shifts the 3D voxel data by translation.

Two 3D voxelized discrete data can be transformed to the frequency domain using 3D FFT, then the relation between the spectrum of the data can be represented as follows:

$$H(\mathbf{k}) = s^3 F(g(\alpha, \beta, \gamma)\mathbf{k}s^{-1}) e^{-i2\pi g(\alpha,\beta,\gamma)\mathbf{k}s\mathbf{t_s}}, \tag{3.11}$$

where $\mathbf{k} = [u, v, w]$ denotes the coordinates in frequency domain. It shows that the translation only has an impact on the phase of the spectrum. Thus, by calculating the magnitude of the spectrum, the 3D translation can be decoupled. The relation can be simplified as:

$$|H(\mathbf{k})| = s^3 |F(g(\alpha, \beta, \gamma)\mathbf{k}s^{-1})|. \tag{3.12}$$

As shown by the equation, the spectral magnitude is influenced by a combination of rotations and scaling. A decoupling process is needed for estimating rotations and scaling separately. Explicitly, the rotations orient the 3D structure of the magnitude of the spectrum in the same way as it does for the original 3D data in the spatial domain. At the same time, the scaling affects the spectral magnitude in two aspects. One aspect is that the cubed scaling s^3 only influences the amplitude of the magnitude spectrum. However, the amplitude does not influence the structural information, indicating that it makes no difference in the phase matching procedure. Another is that the term s^{-1} indicates that the scale difference between signals in the spatial domain shows a reciprocal effect on the spectrum in the frequency domain. It means that scale also influences the structural information of the spectrum. Thus, in order to decouple rotations with scale, the spectral magnitude is radially accumulated. By accumulating spectral data radially, we can obtain a spherical function on which rotations present shifts of the structural information. Then, only rotations remain in terms of the accumulated spectrum. Thus, the general procedure of the seven DoFs transformation estimation is to estimate the rotations using the accumulated spectrum first and subsequently estimate other transformation parameters.

3.2.3 Robust and accurate shift estimation

Via the use of phase correlation, we can convert the spatial translation estimation to the underlying shift estimation of phase angle differences. The main concept of phase correlation is that any shifts between two correlated signals (i.e., 2D images or 3D discrete signals) in the spatial domain can be represented as a phase shift in the frequency domain. Compared with correlation-based solutions which are also widely used, phase correlation seems to be more robust and accurate. Simultaneously, the processing efficiency is improved as a fringe benefit.

However, if we apply classic phase correlation methods (e.g., estimating shifts from the peak of the IFT of the cross-power spectrum) to point cloud registration, we will encounter problems. For example, the estimated shifts can only achieve a voxel-level accuracy, which directly relates to the granularity of voxelization. In Fig. 3.8, we display a sketch showing a comparison between registrations with voxel- and subvoxel-level accuracies. To obtain an accurate registration, a subvoxel level accuracy is mandatory, and this should be addressed by a fine-estimation of phase angle differences in the phase correlation. Moreover, as we have previously mentioned, the outliers and non-overlap areas will result in noise in the frequency domain signals, so we need to overcome these disturbances in the estimation of phase angle differences simultaneously. To this end, a novel multidimensional phase correlation method is proposed using merely the low-frequency components and $\ell 1$-normalized linear fitting for an accurate and robust shift estimation.

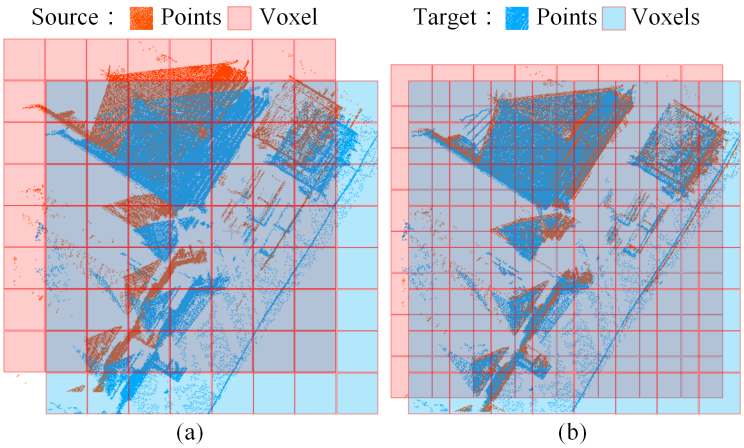

Figure 3.8: Registration (top view) with a) voxel level accuracy and b) sub-voxel level accuracy.

Multidimensional phase correlation

As assumed, the signals to be matched are in three dimensional, thus, the coordinates and shifts can be written as $\mathbf{k} = [u, v, w]$ and $\mathbf{t}_s = [t_x, t_y, t_z]$, respectively. It should be noted that although the solutions are provided in the 3D version, it can be easily adaptive to other multidimensional cases (i.e., 2D). In this case, the normalized cross-power spectrum can be written as:

$$Q(u, v, w) = e^{-i2\pi(ut_x + vt_y + wt_z)}. \tag{3.13}$$

The IFT of $Q(\mathbf{k})$ contains a Dirac delta function in an analytical way. Thus, the phase correlation

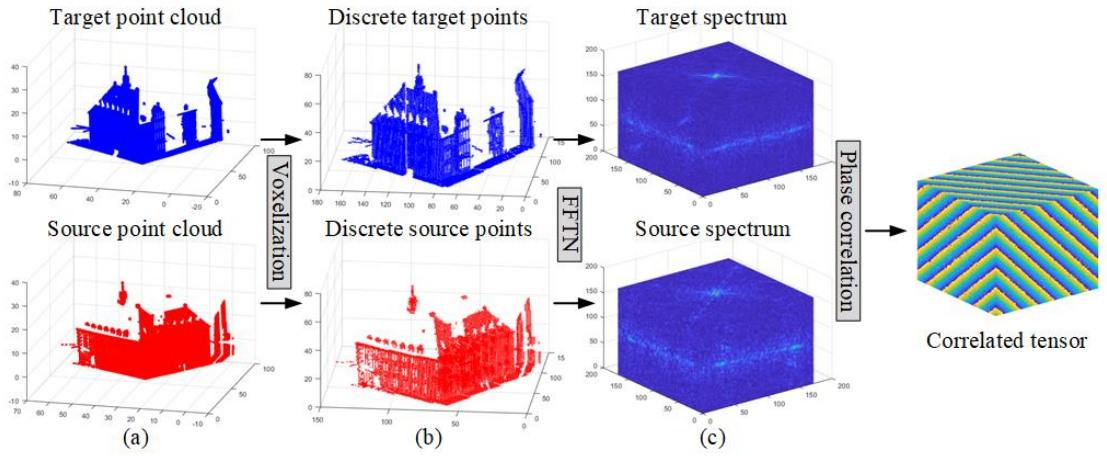

Figure 3.9: Multidimensional phase correlation. a) Original point clouds, b) voxelized 3D points, c) spectrum of discrete 3D signals after FFT, d) correlated tensor from phase correlation of spectrums.

result can be obtained by finding the Dirac peak, whose coordinates corresponds to the estimated parameters. However, this solution has a two-fold drawback. On the one hand, when the noise level is high, it will be hard to find a single peak for the function, which will lead to the failure or mistake in estimating the shifts. On the other hand, this kind of strategy is only able to produce a result in the accuracy of integer voxels or pixels, as shown in Fig. 3.8. This level of accuracy can not fulfill the requirement of registration of different scenes, especially for large-scale scenarios where the voxel size cannot be set as a small value. Although there are some solutions proposed to improve the accuracy of the sub-pixel level by interpolation, the fitting of a high-dimensional

polynomial function is not always robust, especially in high noisy cases. For tackling this problem, rather than sticking to interpolating the peak by some high-dimensional functions, many other methods have been reported aiming at improving the accuracy of phase matching to the sub-pixel level. An elegant way to solve the unknown shift parameters is to fit the phase difference angle, which can be represented as a linear function (i.e., 3D plane function). However, this is only feasible in an ideal situation. The real case is that noise, outliers, and the low-overlapping ratio of point clouds will produce strong disturbances to the cross power spectrum. Furthermore, the phase unwrapping of the high dimensional tensor will face an ill-posed problem with a high-level noisy cross-power spectrum tensor. In the following section, we will present our solution to the problems mentioned above.

Extraction of low-frequency components and signal decomposition

After obtaining fourier spectrum for each individual 3D signals and their corresponding normalized cross-power spectrum, it is of great importance to select from the frequency components and separate those low-frequency parts. Assuming that the 3D phase correlation between point clouds share similar characteristics to the 2D phase correlation between images, the same procedures can be conducted for the 3D signals. For 2D image matching, the concept is that high-frequency components corresponds to aliasing and noise, thus most of energy lies in the low-frequency components of the signals [Leprince et al., 2007]. Thus, for the cross-power spectrum from the 3D phase correlation, a similar strategy is conducted to mask out around 80% of frequency components at the boundary of the tensor Q (see Fig. 3.10). Namely, only the center part of the tensor Q will be preserved for further processing. As for the estimation of the parameters, instead

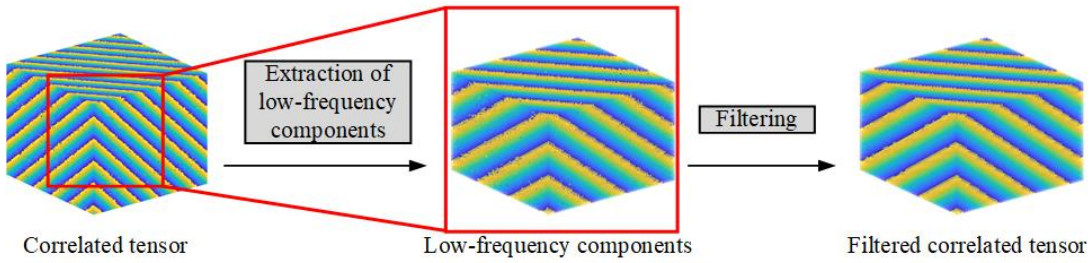

Figure 3.10: Illustration of extraction of low-frequency components.

of fitting the high-dimensional plane, a robust subpixel phase correlation method is applied, which combines the concept of SVD and $\ell 1$ normalization for robust estimation. The normalized cross-power spectrum can be rewitten as:

$$Q(u,v,w) = e^{-i(ut_x+vt_y+wt_z)} = e^{-iut_x}e^{-ivt_y}e^{-iwt_z} = Q_{x0}(u)Q_{y0}(v)Q_{z0}(w). \tag{3.14}$$

The cross-power spectrum can be represented by three rank-one signals. Thus, the task of the 3D shift estimation can be separated to several tasks which exploits the rank-one signals. Firstly, the SVD method can be utilized to divide the cross-power spectrum into several approximate rank-one signals. Thus, instead of solving phase wrapping and eliminating high-frequency components in high dimensions, these problems can be solved by finding 1D solution using the decomposed signals. Compared to the previous one, the 1D solution will be less sensitive to noise and outliers. Simultaneously, the ill-posed problem of high-dimensional phase unwrapping can also be avoided. To calculate the coefficients of the fitted linear function of the decomposed and unwrapped signal, we adopt a robust algorithm in which $\ell 1$ normalization is utilized to add constraint and improve the model's robustness. Compared with $\ell 2$ normalization (e.g., least-squares adjustment), $\ell 1$ is less influenced by noise and outliers by adding constraints for the parameters.

Robust estimation of 3D shifts with $\ell 1$ norm

Although the low-frequency components in the cross-power spectrum are separated and extracted, Eq. 3.15 can still be utilized for the calculated of shift parameters. In order to estimate the parameters of this linear function, a robust estimator with $\ell 1$ normalization is adopted (see Fig. 3.11), which can be presented as follow:

$$\arg\min\left(\frac{1}{2N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2+\lambda\sum_{j=1}^{p}|\beta_i|\right), \tag{3.15}$$

where (x_i, y_i) are N pairs of data values of the decomposed signals and λ is a non-negative regularization parameter. In this problem, $\ell 1$ norm is involved, aiming to add constraints when estimating the linear function parameters. The alternating direction method of multipliers (ADMM) algorithm is utilized to solve the aforementioned optimization problem.

Once the linear functions for each decomposed signals are estimated, parameters of the unwrapped phase angles of the identified components can be converted to the real estimated shift parameters:

$$\begin{cases} \Delta X = \delta x M/(2\pi) \\ \Delta Y = \delta y N/(2\pi) \\ \Delta Z = \delta z L/(2\pi) \end{cases}, \tag{3.16}$$

where M, N, L denote the dimensions of the input tensor, which are from DFT. In DFT that we used for transforming the point cloud into the frequency domain, the dimensions of the samples space in the frequency domain are $M \times N \times L$. Since the shifts are converted to the phase angle difference ranging from $-\pi$ to π, once we get the phase angle differences ΔX, ΔY, and ΔZ, we need to recover the real shifts by the use of Eq. 3.16 based on the sampled dimensions from DFT.

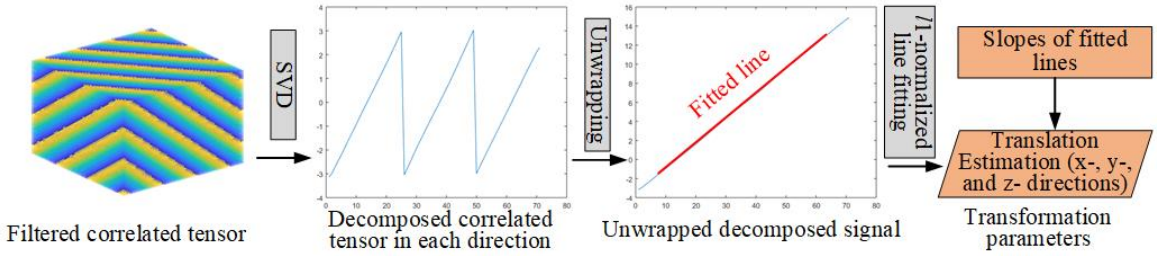

Figure 3.11: Illustration of the robust estimation of 3D shifts using $\ell 1$ norm.

3.2.4 Application to the proposed GRPC method

Based on the proposed principle, we present GRPC for point cloud registration in the frequency domain, decoupling of transformation, and robust multidimensional phase correlation. The way of decoupling rotations and scaling is inspired by [Bülow & Birk, 2012, 2018]. Essential processing steps are summarized as a complete workflow shown in Fig. 3.12. In this workflow, the first step of the registration is the determination of rotations, which can be achieved by matching the accumulated spectrum in the Fourier domain, which is invariant to scaling and translation. Afterward, the scaling can be estimated using the rotationally aligned data by an adaptive FMT method. Finally, the 3D translation can be purely estimated by the shift estimation method, namely the robust 3D phase correlation by matching the 3D data, which has been re-scaled and re-rotated.

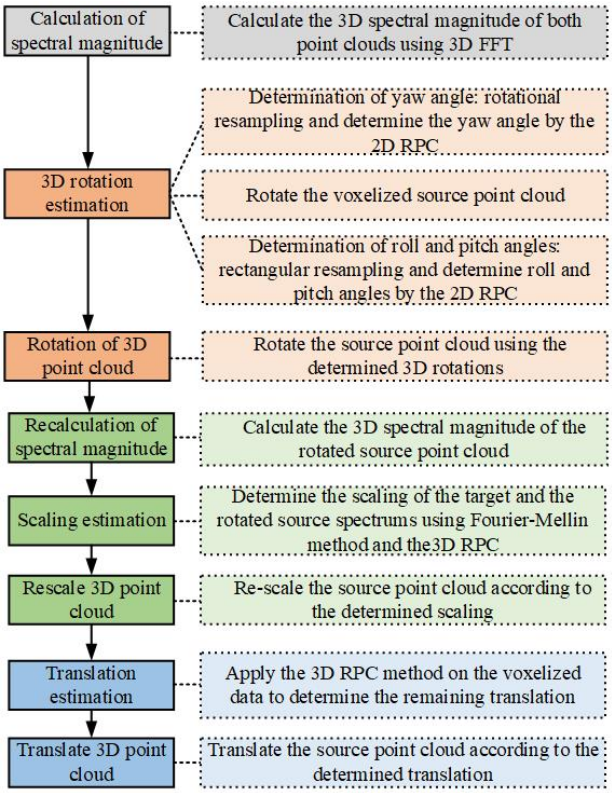

Figure 3.12: Detailed steps of the GRPC method. Gray block stands for the transformation for the spatial domain to the frequency domain. Red blocks denotes the rotation estimation. Green blocks represent the scaling estimation. Blue blocks display the translation estimation. RPC denotes robust phase correlation.

Since the GRPC method is under the framework of coarse registration, if more precise results are required, fine registration methods (i.e., ICP) can be conducted as a subsequent step to improve the registration accuracy.

3D Rotation estimation

As presented in Section 3.2.2, rotations are presented as rotations of points on an accumulated spherical layer. In order to recover 3D rotations from the corresponding rotated spherical structure, we intend to use the similar solutions to the way we use for the translation estimation, which is solved analytically. One solution is to use spherical harmonics [Bülow & Birk, 2018]. However, the main limitation is that the rotational information is recovered based on the standard cross-correlation, but the cross-correlation yields several peaks. There is a same problem as we have mentioned that even though the peak can be found, it is hard to achieve sub-voxel level interpolation since no closed-form way is provided. So the general idea is to resample the hemisphere of the accumulated spectrum. However, since the resampled layer is not an intrinsically 2D rectangular matrix, the structural distortions are dealt with a two-step strategy [Bülow & Birk, 2012]. First, the yaw angle is determined following the rotational behavior of the spherical structure. Then, the 3D spectrum is rotated according to the determined yaw angle. After the rotation, only roll and pitch angles remain their influence on the spherical structure. Thus, the remaining problem is to estimate roll and pitch by resampling the hemisphere in a rectangular way.

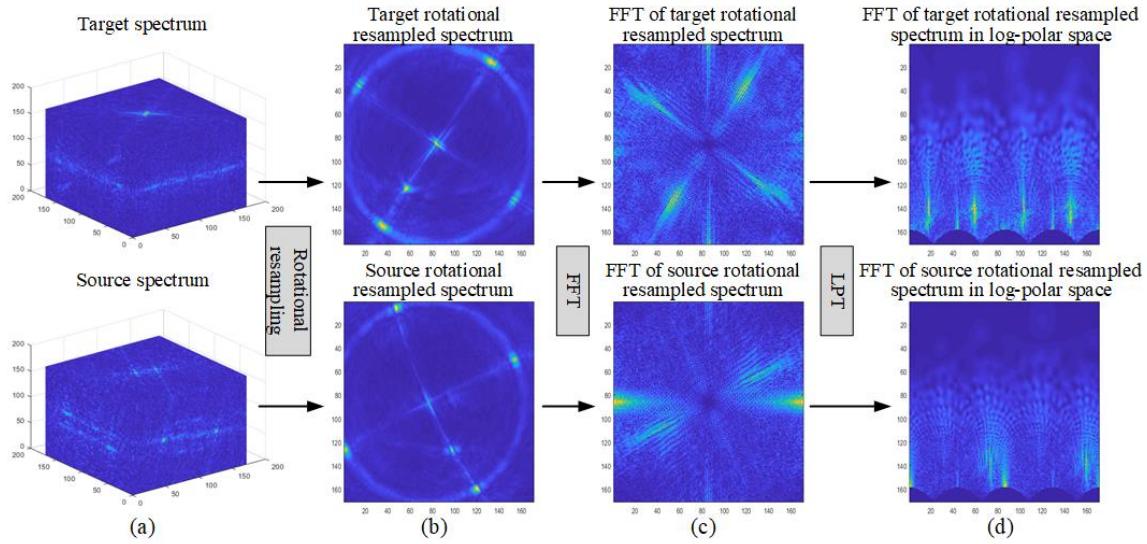

Figure 3.13: Illustration rotational resampling. a) Spectrums from 3D signals, b) rotational resampled spectrum, c) FFT of rotational resampled spectrum, d) FFT of rotational resampled spectrum in log-polar space.

Determination of yaw angle

The general idea to determine the yaw angle is to treat it as a rotation of a resampled structure. The structure is resampled along with spherical coordinates. The accumulated spectrum can be expressed with a resampled spherical coordinate system. The coordinate system is as follows:

$$\begin{cases} v_i = 1, \dots N_{rot} \\ v_j = 1, \dots N_{rot} \\ \phi = arctan(\frac{v_i}{v_j}) \\ \theta = (v_i^2 + v_j^2)^{\frac{1}{2}} \frac{\pi}{N_{rot}} \end{cases}, \tag{3.17}$$

where N_{rot} denotes the size and v_i and v_j present the coordinates of the resampled images.

In accordance with the spectral magnitude, the spherical coordinates can be given as:

$$\begin{cases} u = rsin(\theta)cos(\phi) + \frac{N}{2} \\ v = rsin(\theta)sin(\phi) + \frac{N}{2} \\ w = rcos(\theta) + \frac{N}{2} \end{cases}, \tag{3.18}$$

$$f_{rot}(v_i, v_j) = \sum_{r=r_s}^{r=r_e} F(u, v, w) \tag{3.19}$$

In the resampled matrix, roll and pitch are shown as undesirable interference, which displays roughly as shifts between the matrices in x- and y-directions.

In order to recover the rotation between the resampled images, translation can be decoupled by calculating the Fourier magnitude spectrum. Then, the estimation of rotation can be transformed

to a shift estimation problem by LPT as illustrated in Fig. 3.13, where the spectrum of the two signals can be expressed as:

$$|F(r,\theta)| = |G(r,\theta+\theta_0)|, \tag{3.20}$$

$$|F(\log r,\theta)| = |G(\log r,\theta+\theta_0)|. \tag{3.21}$$

It is clear that the rotation is converted to a shift between the two signals. Thus, by finding the shift (x_0, y_0) in the log-polar space, the rotation can be estimated:

$$\theta_0 = y_0. \tag{3.22}$$

Determination of roll and pitch angles

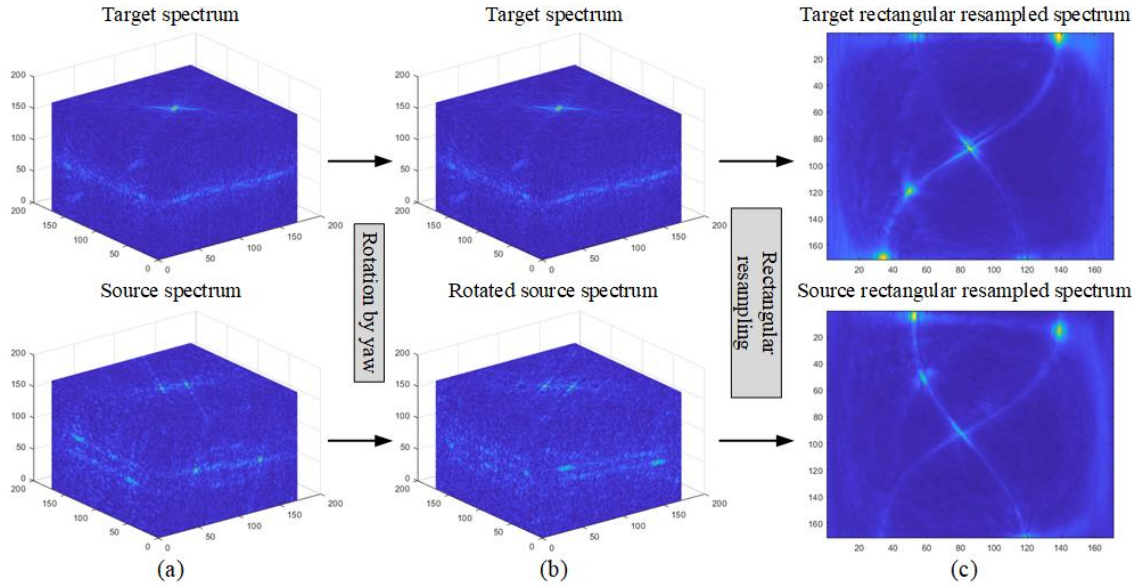

Figure 3.14: Illustration of rectangular resampling. a) Spectrums from 3D signals, b) target spectrum and rotated source spectrum, c) retangular resampled spectrum.

Different from the determination of yaw angle, roll and pitch angles are estimated simultaneously. First, the 3D voxel data is rotated based on the formerly estimated yaw. Then, the rotated spectral magnitude is attained using the same step, as mentioned before. For determining roll and pitch, the spectrum is re-sampled in a rectangular way by a perpendicular projection of the hemisphere into a matrix. Shifts between matrixes can roughly represent the roll and pitch. The resampled coordinate system can be expressed as:

$$\begin{cases} \gamma = -\dfrac{\pi}{2}\left(\dfrac{v_k - N_{rect}/2}{N_{rect}/2}\right), v_k = 1, \dots N_{rect} \\ \psi = \dfrac{\pi}{2}\left(\dfrac{v_l - N_{rect}/2}{N_{rect}/2}\right), v_l = 1, \dots N_{rect} \end{cases}, \tag{3.23}$$

where N_{rect} is the square size of the rectangular resampled images.

Correspondingly, the related accumulated spectrum can be calculated as:

$$\begin{cases} u = r\sin(\gamma)\cos(\psi) + \dfrac{N}{2} \\ v = r\sin(\gamma) + \dfrac{N}{2} \\ w = r\cos(\gamma)\cos(\psi) + \dfrac{N}{2} \end{cases}, \tag{3.24}$$

$$f_{rect}(v_k, v_l) = \sum_{r=r_s}^{r=r_e} F(u, v, w). \tag{3.25}$$

By determining the shifts using the same phase correlation method, roll and pitch can then be estimated. Since all rotation parameters have been estimated, the spectrum can be re-rotated. Only the scaling remains to influence the structure information of the spectrum magnitude.

Scaling estimation using FMT

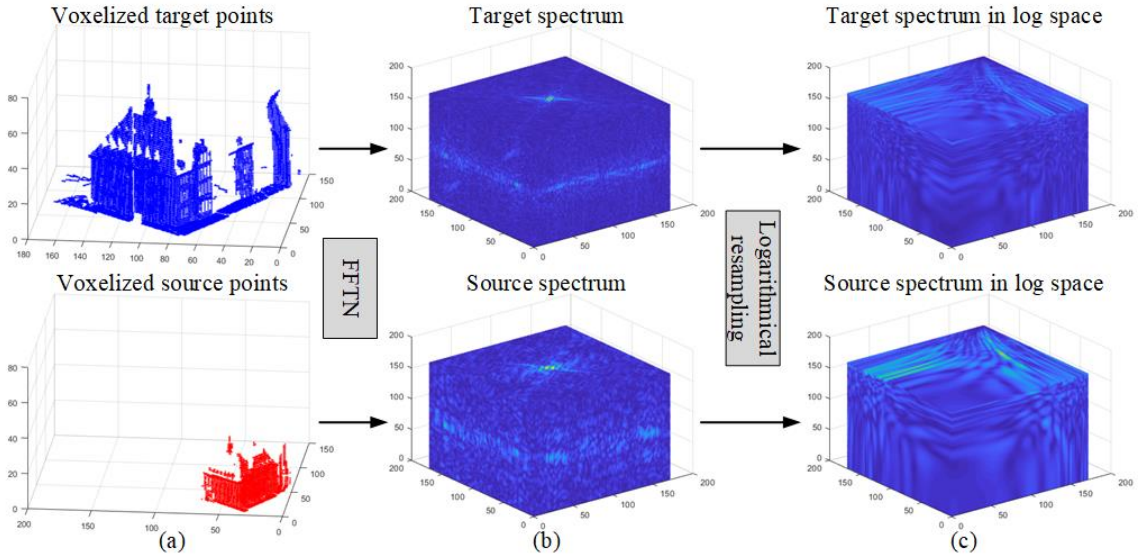

Figure 3.15: Scaling estimation using FMT. a) Discrete 3D signals, b) spectrum of discrete 3D signals, c) Spectrum in log space after FMT.

As mentioned in the previous sections, the radial accumulation of the spectral data is scale-invariant, which allows for the rotation-only registration. In this section, since the rotations have already been determined, Eq. 3.12 can be simplified as:

$$|R(\mathbf{k})| = \psi^3 |S(\mathbf{k}\psi^{-1})|. \tag{3.26}$$

The spectral magnitude can be transformed into a log space by FMT, in which the Fourier magnitude spectrum is related to each other by:

$$|R(log(\mathbf{k}))| = \psi^3 |S(log(\mathbf{k}) - log(\psi))|, \tag{3.27}$$

which illustrates that spectral structure is logarithmically deformed along each direction. By taking the log transformation, the scaling is changed as a shift in each direction. Thus, by finding the shift (x_0, x_0, x_0) between the two spectra in the log space, the scaling factor can be estimated as:

$$\psi = e^{x_0}. \tag{3.28}$$

Note that there are common causes that the shifts in x-, y-, z-directions are different. However, under the assumption that we solve registration with seven DOFs, the influence of scaling change on the spectrum's structure along each direction should be the same. Instead of adding a constraint, we apply an easy way under this situation. By estimating the 3D shift between the spectral structures in the log domain, the shifts in different directions can be determined. Thus, scaling for different directions can be easily calculated using Eq. 3.28. Subsequently, by finding the scaling that can produce the phase correlation's maximum peak, the scaling can be chosen among the three scaling factors.

Translation estimation using 3D phase correlation

Once the rotations and scaling have been determined, point clouds can be aligned according to the estimated parameters. Only translation remains. Thus, the further step is straightforward: the determination of 3D translation and can also be achieved simply by the 3D phase correlation. Without further procedures (i.e., transferring to other domains or conducting a resampling process), the translation can be directly determined by the proposed phase correlation method in the time domain using the aligned voxelized data. Assume that the estimated shifts are calculated as $(\Delta X, \Delta Y, \Delta Z)$. Afterward, considering the difference of the coordinates (X_0, Y_0, Z_0) calculated from the rough alignment, the estimated 3D translations should be (X_0+ΔX, Y_0+ΔY, Z_0+ΔZ), which are the final outputs.

4 Semantic segmentation of urban scenes

Semantic labeling is an essential but challenging task when interpreting point clouds of 3D scenes. As a core step for interpretation, semantic labeling is to annotate every point in the point cloud with a label of semantic meaning, which plays a significant role in many point cloud related applications. In this chapter, three methods [Huang et al., 2019; Huang et al., 2020b, 2021] regarding the semantic segmentation of point clouds will be introduced, including Multi-scale local context embedding for semantic segmentation (MLCE), deep point embedding for semantic segmentation (DPE), and a global relation-aware attentional neural network for semantic segmentation (GraNet). These methods involve feature embedding of local context information, nonlinear manifold learning for feature dimension reduction, global graph-based optimization for refinement, and relation-based modules to a network structure. In Fig. 4.1, we show the key methods corresponds to the ones we have shown in our research frame in Section 1.3.

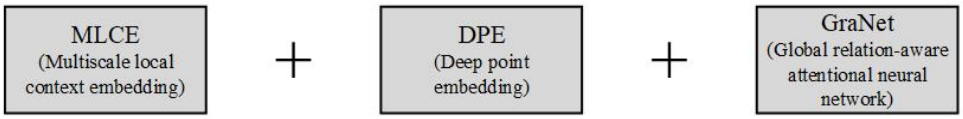

Figure 4.1: The proposed methods for semantic segmentation that included in this chapter.

4.1 Multi-scale local context embedding for semantic segmentation (MLCE)

The first method is multi-scale local context embedding, which includes two major stages: (I) extract point-based multi-scale geometric features; (II) reduce the dimensionality of features. The workflow of the proposed method is illustrated in Fig. 4.2.

4.1.1 Extraction of point-based multi-scale geometric features

Feature extraction is a crucial part in point cloud classification, and its performance plays an important role in the quality of the classification results. However, it remains a challenging task to extract sufficient information from raw points. In this method, we construct a set of geometric features including surface features, statistical features, dimensionality features, height features and orientation features. Specifically, they consist of local density D_k, omnivariance O_k, anisotropy A_k, eigenentropy E_k, local curvature C_k, sum of eigenvalues $\sum_k$, geometric center $\hat{\mathbf{X}}_k$, linearity L_k, planarity P_k, scattering S_k, height mean $\bar{H}_k$, height difference ΔH_k, normal vector $\mathbf{n}_k$, and verticality V_k (see Table 4.1). These features are used to represent the local geometric shape based on eigenvalue decomposition of 3D structure tensor formed by a specific neighborhood of the point p [Weinmann et al., 2015a].

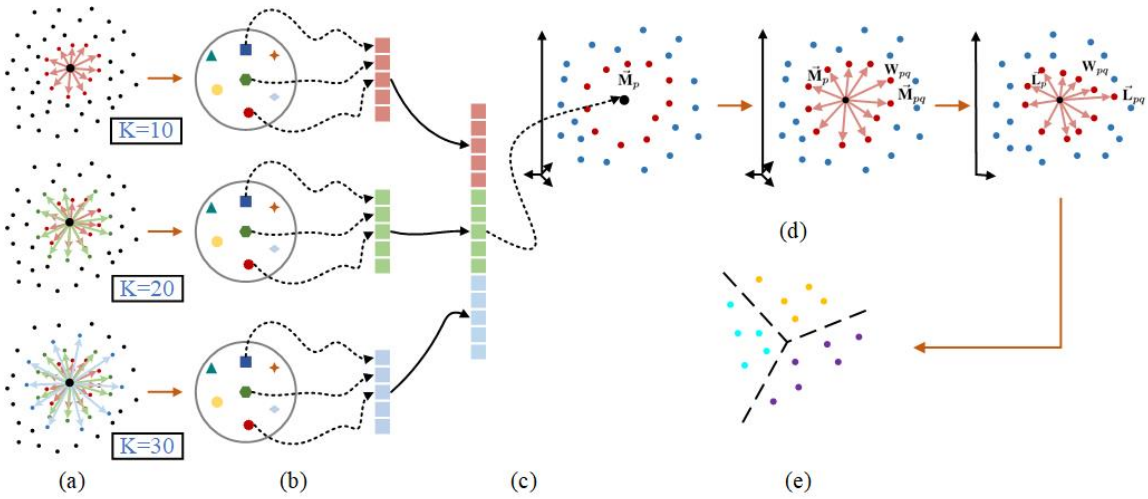

Figure 4.2: Workflow of the MLCE point cloud classification strategy. a) Scaling, b) geometric space, c) concatenation, d) embedding, e) classification.

Table 4.1: List of used features in MLCE.

Category	Features	Definition
Surface	Local density	$D_k = \frac{3n}{4\pi {r_{seed}}^3}$
Statistics [Chehata et al., 2009]	Omnivariance	$O_k = \sqrt[3]{e_1 \cdot e_2 \cdot e_3}$
	Anisotropy	$A_k = (e_1 - e_3)/e_1$
	Eigenentropy	$E_k = -\sum_{i=1}^{3} e_i \ln(e_i)$
	Local curvature	$C_k = \frac{e_3}{e_1+e_2+e_3}$
	Summation	$\sum_k = e_1 + e_2 + e_3$
	Center	$\hat{\mathbf{X}}_k = \frac{1}{k+1} \sum_{i=0}^{k} \mathbf{X}_i$
Dimensionality [Weinmann et al., 2015a]	Linearity	$L_k = \frac{e_1 - e_2}{e_1}$
	Planarity	$P_k = \frac{e_2 - e_3}{e_1}$
	Scattering	$S_k = \frac{e_3}{e_1}$
Height [Maas, 1999]	Height mean	$\bar{H}_k = \frac{1}{n} \sum_{i=1}^{n} Z_i$
	Height difference	$\Delta H_k = Z_{max} - Z_{min}$
Orientation [Rabbani et al., 2006]	Normal vectors	N_x
		N_y
		N_z
	Verticality	$1 - N_z$

The geometric features are also influenced by the neighborhood scale. Thus, to consider different local context, the geometric features of the query point are calculated using three different neighborhood size k. Here, k=k_0, k_1, k_2, and in this paper k_0=10, k_1=20, k_2=30 (determined based on the point density). Then the geometric features are concatenated to form the multi-scale feature vector to represent the 3D point cloud scene.

4.1.2 Low-dimensional embedding of multi-scale features

Multi-scale feature modeling can provide the geometric information more sufficiently by considering the local neighboring context of a given point. To some extent, the redundancy of the features hinders the classification performance from further going better. For this reason, feature selection has been used to, to some extent, handle this issue by filtering some features based on the inter-correlations between features or the coherence between features and given labels. It

should be noted that, however, that the new feature space constructed by those selected features still lies in the original space. Beyond the original feature space, we propose to utilize the local manifold learning (LML)-based method (locality preserving projections (LPP) in our case) on the extracted geometric primitive to learn a compact low-dimensional feature representation by locally embedding the neighboring information.

Compared with other dimensionality reduction (DR) methods that maximize or preserve the specific information, such as principal component analysis (PCA) and linear discriminant analysis (LDA), LML-based method is capable of mining the underlying data structure and considering the correlation between points. More specifically, for each point in the multi-scale feature domain, there is a strong spatial correlation between points, particularly their neighborhoods. Intuitively, LPP learns the low-dimensional representation by constructing a neighboring graph for each sample in the high-dimensional feature domain. This might further enhance the connections between the original features of the points and their neighbors.

Compared with other nonlinear LML's approaches, LPP can explicitly project out-of-samples into the learned subspace, owing to its linearized technique. Moreover, a large number of samples are usually used for the model's learning in practice. For these two purposes, the multi-scale geometric features are fed into the LPP, leading to the geometric primitive embedding as the input of the final classifier.

4.2 Deep point embedding for semantic segmentation (DPE)

The second method is the deep point embedding, which is a further development of the MLCE method. The multi-scale feature learning is improved as hierarchical deep feature learning (HDL). The manifold-learning-based method is further optimized by considering the spatial constraint using joint manifold-based embedding (JME). The result of classification can be further optimized by global graph-based optimization (GGO). Fig. 4.3 illustrates the framework presenting the essential steps of the involved methods and sample results. The detailed explanation of each step in the framework is introduced in the following sub-sections. The HDL step is designed

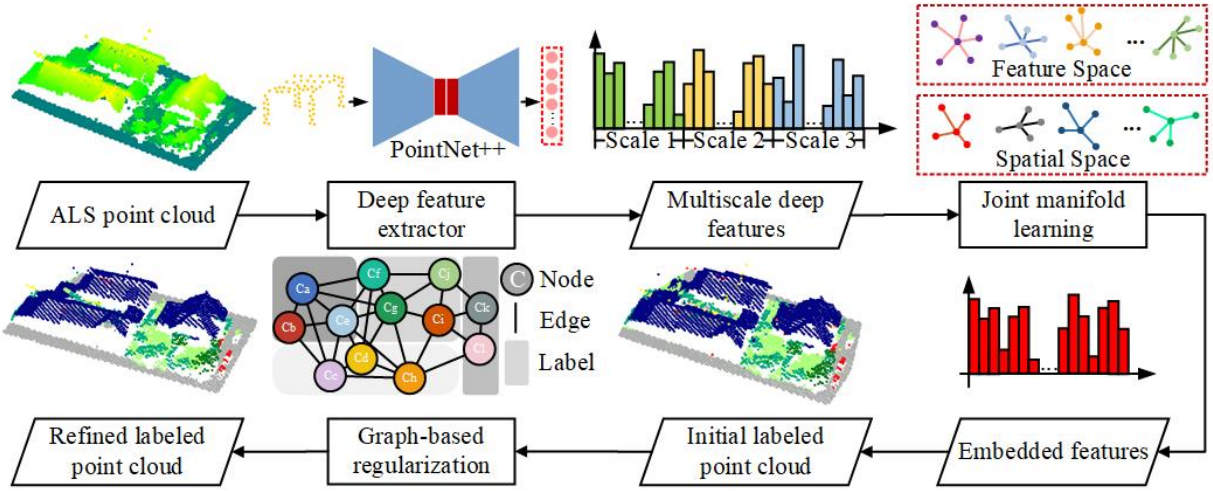

Figure 4.3: Workflow of the DPE method.

for learning the original features of points. Here, hierarchical splitting is applied to the input datasets. Specifically, the training and test point clouds are subdivided into non-overlapped sub-pointsets, and then, these pointsets are downsampled to a fixed number, which fulfills the

input requirement of the network. The features extracted from the network comprise point-based description and texture information of the entire sub-pointsets; therefore, the deep features extracted from hierarchical pointsets can be used to represent texture information from different scales. Consequently, these features of the multi-scale inputs are aggregated into a concatenated feature vector. Moreover, considering the high redundancy and regional dependence of the feature vector, we use a manifold-learning-based algorithm to generate the representation with a lower dimension for describing each point in a globally optimal manner. Considering the high correlation between neighboring points, we apply JME for dimensionality reduction; JME operates in both high-dimensional feature and spatial domains. The initial classification results can be obtained by feeding the embedded features into a classifier, e.g., random forest (RF) or support vector machine (SVM). Finally, GGO is employed for regularizing the initially labeled points to achieve locally smooth and globally optimal classification results. In the following sub-sections, we present the essential algorithms in our workflow in detail.

4.2.1 Hierarchical deep feature learning (HDL)

PointNet proposes a solution for spatial encoding considering the raw nature of 3D points, thereby enabling the processing of 3D points in a direct manner. Compared to PointNet, fine-grained patterns can be recognized in PointNet++, which can help produce better results in the case of scenes with high complexity. Moreover, PointNet++ shows higher robustness to the variation of point densities. Consequently, it is chosen for extracting deep features in the framework of DPE.

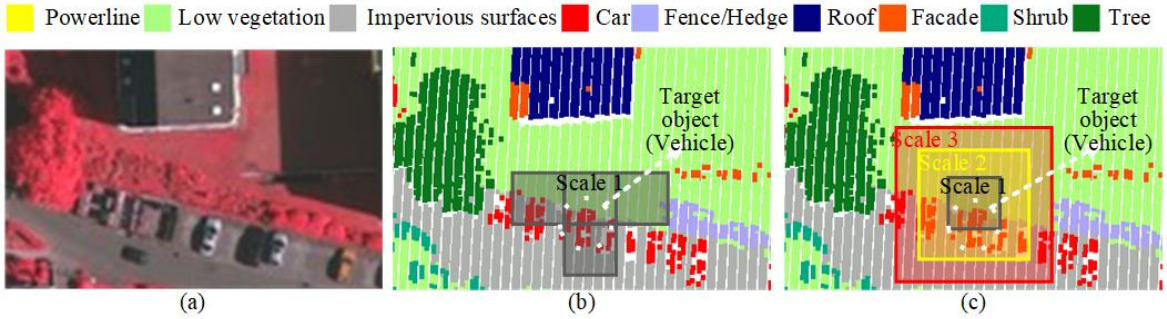

Figure 4.4: Reason for adopting hierarchical subdivision strategy. a) An illustration of the real scene as true orthophoto, b) the target vehicle is cut into pieces by the single-scale subdivision with fixed template size, c) the target vehicle can be fully represented by multiscale sub-pointsets in different template sizes. Labels are used for better illustration.

However, although PointNet++ has higher generalization ability than PointNet, its usage in the classification of large-scale airborne point cloud in complex urban scenes poses challenges. For the training and inference of a network, the process of splitting and sampling is inevitable. However, as shown in Fig. 4.4, artifacts may be induced in this process. As shown in the figure, small objects are cut into pieces; this process is unable to provide sufficient information for identifying the small pieces. Therefore, to improve the ability of the network to deal with different-scale objects, we enhance the ability of PointNet++ to handle complex 3D data via a hierarchical data augmentation method. The hierarchical sampling strategy provides a trade-off solution for object integrity and fine-grained details. Specifically, HDL is conducted in the three steps in Fig. 4.5. First, during the training stage, three rounds of subdivisions are repetitively implemented to the entire point cloud used for training. The scale of the sub-pointset is different in each round, and the sub-pointset is presented as a fixed number of points. In other words, in each round of the subdivision, a different scale will be used to constrain the size of sub-pointsets, in order to

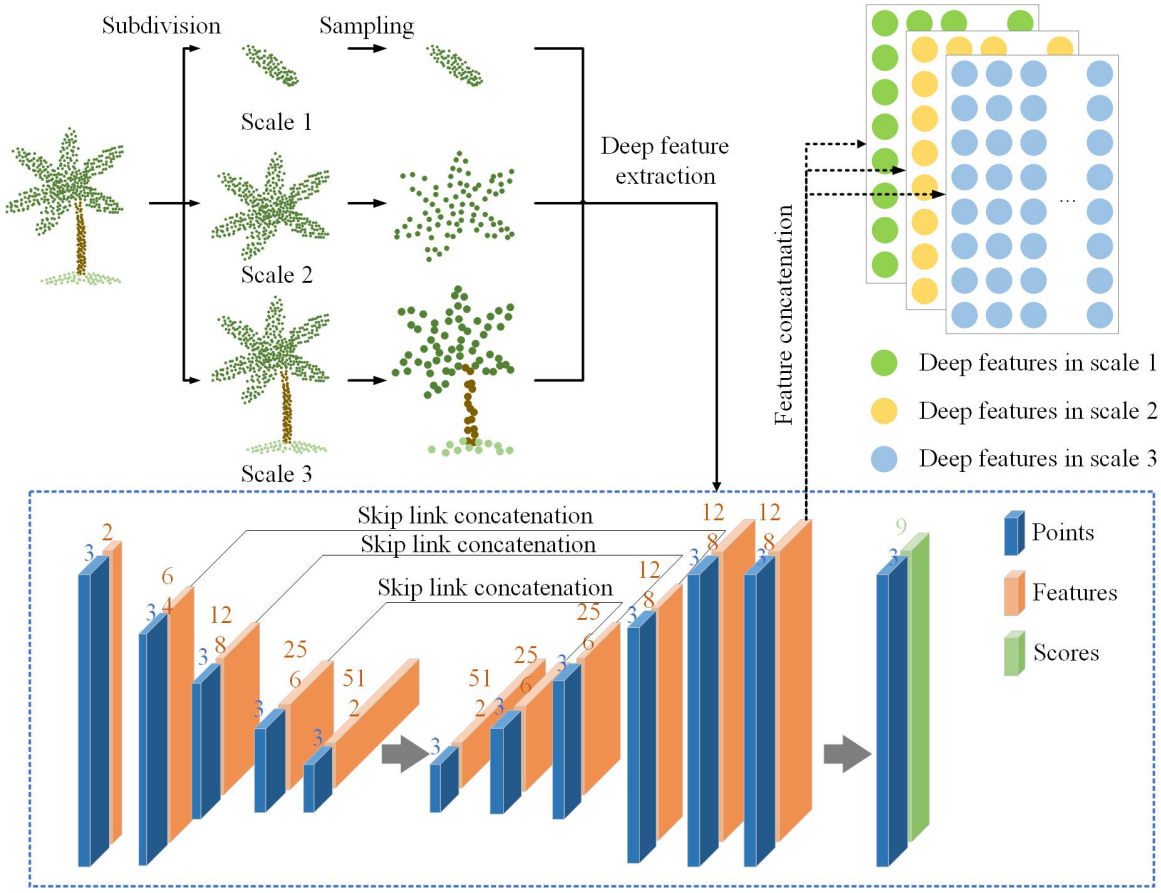

Figure 4.5: HDL for deep feature learning.

subdivide the point cloud into non-overlapped sub-pointsets having a predefined and fixed number of points. Therefore, after the three-round subdivisions, the whole scene is presented with sub-pointsets on different scales. All these sub-pointsets are fed into the network after a downsampling step to ensure the inputs are provided in a consistent manner. In this case, the network can be trained to have a stronger generalization ability while dealing with widely varied-scale objects. Secondly, the point clouds for testing are processed identically for subdivision and downsampling. However, the difference between the training and test processes is that the subdivided pointsets in different scales are not fed into the network together. The subpoints in each scale are fed to obtain the deep features. Then the points in the original point clouds that are not contained in the inputs for the network are interpolated to obtain deep feature vectors. Consequently, all the points subdivided with different scales can be assigned a deep feature vector that contains a different level of contextual information. Finally, the deep features from different scales can be concatenated to form a multiscale deep feature vector.

4.2.2 Joint manifold-based embedding (JME)

Although multi dimensional features (MDF) provide contextual information, they have high redundancy, inter-correlation, and regional dependence, which is counterproductive to the improvement of feature discrimination. Therefore, with an aim to reduce feature correlation and simplify feature representation with a lower dimensionality, we developed a manifold learning algorithm to mine the underlying data properties and leverage the optimal embedded feature domain.

Based on the original local linear embedding (LLE) strategy introduced in Section 2.2, to extract better representative features of 3D points, we apply JME, thereby improving both the robustness of the embedder and the ability of incorporating spatial information. Here, the embedding is calculated with the addition of spatial information under the assumption that neighboring 3D points in a spatial domain should share similar reconstruction weights. In such a scenario, we propose a method that embed both contextual and spatial information of 3D points under the LLE framework with. The JME method comprises three steps: recovery of the spatial neighborhood, computation of reconstruction weight with the spatial constraint, and calculation of joint embedding, which is illustrated Fig. 4.6. It is noteworthy that, for 2D images, spatial correlation can be easily represented by four-neighborhood or eight-neighborhood. However, for 3D point clouds, the distribution of points is unstructured, and consequently, it is hard to be modeled or rasterized. Therefore, we use K-nearest neighborhood (KNN) to define the spatial correlation between points, and the spatial neighborhood is established by these points. Furthermore, the calculation of the reconstruction weights can be formulated by adding the spatial constraints of these neighboring points:

$$\mathbf{r}_i^0 = \arg\min_{\mathbf{r}_i^0} \left\{ \sum_{k=0}^{K} ||\mathbf{x}_{ik} - \mathbf{X}_i^l \mathbf{r}_{ik}||_2^2 \right\}, \tag{4.1}$$

where $k = 0, 1, ..., K$ denotes the spatial neighboring points of the target points defined by KNN, and $\mathbf{X}_i^l = [\mathbf{x}_i^1, \mathbf{x}_i^2, ...\mathbf{x}_i^l]$ denotes the l neighbors in the domain selected by KNN. It should be noted that the selection of K for spatial neighborhood selection is limited by computation resources. Empirically, we find $K = 5$ to be sufficient.

The constraint of this equation is represented as:

$$\begin{aligned} &||\mathbf{X}_i^l (K\mathbf{r}_i^0 - \sum_{k=0}^{K} \mathbf{r}_i^k)|| \leq \epsilon \\ &(\mathbf{r}_i^k)^{\mathrm{T}} \mathbf{r}_i^k = 1, \text{ where } k = 0, 1, ..., K, \end{aligned} \tag{4.2}$$

where ϵ denotes the tolerant error. This problem can be seen as a joint optimization problem

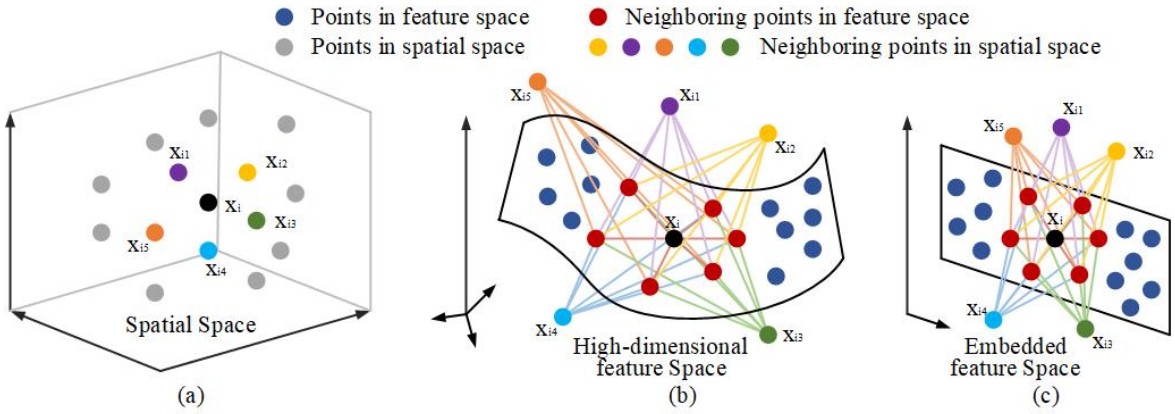

Figure 4.6: Integration of spatial information in joint embedding. a) Recovery of the spatial neighborhood, b) computation of reconstruction weight with the spatial constraint, c) calculation of joint embedding.

that can be rewritten as:

$$\mathbf{q}_i^0 = \arg\min_{\mathbf{q}_i^0} \left\{ \sum_{k=0}^{K} ||\hat{\mathbf{X}}_i^l - \mathbf{A}\hat{\mathbf{R}}_i||_2^2 \right\}, s.t. \mathbf{E}\hat{\mathbf{R}}_i = \mathbf{t}^{\mathrm{T}} = [1, 1, ..., 1]^{\mathrm{T}}$$

$$\mathbf{A} = \begin{bmatrix} K\alpha\mathbf{X}_i^l & -\alpha\mathbf{X}_i^l & -\alpha\mathbf{X}_i^l & \dots & -\alpha\mathbf{X}_i^l \\ \mathbf{X}_i^l & & & & \\ & \mathbf{X}_i^l & & & \\ & & \mathbf{X}_i^l & & \\ & & & \ddots & \\ & & & & \mathbf{X}_i^l \end{bmatrix} \tag{4.3}$$

$$\hat{\mathbf{R}}_i = \begin{bmatrix} \mathbf{q}_i^0 \\ \mathbf{q}_i^1 \\ \vdots \\ \mathbf{q}_i^K \end{bmatrix}, \hat{\mathbf{X}}_i^l = \begin{bmatrix} 0 \\ \mathbf{x}_{i0}^1 \\ \mathbf{x}_{i1}^1 \\ \vdots \\ \mathbf{x}_{iK}^1 \end{bmatrix}, \mathbf{E} = \begin{bmatrix} \mathbf{e} & & & & \\ & \mathbf{e} & & & \\ & & \mathbf{e} & & \\ & & & \ddots & \\ & & & & \mathbf{e} \end{bmatrix}.$$

where the sizes of $\mathbf{t}$, $\mathbf{A}$, $\hat{\mathbf{R}}_i$, $\hat{\mathbf{X}}_i^l$, and $\mathbf{E}$ are $1 \times (K+1)$, $(K+2)L \times (K+1)l$, $(K+1)L \times 1$, $(K+2)L \times 1$, and $(K+1) \times (K+1)l$, respectively. Here, $\mathbf{e}$ is a unit vector with size $1 \times l$. α is the coefficient used to balance the error and the constraint.

$$\mathbf{r}_i^0 = \arg\min_{\mathbf{r}_i^0} \left\{ \sum_{k=0}^{K} ||\hat{\mathbf{X}}_i^l - \mathbf{A}\hat{\mathbf{R}}_i||_2^2 + \gamma||\mathbf{E}\hat{\mathbf{R}}_i - \hat{\mathbf{e}}||_2^2 \right\}, \tag{4.4}$$

where $\hat{\mathbf{e}}$ is a unit vector with size $(K+1) \times 1$.

The weight vector can be solved as:

$$\mathbf{r}_i^0 = (\mathbf{A}^{\mathrm{T}}\mathbf{A} + \gamma\mathbf{E}^{\mathrm{T}}\mathbf{E})^{-1}(\mathbf{A}^{\mathrm{T}}\mathbf{X}_i^l + \gamma\mathbf{E}^{\mathrm{T}}\hat{\mathbf{e}}). \tag{4.5}$$

Finally, we use these weight vectors to solve the embedding problem by using Eq. 2.16 in Section 2.2.

In the proposed representation, each point in the original data is first represented by regional deep features extracted from a point-based neural network. Then, these features are embedded with spatial information by excavating the local manifold structure, wherein these features are optimized based not only on the local data but also the global feature correlations.

Large-scale LML

For commonly used small-sized 2D datasets, the LLE-based methods always work quite well and provide satisfactory outputs [Bachmann et al., 2005; Ma et al., 2010; Hong et al., 2017]. However, for an enormous number of points in a point cloud, the computation of low-dimensionality embedding is challenging owing to the massive amount of data. More than millions of points are collected for each point cloud; this renders algorithms with high demands for memory and computational cost impractical. To mitigate this problem, it is essential that both construction of the affinity matrix and spectral decomposition of the matrix are intractable for a large number of samples. Therefore, under the framework of LML, we propose a new algorithm, KNN joint manifold embedding (K-JME), to improve efficiency by adopting k-means clustering before embedding. This strategy can be applied not only to our joint embedding methods but also in other LML-based algorithms. The improved algorithm mainly comprises three steps. Firstly, M cluster centers are obtained by the k-means clustering, and they are denoted as: $\mathbf{X}_c = [x_{1c}, x_{2c}, ..., x_{Mc}] \in \mathcal{R}^{M \times D}$.

Then the joint manifold learning algorithm is executed to acquire a representation with reduced dimensionality of the M cluster centers by using Eq. 2.16 in Section 2.2.

$$\hat{\mathbf{Y}}_c = \arg\min_{\mathbf{Y}} \left\{ \mathbf{Y}_c \mathbf{L_c} \mathbf{Y}_c^{\mathrm{T}} \right\}, s.t. \mathbf{Y}_c \mathbf{P} \mathbf{Y}_c^{\mathrm{T}} = \mathbf{I}, \tag{4.6}$$

where $\mathbf{L}_c = \mathbf{D}_c - \mathbf{W}_c$, and $\mathbf{W}_c$ can be calculated using Eq. 4.1.

Finally, the local manifold structures obtained with these M cluster centers serve as the manifold structure of the original data. In this manner, we can reconstruct the manifold structure of the original data using the correlations between the original data points and the clusters. The reconstruction weights can be calculated by using Eq. 4.1 as follows:

$$\mathbf{w}_{ci}^0 = \arg\min_{\mathbf{w}_{ci}^0} \left\{ \sum_{k=0}^{K} ||\mathbf{x}_{ik} - \mathbf{X}_{ci}^l \mathbf{w}_{cik}||_2^2 \right\}, \tag{4.7}$$

where $k = 0, 1, ..., K$ denotes the spatial neighboring clusters of the target points in the original data defined by KNN, and $\mathbf{X}_{ci}^l = [\mathbf{x}_{ci}^1, \mathbf{x}_{ci}^2, ...\mathbf{x}_{ci}^l]$ denotes the l neighbors in the feature domain selected by KNN.

Assuming that the local manifold structures in high-dimensional and low-dimensional spaces are consistent, the low-dimensional representation of the original data points can be obtained by

$$\mathbf{y}_c^i = \mathbf{w}_c^i \times \mathbf{Y}_c^i, \tag{4.8}$$

where $\mathbf{y}_c^i \in \mathcal{R}^{1\times d}$ denotes the reduced dimensional representation of the i-th data point. $\mathbf{w}_c^i = [\mathbf{w}_{c1}, \mathbf{w}_{c2}, ..., \mathbf{w}_{cl}] \in \mathcal{R}^{1\times l}$ stands for the reconstruction weights calculated between the data point and its l-nearest cluster centers in the high-dimensional feature space constrained by K spatial neighboring clusters, and $\mathbf{Y}_c^i \in \mathcal{R}^{l\times d}$ is the selected l d-dimensional representation from the obtained low-dimensional representation of M cluster centers, $\mathbf{Y}_c^i = [\mathbf{y}_{c1}, \mathbf{y}_{c2}, ..., \mathbf{y}_{cl}] \in \mathcal{R}^{M\times d}$.

Finally, the high-dimensional deep features are embedded into an optimal low-dimensional space. The embedded features are classified with a specific classifier. We use RF in the classification step to obtain initial classification results because RF is a robust classifier, and it is influenced by less parameters compared to other classic classifiers. Furthermore, the optimal number of decision trees is determined by cross-validation.

4.2.3 Global graph-based optimization (GGO) for labeling refinement

After the classification of the aforementioned embedded features with a specific classifier, the initial results can be further optimized to obtain locally smoothed results. Under the mathematical framework presented in Landrieu et al. [2017], a probabilistic model is applied to use a graph structure to model spatial correlations. Labeling refinement is conducted by searching for the optimal labels to improve the regularity with the graph-structured optimization. As illustrated in Fig. 4.7, it can be divided into three steps: construction of the graph, construction of the energy cost, and the solution. The regularization can be converted into a graph-structured problem. The graphical model is constructed by the undirected adjacency graph, which is defined by ten KNN neighboring points, and is expressed as $\mathcal{G} = \{V, E\}$, where V denotes the nodes (representing the points in 3D space) and E denotes a set of edges defined by the correlations of the neighboring points. With respect to the edge weights, we did not use the commonly adopted spatial distances or constant values for estimating weights. Instead, we measured the custom distance in the feature space (i.e., the feature space with reduced dimensions) between points, and used this distance to weight the edge. To obtain optimal labels using the constructed graph, the problem can be

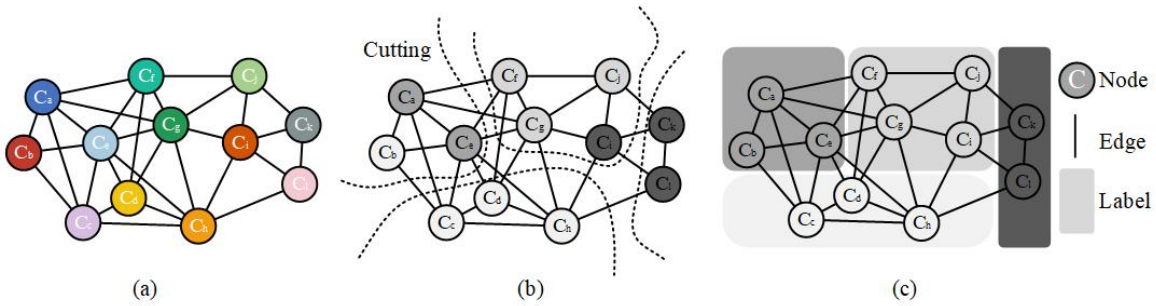

Figure 4.7: GGO. a) Graph construction, b) graph cut, c) refinement of labels.

formulated as a problem that aims at minimizing the energy cost, and it is defined as :

$$\mathbf{E}(L) = \alpha \mathbf{R}(L) + \mathbf{\Phi}(L), \tag{4.9}$$

where $L = [l_1, l_2, ...l_n]$ $(1 \leq l_i \leq m)$ denotes a set of labels with m categories. Moreover, α is a factor used to balance the influence between the local smoothness, $\mathbf{R}(L)$, and the penalty inducing the fidelity, $\mathbf{\Phi}(L)$. With this configuration, the optimization process of the classification results can be achieved to make the labels locally continuous and globally optimal. The two terms on the right-hand side have different definitions in various energy functions. In our framework, the term for regional smoothing can be defined as:

$$\mathbf{R}(L) = \sum W_{u,v} \delta(V_u, V_v), \tag{4.10}$$

where $\delta(V_u, V_v)$ penalizes adjacent points, V_u and V_v, of different labels, l_u and l_v, and it is obtained by the Potts model [Potts, 1952]:

$$\delta(V_a, V_b) = \begin{cases} 0 & \text{if } l_a = l_b \\ 1 & \text{if } l_a \neq l_b \end{cases}, \tag{4.11}$$

where $W_{u,v}$ is a positive value related to the strength of the penalty using the distance constraint.

$$W_{u,v} = \exp(-||X_u - X_v||^2 / 2\sigma^2), \tag{4.12}$$

where X_u and X_v denote the representations of the points in the given feature space (i.e., the feature vectors). Furthermore, the fidelity term, $\mathbf{\Phi}$, enforces the influence of the labels

$$\mathbf{\Phi}(L) = -\sum l_k \log(\hat{l}_k), \tag{4.13}$$

where l_k denotes the initial label of the point, whereas $\hat{l}_k$ represents the optimized label of the point.

The aforementioned cost function can be easily solved by employing a graph-cut strategy using the alpha-expansion, which can quickly find an approximate solution with a few graph-cut iterations [Boykov et al., 2001; Kolmogorov & Zabih, 2004; Boykov & Kolmogorov, 2004]. Here, the labeling cost is not considered because we assume that the labels of all objects are independent; consequently, all elements in the labeling cost matrix are set to 1, except the diagonal ones that are set to 0. The optimization results automatically adapt to the underlying scene without the need for predefined features of certain potential objects.

4.3 A global relation-aware attentional neural network for semantic segmentation (GraNet)

The third method is a novel neural network focusing on semantic labeling of point clouds, which investigates the importance of long-range spatial and channel-wise relations and is termed as global relation-aware attentional network (GraNet). GraNet first learns local geometric description and local dependencies using a local spatial discrepancy attention (LoSDA) convolution module. In LoSDA, the orientation information, spatial distribution, and elevation information are fully considered by stacking several local spatial geometric learning modules and the local dependencies are learned by using an attention pooling module. Then, a global relation-aware attention (GRA) module, consisting of a spatial relation-aware attention (SRA) module and a channel relation-aware attention (CRA) module, is presented to further learn attentions from the structural information of a global scope from the relations and enhance high-level features with the long-range dependencies. The aforementioned two important modules are aggregated in the multi-scale network architecture to further consider scale changes in large urban areas.

Here, we first introduce the local spatial discrepancy attention convolution module and the design of each module involved in this local encoding module. Then the global relation-based attention module is explained, in which the contextual information is further investigated under the global scope. In Section 4.3.3, the network configuration of the GraNet is presented, in which an encoder-decoder framework is developed on the basis of PointNet++ [Qi et al., 2017b] architecture.

4.3.1 Local spatial discrepancy attention (LoSDA) convolution module

The LoSDA module consists of spatial distribution encoding (SDE), directional feature encoding (DFE), and elevation feature encoding (EFE), and an attention pooling module, which sequentially implements the description of local geometric characteristics and considers the local dependencies. In Fig. 4.8, we illustrate the scheme of the LoSDA module we used, which is applied to each 3D point and outputs aggregated features. It shows how the three modules are integrated to learn the output features. For each input point p_i, the further operations are conducted on the point itself and its KNN points P_i together with attributes or intermediate learned features f_i. First, the directional features are obtained using the DFE module. The former features are concatenated with the directional features. A shared multi-layer perceptron (MLP) is then followed to obtain fused features from the combination of the original features and the directional features to generate orientation-augmented features. Second, the spatial information is processed by two feature encoding modules. Based on the spatial information provided by the neighboring points, the 3D coordinates are fed into the SDE module and the height information is fed into the EFE module. Then, the learned features from these two modules are concatenated and followed by a shared MLP to obtain the spatial features. Finally, the orientation-augmented features and the spatial features obtained in the last two steps are concatenated and fed into the attention pooling module to generate the aggregated features based on all the neighboring features. By encoding information using the aforementioned three modules, we can get an explicit representation of the local spatial information for a given point. Then, we will provide detailed explanations for each module.

Spatial distribution encoding (SDE)

As stated in Hu et al. [2020], given the input point p_i and its KNN point set $P_i = \{p_i^1, p_i^2, p_i^3, ..., p_i^K\}$, the use of KNN points can make point features be aware of their relative spatial locations, indicating the local spatial distribution of points. To encode the spatial distribution of an input

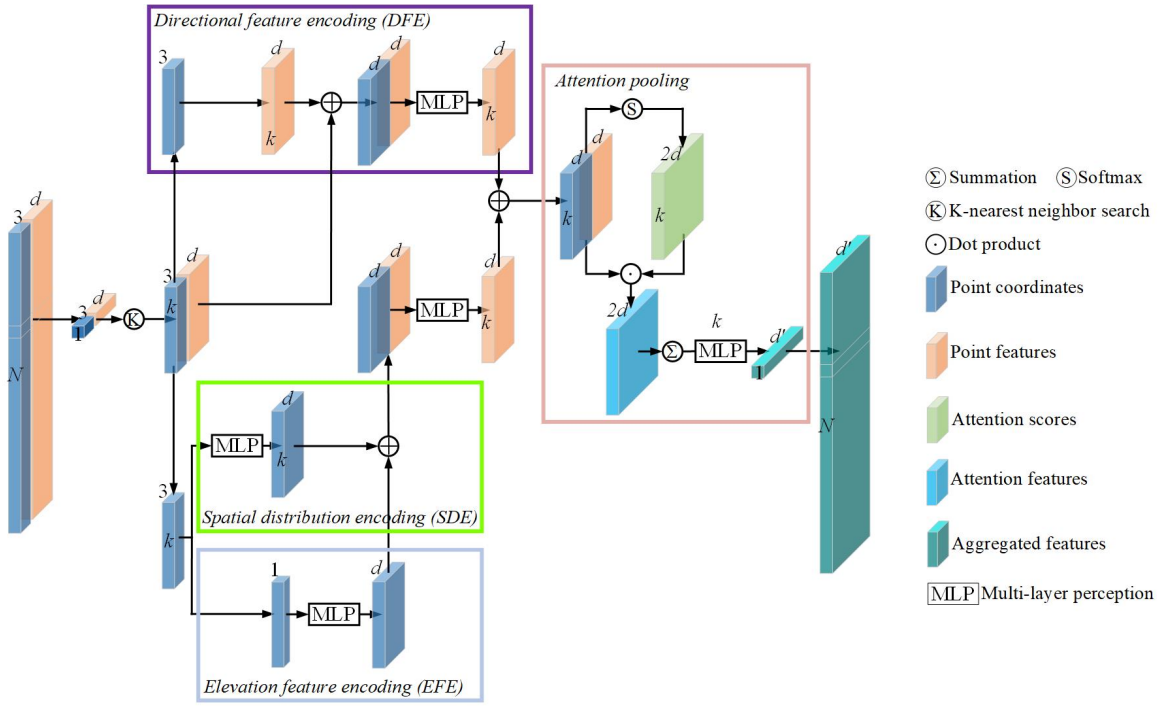

Figure 4.8: Exploited local spatial discrepancy attention encoding module. The purple panel shows the directional feature encoding module. The pink panel shows the attention pooling module. The light green panel shows the spatial distribution encoding module and the blue one shows the elevation feature encoding module.

point p_i in its KNN point set P_i, their relative point positions will be utilized, which is calculated as follows:

$$r_i^j = MLP(p_i \oplus p_i^j \oplus (p_i - p_i^j) \oplus ||p_i - p_i^j||), \tag{4.14}$$

where $\oplus$ is the concatenation operation, and $||\cdot||$ calculates the Euclidean distance between the neighboring and input points.

Directional feature encoding (DFE)

The directional feature of a point can also be encoded from the spatial locations of points in the local area, which describes the orientation information in the horizontal directions. To encode the directional feature of the input point p_i, we utilize the point-wise local feature descriptor proposed by Jiang et al. [2018], which depicts directional information of eight orientations. Specifically, we implement a selection of the nearest neighbors for point P in each of the eight octants using the Stacked 8-neighborhood Search (S8N). Since distant points provide little information for a description of local patterns, when no point appears within the searching radius in some actant, we duplicate p_0 as the nearest neighbor of itself. We further process features of those neighbors that reside in a $2 \times 2 \times 2$ cube for local pattern description centering at p_0. Many previous works conduct max-pooling on unordered feature vectors to get new features, while we believe that ordered operations such as convolution can better exploit the local geometric structure information. Thus we utilize orientation-encoding convolution, which is a three-stage operator that convolves the $2 \times 2 \times 2$ cube along $x-$, $y-$, and $z-$ axis successively. Formally, the features of neighboring points is a vector V of shape $2 \times 2 \times 2 \times d$, where the first three dimensions correspond

to three axes. Slices of vector M are feature vectors, for example $M_{1,1,1}$ represents the feature from top-frontright octant. The three-stage convolution is formulated as:

$$\begin{aligned} V_x &= g(Conv(Wx, V)) \in R_{1\times2\times2\times d} \\ V_{xy} &= g(Conv(Wy, V_x)) \in R_{1\times1\times2\times d} \\ V_{xyz} &= g(Conv(Wz, V_{xy})) \in R_{1\times1\times1\times d} \end{aligned}, \tag{4.15}$$

where $W_x \in R_{1\times2\times2\times d}$, $W_y \in R_{1\times2\times1\times d}$, and $W_z \in R_{1\times1\times1\times d}$ are weights of convolution operator (bias is omitted for clarity). In this work, we set $g(\cdot) = ReLU(\cdot)$. Finally, we will get a d dimension feature by reshaping $V_{xyz} \in R_{1\times2\times2\times d}$, integrating information from eight spatial orientations and obtains a representation that encodes orientation information.

Elevation feature encoding (EFE) (Optional)

Since the elevation information plays an important role in ALS point cloud classification, elevations should be further emphasized by encoding the elevations provided by the neighboring points to the high dimensional features. In the classification of point clouds of construction site, EFE is not utilized. The elevations of points are encoded to describe the positioning discrepancy in the vertical direction. It should be noted that the elevation encoding module will be removed when classifying MLS or photogrammetric point clouds. To encode elevation information of the input point p_i, the coordinates in z- direction of the local neighboring point set of p_i is extracted and formed a 1-dimensional vector, and then the feature vector is embedded to the same d-dimensional feature using a MPL layer. The MLP layer is formed by a 1 $\times$ convolution layer, a batch normalization layer, and a ReLU activation layer. Thus, the elevation information encoding can be formulated as:

$$\tilde{Z}_i = MLP(Z_i), \tag{4.16}$$

where $Z_i = z^1, z^2, ..., z^K$ is the z coordinates of the input KNN point set P_i and $\tilde{Z}_i$ is the extracted elevation features for the KNN point set.

Attention pooling

Assuming a given point p_i with K neighboring points together with point features $\hat{F}_i = \hat{f}^1, \hat{f}^2, ..., \hat{f}^K$, the attentive pooling is used to aggregate the obtained point features $\hat{F}_i$ of point p_i [Hu et al., 2020]. As shown in Fig. 4.8, the attention scores are achieved by a shared MLP and a followed softmax, which is defined as follows:

$$a_i^j = g(\hat{f}_i^j, W), \tag{4.17}$$

where $j \in 1, ..., K$ is the neighoring point number and W stands for the learnable weights of a shared MLP. The function $g(\cdot)$ is the shared MLP followed by softmax. Then, these features are weighted by the attention scores and summed as follows:

$$\tilde{f}_i = \sum_{j=1}^{K} (\hat{f}_i^j \cdot a_i^j), \tag{4.18}$$

where $\tilde{f}_i$ stands for the output aggregated feature of the input point p_i.

4.3.2 Global relation-aware attention (GRA) module

Inspired by the work in Zhang et al. [2020], we design the GRA module, which learns global attentions from the structural information presented by relations from all the points and all the channels in a global scope and strengthens high-dimensional features using the exploited global attention maps. The GRA module consists of two sub-modules: SRA and CRA.

Spatial relation-aware attention (SRA) module

In order to capture the long-range dependencies between points, the SRA module is applied to 3D points with intermediate deep features. In the former local attention module (i.e., attention pooling module in LoSDA), convolution operations are conducted with small receptive fields on feature maps. However, in order to learn discriminative features for each position, global structural information can be utilized by collecting relations for all the points to enhance the feature representation. Thus, we introduce the SRA module, which consists of three major steps, namely the calculation of relations (named as affinity matrix) between all nodes, the formation of relation-augmented features, and the calculation of attention scores for each position. The details will be explained in the following texts, as illustrated in Fig. 4.9.

Given a feature tensor $\mathbf{X} \in \mathcal{R}^{N \times C}$ presenting a point set with N points and C channels from an immediate layer in the network. We regard the C-dimensional feature vector of each spatial point as a node in the graph. Then, all the feature nodes representing the spatial positions can form a graph G^s with N nodes. The pairwise relation from node n_i to node n_j can be calculated using a dot product, which is termed as affinity:

$$r^s_{i,j} = \alpha^s(x_i)^T \beta^s(x_j), \tag{4.19}$$

where α^s and β^s are two embedding functions implemented by a MLP layer. Then, the bidirectional relations between node n_i and n_j are described by the pairwise affinity value and the relations between all the points can be presented by an affinity matrix $\mathbf{A}^s \in \mathcal{R}^{N \times N}$.

As illustrated in Fig. 4.9, relation features are generated by extracting the column and the row at the spatial position from the affinity matrix. For each node n_i, the relation vectors can be obtained as $r^s_i = [\mathbf{A}^s(i,:), \mathbf{A}^s(:,i)] \in \mathcal{R}^{2N}$. In the figure, we present the relations between node n_7 with other feature node as an example. When calculating the attention scores of the n_i feature node, the original feature vector from the point itself should also be considered to further utilize both the non-local information and its relation to the local original information. Thus, the spatial relation-augmented features can be denoted as:

$$y^s_i = [pool(\psi^s(x_i)), r^s_i], \tag{4.20}$$

in which ψ^s denote embedding functions for the original features. α^s denotes a MLP layer and the embedded features are further fed into a pooling layer after the embedding operation using α^s. Here a max pooling is utilized. $y^s_i \in \mathcal{R}^{2N+1}$ denotes the spatial relation-augmented features for node i and is obtained by connecting the original features with relation feature vector.

The global relation features contain rich contextual information. In order to further mine valuable information from them, we propose to infer attention via a learnable model. Since the dimensions of the spatial relation-augmented features are different from the original feature vector for each point, the spatial relation-augmented features are further embedded using a shared MLP to obtain attention maps for all the point for each feature channel. The spatial attention values can be obtained by:

$$a^s_i = Sigmoid(\gamma^s(y^s_i)). \tag{4.21}$$

where γ^s denotes the embedding functions for the relation-augmented features. A sigmoid function is followed to obtain attentions. $a^s_i \in \mathcal{R}^C$ denote a sequence of attention scores for each feature channel. Then, the attention scores of each nodes can formed a attention matrix $\mathbf{a}_s \in \mathcal{R}^{N \times C}$. Finally, the spatial relation-aware features can be obtained by:

$$\mathbf{Y}^s = \mathbf{a}^s * \mathbf{X}. \tag{4.22}$$

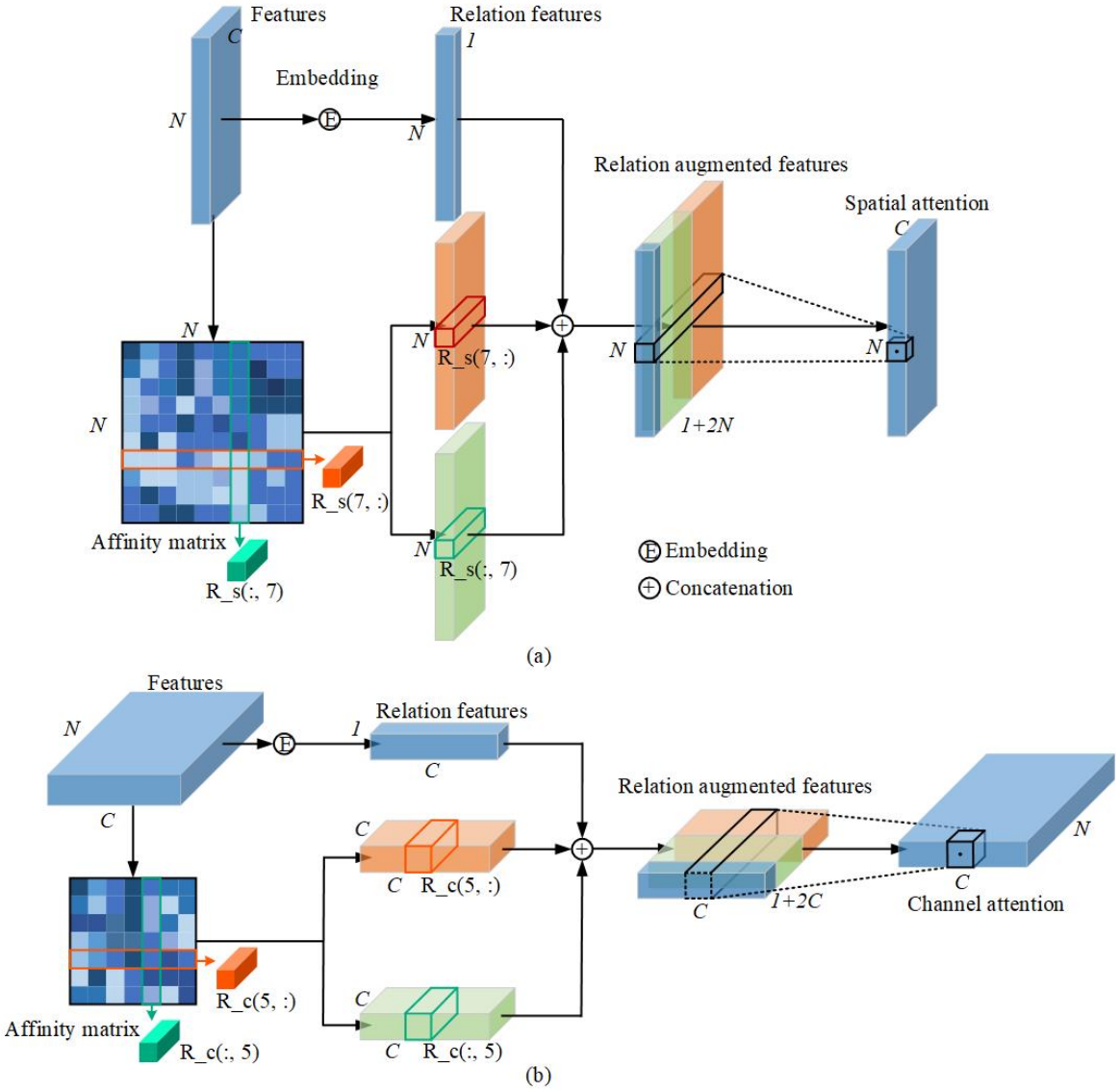

Figure 4.9: Illustration of the GRA module. a) SRA module, b) CRA module.

Channel relation-aware attention (CRA) module

To further consider the interdependencies between channels, as shown in Fig. 4.9, we use a similar strategy and design a channel relation-aware attention module. The main difference is that the relations are calculated between channel-wise nodes instead of spatial positions.

Given an intermediate feature tensor $\mathbf{X} \in \mathcal{R}^{N \times C}$ presenting a point set with N points and C channels. We treat the point set as a graph G^c with C nodes. Thus, the pairwise relation from node c_k to node c_l can be calculated as:

$$r^c_{k,l} = \alpha^c(x_k)^T \beta^c(x_l), \tag{4.23}$$

where α^c and β^c are two embedding functions, which are achieved by a MLP layer. Then, the relations between all nodes can be represented by an affinity matrix $\mathbf{A}^c \in \mathcal{R}^{C \times C}$. Subsequently, for each node c_k, we obtained the relation vector as $r^c_k = [\mathbf{A}^c(k, :), \mathbf{A}^c(:, k)] \in \mathcal{R}^{2C}$. In Fig. 4.9, we use node c_5 as an example for illustration. Then, the original feature vector is embedded

and concatenated with the pairwise relations between all the nodes and generate a new channel relation-augmented feature vector as:

$$y_k^c = [pool(\psi^c(x_k)), r_k^c], \tag{4.24}$$

where ψ^c also denotes the embedding functions implemented by a MLP layer, and the embedding function ψ^c is followed by a pooling layer. The channel attention value can then be obtained using a similar procedure to the procedure conducted on the corresponding spatial module:

$$a_i^c = Sigmoid(\gamma^c(y_k^c)), \tag{4.25}$$

where where γ^c denotes the embedding functions for the relation-augmented features. $a_i^c \in \mathbf{R}^N$ is attention score for each node and forms the attention matrix $\mathbf{a}^c \in \mathcal{R}^{N \times C}$. Finally, the channel relation-aware features can be obtained by multiplying attention scores:

$$\mathbf{Y}^c = \mathbf{a}^c * \mathbf{X}. \tag{4.26}$$

Integration of the GRA module

In order to make full use of the SRA and CRA modules, we further integrate these two modules. In this work, we investigate three patterns for the configuration. In the following text, to make the elaboration clearly and briefly, the configurations of SRA and CRA modules are terms as the GRA module. As illustrated in Fig. 4.10, there are three modes in constructing the GRA module by combining the SRA and CRA modules. Mode 1 is a serial configuration, where the SRA module follows the CRA module. Mode 2 is also a serial configuration, but in this configuration, the CRA module comes after the SRA module. Mode 3 is a parallel pattern in which the intermediate deep features obtained after feeding to the CRA and SRA modules parallelly are concatenated to obtain the final output of the GRA module. The effect of different configurations will be tested in Section 7.2.3.

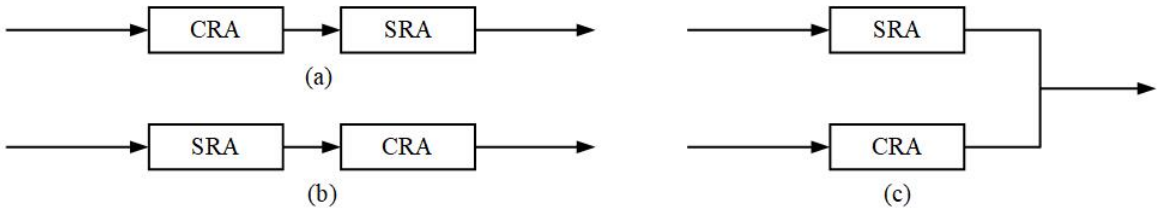

Figure 4.10: Illustration of different configurations of the GRA module. a) Mode 1, b) Mode 2, c) Mode 3.

4.3.3 Details of the network architecture

The multi-scale network framework is inspired by the framework of PointNet++ [Qi et al., 2017b]. Fig. 4.11 illustrates the detailed architecture of our network, which is implemented by a classic encoder-decoder architecture with skip connections. The input point cloud is initially fed to a fully connected layer to extract point features based on the original point features except for point coordinates. Four encoding and decoding layers are then used to learn features for each point. At last, a fully-connected layer is used to predict the semantic label of each point. The input is a large-scale original point cloud with a size of $N \times d$ (the batch dimension is dropped for simplicity), where N is the number of points, d is the feature dimension of each input point. For our testing datasets, each point is represented by its 3D coordinates, intensity, and numbers

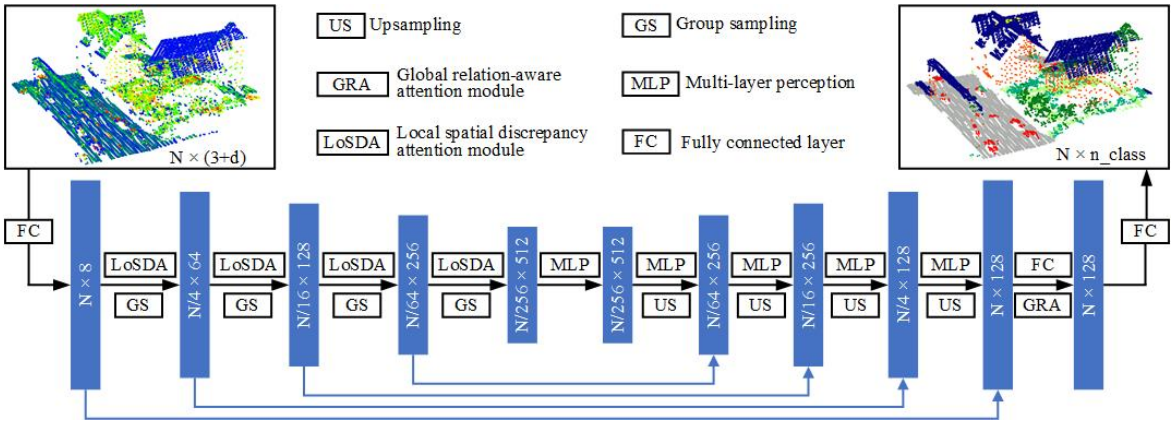

Figure 4.11: Detailed network architecture of GraNet.

of return. The encoder has four layers, progressively reducing the size of the point clouds and increasing the feature dimensions of each point. Each encoding layer consists of the LoSDA module and a grouping and sampling module. The point cloud is downsampled with a four-fold decimation ratio. In particular, only 25% of the point features are retained after each layer. Meanwhile, the feature dimension of each point is gradually increased each layer to preserve more information. The decoder also has four layers that are used after the above encoding layers. For each layer in the decoder, we first use the KNN algorithm to find one nearest neighboring point for each query point. The point feature set is then upsampled through a nearest-neighbor interpolation. Next, the upsampled feature maps are concatenated with the intermediate feature maps produced by encoding layers through skip connections, after which the MLP module is applied to the concatenated feature maps. After all the points have been decoded to the original point size, the GRA module is applied to the feature maps. Then, the semantic label of each point is predicted through a fully-connected layer.

5 Change detection

In this chapter, we present a method of detecting changes of construction sites using photogrammetric point clouds. The overall goal is to detect changes considering both geometric changes and semantic changes. Fig. 5.1 shows the core methods corresponds to the ones we have shown in our research frame in Section 1.3. The entire workflow of our change detection method is presented in Fig. 5.2, with involved algorithms, methods, and illustrations of intermediate results shown. Our change detection method follows a two-step strategy. In the first step, the geometric changes are detected using an occupancy-based change detection method using multi-stereo vision (OBCD-M), which is inspired by previous studies [Hebel et al., 2013]. In the second step, for the detected geometrically consistency occupancy, semantics can be further considered to contribute to detect the semantic consistency and conflicts. Here, a semantic-aided change detection method (SACD) is applied.

Figure 5.1: Proposed methods for change detection that included in this chapter.

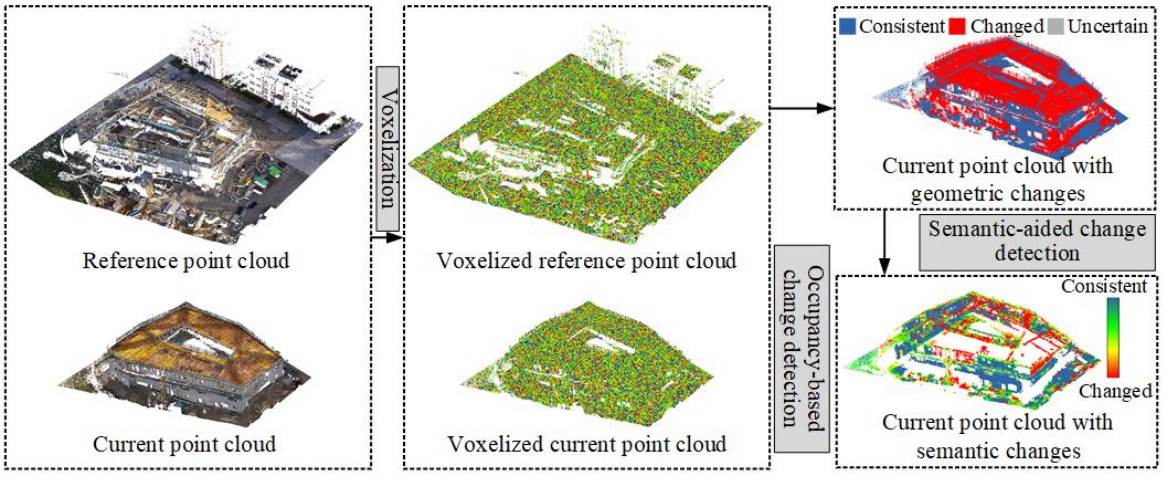

Figure 5.2: Workflow of change detection procedure.

5.1 Occupancy-based change detection using multi-stereo vision (OBCD-M)

The general principle of the OBCD-M method is that changes can be determined by evaluating occupancy conflicts of occupied space and empty space along the viewing rays of cameras considering the sensor positions. By using the occupancy-based change detection, both occlusions and changes can be handled implicitly. Here, photogrammetric point clouds acquired at two different epochs are taken for comparison. We name point clouds acquired at the former epoch as the reference database. The point cloud acquired at the latter epoch as the current database. It should be noted that the raw data, including the camera positions and camera intrinsics, is also required.

To define conflicts between the reference database and the current database, the whole data processing mainly consist of two steps. First, the reference data is assigned to a 3D grid that covers the complete measured area. Second, based on the grid structure, we can evaluate whether the current measurements confirm or contradict previous information in the reference database.

5.1.1 Generation of the reference database

Before evaluating the occupancy conflicts between the reference data and the current measurements, we first assign the reference data to a 3D grid that covers the entire measured area. Here, the grid cells are used not only for searching but also for representing changes, which is different from the process introduced in Section 2.5. The different function of grid cells is resulted from the different data acquisition methods. For photogrammetric point clouds, we can generate point clouds with different densities using different interpolations. 3D points in a small neighborhood would probably be generated from the same camera rays. Here, two types of 3D grid $\mathbf{V}_P$ and $\mathbf{V}_R$ are utilized for storing indices of camera measurements according to positions of 3D points. The same as the procedures in Section 2.5, $\mathbf{V}_P$ presents the index-based rasterization of point clouds, while $\mathbf{V}_R$ denotes the grid cells that are traversed by camera rays. The camera rays are determined by camera positions and the field of view identified by intrinsic parameters. For both types of grid cells, the index of cameras are stored in the cell. In Fig. 5.3, we illustrate how the camera indices are stored in grid cells of $\mathbf{V}_P$ and $\mathbf{V}_R$.

5.1.2 Occupancy modelling of 3D space

Similar to the occupancy model of LiDAR rays introduced in Section 2, the state of occupancy can be represented by a universal set $U = \{empty, occupied\}$. The power set 2^U of U is given as the set $\{\emptyset, \{empty\}, \{occupied\}, \{empty, occupied\}\}$, which contains all possible state. For the occluded spaces, since there is no information acquired from these shadowed area, the occupancy space can be either empty or occupied. Thus, it is actually implicitly modeled as unknown space represented by the set $\{empty, occupied\}$. According to Dempster-Shafer theory (DST), a belief mass within range $[0, 1]$ is assigned to each element of the power set. In addition, the empty set $\emptyset$ has zero mass, and the masses of the remaining set are summed to one:

$$m : 2^U \rightarrow [0, 1], m(\emptyset) = 0, \sum_{A \in 2^U} m(A) = 1. \tag{5.1}$$

An assignment that obeys the aforementioned rules is called as "basic belief assignment". DST makes use of the mass assignment and sets a range for each state of occupancy. The mass of each state, $m(\{empty\})$, $m(\{occupied\})$, and $m(\{unknown\})$ are abbreviated as e, o, and u respectively. As illustrated in Fig. 5.4, the occupancy can be easily modelled as occupied, empty, and unknown. The space that is traversed by at least one ray that in combination with a ray of a

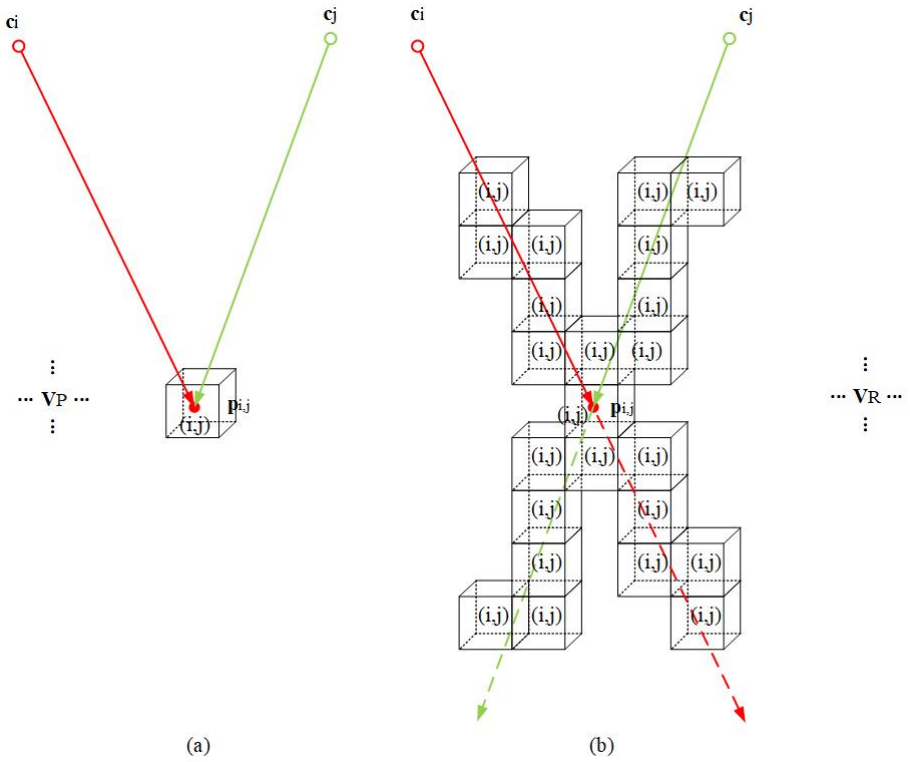

Figure 5.3: Storage of camera indices in 3D grids. a) Rays from two camera positions $\mathbf{c}_i$ and $\mathbf{c}_j$ intersect in position $\mathbf{p}_{i,j}$ located in the 3D grid cell $\mathbf{V}_P$. Corresponding camera indices are stored in the cell, b) rays from camera positions $\mathbf{c}_i$ and $\mathbf{c}_j$ traverse 3D grid cells $\mathbf{V}_R$. The same indices are stored in the traversed grid cells.

second camera led to a 3D point is determined as $e = 1, o = 0, u = 0$. The space that are occupied by 3D points can be defined as occupied space, and the belief masses would be $e = 0, o = 1, u = 0$. For anywhere else, it would be unknown and the belief masses are $e = 0, o = 0, u = 1$.

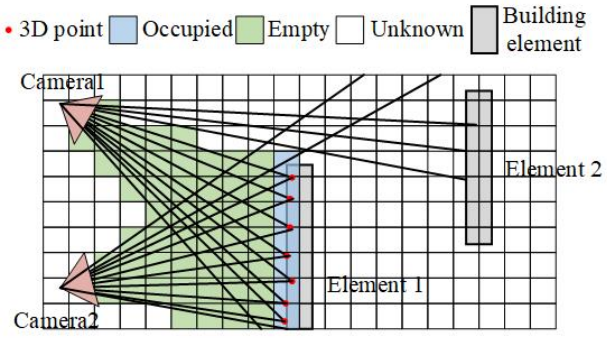

Figure 5.4: Occupancy modelling of stereo vision. We get 3D points on the building element if camera rays intersect on the surface of the element. The corresponding grid cell is observed as occupied. The space along the corresponding camera rays is observed as empty.

In case more than two cameras are used for data acquisition, we need to combine the mass assignment. Assume that k images are acquired from k camera positions and the image sequence is denoted as $\mathbf{I}_m$. For each image pair (I_i, I_j), the belief masses would be $(e_{i,j}, o_{i,j}, u_{i,j})$ deter-

mined following the occupancy modelling of stereo vision. Then, the belief masses under k-image acquisition can be modelled by considering all the possible choices between the image sequence. First, assume that we now have two image pairs (I_{i1}, I_{ji}) and (I_{i2}, I_{j2}) and the corresponding belief masses are $(e_{i1,j1}, o_{i1,j1}, u_{i1,j1})$ and $(e_{i2,j2}, o_{i2,j2}, u_{i2,j2})$. The combination of belief masses would be:

$$\begin{aligned} e &= (e_{i1,j1} \cdot e_{i2,j2} + e_{i1,j1} \cdot u_{i2,j2} + e_{i2,j2} \cdot u_{i1,j1}) \cdot \frac{1}{K} \\ o &= (o_{i1,j1} \cdot o_{i2,j2} + o_{i1,j1} \cdot u_{i2,j2} + o_{i2,j2} \cdot u_{i1,j1}) \cdot \frac{1}{K}. \\ u &= (u_{i1,j1} \cdot u_{i2,j2}) \cdot \frac{1}{K} \end{aligned} \tag{5.2}$$

The operation can be simplified as $m = m_{i1,j1} \oplus m_{i1,j2}$. Thus, the combination of k images can be obtained by:

$$m(\mathbf{I}_k) = \oplus_{\mathbf{I}_l \subset \mathbf{I}_L} m_{\mathbf{I}_l} \tag{5.3}$$

where $\mathbf{I}_L$ denotes the set of image sequence when randomly selecting two images from k images. $\mathbf{I}_l$ is one sample of image sequence in the set. The combination of the occupancy images is commutative and associative. Thus, the order is arbitrary. The corresponding belief masses under k-image acquisition can be written as $(E(\mathbf{I}_k), O(\mathbf{I}_k), U(\mathbf{I}_k))$.

5.1.3 Change detection

After generating the reference database and modelling the occupancy, we can decide whether the new image measurements confirm or contradict the mass assignments based on the former image measurements. Here, we define two types of conflicts: (i) conflict type A (Fig. 5.5a): 3D point $\mathbf{q}$ appear at empty space defined by former measurement; (ii) conflict type B (Fig. 5.5): camera rays traverse occupied space in front of $\mathbf{q}$.

For conflict type A, $\mathbf{v}_q \subset \mathbf{V}_R$ is the grid cell where q is located. Assume that there are camera rays from two image acquisitions, the mass assignment to the grid cell based on the current measurement should be $(E(\mathbf{I}_2), O(\mathbf{I}_2), U(\mathbf{I}_2))$. Let $\mathbf{I_q}$ denote the image indices stored in $\mathbf{v_q}$, then, the joint mass resulting from former measurements should be $(E(\mathbf{I_q}), O(\mathbf{I_q}), U(\mathbf{I_q}))$. Thus, a measure of conflict can be determined as $C_\mathbf{q} = E(O(\mathbf{I}_2)\mathbf{I_q})$.

Then, we address the other type of conflict. For each grid cell $\mathbf{v_q} \subset \mathbf{V}_P$ that the camera rays of new measurements traverse, we retrieve the image indices of former measurements as $\mathbf{I_p}$. The current measurements are denoted as $\mathbf{I_q}$. Thus, a measure of conflict can be calculated as $C_\mathbf{q} = O(\mathbf{I_p})E(\mathbf{I_q})$. To summarize, conflicts always occur when the space is occupied at one time epoch but empty at the other one. Thus, the belief mass which denotes whether the space changes or nor can be determined as:

$$\begin{aligned} Conflicting &= E(\mathbf{I_q}) \cdot O(\mathbf{I_p}) + O(\mathbf{I_q}) \cdot E(\mathbf{I_p}) \\ Consistent &= E(\mathbf{I_q}) \cdot E(\mathbf{I_p}) + O(\mathbf{I_q}) \cdot O(\mathbf{I_p}) \\ Unknown &= U(\mathbf{I_q}) \cdot (E(\mathbf{I_q}) + O(\mathbf{I_p})) + U(\mathbf{I_p}) \cdot (E(\mathbf{I_q}) + O(\mathbf{I_q})) + U(\mathbf{I_q}) \cdot U(\mathbf{I_p}) \end{aligned} , \tag{5.4}$$

where $\mathbf{I_q}$ and $\mathbf{I_p}$ denote the image acquisitions of the current and former measurements, respectively.

In the final results, the occupancy can be classified into three different folds, geometrically conflicting, geometrically consistent, and geometrically unknown. The geometrically conflicting parts can be defined as geometric changes.

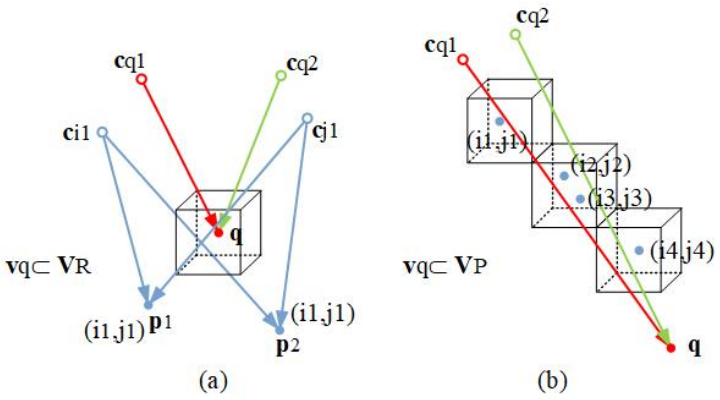

Figure 5.5: Conflicts between reference data and current measurements under image acquisition. a) Conflict type A: 3D point $\mathbf{q}$ with the intersection of rays from $\mathbf{c_{q1}}$ and $\mathbf{c_{q2}}$ appear at empty space defined by former measurements with cameras $\mathbf{c}_{i1}$ and $\mathbf{c}_{i2}$, b) Conflict type B: camera rays traverse occupied space in front of $\mathbf{q}$ obtained based on image acquisition from camera $\mathbf{c_{q1}}$ and $\mathbf{c_{q2}}$.

5.2 Semantics-aided change detection (SACD)

However, geometric changes are not sufficient to present changes of the observed scene. Apart from geometric changes which can be presented by the changes of occupancy states, semantic changes also widely appear in the construction progress and are vital for monitoring the construction process. As illustrated in Fig. 5.6, the images shows that there is no building roof in Fig. 5.6a but the formwork of the building roof has been finished in Fig. 5.6b. The illustration of corresponding point clouds is shown in Fig. 5.6c-d. The geometric changes on the top of the building is clearly illustrated. The corresponding point cloud is shown in Fig. 5.6g-h. For Fig. 5.6e, a formwork appear in the scene, while the formwork is replaced by a almost finished building wall in Fig. 5.6f. Although the detailed geometric change between formwork and as-built building structure can be determined with high-precision data, we detect this type of changes as semantic changes due to the limitation of data quality and parameter settings, i.e., the grid sizes. Generally, geometric changes appears when the occupancy of space changes, while the semantic changes occur when the semantic categories of occupied space changes.

To detect semantic changes, semantic information of gird cells should be considered. In our method, semantic information can be obtained using the semantic segmentation method presented in Section 4. Here, GraNet are utilized to obtain semantic labels for each points. The labels are presented in a soft way, namely soft labels without directly indicating the exact class of objects. It means that for each grid cell we can obtain a sequence of possibilities which denote the possibilities that the grid cell belongs to each category. Based on the results of geometric changes in the former section, the semantics can be further considered. It should be noted that there is no need to compare semantics when the grid cell has geometric changes. Thus, before considering the semantic information, grid cells with uncertainties and geometrically changed occupancy status are filtered out. Only grid cells which are geometrically consistent will be considered for further semantic comparison.

Since the results of the former semantic segmentation of construction sites are presented with soft labels, the estimation of confirming and conflicting is also achieved based on the possibilities of each categories. If the category of the grid cell remains unchanged, the grid cell will be defined as semantically consistent. Otherwise, although the grid cell is geometrically consistent,

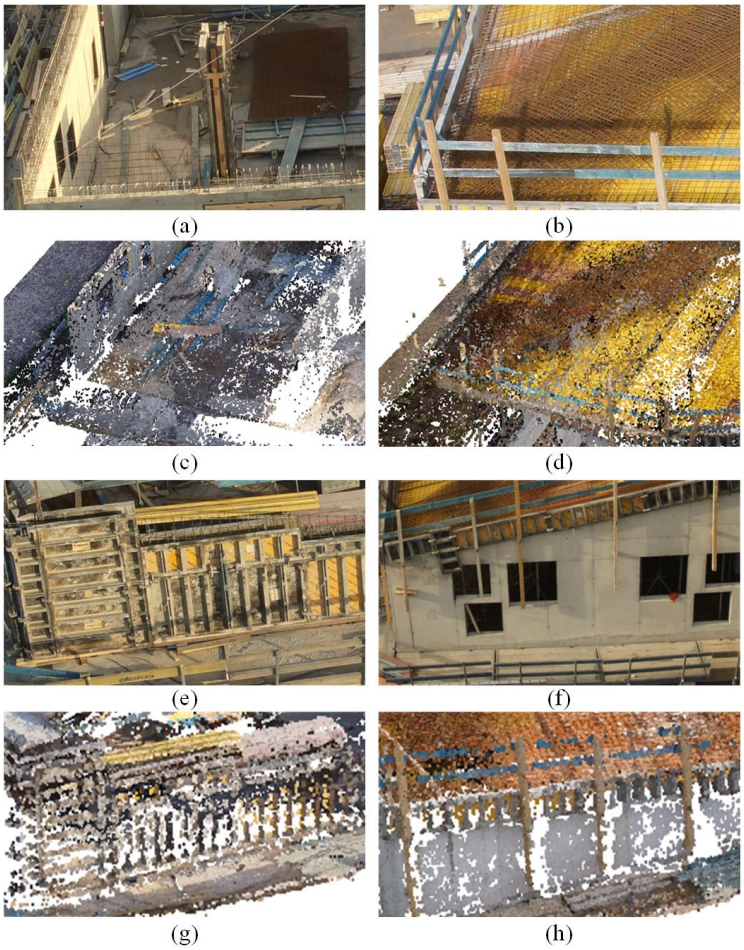

Figure 5.6: Illustration of different change types. a) Scene section before placing the formwork, b) scene section corresponding to 5.6a with formwork and iron for a concrete building floor, c) point cloud corresponding to 5.6a, d) point cloud corresponding to 5.6b, e) scene section showing the formwork for a building wall, f) scene section showing the building wall corresponding to 5.6e, g) point cloud corresponding to 5.6e, h) point cloud corresponding to 5.6f.

its semantics have changed and this will also be annotated as changes in the final change detection result.

$$\begin{aligned} Consistent &= \sum_k p_i^k \cdot p_j^k \\ Conflicting &= 1 - \sum_k p_i^k \cdot p_j^k \end{aligned}, \tag{5.5}$$

where $p_i^k \in p_i^1, p_i^2, ..., p_i^K$ and $p_j^k \in p_j^1, p_j^2, ..., p_j^K$. K is the number of categories. By considering the possibilities of semantic changes, the whole change detection process is finished, with four types of change status: geometrically changed, geometrically and semantically consistent, geometrically consistent but semantically changed, and uncertain.

6 Experiments

In this chapter, we will introduce the experiment design, the experiments, including datasets and preprocessing, and the evaluation metrics for the analysis of experimental results.

6.1 Experiment design

For testing the performance of the methods introduced in Section 3-5, various experiments were conducted. These experiments can be divided into three major groups.

The first group is the test of the registration methods using several benchmark datasets, including three published TLS benchmark datasets, including the Bremen dataset, the large-scale TLS point clouds registration benchmark (WHU-TLS) dataset [Dong et al., 2020], and the Real-world Scans with Small Overlap (RESSO) dataset [Chen et al., 2019]. The performance of PBPC and GRPC were compared with several baseline methods. Since GRPC is a 3D improved solution for PBPC. Thus, we mainly conducted sensitivity analysis on GRPC, including parameters involved in the method and some attributes of datasets, such as noise levels and overlapping ratios between registration pairs. The experiments were implemented using Matlab R2017b and conducted on a computer with an Intel i7-4710MQ CPU and 16GB RAM.

The second group is the test of the performance of the semantic segmentation methods. Since the semantic segmentation methods were involved in different practical applications, these proposed methods were tested using different benchmark datasets. The MLCE method was tested on an ALS dataset (i.e., the DFC2018 dataset). The DPE method was tested on two ALS benchmark datasets, including the ISPRS benchmark dataset [Cramer, 2010; Rottensteiner et al., 2012], the ALS dataset of a selected area provided by Actueel Hoogtebestand Nederland (AHN)* [Xu et al., 2014; Vosselman et al., 2017]. The GraNet method was tested on three benchmark datasets, including the ISPRS benchmark dataset, the Large-scale Aerial LiDAR Dataset for Semantic Labeling in Dense Urban Areas (LASDU) [Ye et al., 2020], and the Dayton Annotated LiDAR Earth Scan (DALES) data set [Varney et al., 2020]. Since in the semantic segmentation methods, both non deep learning methods and deep learning methods were involved, the experiment environments contained two parts. The non deep learning algorithms were tested using Matlab R2017b and implemented on a computer with an Intel i7-4710MQ CPU and 16GB RAM. The deep learning methods were implemented in the framework of Tensorflow and performed on an NVIDIA TITAN X (Pascal) 12 GB GPU.

The third group is the test of the whole change detection workflow on a construction dataset, which is a real practical application for monitoring the construction process. The construction dataset was registered using the proposed registration method (i.e., GRPC). Then, the construction dataset was using the semantic segmentation method (i.e., GraNet). Finally, based on the former registration and semantic segmentation results, the changes between datasets of different

*https://downloads.pdok.nl/ahn3-downloadpage/

acquisition dates were conducted using the proposed change detection method. The experiments about change detection was conducted using Matlab R2017b and on a computer with an Intel i7-4710MQ CPU and 16GB RAM.

The details about experimental datasets and the prepossessing of the datasets will be provided in the following sections.

6.2 Experiments

We have designed three groups of experiments. The first one is for the test of the registration methods. The second one is for the test of the semantic segmentation methods. The third one is for the test of the whole change detection procedure on a construction datasets, including registration, semantic segmentation, and change detection.

6.2.1 Experiments for registration

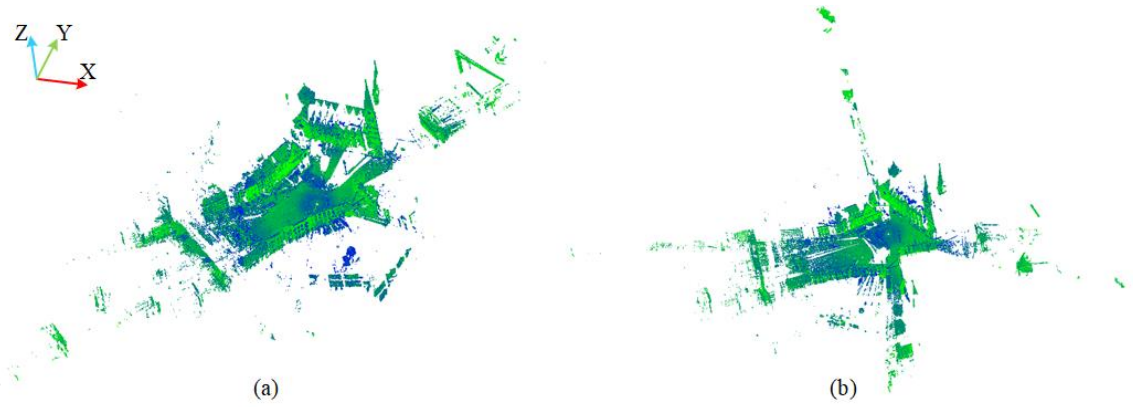

Figure 6.1: The Bremen TLS dataset. a) Target and b) source point clouds color-coded with intensities.

In order to test the versatility of the registration methods, three benchmark datasets with different point densities and different characteristics of scenes were utilized for testing the performance. In this section, we will introduce the test datasets. For evaluating the performance of PBPC and GRPC, experiments were conducted using three benchmark datasets, and their results were evaluated and analyzed.

The Bremen dataset

The Bremen dataset is a pair of TLS point clouds from the ThermalMapper project acquired by the Jacobs University Bremen covering a large urban area (see Fig. 6.1). Table 6.1 shows the detailed information of the Bremen dataset, including the area size of the observed scene, the number of points, and the overlap ratios between scans. It should be noted that the target point cloud serves as a reference and the source point cloud is the one to be registered.

The WHU-TLS dataset

The WHU-TLS dataset is published by Wuhan University, which provides multiview point clouds with varying point densities acquired from different scenes. We selected three representative scenes from the WHU-TLS dataset: a subway station, a park, and a cliff of a mountain. As shown by Fig. 6.2, point clouds acquired from these three different scenes show different geometric characteristics, which provide us valuable opportunities to test the strength and weakness of the

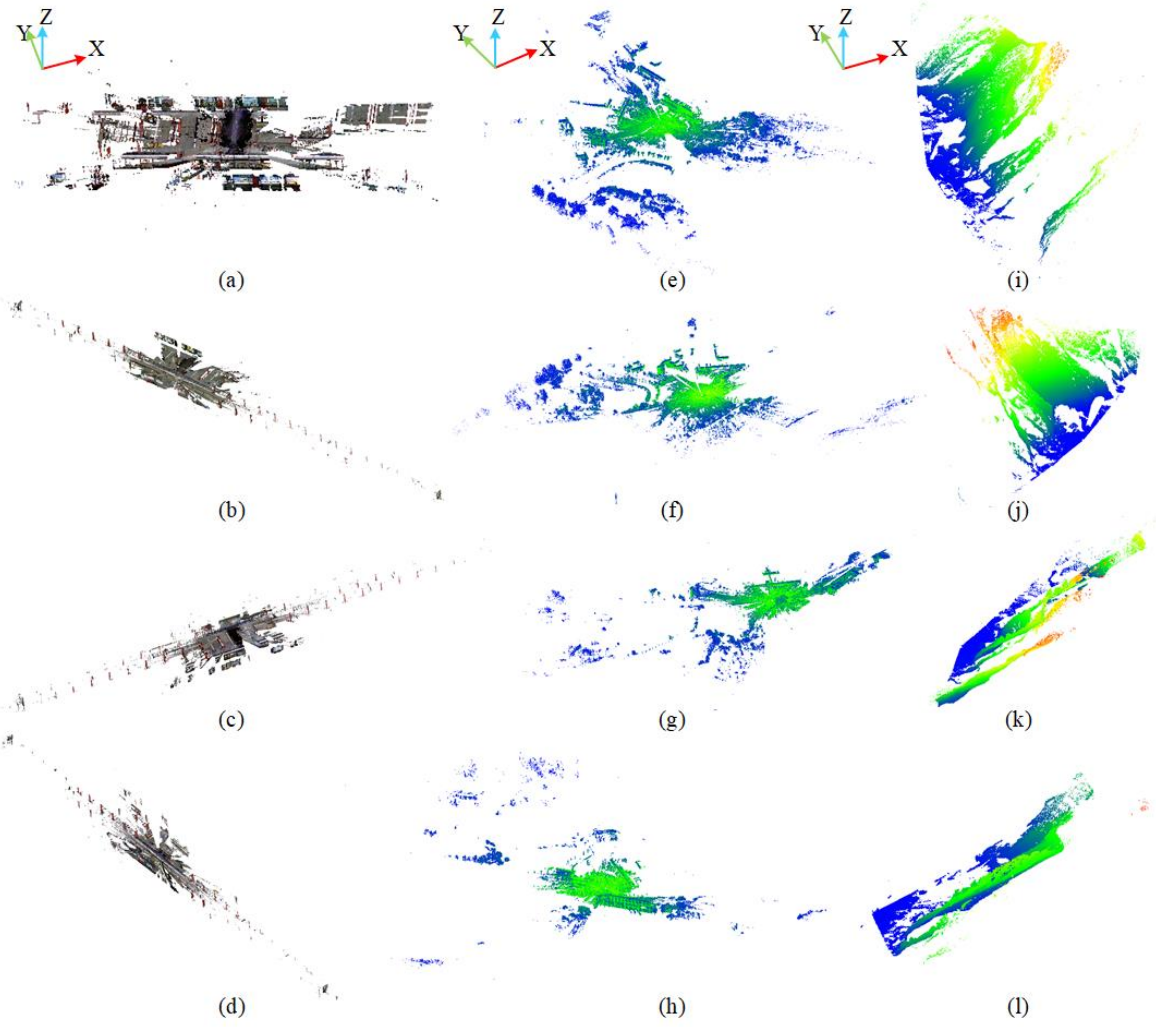

Figure 6.2: The WHU-TLS dataset. a) - d) the selected point clouds of the subway station dataset textured with RGB color, e) - f) the selected point clouds of the park dataset color-coded with intensities, i) - l) the selected point clouds of the mountain dataset color-coded with heights.

proposed method. Detailed information for the selected point clouds is listed in Table 6.2. For this multiview dataset, the reference scan for each registration pair is also listed in Table 6.2.

The RESSO dataset

The last one is the RESSO dataset (see Fig. 6.3). For this dataset, we used six TLS point clouds, which generated five registration test pairs. For each registration pair, Scan 2 is regarded as the reference scan. The detailed information of these scans is listed in Table 6.3. As shown by the table, the five pairs have different overlap ratios. By utilizing this dataset, the influence of different overlap ratios can be tested on GRPC.

Table 6.1: Information of the Bremen TLS dataset.

Parameters	Target	Source
Area (m^2)	451 ×587	585 ×422
Number of points (million points)	15.2	15
Approxi. overlap ratio	0.85	

Table 6.2: Information of the WHU-TLS dataset.

Scene	Scan index	Area (m^2)	Number of points (million points)	Approxi. overlap ratio	
				Reference scan index	ratio
Subway	5	73 ×60	41.7	/	/
	3	111 ×161	39.0	5	0.96
	4	203 ×35	39.1	5	0.92
	6	67 ×179	40.5	5	0.86
Park	14	461 ×564	3.9	/	/
	13	600 ×638	3.8	14	0.55
	15	526 ×552	4.9	14	0.47
	16	434 ×534	4.8	15	0.67
Mountain	4	255 ×277	3.4	/	/
	2	130 ×331	3.7	3	0.54
	3	122 ×346	3.5	4	0.50
	5	209 ×329	2.7	4	0.69

Table 6.3: Information of the RESSO TLS dataset.

Parameters	Target	Source (a)	Source (b)	Source (c)	Source (d)	Source (e)
Scan number	2	1	3	5	6	7
Area (m^2)	275 ×280	195 ×273	256 ×258	231 ×192	218 ×183	177 ×260
Number of points (million points)	0.82	0.45	0.62	0.78	0.60	0.22
Approxi. overlap ratio	/	0.52	0.60	0.48	0.39	0.24

6.2.2 Datasets for semantic segmentation

As mentioned in Section 6.1, the semantic segmentation methods were tested using five benchmark datasets.

The DFC2018 dataset

The DFC2018 dataset is a classification-related benchmark dataset acquired by the National Center for Airborne Laser Mapping using an Optech Titan MW (14SEN/CON340) with an integrated camera (a LiDAR sensor operating at three different laser wavelengths, namely 1550 nm, 1064 nm, and 532 nm), including 20 land use and land cover categories. Here, experiments are carried out using LiDAR point clouds collected by laser wavelength 1550 nm. The dataset is illustrated in Fig. 6.4. For the dataset, we adopted a random sampling strategy to select training and test samples. We randomly select 500 samples for each class as training samples and the rest of samples with labels is assigned as test samples. Labels are selected as training, and test samples can be seen in Fig. 6.4.

The ISPRS benchmark dataset

The ISPRS benchmark dataset was acquired in August 2008 by an airborne Leica ALS50 system with an average flying height of 500 m and a 45° field of view. The point density is approximately 4 points/m^2, and there are approximately 753,000 points for training 412,000 for testing. In this dataset, all LiDAR points contain the $x-$, $y-$, $z-$coordinates in Euclidean space, intensity values, and the number of returns. These points have been manually assigned to the following nine classes of different objects: powerline, low vegetation, impervious surfaces, cars, fences/hedges, roofs, facades, shrubs, and trees. The test areas are located in the city center of Vaihingen, Germany, where buildings are present in a dense and complex pattern with an area of 389 m ×

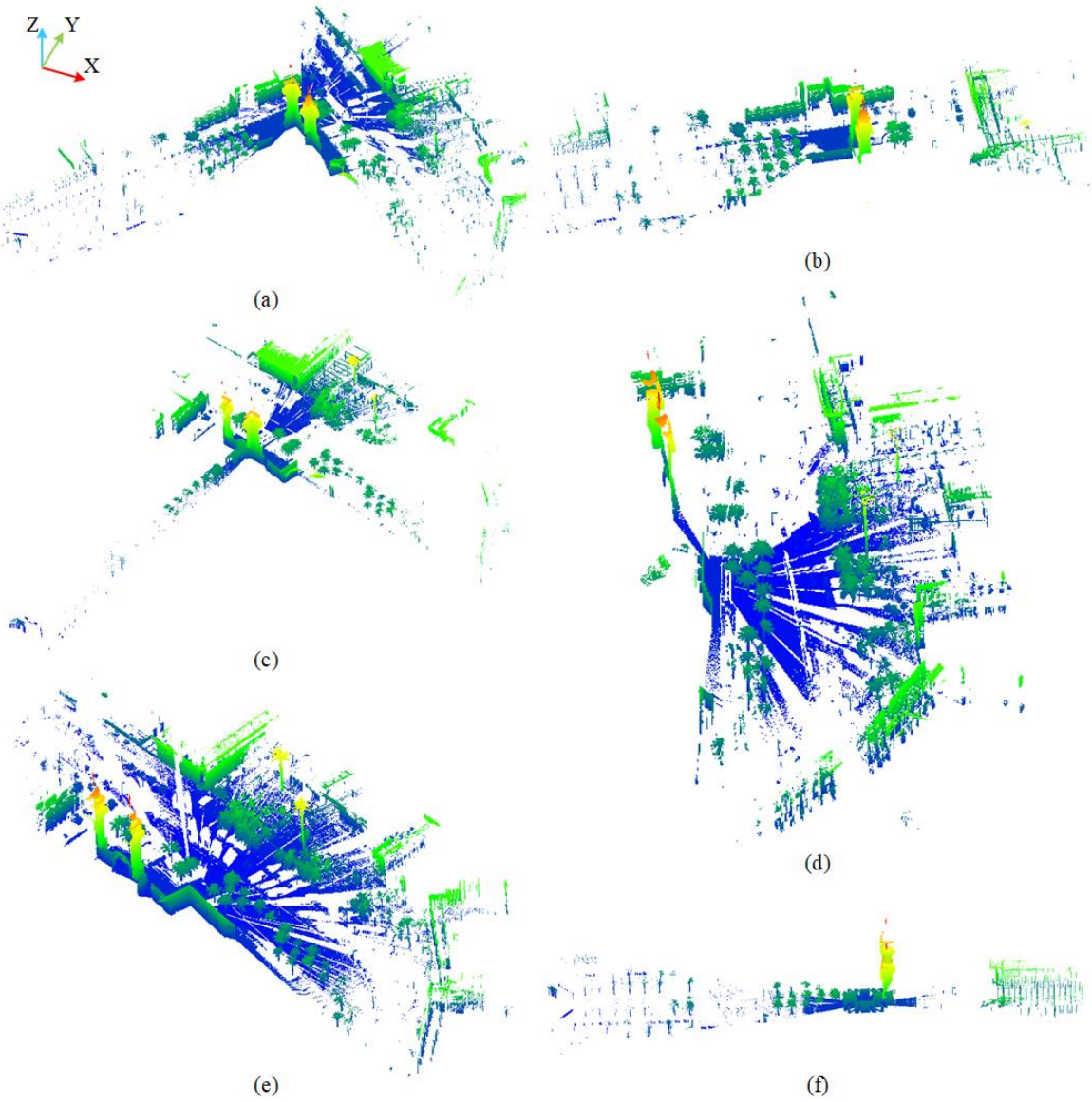

Figure 6.3: The selected scans from the RESSO TLS dataset. a) Target point cloud, b) - f) source point clouds to be registered. All point clouds are color-coded with heights.

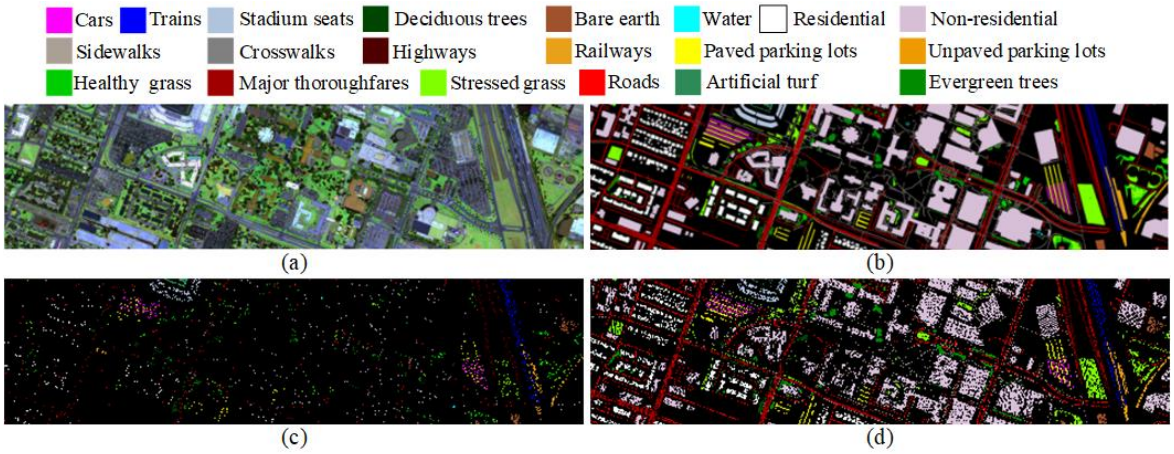

Figure 6.4: Top view of the DFC2018 dataset. a) Color composite of MS-LiDAR intensity data, b) ground truth, c) training samples, d) test samples. The map of training samples is dilated for better illustration. And the black points in classification maps are pixels without projected LiDAR points.

419 m. Moreover, the training area contains mainly residential houses and high-rise buildings covering an area of 399 m × 421 m. The test and the training areas are illustrated in Fig. 6.5.

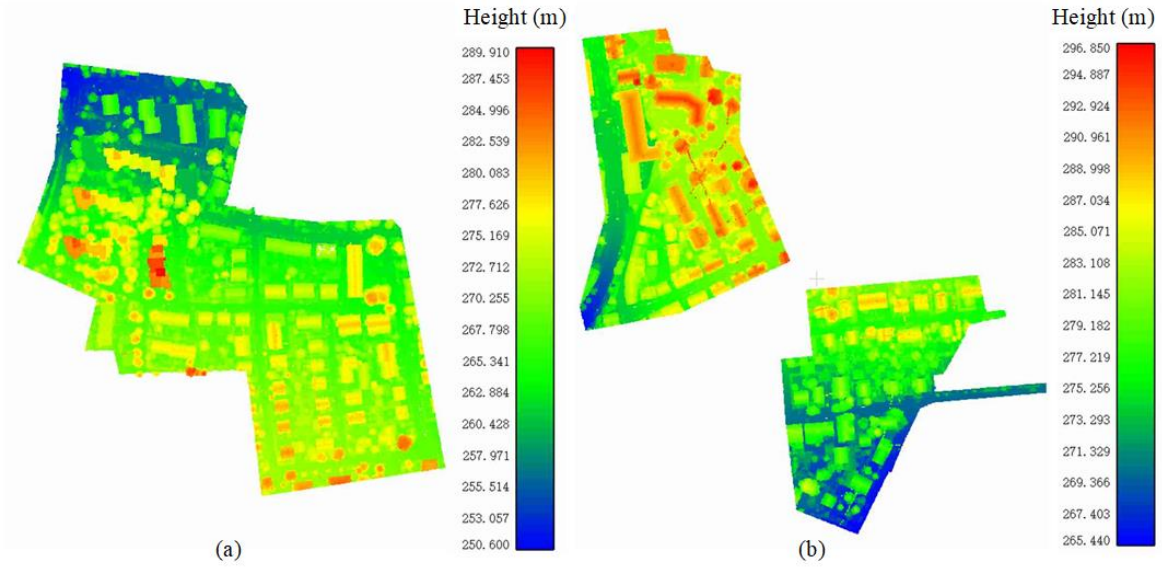

Figure 6.5: Top view of the ISPRS benchmark dataset colored by elevations. a) Training area, b) test area. It should be noted that the scales of two color bars are different.

In order to conduct the training process, we divided the dataset into several point blocks as illustrated in Fig. 6.6, similar to the strategy used in Li et al. [2020a]. Considering the covering area of the dataset, the target size for each block in this step was set to be 100 m × 100 m. The training dataset was then divided into 13 non-overlapped blocks, which consisted of two blocks for validation and 11 blocks for training. Among these blocks, block 13 and block 4 were selected for validation considering the balance of different categories in the validation data. Each block was further divided into 25 m × 25 m blocks with a 12.5 m overlap in both $x-$ and $y-$directions. Besides, the test dataset was also processed using the same procedure for

the training and validation datasets. After the two-step subdivision, the uneven densities of the dataset and each block hold points whose data amount varies from hundreds to ten thousand points. To meet the requirement for the input for the networks, we randomly sampled points for each block to generate sub-pointsets for training and test processes. A fixed value of 4096 was chosen as the size for each sub-pointset. Moreover, each point within the pointset was represented by a 5D vector considering the provided information that comprises the $x-$, $y-$, $z-$coordinates, intensities, and return numbers.

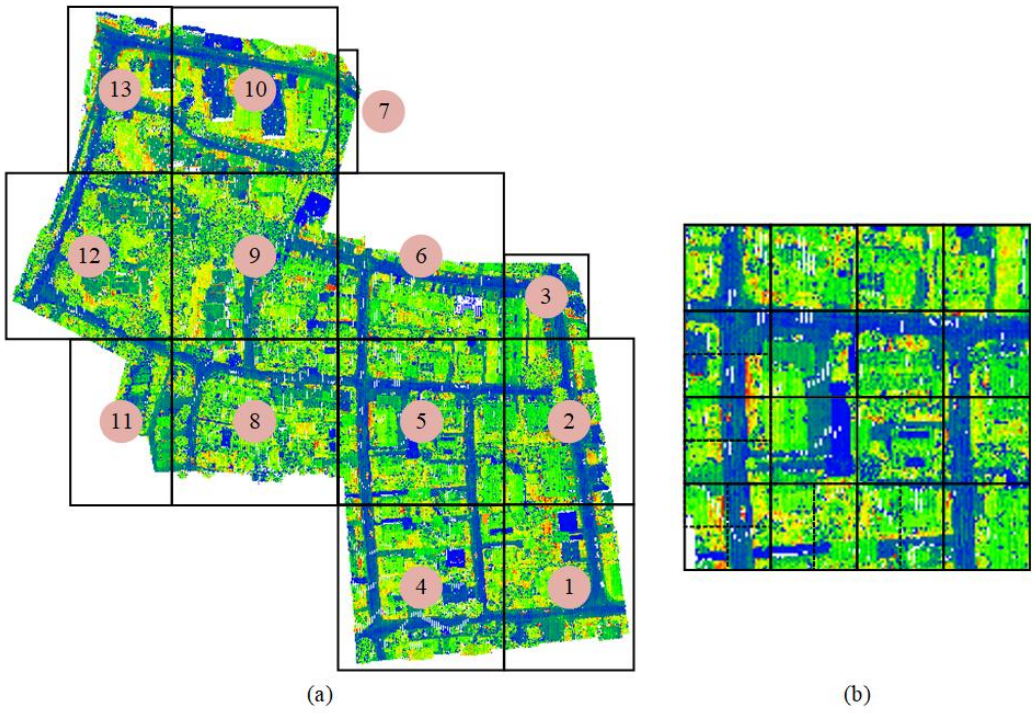

Figure 6.6: Illustration of the data division of the ISPRS benchmark dataset. a) The division of the large training blocks, b) the further subdivision of each training block.

In DPE and GraNet, when constructing a local neighborhood for each point, the number of neighboring points was all set as 32, the same as the hyperparameter in PointNet++. In the training process, we used an Adam optimizer with an initial learning rate of 0.001, a momentum value of 0.9, and a batch size of 4. The learning rate was iteratively reduced based on the current epoch by the factor of 0.7. Moreover, the training process lasted for 1000 epochs totally, and the weights were saved if the loss decreased.

The AHN3 ALS dataset

AHN is a digital height map with detailed and precise altitude data for the entire Netherlands; it additionally provides a high resolution and precise LiDAR database. The AHN data are mainly used for water management. There are multiple versions of AHN, and the most recent and detailed one is AHN3. In AHN3, each point has x-, y-, and z- values, and it is classified into the ground surface, water, buildings, artificial objects, and unclassified (including vegetation). Moreover, each point contains scanning information similar to that of the Vaihingen dataset such as the number of returns, intensity, and GPS time. The experimental dataset is a part of the AHN3 dataset, which covers an area of 2km $\times$ 2km in the city center of Rotterdam, The Netherlands,

as illustrated in Fig. 6.7. The data were acquired with the FLI-MAP laser scanning in April 2010. The point density is approximately 10 points/m^2. The number of points in the training area is approximately 19,008,000, and that in the two test areas is approximately 9,860,000 and 7,828,300, respectively.

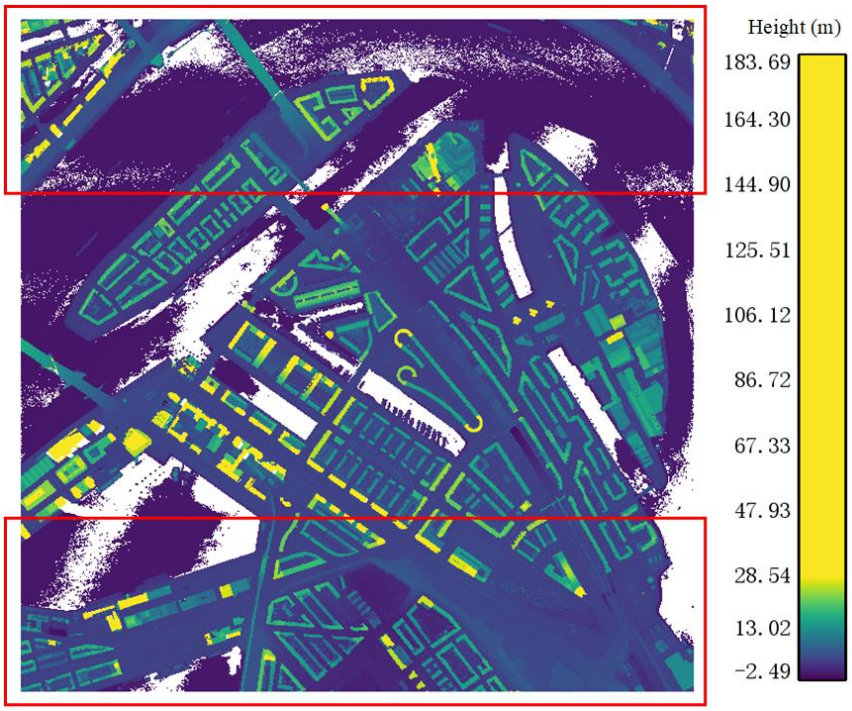

Figure 6.7: Top view of the selected area in the AHN3 dataset (color coded with heights). Area with available reference labels is framed. (Red box: test areas, Other: training area).

The LASDU dataset

The LASDU dataset is a part of data obtained in the campaigns from HiWATER (Heihe Watershed Allied Telemetry Experimental Research) project [Li et al., 2013]. The study area is located in the valley along the Heihe River in the northwest of China. The covering study area is nearly flat, and the average elevation is about 1550 m. The dataset was acquired in July 2012 by a Leica ALS70 system onboard an aircraft with a flying height of about 1200 m. The average point density was approximately 3 points/m^2, and the vertical accuracy ranges between 5-30 cm. One part of the dataset was manually labeled, covering an urban area of around 1 km^2, with highly dense residential and industrial buildings. The total number of annotated points is approximately 3.12 million. In the dataset, five categories of urban objects were considered: ground (e.g., artificial ground, roads, and bare land), buildings, trees (e.g., tall and low trees), low vegetation (e.g., bushes, grass, and flower beds), artifacts (e.g., walls, fences, light poles, vehicles, and other artificial objects). The entire labeled point cloud of the investigating area has been divided into four areas. The numbers of points in these four areas are around 0.77 million, 0.59 million, 1.13 million, and 0.62 million, respectively. In Fig. 6.8, the training and testing datasets are shown, with the elevation of the study area given. From Fig. 6.8, we can find that the annotated area is nearly flat, and the maximum difference of elevation is only around 70 m.

Considering that the LASDU dataset has a similar point density to the ISPRS benchmark dataset, we conducted similar division procedures on the LASDU dataset to generate the data

for the training and validation process. The whole training data was split into several 100 × 100 large blocks without overlaps. Then, 90% of blocks were used for training, and the other 10% blocks were treated as validation data. All these blocks were subdivided into 25 × 25 subblocks with 12.5-m overlaps in both $x-$ and $y-$directions. Finally, before putting all these small blocks into the network for the training process, the points were resampled from each point set to meet the input requirement. The number of points as the input size was set as 4096. Additionally, the hyperparameters set for the training process was also similar. The batch size, the initial learning rate, the decay rate, and the max epoch were set to 4, 0.001, 0.7, and 1000, respectively.

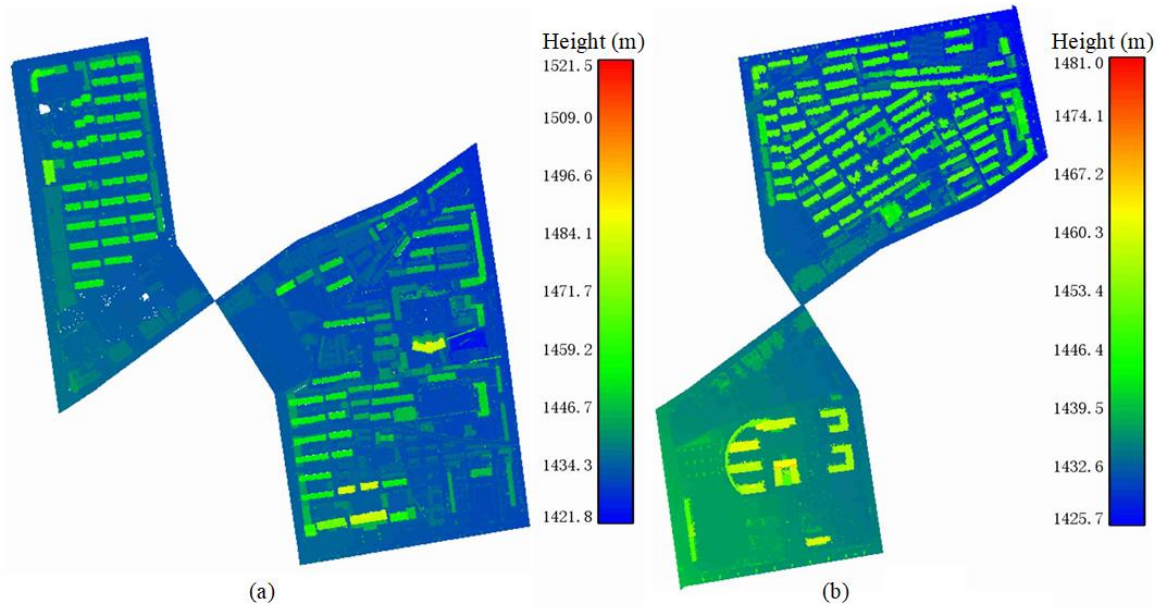

Figure 6.8: Top view of the LASDU dataset colored by elevations. a) Training area, b) test area. It should be noted that the scales of two color bars are different.

The DALES dataset

The DALES dataset is an ALS benchmark dataset with over a half-billion hand-labeled points spanning 10 km^2 of area. DALES is the most extensive publicly available ALS dataset with over 400 times the number of points and six times the resolution of other currently available annotated aerial point cloud data sets. The average point density was about 50 points/m^2. In the dataset, eight categories of objects are considered: ground, vegetation, cars, trucks, powerlines, poles, fences, and buildings. The dataset contains 40 tiles in total, with 29 tiles for training and 11 for testing. Each tile covers an area of 0.5 km^2 with about ten million points. This data set gives a critical number of expert verified hand-labeled points for evaluating new 3D deep learning algorithms, helping to expand the focus of current algorithms to aerial data. In Fig. 6.9, since the dataset is enormous and it is impossible to show the whole dataset in this paper, two examples from the test dataset are illustrated.

The DALES dataset is a much larger dataset compared with the former two datasets. Thus, different division procedures are conducted on the DALES dataset. First, three tiles were selected from the training tiles as validation data. Then, all tiles were subdivided into 20 m × 20 m subblocks without overlaps. In order to fulfill the input requirement of the network, the points of each subblock were resampled to a point set with a size of 8092 points. The batch size was set

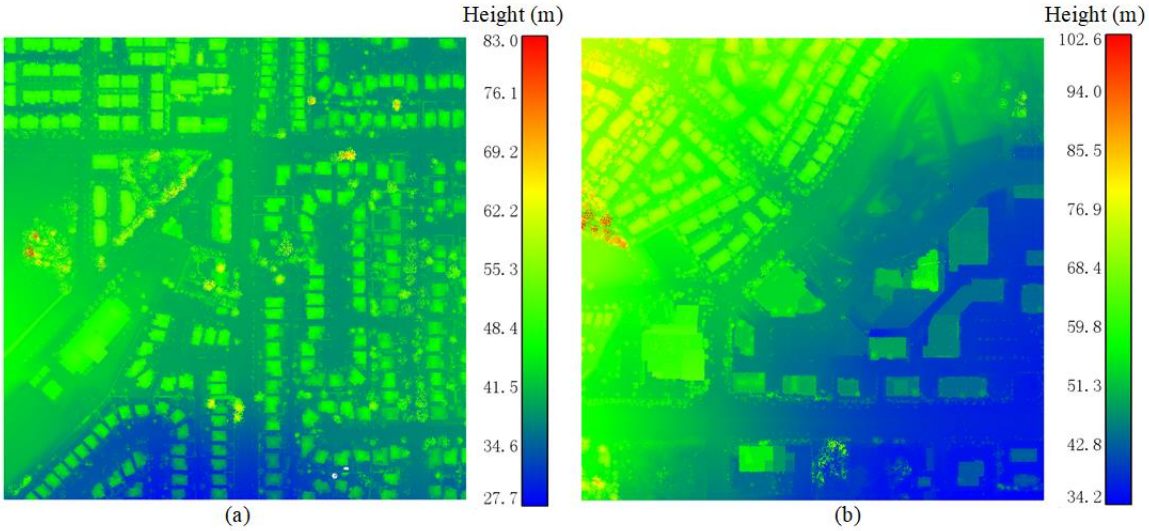

Figure 6.9: Top view of the DALES dataset colored by elevations. a) - b) Two selected tiles from the testing dataset. It should be noted that the scales of two color bars are different.

to 8. The initial learning rate was set as 0.0005. The decay rate and the max epoch were set to 0.7, and 1000, respectively.

6.2.3 Dataset for change detection

The performance of the change detection method is tested using one photogrammetric point cloud dataset acquired at a construction site.

The construction site is the Haus für Kinder (HFK), located in Moosach, Munich, Germany, which is a peripheral construction site with neighboring houses on one side and grassland on the other. The images are acquired via a UAV platform with the nadir and oblique view directions. The focal length of the camera is 18 mm and the image size is 6000 $\times$ 4000 pixels. Table 6.4 shows the details for the HFK dataset, including acquisition date, area of the scene, number of points, as well as approximate overlap ratios. In Fig. 6.10, positions and configuration of images for the HFK dataset, as well as the generated point clouds, are illustrated. It should be mentioned that the overlap ratio is calculated based on the reference point cloud which is acquired on the former acquisition date for each dataset. To obtain the semantics of the acquired point clouds, the dataset was manually labeled to 17 categories, including cranes, structures under construction (strut_const), wooden piles (wood_piles), metal pipe piles (metal_piles), concrete blocks (concrete), waster baskets (waste), others, pipe or sticks (pipes), planes or battens (planes), scaffolds or ladders (scaffolds), formworks, containers or boxes (containers), sheds or tens (sheds), buildings, naturals, impervious layer (impervious), bare lands. In the semantic segmentation of the construction dataset, the point clouds acquired on Oct 20, Nov 20, and Dec 12, 2014 were used for training and the other two point clouds are used for testing. As for the change detection of point clouds, we evaluate our method on two pairs. The first comparison was conducted between point clouds acquired on Dec 12, 2014 and Jan 16, 2015. The second one was conducted on point clouds acquired on Jan 16 and Feb 26, 2015.

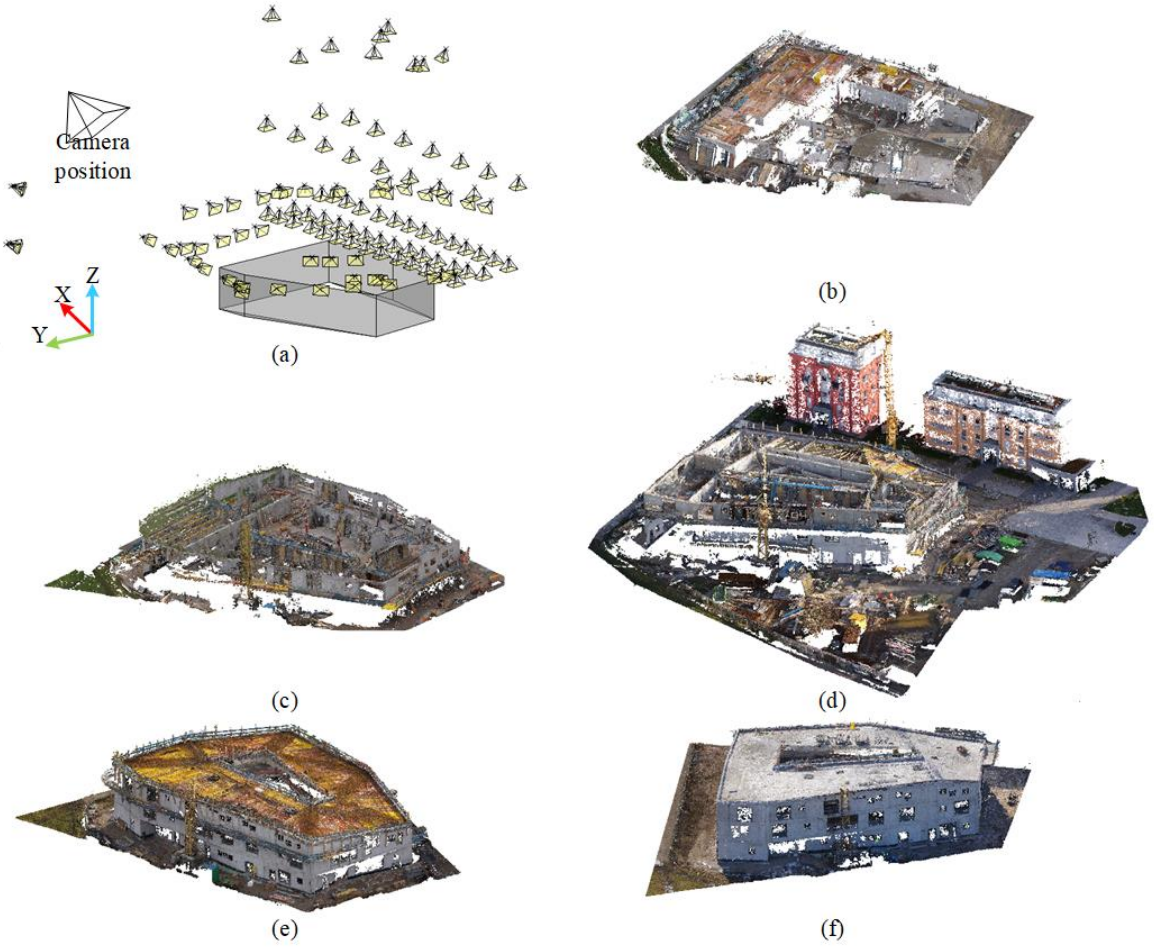

Figure 6.10: Positions and configurations of cameras taken in the construction site HFK. a) Positions of cameras in HFK on Jan 16, 2015, b) - f) point clouds of HFK acquired on Oct 20, Nov 20 and Dec 12, 2014, and Jan 16, and Feb 26, 2015, respectively.

Table 6.4: Information of the construction dataset.

Acquisition date	Size (m^2)	Number of points (Million points)	Approxi. overlap ratio
2014/10/20	61 ×56	10.2	/
2014/11/20	62×63	25.6	0.09
2014/12/12	105×87	30.1	0.08
2015/01/16	61×56	24.2	0.13
2015/02/26	62×77	34.1	0.31

6.3 Evaluation metric

6.3.1 Evaluation metric of registration

The performance of the registration methods was evaluated from two different aspects. The first evaluation criterion is registration accuracy. First, ground truth is needed for the evaluation of registration accuracy. For the Bremen dataset, we manually aligned source and target point clouds, followed by an ICP refinement as ground truth. As a registration benchmark, the WHU-TLS dataset and the RESSO dataset provided the accurately aligned source and target scans as ground truth. Then, the matching was performed between the source and target point clouds.

The matching results can then be compared with the ground truth. The ground-truth transformation information of the two construction datasets was calculated based on the ground control information. The comparison between different algorithms was conducted using the rotation error e^r and the translation error e^t:

$$\Delta\mathbf{T} = \mathbf{T}_g(\mathbf{T}_r)^{-1} = \begin{bmatrix} \Delta\mathbf{R} & \Delta t \\ 0 & 1 \end{bmatrix}, \tag{6.1}$$

$$e^r = arccos(\frac{tr(\Delta\mathbf{R}) - 1}{2}), \tag{6.2}$$

$$e^t = \|\Delta t\|, \tag{6.3}$$

wherein $tr(\cdot)$ denotes the trace. Furthermore, $\mathbf{T}_g$ and $\mathbf{T}_r$ represent the transformation matrix of the ground truth and the estimated one, correspondingly.

The second one is the time performance, which is used to test the efficiency of the proposed method. In our experiments, the execution time for the whole registration process was recorded.

Additionally, the GRPC method can also achieve an estimation of a seven DoFs transformation, including scaling and experiments, so evaluating the performance on scaling estimation is also conducted. The estimation of scaling is evaluated by scaling error:

$$\Delta s = |\frac{s_r}{s_g} - 1|, \tag{6.4}$$

where s_r and s_g are the scaling factor of ground truth and the estimated one, respectively.

6.3.2 Evaluation metric of semantic segmentation

For evaluation of the performance of semantic segmentation, we utilized the following evaluation metrics: F_1 measure (F_1), overall accuracy (OA), average accuracy (AA), and average F_1 score ($AvgF_1$). Among them, F_1 is specifically used to evaluate the classification performance on every single class, whereas OA, AA, and $AvgF_1$ score are applied to assess the performance on the whole test dataset.

6.3.3 Evaluation metric of change detection

In order to evaluate the performance of change detection, ground truth has been generated from the point clouds which were manually labeled with semantic categories. Precision (pr), recall (r), and F_1 are utilized to evaluate the performance of our method in each category, including changes and consistency. The overall performance is evaluated using OA and $AvgF_1$.

7 Results and Analysis

In this chapter, we will provide the qualitative and quantitative results of the proposed algorithms and methods listed in Chapters 3-5. Based on the results, we will analyze the methods from different aspects.

7.1 Registration results

7.1.1 Registered multi-station TLS point clouds

Registration results of the Bremen dataset

Experimental results of PBPC and GRPC using the Bremen TLS dataset are listed in Table 7.1. In the experiment of PBPC, the gridding size was set as 1 m. The maximum iteration number was set as 1000. The thresholds of the ratio of inliers and for estimating inliers in the RANSAC process were set to 0.9 and 0.1 respectively. In experiment of GRPC, the voxel size was set to 1 m. The filtering threshold for voxelization and binarization was set to 5.0. As shown in the table, the rotation error of PBPC is about 0.1 degrees and the translation error is nearly 0.2 m. The rotation error of GRPC was about 0.04 degrees, and the translation error was nearly 0.25 m. In light of the requirement of coarse registration, the two registration method's results are satisfactory. Additionally, the processing time was around 1 minute. Fig. 7.1 shows coarse registration results of the Bremen dataset using GRPC. As illustrated in the figure, the source point cloud and the target point cloud were well aligned. As illustrated in Fig. 7.2, it can be seen that the spires of the the Bremen bank and the facades were well matched.

In order to validate the effectiveness and efficiency of the GRPC method, we selected several baseline methods for comparison, which were the method using FPFH and a RANSAC process (FPFHSAC) [Holz et al., 2015], K4PCS, and Voxel-based 4-plane congruent sets (V4PCS) [Xu et al., 2019a]. FPFASAC is a feature-based method, which combines FPFH features and a RANSAC process for estimating transformation parameters. K4PCS and V4PCS are both improved strategies in the framework of 4PCS. In K4PCS, keypoints are utilized to replace points in point clouds to reduce the number of candidates and improve the robustness of selected points. Differing from 4PCS and K4PCS, V4PCS replaces points by planes as candidates for the congruent pairs. The baseline results of these methods were provided in [Xu et al., 2019a]. As shown in the table, all registration methods provided acceptable results for a coarse registration. Compared with these baseline methods, PBPC and GRPC achieved better registration results considering both rotation and translation errors. Additionally, PBPC and GRPC also showed their superiority in their efficiency. Meanwhile, compared with PBPC, GRPC showed better results with smaller rotation errors and same level of translation errors. However, PBPC is faster compared to GRPC.

It should also be noted that since the ground truth was generated by manual alignment followed by an ICP refinement, we also evaluated the quality of the ground truth by calculating

Table 7.1: Comparison of four registration methods using the Bremen dataset.

Methods	Rot_err (degrees)	Trans_err (m)	Time (s)
FPFHSAC [Holz et al., 2015]	0.3601	0.0692	318
K4PCS [Theiler et al., 2014]	0.3682	0.4826	256
V4PCS [Xu et al., 2019a]	0.1916	0.6312	78
PBPC	0.1010	0.2181	57
GRPC	0.0453	0.2436	67

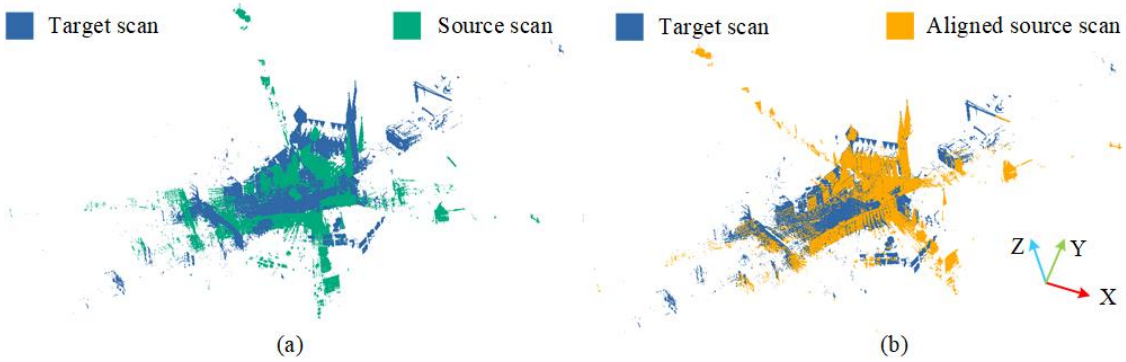

Figure 7.1: Registration result of the Bremen dataset using GRPC. a) Source and target point clouds shown in the same coordinate frame, b) aligned source and target point clouds.

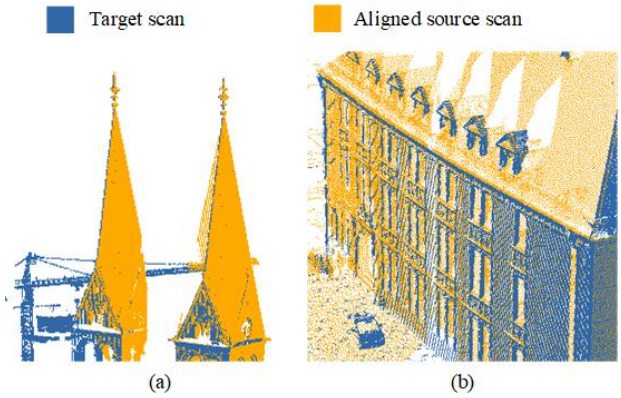

Figure 7.2: Details of the registration result of the Bremen dataset using GRPC. a) - b) Scene sections of the registered point clouds.

the residual distances between corresponding points between the aligned source using GRPC and target point clouds. In Fig. 7.3, the histograms of residual distances between corresponding points in the alignment results using the given ground truth and the aligned source scans using our method are shown. It can be seen that regarding the residual distances, the GRPC method provides better alignment results compared with ground truth created by manual alignment, with smaller mean residual distances and lower standard deviations being obtained.

Results of the WHU-TLS dataset

To further evaluate the versatility of the GRPC method to different scenes, three different scenes were selected from the WHU-TLS dataset for testing, including both regular-shaped areas (i.e., urban areas) and irregular-shaped areas (i.e., mountain cliffs). Since multiple scans were acquired for each scene, four scans were selected for testing, and each scan was matched to the corresponding reference scan. The voxel sizes used for registering point clouds of the scenes of the subway, the

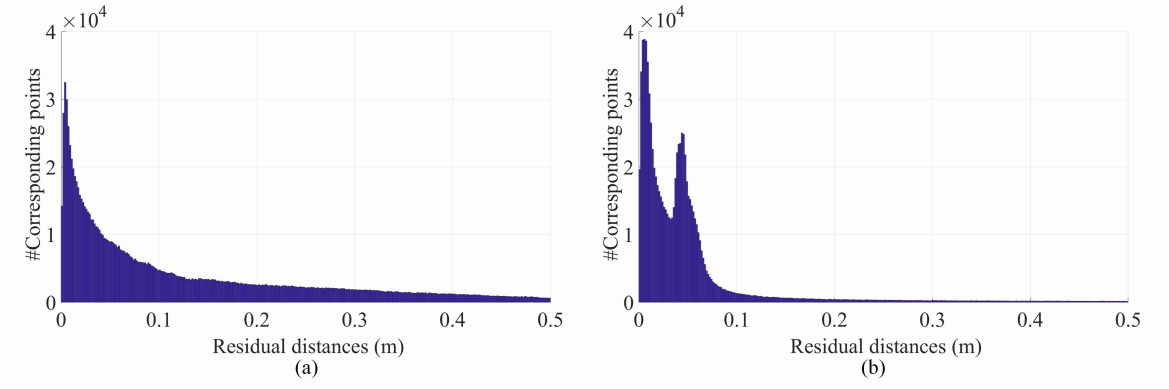

Figure 7.3: Histogram of residual distances between corresponding points in Bremen dataset between a) ground truth and the align source scan and between b) target and the align source scans.

park, and the mountain were 1.0 m, 1.5 m, and 1.0 m, respectively. Additionally, the filtering thresholds for voxelization and binarization were set to 3.0, 5.0, and 2.0. Table 7.2 lists the registration results of the GRPC method and the baseline results using Hierarchical merging-based multiview registration (HMMR) [Dong et al., 2018] provided by the publisher of the WHU-TLS dataset. The baseline method is also a hybrid method combining both global (for initial orientation) and local features (for fine registration) [Dong et al., 2020]. As shown in Table 7.2, for the scene of the subway, the rotation errors of the GRPC method were less than 0.2 degrees, and the translation errors were less than 0.6 m. Meanwhile, the processing time was less than two minutes. Compared with the baseline method, GRPC provided better registration outputs in several cases (i.e., the matching between Scans 5 and 3 and Scans 5 and 6), with better results achieved in both rotations and translations.

Table 7.2: Performance comparison of the GRPC method and the baseline using the WHU-TLS dataset.

Scene	Registration pair (Target & Source)	Baseline [Dong et al., 2018] Rot_err (degrees)	Baseline [Dong et al., 2018] Trans_err (m)	Our GRPC Rot_err (degrees)	Our GRPC Trans_err (m)	Our GRPC Time (s)
Subway	5 & 3	Failed		0.0490	0.4848	65
	5 & 4	0.0722	0.7025	0.1841	0.2125	120
	5 & 6	0.0931	1.0286	0.0692	0.5493	93
Park	14 & 13	0.0864	0.0438	0.0795	0.4059	158
	14 & 15	0.0572	0.0358	0.0646	0.3202	137
	15 & 16	0.0256	0.0112	0.0862	0.7704	135
Mountain	3 & 2	0.0495	0.0180	0.2338	0.4010	79
	4 & 3	0.0422	0.0090	0.1827	0.2946	74
	4 & 5	11.1691	7.9453	0.1332	0.3263	69

For the scene of the park, the GRPC method achieved less than 0.1 degrees rotation errors, which was the same level as the results provided by the baseline method. However, GRPC's translation errors were larger than 0.3 m, while the baseline method can provide translation errors at a centimeter-level. Compared with the first scene, the processing time is longer, about two and a half minutes. For the scene of the mountain, GRPC provided results about a rotation error of around 0.2 degrees and a translation error of about 0.3 m. In general, compared with the GRPC method, baseline [Dong et al., 2018] achieved better results in most cases with rotation errors less than 0.1 degrees and most of the translation errors in centimeter-level, which may benefit from the iterative optimization procedure. However, it also failed in some cases, namely, the matching of Scans 4 and 5 of the mountain scene when details of scans changed in a broad range. It shows that one advantage of the GRPC method is that it runs stable under different

situations in various registration datasets. Additionally, the GRPC method is efficient concerning the processing time. Fig. 7.4 depicts the registration results of multiple scans in different scenes from the WHU-TLS dataset. The reference scans for the multiscan registration in the scene of the subway, the park, and the mountain were Scan 5, Scan 14, and Scan 4, respectively. As illustrated in the figure, it can be observed that walls, buildings in the park, and the mountain's valley were well aligned.

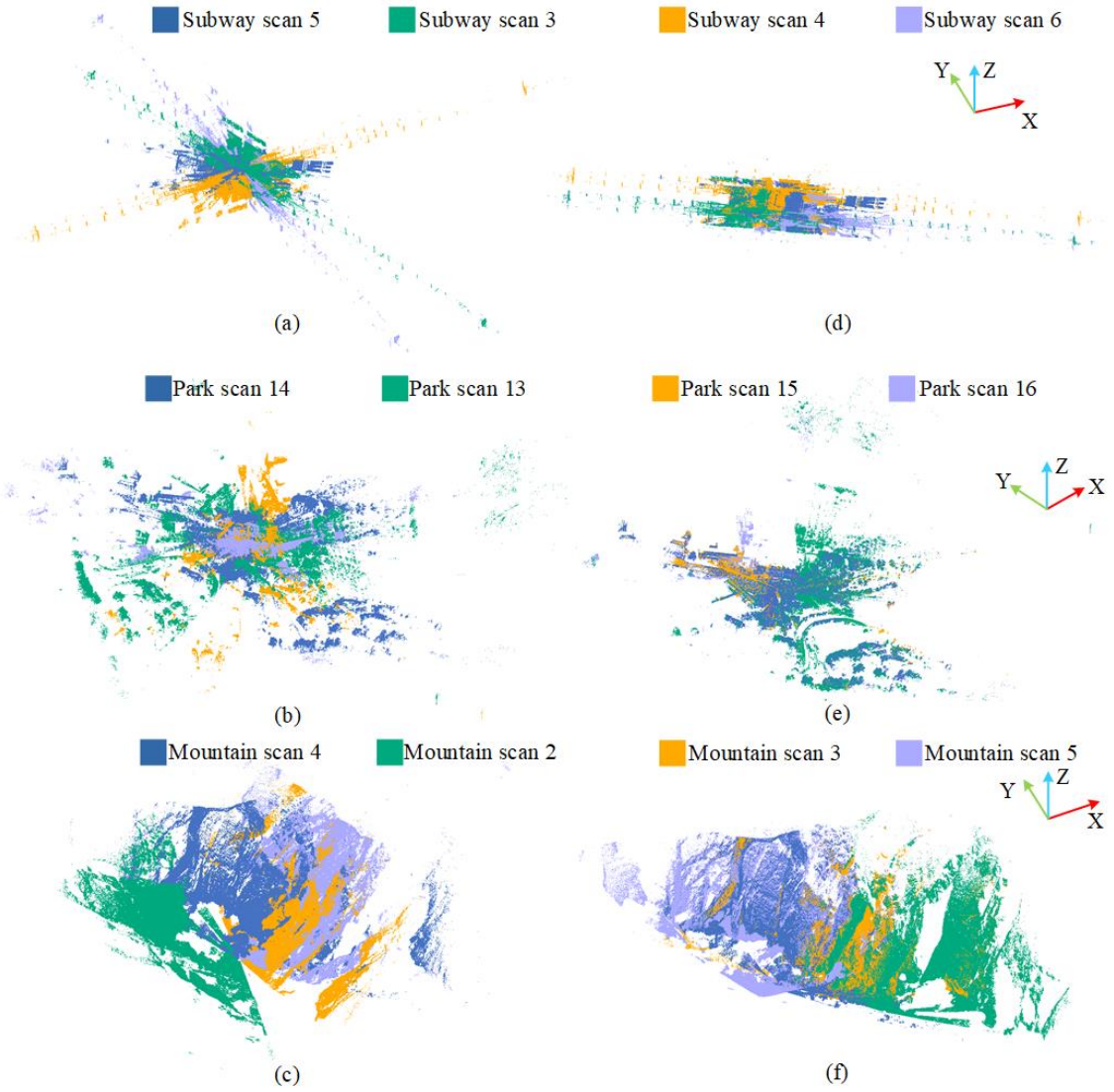

Figure 7.4: Registration results of the WHU-TLS dataset using GRPC, with color representing different scans. a) - c) Source and target point clouds shown in the same coordinate frame, d) - f) aligned source and target point clouds.

Results of the RESSO dataset

Apart from the Bremen-TLS and WHU-TLS datasets, we further tested the GRPC method using another benchmark dataset, namely the Resso dataset. In the experiments, the voxel size was set to 1.0 m. The filtering threshold for voxelization and binarization was set to 3.0. In Table 7.3, it can be seen that the rotation errors were all smaller than 0.3 degrees, and the translation errors were less than 0.6 m. Besides, the processing is comparatively fast, with processing time less than 50 s. The registration results of the baseline method, Plane-based descriptor (PLADE) [Chen

et al., 2019], provided by the data publisher, are also given in Table 7.3. In PLADE, a plane and line-based descriptor are utilized to establish correspondences between point clouds. It can be seen that the GRPC method always performed better in estimating rotations. However, as for the estimation of translations, GRPC and PLADE achieved almost the same level results.

Table 7.3: Performance comparison of the GRPC method and the baseline using the Resso dataset.

Scene	Registration pair (Target & Source)	Baseline [Chen et al., 2019] Rot_err (degrees)	Trans_err (m)	Our GRPC Rot_err (degrees)	Trans_err (m)	Time (s)
7a	2 & 1	0.3265	0.2082	0.2650	0.5610	45
	2 & 3	0.0810	0.0854	0.0727	0.3075	46
	2 & 5	0.4475	0.3626	0.1951	0.4000	43
	2 & 6	0.5060	0.5741	0.0485	0.2909	48
	2 & 7	0.2497	0.4057	0.2844	0.2391	44

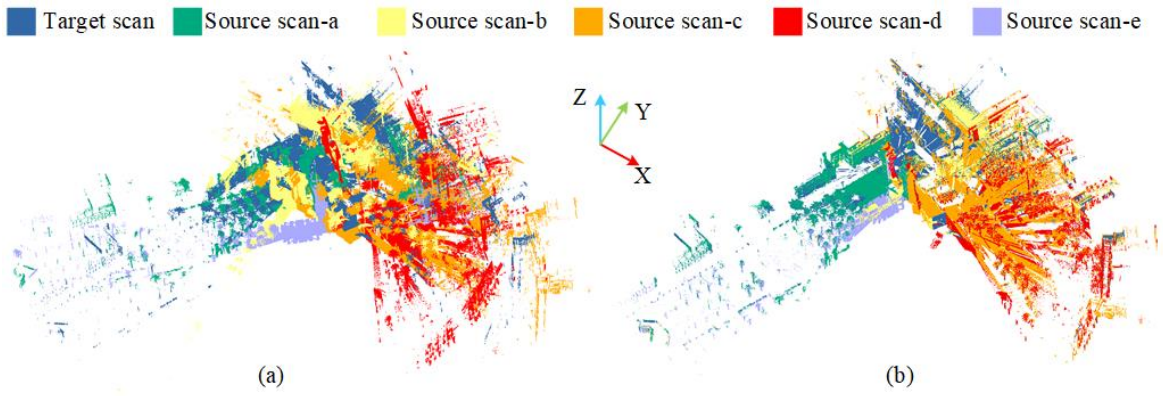

Figure 7.5: Registration results of the Resso dataset using GRPC, with color representing different scans. (a) Source and target point clouds shown in the same coordinate frame. (b) Aligned source and target point clouds.

The visualized results of the registration of the selected scene in the Resso dataset are shown in Fig. 7.5. It can be observed that the spires, palm trees, and walls of buildings were well matched.

7.1.2 Registered multi-temporal construction dataset

Fig. 7.6 presents the registration results of the multitemporal dataset acquired on the HFK construction sites achieved by the GRPC method. As shown in the figure, we can see that the point cloud sequence is well united to the same reference system from different views. Table 7.4 provides the quantitative results for the registration of the point cloud sequence. We can achieve registration of all source point clouds using the GRPC method, with satisfactory results. The rotation errors are all smaller than 0.4 degrees, and the translation errors are less than 0.2 m. By comparing the multitemporal dataset, we can further detect changes during the construction process. However, we register all the other point clouds sequentially referencing to the point cloud acquired on the last acquisition time. When the time sequence is long, the registration will produce high translation or rotation errors or even fail since the errors are accumulated during the sequential process. Therefore, in the practical cases, the registration of sequential point clouds can be conducted in some ways to eliminate the accumulating effect, i.e., registration of point clouds in a closed loop and eliminating closure errors.

Table 7.4: Registration results of the construction dataset using GRPC.

Acquisition date	Rotation error (degrees)	Translation error (m)
2014/11/20	0.3201	0.0172
2014/12/12	0.2549	0.1872
2015/01/16	0.1067	0.0328
2015/02/26	0.2385	0.1507

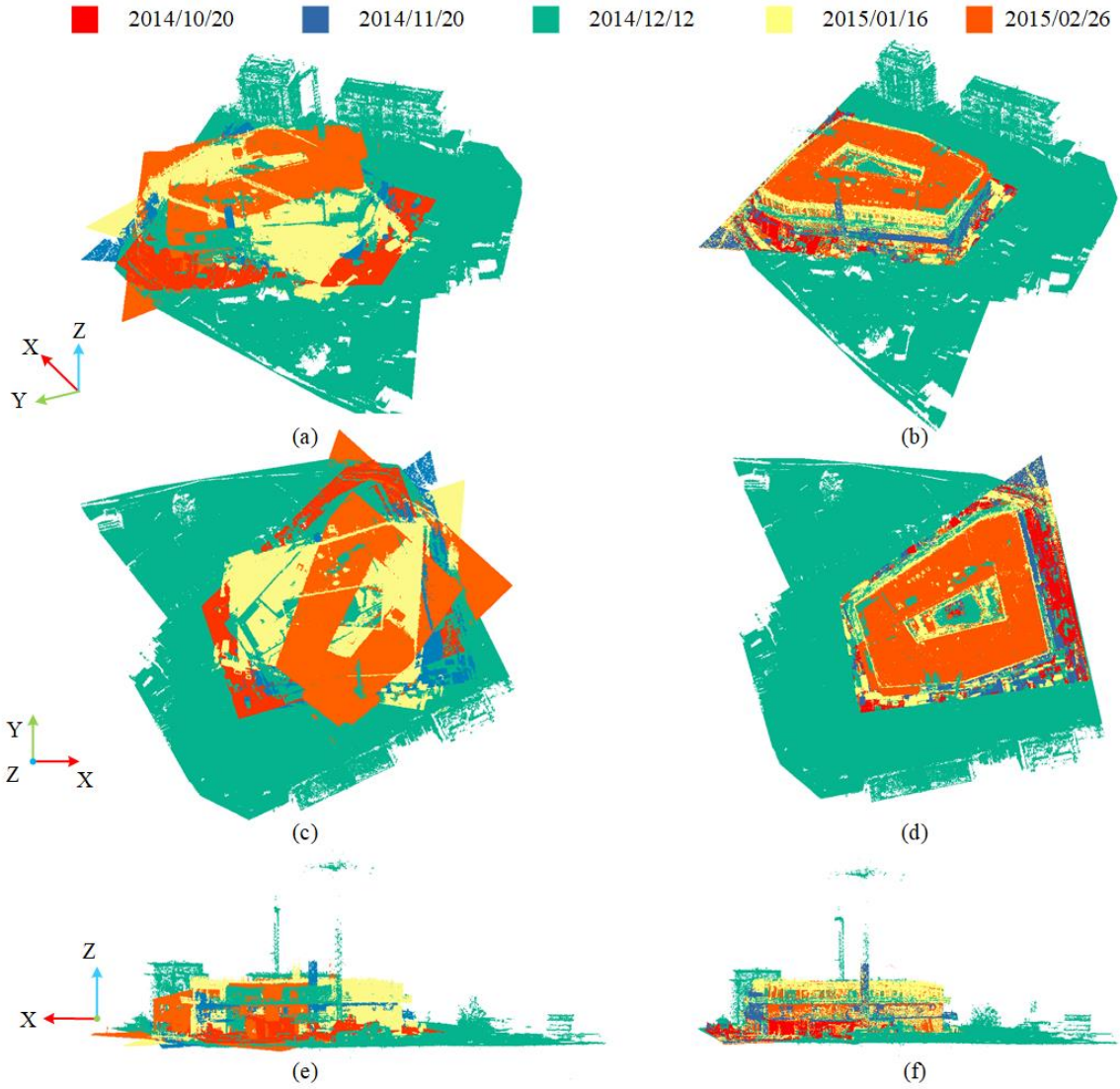

Figure 7.6: Registration results using the construction dataset using GRPC. a) - b) Points clouds before and after registration from oblique view, c) - d) points clouds before and after registration from top view, e) - f) are points clouds before and after registration from side view.

7.1.3 Sensitivity analysis

Performance under different data properties

Three benchmark datasets for point cloud registration were tested in the experiments, including different point densities, different coverage areas, and different scenes. In Fig. 7.7, mean values and standard deviations of the residual distances between corresponding points in the aligned

results are shown, in which both ground truth and aligned source scans using the GRPC method were used as references.

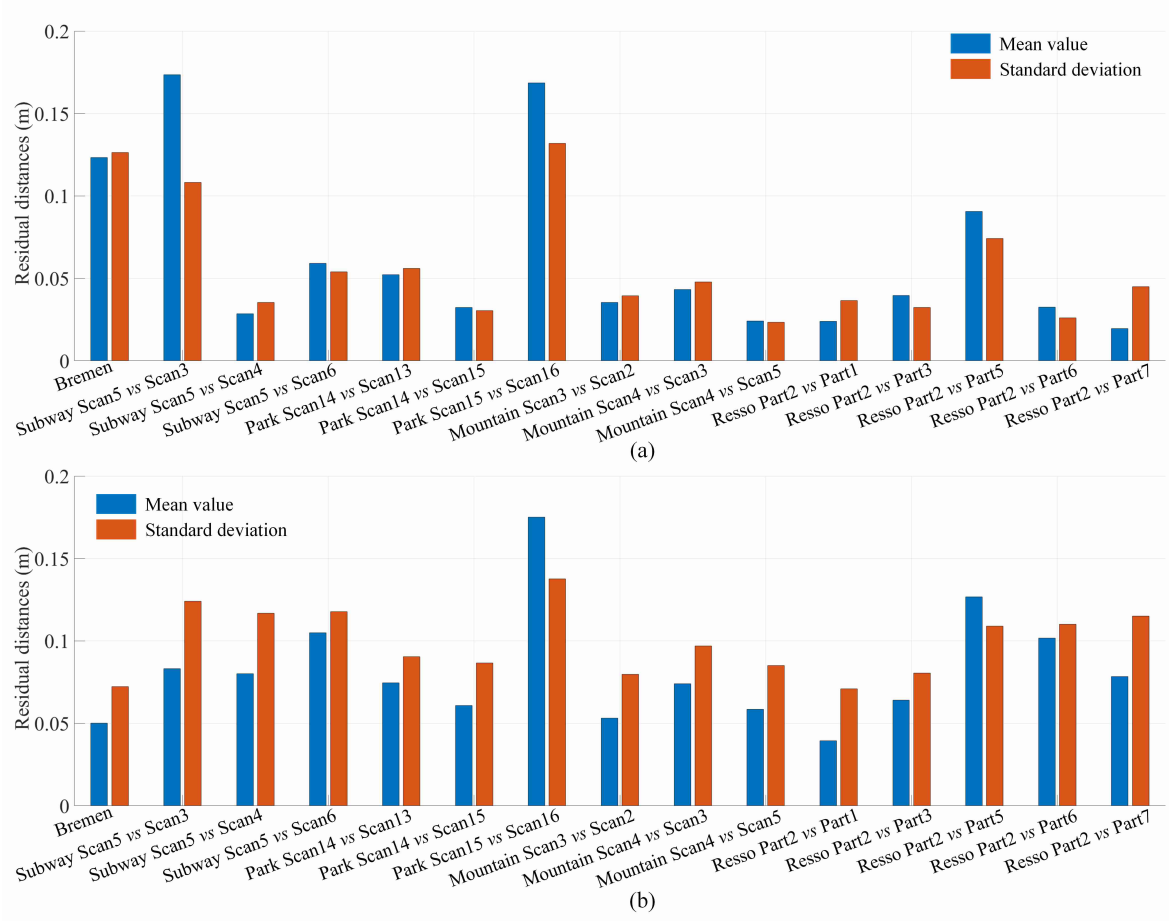

Figure 7.7: Mean values and standard deviation of residual distances between corresponding points in all pairs of scans between a) ground truth and the align source scan and between b) target and the align source scans.

For most registration pairs, the mean values and standard deviations of residual distances in results using the GRPC method were close to those using ground truth. It means that the GRPC method can provide acceptable results under different evaluation criteria, even employing checking point-by-point details. Additionally, we selected several representative registration pairs and illustrate the distribution of registration errors in Fig. 7.8. As shown in the figure, for most registration pairs, the distance errors were less than 0.25 m. By comparing different registration pairs, it can be seen that although the geometric characteristics of the acquired scene changes and data property changes, the GRPC method can produce nearly equal and high quality of registration.

Influence of voxelization resolutions

The resolution of voxels is a significant factor influencing the result of registration. The resolution represents the geometric size of each voxel used in the step of voxelization and binarization. In the experiments, two registration pairs were selected from the aforementioned tested datasets. The first one is the pair of Scans 2 and 3 from the Resso dataset, which serves as a representative of regular-shaped areas. The other one is the pair of Scans 3 and 2 from the scene of a mountain

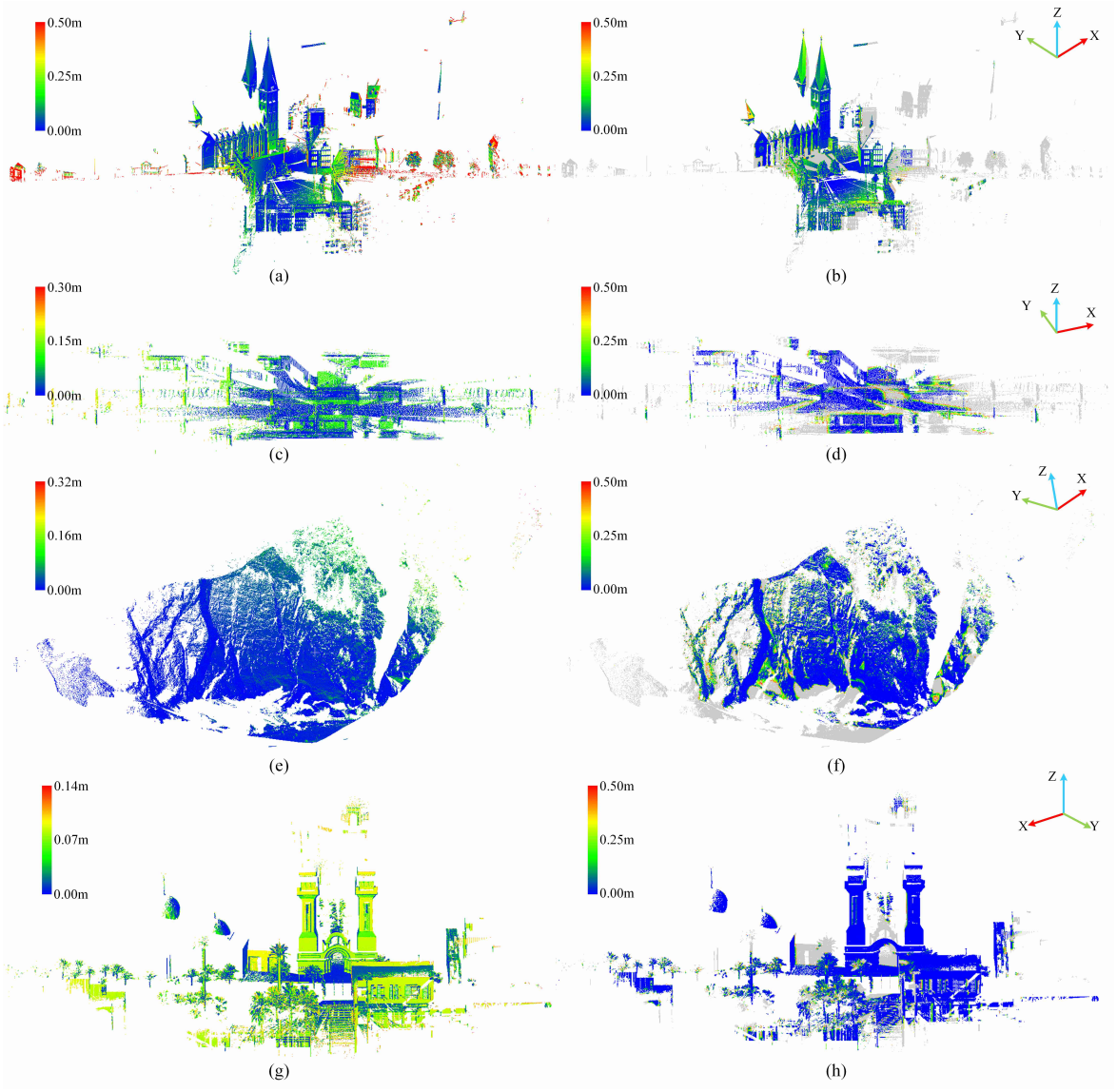

Figure 7.8: Selected registered results colored by the residual distances between corresponding points, where the gray points represent the non-overlap areas. Point distances between ground truth and the align source scan in a) Bremen, c) WHU-TLS subway, e) WHU-TLS mountain, and g) Resso. Point distances between target and the aligned source scans in b) Bremen, d) WHU-TLS subway, f) WHU-TLS mountain, and h) Resso.

cliff in the WHU-TLS dataset, which stands for a representative of irregular-shaped areas. In the experiments, the sizes of voxels are set to 1.0 m, 2.0 m, and 3.0 m. In Fig. 7.9, the registration results using the GRPC method, including rotation errors, translation errors, and processing time, are provided.

As we can predict, when the voxel size gets larger, the execution time will decrease. The results perfectly proved this assumption. For both datasets, the processing time experienced a remarkable drop along with the increment of voxel resolution. On the other hand, it is also noticeable that both rotation errors and translation errors for the two datasets showed drastic improvements. For the Resso dataset, the rotation errors increased from less than 0.1 degrees to 0.7 degrees, and the translation errors rose from 0.3 m to larger than 1 m. For the WHU-TLS mountain dataset, the rotation errors increased from 0.2 degrees to about 0.8 degrees, while the

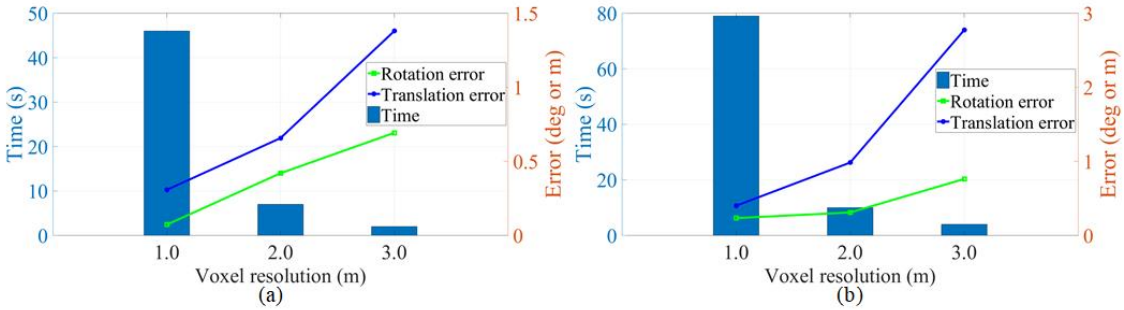

Figure 7.9: Sensitivity analysis of GRPC on voxel resolutions. a) Results using the Resso dataset, b) results using the WHU-TLS mountain dataset.

translation errors expanded from 0.4 m to almost 2.8 m. It can be seen that no matter for regular-shaped areas or irregular-shaped areas, the voxel resolution is an essential factor that influences registration results. One reason may be that the voxel size actually defines the sampling rate in the process of voxelization. When the voxel size is large, a sparse sampling is conducted on point clouds, which leads to strong aliasing effect.

Influence of scaling changes

All tested datasets we used in the experiments provide no scaling changes. To investigate the effectiveness of the scaling estimation and the influence on the estimation of other transformation parameters, we generated several simulated registration pairs of point clouds with scaling changes by zooming out the source point cloud. We selected the pair of Scans 2 and 3 from the Resso dataset and the pair of Scans 3 and 2 from the WHU-TLS mountain dataset as registration pairs for testing. The source scans, namely Scan 3 from the Resso dataset and Scan 2 from the mountain dataset, were zoomed out with various scaling factors. The target scans remained no changes. As illustrated in Fig. 7.10, it is clear that when the scale difference gets larger, the registration accuracies decrease with larger rotation, translation, and scaling errors no matter for regular-shaped areas and irregular-shaped areas. It could be explained that when the point cloud is zoomed out with a large scale factor, the aliasing effect will be caused by a relatively sparse sampling process on the point cloud.

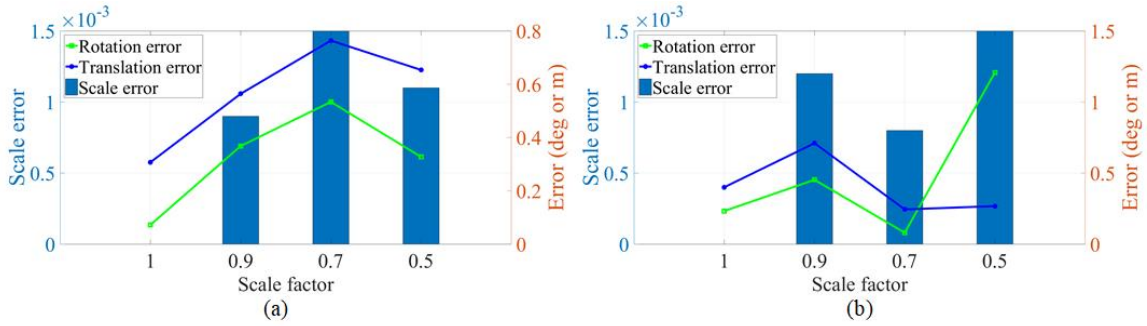

Figure 7.10: Sensitivity analysis of GRPC on changes of scale. a) Results using Resso dataset, b) results using WHU-TLS mountain dataset.

Additionally, as shown in the figure, the influences of scaling changes on rotation errors and translation errors are almost with the same trend except for some individual cases. Since point clouds have been zoomed to approximately the same scale after scaling, the accuracy of estimated

translations will not be influenced by the aliasing effect caused by a sampling process but merely influenced by errors in the estimation of scaling and rotations.

Influence of signal-to-noise ratios

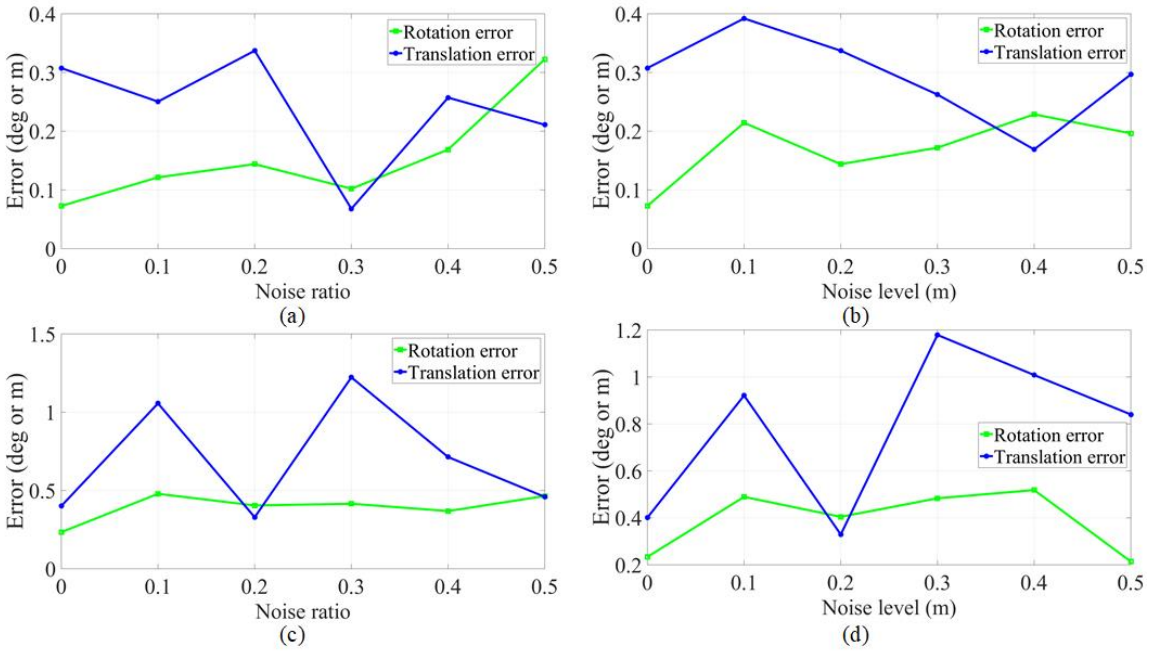

Figure 7.11: Rotation and translation errors with different noise ratios and noise levels using GRPC. a) - b) Results using the Resso dataset, c) - d) results using WHU-TLS mountain dataset.

To validate the robustness of the GRPC method, we conducted further experiments, which added noises to the original point clouds. In experiments, we selected two registration pairs from the aforementioned datasets. One pair is Scans 2 and 3 from the Resso dataset, and another pair is Scans 3 and 2 from the WHU-TLS mountain dataset. Meanwhile, Gaussian noises with different noise ratios and different noise levels were added to corresponding point clouds. It should be noted that noise ratio means the proportion of points that are changed to noise, while the noise level means the amplitude of the added noise. Thus, the influence of noise on regular-shaped and irregular-shaped datasets can also be investigated. The voxel size and the filtering threshold for voxelization and binarization were set as 1.0 m and 3.0. As shown in Fig. 7.11, the registration accuracies vary in a small range with changes in noise ratios and noise levels. It demonstrates the robustness of the GRPC method and proves that GRPC can still be effective in a highly noisy situation. Comparatively, the registration of the mountain dataset is more sensitive to the influence of noise, with higher translation errors gained. However, for the mountain dataset, the estimation of rotations seems to be more stable under the changes of both noise ratios and noise levels.

Influence of overlapping ratios

In the real world, occasionally, it is unpredictable for a pair of scans to have varying overlap ratios, which is challenging work for point cloud registration. Thus, we also investigated the influence of overlap ratios on registration results using the GRPC method. As depicted in Table 6.3, the dataset provides several scans with different overlap ratios varying from 0.60 to 0.24, but with the same data quality. The voxel sizes were all set to 1.0 m, and the filtering thresholds were

set to 3.0, as mentioned in Section 7.1.1. In Fig. 7.12, overlap ratios, and their corresponding rotation and translation errors are shown. As illustrated in Fig. 7.12, the registration results are not directly positively influenced by overlap ratios. When the overlap ratio dropped from 0.52 to 0.24, the registration accuracy was still acceptable with a rotation error by about 0.3 degrees and a translation error by 0.25 m. Generally, it shows that GRPC is kind of robust to the variations of overlap ratios.

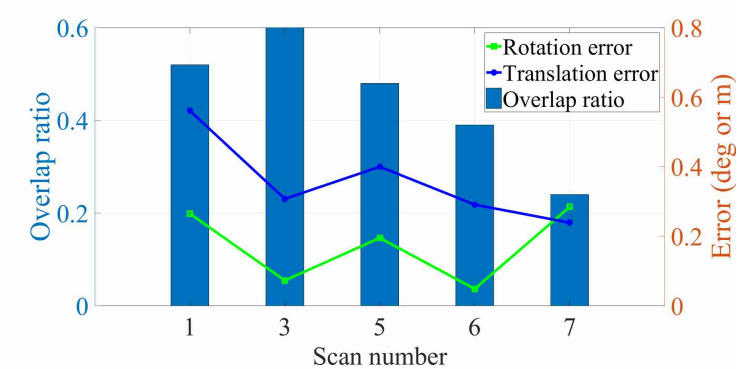

Figure 7.12: The rotation and translation errors of the registration results using the Resso dataset with different overlap ratios.

7.2 Semantic segmentation results

7.2.1 Semantically segmented urban scenes using MLCE

The performance of MLCE was tested on the 2018 GRSS DFC LiDAR dataset. We compare the classification results on multi-scale features with that of a single-scale neighborhood, and the classification results on dimensionality-reduced data using LPP with those using some benchmark DR methods (PCA and LDA) and original geometric features (OGF). Since our main focus is to assess the discriminative performance of the learned geometric features, we, therefore, use a classic classifier, namely random forest whose number of trees is selected to be 100 by cross-validation. Moreover, ten replications were performed for selecting training and test samples.

Performance comparison and analysis between multi-scale feature extraction and single scale feature extraction

The representation of the neighborhood of points is a crucial issue directly influencing classification performance. Thus, the neighborhood for each point is of great significance. Table 7.5 shows the classification accuracy obtained by using different neighborhood scales for feature extraction.

The multi-scale feature extraction outperforms the other two feature extraction methods. Compared to the single scale and dual scale, multi-scale feature extraction increases OA by 4.52% and 0.96%, respectively. As for AA, on the other hand, the corresponding increase is, 7.14% and 1.52% respectively. Overall, with the increase of scales of neighborhoods, the performance of classification also increases. It means that multi-scale features provide a more distinctive representation of local context.

Performance comparison and analysis between LPP and classical DR methods

Table 7.6 lists the OA and AA of four different methods with optimal parameters determined by cross-validation on the training set.

Table 7.5: Comparison of different feature extraction methods using the DFC2018 dataset (All values are in %). Noted that the highest values in OA and AA are marked with bold texts.

Method	Neighborhood size	OA	AA
Single-scale	k_1=10	74.98	60.70
Dual-scale	k_1=10, k_2=20	78.54	66.32
Multi-scale	k_1=10, k_2=20, k_3=30	**79.50**	**67.84**

Table 7.6: Comparison of different dimensionality reduction methods using optimal parameters with the RF classifier on the DFC2018 dataset (All values are in %). Noted that the highest values in OA and AA are marked with bold texts.

Method	Neighborhood size	OA	AA	Running time (s)
OGF	/	79.50	67.84	/
PCA	d = 10	80.32	67.62	0.09
LDA	d = 19	70.71	56.36	0.15
LPP	d = 4, k = 15	**86.66**	**77.62**	1.78

The LML method outperforms other methods. Compared to OGF, PCA, and LDA, LPP increases OA by 7.16%, 6.34%, and 15.95%, respectively. For the AA, on the other hand, the corresponding increase is 9.78%, 10%, and 21.26%, respectively. The classification maps are shown in Fig. 7.13. These results demonstrate the effectiveness of this LML method and imply that it successfully contributes to extracting robust and discriminative low-dimensional features.

Moreover, we also provided the classification results with the MLCE method on the test scene given in the DFC2018 dataset, yielding an OA of 40%. To our knowledge, the resulting accuracy is reasonable to a large extent, since our task is point cloud classification without considering intensities.

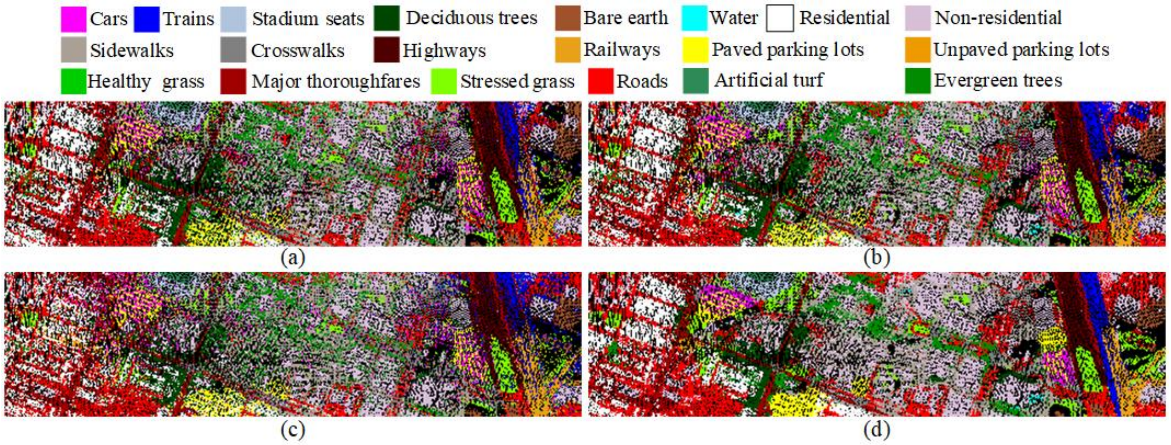

Figure 7.13: Classification maps of different algorithms obtained using the RF classifier on the DFC2018 dataset. a) Classification result uisng OGF, b) classification result using PCA, c) classification result using LDA, d) classification result using LPP.

Sensitivity analysis of parameters in low-dimensional embedding

The sensitivity of parameters is tested by varying the number of neighbors k, the size of reduced dimensionality d and the variance of Gaussian kernel in weight determination σ for LPP. As

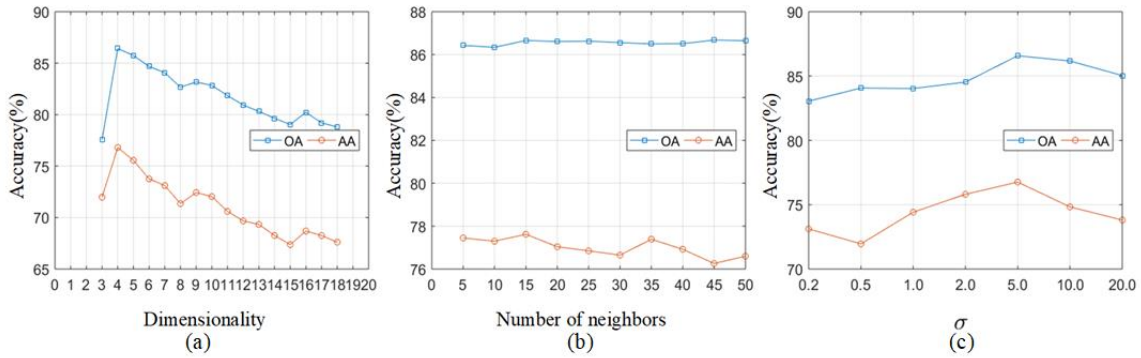

Figure 7.14: Sensitivity analysis of MLCE on three parameters: a) reduced dimensionality, b) number of neighbors, c) variance of Gaussian kernel.

shown in Fig. 7.14, the performance of LPP is some kind of sensitive to the parameters. In general, from the perspective of reduced dimensionality, the classification accuracy increases when decreasing dimensionality. When the reduced dimensionality reaches approximately four, the accuracy reaches the nearly optimal level. Compared with reduced dimensionality, LPP is much less sensitive to the other two parameters, namely the number of neighbors and the variance of the Gaussian kernel. When the number of neighbors gradually increases, the corresponding classification accuracy increases moderately to a peak (e.g., k is equal to around 15) and then fluctuates. A large number of neighbors may obscure the local structure, whereas a small number of neighbors may be not sufficient to represent the local structure, causing the fluctuation of the performance of LPP. The accuracy reaches a peak when the variance of the Gaussian function in weight determination is 5.0. Thus we can determine the optimal parameters for LPP in the application of classifying the DFC2018 dataset for reduced dimensionality, number of neighbors, and the variance of Gaussian function to be 4, 15, and 0.5.

7.2.2 Semantically segmented urban scenes using DPE

The performance of the DPE method was tested on two benchmark datasets, including the ISPRS benchmark dataset and the AHN3 dataset. We also conducted ablation study on the DPE method to test the effectiveness of each module in the DPE method using the two datasets. In addition, the sensitivity of parameters were tested using the ISPRS benchmark dataset.

Classification results of the ISPRS benchmark dataset

To test the performance of the DPE method, we conducted comparisons between the results with the following conditions: (i) using original single-scale deep features (SDF) with original PointNet++, (ii) using multi-scale deep features (MDF) with the RF classifier, and (iii) using the embedded features obtained using JME with the RF classifier. Furthermore, to evaluate the performance of the feature dimensionality reduction in addition to the DPE method, we implemented several classic dimensionality reduction methods to provide fair comparison; with such a comparison, we can investigate the effectiveness of JME. Moreover, to validate the feasibility of GGO used for smoothing the classification results, we conducted comparisons between the results with and without GGO. Both quantitative and qualitative results prove the effectiveness of the DPE. More specifically, we finally achieved an OA of 83.2% for labeling the nine classes.

Comparison between SDF and MDF

For the point-based deep neural network, the sampling of point clouds in large urban scenes is an important issue that directly affects the classification results. To validate the performance of our MDF, we compared the results obtained by extracting deep features using HDL (i.e., MDF) to those by extracting features directly from the original PointNet++ (i.e., SDF). Here, Table 7.7 lists the classification results of SDF and MDF with the RF classifier. In experiments, the numbers of trees used in the RF classifier were set to 100 and 200, respectively. Concerning OA, MDF

Table 7.7: Comparison of different feature learning methods using the ISPRS benchmark dataset (All values are in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	power	low_veg	imp_surf	car	fence_hedge	roof	fac	shrub	tree	OA	$AvgF_1$	Time
SDF	58.5	81.4	98.1	55.3	19.2	86.9	41.0	33.2	71.2	79.1	60.5	/
MDF	64.2	85.1	99.2	68.9	19.2	88.2	36.5	37.7	69.2	81.2	63.1	1h17m30s
JME	**66.1**	**86.4**	**99.4**	**74.1**	**20.5**	**90.9**	**41.9**	**39.2**	**72.2**	**83.0**	**65.6**	39m12s+20m27s

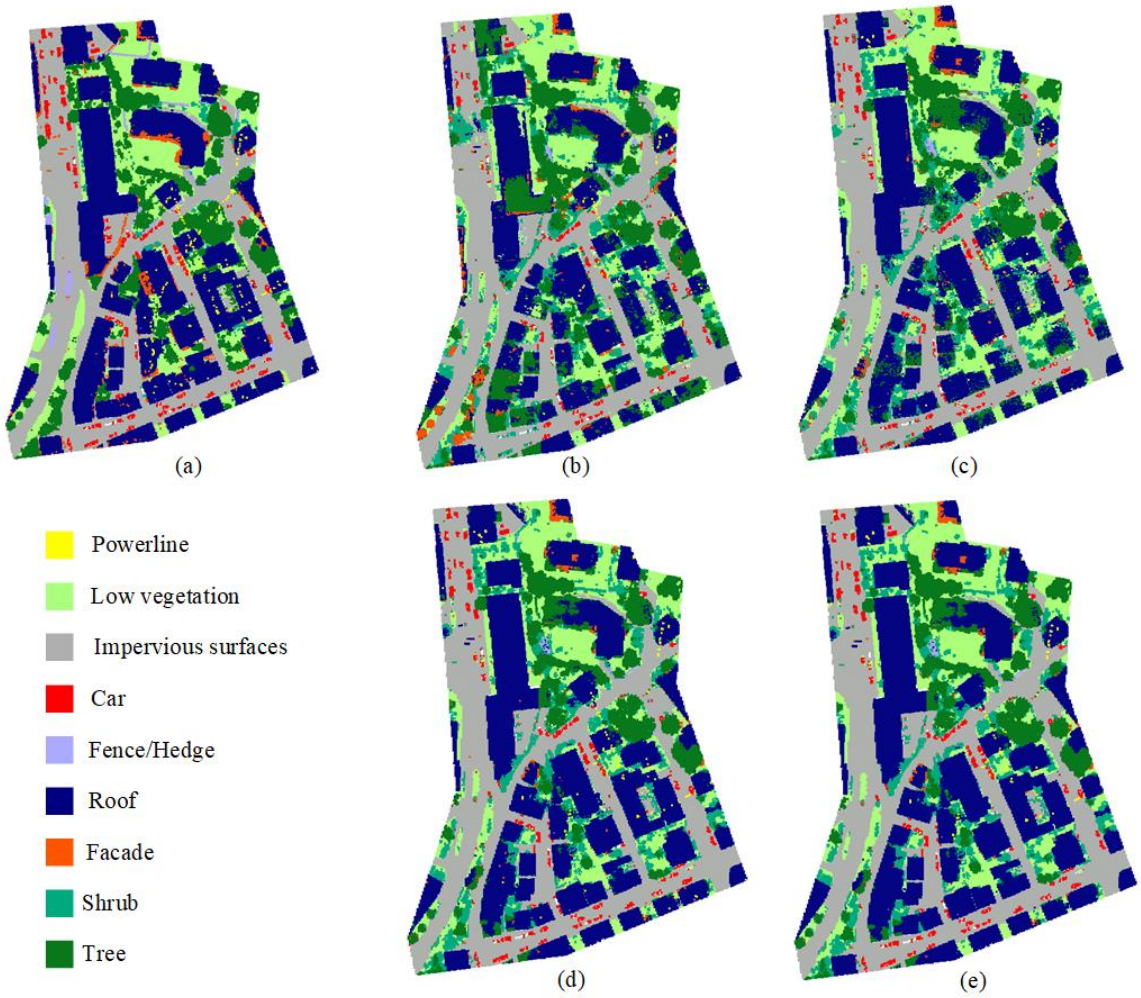

Figure 7.15: Classification results of test area 1. a) Ground truth, b) the classification result with SDF, c) the classification result with MDF, d) the classification result with JME, e) the smoothed classification result with GGO.

can largely improve the performance of classification with a 2.1% increment in OA. Additionally, an increase of 2.6% is seen in $AvgF_1$. Meanwhile, the classification accuracy show improvement

for a majority of the classes. The F_1 of powerline, low vegetation, impervious surface, car, roof, and shrub show considerable increase by 5.7%, 3.7%, 1.1%, 13.6%, 1.3%, and 4.5%, respectively. This indicates that the HDL strategy can produce higher classification accuracy, especially for the classification of small objects such as cars and pole-like objects (e.g., parts of powerlines). This can be attributed to the reason that the induction of multi-scale contextual information can provide better representation for such small objects, especially in improving object integrity. The visualization of the classification results is presented in Fig. 7.15 and Fig. 7.16. Here, PointNet++ provides a good initial result, which indicates the efficacy of the deep neural network in providing informative features. Moreover, through the utilization of the hierarchical subdivision strategy, some wrongly classified areas can be corrected.

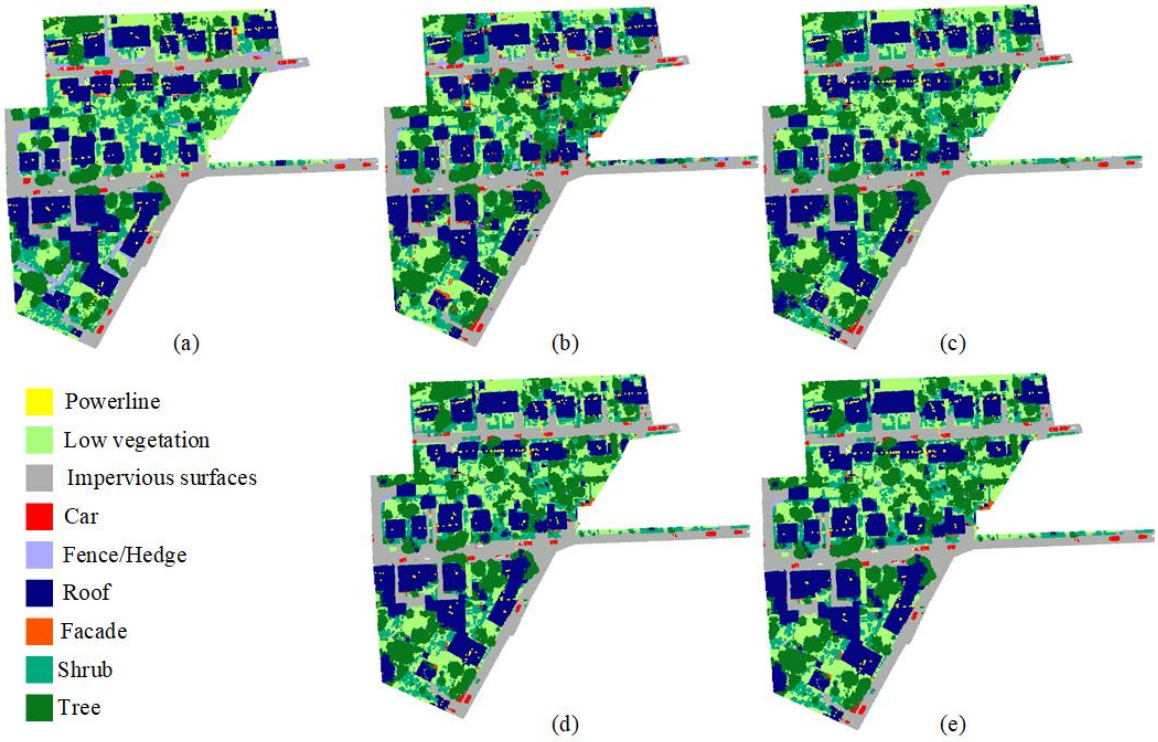

Figure 7.16: Classification results of test area 2. a) Ground truth, b) the classification result with SDF, c) the classification result with MDF, d) the classification result with JME, e) the smoothed classification result with GGO.

Effectiveness of feature embedding with JME

Regarding the embedding process on the obtained MDF, to determine the effectiveness of the JME method, the original MDF and the embedded features testing with the same classifier (i.e., RF) were compared. The 384-dimensional deep features were embedded into a 50-dimensional feature vector. The results of the comparison are listed in Table 7.7. Compared to the classification results using MDF, JME achieves an increment of 1.8% and 2.5% for OA and $AvgF_1$, respectively. Regarding the classification accuracy of each class, all F_1 values show improvement with the embedding process, especially for cars and facades that show improvement of more than 5%. These results indicate that the JME method has proved to be effective for the classification task. More specifically, the features embedded with JME can provide improved classification results compared to either MDF or SDF with the integration of contextual information in both the feature and the spatial domains. Furthermore, the processing time in the classification step of MDF and JME is provided. It should be noted that the time provided for MDF comprises two

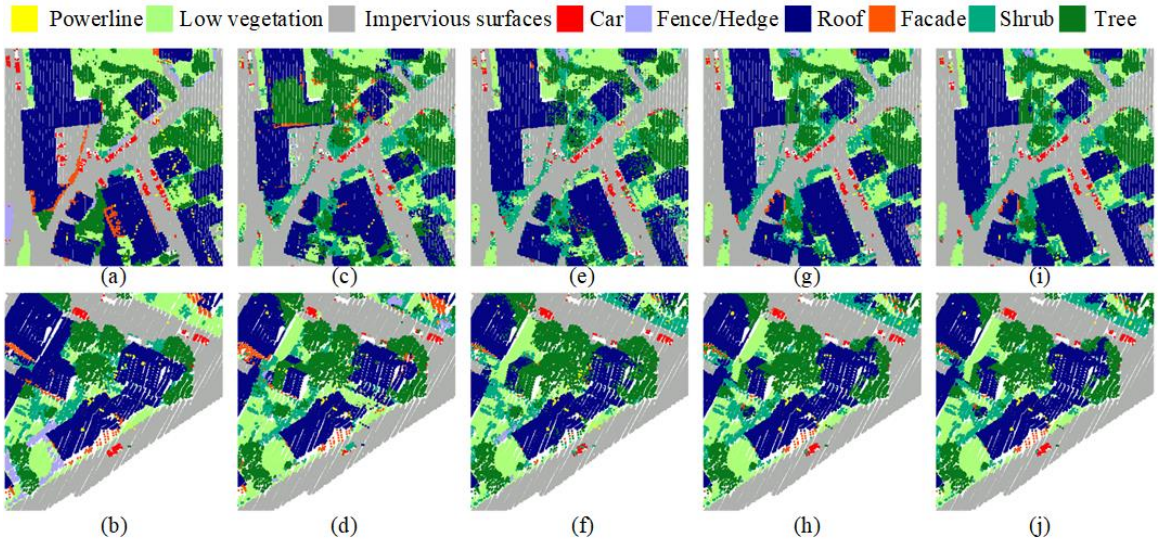

Figure 7.17: Details of classification results of the ISPRS benchmark dataset for comparison. a) - b) Ground truth, c) - d) the classification result with SDF, e) - f) the classification result with MDF, g) - h) the classification result with JME, i) - j) the smoothed classification result with GGO.

Table 7.8: Comparison of existing feature dimensionality reduction methods (All values are in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	Optimal parameters	power	low_veg	imp_surf	car	fence_hedge	roof	fac	shrub	tree	OA	$AvgF_1$
PCA	d=10	0.0	33.7	50.5	3.1	0.1	59.7	4.1	2.3	38.4	43.7	21.3
LLE	d=30, k=10	33.4	85.1	98.9	65.9	15.2	89.1	35.6	41.0	72.3	81.6	59.6
SE	d=30, k=10	0.0	83.2	94.0	71.7	0.0	85.3	0.0	32.1	72.4	79.7	48.8
LPP	d=15, k=30	50.4	79.7	95.8	46.7	16.2	84.3	39.9	30.3	69.3	77.2	56.9
1D-NN	/	**66.9**	85.8	**99.4**	73.3	12.5	90.3	40.5	**39.8**	71.6	82.4	64.4
JME	d=50, k=5	66.1	**86.4**	**99.4**	**74.1**	**20.5**	**90.9**	**41.9**	39.2	**72.2**	**83.0**	**65.6**

parts: the time required for feature dimensionality reduction and for classification. It is evident that the application of JME largely reduces the processing time spent on classification. A detailed visualization of the performance using the embedded features is illustrated in Fig. 7.17. It can be seen from the figure that some small wrongly labeled areas such as the boundaries of buildings adjacent to trees can be corrected. Furthermore, the "salt and pepper" effect in the classification map is considerably eliminated compared to the classification map of MDF, which corroborates the effectiveness of embedding contextual information in producing smoother results.

Comparison with other dimensionality reduction methods

To further demonstrate the strength of the integration of spatial information in JME, we compared it with other commonly utilized dimensionality reduction methods such as PCA, LLE, spectral embedding (SE) [Belkin & Niyogi, 2003], and LPP. Apart from the aforementioned commonly-used dimensionality reduction methods, we constructed a simple 1D neural network (1D-NN) for dimensionality reduction. It is composed of a fully connected layer that reduces the feature dimensionality to 64, a fully connected layer, and a softmax layer for classification. The classification results are presented in Table 7.8. JME increases OA by 39.3%, 1.4%, 3.3%, 5.8%, and 0.6% compared to PCA, LLE, SE, LLP, and 1D-NN, respectively. Besides, the corresponding increases on the $AvgF_1$ are 44.3%, 6.0%, 16.8%, 8.7%, and 1.2%, respectively. These results imply the efficacy of the manifold learning method-JME, as the accuracies of JME outputs are higher than those gained by means without using manifold learning. Additionally, comparison between

these manifold learning-based methods indicates that JME can provide a more discriminative feature representation with lower dimensionality. Especially, compared to the results of LLE, JME achieves remarkable improvement through approximately a 6% increment in $AvgF_1$. The joint learning method shows improvement on the LLE by incorporating contextual information in the spatial domain. Moreover, the joint learning method provides improved performance when compared to the 1D-NN, which proves the efficacy of this method.

Effectiveness of labeling smoothing using GGO

To evaluate the usefulness of smoothing, we tested a graph-based regularization for obtaining optimal global results. Here, the regularization strength was set to 1.0. The initial and optimized classification results are presented as a comparison in Table 7.9. It can be seen that the implementation of GGO improves the OA and the $AvgF_1$ by 0.2% and 0.6%, respectively. Although the overall performance does not show remarkable improvements, the influence of GGO is clear in some specific classes such as powerlines, facades, and cars. In Fig. 7.17, the detailed visual illustration indicates an improvement in both smoothness and classification performance with global contextual information. As aforementioned, through a powerful feature representation method, most areas in the urban scenes such as roofs and high vegetation are correctly labeled. Most powerlines, whose density is relatively sparse, are labeled correctly as well. A small amount of noise (i.e., wrongly labeled points) still presents in the classification results. In the results of GGO, the wrongly labeled points can be corrected based on the contextual information. However, although the use of the optimization procedure corrected the labels of those wrongly classified points, the change in the statistic of the classification accuracies seems not apparent. One possible explanation is that our feature engineering strategy has already provided contextual properties with high quality, especially the implementation of the multi-scale strategy. Thus, the initial classification output is already of high accuracy, so that the graph-based optimization can merely serve as a smoothing.

Table 7.9: Comparison between initial and smoothed classification results using the ISPRS benchmark dataset (All values are in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	power	low_veg	imp_surf	car	fence_hedge	roof	fac	shrub	tree	OA	$AvgF_1$
Initial	66.1	86.4	**99.4**	74.1	**20.5**	90.9	41.9	39.2	72.2	83.0	65.6
Smoothed	**68.1**	**86.5**	99.3	**75.2**	19.5	**91.1**	**44.2**	**39.4**	**72.6**	**83.2**	**66.2**

Classification results of the AHN3 dataset

In addition to the ISPRS benchmark dataset, we obtained the point-wise classification results using the AHN3 dataset to further investigate the versatility of the DPE method on a large-scale ALS point cloud dataset. The results are listed Tables 7.10 and 7.11 and Figs. 7.18 and 7.19.

For the part of deep feature extraction, hyperparameters of the training process are the same as the setting given in the description of the processing of the ISPRS benchmark dataset. For the feature embedding part, the 384-dimensional deep features were embedded into a 30-dimensional feature vector with the neighborhood size set as 20. The number of trees used in the RF classifier is 100. As for the graph-based optimization, the regularization strength is set as 1.0.

As seen in Table 7.10, JME outperforms the other feature extraction methods with the AHN3 dataset. Compared to the baseline provided by PointNet++, after the hierarchical subdivision process and the embedding of deep features, the OA and the $AvgF_1$ are incremented by 5.1% and 8.2%, respectively. Effectiveness of JME in providing a good representation of deep features is

Table 7.10: Comparison of different feature learning methods using the AHN3 dataset (All values are in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	Ground surfaces	Buildings	Water	Artificial objects	Other	OA	$AvgF_1$
SDF	90.6	85.8	78.9	34.3	84.0	86.0	74.7
MDF	93.2	87.5	93.3	41.8	86.0	89.8	80.4
JME	**94.2**	**89.1**	**95.3**	**49.1**	**86.9**	**91.1**	**82.9**

Table 7.11: Comparison between the initial and smoothed classification results using AHN3 dataset (All values are in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	Ground surfaces	Buildings	Water	Artificial objects	Other	OA	$AvgF_1$
Initial	**94.2**	89.1	95.3	49.1	**86.9**	91.1	82.9
Smoothed	94.1	**89.8**	**96.3**	**52.2**	86.2	**91.2**	**83.7**

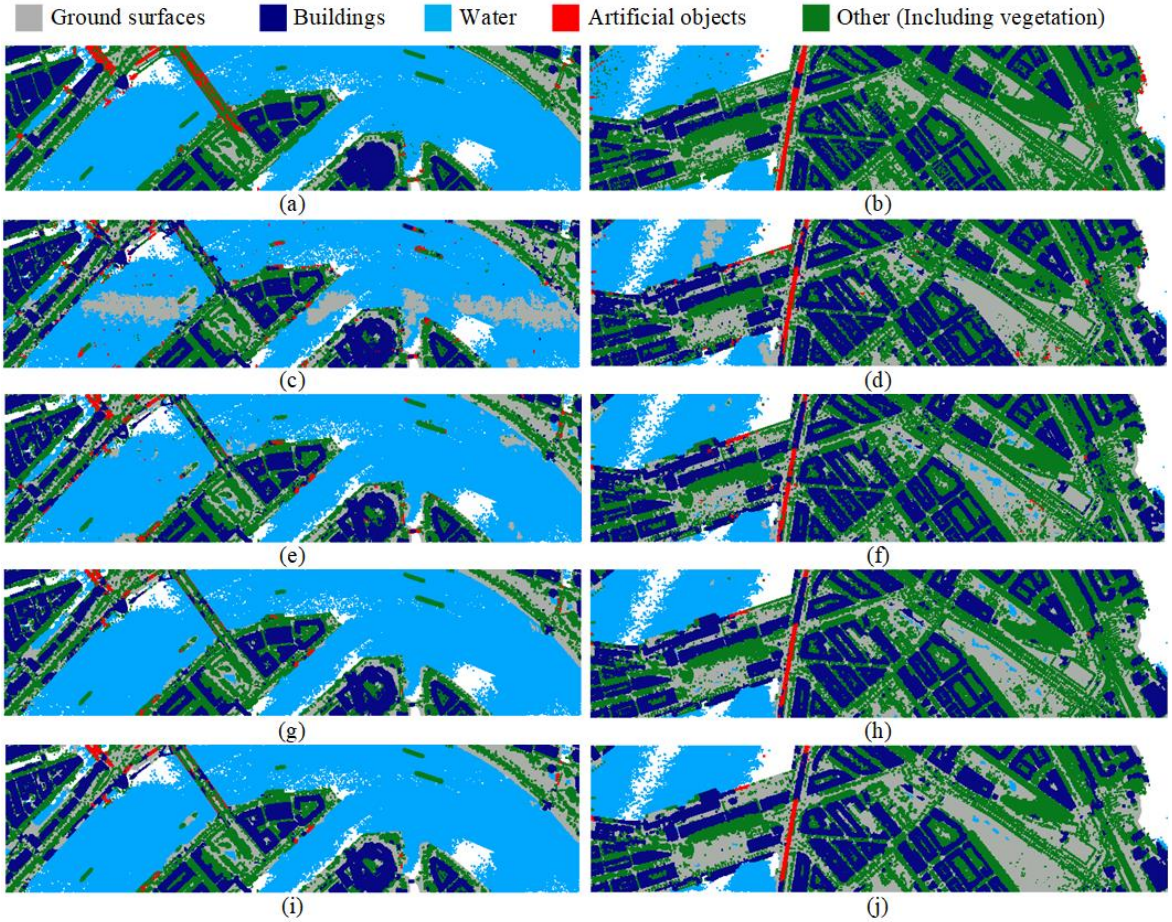

Figure 7.18: Classification results of the AHN3 dataset. a) - b) Ground truth of test areas, c) - d) classification results with SDF, e) - f) classification results with MDF, g) - h) classification results with JME, i) - j) the smoothed classification results.

clearly demonstrated by the results. Additionally, as seen in Fig. 7.18, with the embedded features, most areas of our interest such as buildings, ground surfaces, and water can be correctly classified. Compared to the results with PointNet++ that are denoted as SDF, some large uncorrected

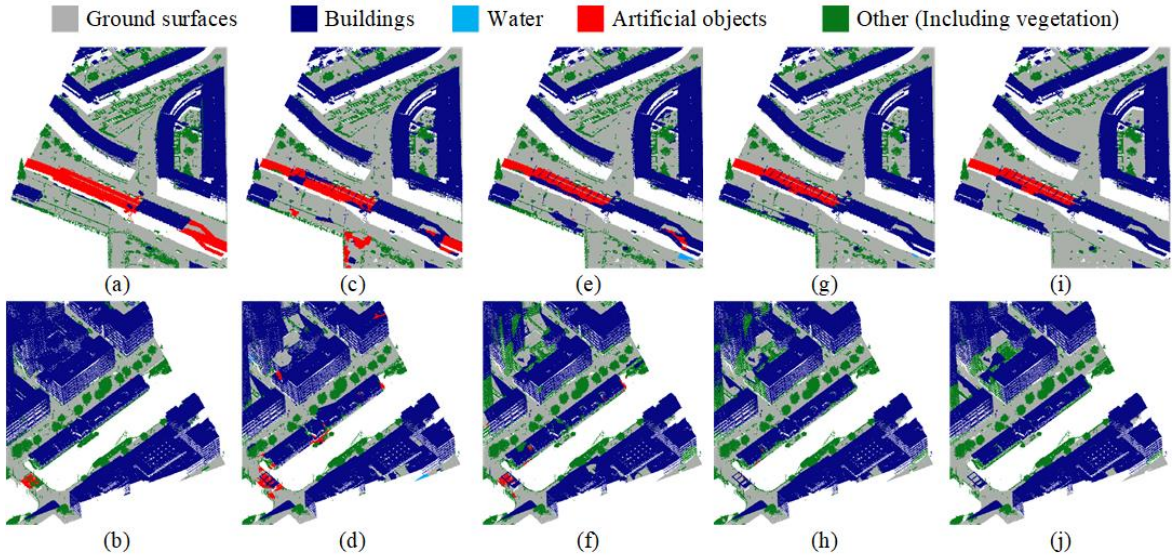

Figure 7.19: Details of the classification results with the AHN3 dataset for comparison. a) - b) Ground truth of test areas, c) - d) classification results with SDF, e) - f) classification results with MDF, g) - h) classification results with JME, i) - j) the smoothed classification results.

labeled areas can be refined with MDF; this is obvious for some areas covered by water. The classification maps shown in Fig. 7.18 and details shown in Fig. 7.19 indicate the remarkable improvement with the utilization of feature embedding.

Regarding the classification results after labeling smoothing, although the improvement is not significant compared to that provided by the feature engineering part, the $AvgF_1$ still increases by 0.8% as shown in Table 7.11.

Sensitivity analysis of parameters in DPE

Influence of different hyperparameters on embedding performance

To effectively obtain optimal feature representation, high-dimensional features are embedded into a reduced dimensional feature space by using JME; however, the performance of JME is highly dependent on certain hyperparameters. To investigate the influence of the hyperparameters on the embedding performance, we assessed the sensitivity of these parameters by altering their values in reasonable ranges. Two key parameters, number of neighbors and dimensionality, were tested in the experiments.

As seen in Fig. 7.20, the sensitivity of these parameters on the performance of JME is evident to some extent. Generally, the performance of JME improves with increasing dimensionality at an early stage. Then, the classification accuracy reaches the peak when the dimensionality is approximately 50. Furthermore, classification results remain almost stable when the reduced dimensionality continues to increase; this can be explained as: the 50-dimensionality embedded features are sufficient to provide a satisfying representation of the classification task. It should be noted that owing to the biased distribution of different classes in the test areas, the improvement of classification accuracy can be better presented by the $AvgF_1$ score. By contrast to the case of the parameter, reduced dimensionality, JME seems to be more stable in facing the variation of the other parameter, the number of selected neighbors. As seen in Fig. 7.20, the classification accuracy decreases with a very small margin when the number of neighbors increases. This can be

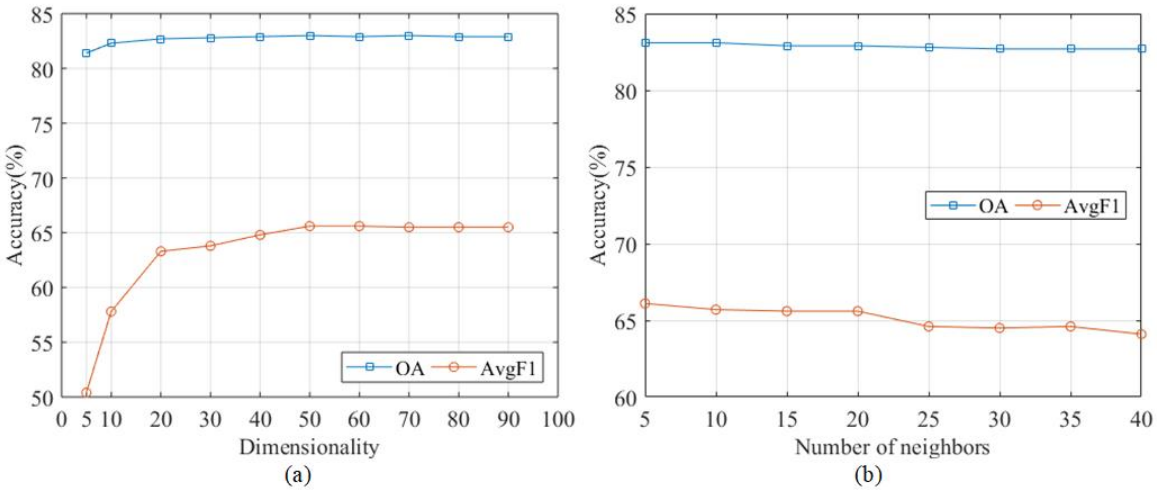

Figure 7.20: Sensitivity analysis of JME on two hyperparameters: a) reduced dimensionality, b) number of neighbors.

attributed to the reason that the local data structure may be concealed by a large neighborhood. In typical cases, the optimal number of neighbors is neither too large nor too small because the local data structure cannot be efficiently represented by a small number of neighbors. However, in our case, the results reach a peak at an early stage. The application of the k-clustering strategy in the embedding process may be the reason for the same. The optimal parameter values utilized in the classification of the ISPRS benchmark dataset were determined to be 50 and 5 for the dimensionality and the number of neighbors, respectively.

Effect of regularization strength on the performance of GGO

GGO is achieved by minimizing the cost function that is a weighted sum of a local smoothness term and a fidelity term. We tested different values for the regularization strength. As shown in Fig. 7.21, generally, the regularization strength slightly affects the classification accuracies, and its effect on $AvgF_1$ is more pronounced than that on OA. Furthermore, the classification accuracies including both OA and $AvgF_1$ reach their best values when the regularization strength is set to 1.0.

7.2.3 Semantically segmented urban scenes using GraNet

The performance of GraNet was tested on three benchmark dataset, consisting of the ISPRS benchmark dataset, the LASDU dataset, and the DALES dataset. The ablation study was conducted on the GraNet. Moreover, the result of GraNet is compared with start-of-the-art methods, also including the DPE method.

Classification results of the ISPRS benchmark dataset

Comparing with other PointNet-based methods

Based on the obtained classification results, we compare the results of GraNet with the other four methods, including PointNet [Qi et al., 2017a], PointNet++ [Qi et al., 2017b], Hierarchical Data Augmented PointNet++ (HDA-PointNet++) [Huang et al., 2019], and PointSIFT [Jiang et al., 2018].

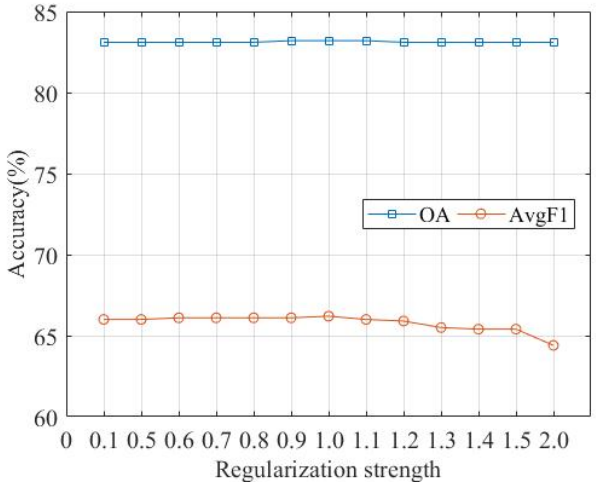

Figure 7.21: Sensitivity analysis of GGO on the regularization strength.

Table 7.12 lists the classification results of the four aforementioned methods and the GraNet method, which are all under the framework of the multi-scale network structure. As shown by the table, GraNet can outperform the other four reference PointNet++ based methods. In respect of the OA, we can find that GraNet can achieve the best results with an OA of 84.5%. First, it can be seen that PointNet shows lower ability when dealing with this kind of fine-grained situation, especially when considering the classification accuracy of some small objects, such as powerlines, cars, and fence hedges. Compare to the baseline method for our strategy, PointNet++, the OA increases by 5.4%. Additionally, compared with the improved solution, PointSIFT, our method also achieve better results with an increment of OA by 1.8% and $AvgF_1$ by 5.7%. It indicates the effectiveness of the local spatial encoding method and relation-aware strategy in GraNet compared with purely encoding local neighborhood and orientation information with MLP. Furthermore, GraNet also achieves higher classification accuracy for most categories than the other PointNet++ based methods. It is also worth mentioning that HDA-PointNet++ shows the best results in some large-scale categories, such as low vegetation and impervious surface, compared with the other methods, although it does not provide competitive OA and $AvgF_1$. The main reason is that for deep neural networks directly handling describe points, the subdivision and sampling of the entire point clouds in urban scenarios is a critical step that directly impacts the classification results. By the use of the multi-scale subdivision and sampling, the scale information can be considered, especially for the objects with larger-scale dependencies.

Table 7.12: Comparing of the GraNet method and different PointNet++ based methods using the ISPRS benchmark dataset. (Values in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	Power	Low_veg	Imp_surf	Car	Fence_hedge	Roof	Fac	Shrub	Tree	OA	$AvgF_1$
PointNet [Qi et al., 2017a]	0.0	77.1	92.6	0.0	0.0	77.6	5.4	28.5	37.0	71.2	69.3
PointNet++ [Qi et al., 2017b]	58.5	81.4	98.1	55.3	19.2	86.9	41.0	33.2	71.2	79.1	60.5
HDA-PointNet++ [Huang et al., 2020b]	64.2	**85.1**	**99.2**	68.9	19.2	88.2	36.5	37.7	69.2	81.2	63.1
PointSIFT [Jiang et al., 2018]	50.6	80.3	91.3	74.9	38.4	92.9	59.2	41.8	81.4	82.7	67.9
GraNet	**67.7**	82.7	91.7	**80.9**	**51.1**	**94.5**	**62.0**	**49.9**	**82.0**	**84.5**	**73.6**

In Fig. 7.22, we provide the classification map obtained using GraNet. From the figure, we can see that GraNet shows good ability in recognizing complicated and fine-grained patterns. It performs well in classifying small objects, such as powerlines, cars, and, fences. However, it showed deficiency in distinguishing the boundaries between facades and buildings. For better comparison

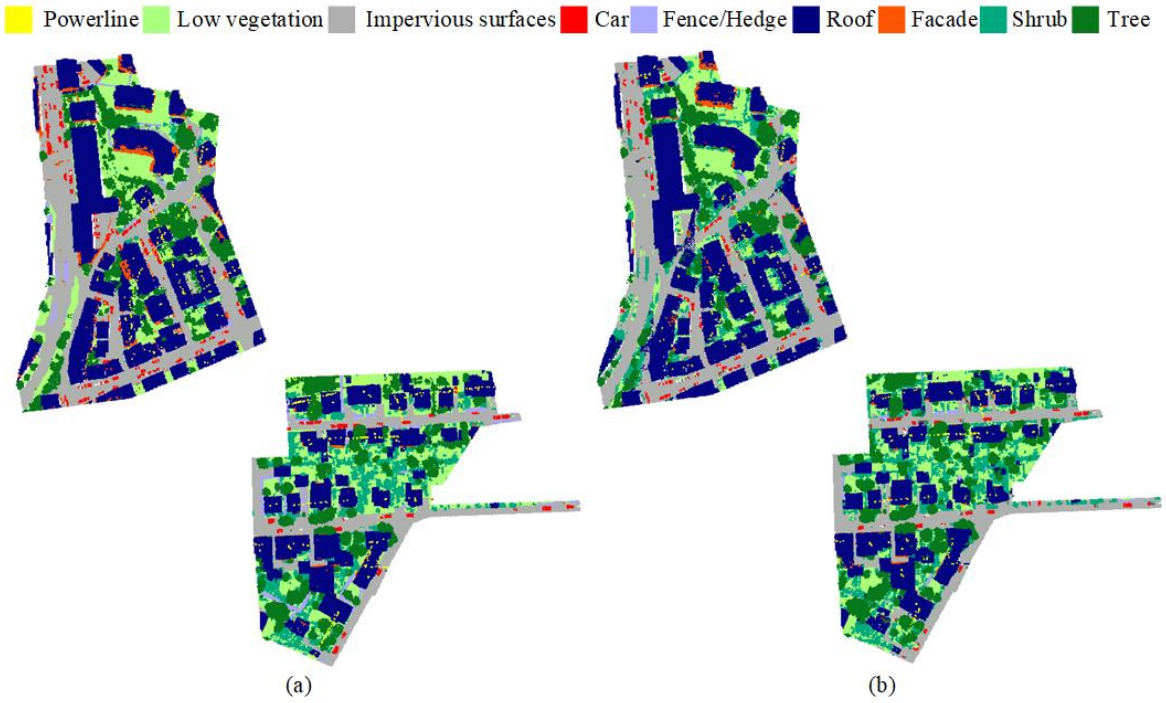

Figure 7.22: Classification results of the ISPRS benchmark dataset. a) Ground truth, b) GraNet.

with baseline method PointSIFT, we also provide some details for the classification results in Fig. 7.23. As shown in the figure, PointSIFT has a strong ability to distinguish details since it constructs local orientation features in different scales. It also performs well in persevering boundaries between objects. However, in a larger scope, it may cause errors. For example, it wrongly recognized buildings as low vegetation due to similar local geometric characteristics. However, for the GraNet method, there are fewer misclassified points between categories. Meanwhile, it is obvious that our method works better in classifying powerlines that are sparsely distributed in the dataset.

Error map of classification results

Fig. 7.24 illustrates the error map of the classification results using the GraNet method on the ISPRS benchmark dataset. From the figure, it can be seen that the majority of 3D points can be assigned with correct labels. However, there are still some errors in the classification maps. With reference to the ground truth provided in Figs. 7.15, most of the errors lie in the boundaries between buildings and facades and also the boundaries between vegetation, including low vegetation, tree, and shrub. These parts do indeed share similar geometric characteristics, and the local neighborhood is also similar when the point cloud is sparse. Additionally, for the classification of vegetation, the lack of sufficient spectral information may also be one reason that it is hard to distinguish these three categories, especially when the geometric attributes are similar.

Comparing with results from other published methods

For further evaluation, we also compare the GraNet method with other published methods achieving baseline results, which can be regarded as reference methods, including LUH [Niemeyer et al., 2016], NANJ2 [Zhao et al., 2018], WhuY4 [Yang et al., 2018], RIT_1 [Yousefhussien et al., 2018], DPE, and a geometry attentional network (GANet) [Li et al., 2020a], DANCE-NET [Li

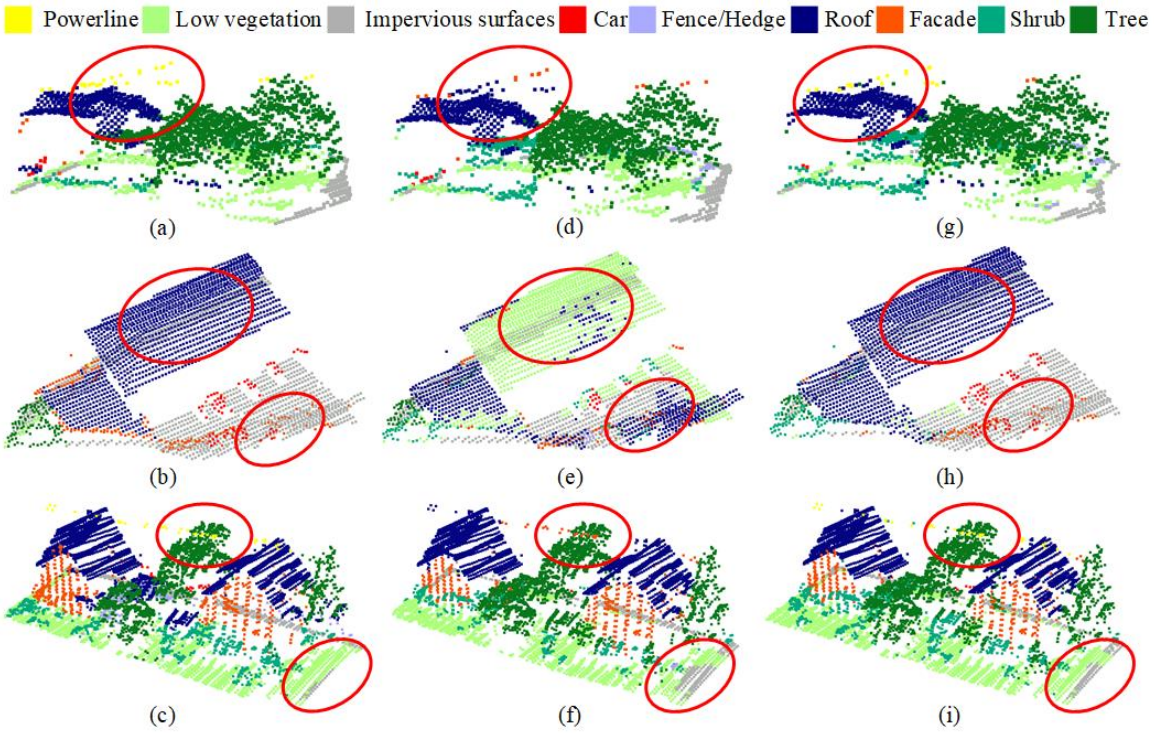

Figure 7.23: Details of classification results of the ISPRS benchmark dataset. a) - c) Ground truth, d) - f) PointSIFT, g) - i) GraNet.

et al., 2020b], a directionally constrained fully convolutional neural network (D-FCN) [Wen et al., 2020], KPConv [Thomas et al., 2019], and RandLA-Net [Hu et al., 2020]. The published results of these methods can be checked from the ISPRS 3D semantic labeling dataset website*, published papers, or based on the published codes. In Table 7.13, a comparison of the classification results obtained by these methods is provided and displayed with the aforementioned evaluation metrics. LUH utilized a two-layer hierarchical CRF. For these two layers, one layer operated directly on points, while the other operated on generated segments using manual features. Apart from LUH, all the other methods are deep learning-based methods. Noted that not all the abovementioned methods directly handle the discrete points. NANJ2 and WUY4 methods rely on 2D deep learning strategies. The NANJ2 method utilized a 2D CNN to predict labels of ALS point clouds in an urban scenario by learning depth features of multiple scales with several selected attributes, including height, intensity, roughness, and a color vector. The WUY4 method had a similar strategy as that uses in the NANJ2 method, wherein the pixel-wise features, generated based on the geometric attribution of 3D points, were fed into a 2D CNN to obtain the classification results. Unlike 2D-based methods, RIT_1 was implemented directly on 3D points by using a proposed multi-scale 1D fully convolutional architecture. Since the other deep-learning-based methods have been introduced in Section 1.2.2, we do not give further explanations on these methods. When compared with other aforementioned methods, one major strength of GraNet is that this method ranks first among all the listed state-of-the-art methods in terms of $AvgF_1$ of 73.6%. Its OA is the same as the value of GANet, but its $AvgF_1$ is 0.4% higher than GANet. Besides, compared to NANJ2 and WhuY4, although we did not achieve higher OA, the $AvgF_1$ of our method is higher by 4.2%, and 4.3%, respectively. Additionally, the GraNet method is an end-to-end method that directly works on 3D points compared with these two methods. Compared with the other 3D

*http://www2.isprs.org/commissions/comm2/wg4/vaihingen-3d-semantic-labeling.html

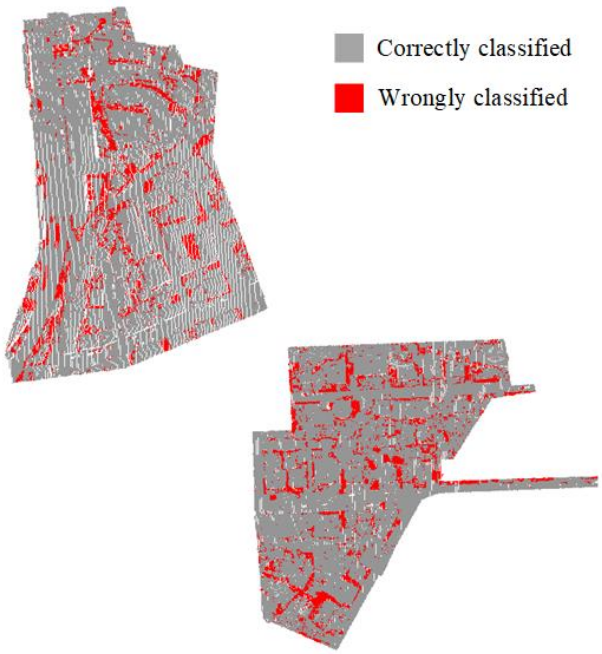

Figure 7.24: The error map of the classification results of the ISPRS benchmark dataset using GraNet.

Table 7.13: Comparison of the GraNet method and other published methods using the ISPRS benchmark dataset (Values in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	Power	Low_veg	Imp_surf	Car	Fence_hedge	Roof	Fac	Shrub	Tree	OA	$AvgF_1$
LUH [Niemeyer et al., 2016]	59.6	77.5	91.1	73.1	34.0	94.2	56.3	46.6	83.1	81.6	68.4
NANJ2 [Zhao et al., 2018]	62.0	**88.8**	91.2	66.7	40.7	93.6	42.6	**55.9**	**82.6**	**85.2**	69.3
WhuY4 [Yang et al., 2018]	42.5	82.7	91.4	74.7	**53.7**	94.3	53.1	47.9	82.8	84.9	69.2
RIT_1 [Yousefhussien et al., 2018]	37.5	77.9	91.5	73.4	18.0	94.0	49.3	45.9	82.5	81.6	63.3
DPE [Huang et al., 2020b]	68.1	86.5	**99.3**	75.2	19.5	91.1	44.2	39.4	72.6	83.2	66.2
GANet [Li et al., 2020a]	**75.4**	82.0	91.6	77.8	44.2	94.4	61.5	49.6	**82.6**	84.5	73.2
DANCE-NET [Li et al., 2020b]	68.4	81.6	92.8	77.2	38.6	93.9	60.2	47.2	81.4	83.9	71.2
D-FCN [Wen et al., 2020]	70.4	80.2	91.4	78.1	37.0	93.0	60.5	46.0	79.4	82.2	70.7
KPConv [Thomas et al., 2019]	73.5	78.7	88.0	79.4	33.0	94.2	61.3	45.7	82.0	81.7	70.6
RandLANet [Hu et al., 2020]	68.8	82.1	91.3	76.6	43.8	91.1	61.9	45.2	77.4	82.1	70.9
GraNet	67.7	82.7	91.7	**80.9**	51.1	**94.5**	**62.0**	49.9	82.0	84.5	**73.6**

methods, our method has a higher OA and $AvgF_1$. It is worth mentioning that KPConv provides higher classification accuracies when classifying powerlines and trees compared with our method. For powerlines that are sparsely distributed on the scene, it shows its adaptiveness to various densities. As for the RandLA-Net method, the major advantage is its efficiency. It is much faster compared to our method.

Classification results of the LASDU dataset

The same as the comparison in the evaluation using the ISPRS benchmark dataset, we also compare the results with the other method under the framework of PointNet++. Table 7.14 lists the classification results of four different PointNet++ based methods, consisting of PointNet, PointNet++, HDA-PointNet++, and PointSIFT. Since the urban scene presented in the LASDU dataset is less complicated than the ISPRS benchmark dataset, the performance of PointNet is slightly better. At least, most urban objects can be distinguished. Compared to the baseline method, PointNet++, using the proposed method, OA and $AvgF_1$ are incremented by 3.5% and

4.7%. By stacking features from multiple scales, recognizing small objects and distinguishing objects from different scales can be strengthened using HDA-PointNet++. However, compared to the GraNet method, OA and $AvgF_1$ are still lower by 1.9% and 2.6%. Compared to PointSIFT, which considers the orientation and scale information, GraNet still performs better by considering additional elevation information and long-range dependencies. Additionally, when it comes to the classification accuracies, it can be seen from the table that the GraNet method achieves the best results in most categories among the aforementioned methods. As shown in Figs. 7.25, using

Table 7.14: Comparing of the GraNet method and different PointNet++ based methods using the LASDU dataset (all values are in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	Artifacts	Buildings	Ground	Low_veg	Trees	OA	AvgF_1
PointNet [Qi et al., 2017a]	13.2	86.2	86.3	51.0	59.9	77.5	59.3
PointNet++ [Qi et al., 2017b]	31.3	90.6	87.7	63.2	82.0	82.8	71.0
HDA-PointNet++ [Huang et al., 2020b]	36.9	93.2	88.7	**65.2**	82.2	84.4	73.2
PointSIFT [Jiang et al., 2018]	38.0	94.3	88.8	64.4	85.5	84.9	74.2
GraNet	**42.4**	**95.8**	**89.9**	64.7	**86.1**	**86.2**	**75.8**

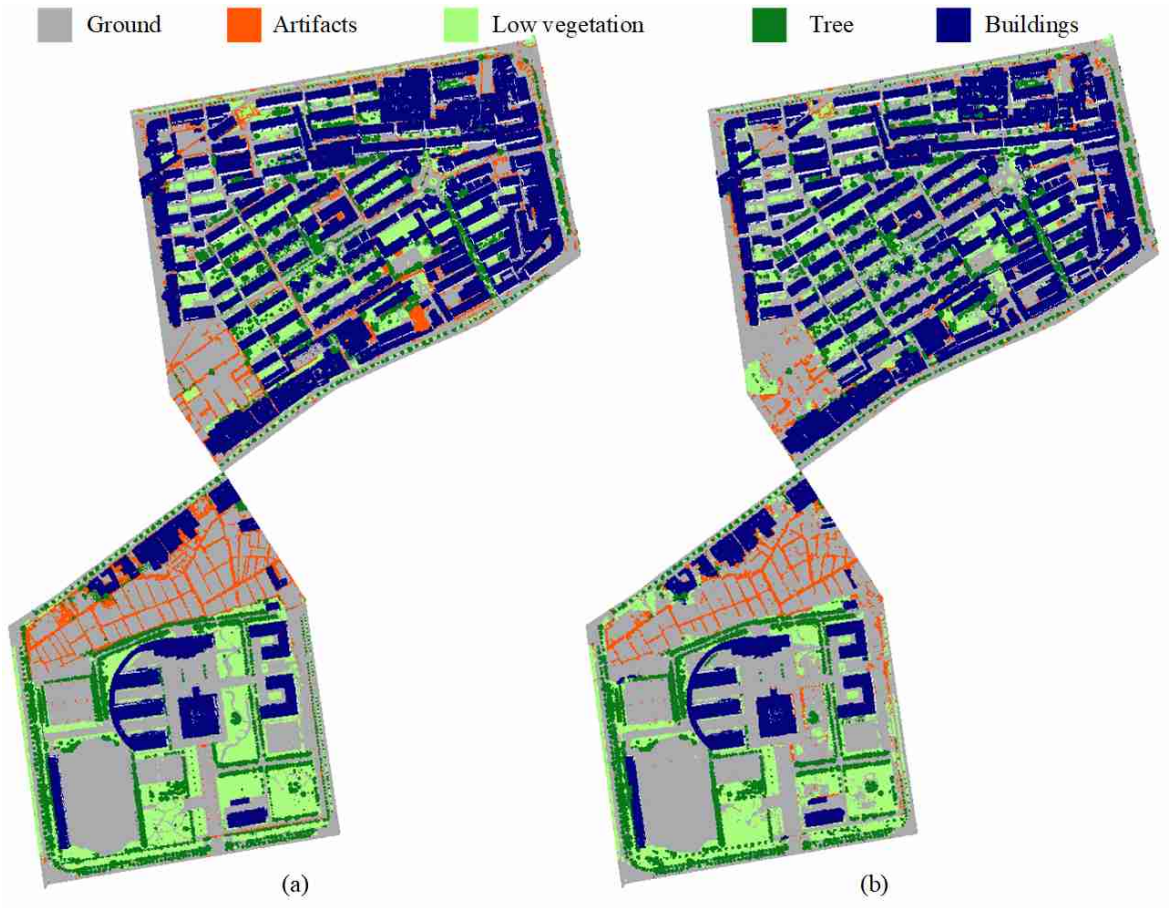

Figure 7.25: Classification results of the LASDU dataset using GraNet. a) Ground truth, b) GraNet.

the GraNet method, most areas of our interest, such as buildings, artifacts, and vegetation, can be correctly classified. It is clear that the GraNet method performs well in terms of recognizing artifacts and preserving their boundaries between the ground. In Fig. 7.26, a detailed comparison with the baseline method PointSIFT is also provided. From the figure, we can see that our method

performs better in classifying the boundary points between buildings and ground. Additionally, for the building roof which shows the different local structure, our method performs better than PointSIFT with long-range constraint. However, both the GraNet method and PointSIFT meet some problems in classifying low vegetation and ground for some specific areas.

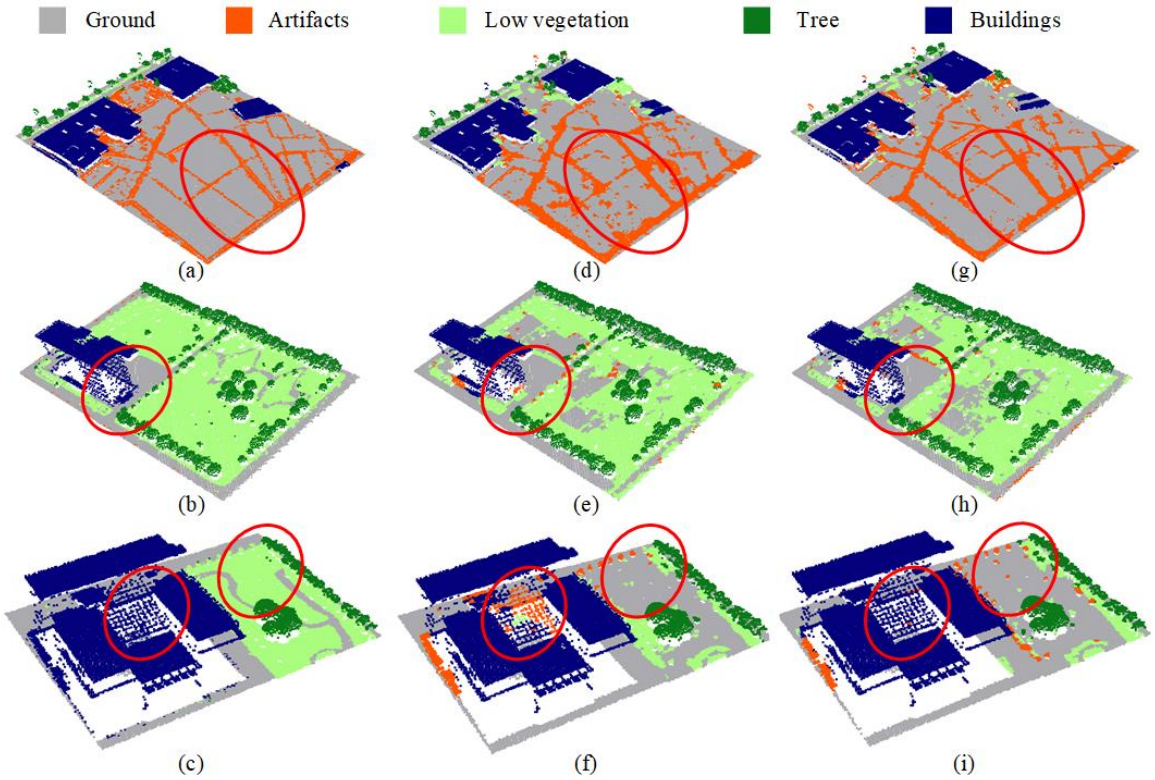

Figure 7.26: Details of classification results of the LASDU dataset using GraNet. a) - c) Ground truth, d) - f) PointSIFT, c) - i) GraNet.

Classification results of the DALES dataset

Since the results of some start-of-the-art methods on the DALES dataset, we directly compare the results using GraNet with these baseline methods. Table 7.15 displays the classification results of six baseline methods, containing KPConv, Pointnet++, ConvPoint, SuperPoint, PointCNN, and ShellNet. These methods are all deep-learning-based ones, which have been introduced in Section 1.2.2. From the results, it can be seen that the KPConv architecture has a stronger performance on the DALES dataset with the highest OA and $mIoU$. The GraNet method also achieves satisfying classification results, which ranks second compared with the other start-of-the-art deep-learning methods. The classification maps using the GraNet method are shown in Fig. 7.27, which shows that the GraNet method can classify the majority of points from various categories correctly. In Fig. 7.28, details of the classification results are illustrated. From the result, we can see that there are both large batches with low contrast, which were wrongly classified. One reason may be the selection of block size. Although the GraNet method tends to learn long-range dependencies from other points globally, the connections are only limited inside the bounding box of the block size. For large-scale datasets, a small block size would be enough to obtain contextual information for small objects correctly. However, for large objects, small blocks are not efficient enough for providing important contextual information. On the other hand, a large block size will increase memory and run time. For the GraNet method, the memory

increases quadratically. This is the main drawback of this method. The difference between the KPConv architecture and other methods (except for Superpoint Graphs) is that KPConv did not rely on selecting a fixed number of points within a bounding box. This may also be the reason why KPConv performed better on the DALES dataset.

Table 7.15: Comparison of the GraNet method and different baseline methods using the DALES dataset (all values are in %). Noted that the highest values in OA, $mIoU$, and IoU for each category are marked with bold texts.

Methods	Ground	Buildings	Cars	Trucks	Poles	Powerlines	fences	Vegetation	OA	$mIoU$
KPConv [Thomas et al., 2019]	97.1	**96.6**	**85.3**	**41.9**	75.0	**95.5**	63.5	94.1	**97.8**	**81.1**
PointNet++ [Qi et al., 2017b]	94.1	89.1	75.4	30.3	40.0	79.9	46.2	91.2	95.7	68.3
ConvPoint [Boulch, 2020]	96.9	96.3	75.5	21.7	40.3	86.7	29.6	91.9	97.2	67.4
SuperPoint [Landrieu & Simonovsky, 2018]	94.7	93.4	62.9	18.7	28.5	65.2	33.6	87.9	95.5	60.6
PointCNN [Li et al., 2018]	**97.5**	95.7	40.6	4.8	57.6	26.7	52.6	91.7	97.2	58.4
ShellNet [Zhang et al., 2019a]	96.0	95.4	32.2	39.6	20.0	27.4	60.0	88.4	96.4	57.4
GraNet	96.3	93.3	80.6	40.8	**91.8**	61.1	**65.6**	**94.5**	97.3	78.0

Ablation study of GraNet

In order to investigate the effectiveness of each individual module in the GraNet method, we conduct the following ablation studies. All the experiments were conducted on the ISPRS benchmark dataset. The detailed design of the ablation studies and the evaluation results are provided in the following.

Effectiveness of the LoSDA module

To vindicate the LoSDA module and explore the effectiveness of each module in LoSDA, we trained five models using the same network architecture with different local spatial encoding modules, namely only SDE (model **A**), only DFE (model **B**), the combination of SDE and DFE (model **C**), the combination of SDE, DFE and EDE (model **D**), the full local spatial encoding module using the final combination of SDE, DFE, and EDE with attention pooling (model **E**). All the experimental results are listed in Table 7.16.

Table 7.16: Comparison of models with different local spatial encoding methods using the ISPRS benchmark dataset (Values in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	Power	Low_veg	Imp_surf	Car	Fence_hedge	Roof	Fac	Shrub	Tree	OA	$AvgF_1$
A	49.8	81.5	90.5	76.8	38.4	91.6	54.5	**47.8**	78.0	81.8	67.6
B	50.6	80.3	91.3	74.9	38.4	92.9	59.2	41.8	**81.4**	82.7	67.9
C	49.5	82.2	91.7	73.6	43.2	93.8	60.6	45.9	80.0	83.6	68.9
D	**60.8**	**82.4**	91.2	78.8	41.9	93.8	62.7	46.0	**81.4**	83.9	71.0
E	60.1	82.1	**92.1**	**80.2**	**46.2**	**94.1**	**63.1**	47.7	81.1	**84.1**	**71.9**

Results in Table 7.16 show that directional features can better present the local geometry compared with features based on local distribution. However, since the SDE and the DFE module actually describe the local geometry from different perspectives, we further conduct experiments to test the effectiveness of combining these two local feature embedding modules. We can see from the results that the classification results are further improved compared with the results using SDE and DFE individually. Additionally, for the ALS point clouds, elevations of points play an essential role. It is shown in the table that model **D** has a higher OA by 0.3% and a higher $AvgF_1$ by 2.1%, compared with model **C**. The results prove that the elevation-based module benefits the description of local geometry. The dependencies between local points are further investigated by

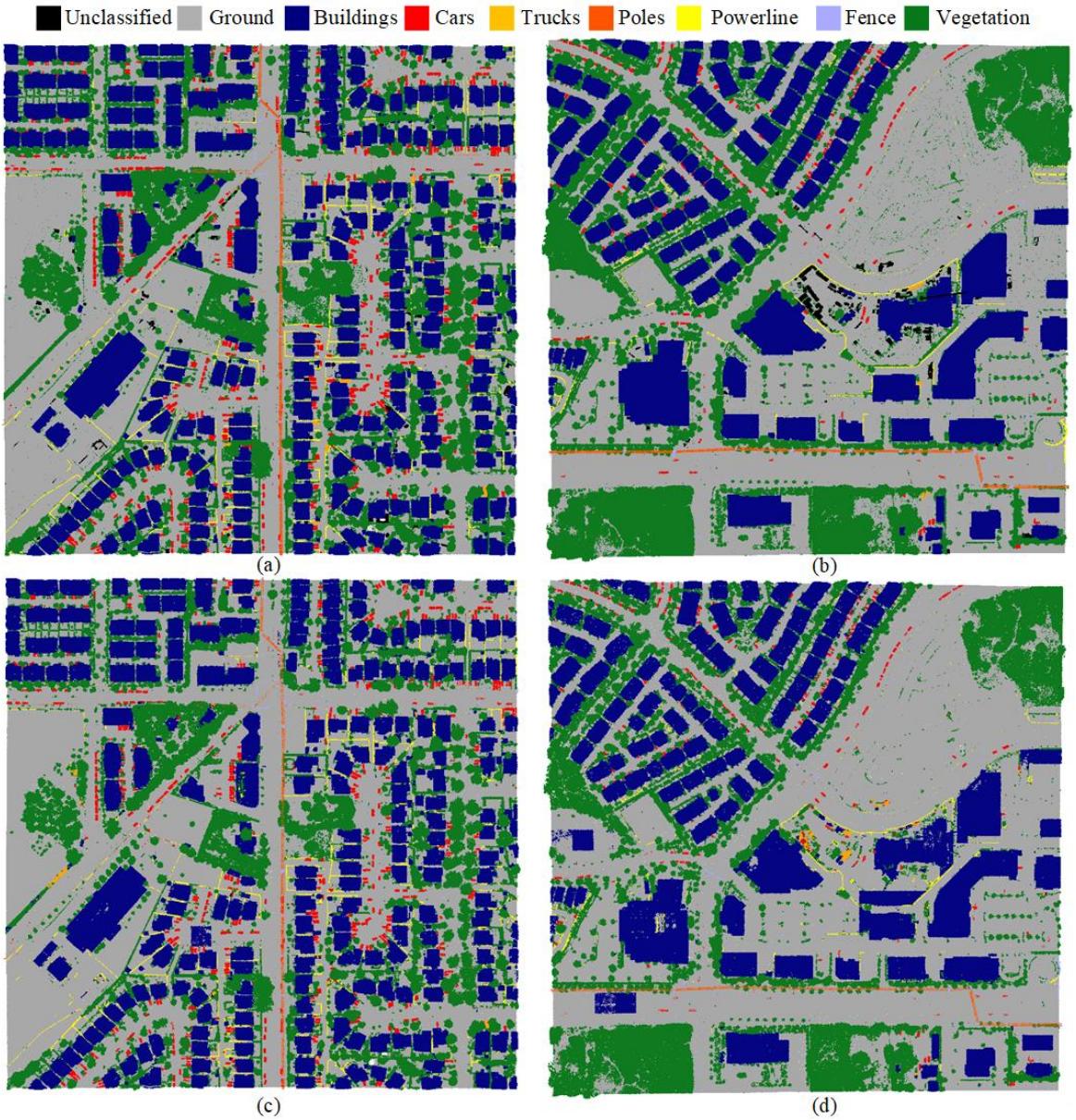

Figure 7.27: Classification results of the DALES dataset using GraNet. Only two selected tiles are shown. a) - b) Ground truth, c) - d) the classification maps using GraNet.

adding the attention pooling module to the current local spatial encoding module. As shown by the results, the classification accuracy is further improved. Compared with different combinations of each module in the LoSDA module, we can see that each module is useful in learning local feature descriptions. Moreover, compared with the baseline method PointSIFT, the local spatial encoding module in the GraNet method can better encode the geometric structures in the local scope and is efficient for both large-scale and small-scale categories.

Effectiveness of the GRA module and different modes

In order to validate the effectiveness of the GRA module and compare the different combination modes of SRA and CRA, we conducted further experiments on the ISPRS benchmark dataset. Namely, based on the LoSDA module and the baseline architecture, we add different modes of

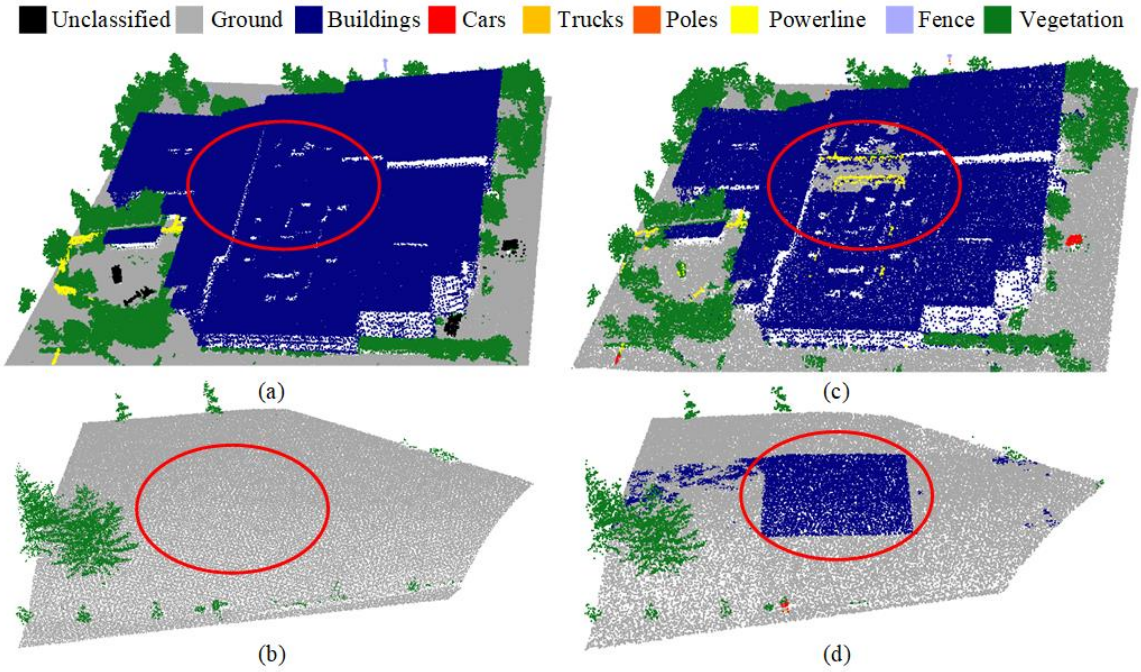

Figure 7.28: Details of classification results of the DALES dataset using GraNet. Only two selected examples are shown. a) - b) ground truth, c) - d) GraNet.

the GRA module to the network and evaluate the results. It should be mentioned that here the baseline method refers to the network which uses the LoSDA module for local spatial encoding.

Table 7.17: Comparison of models with different configurations of the GRA module using the ISPRS benchmark dataset (Values in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	Power	Low_veg	Imp_surf	Car	Fence_hedge	Roof	Fac	Shrub	Tree	OA	$AvgF_1$
Non GRA	60.1	82.1	92.1	80.2	46.2	94.1	63.1	47.7	81.1	84.1	71.9
Only CRA	62.8	82.3	91.5	79.6	49.4	94.4	62.5	48.4	81.1	84.1	72.4
Only SRA	61.8	82.8	**92.2**	74.7	49.4	94.2	61.8	**52.2**	81.7	84.4	72.3
Model 1	**67.7**	82.7	91.7	**80.9**	**51.1**	**94.5**	62.0	49.9	**82.0**	**84.5**	**73.6**
Model 2	66.8	**82.9**	91.9	80.4	47.6	94.1	62.7	48.4	81.2	84.3	72.9
Model 3	67.5	83.0	92.5	80.7	48.2	94.3	**63.8**	50.0	80.6	83.3	73.4

All the results using different configurations of the GRA modules are listed in Table 7.17. As can be seen from the table, it is clear that the relation-aware attentional module can provide an improvement compared with the baseline method. In detail, by adding the CRA module, the $AvgF_1$ experiences an increase by 0.5%. The use of the SRA module produces an increment of OA by 0.3% and $AvgF_1$ by 0.4%. Besides, by combining the SRA and CRA modules, the performance of our proposed method is further augmented. By using the configuration of Mode 1, the results are improved with an increase of OA by 0.4% and $AvgF_1$ by 1.7%. Mode 2 and Mode 3 also provide improvement for the classification results. It should be mentioned that Mode 1 outperforms all the other configurations of the GRA module, producing the highest OA with 84.5% and $AvgF_1$ with 73.6%. In Fig. 7.29, a detailed illustration of the classification results using the GraNet and the network without the GRA module is provided. Here, model 1 was utilized. By using the effective local spatial encoding module, most of the area has been correctly classified, including the roofs, impervious surfaces, and trees. By stacking the GRA module, some

small areas are corrected, such as cars and part of building roofs. However, even using the GRA module, it is still hard to correctly distinguish fences with shrubs and vegetation.

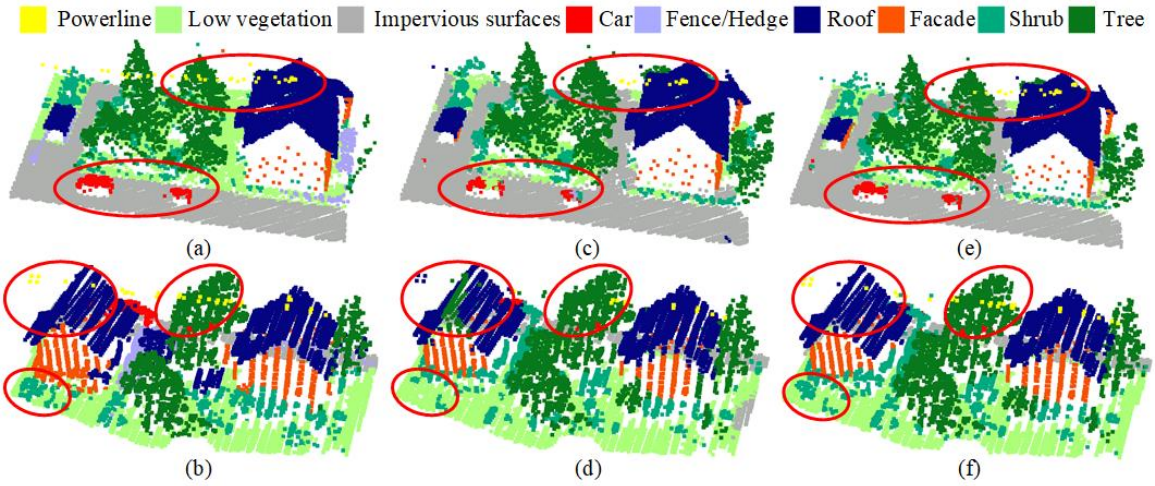

Figure 7.29: Classification results with the GRA module and without the GRA module. a) - b) Ground truth, c) - d) the classification result using the model without GRA, e) - f) the classification result using the model with GRA (Model 1).

Sensitivity analysis of input block size of the network

For fulfilling the input requirement, the entire point clouds are divided into small blocks. However, the number of parameters of the network depends on the input size. To investigate the influence of the input block size on the classification results, we added further experiments by improving the input block size to 40 $m \times 40$ m. Correspondingly, the input size was changed to 8912 points. Table 7.18 list the classification results using two different block sizes. From the table, we can see that the OA increases by 0.1% but the $AvgF_1$ decreases by 0.4%. The decrease is mainly due to the decrease in the classification accuracy of powerlines. However, the classification accuracies increase for some comparatively large objects, i.e., low vegetation, impervious surfaces, trees. Thus, although the overall performance does not change by a large margin, it can be seen that a larger block size will improve the classification performance on large objects. However, the small objects will have a low classification accuracy.

Table 7.18: Comparison of results using different input block sizes using the ISPRS benchmark dataset (Values in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Methods	Power	Low_veg	Imp_surf	Car	Fence_hedge	Roof	Fac	Shrub	Tree	OA	$AvgF_1$
25 $m \times 25$ m	**67.7**	82.7	91.7	80.9	51.1	94.5	**62.0**	**49.9**	82.0	84.5	**73.6**
40 $m \times 40$ m	64.0	**83.1**	**91.9**	**81.1**	**52.2**	**94.6**	60.3	49.6	**82.3**	**84.6**	73.2

Complexity and runtime analysis

To evaluate the complexity and training efficiency of GraNet, we list the numbers of parameters (params) and running time of eight networks in Table 7.19, including two baseline methods, the combination of PointNet++ structure and the LoSDA module, and five different modes of GraNet. The input size for all networks is set to 4096. It should be mentioned that the running time refers to the time used for one epoch during the training process, including the time used

for both training and validation. It is clear that the number of parameters is not increased, and the training process is faster using the LoSDA module compared with PointSIFT. However, after stacking the GRA modules, the complexity of the model increases dramatically, especially when stacking the CRA module. For the different configurations of the GRA module, no matter the parallel or serial configurations, the number of parameters and the running time do not vary too much. In general, the complexity of GraNet is higher than the baseline methods. The increase of the classification accuracy may partially arise from the increase of the network complexity. However, since the training dataset is not a dataset with a large data amount, apart from the PointNet++, the running time of all the other networks varies in a small range.

Table 7.19: The number of parameters and running time of different network models.

Method	Params (Millions)	Running time (s)
PointNet++	0.97	32
PointSIFT	13.56	65
PointNet++ & LoSDA	9.58	52
GraNet (only CRA)	61.02	61
GraNet (only SRA)	10.68	60
GraNet mode 1	62.13	67
GraNet mode 2	62.13	65
GraNet mode 3	62.14	67

7.2.4 Semantically segmented construction scenes

The experimental results of semantic segmentation of the construction dataset are listed in Table 7.20. Here, the results of the GraNet method are compared with the results using the LoSDA method which has a similar network structure but without the GRA modules. As seen from the table, for the dataset acquired on Jan 16th, 2015, the GraNet method can achieve the results with an OA of 53.5% and $AvgF_1$ of 26.8%, while the LoSDA method can only obtain results with an OA of 52.5% and $AvgF_1$ of 24.1%. For the other dataset acquired on Feb 26, 2015, the OA of the semantic segmentation result using the GraNet is 55.8% and the $AvgF_1$ of the semantic segmentation result is 35.9%. The LoSDA shows a decrease of OA by 0.8% and $AvgF_1$ by 1.7%. From the result, we can see that the two deep learning-based methods can not achieve satisfying semantic segmentation results on the construction dataset, especially for some categories, such as waste, pipes, and planes. Even though the long-range relations can augment the performance of the network as shown in former experiments, the contribution of the relation modules is not that obvious on the construction dataset.

Table 7.20: Semantic segmentation results of the construction dataset. (Values in %). Noted that the highest values in OA, $AvgF_1$, and F_1 for each category are marked with bold texts.

Date	Methods	Cranes	Struct_const Formworks	Wood_piles Containers	Metal_piles Sheds	Concrete Buildings	Waste Naturals	Others Impervious	Pipes Bare lands	Planes OA	Scaffolds $AvgF_1$
2015/01/16	LoSDA	16.0	58.8	**16.2**	/	**19.0**	0	/	**9.4**	11.0	**56.1**
			2.4	0	/	/	/	**20.3**	**80.0**	52.5	24.1
	GraNet	**39.0**	**60.0**	2.4	/	17.8	**16.9**	/	1.5	**20.1**	54.2
			13.5	**1.2**	/	/	/	16.5	78.2	**53.5**	**26.8**
2015/02/26	LoSDA	22.5	**63.0**	23.4	**44.0**	37.5	0.7	/	0	12.7	50.1
			22.2	19.9	/	/	/	65.6	**83.6**	55.0	34.2
	GraNet	**36.8**	62.5	**25.5**	32.7	37.5	**13.4**	/	2.2	7.4	53.2
			18.6	29.6	/	/	/	65.6	81.5	**55.8**	**35.9**

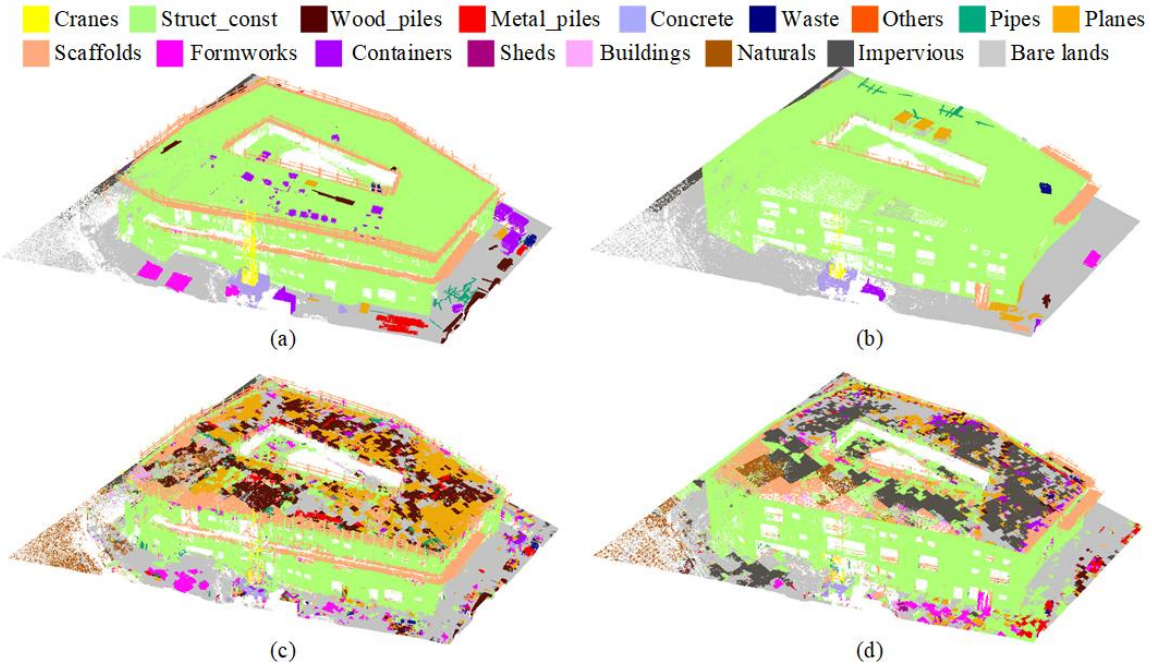

Figure 7.30: Semantic segmentation results of the construction dataset. a) - b) Ground truth, c) - d) the semantic segmentation results of the construction dataset using GraNet.

Fig. 7.30 illustrates the classification maps using the GraNet method. From the figure, it is obvious that GraNet shows satisfying classification results on categories, such as cranes, structures under construction, and scaffolds, which have evident geometric characteristics. In Fig. 7.31, we provide further details for the classification results. In this figure, we provide not only the comparison between the labeled ground truth and the classification result but also an illustration of the original data. From the comparison, it is clear that there are strongly different visual characteristics even for the same categories, such as structure under construction. This may be one of the reasons that lead to the wrongly classified result as illustrated in Fig. 7.31k. Generally speaking, there may be four major reasons that lead to the unsatisfying results on the construction dataset:

- The training process was conducted on the former three datasets which were acquired in the early stage of the construction process but the test was performed on the two datasets of the later construction stage. The environment of the construction scene and the corresponding objects all show a big difference.
- The construction dataset has a strong bias between different categories. For some categories, the number of samples is not enough to provide sufficient geometric and radiometric information for training.
- The difference between objects in the same categories is large, as shown by the previous example of the structure under construction (see Fig. 7.31). For instance, the visual difference may result from different illustrations or different covering materials. Geometric differences may result from the incompetence of data.
- One deficiency resulted from the deep learning-based method is that the result is influenced by the way of dividing. As shown in Fig. 7.31k, the boundaries of dividing blocks are obvious in the classification result.

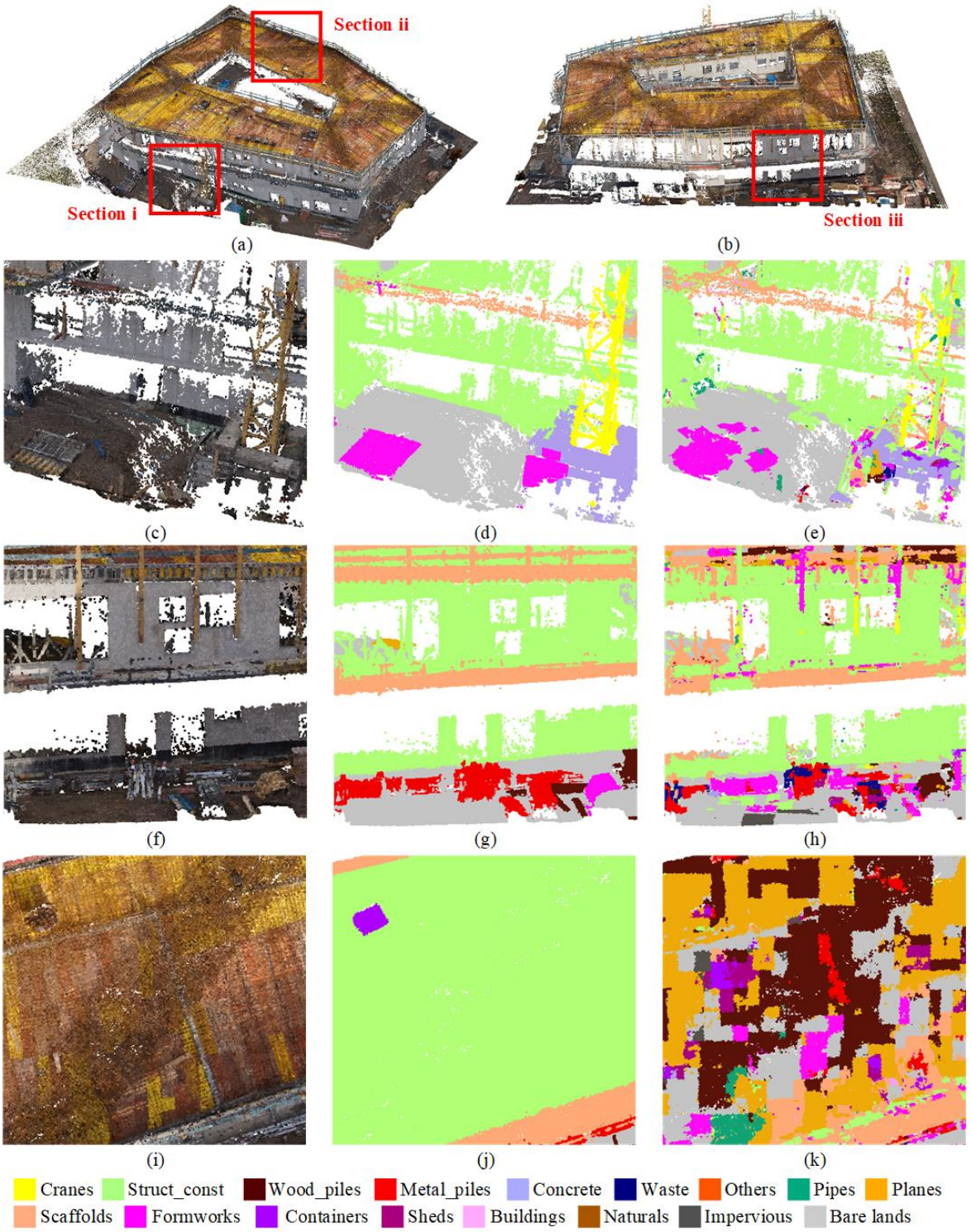

Figure 7.31: Details of semantic segmentation results of the construction dataset using GraNet. a) - b) Illustration of scene sections in point clouds, c) - e) original point clouds, ground truth, and semantic segmentation results of section i, f) - h) original point clouds, ground truth, and semantic segmentation results of section ii, i) - k) original point clouds, ground truth, and semantic segmentation results of section iii.

7.3 Change detection results

7.3.1 Geometrical changes of construction scenes

The experimental results of geometric change detection of the construction site are listed in Table 7.21. For the dataset acquired on Jan 16, 2015, the change detection was referenced with the dataset acquired on Dec 12, 2014. The dataset acquired on Feb 26, 2015 was compared with the dataset acquired on Jan 16. The voxel sizes for the occupancy-based change detection are all set to 0.3 m, considering the object sizes and the observed area. Here, we list the numbers of voxels detected as conflicting, consistent, and unknown, respectively.

Table 7.21: Results of geometric changes.

Acquisition Date	Number of voxels		
	Conflicting	Consistent	Unknown
2015/01/16	63403	26869	50598
2015/02/26	44349	38243	4788

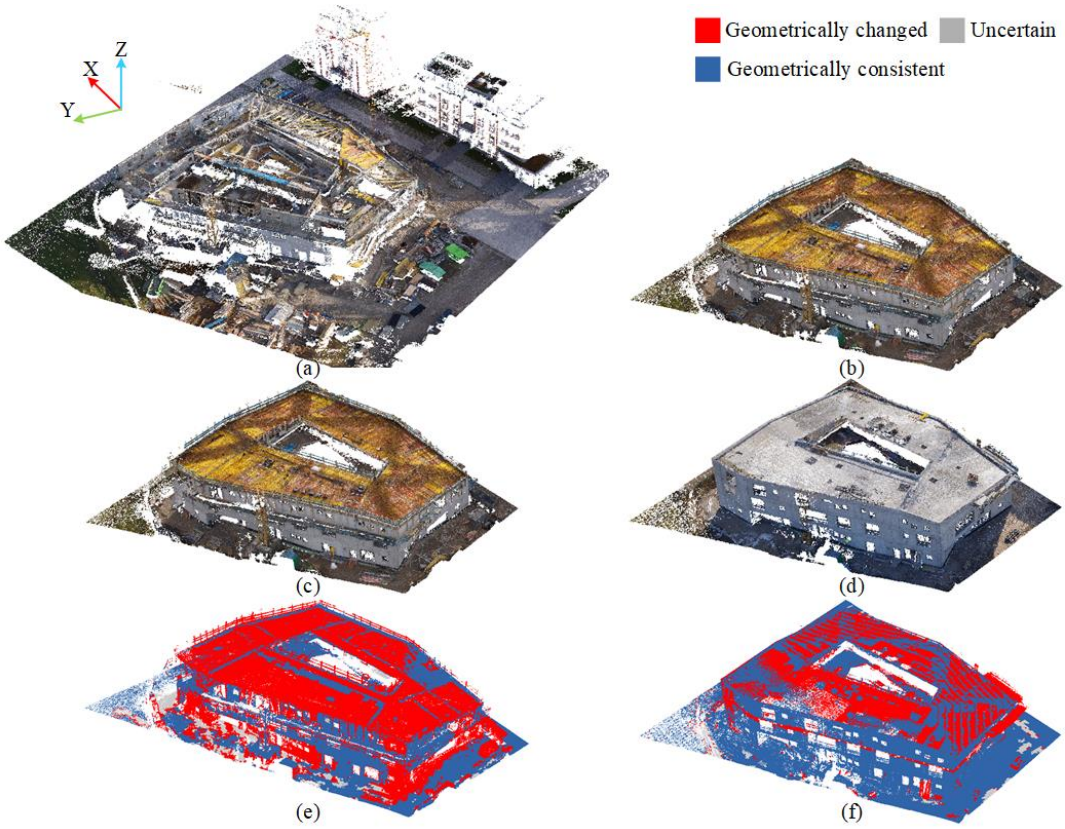

Figure 7.32: Experimental results of geometric change detection using construction point cloud sequence. a) - b) The construction data acquired at Dec 12, 2014 and Jan 16, 2015, respectively, c) - d) the construction data acquired at Jan 16, 2015 and Feb 26, 2015. e) - f) the detected geometric changes between the aforementioned two construction data pairs.

From the results, we can see that even though we only considered the union of occupied space provided by the datasets used for change detection and did not count the unobserved areas, there was still much space which had observing conflicts between different acquisition time. It also makes the process of detecting unknown areas and consideration of occlusions important.

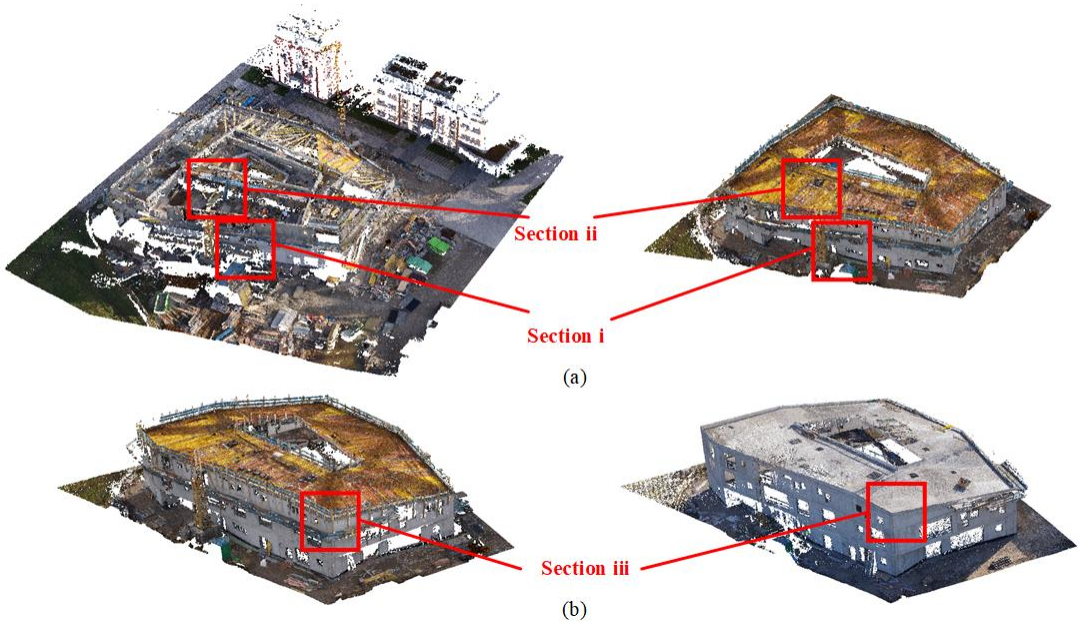

Figure 7.33: Illustration of selected scene sections for showing details of geometric changes. a) Section i and ii are selected from the scene, b) section iii are selected from the scene.

In addition, we can see that the number of unknown voxels decreases dramatically. It can be concluded that when the construction process went to the late stage, the change of observation environment decreased. As for the detected changes, compared with the data acquired in Dec 2014, about 75% of occupied space changed. However, when the construction process went further to the end stage, the changes occupied less than 50%.

In Fig. 7.32, the results of geometric change detection are illustrated. As shown in the figure, from Dec 12, 2014 to Jan 16, 2015, the main changes lay in the top of the structure under construction, as well as some temporary objects around the construction site. The unknown area mainly existed in the corner of walls or some areas which were occluded by temporal objects. Most of the vertical walls were consistent during the one month. However, there was still some part of the vertical structure which were under construction. From Jan 16, 2015 to Feb 26, 2015, the geometric changes decreased obviously, with fewer changes on the construction environment and construction structure. The vertical structure has much fewer changes compared with the last construction process stage. The main changes still lay in the area of top of the building under construction. Fig. 7.33 illustrated the scene sections we selected for detailed illustration. Fig. 7.34 illustrates the details of the detected geometric changes of selected sections. As shown in Fig. 7.34d, there is no formworks placed on the scene section, while in Fig. 7.34e, we can see that the formworks and iron for a building roof. The geometric changes were detected using our method as illustrated in Fig. 7.34f. the construction of the top plane of the structure was detected. In Figs. 7.34g-i, the changes from planes of scaffolds to a concrete building wall were also detected. However, there are also some drawbacks of this method. First, as illustrated in Fig. 7.34a, the points of the vertical structure were not reconstructed during the MVS process and this area was detected as changed areas in the result. In the occupancy-based geometric change detection method, the errors during the MVS process are not fully considered. In addition, for some areas whose semantic categories changed but the occupancy remained, the occupancy-based change detection can not detect them as changes. The semantic changes should also be taken into consideration.

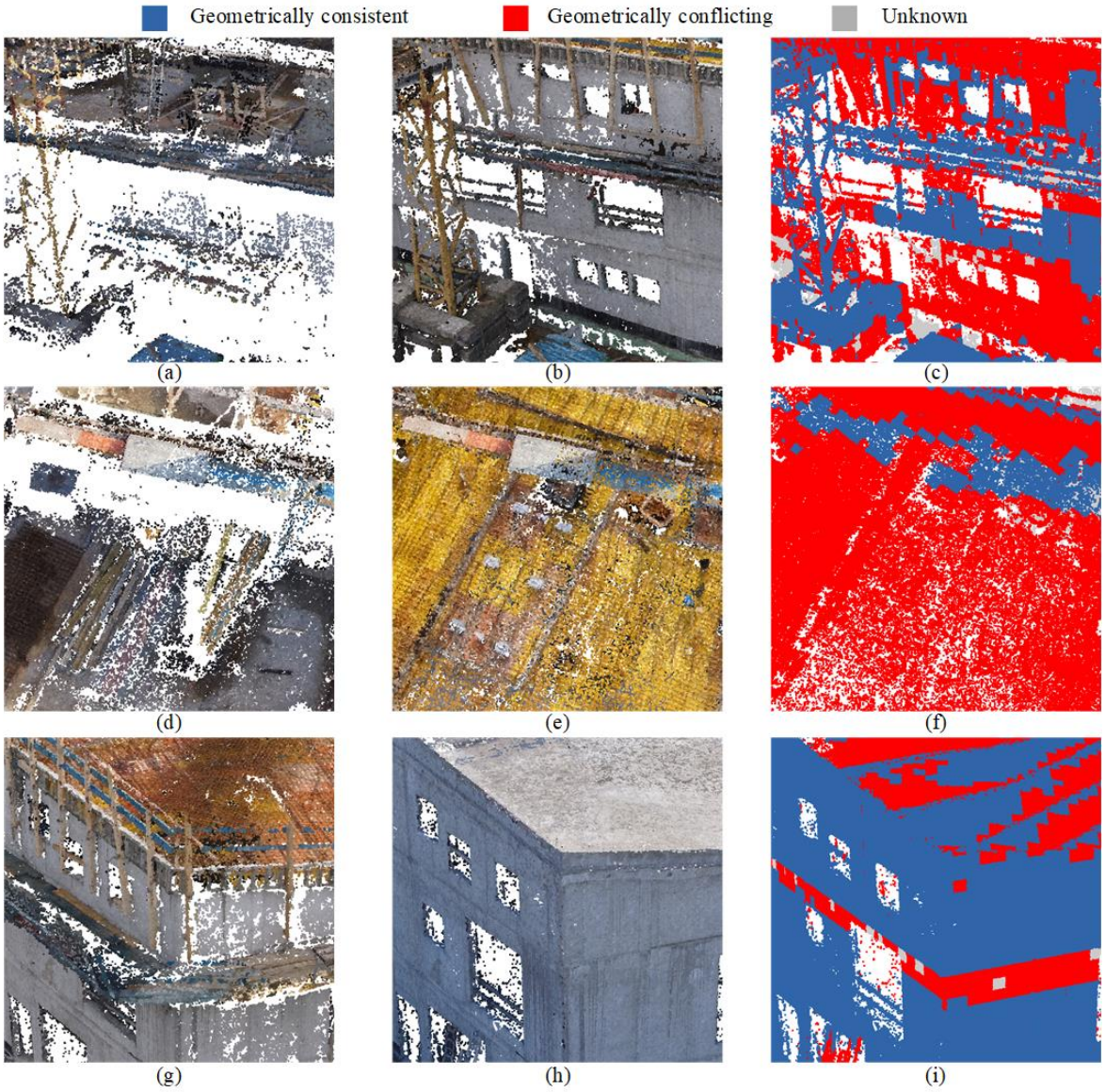

Figure 7.34: Details of geometric change detection results using the the OBCD-M method. a) - c) The original reference database, the current database, and the geometric change detection results of section i, d) - f) the original reference database, the current database, and the geometric change detection results of section ii, g) - i) the original reference database, the current database, and the geometric change detection results of section iii.

Table 7.22: Accuracy assessment of the SACD method using the construction dataset (Values in %).

Acquisition Date	Evaluation metric	Changed	Consistent	OA	$AvgF_1$
2015/01/16	pr	87.0	89.9	88.2	88.0
	r	97.8	82.2		
	F_1	89.9	86.1		
2015/02/26	pr	51.7	94.3	67.9	73.6
	r	93.6	54.6		
	F_1	72.6	74.4		

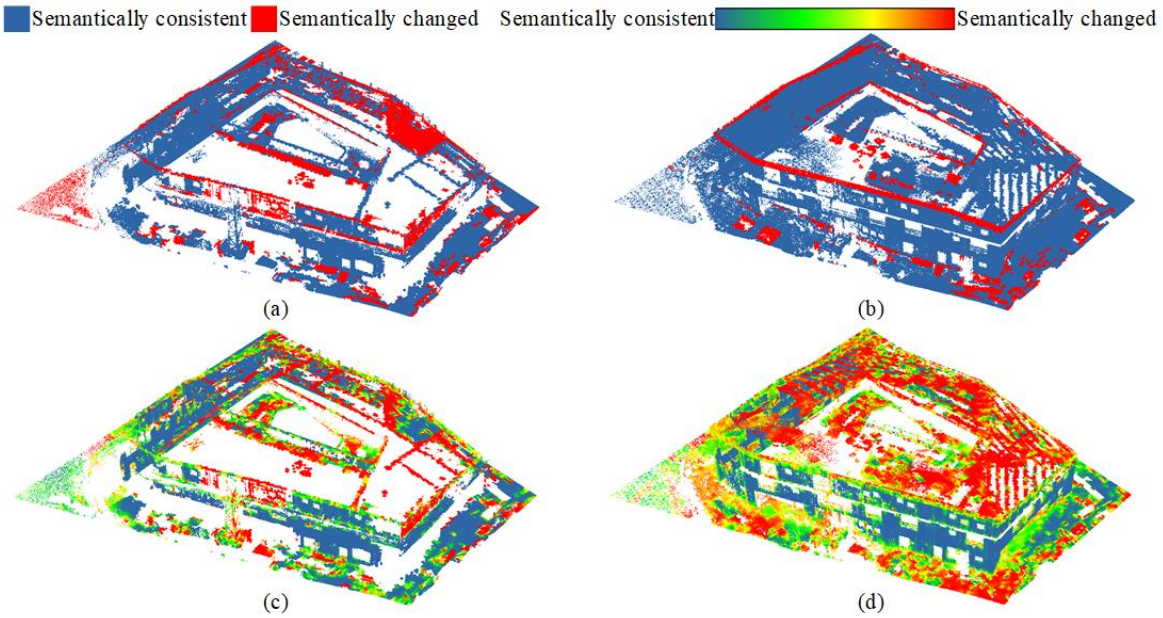

Figure 7.35: Experimental results of semantic change detection. Subplots a) and b) are the ground truth. Subplots c) and d) present the semantic change detection results.

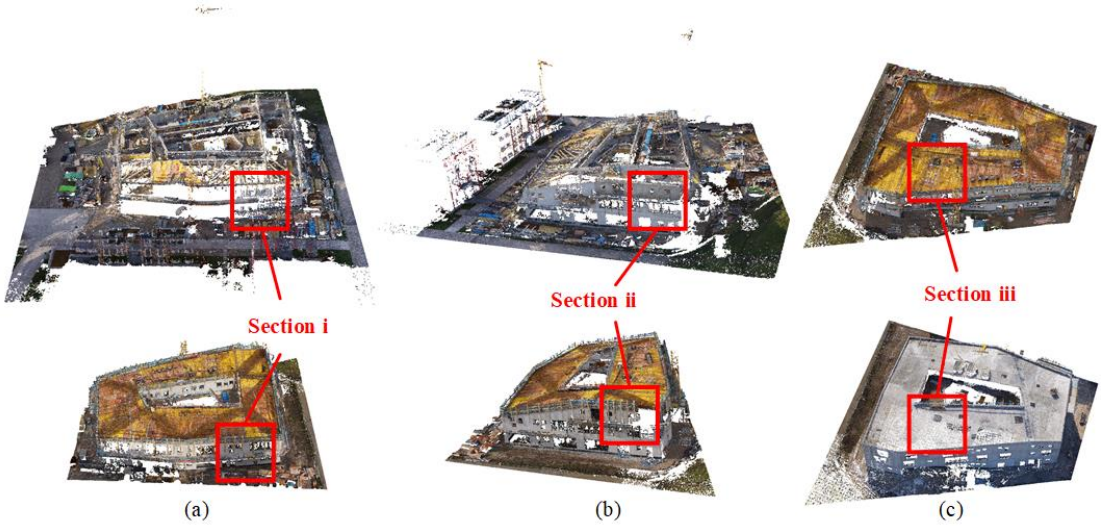

Figure 7.36: Illustration of selected scene sections for showing details of semantic changes. a) Section i, b) section ii, c) section iii.

7.3.2 Semantics-based changes of construction scenes

As stated in the last section, semantics are also an important indicator of changes. Thus, we further involved the semantic segmentation results obtained in Section 7.2.4 and obtained the changes with semantics. Table 7.22 lists the accuracy assessment of semantics-aided change detection. As seen from the table, we can see that the OA of semantics-aided change detection from Dec 12, 2014 to Jan 16, 2015 are 88.2% and the $AvgF_1$ is 88.0%. The F_1 of detecting changed is 89.9% and the F_1 of detecting consistent is 86.1%. As for detecting changes between the dataset of Jan 16, 2015 and Feb 26, 2015, the SACD method can achieve results with OA of 67.0% and 73.6%. The accuracies presented by F_1 of detecting changes and consistent are 72.6%

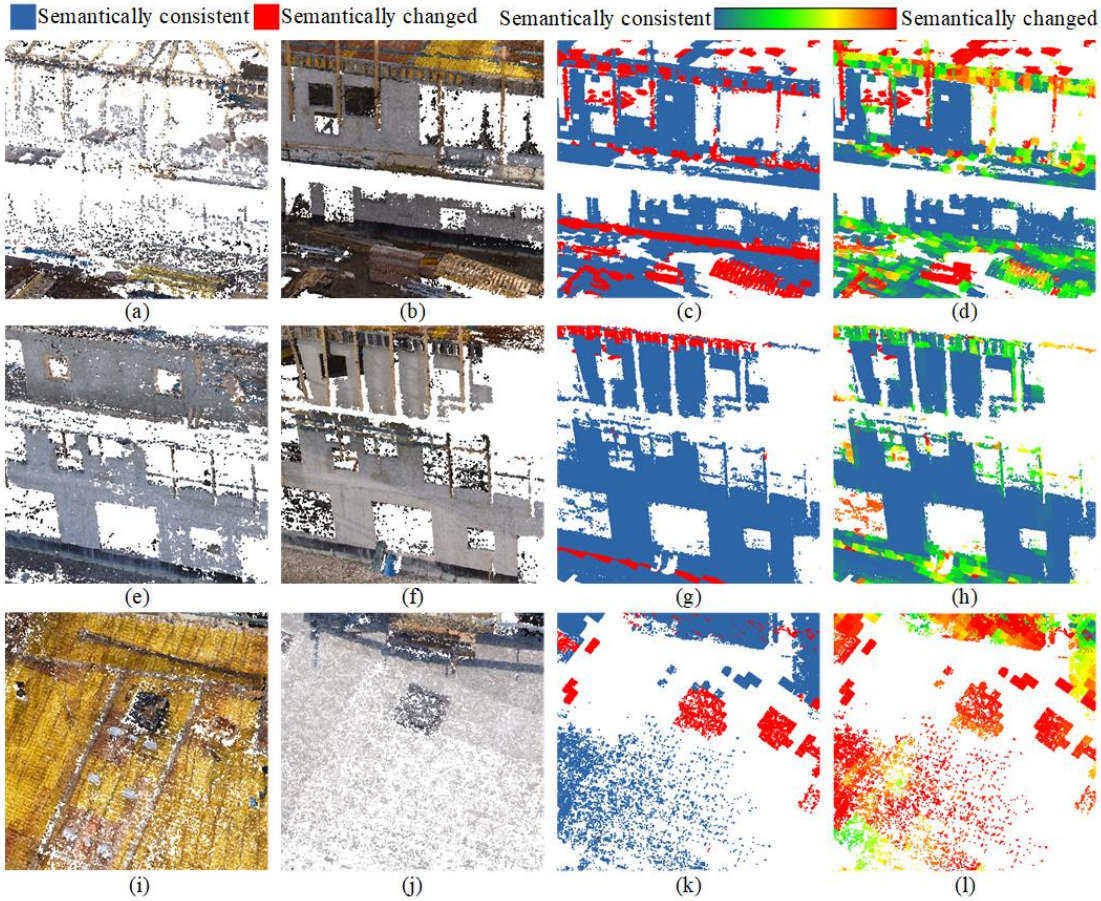

Figure 7.37: Details of experimental results of semantic change detection using the construction dataset. a) - d) the original reference database, current database, ground truth, the results of semantic change detection of section i, e) - h) the original reference database, current database, ground truth, the results of semantic change detection of section ii, i) - l) the original reference database, current database, ground truth, the results of semantic change detection of section iii.

and 74.4%, respectively. It should be mentioned that the first change detection was conducted on a dataset (Dec 12, 2014) with manually semantic labels and a dataset with predicted labels using our proposed method. However, the second one was performed on two datasets with predicted labels.

Fig. 7.35 depicts the results of semantics-aided change detection. Compared with the ground truth, the majority of changed and consistent areas coincide, especially for the changes of the vertical structure under construction and the temporal objects on the ground around the construction scene. However, the main mis-detected areas lay on the top plane of the structure under construction. It results from the misclassification of the top plane of the structure under construction. The selection of scene sections are illustrated in Fig. 7.36. In Fig. 7.37, the details of the results of the semantics-aided change detection are illustrated. Here, we also provide an illustration of the comparison between the original data. As shown in Fig. 7.36a-d and Fig. 7.36e-h, the vertical structures which remain consistent were correctly determined as semantic consistent areas. The area in the boundary of structure under construction, i.e., the top boundary, was detected as changed area as illustrated in Fig. 7.36a-d, which also coincides with the ground truth. However, as illustrated in Fig. 7.36i-l, section iii lies in the top plane of the building structure. The covering material changed, which led to a strong visual difference between the two observed scenes. That is also the reason why this area was detected as changes in our results. Thus, there are two

main drawbacks of our semantics-aided change detection method. The first one is that the result is influenced by both the results of semantic segmentation and the results of geometric change detection. Second, from the visual results of change detection, we can see that the boundary of detected changes is also influenced by the selection of voxels as illustated in Fig. 7.36l.

8 Discussion

In this chapter, we will discuss the three key tasks in the change detection of the construction site, namely registration, semantic segmentation, and change detection from several aspects based on the experimental results, including the major problems of the tasks, the advantages, the limitations of the proposed methods, and their sensitivity to the parameterization and the attributes of data.

8.1 Discussion on 3D point cloud registration

The experimental results described in Section 7.1 demonstrated that datasets could be well registered using our proposed methods. Two registration methods were proposed for the registration tasks, including the PBPC method and the GRPC method. By improving the projection-based method to a full 3D solution, the presented method is not limited to the application in urban areas or construction scenes. However, it is also applicable for non-regular-shaped areas, such as mountains. The registration result is also improved, especially for the estimation of rotations. The PBPC method relies on the determination of the principle, which is limited by the geometric characteristics. The estimation of principle will bring additional errors in the estimation of rotations. However, the GRPC method estimates all the transformations using the global information and in the frequency domain. In terms of registration accuracy, our proposed method can produce better results compared with other feature-based methods. We evaluated our proposed method under different scenarios. For example, the Bremen dataset and the RESSO dataset observed urban scenes, while in the WHU-TLS dataset, a mountain area was presented using TLS point clouds. These datasets also show different geometric characteristics, different point densities, and different overlapping cases. Our proposed method can achieve satisfying results on all these datasets. Meanwhile, the registration results on the construction datasets demonstrate that our proposed method is also efficient in registering multitemporal datasets. We also evaluated the robustness of our proposed method under different noise ratios and noise levels. The results show that the increase in noise levels and noise ratios did not significantly increase registration errors. The proposed method is robust under different noise situations. We evaluated different parameter settings for the proposed GRPC method. The experimental result shows that the setting of voxel size influenced the performance of the GRPC method. Large voxel size can increase the processing efficiency but decrease the presentation of details, leading to a deficiency in registration. A reasonable set of parameters represents a compromise between the level of details and the computational sources.

Generally, by utilizing global features and matching low-frequency components in the frequency domain, we can align point clouds under different scenarios, including datasets acquired from different viewpoints and at different times. Promising results also prove the versatility of the GRPC method to different datasets with regular-shaped or irregular-shaped geometric characteristics. The proposed GRPC method can efficiently achieve registration with majority rotation and translation errors, reaching less than 0.2 degrees and 0.5 m, and outperform state-of-the-art

methods on several benchmark datasets. Meanwhile, the experiments also proved that the GRPC method is kind of robust to noise and is still effective and efficient under low-overlapping cases. However, the registration is influenced by the selection of voxel size. The voxelization process will also lead to the aliasing effect. Additionally, the decoupling of rotation, scaling, and translation and the sequential process manner of these parameters will bring accumulated estimation errors for the parameters estimated in the late steps compared with conventional registration methods, which simultaneously estimate all the transformation parameters.

8.2 Discussion on semantic segmentation of urban scenes

As for the experiments of semantic segmentation, we evaluated the performance of the proposed method under different applications using several benchmark datasets. The results demonstrated that our proposed methods could achieve satisfying results for recognizing different urban objects presented in the benchmark datasets.

For the MLCE method, an ablation study was conducted on the proposed point cloud classification workflow that combined multi-scale geometric feature extraction with LPP for the DR of features. The experimental results demonstrated the importance of broadening the receptive field of local feature reconstruction and the effectiveness of DR in feature engineering. The qualitative and quantitative results revealed that our method could outperform other feature DR methods in classification and provide an effective and distinctive geometric feature representation. In addition, different parameter settings were tested on LPP. It shows that the performance of LPP is sensitive to the parameters. A proper setting of the parameters is vital in the process of DR.

As for the DPE method, it made an improvement based on the former strategy. First, we used a deep neural network, PointNet++, to directly extract deep features from pointsets instead of handcrafted features. Second, the extension of local receptive fields was achieved using a hierarchical subdivision strategy. The first two steps were named HDL in the DPE method. Third, a novel robust manifold-learning-based algorithm, namely JME, was employed for point feature embedding by integrating spatial information. Fourth, the initial classification results can be optimized by a graph-cuts-based regularization method, namely GGO. The experimental results indicate that the strategy of DPE can provide classification results with high accuracy, especially while providing fine-grained results. The results of the ablation study indicate the effectiveness of each part of the procedure. Moreover, these embedded features using JME were proved to be capable of producing smoother results, although degradation of the effect of GGO on the smoothing of the initial classification results was seen to some extent. We also evaluate the performance of the proposed method under different parameterizations. The performance of JME is sensitive to the change of reduced dimensionality but less sensitive to the change of the number of neighbors. The regularization strength slightly influences the classification results. Generally, a proper selection of dimensionality in feature embedding is of great importance in DPE. However, DPE showed weakness in identifying the boundaries of buildings and trees and classifying specific spectral and geometric similar categories, such as fence_hedge, shrubs, and trees. The JME algorithm is not efficient enough and has a high requirement on the computation memory, limiting the application of DPE to an extremely large-scale dataset.

The GraNet method is a novel deep-learning-based method investigating the construction of efficient local neighborhood representation and the importance of long-range dependencies provided by relation-aware modules. Experimental results demonstrated that the GraNet method achieved high classification results on three benchmark datasets and outperformed other commonly used advanced point-based strategies. Besides, comparative results also validated the feasibility of using the orientation information, the elevation information, and the local dependencies achieved

by local attention pooling to enhance the feature importance and the importance of considering long-range relations in complicated scenes, especially for urban scenes. Compared with the former two methods, GraNet does not require a careful selection for the hyperparameters. However, the way of subdivision of the training and test dataset affects the semantic segmentation results. The effect is apparent when classifying a large dataset, such as the DALES dataset and the construction dataset. The other drawback of GraNet is that the relation-aware modules involved in the network result in a heavy configuration of the network. In addition, the experimental results of GraNet on the construction dataset were not satisfying. It revealed that our proposed method still shows deficiency when dealing with datasets with high complexity.

Overall, we can identify the categories of objects in urban scenes and building objects in the construction scenes using the proposed semantic segmentation methods. High classification accuracies on different datasets prove the effectiveness of the proposed methods. The experimental results also revealed that our proposed methods outperformed other commonly used advanced point-based strategies and state-of-the-art methods. In addition, the DR methods, including LPP and JME, are sensitive to the hyperparameter setting. As for GraNet, the performance of this method is influenced by the data subdivision during the training and test process. Moreover, the performance of our semantic segmentation methods is also limited by the geometric and radiometric information provided by the datasets.

8.3 Discussion on change detection of the construction site scene

The experimental results in Section 7.3 demonstrated that geometric changes and semantic changes can be detected, provided that data are appropriately aligned and segmented in the former experiments of registration and semantic segmentation. The geometric changes can be reliably detected using the occupancy-based method. The detected unknown space can also reflect the changes of the observation environments. Although the result is specific to the experimental system, including the camera configurations and reconstructing 3D point clouds from images, the idea of retrieving rays for finding occlusions and geometric conflicts applies to other systems. However, for the occupancy-based change detection, the size of detectable changes is limited by the data quality and the size of 3D grid cells. The results reveal that there are obvious "zig zag" effect in the boundaries between areas with detected geometric changes and areas that are consistent. The natural boundaries can not be well presented. As for the semantic-aided change detection, we actually conduct a direct differencing between the semantic segmentation results of different time epochs. In this case, the results highly rely on the semantic segmentation quality. If the classification correctness is too low, we can not use the semantic segmentation results for detecting semantic changes. In addition, due to the influence of changes of illumination and covering materials on the results of semantic segmentation, these factors also have an obvious impact on the results of semantic change detection.

9 Conclusion and Outlook

In this chapter, we will provide conclusions drawn based on the work presented in this thesis and propose new possible directions for further research work based on the limitations of the proposed methods. The conclusions are grouped according to the their relationship to specific goals of this thesis and also regarding to the research question in Section 1.1.

9.1 Conclusion

Research question I: To what extent of robustness, accuracy and efficiency could a marker-free alignment of point clouds achieve?

First, the task of point cloud registration is addressed. Global features and low-frequency components are utilized for achieving point cloud registration. The qualitative and quantitative results reveal that the GRPC method can outperform other representative registration methods as well as our former method (e.g., PBPC), with the majority of rotation and translation errors reaching less than 0.2 degrees and 0.5 m. Additionally, it has been proved that the proposed registration method shows its superiority in the matching of point clouds with high-level noise and with a wide range of overlaps and various geometric characteristics, which shows the robustness of the proposed registration method. Furthermore, the case study on the registration of the construction dataset demonstrates that the proposed registration method is also robust to temporal changes and incompleteness of data. Regarding the efficiency of the proposed method, when registering large-scale datasets (i.e., the Resso dataset), the processing time was less than 50 s. However, the proposed method is sensitive to voxel size, indicating the level of details presented by the global features obtained after the voxelization step. A large voxel size will lead to a strong blurring effect on the presentation of details, while a small voxel size will significantly decrease processing efficiency. Additionally, since the registration method follows a sequential estimation procedure and the transformation parameters are not estimated synchronously, estimation errors will be accumulated. These factors should be considered in our future work.

Research question II: What are the necessary aspects to be considered when learning features for an accurate interpretation of complex scenarios using point clouds?

Here, the task addressed is the semantic segmentation of point clouds. For designing robust and discriminative features for the semantic segmentation task, we develop feature embedding strategies using traditional methods and deep learning-based methods, whose main focus lies in improving receptive fields of points, the engineering of point correlations, and the involvement of attention mechanism. The improvement on semantic segmentation results when applying multi-scale neighborhood construction strategy reveal the importance of improving receptive fields. The correlation of local points can be considered by utilizing manifold-learning-based methods. From the result of JME, we can conclude that the consideration of local point correlations in both spatial and feature domain could promote the accuracy of semantic segmentation and also

improve the smoothness of semantic segmentation results. On the other hand, the correlations between non-local points are investigated by building global relations between points, which is achieved by utilizing GGO in the DPE method and the GRA modules in the GraNet method. In addition, the attention mechanism is also involved in improving the discriminate features and suppressing interference. The experiments on several benchmark datasets presenting various urban scenarios have validated the effectiveness of consideration of non-local point correlations and the attention mechanism in the task of semantic segmentation. The aforementioned aspects, including the improvement of receptive fields, engineering of point correlations, especially in a global way, and the attention mechanism, ensure fine-grained and highly accurate semantic segmentation results under different scenarios using point clouds with different point densities or intense noise and outliers. We can obtain a classification result with $AvgF_1$ of 73.6 % on the ISPRS benchmark dataset, outperforming most start-of-the-art methods. However, despite that the aforementioned aspects show promising results in semantic segmentation, there are still some drawbacks. For instance, the point embedding achieved by manifold-learning-based methods is sensitive to hyperparameters. It takes much effort to define optimal parameters. The high computation requirement has also limited the application of the proposed manifold-learning-based method to a large-scale dataset. The GraNet method is a fully end-to-end deep learning-based solution, and it can be utilized in various scenarios. However, the way of division of training and test has a high effect on the semantic segmentation results.

Research question IIIa: To what extent of automation could be achieved for change detection?

The final task addressed is change detection during the construction progress. Here, we identified changes from two different perspectives. The first one is the geometric changes, which indicate the changes of occupancy in 3D space. It can be the change of appearance or shapes of building objects. The second one is the changes in semantics. It indicates that the occupancy of 3D space does not change, but the categories of objects change. Using the two different types of changes, we can fully consider the changes that may happen on the construction sites and define the difference between the changes. For the change detection, an overall accuracy of about 75% can be finally achieved. The whole process of change detection of building objects in the construction site, including the procedures: robust registration, semantic segmentation, detection of geometric and semantic change detection with a voxel representation. The three parts involved in the process are automated. However, there are some obvious limitations of the current workflow. The final change detection results rely highly on the results of registration and semantic segmentation. An optimal parameterization for each former step is vital for the success of change detection. It needs a lot of computation effort. Additionally, the semantic segmentation of point clouds requires a large number of training samples. Here, the ground truth was created based on manual work, limiting the level of automation of the whole procedure.

Research question IIIb: Is it sufficient enough to detect changes by purely comparing segmentation results of building objects?

In the proposed framework of change detection, apart from the conflicts in semantics, one crucial part is to take occlusions and missing information into account. Besides, as explained previously, the change detection by purely comparing semantic segmentation results of building objects is influenced by the accuracy of semantic segmentation results. It is also one of drawbacks of our framework. Poor semantic segmentation results will lead to a hard interpretation of the results of change detection. Some geometric constraints could be considered to improve the results. This is also a further direction we could consider in our future work. Additionally, in our proposed

method, the voxels are the basic element presenting the changes. A proper selection of voxel size is vital for compromising the presentation of detail and computation efforts. The voxel representation also limits the presentation of boundaries of changes.

To summarize, we proposed a framework for detecting changes of the construction sites, including a series of methods and algorithms addressing the specific tasks, namely point cloud registration, semantic segmentation, and change detection. These methods can provide effective tools for fast point cloud alignment, scene interpretation, overall change detection when dealing with point clouds acquired from construction sites. However, for real applications in construction monitoring, the connections to the construction plans and schedules should be considered. For example, an alignment between acquired point clouds and as-planned model should be conducted. The interpretation of construction scene should be on object-level. The changes should be also be on the object-level thus a detailed monitoring on each object of interest on the construction sites could be achieved. There is still a long way to go for achieving a detailed construction monitoring using point clouds.

9.2 Outlook

Considering the drawbacks of the proposed methods and the deficiency in real applications, in the future, the following aspects can be investigated:

- The first task is the optimization of semantic interpretation of point clouds. First, the performance should be further improved. Although our method performed well in some benchmark datasets, the performance on the construction datasets was not fully satisfactory, which also limited the performance of semantic change detection. The current method focus more on the learning of geometric features based on 3D coordinates of 3D points. However, the color information is not fully investigated. The further work could be conducted on the investigation of attribute information provided by point clouds. Second, the point-based method will produce results influenced by the way of division of training and test. The division boundary will show on the semantic segmentation results. Currently, there are some deep learning methods working on clusters of point instead of directly on 3D points. The natural boundaries could be preserved by the point clusters and the use of point clusters could also greatly decrease the size of inputs and thus improve the efficiency.

- The point cloud registration could be further optimized. First, the influence of voxel size could be decreased by building a hierarchical voxel selection. The voxel size is selected following an adaptive step, depending on the demand for accuracy of the registration, the data quality, and the scene size. Then, the point clouds could be separated partially and segments could be further registered with small voxel size. In the procedure, the registration can be optimized by gradually registering point clouds from globally to locally. The accumulation errors appearing in both feature-based registration and our proposed method should be further addressed. Currently, there are many deep learning-based methods developed for achieving finding of correspondence and transformation estimation synchronously. The use of some newly developed deep learning techniques for point cloud registration may help us overcome the problems.

- The process of change detection could be further improved. In this work, we present changes in the voxel element. However, in real applications, object-based monitoring is usually required for updating the state information of construction progress. Further research could be conducted on the optimization of the whole change detection procedure. Instead of semantic segmentation of point clouds, construction datasets should be segmented to instance- and

object-based levels. Then, changes could be conducted by comparing corresponding objects for detailed change detection. In addition, in the context of occupancy models, we have only considered the belief masses of measurements from different acquisition epochs. Other conditions or constraints are not taken into consideration. In future work, the probabilistic approaches, i.e., Bayes' theorem, could be applied to substitute the Dempster-Shafer theory for taking other conditions into consideration.

Bibliography

Abeid J, Allouche E, Arditi D, Hayman M (2003) Photo-net ii: a computer-based monitoring system applied to project management. Automation in Construction, 12 (5): 603–616.

Aijazi AK, Checchin P, Trassoudaine L (2013) Detecting and updating changes in lidar point clouds for automatic 3d urban cartography. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, II-5/W2: 7–12.

Al-Durgham M, Habib A (2013) A framework for the registration and segmentation of heterogeneous lidar data. Photogrammetric Engineering & Remote Sensing, 79 (2): 135–145.

Arayici Y (2007) An approach for real world data modelling with the 3d terrestrial laser scanner for built environment. Automation in Construction, 16 (6): 816–829.

Armeni I, Sener O, Zamir AR, Jiang H, Brilakis I, Fischer M, Savarese S (2016) 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 1534–1543.

Bachmann CM, Ainsworth TL, Fusina RA (2005) Exploiting manifold geometry in hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing, 43 (3): 441–454.

Bae KH, Lichti DD (2008) A method for automated registration of unorganised point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 63 (1): 36–54.

Basgall PL, Kruse FA, Olsen RC (2014) Comparison of lidar and stereo photogrammetric point clouds for change detection. In: Turner MD, Kamerman GW, Thomas LMW, Spillar EJ (eds) Laser Radar Technology and Applications XIX; and Atmospheric Propagation XI, 9080: 214 – 227.

Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 15 (6): 1373–1396.

Bellekens B, Spruyt V, Berkvens R, Penne R, Weyn M (2015) A benchmark survey of rigid 3d point cloud registration algorithms. International Journal on Advances in Intelligent Systems, 8: 118–127.

Besl PJ, McKay ND (1992) Method for registration of 3-d shapes. In: Robotics-DL Tentative: 586–606.

Biber P, Straßer W (2003) The normal distributions transform: A new approach to laser scan matching. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453), 3: 2743–2748.

Bosché F (2010) Automated recognition of 3d cad model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. Advanced Engineering Informatics, 24 (1): 107–118.

Bosché F, Guillemet A, Turkan Y, Haas CT, Haas R (2014) Tracking the built status of mep works: Assessing the value of a scan-vs-bim system. Journal of Computing in Civil Engineering, 28 (4): 05014004.

Boulch A (2020) Convpoint: Continuous convolutions for point cloud processing. Computers & Graphics, 88: 24–34.

Boulch A, Guerry J, Le Saux B, Audebert N (2018) Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. Computers & 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, (9): 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 (11): 1222–1239.

Braun A, Tuttas S, Borrmann A, Stilla U (2015) Automated progress monitoring based on photogrammetric point clouds and precedence relationship graphs. In: ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, 32: 1.

Bresenham JE (1965) Algorithm for computer control of a digital plotter. IBM Systems journal, 4 (1): 25–30.

Bülow H, Birk A (2012) Spectral 6dof registration of noisy 3d range data with partial overlap. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (4): 954–969.

Bülow H, Birk A (2018) Scale-free registrations in 3d: 7 degrees of freedom with fourier mellin soft transforms. International Journal of Computer Vision, 126 (7): 731–750.

Chan JCW, Paelinckx D (2008) Evaluation of random forest and adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. Remote Sensing of Environment, 112 (6): 2999–3011.

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 (Part 3): W8.

Chen S, Nan L, Xia R, Zhao J, Wonka P (2019) Plade: A plane-based descriptor for point cloud registration with small overlap. IEEE Transactions on Geoscience and Remote Sensing, 58 (4): 2530–2540.

Chen X, Ma H, Wan J, Li B, Xia T (2017) Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 1907–1915.

Chi S, Caldas CH, Kim DY (2009) A methodology for object identification and tracking in construction based on spatial modeling and image matching techniques. Computer-Aided Civil and Infrastructure Engineering, 24 (3): 199–211.

Choy C, Gwak J, Savarese S (2019) 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 3075–3084.

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

Dai A, Chang AX, Savva M, Halber M, Funkhouser T, Nießner M (2017) Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 5828–5839.

Dong Z, Liang F, Yang B, Xu Y, Zang Y, Li J, Wang Y, Dai W, Fan H, Hyyppäb J et al. (2020) Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. ISPRS Journal of Photogrammetry and Remote Sensing, 163: 327–342.

Dong Z, Yang B, Liang F, Huang R, Scherer S (2018) Hierarchical registration of unordered tls point clouds based on binary shape context descriptor. ISPRS Journal of Photogrammetry and Remote Sensing, 144: 61–79.

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

El-Omari S, Moselhi O (2008) Integrating 3d laser scanning and photogrammetry for progress measurement of construction work. Automation in Construction, 18 (1): 1–9.

Engelcke M, Rao D, Wang DZ, Tong CH, Posner I (2017) Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In: 2017 IEEE International Conference on Robotics and Automation (ICRA): 1355–1361.

Feng M, Zhang L, Lin X, Gilani SZ, Mian A (2020) Point attention network for semantic segmentation of 3d point clouds. Pattern Recognition, 107: 107446.

Flitton GT, Breckon TP, Bouallagu NM (2010) Object recognition using 3d sift in complex ct volumes. In: BMVC (1): 1–12.

Fu J, Liu J, Tian H, Li Y, Bao Y, Fang Z, Lu H (2019) Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 3146–3154.

Ge X (2017) Automatic markerless registration of point clouds with semantic-keypoint-based 4-points congruent sets. ISPRS Journal of Photogrammetry and Remote Sensing, 130: 344–357.

Ge X, Hu H (2020) Object-based incremental registration of terrestrial point clouds in an urban environment. ISPRS Journal of Photogrammetry and Remote Sensing, 161: 218–232.

Ge X, Wunderlich T (2016) Surface-based matching of 3d point clouds with variable coordinates in source and target system. ISPRS Journal of Photogrammetry and Remote Sensing, 111: 1–12.

Gehrung J, Hebel M, Arens M, Stilla U (2016) A framework for voxel-based global scale modeling of urban environments. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W1: 45–51.

Gehrung J, Hebel M, Arens M, Stilla U (2018) A voxel-based metadata structure for change detection in point clouds of large-scale urban areas. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2: 97–104.

Gehrung J, Hebel M, Arens M, Stilla U (2019) A fast voxel-based indicator for change detection using low resolution octrees. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W5: 357–364.

Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition: 3354–3361.

Ghamisi P, Höfle B (2017) Lidar data classification using extinction profiles and a composite kernel support vector machine. IEEE Geoscience and Remote Sensing Letters, 14 (5): 659–663.

Golparvar-Fard M, Peña-Mora F, Savarese S (2009) D4ar–a 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. Journal of Information Technology in Construction, 14 (13): 129–153.

Golparvar-Fard M, Pena-Mora F, Savarese S (2015) Automated progress monitoring using unordered daily construction photographs and ifc-based building information models. Journal of Computing in Civil Engineering, 29 (1): 04014025.

Gordon C, Akinci B (2005) Technology and process assessment of using ladar and embedded sensing for construction quality control. In: Construction Research Congress 2005: Broadening Perspectives: 1–10.

Gorgens EB, Valbuena R, Rodriguez LCE (2017) A method for optimizing height threshold when computing airborne laser scanning metrics. Photogrammetric Engineering & Remote Sensing, 83 (5): 343–350.

Gressin A, Mallet C, Demantké J, David N (2013) Towards 3d lidar point cloud registration improvement using optimal neighborhood knowledge. ISPRS Journal of Photogrammetry and Remote Sensing, 79: 240–251.

Guo Y, Sohel F, Bennamoun M, Lu M, Wan J (2013) Rotational projection statistics for 3d local surface description and object recognition. International Journal of Computer Vision, 105 (1): 63–86.

Haas C, Shen H, Phang W, Haas R (1984) Application of image analysis technology to automation of pavement condition surveys. Publication of: Balkema (AA).

Habib A, Detchev I, Bang K (2010) A comparative analysis of two approaches for multiple-surface registration of irregular point clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 38 (1): 61–66.

Habib A, Ghanma M, Morgan M, Al-Ruzouq R (2005) Photogrammetric and lidar data registration using linear features. Photogrammetric Engineering & Remote Sensing, 71 (6): 699–707.

Hackel T, Savinov N, Ladicky L, Wegner JD, Schindler K, Pollefeys M (2017) SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1-W1: 91–98.

Hebel M (2012) Änderungsdetektion in urbanen Gebieten durch objektbasierte Analyse und schritthaltenden Vergleich von Multi-Aspekt ALS-Daten. Dissertation, Technische Universität München, Fakultät für Bauingenieur- und Vermessungswesen, Photogrammetrie und Fernerkundung.

Hebel M, Arens M, Stilla U (2009) Utilization of 3D city models and airborne laser scanning for terrain-based navigation of helicopters and UAVs. In: Stilla U, Rottensteiner F, Paparoditis N (eds) CMRT09. International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation, 38 (3/W4): 187–192.

Hebel M, Arens M, Stilla U (2011) Change detection in urban areas by direct comparison of multi-view and multi-temporal ALS data. In: Stilla U, Rottensteiner F, Mayer H, Jutzi B, Butenuth M (eds) Photogrammetric Image Analysis, ISPRS Conference, PIA 2011. Lecture Notes in Computer Sciences (LNCS) 6952, Heidelberg: Springer: 185–196.

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

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

Hebel M, Stilla U (2010) LiDAR-supported navigation of uavs over urban areas. Surveying and Land Information Science, 70 (3): 139–149.

Hebel M, Stilla U (2012) Simultaneous calibration of ALS systems and alignment of multiview LiDAR scans of urban areas. IEEE Transactions on Geoscience and Remote Sensing, 50 (6): 2364–2379.

Hirschmuller H (2007) Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30 (2): 328–341.

Hoge WS (2003) A subspace identification extension to the phase correlation method [mri application]. IEEE Transactions on Medical Imaging, 22 (2): 277–280.

Holz D, Ichim AE, Tombari F, Rusu RB, Behnke S (2015) Registration with the point cloud library: A modular framework for aligning in 3-d. IEEE Robotics & Automation Magazine, 22 (4): 110–124.

Hong D, Yokoya N, Zhu XX (2017) Learning a robust local manifold representation for hyperspectral dimensionality reduction. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10 (6): 2960–2975.

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

Huang J, Kwok TH, Zhou C (2017) V4pcs: Volumetric 4pcs algorithm for global registration. Journal of Mechanical Design, 139 (11).

Huang R, Hong D, Xu Y, Yao W, Stilla U (2019) Multi-scale local context embedding for lidar point cloud classification. IEEE Geoscience and Remote Sensing Letters, 17 (4): 1–5.

Huang R, Xu Y, Hoegner L, Stilla U (2020a) Temporal comparison of construction sites using photogrammetric point cloud sequences and robust phase correlation. Automation in Construction, 117: 103247.

Huang R, Xu Y, Hong D, Yao W, Ghamisi P, Stilla U (2020b) Deep point embedding for urban classification using ALS point clouds: A new perspective from local to global. ISPRS Journal of Photogrammetry and Remote Sensing, 163: 62–81.

Huang R, Xu Y, Stilla U (2021) Granet: Global relation-aware attentional network for semantic segmentation of als point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 177: 1–20.

Huang R, Xu Y, Yao W, Hoegner L, Stilla U (2020c) Robust global registration of point clouds by closed-form solution in the frequency domain. ISPRS Journal of Photogrammetry and Remote Sensing, 171: 310–329.

Huang R, Ye Z, Boerner R, Yao W, Xu Y, Stilla U (2019) Fast pairwise coarse registration between point clouds of construction sites using 2d projection based phase correlation. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W13: 1015–1020.

Ibrahim Y, Lukins TC, Zhang X, Trucco E, Kaka A (2009) Towards automated progress assessment of workpackage components in construction projects using computer vision. Advanced Engineering Informatics, 23 (1): 93–103.

Jiang M, Wu Y, Zhao T, Zhao Z, Lu C (2018) Pointsift: A sift-like network module for 3D point cloud semantic segmentation. arXiv preprint arXiv:1807.00652.

Jutzi B, Gross H (2010) Investigations on surface reflection models for intensity normalization in airborne laser scanning (ALS) data. Photogrammetric Engineering & Remote Sensing, 76 (9): 1051–1060.

Kang Z, Yang J (2018) A probabilistic graphical model for the classification of mobile lidar point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 143: 108–123.

Kang Z, Zhang L, Yue H, Lindenbergh R (2013) Range image techniques for fast detection and quantification of changes in repeatedly scanned buildings. Photogrammetric Engineering & Remote Sensing, 79 (8): 695–707.

Kim C, Son H, Kim C (2013) Automated construction progress measurement using a 4d building information model and 3d data. Automation in Construction, 31: 75–82.

Klokov R, Lempitsky V (2017) Escape from cells: Deep kd-networks for the recognition of 3D point cloud models. In: Proceedings of the IEEE International Conference on Computer Vision: 863–872.

Kolmogorov V, Zabih R (2004) What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2): 147–159.

Kuhn A, Hirschmüller H, Scharstein D, Mayer H (2017) A tv prior for high-quality scalable multi-view stereo reconstruction. International Journal of Computer Vision, 124 (1): 2–17.

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. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 4558–4567.

Lee J, Son H, Kim C, Kim C (2013) Skeleton-based 3d reconstruction of as-built pipelines from laser-scan data. Automation in construction, 35: 199–207.

Leprince S, Barbot S, Ayoub F, Avouac JP (2007) Automatic and precise orthorectification, coregistration, and subpixel correlation of satellite images, application to ground deformation measurements. IEEE Transactions on Geoscience and Remote Sensing, 45 (6): 1529–1558.

Li N, Liu C, Pfeifer N (2019a) Improving lidar classification accuracy by contextual label smoothing in post-processing. ISPRS Journal of Photogrammetry and Remote Sensing, 148: 13–31.

Li W, Wang FD, Xia GS (2020a) A geometry-attentional network for ALS point cloud classification. ISPRS Journal of Photogrammetry and Remote Sensing, 164: 26–40.

Li X, Cheng G, Liu S, Xiao Q, Ma M, Jin R, Che T, Liu Q, Wang W, Qi Y et al. (2013) Heihe watershed allied telemetry experimental research (hiwater): Scientific objectives and experimental design. Bulletin of the American Meteorological Society, 94 (8): 1145–1160.

Li X, Wang L, Wang M, Wen C, Fang Y (2020b) Dance-net: Density-aware convolution networks with context encoding for airborne lidar point cloud classification. ISPRS Journal of Photogrammetry and Remote Sensing, 166: 128–139.

Li Y, Bu R, Sun M, Wu W, Di X, Chen B (2018) Pointcnn: Convolution on x-transformed points. In: Advances in Neural Information Processing Systems: 820–830.

Li Y, Chen D, Du X, Xia S, Wang Y, Xu S, Yang Q (2019b) Higher-order conditional random fields-based 3D semantic labeling of airborne laser-scanning point clouds. Remote Sensing, 11 (10): 1248.

Lu Y, Rasmussen C (2012) Simplified markov random fields for efficient semantic labeling of 3d point clouds. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems: 2690–2697.

Ma L, Crawford MM, Tian J (2010) Local manifold learning-based k-nearest-neighbor for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 48 (11): 4099–4109.

Maas HG (1999) The potential of height texture measures for the segmentation of airborne laserscanner data. In: Fourth International Airborne Remote Sensing Conference and Exhibition/21st Canadian Symposium on Remote Sensing, 1: 154–161.

Magnusson M, Lilienthal A, Duckett T (2007) Scan registration for autonomous mining vehicles using 3d-ndt. Journal of Field Robotics, 24 (10): 803–827.

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 (6): S71–S84.

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

Mellado N, Aiger D, Mitra NJ (2014) Super 4pcs fast global pointcloud registration via smart indexing. In: Computer Graphics Forum, 33 (5): 205–215.

Munoz D, Bagnell JA, Vandapel N, Hebert M (2009) Contextual classification with functional max-margin markov networks. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition: 975–982.

Myronenko A, Song X (2010) Point set registration: Coherent point drift. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (12): 2262–2275.

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.

Pagac D, Nebot E, Durrant-Whyte H (1998) An evidential approach to map-building for autonomous vehicles. IEEE Transactions on Robotics and Automation, 14 (4): 623–629.

Potts RB (1952) Some generalized order-disorder transformations. In: Mathematical Proceedings of the Cambridge Philosophical Society, 48 (1): 106–109.

Puri N, Turkan Y (2020) Bridge construction progress monitoring using lidar and 4d design models. Automation in Construction, 109: 102961.

Qi CR, Liu W, Wu C, Su H, Guibas LJ (2018) Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 918–927.

Qi CR, Su H, Mo K, Guibas LJ (2017a) Pointnet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 652–660.

Qi CR, Su H, Nießner M, Dai A, Yan M, Guibas LJ (2016) Volumetric and multi-view cnns for object classification on 3D data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 5648–5656.

Qi CR, Yi L, Su H, Guibas LJ (2017b) Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems: 5099–5108.

Qin N, Hu X, Wang P, Shan J, Li Y (2019) Semantic labeling of ALS point cloud via learning voxel and pixel representations. IEEE Geoscience and Remote Sensing Letters.

Rabbani T, Van Den Heuvel F, Vosselmann G (2006) Segmentation of point clouds using smoothness constraint. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 36 (5): 248–253.

Rothermel M, Wenzel K, Fritsch D, Haala N (2012) Sure: Photogrammetric surface reconstruction from imagery. In: Proceedings LC3D Workshop, 8 (2).

Rottensteiner F, Sohn G, Jung J, Gerke M, Baillard C, Benitez S, Breitkopf U (2012) The isprs benchmark on urban object classification and 3d building reconstruction. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1, 1 (1): 293–298.

Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science, 290 (5500): 2323–2326.

Rusu RB, Blodow N, Beetz M (2009) Fast point feature histograms (fpfh) for 3d registration. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on: 3212–3217.

Schachtschneider J, Schlichting A, Brenner C (2017) Assessing temporal behavior in lidar point clouds of urban environments. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-1/W1: 543–550.

Schnabel R, Wahl R, Klein R (2007) Efficient ransac for point-cloud shape detection. In: Computer Graphics Forum, 26 (2): 214–226.

Shih NJ, Wang PH (2004) Using point cloud to inspect the construction quality of wall finish. In: Proceedings of the 22nd Education and research in Computer Aided Architectural Design in Europe (eCAADe) Conference: 573–578.

Simonovsky M, Komodakis N (2017) Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 3693–3702.

Singh A (1989) Review article digital change detection techniques using remotely-sensed data. International Journal of Remote Sensing, 10 (6): 989–1003.

Stone W, Cheok G (2001) Ladar sensing applications for construction. Technical Paper, National Institute of Standards and Technology.

Su H, Jampani V, Sun D, Maji S, Kalogerakis E, Yang MH, Kautz J (2018) Splatnet: Sparse lattice networks for point cloud processing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 2530–2539.

Su H, Maji S, Kalogerakis E, Learned-Miller E (2015) Multi-view convolutional neural networks for 3D shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision: 945–953.

Sun Z, Xu Y, Hoegner L, Stilla U (2018) Classification of mls point cloud in urban scenes using detrended geometric features from supervoxel-based local contexts. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4 (2).

Tang P, Huber D, Akinci B, Lipman R, Lytle A (2010) Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. Automation in Construction, 19 (7): 829–843.

Theiler PW, Wegner JD, Schindler K (2014) Keypoint-based 4-points congruent sets–automated marker-less registration of laser scans. ISPRS Journal of Photogrammetry and Remote Sensing, 96: 149–163.

Thomas H, Qi CR, Deschaud JE, Marcotegui B, Goulette F, Guibas LJ (2019) Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision: 6411–6420.

Tombari F, Salti S, Di Stefano L (2010) Unique signatures of histograms for local surface description. In: European Conference on Computer Vision: 356–369.

Tsin Y, Kanade T (2004) A correlation-based approach to robust point set registration. In: European Conference on Computer Vision: 558–569.

Turkan Y, Bosché F, Haas CT, Haas R (2012) Automated progress tracking using 4d schedule and 3d sensing technologies. Automation in Construction, 22: 414–421.

Tuttas S, Braun A, Borrmann A, Stilla U (2015) Validation of bim components by photogrammetric point clouds for construction site monitoring. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, II-3/W4: 231–237.

Tuttas S, Braun A, Borrmann A, Stilla U (2017) Acquisition and consecutive registration of photogrammetric point clouds for construction progress monitoring using a 4D BIM. PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 85 (1): 3–15.

Varney N, Asari VK, Graehling Q (2020) Dales: a large-scale aerial lidar data set for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops: 186–187.

Vo AV, Truong-Hong L, Laefer DF, Bertolotto M (2015) Octree-based region growing for point cloud segmentation. ISPRS Journal of Photogrammetry and Remote Sensing, 104: 88–100.

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, Gorte B, Sithole G (2004) Change detection for updating medium scale maps using laser altimetry. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 34 (B3): 207–212.

Wang DZ, Posner I (2015) Voting for voting in online point cloud object detection. In: Robotics: Science and Systems, 1 (3): 10–15607.

Wang PS, Liu Y, Guo YX, Sun CY, Tong X (2017) O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM Transactions on Graphics (TOG), 36 (4): 72.

Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, Solomon JM (2018) Dynamic graph cnn for learning on point clouds. arXiv preprint arXiv:1801.07829.

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, Schmidt A, Mallet C, Hinz S, Rottensteiner F, Jutzi B (2015b) Contextual classification of point cloud data by exploiting individual 3d neigbourhoods. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3 (2015), Nr. W4, 2 (W4): 271–278.

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

Wen C, Yang L, Li X, Peng L, Chi T (2020) Directionally constrained fully convolutional neural network for airborne lidar point cloud classification. ISPRS Journal of Photogrammetry and Remote Sensing, 162: 50–62.

Wolf D, Sukhatme G (2004) Online simultaneous localization and mapping in dynamic environments. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, 2: 1301–1307 Vol.2.

Wu Y, Kim H, Kim C, Han SH (2010) Object recognition in construction-site images using 3d cad-based filtering. Journal of Computing in Civil Engineering, 24 (1): 56–64.

Xiao J, Adler B, Zhang J, Zhang H (2013) Planar segment based three-dimensional point cloud registration in outdoor environments. Journal of Field Robotics, 30 (4): 552–582.

Xiao W, Vallet B, Brédif M, Paparoditis N (2015) Street environment change detection from mobile laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 107: 38–49.

Xu S, Vosselman G, Elberink SO (2014) Multiple-entity based classification of airborne laser scanning data in urban areas. ISPRS Journal of photogrammetry and remote sensing, 88: 1–15.

Xu S, Vosselman G, Oude Elberink S (2015) Detection and classification of changes in buildings from airborne laser scanning data. Remote Sensing, 7 (12): 17051–17076.

Xu Y, Boerner R, Yao W, Hoegner L, Stilla U (2019a) Pairwise coarse registration of point clouds in urban scenes using voxel-based 4-planes congruent sets. ISPRS Journal of Photogrammetry and Remote Sensing, 151: 106–123.

Xu Y, Stilla U (2019) Contour extraction of planar elements of building facades from point clouds using global graph-based clustering. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4.

Xu Y, Stilla U (2021) Toward building and civil infrastructure reconstruction from point clouds: A review on data and key techniques. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 2857–2885.

Xu Y, Tong X, Stilla U (2021) Voxel-based representation of 3d point clouds: Methods, applications, and its potential use in the construction industry. Automation in Construction, 126: 103675.

Xu Y, Tuttas S, Hoegner L, Stilla U (2017) Geometric primitive extraction from point clouds of construction sites using vgs. IEEE Geoscience and Remote Sensing Letters, 14 (3): 424–428.

Xu Y, Tuttas S, Hoegner L, Stilla U (2018a) Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor. Automation in Construction, 85: 76–95.

Xu Y, Tuttas S, Hoegner L, Stilla U (2018b) Voxel-based segmentation of 3d point clouds from construction sites using a probabilistic connectivity model. Pattern Recognition Letters, 102: 67–74.

Xu Y, Ye Z, Yao W, Huang R, Tong X, Hoegner L, Stilla U (2019b) Classification of lidar point clouds using supervoxel-based detrended feature and perception-weighted graphical model. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, (99): 1–17.

Yang B, Dong Z, Liu Y, Liang F, Wang Y (2017a) Computing multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 126: 180–194.

Yang B, Zang Y (2014) Automated registration of dense terrestrial laser-scanning point clouds using curves. ISPRS Journal of Photogrammetry and Remote Sensing, 95: 109–121.

Yang J, Li H, Jia Y (2013) Go-icp: Solving 3d registration efficiently and globally optimally. In: Proceedings of the IEEE International Conference on Computer Vision: 1457–1464.

Yang Z, Jiang W, Xu B, Zhu Q, Jiang S, Huang W (2017b) A convolutional neural network-based 3d semantic labeling method for als point clouds. Remote Sensing, 9 (9): 936.

Yang Z, Tan B, Pei H, Jiang W (2018) Segmentation and multi-scale convolutional neural network-based classification of airborne laser scanner data. Sensors, 18 (10): 3347.

Yao W, Polewskia P, Krzystek P (2017) Semantic labeling of ultra dense mls point clouds in urban road corridors based on fusing crf with shape priors. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 42.

Ye Z, Xu Y, Huang R, Tong X, Li X, Liu X, Luan K, Hoegner L, Stilla U (2020) Lasdu: A large-scale aerial lidar dataset for semantic labeling in dense urban areas. ISPRS International Journal of Geo-Information, 9 (7).

Yin Z, Collins R (2007) Belief propagation in a 3d spatio-temporal mrf for moving object detection. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition: 1–8.

Yousefhussien M, Kelbe DJ, Ientilucci EJ, Salvaggio C (2018) A multi-scale fully convolutional network for semantic labeling of 3D point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 143: 191–204.

Yu Y, Li J, Wen C, Guan H, Luo H, Wang C (2016) Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 113: 106–123.

Zhang J, de Gier A, Xing Y, Sohn G (2011) Full waveform-based analysis for forest type information derivation from large footprint spaceborne lidar data. Photogrammetric Engineering & Remote Sensing, 77 (3): 281–290.

Zhang Z, Hua BS, Yeung SK (2019a) Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision: 1607–1616.

Zhang Z, Lan C, Zeng W, Jin X, Chen Z (2020) Relation-aware global attention for person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 3186–3195.

Zhang Z, Sun L, Zhong R, Chen D, Xu Z, Wang C, Qin CZ, Sun H, Li R (2019b) 3-d deep feature construction for mobile laser scanning point cloud registration. IEEE Geoscience and Remote Sensing Letters, 16 (12): 1904–1908.

Zhao R, Pang M, Wang J (2018) Classifying airborne lidar point clouds via deep features learned by a multi-scale convolutional neural network. International Journal of Geographical Information Science, 32 (5): 960–979.

Zhou Y, Tuzel O (2018) Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 4490–4499.

Zhu B, Liu JZ, Cauley SF, Rosen BR, Rosen MS (2018) Image reconstruction by domain-transform manifold learning. Nature, 555 (7697): 487–492.

Curriculum Vitae

Name	Rong Huang
Date/Place of birth	10.05.1994 in Hubei, China
Nationality	Chinese
Mailing address	Leonrodstr. 27
	Munich, Germany
Phone	+49-17643359015
Email	rong.huang@tum.de

Education/Internship

1999 – 2005	Elementary school in Xianning, Hubei, China.
2005 – 2008	Junior high school in Xianning, Hubei, China.
2005 – 2008	High school in Xianning, Hubei, China.
2011 – 2015	Study of Surveying and Mapping Engineering at Tongji University, Shanghai, China. Degree: Bachelor of Science
2015 – 2018	Study of Earth Oriented Space Science and Technology (ESPACE) at Technical University of Munich, Munich, Germany. Degree: Master of Science
Since 2019.03	Doctoral Candidate under the Supervision of Prof. Uwe Stilla at Technical University of Munich.
2018.04 – 2018.09	Research Assistant at Chair of Signal Processing in Earth Observation, Technical University of Munich.
2020.02 – 2020.08	Student Research Assistant at Chair of Photogrammetry and Remote Sensing, Technical University of Munich.

Acknowledgment

Looking back on the day I started my PhD study, to the completion of the doctoral dissertation, I have devoted myself to the development of point cloud processing techniques with great anticipation and passion for making them applicable in the construction industry. Although the task is challenging and my intelligence and energy is limited, I have tried my best and made all my efforts for achieving a very tiny contribution on this research topic. At this special time when I am about to finish my PhD study, I would like to express my great gratitude to the professors, colleagues, friends, and family who helped and supported me for surviving this period of time!

First and foremost, I would like to convey my deepest gratitude to Prof. Uwe Stilla, who supervises my doctoral work at the Technische Universität München, for giving me the chance to be one of his students and to purse my doctoral degree in his research team. I always feel lucky to be one member of this research group, because, during my PhD work, Prof. Uwe Stilla provided me many valuable suggestions in the academic studies and was also available for giving me help in my life. He also provided me lots of practical tips regarding the publishing process of scientific manuscripts, including how to give appropriate response to reviewer comments, how to structure the paper properly, how to give presentation, and so on. His help contributed significantly to my successful completion of the doctoral study. In addition, I would like to thank Prof. Helmut Mayer of Universität der Bundeswehr München for being the reviewer for my doctoral dissertation and spending time reading it carefully and providing many valuable comments, which help me significantly improve the quality of my dissertation. I would also like to express my gratitude to Prof. André Borrmann for his chairmanship of the examination committee and providing some beneficial suggestions from the perspective of construction industry.

Besides my supervisors, my sincere thanks also go to all my colleagues and friends, Dr. Richard Boerner, Dr. Tobias Koch, Dr. Sebastian Tuttas, Lukas Liebel, Philipp-Roman Hirt, Christian Albrecht, and Jingwei Zhu from both Chairs of Photogrammetry and Remote Sensing and Remote Sensing Technology, Dr. Markus Hebel and Joachim Gehrung from Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB. Here, I also would like to convey my special thanks to Dr. Yusheng Xu and Dr. Ludwig Hoegner for providing many valuable suggestions and discussions on my research topic and their great support in both my researches and my daily work. Besides, I would covey my thanks to Prof. Wei Yao and Prof. Pedram Ghamisi for providing me many helpful advises in the academic field. I would also like to express my thanks to my friends, Prof. Jian Kang, Prof. Li Fang, Dr. Ruoxin Zhu, Dr. Danfeng Hong, Chenyu Zuo, Shirui Wang, Xiangtian Yuan, and Yuxin Xie for their support, help, and expertness in their own research fields.

Last, but not least, I would like to thank my parents, thank you for always being proud of their daughter and keeping company with me. Only with their selfless love and unconditional support and encouragement, I can go through the tough period and successfully finish my study in Germany.