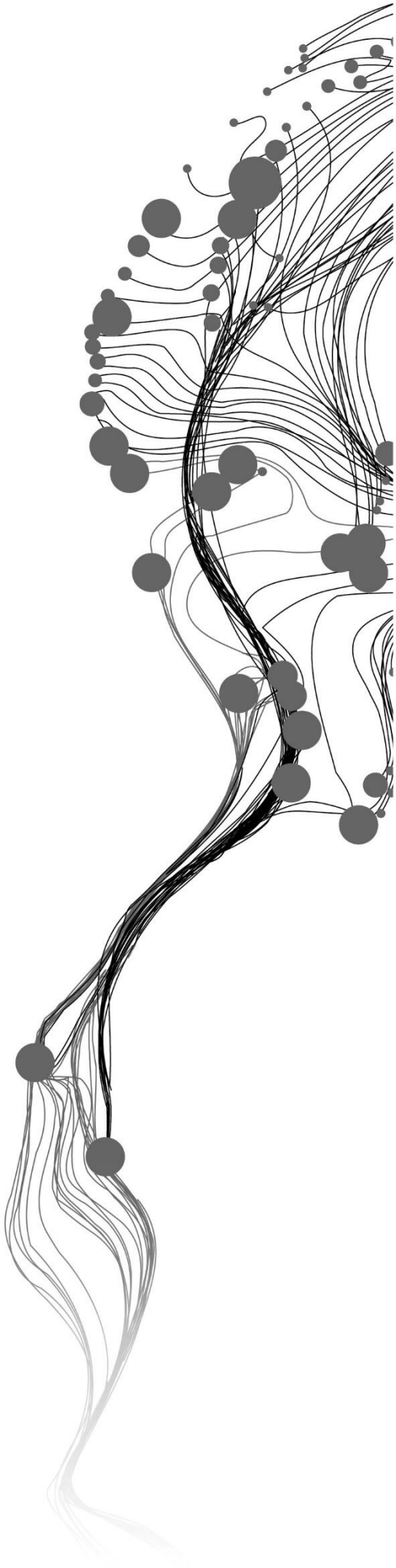
Scan-vs-BIM Automated Registration Using Columns Segmented by Deep Learning for Construction Progress Monitoring

GIRMA ZEWDIE TSIGE

June,2022

SUPERVISORS:
Dr. B. ALSADIK
Dr.IR. S.J. OUDE ELBERINK

External Advisor:
Dr.Martin Bueno Esposito, University of Edinburgh, UK



Scan-vs-BIM Automated Registration Using Columns Segmented by Deep Learning for Construction Progress Monitoring

GIRMA ZEWDIE TSIGE

Enschede, The Netherlands, June,2022

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.

Specialization: Geoinformatics

SUPERVISORS:
Dr. B.S.A. ALSADIK
Dr.IR. S.J. OUDE ELBERINK

External Advisor:
Dr.Martin Bueno Esposito, University of Edinburgh, UK

THESIS ASSESSMENT BOARD:
Prof.Dr.Ir.M.G. Vosselman (Chair)
Dr.Ing.Maarten Bassier (External Examiner , KU Leuven, Belgium)

ABSTRACT

In construction automation applications, registration between 3D Building Information Modelling (BIM) and the point cloud obtained by remote sensing technology is critical for smart monitoring of construction progress. However, in Architecture, Engineering, Construction & Facility Management (AEC/FM) context, automating such registration has limitations because building designs often contain a considerable self-similarity and symmetry, and the as-built point cloud can be incomplete and contain significant amounts of spurious data (outliers) from the construction site or lack texture. The existing AEC/FM software packages for 3D data registration implement coarse registration manually, which is less accurate and more time-consuming.

In this work, we present a novel automated column-based coarse registration method for 3D point clouds with the 3D BIM model from building construction sites. The method is based on the extraction of columns from the as-built point cloud and the 3D BIM model. For the point cloud data, fully automated column extraction techniques are used by applying deep learning, whereas columns are extracted from the structural-semantic component for the BIM model. We selected the point-based KPConv deep learning model and applied a transfer learning technique to semantically segment the main structural construction component. The inputs to the model are an unstructured set of points and the normal vector component of each point. The model is trained and tested with the published publicly available indoor point clouds, the point clouds from the real construction site, and the simulated point cloud. The trained model was evaluated on the test dataset and has achieved the mean intersection over union(mIoU) of 73% of overall classes and column detection accuracy of 69%. The achieved column detection result improved over the 16% mIoU in the developer's original work.

Then, we propose an automatic coarse registration method that is motivated by the Random Sample Consensus (RANSAC) algorithm.; we call it the 'congruent column sets' algorithm. This approach is presented to estimate the transformation parameters that best align the point cloud in the coordinate frame of the BIM model by matching the corresponding columns. Experiments were carried out on as-built point clouds acquired from the real building construction site using both terrestrial laser scan (TLS) and unmanned aerial vehicles (UAV) to validate the proposed method. The results show that our proposed column-based registration method achieved a rotation error of 0.02 degrees and RMSE of 0.12 meters for the TLS dataset and 0.03 degrees and 0.17meters for the UAV dataset. The computational time required is 55 seconds for the TLS dataset and 86 seconds for the UAV datasets. We conclude that our proposed approach contributes to automating the registration between the as-built point cloud and the as-planned BIM model to monitor the construction progress.

Keywords: BIM, Point cloud, Laser Scan, UAV, Deep Learning, Coarse Registration

ACKNOWLEDGEMENTS

First and foremost, I want to thank the almighty God for blessing me with the gift of life and good health. It would not have been possible to pursue and complete this program without the help of God.

I would take this opportunity to thank the Netherlands Government and the Orange Knowledge Programme (OKP) for the scholarship, which covers all financial costs and allows me to focus solely on academics.

My heartfelt thanks go to my first and second supervisors, Dr.B. Alsadik and Dr.Ir.S.J.Oude Elberink, for their guidance, encouragement, valuable suggestions, and insightful criticisms throughout the thesis work. It was a pleasure to work with you both, and I learned a lot.

I want to express my sincere thanks to my advisor Dr.Martin Bueno Esposito for his constructive feedback and valuable advice during the research phase. Thank you for taking the time to help me through my research work; it was fun to collaborate with you. Special thanks to Dr. Frédéric Bosché of the University of Edinburgh for allowing me to do my internship in his organization and provide data for my research.

I appreciate Tim Robert taking the time to assist me in obtaining the UAV images from the construction site. Many thanks to Mr. Ruben Nijhuis, the manager of Trebbe Bouw B.V construction company, for granting us permission to collect data and provide building design documents.

Most importantly, I am grateful to my family and friends for your love and support during these 24 months of my academic journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background and motivation

Monitoring work progress in building construction is crucial to enhance the performance of construction management: progress measurement(Arditi et al., 2015), material tracking (Scott & Assadi, 1999), and quality control (Arditi & Gunaydin, 1997). Conventionally, construction work progress can be tracked using daily or weekly reports collected by foremen(Scott & Assadi, 1999). The construction progress is then estimated by reading the collected daily work activities jointly with the design documents (i.e., 2D CAD drawings, bill of quantity, and specifications) (Turkan et al., 2012a). Then, site engineers use the construction schedule to retrieve the planned estimate of the construction activity by the exact date. This process of work progress estimation requires an intensive manual intervention which costs a lot of time and effort (Kiziltas & Akinci, 2005). However, with the evelopment of Building Information Modelling (BIM) and reality capture technologies, such as laser scanning and Photogrammetry in the Architecture, Engineering, Construction &Facility Management (AEC/FM) industry, construction progress monitoring activities are currently becoming more automated. This has a significant value in project management because it enhances the decision-making process allowing quick project delivery (Alsadik & Nex, 2021b; S. Kim et al., 2020).

The information needed to track the work progress of a construction project basically includes as-planned and as-built activities of the project. The main as-planned tools used for the building construction projects are the three-dimensional (3D) Computer-Aided Design (CAD) models, project specifications, and schedules. These are combined all together in Building Information Models (BIMs) (Turkan et al., 2012). The BIMs are semantically rich digital models currently popular in AEC/FM industry. BIM can enhance the collaboration between various parties involved in the different stages of the facility life (US General Services Administration, 2009). These as-planned documents constitute the primary information to track the forward flow of the designed concepts. On the other hand, the as-built construction site activities that convey feedback about the current state are usually obtained from the construction progress tracking activities which are currently becoming more automated using 3D sensing technologies(Omar & Nehdi, 2016a).

Recently, reality capture technologies, such as terrestrial laser scanners (TLS) and Photogrammetry, are becoming the common methods of as-built data acquisition on construction sites and operational facilities to help automation in the AEC sector(Bognot et al., 2018; Nahangi et al., 2018; Turkan et al., 2012b). The use of TLS for as-built BIM recording during construction is on the rise due to recent improvements in data capture speed and quality, as well as affordable equipment costs. TLS collects point clouds, which are 3D coordinates of the surrounding surfaces (see Fig.1.1.). Compared to conventional techniques, TLS point clouds provide very accurate information quickly and in large quantities. Representative research studies in this area have reviewed progress tracking techniques for building construction by comparing 3D models of actual progress data acquired by LiDAR and the as-planned 3D BIM model (S. Kim et al., 2020; Lin et al., 2015; Turkan et al., 2012b). However, as data acquisition using LiDAR requires the emission of laser beams, if a target has a high reflectance, the data quality decreases(Omar & Nehdi, 2016b). Besides, the high cost and limited applicability of LIDAR in complex indoor atmospheres are usually barriers to its applicability (Dai et al., 2013).

On the other hand, Photogrammetry is a strategy that generates a point cloud model by putting together digital pictures/videos captured by sensing technology (Camarillo et al., 2020). Currently, unmanned aerial vehicles (UAVs or drones) are becoming popular in construction site data acquisition. The UAVs have improved the level of autonomy, efficiency, and quality of reality capture, which increases the safety, accuracy, and speed of civil engineering inspections. They also allow the temporal monitoring of as-built changes over time(Alsadik and Nex2021; Greenwood et al., 2019). The quantity of work executed on construction sites for a particular time was estimated based on the 3D reconstruction of digital videos and

images acquired over the corresponding period(El-Omari & Moselhi, 2009; Omar & Nehdi, 2016b). Bognot et al. (2018) integrated a UAV image-based 3D point cloud model with the as-planed BIM model to track the building construction progress over time. According to their study, an as-built 3D model is generated from UAV images and carried out the construction progress monitoring in ArcGIS software by integrating the as-planned 4D BIM model with an as-built 3D model. The increasing autonomy and higher production rate benefits of reality capture using UAV lie in reduced costs, low risk, short time, and ease of data acquisition (McCabe et al., 2017).

Figure 1.1: Example of as-built construction site point cloud acquired by a laser scanner (Bosché,2012)

Construction Progress monitoring requires acquiring the as-built 3D construction site status and comparing it with the design 3D BIM model to retrieve useful as-built information(Bosche et al., 2008, 2009; Jacob-Loyola et al., 2021; Kim et al., 2013). This can be substantially simplified and automated by 3D registration of the as-built point cloud with the as-planed BIM model so that they have a common coordinate frame (Bueno et al., 2018a; J. Chen & Cho, 2018). 3D registration can be carried out by the two main consecutive steps: coarse registration to roughly align the datasets, followed by an automated fine registration process to optimally register them. The Fine registration of the 3D dataset is an extensively studied problem with well-known computational solutions; those are primarily variants on the fundamental approach of minimizing the Euclidean distance between neighboring points. The most popular approach is the Iterative Closest Point (ICP) algorithm(Besl & McKay, 1992) and its variants(Y. Chen & Medioni, 1992; Segal et al., 2009; Rusinkiewicz, 2019). ICP iterates by using a nearest-neighbor search to obtain point correspondence and using those correspondences to update transformation parameters by performing a local minimization of a non-convex error function using the least-squares method(Pottmann et al., 2006). On the other hand, coarse registration is a compelling step that is required to give an initial rough alignment between both datasets before applying the refinement registration using the ICP(Bosché, 2012). The coarse registration result should be sufficiently accurate for fine registration to provide more optimal 3D transformation parameters (Aiger et al., 2008).

Coarse registration of two 3D datasets can be performed when corresponding distinct features (highly dominant elements) are extracted and matched in both datasets. The researcher community makes efforts to develop geometric-based methods to automatically extract 3D distinct geometric features with strong candidate correspondences between both datasets. These approaches apply mathematical algorithms to segment point clouds into primitive shapes, such as lines(J. J. Jaw & Chuang, 2008b; Kaiser et al., 2022), planes(Bosché, 2012; Bueno et al., 2018a), cones, and cylinders(Ip & Gupta, 2007a). Their segmentation method begins with the extraction of multi-scale features, which describe surface roughness and curvature in the area surrounding each 3D point. The local points in the point cloud which have common geometric properties are clustered based on the geometric descriptors, such as planarity, linearity, verticality, or surface curvature(Hackel, Wegner, & Schindler, 2016). Then, different model fitting algorithms, such as RANSAC or Hough transform (Adan and Huber, 2011; Jung et al., 2016), are used to extract the segmented geometric features from both point cloud and model by making an assumption about typical structures based on a prior understanding. Then, the extracted geometric features from both the point

cloud and model are used as a corresponding candidate for the 3D transformation parameter estimation (Ip & Gupta, 2007a). However, the geometric-based feature extraction techniques require an intensive handcrafted feature extraction and are semi-automated.

Currently, fully automated feature extraction methods from the point cloud using deep learning are becoming state of the art. Deep learning is a subset of machine learning in which the algorithm learns patterns by analyzing labeled training data(Kaiming et al., 2016). Deep learning methods have become popular with large data availability and faster processing units such as Graphics Processing Units (GPUs). In computer vision, deep learning has progressed to produce state-of-the-art outcomes in semantic segmentation and recognition tasks(He et al., 2016). Deep learning methods have been adopted for the point cloud segmentation to separate the point cloud into several subsets according to the semantic meaning of the points (Qi, Su et al., 2017a). The evolvement of deep neural networks has achieved significant popularity, which eases the task of feature extraction by semantic segmentation of the as-built point clouds into the component building elements(Guo et al., 2021). In the AEC/FM context, the deep learning methods are used to semantically segment the as-built point clouds acquired from the building construction site into the building components, such as beams, columns, walls, slabs, etc. (Hu et al., 2019; Perez-Perez et al., 2021a; Thomas et al., 2019b; Zhao et al., 2021). Deep learning outperforms other supervised classical machine learning or mathematical algorithms because it eliminates the requirement to extract and select feature descriptors and offers a more robust non-parametric classification model (Chen et al., 2019). The availability of open-source trained models and the benchmark datasets for semantic segmentation of the as-built point cloud and their reliable segmentation accuracy made the extraction of the building components more feasible than the classical machine learning or geometric approaches.

This research proposes an automatic column-based coregistration approach to automatically align the asibuilt point cloud from the buiing construction site to the as-planned BIM model for construction progress monitoring application. The method uses deep learning to automatically extract columns from the as-built point cloud by semantic segmentation to match with the corresponding columns in the as-planned BIM model. The feature extraction using deep learning and automating coarse registration based on the matching extracted features is the major innovative part of the research, and there is no study done based on deep learning to extract the building component for the feature-based coarse registration. Columns were chosen as the main distinctive features for coarse registration because they are the most dominant building structural components usually constructed at early stages in the building construction process(Truong-Hong & Lindenbergh, 2022), and their center of mass can be easily calculated from geometric data. The proposed method is tested in two real-life case studies (TLS and UAV point clouds from the construction site), with experiments conducted at various levels of data completeness and orientation. A detailed explanation of our developed approach is found in the research methodology section.

1.2. Problem statement

Automated progress monitoring of a construction site where a BIM model is available is a challenging task. Accordingly, it requires the coregistration of the as-built point cloud with the as-planned BIM model. This basically consists of two parts: a coarse registration step to roughly register two 3D datasets followed by automated fine registration to optimize the alignment based on corresponding features(Dorai et al., 1994). The optimal solution using fine registration of two 3D point data is a well-explored problem with fully automated known solutions according to Iterative Closest Point (ICP) algorithm. The 3D coarse registration in the AEC/FM industry is currently implemented manually by picking the corresponding dominant points from the as-built point cloud and the 3D BIM model. However, this method is particularly time-consuming due to the large-scale point cloud obtained from the construction scene, with a significant amount of noise and clutter. It is true that browsing and visualizing point clouds to discover and pick locations of interest is tough, and points that are incorrect are prevalent. This may lead to circumstances where the achieved registration is insufficient for a future fine-registration phase to achieve an optimal alignment. Therefore, automated, reliable approaches for extracting highly distinct features in both datasets are necessary to speed up the correspondences' search and help achieve accurate alignment between both datasets.

Even though efforts have been made to automate the coarse registration process using various geometric-based approaches (further described in section 2.1), such as extracting corresponding lines, planes, or surfaces from both datasets, the mathematical segmentation algorithms used to extract geometric features require tedious handcrafted feature extraction and semi-automated. In addition to this, these approaches in the AEC/FM context are likely to fail due to the lack of textures, significant self-similarities and design symmetries in the building components, and the presence of significant noise and clutter from the construction site. So a robust, efficient, and completely automatic coarse registration process is still a research problem. Furthermore, there is a considerable research gap in applying deep learning to detect features for coarse registration.

1.3. Contributions

This thesis applies an automated column-based coregistration approach to align the as-built data from the construction site in the coordinate system of the as-planned BIM model to enable smart monitoring of indoor construction progress over time. The method detects the corresponding columns in both data sets as distinct corresponding features for coregistration. The expected major contributions from this thesis are (i) automatic detection of columns from the as-built point cloud by applying deep learning for semantic segmentation and (ii) automated coarse registration of as-built and as-planned datasets based on the detected corresponding columns for progress monitoring. The proposed automated co-registration method can be implemented on the as-built point cloud captured by 3D sensing technology, such as Photogrammetry or laser scanner from the construction site. The proposed automated cloud-model registration system is illustrated in Fig.1.2.

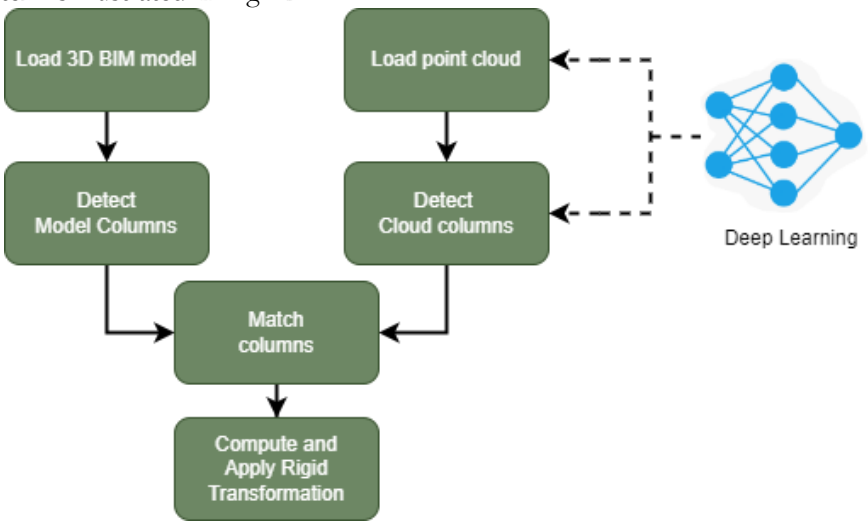


Figure 1.2:Schema of the proposed automated model-cloud coarse registration

1.4. Objective and research questions

1.4.1. Research objectives

In this research, we aim to develop an automated coregistration method by using deep learning for semantic segmentation to detect the construction columns in the as-built point cloud to match with the corresponding columns in the as-planned BIM model to automate construction progress monitoring.
The following sub-objectives will be addressed:

1. To automatically detect columns from the as-built point cloud by applying a suitable deep learning network for semantic segmentation.
2. To assess the column detection accuracy of the deep learning model and the mechanisms to improve the accuracy.
3. To develop a column-based coregistration between the as-built point cloud and the as-planned BIM model

1.4.2. Research questions

The research will mainly focus on the following sub-questions under each specific objective:

1. To detect columns from the as-built point cloud by applying a suitable deep learning network for semantic segmentation.
    - 1.1. Which deep learning network is suitable for semantic segmentation of indoor point clouds for column detection, and what are the determining criteria?
    - 1.2. What are the preprocessing techniques before loading the training data into the deep learning network?
2. To assess the column detection accuracy of the deep learning model and the mechanisms to improve the accuracy.
    - 2.1. Which accuracy metrics are suitable for assessing the model's column detection accuracy?
    - 2.2. To what extent can the semantic segmentation using deep learning help accurately detect the columns, and what are the mechanisms to improve the detection accuracy?
3. To develop a column-based coregistration between the as-built point cloud with the as-planned BIM model
    - 3.1. What are the appropriate tools to extract and convert columns from the as-planned BIM model to mesh/point cloud?
    - 3.2. Given the extracted columns in both datasets, how to identify their correspondences and then coregister the as-built point cloud in the coordinate system of the as-planned BIM model?
    - 3.3. What is the accuracy of coregistration using TLS and UAV point clouds, and what are the suitable techniques to evaluate the performance of the proposed registration?

2. LITERATURE REVIEW

2.1. Coarse registration of 3D datasets

3D coarse registration is a long-explored research problem, with most of the research developed on the registration of two or more-point clouds acquired by sensing technologies (J. Jaw & Chuang, 2008; J. J. Jaw & Chuang, 2008a, 2008c), and fewer studies on the registration of point clouds with 3D BIM models(Bueno et al., 2018b; Kaiser et al., 2022; Mahmood et al., 2020). Coarse registration of two 3D datasets can be performed when corresponding geometric features, such as points, lines, and planes, are identified and matched in the model and data points. The rigid 3D transformation parameters that coregister the two datasets are estimated based on the corresponding data association between two datasets. This is usually achieved by expressing the 3D rigid transformation problem as a closed-form solution (Horn, 1987) or nonlinear least-squares optimization model (Arun et al., 1987; Low, 2004). Rigid transformation can only be recovered once the correct correspondence between the minimum required pairs of distinct features is established. To extract such a limited number of distinct features with strong candidate correspondences, invariant local descriptors are typically used. Researchers have developed various approaches for the registration of multiple scans or scans to 3D BIM/CAD models.

2.1.1. Scan-to-Scan coarse registration

Multiple TLS scans belonging to a built environment scene should be registered to the target coordinate system to get a complete view of the facility. This can be done by extracting corresponding features from the overlapping region of the scan and matching them using one of the following approaches:

i. Manual Point-based registration

To apply this strategy, the user manually selects at least three corresponding key points, e.g., corner points, surveying control points, etc., in the datasets. Golparvar-Fard et al. (2009) employed control points to register two as-planned and as-built models in order to track building construction progress. Nguyen and Choi (2018a, 2018b) used an iterative closest point (ICP) to register point clouds from an industrial plant inspection after manually aligning an as-built model with an as-planned model. This method is not always accurate, mainly due to that navigating through the entire point cloud to pick the exact points of interest is a tedious and challenging task. Most often, these picked points result in inaccurate initial transformation results.

ii. Target-based registrations

This approach requires the user to place an easily recognizable scene object manually, often called targets, in the various scans. For the registration of the subsequent scans, at least three targets should be detected in the corresponding scans. For the recognized targets in each scan, the coordinates of their centroid are computed. The targets detected in both scans are matched, and 3D rigid transformation can be estimated (Akca, 2003).

iii. Geometric Feature-based registration

This technique focuses on the extraction of corresponding geometric/texture features, such as 3D points (Aiger et al., 2008; Johnson & Hebert, 1998), lines (J. J. Jaw & Chuang, 2008b), surfaces (Dold & Brenner, 2006) or combination of these(Bae, 2009; Ip & Gupta, 2007a) that can be automatically and effectively matched between target and source 3D point clouds. The distinct geometric features are identified and extracted from both scans using feature descriptors that specify the local geometry of point clouds(Han et al., 2018), which are then utilized to construct the transformation matrix. For instance, Mahmood and Han (2019)used a fast point feature histogram(Rusu et al., 2009) as a feature descriptor and a RANSAC

method to reject erroneous correspondences in aligning multiple scan 3D point clouds. Aiger et al. (2008) developed a '4-point congruent set' (4-PCS) algorithm that uses '4 coplanar point bases' as distinctive features for the alignment of target and source point clouds. Their method extracts all coplanar 4-points sets from a 3D point cloud that are approximately congruent to a given set of coplanar 4-points from the target point cloud. For each candidate's congruent sets, based on the correspondence information, they estimated the best rigid transformation parameters using the randomized alignment algorithm inspired by the RANSAC algorithm.

2.1.2. Scan-to-Model/Mesh coarse registration

Bosch (2012) presented some basic registration advantages and constraints in the context of the AEC/FM sector. The advantage is that the vertical (Z) axis of laser scans used for acquiring the as-built point cloud generally corresponds to the vertical axis of the BIM model. On the other hand, the building designs often contain several self-similarities between the simple geometrical elements (e.g., the dimension of the structural elements and the spacing between them are similar), which contributes to a challenge to the correspondence search during the feature-based registration. The other main challenge comes from the large portion of outliers (often 10% to 20%) in the as-built data, such as temporary objects, equipment, or people. These outliers significantly affect the registration process, and cleaning these noisy as-built point clouds is not efficient. Therefore, a coarse registration method should therefore be robust to these noisy data.

The same approaches used in scan-scan registration can also be used to accomplish 3D coarse registration of as-built point clouds to 3D BIM models, as the model can always be transformed into a point cloud. However, due to the lack of '3D textures' in the BIM model and the limitations explained in the above section in the AEC/FM context, most of the automated methods applied for the scan-scan registration are likely to fail. The most popular approach used to align the point cloud to the 3D BIM model is a geometric-based approach. The geometric features, basically geometric primitives, are extracted from both datasets using mathematical algorithms and matched to estimate the transformation parameters. Kim et al. (2011) proposed an automated coarse registration method suing PCA for the alignment of a 3D CAD model with the as-built point cloud for the construction progress monitoring. They first converted the 3D CAD model into a point cloud representation. Their method resampled both point cloud and CAD models to have a uniform point resolution without extracting corresponding geometric features. This method works in a simple construction site used by Kim et al. (2011). Ip and Gupta (2007) suggested an automated coarse registration based on the PCA. Their technique is based on segmenting both the as-planned 3D CAD model and the as-built point cloud into similar curvature surfaces

Bosché (2012) proposed a plane-based semi-automated coarse registration approach to extract and match planes from both as-planned and as-built 3D models. He first converted the 3D BIM model into mesh for extracting the plane patches based on the normal vector information. His method requires a user to pick at least three non-parallel planes from both datasets and computes the 3D rigid transformation using a least-squares alignment approach(Horn, 1987). Inspired by the 4PCS, Bueno et al.(2018a) proposed a 4-plane congruent set (4-PlCS) for the registration of as-built scanned data with the as-planned BIM mesh/model. The 4-PlCS developed by Bueno et al. (2018a) starts with extracting all the planar patches from both the 3D BIM model and point cloud. For each '4-plane base' of the BIM model point cloud, the matching congruent '4-plane bases' are searched as a candidate set within point cloud plane patches using geometric descriptors, such as parallelism, orthogonality, and distance. Then, the 3D rigid transformation matrix is computed for each pair of the 4-PlCS, and the optimal one is selected based on the maximum number of scores from the rest of the congruent sets.

Kaiser et al. (2022) proposed a 'line-in-plane' automated co-registration approach for photogrammetric point clouds with 3D building models. Their developed approach uses the BIM model in the IFC standard without the need to convert to a point cloud or mesh. Their method uses the structure from motion technology (SfM) to extract 3D line segments from the photogrammetric images and extract boundary

surfaces from the BIM model using the 'IfcSpace' entity to represent a single room and 'SpaceBoundaries' to represent a single bounding surface of the room. Their method considers only the planar boundary surfaces for matching with the extracted 3D line segments. The transformation parameters are estimated based on the correctly located 3D line segments on a corresponding planar boundary surface. As mentioned in their paper, their method extracts several possible 3D line segments which are not part of a boundary surface of interest, so a lot of preprocessing steps on the 3D line segments are required to get the correct matches. Another drawback of their method is that the SfM for the estimation of image orientation and point cloud reconstruction fails if the images captured from the built environment are not sufficiently textured. In addition to this, their developed method is tested on a very simple one-story building; however, in reality, the built environment contains multiple stories with symmetry and self-similarities within the building component.

2.1.3. Fine registration using ICP

The ICP (Iterative Closest Point) algorithm has been the most extensively used approach for fine-registration of three-dimensional structures after rough alignment using coarse registration. According to the ICP algorithm proposed by Besl and McKay (1992), every point in one data set is associated with the closest point in the other data set to generate correspondence matches. The nearest corresponding point is searched using the KD-tree data representation(Muja & Lowe, 2009). Then a point-to-point error metric is applied, which optimizes the sum of the squared distance between points in each matching set. The point-to-point metric is solved using the Singular Value decomposition (SVD) approach (Horn, 1987) Chen and Medioni (1992), on the other hand, employed a point-to-plane error metric, in which the aim of optimization is the sum of the squared distance between a point and the tangent plane at its matching point. Rusinkiewicz & Levoy(2001) have shown that the point-to-plane ICP algorithm has a faster convergence speed and gives more accurate results than the point-to-point ICP algorithm.

2.2. Feature extraction techniques from the construction site point cloud for registration

Point clouds captured using 3D sensing technology from the building construction site only contain 3D coordinates of the surrounding surface (often augmented with the RGB color). Anil et al. (2011) have explained the main factors which affect the extraction of building structural elements from the as-built 3D point clouds. Firstly, the point cloud density, which is defined as the average distance between points in Euclidean space, specifies the smallest object that can be represented. The density of the point cloud varies depending on the object's orientation and the reality capture device's range. This variation in point density affects the local nature and distribution of point clouds around the point (Dimitrov & Golparvar-Fard, 2015). Second, occlusions due to surface irregularity and neighboring objects can create missing data, making it difficult to cover the as-built information of the full scene. This is especially true in indoor areas where equipment and furniture are prevalent(Adan & Huber, 2011; Previtali et al., 2014). Besides partially occluded objects, some building components are entirely embedded in the walls and floors, such as columns and beams(Pătrăucean et al., 2015). Finally, measurement errors imposed by the point cloud captured with the reality capture device may have an impact on the quality of the extracted structural elements. Wang et al. (2019) also suggested the quantitative relationship between point cloud quality and the reliability of the segmented structural element for its intended purpose should be identified, as well as the standard requirements that are permitted for specific applications. The General service administration (GSA) of the USA (GSA, 2009) has developed a BIM guide for 3D imaging, which specifies the allowable standard requirements of the as-built BIM for various application domains.

Processing of these point clouds to extract various features of the scene for registration remains mainly a manual affair with its own challenges. The traditional methods to extract highly dominant features from the as-built point cloud apply handcrafted geometric features using a mathematical segmentation algorithm or classical machine learning. These methods are semi-automated or require intensive human interaction to some extent. However, currently, fully automated column extraction methods using deep learning are becoming state of the art. A detailed explanation of both approaches is found in the following section.

2.2.1. Conventional feature extraction techniques

a) Geometric based mathematical algorithms

This approach segments a point cloud into geometric primitive shapes of building elements such as planes, cylinders, and cones (Pu and Vosselman, 2009; Ruwen Schnabel et al., 2010). These methods are based on the assumption that primitive parametric models can represent the objects in a point cloud scene. To detect the objects, mathematical algorithms such as random sample consensus (RANSAC) or the Hough transform can be used(Adan and Huber, 2011; Jung et al., 2016). Jung et al. (2016) proposed an automatic approach to extract the interior building structures from a terrestrial point cloud. Their strategy began by making an assumption about typical interior structures based on a prior understanding of building components. Then, they integrated the constrained RANSAC and least square adjustments to extract the internal building objects from the point cloud. Semi-automatic software such as Realworks (Trimble), CloudCompare (Girardeau-Montaut, 2015), or 3D Reshaper (Technodigit) allows users to use plugins to extract geometric primitives such planes, cylinders, and spheres by fitting a plane to a 3D as-built point cloud data.

Another method presented was to use a region growing segmentation method for extracting structural and architectural components of the building by initializing the seed points at a certain point in the point cloud and forming a small section of a specific shape, such as a plane(Dimitrov and Golparvar-Fard, 2015; Romero-Jarén and Arranz, 2021b; Wang et al., 2015).Dimitrov and Golparvar-Fard (2015) provided a region-growing method for clustering patches of points that belong to individual building surfaces and then fitting surfaces and solid geometry objects suitable for the study. Their segmentation method begins with the identification of multi-scale features, which describe surface roughness and curvature in the area surrounding each 3D point. Following that, seed finding and region expanding stages are used to partition points that belong to the same region, resulting in a 3D as-built model of the indoor building environment. However, these methods often depend on prior knowledge of a point cloud class to extract the primitive elements from the point cloud. Others have devised a method for detecting a point cloud object using matching computer-aided design models from an AutoCAD library(J. Chen et al., 2018; Cho et al., 2014) or detection from as-designed BIMs (Turkan et al., 2012c). The latter methods, on the other hand, are inapplicable in situations where an as-designed 3D model of an existing facility (e.g., a building) is missing (e.g., a historical building scene).

b) Classical Machine learning techniques

Machine learning techniques have also been used in research on semantic segmentation of indoor point clouds to detect structural components, where computer models learn how to accomplish a task through supervised feature learning(Swetha Koppula et al., 2011). To learn a parametric model and classify objects based on the feature vector, supervised machine learning classifiers such as random forest or support vector machines can be employed(Weinmann et al., 2015). Huang and You (2013) proposed a framework for recognizing various objects in point cloud data, such as primitive shapes like walls and floors, using a Support Vector Machine (SVM). Swetha Koppula et al. (2011) used a Markov Random Field(MRF) to segment the indoor scene of a building semantically. They trained a graphical model which can capture various local features and contextual relations. They trained their method over 52 3D scenes and homes and applied the trained model to label the indoor scenes semantically. However, this technique requires manual handcrafted features which can characterize individual points in the point cloud.

2.2.2. Fully automated deep learning techniques

Many researchers have recently been working to identify a function that maps from a 3D point cloud to semantic labeling of building components using a deep learning model. In computer vision, deep learning has progressed to produce state-of-the-art outcomes in semantic segmentation and recognition tasks(He et al., 2016). Deep learning outperforms other supervised classical machine learning algorithms because it eliminates the requirement to extract and select feature descriptors and offers a more robust non-parametric classification model(Chen et al., 2019). Since our proposed approach applies deep learning, we review it in detail in the following sections.

a. Deep learning methods for semantic segmentation of point cloud

Guo et al.(2020) provide a comprehensive evaluation of deep learning algorithms for semantic segmentation of point clouds. According to their research, deep learning-based 3D point cloud segmentation needs information on both the global and local geometric structures of each point in the point cloud. Based on how these methods tackle the problem of irregularity in the point cloud, they grouped the semantic segmentation of point clouds using deep learning into four paradigms: projection-based, discretization-based, point-based, and hybrid methods. Fig.2.1 illustrates a classification of existing deep learning approaches for semantic segmentation of 3D point clouds.

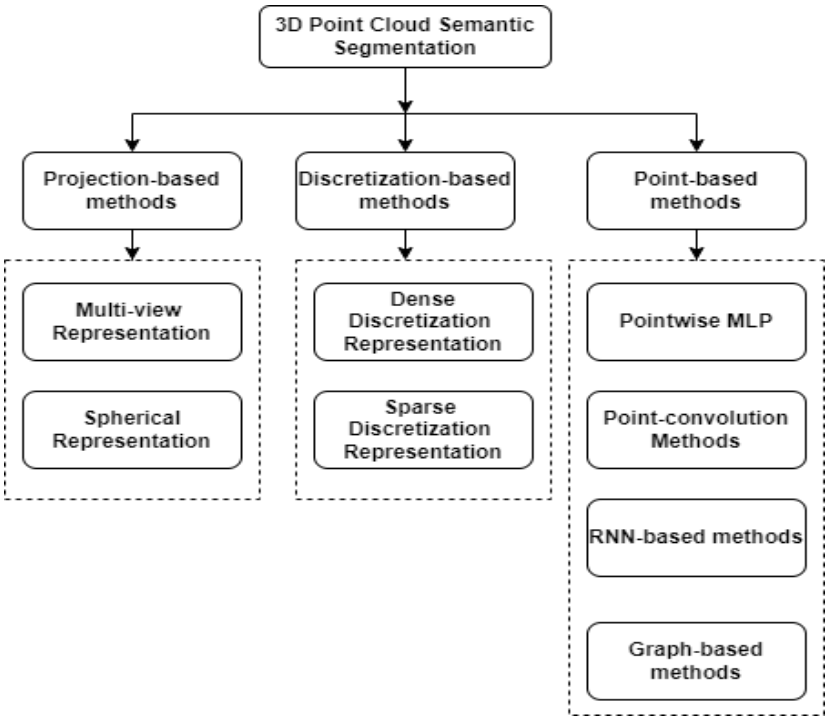


Figure 2.1: Classification of the existing deep learning methods for the semantic segmentation of point clouds. Source: (Bello et al., 2020).

As mentioned in Fig.2.1, there are several approaches for semantic segmentation of 3D point cloud, but in this work, we use only the point-based approach. Hence, we provide a dsetailed explanation of point-based semantic segmentation models for 3D point cloud in the following section:

Point-based methods learn the pointwise features directly from the unstructured point cloud. Point clouds are disordered and unstructured, making it impossible to apply ordinary CNNs as in the 2D images. So, many point-based methods are being proposed. In general, point-based methods can be generally divided into pointwise multi-layer perceptron (MLP) methods, point convolution methods, RNN-based methods, and graph-based methods (Guo et al., 2021)(see Fig.2.2).

Pointwise MLP methods make use of the shared MLP as the basic unit in their network architecture for its high efficiency. PointNet (Qi, Su et al., 2017)is the first method for deep learning on point-based feature learning of point clouds. The network utilized a multi-layer perceptron (MLP) with shared weights to learn and extend the feature space for each point. The global features of point sets are collected by all per-point local attributes in the neighborhood using an asymmetric function called max pooling. The global feature is linked to some other MLP and a classification prediction head for a classification problem. To acquire the final per-point label results for a semantic segmentation task, the global feature is concatenated with the feature of each point and two other MLPs. PointNet considered the problem of the point cloud invariance challenge through a learned rotation matrix T-Net to handle a rotation invariance challenge,

asymmetry function global pooling for permutation invariance, and it uses local coordinates instead of original coordinates to handle translational invariance. PointNet does not, however, incorporate the hierarchical local feature learning layer in the network design, which is one of its limitations. As a result, its ability to detect fine-grained patterns and generalize to complicated settings is hindered. To overcome this problem, several improved networks have been developed, such as methods based on neighboring feature pooling, attention-based aggregation, and local-global feature-based concatenation.

Neighboring feature pooling enables learning local geometric patterns for each point in the point cloud by aggregating the information from the local neighboring points. PointNet++(Qi, Yi et al., 2017b) presented a hierarchical approach to feature learning for point clouds, as shown in Fig.2.2(a). This network sample is a collection of keypoints using an iterative farthest point sampling method; group each keypoint with its surrounding points; and use PointNet to learn features on each keypoint. Their network architecture introduced two sets of abstraction layers that aggregate multi-scale information according to local point density variation. These layers capture the geometric pattern by integrating multi-scale information from local nearest neighbors to learn the feature for every point. To get a prediction for a classification task, the highest-level feature is linked with a fully connected layer. To obtain per-point classification results for a semantic segmentation problem, the point features are reversed and concatenated in a U-Net fashion. However, the point sampling method used in this method limited the scalability of the point cloud and was presumed to be computationally and memory inefficient for large-scale real point cloud semantic segmentation. M Jiang et al. (2018) proposed a PointSIFT module that incorporates both orientation and scale encoding units to capture information from eight different orientations through a three-phase ordered convolution. The scale encoding unit concatenates the muli-scale features to represent the 3D objects at various scales.

For large-scale semantic segmentation of point cloud scenes, Hu et al. (2019) presented RandLA-Net, an efficient and compact deep learning network. This framework utilizes a random point selection strategy rather than a neighborhood-based point sampling approach to overcome the problem of heavy computation and memory consumption. The author introduced the local feature aggregation (LFA) module to learn the hierarchical features of the point cloud to capture and preserve local features. Random sampling, on the other hand, may miss certain significant points, which can make feature extraction of input point clouds difficult. As a result, the authors proposed a U-Net-like design in which local features that were missed might be recreated using skip connections. This method was trained and tested on multiple benchmarks and demonstrated high efficiency and state-of-the-art performance.

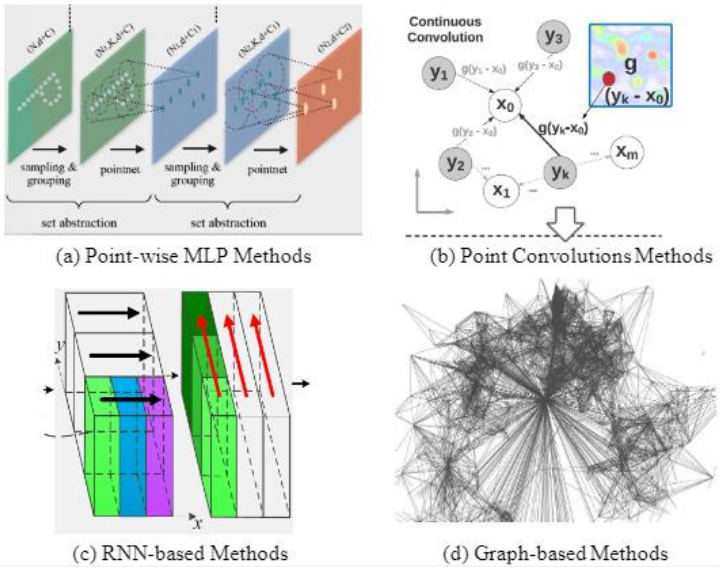


Figure 2.2: An illustration of point-based methods(Landrieu & Simonovsky, 2017; Qi, Yi, et al., 2017a; Wang et al., 2018; Ye et al., 2018)

Recurrent Neural Networks (RNN) have also been utilized for semantic segmentation of point clouds in order to capture inherent context information. Engelmann et al. (2017) first turned a block of points into multi-scale blocks and grid blocks to obtain input-level context. Then, the block-wise information retrieved by PointNet is successively fed into Consolidation Units (CU) or Recurrent Consolidation Units (RCU) to get output-level context. Li et al. (2018) developed a Pointwise Pyramid Pooling (3P) module to aggregate the coarse-to-fine local structure and then utilized two-direction hierarchical RNNs to further obtain long-range spatial dependencies (see Fig.2.2c). RNN is then applied to achieve end-to-end learning.

Graph-based techniques treat each point in a point cloud as a graph vertex, then build directed edges for the graph based on each point's neighbors. The graph structure is useful for modeling correlation between points, as the graph edges are directly depicted. As illustrated in Fig.2.2d, Landrieu and Simonovsky (2017) described a point cloud as a series of interlinked simple forms and superpoints, with the structure and context information captured using an attributed directed graph (i.e.superpoint graph). The problem of segmenting large-scale point clouds is then divided into three sub-problems: geometrically homogenous partition, super-point embedding, and contextual segmentation.

Point Convolution Methods propose effective convolution operators for point clouds. Hua et al.(2018) developed a pointwise convolution kernel, in which adjacent points are collected into kernel cells and then convolved with kernel weights. Thomas et al. (2019) developed a Kernel Point Fully Convolutional Network (KP-FCNN) based on Kernel Point Convolution (KPConv). Since our proposed method applies the KPConv(Thomas et al., 2019a) model for the column extraction, we explain this method in detail in the following section.

KPConv is a convolutional point-based semantic segmentation network motivated by an image-based convolution neural network(CNN), but instead of kernel pixels in the image convolution, as illustrated in Fig.2.3, the model utilizes a set of kernel points to describe the location where each kernel weight is applied. The points carry kernel weights, and a correlation function determines their region of effect to learn the point cloud's local geometric structure. Kernel points are points that are arranged in a regular way around each point in the point cloud. A weight matrix for each kernel point is applied to the point cloud's neighbors.

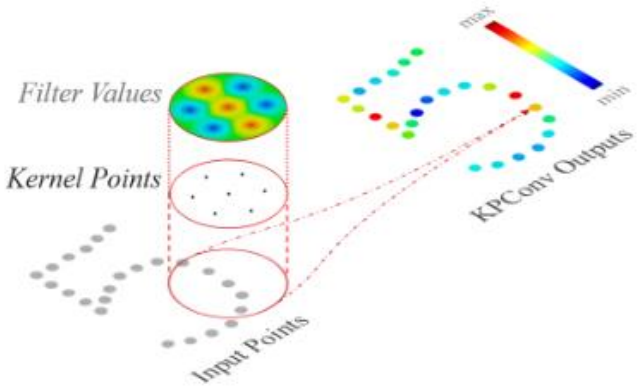


Figure 2.3: KPConv is depicted on 2D points. Source input points with a constant scalar feature (in grey) are convolved using a KPConv, which is defined by a set of kernel points with filter weights (in black) (Thomas et al., 2019a).

In CNN, every pixel is represented by a list of values known as 'channels,' which are subsequently processed by k filters. The new representation of the pixel is the dot product of the channels of a neighboring pixel by the filters. Similarly, in a KPConv layer, each point has features (similar to channels in CNN) and is multiplied by k kernel points (similar to filters in CNN). A new representation of the point is obtained by multiplying all of the kernel values by the neighbor's features. Each point in the point cloud has a KPConv layer enabled. It takes a point (location and features) and its neighbors (location and features for each neighbor) and generates additional features for the given point. The neighborhood of a

point is made up of all the points in a sphere with a fixed radius around it. The neighbors' locations are centered on the given point, similar to a convolution layer filter.
Several kernel points define a KPConv layer (similar to filters in CNN). The kernel points are arranged in a sphere around the given point in a regular pattern (see Fig. 2.4).

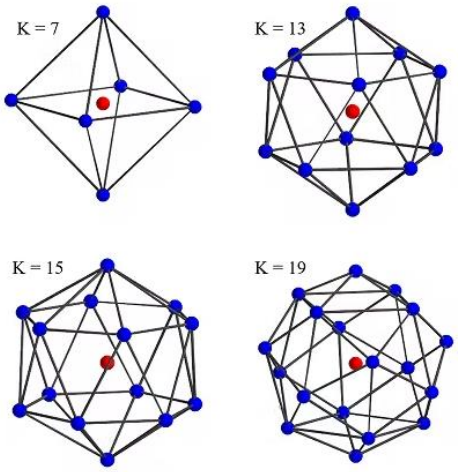


Figure 2.4: The blue points are kernel points, while the red point is the sphere's center, which is the given point of the KPConv layer (Thomas et al., 2019) .

Each kernel point has a set of weights (like filters) that are applied to nearby points. Each kernel point's weights are applied to nearby points in proportion to their distance from the kernel point. As a result, each kernel point has a correlation function that defines the area of influence. As illustrated in Fig. 2.5, the correlation function specifies whether a kernel point influences a point and how the neighbor point affects the calculation of the kernel value.

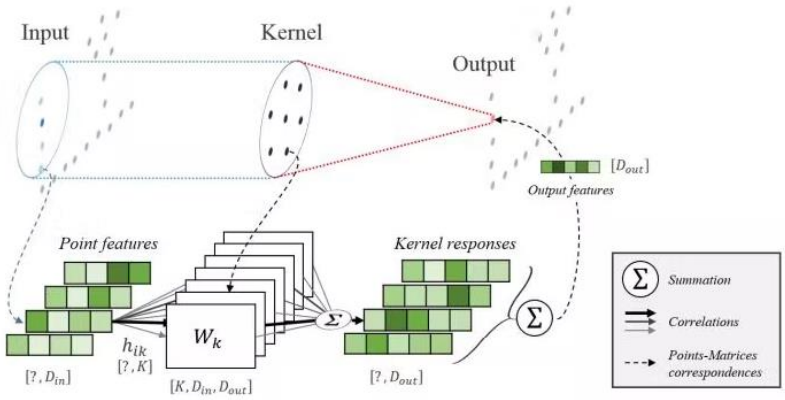


Figure 2.5: An illustration of a neighboring point to each kernel point correlation (Thomas et al., 2019) .

The new feature values of a given point are calculated by adding the features of all neighbors and multiplying them by their correlation and learned weights. At the input layer, the calculated features of each point are assigned to the point's attributes, such as RGB, intensity, and so on. It's similar to feeding RGB channels to a CNN in some ways. After the first operation of the KPConv layer on the point cloud, each point has new features that are used to calculate the next layer of KPConv. Multiple kernel points allow information from different directions of the processed point's surrounding to be distinguished. The regular arrangement of the kernel points results in uniform resolution in all directions. Like CNN, each kernel has a unique set of weights with the shape of $[f_{in}, f_{out}]$, where the f_{in} is the number of features in the current layer and f_{out} is the number of features in the next layer.

b. Benchmark datasets for evaluating a deep learning network

A significant number of datasets were collected to examine the performance of deep learning methods for various 3D point cloud applications. The datasets have been acquired by various kinds of sensors, such as Terrestrial Laser scanners (TLS), Mobile Laser Scanners (MLS), Aerial Laser Scanners (ALS), RGB-D cameras, and other 3D scanners. Some of the most used dataset lists for 3D semantic segmentation tasks are given in Table 2.1 below.

Table 2.1: A summary of existing datasets for 3D point cloud semantic segmentation

Name and reference	Year	#Points	#Classes	#Scans	RGB	Sensor	Environment
S3DIS	2017	273M	13	272	yes	RGB-D	Indoor
ScanNet	2017	-	20	1513	yes	RGB-D	Indoor
Semantic3D	2017	4000M	8	15	yes	TLS	Outdoor
Paris-Lille-3D	2018	143M	9	3	N/A	MLS	Outdoor
Toronto-3D	2020	78.3M	8	4	yes	MLS	Outdoor
SemanticKITTI	2019	4549M	25	23201	N/A	MLS	Outdoor
DALES	2020	505M	8	40	N/A	ALS	Outdoor

c. Model performance evaluating metrics

Overall accuracy (OA) and mean Intersection over Union (mIoU) are the most commonly used metrics for evaluating model performance in 3D point cloud segmentation(Armeni et al., 2016; Hackel et al., 2017). The IoU is calculated by dividing the area of overlap between the predicted segmentation and the ground truth by the area of union between the two, as illustrated in Fig.2.7. Intuitively, a successful prediction is one that maximizes the overlap between the predicted and ground truth. The mIoU, on the other hand, is the average of the IoU calculated for each category.

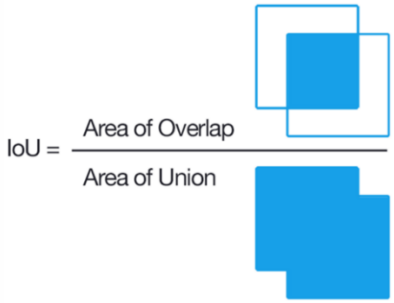


Figure 2.6: Illustration of the IoU metrics from the evaluation of the semantic segmentation model

IOU score ranges from 0 to 1, which represents the amount of overlap between the predicted and ground truth points in the point cloud in semantic segmentation.

Table 2.2: Table 2.2:Confusion matrix for the binary semantic segmentation problem. Source: (Chicco & Jurman, 2020)

		Predicted Condition	
	Total Point cloud =P+N	Positive (PP)	Negative (PN)
Actual Condition	Positive(P)	True positive (TP)	False negative (FN)
	Negative(N)	False-positive (FP)	True negative (TN)

In terms of the confusion matrix shown in table 2.2, the IoU can be rephrased in terms of true/false positives /negatives :

$$\boldsymbol{IoU = TP/(TP + FP + FN)} \quad (2.1)$$

Furthermore, the overall accuracy is calculated by taking the average of the accuracy of each class ratio of correct predictions to total predictions. However, accuracy metrics are not recommended when dealing with an imbalanced test set as it gives biased results. In terms of the confusion matrix shown in table 2.2, the accuracy metric can be rephrased in terms of true/false positives /negatives :

$$\boldsymbol{Accuracy = TP/(TP + FN)} \quad (2.2)$$

Based on our findings in the literature, we learned that the lack of textures, self-similarities, incompleteness and clutters in the construction site provides a considerable obstacle to achieving effective automatic coarse registration in the construction scene. The research community has made considerable efforts to extract corresponding features from both the as-built point cloud and the BIM model based on the geometric descriptors of the local point structure to extract corresponding features from the point cloud to estimate the transformation parameters. These methods are semi-automated or require intensive human interaction. On the other hand, recently, fully automated feature extraction methods using deep learning are becoming state of the art. We have noticed that there is a research gap in using deep learning methods for feature-based coarse registration between the as-built point cloud and as-planned BIM model. So, it is of interest to apply deep learning to detect columns as the most distinct features from the as-built point cloud and propose an automated alignment approach to match and align both datasets based on the detected columns.

3. STUDY AREA AND DATASET

3.1. Case study construction site

The building construction site where the experimental dataset was acquired is the new ITC building (a one-story building composed of steel and concrete columns, beams, and concrete slab) under renovation in Langezijds at the University of Twente (UT) campus in Enschede(Fig.3.1). It is a one-story building with a length of about 220m and a width of about 50m. This case study building is of interest because the construction is actively ongoing, and the structural components of the building were visible during the data acquisition phase. Moreover, the data acquisition was made possible as the building belongs to the University, and the large-scale construction site is more accessible for the flying of a UAV in the building for data collection.

Figure 3.1: The location of the case study building source: google earth

3.2. Required dataset

This research requires two basic datasets:

- The real building construction site 3D BIM model and as-built point cloud acquired by TLS and UAV for testing the developed automated registration method and
- Point cloud for training the deep learning model for semantic segmentation of the test data.

3.2.1. Experimental dataset from the building construction site

The developed method requires both the as-built point cloud and the as-planned BIM model of the construction site building to test the developed registration method. Both the TLS and UAV point clouds were acquired on the same date from the case study building construction site. The UAV point clouds (see Fig.3.3c) are reconstructed from UAV images of the ITC building under construction. The images are acquired by flying a UAV (DJI Phantom 4 pro v2) on the building construction site in collaboration with the construction company. The technical characteristics of the UAV camera are summarized in Table 3.1.

Table 3.1: Technical characteristics of the UAV camera (DJI Phantom 4 Pro v2) according to the manufacturer datasheet

Technical Characteristics	
Sensor	1-inch CMOS
Image resolution	20 megapixels
Field of view	84^0
Focal length	24mm
Mechanical shutter speed	8-1/2000 s
Max flight time	30 minutes

The images were acquired with a resolution of 1.5cm GSD with 90% front and side overlap. Then, dense point clouds were reconstructed using the commercial photogrammetry software Agisoft Metashape. We used several GCP targets mounted on the columns by the construction company to locally orient our UAV images/point cloud. The sensor type used to obtain the TLS point cloud is Riegl VZ-400i. A total number of 155 scan positions were used to cover the complete building interior. Table 3.2 summarizes the technical specification of the TLS device used to acquire the construction site point cloud. Fig. 3.3a illustrates the as-built point cloud acquired by the TLS.

Table 3.2: Technical specifications of the Riegl VZ-400i according to the manufacturer datasheet

Technical Characteristics	
Measurement range	1.5m to 600m
Ranging error (100 m, one sigma)	3mm
Beam divergence	0.3mrad
Field of view (vertical/horizontal)	1000/3600
Pulse repetition rate (points per second)	Up to 300,000
Laser wavelength	1550 nm
Minimum stepping angle	0.00240
Spot size	3 cm at 100m distance
Accuracy	5mm

The as-planned BIM model of the same building was provided by the University of Twente, ITC, and illustrated in Fig.3.2. The provided BIM model is in the IFC format. The open-source software tools and libraries, such as CloudCompare and Python, are used to preprocess the generated as-built point cloud, whereas we used REVIT BIM software for the visualization and querying of the elements of the as-planned BIM model.

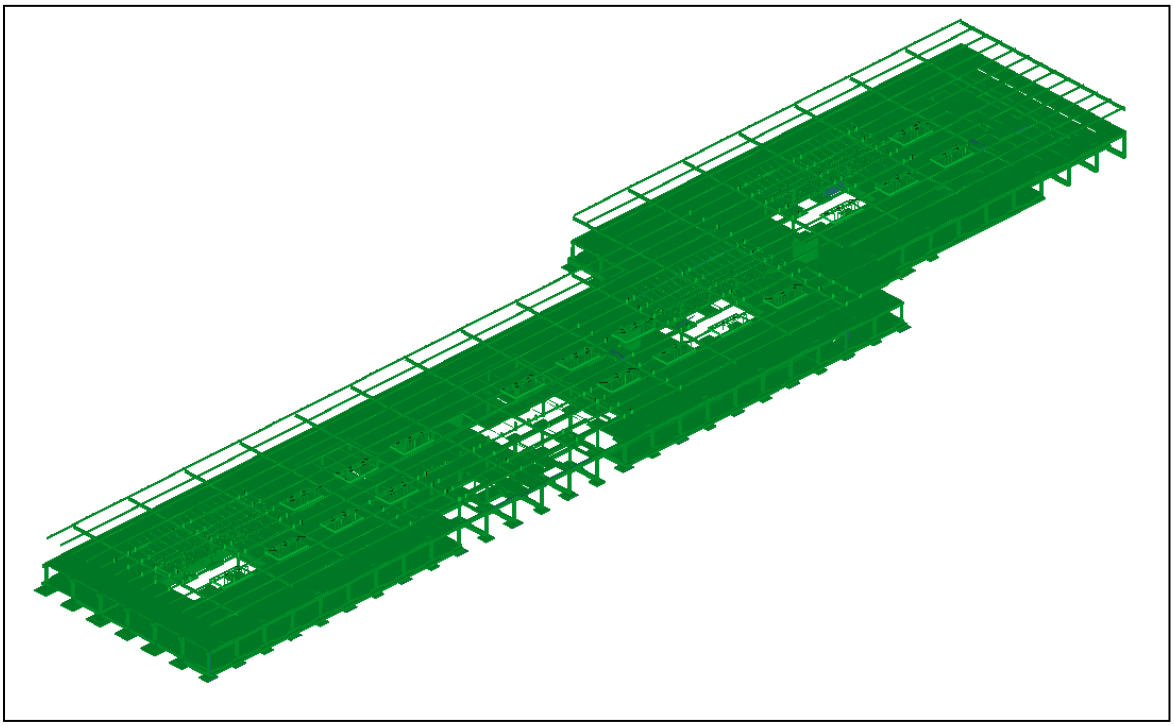
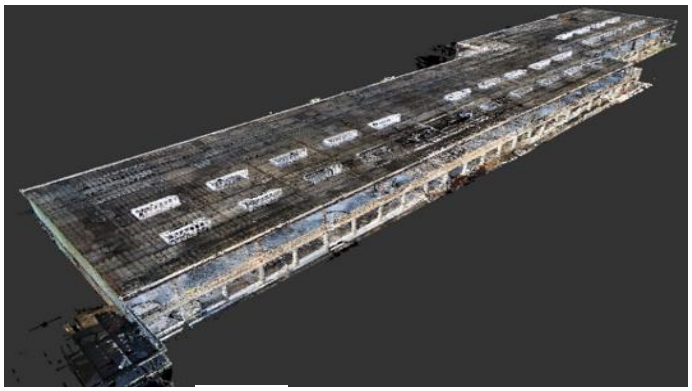


Figure 3.2:: As-planned BIM model of the case study of the ITC building, only the structural components of the BIM model are included.



a)

b)

c)

Figure 3.3::As-built point cloud of the new ITC building under renovation: A) TLS point cloud, b) & C) Point cloud reconstructed from the sequence of images acquired by flying a UAV (DJI Phantom 4 pro v2) Sept. 24 to Nov.02, 2021.

3.2.2. Dataset for training the semantic segmentation model

The datasets for training the deep learning model are collected from four sources: the construction site point cloud from the University of Waterloo, the publicly available published Raamac Lab dataset (Perez-Perez et al.,2021b), the S3DIS datasets (Armeni et al., 2017), and synthetic point cloud sampled from the BIM model exported from the Revit software (BIM model structural projects). The construction site point clouds from the University of Waterloo were provided by the host organization, the University of Edinburgh, School of Engineering. These point clouds are captured by terrestrial laser scanners at different stages of building construction. The S3DIS dataset is made up of 3D RGB point clouds, which contain six large-scale indoor areas with 271 rooms. Each point in the point cloud is annotated with one of the 13 semantic categories. For our semantic segmentation task, only the classes with the column, wall, beam, and ceiling/floor annotations were filtered out and post-processed. The Raamac Lab dataset was captured using laser scanners. They represent commercial and industrial buildings with ground truth data for column, beam, wall, ceiling, and other indoor industrial object labels, such as pipes. The synthetic point clouds were generated from the BIM model, which is already available in Revit software as a part of a sample BIM model for the structural project. The structural components of the BIM model (columns, beam, and floor) were retrieved and exported in mesh (STL) format from Revit. The exported mesh was post-processed for the unit conversion and synthetic point cloud sampling. Finally, we added 6 mm standard deviation Gaussian noise to the sampled synthetic point clouds to simulate the real behavior of the laser scanner. Fig.3.4 illustrates the examples of these point clouds used for the training purpose.

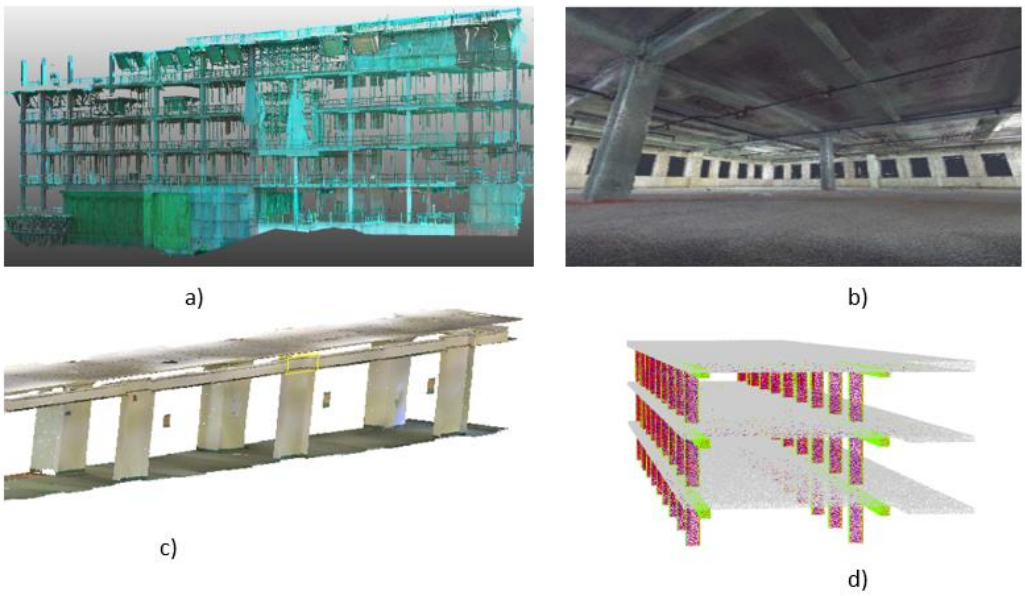


Figure 3.4::Examples of point clouds used for training and testing purposes: a) from the University of Waterloo, b) Raamac Lab dataset, c) S3DIS dataset, d) synthetic point cloud sampled from Revit BIM model.

4. RESEARCH METHODOLOGY

In this study, we propose a new method for registering the as-built point cloud with the as-planned BIM model for building construction progress monitoring. Noticeably, if the correspondences in the as-built point clouds and the as-planned BIM model are unknown, it is generally impossible to determine the optimal transformation parameters. Therefore, extracting highly distinct corresponding features in both datasets speeds up the correspondences' search and helps achieve accurate alignment.
So, this study proposes a column-based registration by extracting columns from the as-built structural construction point cloud to match with the corresponding columns in the BIM model. Columns are chosen as the basis for the registration because:

- Columns are the main structural components of all building types (concrete or steel building structure)
- They are constructed in the early stages of the structural construction, while other building components such as beams, floors, external and internal wall partitions, doors, and windows come after the column erection.
- They are usually distributed over the entire building layout.

The proposed method applies deep learning to semantically segment the as-built structural construction point cloud to detect the columns. Deep learning is chosen due to its capability to automate the column detection process and provides high accuracy when compared to the other classical machine learning and mathematical segmentation approaches because it eliminates the requirement to extract and select feature descriptors and offers a more robust non-parametric classification model(Chen et al., 2019). The method makes use of the semantic information available in the BIM model to retrieve the corresponding columns to match with the extracted column from the as-built data during the registration process. Once the corresponding columns are detected and identified in both as-built and as-planned models, the co-registration of the as-built point cloud will be done in the coordinate system of the as-planned model using the modified version of the RANSAC algorithm. This technique will be utilized to successfully handle column matching in the presence of outliers for the robust estimation of a transformation function.

Accordingly, the registration process of the as-planned BIM model and an as-built point cloud summarized in Fig.4.1 consists of four steps: (i) Data preprocessing, (ii) Point cloud Column detection, (iii) BIM model column detection, and (iv) Registring the as-built data with the as-planned BIM model. A detailed discussion of the research methodology is given below.

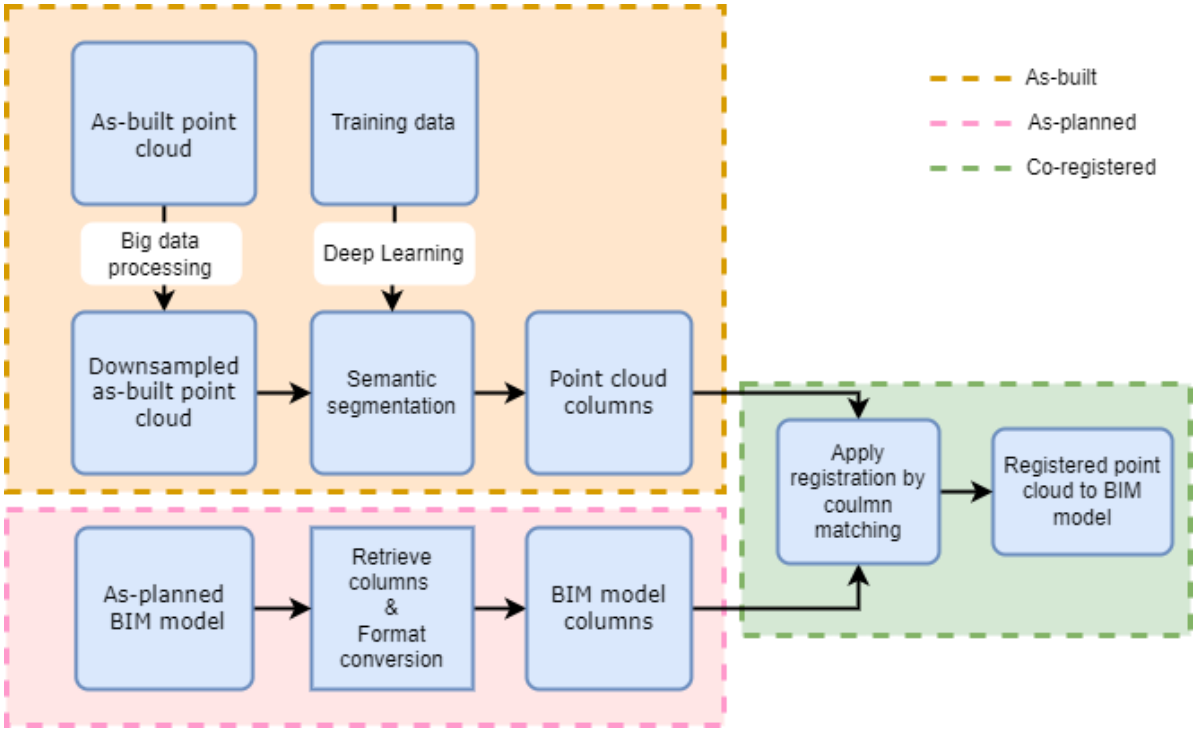


Figure 4.1: Schema of overall research methodology

4.1. Preprocessing the large-scale building construction site point cloud

The experimental data required for this research mentioned in section 3.2.1 are required to be preprocessed to efficiently handle the massive amount of the as-built point clouds acquired by TLS & UAV for the ITC construction site. Currently, ≅20 GB of TLS point cloud data (las file) and ≅4.4 GB of UAV-based point clouds (laz file) are available.

Accordingly, the TLS and UAV point cloud were subjected to the following preprocessing step to reduce the amount of data without losing the reliable representation. The open-source CloudCompare software was used for preprocessing the point cloud. Workstation computers with high processing and memory capacity (128GB RAM) have been used as the hardware. The preprocessing of the large-scale construction point cloud is summarized into:

- **Floor-based point cloud divisions**: Each floor has its own point cloud separated from the other floors to reduce the amount of data.
- **Subsampling:** Downsampling has been applied based on the 3cm spacing between the points in the point cloud for each floor point cloud.
- **Outlier removal**: The point cloud acquired by the UAV encompasses both the target object and the surrounding environment (e.g., trees, cars). As a result, the point cloud that belongs to the surrounding environment(e.g., trees, cars) has been eliminated to retrieve only the relevant component (indoor point cloud).

4.2. Point cloud column detection using deep learning

Among various point-based deep learning models reviewed in section 2.3.2, we selected a point convolution-based deep learning model for the 3D point cloud semantic segmentation task to extract the columns from the construction site point cloud. The selection process of the specific model is discussed in the base model selection section below. The proposed method applies transfer learning by customizing a pre-trained base model which has been trained on the S3DIS benchmark datasets (Armeni et al., 2017). Various techniques were applied to collect and preprocess point cloud data for training and validating the selected base model. The training data was labeled into five (5) interest classes: Column, Wall, Beam, Slab, and Clutter. The detailed description of the classes and the quantity of prepared training data for each

class is illustrated in Table 4.3. Then, the pre-trained base model was modified and trained on our training dataset to semantically segment as-built point clouds from the building construction site to classes of our interest. Finally, the performance of the trained model was tested using a suitable accuracy metric. A detailed description of the proposed method for semantic segmentation can be found in the following sections.

4.2.1. Transfer learning and fine-tuning

Transfer learning is a method of leveraging feature representations from a pre-trained model and applying them to a different but related problem(Torrey & Shavlik, 2010). Pre-trained models are typically trained on large datasets, which are a common benchmark in the computer vision field. The weights learned by the models (especially those of the first layers) can be used for various computer vision applications. These models can be used to make predictions on new tasks directly or as part of the training process for a new model. When pre-trained models are used in a new model, the training time and generalization error are normally reduced. Transfer learning is very important when the available training datasets are small. In this situation, weights from the pre-trained models are used to initialize the weights of the new model. Building a new model and training from scratch on the small dataset would most likely lead to over fittings. However, transfer learning will fail when the high-level features learned by the later layers of the network are insufficient to discriminate the classes in the semantic segmentation task (Yosinski et al., 2014). The other scenario where transfer learning is not applicable is when the datasets are very dissimilar; then, all features may transfer poorly.

Fine-tuning the network is the process of adjusting the parameters, such as the learning rate, the number of epochs, the optimizer, and the regularization parameters to achieve the best possible results. Fine-tuning requires unfreezing the entire pre-trained model (or part of it) and training it on the new data with a very low learning rate. However, since the entire model needs to be retrained, the model is also likely to overfit. The proposed transfer learning workflow for the semantic segmentation of an as-built point cloud using deep learning is summarized in Fig.4.2.

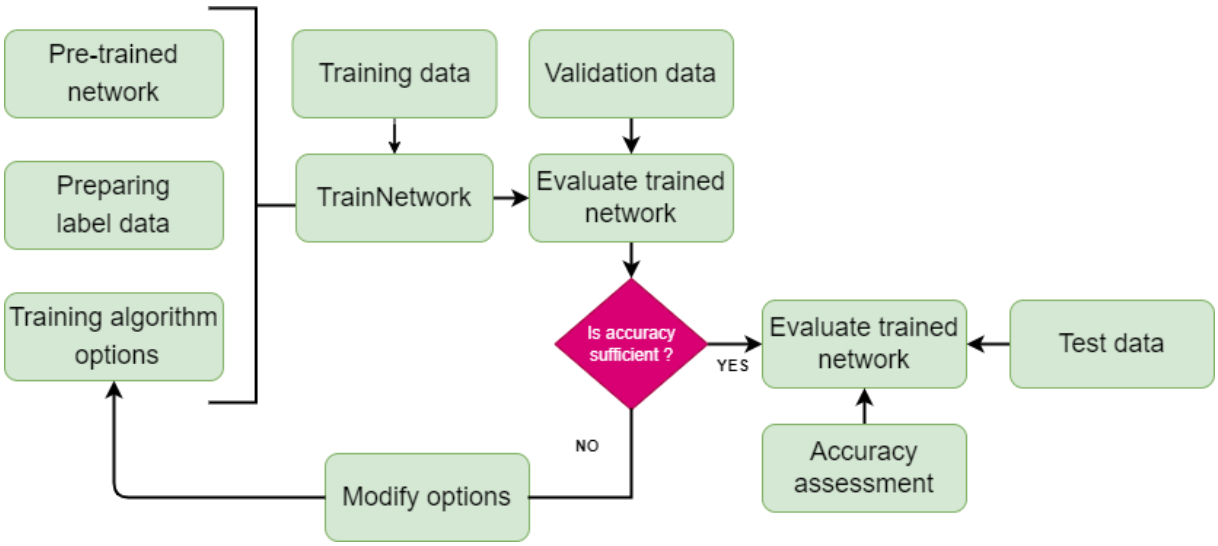


Figure 4.2:Transfer learning workflow for semantic segmentation of point cloud using deep learning

The following section will give a detailed description about the model selection,and training data preparation

a) Base model selection

The point convolution-based deep learning model was selected based on the literature review in section 2.3.2. For the pre-trained model selection for the task of 3D point cloud semantic segmentation, the machine learning extension of Open3D (Zhou et al., 2018) called Open3D-ML was used. This repository was chosen because it integrates various state-of-the-art deep learning models for the 3D point cloud

semantic segmentation task. Table 4.1 summarizes the available trained models and datasets integrated into the Open3D-ML for the semantic segmentation tasks and the respective scores based on the mean intersection-over-union (mIoU) over all classes.

Table 4.1:Performance of various models on different benchmark datasets for the task of semantic segmentation. Source: Open3D-ML GitHub repository (Q.-Y. Zhou et al., 2018)

Model / Dataset	SemanticKITTI	Toronto 3D	S3DIS	Semantic3D	Paris-Lille3D	ScanNet
RandLA-Net (tf)	53.7	73.7	70.9	76.0	70.0*	-
RandLA-Net (torch)	52.8	74.0	70.9	76.0	70.0*	-
KPConv (tf)	58.7	65.6	65.0	-	76.7	-
KPConv (torch)	58.0	65.6	60.0	-	76.7	-
SparseConvUnet (torch)	-	-	-	-	-	68
SparseConvUnet (tf)	-	-	-	-	-	68.2
PointTransformer (torch)	-	-	69.2	-	-	-
PointTransformer (tf)	-	-	69.2	-	-	-

(*) Using weights from the original author

Among the benchmark datasets mentioned in Table 2.1, we have chosen the pre-trainessd model on the S3DIS (Armeni et al., 2017) dataset because it includes some of the classes of our interest, such as column, beam, wall, and slab. The S3DIS is made up of 3D point clouds from six floors in three different buildings, divided into individual rooms. These datasets are semantically annotated into thirteen (13) classes containing both the indoor structural and non-structural components, such as ceiling, floor, wall, column, chairs, tables, etc. As seen in Table 4.1, RandLA-Net (Hu et al., 2019) and Point Transformer (Zhao et al., 2021) have achieved the state-of-the-art performance (70% mIoU) on the S3DIS dataset, followed by the KPConv model (Thomas et al., 2019a) which has relatively lower performance 65% of mIoU in overall classes. We have loaded the pre-trained weights of these trained models to adapt to our semantic segmentation task using the proposed transfer learning approach. Unfortunately, for the RandLA-Net and Point Transformer models, the available trained weights were not complete. As a result, it was not possible to adopt our proposed transfer learning approach to these models. So, we selected the KPConv model, a point-based semantic segmentation model, due to the availability of the pre-trained weights on the S3DIS dataset. The semantic segmentation architecture of KPConv is built with convolutional blocks, designed like bottleneck ResNet blocks (Targ et al., 2016), as illustrated in Fig. 4.3. The KPConv model was trained on the S3DIS indoor dataset to semantically segment the 3D point cloud into thirteen classes. We modified the output layer of the pre-trained model into five (5) classes of our interest and trained again on our dataset. A detailed explanation of the KPConv model was given in section 2.2.2

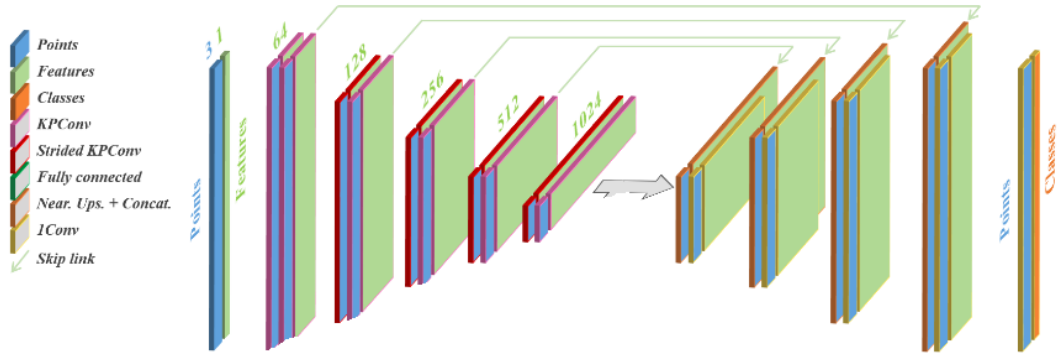


Figure 4.3: The adopted KPConv network architecture for semantic segmentation of 3D point clouds from the construction site.

As illustrated in Table 4.2, the developers of the KPConv model (Thomas et al., 2019a) provided the semantic segmentation IoU scores of each of the classes which are common to our interest classes.

Table 4.2:Semantic Segmentation IoU scores on S3DIS for only five classes Source: Thomas et al.(2019)

Class name	Ceiling	Floor	Wall	Beam	Column
IoU score (%)	92.6	97.3	81.4	0.0	16.5

This shows a very low performance of their trained model for predicting columns and beams relative to the other common classes (ceiling and wall).

b) Training data preparation

The training data described in section 3.2.2 was preprocessed before being loaded into the network. The preprocessing involves various techniques to clean and organize the raw data to make it suitable for the deep learning model. The main preprocessing techniques used were outlier removal, downsampling, additional attribute extraction, and labeling training data. We used Python libraries, such as NumPy, Pandas, and Open3D, jointly with MATLAB and CloudCompare (Girardeau-Montaut, 2015). The main preprocessing activities used for the training data preparation are described in the following section:

Subsampling: We have various point clouds with varying point densities from different sources. So, we subsampled a point cloud using a voxel size equal to 3cm to get a uniform point density in all point clouds.

Additional attribute extraction: to help the model learn the essential patterns of the geometric data, features play a major role in addition to the geometric components. Most of the collected point clouds include attributes, such as color and reflectance values, in addition to the geometric coordinates. However, since we are interested in the semantic segmentation of the structural building components, we argue that colors and intensity attributes don't help distinguish our classes of interest. Accordingly, the color and other additional scalar fields that came with the raw point cloud are removed. Instead, the normal vectors for each point in the subsampled point cloud were computed and added as an additional attribute for the entire point cloud used for training, validation, and testing. The normal vectors are computed for the downsampled point cloud by setting the maximum size of the point's neighborhood and radius equal to 50 points and 20cm, respectively.

Labeling training data: the training dataset was prepared by labeling the collected point clouds into four structural components of the building, i.e., column, beam, slab, and wall, using CloudCompare software. The rest of the point clouds, which do not belong to the four classes, such as formworks, safety fences, construction equipment, and people, are labeled under the 'clutter' category. Then, the dataset was split into a training, validation, and test using the 70:20:10 ratio from the original dataset. The model was

trained using the training dataset and evaluated using the validation dataset. A description of the classes and data used for training, validation, and testing are shown in Table 4.3.

Table 4.3: Description of the number of points per class used for training and testing the architecture

Data type	# Column points	# Wall points	# Beam points	# Slab points	# Clutter points	Total
Training	6.5M	14.4M	7.7M	40.1M	0.4M	69.1M
Validation	1.9M	4.1M	2.2M	11.5M	0.2M	19.9M
Testing	0.9M	2.1M	1.1M	5.7M	0.1M	9.9M

4.3. BIM model column detection

Column detection from the as-planned BIM model consists of retrieving columns from the BIM model and converting the format.

Retrieving columns from the BIM model: The original BIM model was in Industry Foundation Classes (IFC) format. The BIM model software (Revit) was used to retrieve the column components from the BIM model. First, the BIM model was imported into the Revit software in IFC format, and then columns were filtered out based on their floor location from the structural components of the BIM model.

Convert the format of the retrieved columns: Our proposed method requires both the BIM model and point cloud should be represented in a point cloud format. In order to use the 3D geometric information stored in the BIM model, it is easier to be converted it into a point cloud format. Accordingly, the retrieved columns (IFC format) were exported in Stereolithography (STL) format from the Revit software. The STL or OBJ format is a 3D data representation consisting of normal and triangular facets. This format allows a BIM model to be converted without losing significant information (Bosche et al.,2008). Uniform resolution point clouds can then be generated from the triangulated facets. We sampled uniform resolution point cloud for the columns using CloudCompare software.

4.4. Preprocessing the detected columns

The detected columns from both the point cloud and BIM model were subjected to the following preprocessing techniques before being loaded into the proposed registration algorithm:

i. Cleaning the detected point cloud columns:

This process only applies to the columns extracted from the point cloud using semantic segmentation. As will be discussed in section 5, the columns extracted from the semantic segmentation model contain some misclassified points from other classes, especially walls and clutters. So, these misclassified points (outliers) contribute to the bias in the estimated column centroid in the later stage. Since the outlier points are scattered in a small group further from the column clusters, we selected an outlier filtering technique that removes the outliers based on the distance from the column points. For this task, we used the 'Statistical Outlier Filter(SOR)' tool in CloudCompare to remove the outlier points which are far from the column points. This tool removes the points far from their neighbors based on the mean distance and standard deviation multiplier threshold from the given point. Below we elaborate on how this algorithm works :

Geven the extracted column points from the semantic segmentation result, select N points used for the calculation of the mean distance from the given point O. Let d_i represents the distance from the given point to each of the selected points. The mean distance between these points can be calculated by:

$$\bar{d} = \frac{\sum_{i=1}^{N} d_i}{N} \tag{4.1}$$

The standard deviation is calculated by:

$$\delta = \frac{\sum_{i=1}^{N} \left(d_i - \bar{d}\right)^2}{N} \tag{4.2}$$

The distance threshold that the algorithm accounts for the outlier removal is calculated by:

$$d_s = \bar{d} + n * \delta \quad (4.3)$$

Where n is the standard deviation multiplier, if the chosen value of n is low (1 or below 1), then the filter would be much more aggressive and removes many points around the selected point. On the other hand, if the assigned value for n is high, then the filter will be less aggressive. Based on the above calculation, the SOR considers all points with $d_i \geq d_s$ as outliers and removes them, whereas the points with $d_i < d_s$, remains as the column points. We set the optimal parameters for standard deviation multiplier n and the number of points N to remove these outliers for the detected point cloud columns.

ii. Clustering the detected columns:

This process is required to cluster the column instances from the entire patch of detected columns from both datasets. For this task, we used a clustering tool called 'Label Connected Component' in CloudCompare. This tool divides the selected cloud(s) into smaller parts separated by a minimum distance. Each part is a connected component (a group of 'connected' points). This tool uses two parameters to extract the connected components of the point clouds: the minimum number of points and the octree level.

Octree level: CloudCompare extracts the connected components using a 3D grid. The octree structure is used to generate this grid. The octree level specifies how small the minimum gap between two components should be (the corresponding cell size is displayed next to the level). The higher the level, the narrower the gap (so the more components can be extracted).

Minimum points per component: Components with fewer than the number of points specified will be ignored (useful to remove the smallest components).

4.5. Proposed registration method

The registration problem is to find the global 3D rigid similarity transformation parameters that best align the as-built point cloud into the same coordinate system as the as-planned BIM model to track construction progress over time. We assume that this transformation is rigid, so we do not consider the deformations inside the point cloud (no shear deformation). The 3D transformation can be made along three orthogonal axes, and is characterized by seven unknown 3D similarity transformation parameters, i.e., three rotation, one scale, and three translation parameters, are to be estimated. However, we expect a similar scale in the two data sets (the scale along all the three directions is equal to unity), assuming the BIM model scale is known, and the as-built point cloud is also scaled into reality. The registration problem is to find the correct corresponding columns in both data sets and then estimate the unknown 3D similarity transformation parameters.

The building construction context has some unique benefits that can be taken advantage of during the initial estimation of transformation parameters, but it also has some unique limits that must be addressed (Bosché, 2012). The advantage is that the as-built point clouds are reconstructed with the vertical axis (Z) orthogonal to the ground (i.e., the axis along which the earth's gravitational force is exerted) this axis often correlates to the design 3D (BIM) model's vertical (Z) axis. So, based on this argument, our developed method assumes that the rotation mainly occurs along the vertical direction, and the rotation along the horizontal (X, Y) axis is assumed to be negligible. On the other hand, construction site as-built data are usually taken in noisy areas with a significant amount of clutters that are not part of the structure under target. These components cause occlusion that creates gaps in the as-built data acquired from the scene of interest. Consequently, the coarse registration algorithm should take advantage of this vertical axis symmetry and be robust to such a noisy as-built point cloud.

We propose an automated coarse registration approach which is motivated by the randomized alignment approach first developed by (Fischler & Bolles, 1981). The motivation behind this approach is to apply a

trial and error coregistration approach that is robust to the outliers in the detected columns of the as-built point cloud from the semantic segmentation. The outliers should be detected and removed from the process of estimating transformation parameters. The key idea is to find the best partition of points in the inlier set and outlier and estimate the transformation model from the inliers set. Our proposed method is summarized in Fig.4.4, which is explained below. The method assumes that both the model and point cloud columns have previously been detected.

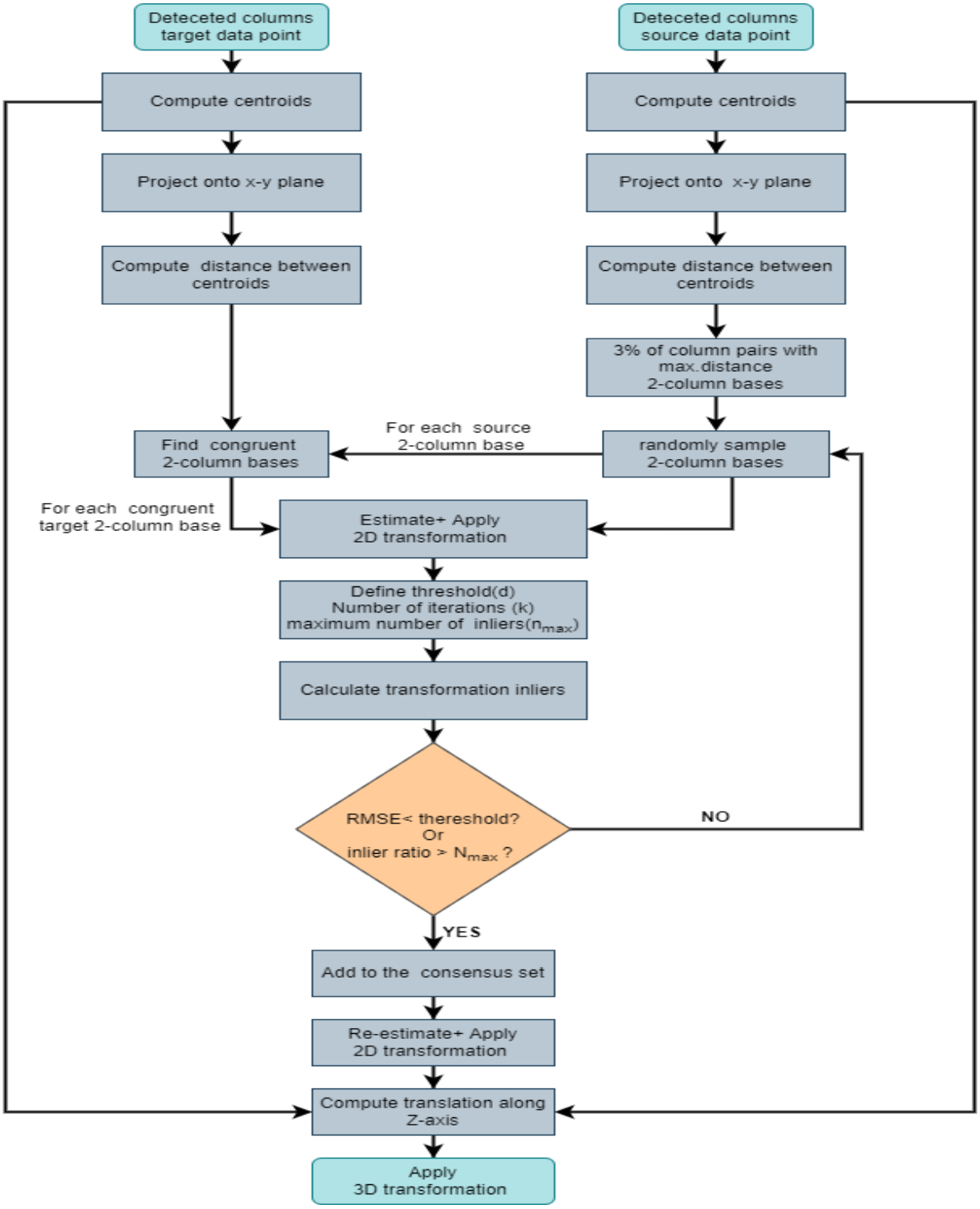


Figure 4.4: Workflow of the proposed coarse registration algorithm

i. Problem setting

Given the BIM model mesh (target data points) and as-built point cloud (source data points) in the initial arbitrary coordinate frame, the registration problem is to find the 3D rigid similarity transformation parameters that best align the source data points in the coordinate system of the target data points. Initially, the two datasets are misaligned because they are placed in different local coordinate frames. Accordingly, we use the detected columns from both datasets as a distinct feature to estimate the optimal 3D rigid similarity transformation.

Our approach starts with computing the 3D centroids of the detected columns for both data points to make use of the center points as a basis for the correspondence search for estimating the 3D transformation parameters. The method first determines the 2D optimal transformation parameters, then is followed by estimating the vertical translation vector using the center of mass of both datasets. The 2D coarse-registration is done in the horizontal XY plane to obtain four unknown parameters out of the 6-parameters of the 3D rigid similarity transformation(in our case, scale is assumed to be unity). If (x,y) represent the centroids of the source (as-built point cloud) detected columns coordinates, and (u,v) represent the centroids of the target (BIM model) detected columns coordinates, the 2D similarity transformation is represented by a homogenous matrix A:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.4}$$

where $\mathrm{A} = \begin{bmatrix} \mathrm{a} & -\mathrm{b} & \mathrm{c} \\ \mathrm{a} & \mathrm{b} & \mathrm{d} \\ 0 & 0 & 1 \end{bmatrix}$

Where a and b are elements of the scale and rotation part, and c and d are the elements of the translation vector. The rotation and scale parameters can be determined from the values of a and b as follows:

$$s = \sqrt{a^2 + b^2} \tag{4.5}$$

$$\theta = tan^{-1}\left(\frac{b}{a}\right) \tag{4.6}$$

A 2D similarity transformation has four unknown parameters, i.e., rotation, scale, and translation along X and Y directions, so we need a non-singular system of four linear equations to reconstruct the transformation. A minimum of two matched pairs of the column centroids from both data points are needed to solve the transformation equation. The 2D transformation parameters are determined by our proposed alignment.

ii. Implementation of the proposed algorithm

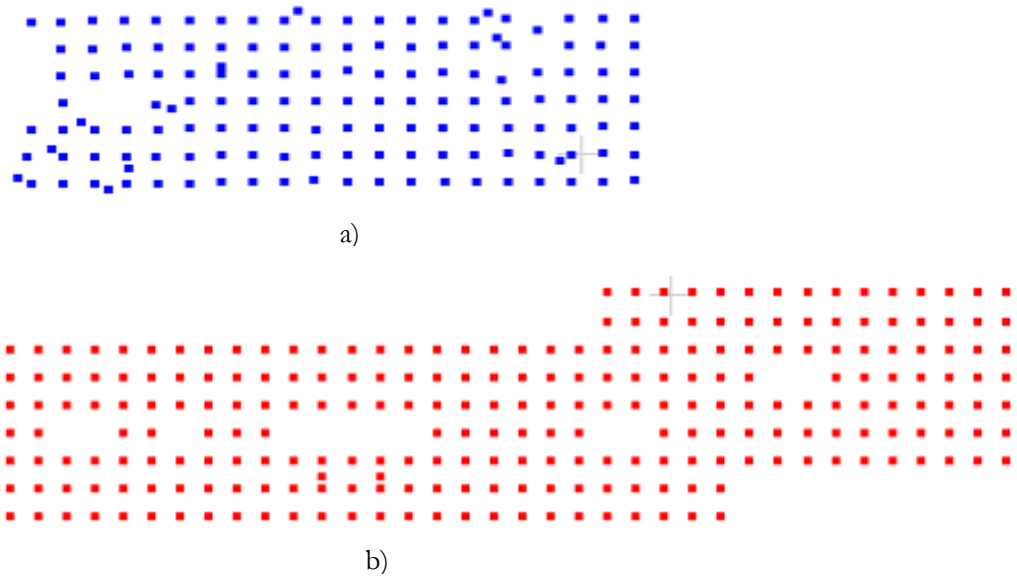
The motivation behind this approach is to apply a randomized alignment approach that is robust to the outliers in the detected columns of the as-built point cloud from the semantic segmentation. The proposed randomized alignment approach randomly picks two-column centroids from the candidate bases of source data point P and computes the 2D similarity transformation for all possible congruent pairs of column centroids from the target data points Q. The congruent bases from the target data points are determined based on the distance between the randomly selected source column centroid pair. This trial-and-error approach to get the optimal transformation continues until the best transformation is found. The best transformation is selected based on the largest number of inliers (i.e., source data points within a certain distance δ from the target data points) from the source data points. We modified the basic RANSAC algorithm to lower the computational time by randomly selecting a column pair with the centroidal distance between them. Then, the rest is to find all possible congruent sets from the target data point for the estimation of the best transformation parameters. We call our proposed registration method the 'congruent column sets' algorithm.

The main steps of the developed "congruent column sets" algorithm are:

1. Choosing the bases: Unlike basic RANSAC, we randomly pick pairs of columns centroid bases from the source data point (as-built point cloud data) and look for the matching possible pairs of columns

centroid bases from the target data point (as-planned BIM model data) based on distance congruence, i.e., the distance between target bases should be equal to the distance between the source bases within a certain tolerance δ. Also, instead of exhaustively sampling all possible bases from the source data point, which is computationally expensive, we only consider the top 3% of candidates with the maximum 2D distance between column centroids in the source data point. Note that we use the entire target point column centroids for the selection of all possible matching candidate sets based on the congruent distance. Below, we illustrate with one example how selecting the 3% candidate bases from the source data points is implemented. For this illustration, we used part of the point cloud column centroids (149 column centroids) to show the worst-case scenario where the captured point clouds don't cover the entire area of the building represented by the BIM model (see Fig 4.5).

Let p_i and q_i represent the computed 2D column centroids in source data points P and target data points Q, respectively. Let s_i represents the distance between each column centroids in p_i , whereas t_i represent the distance between each column centroids in q_i. The user loads P and Q as illustrated in Fig.4.5 to our developed method. Then, the centroids of both datasets are projected onto the XY plane and the maximum distance s_{max} is computed from s_i . Based on the computed maximum distance, the method selects and stores those column centroid pairs with $s_i \geq 0.97 * s_{max}$, which are the top 3% of pairs of column centroids with s_{max}, as illustrated in Fig.4.6a (only four pairs of 2D column centroids are selected out of 149); we call those selected pairs of column centroids 'bases'. The method randomly samples a base (a pair of column centroids) among the candidate bases (four pairs in this example) in the source data point and then computes the distance d_c between them (see Fig.4.6a.). Based on the d_c the method returns all possible target candidate bases with the congruent distance value in t_i within tolerance distance δ where $d_c - \delta \leq t_i \leq d_c + \delta$. Considering, for example, 6m adjacent distance between the column centroids, we assigned 50cm for δ.



a)

b)

Figure 4.5: Illustrates the inputs to the proposed coarse registration approach: a) the detected point cloud column centroids after projecting onto the XY plane, b) the target (reference) BIM model column centroids projected onto the XY plane.

Fig. 4.6b illustrates the selected target 2D column centroids (170 pairs of columns with a distance $d_c \pm \delta$) based on the sampled pair of a base from the 3% candidate bases in P with distance d_c (see Fig. 4.6a).

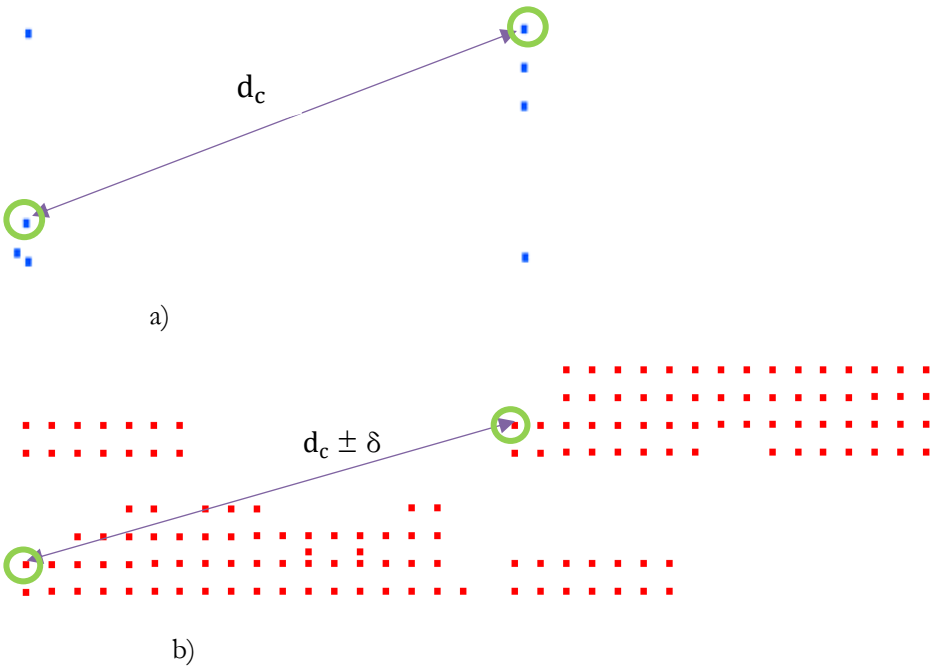


Figure 4.6: The portion of candidate bases selected by our proposed 'congruent column sets' method: a) the top 3% of source column centroid pairs with the maximum distance between them (selected from Fig. 4.5a), b) the possible candidate bases selected from an entire target data point in Fig.4.5b based on the randomly sampled base from a source data point with distance d_c as shown in Fig. 4.6a. The correct match in the target data point is shown with a congruent distance $d_c \pm \delta$

2. Computing the transformation: In this step, hypothetical 2D transformation parameters (rotation and translation) are estimated based on pairs of randomly sampled source column bases and all possible congruent column bases from the target data points. The transformation parameters are estimated by 2D similarity transformation using equation (4.4).

3. Scoring transformation support: In this stage, the entire set of columns from the source data points are transformed at each iteration based on the computed transformation parameters from step 2, and the alignment is scored based on the correctly placed columns (e.g., within a threshold distance δ). The value of δ varies for both TLS and UAV point clouds, as discussed in section 5.4. This step is more computationally expensive than the others. The main tasks carried out in this step are detailed as below:

- First, after each internal iteration, the full set of source data point column centroids are transformed using the estimated transformation parameters (rotation matrix and translation vector) from step 2.
- Second, compute the distance between the transformed source data point columns and the corresponding closest target data point columns. The distance between the corresponding closest columns is determined by comparing the distance between each transformed source data point with all the other target data points, and the list of distances with the minimum values is stored.
- Finally, the threshold distance is established to score the number of inliers. The threshold distance represents how far the transformed source data points should be from the target data points. All the source data point columns within 30cm from the corresponding target columns are selected as inliers.

The optimal transformation parameters are updated at the end of every iteration if a transformation has more inliers than the best transformation found so far. Repeat 1-3 until the best transformation parameters are found with more than the preset maximum number of inliers or less than the preset

threshold of RMSE. The method returns the optimal transformation parameters (2D rotation and translation) corresponding to the maximum number of inliers.

iii. **Run time of our proposed 2D registration** :

The 'congruent column sets' carries out two different iterations for optimal 2D transformation parameters estimation. The main iteration is implemented for randomly selecting a pair of base from 3% of the source data point candidates, whereas the internal iteration takes place for all possible target data point candidates (congruent column sets) based on the distance congruence. So, the total run time is a combination of the computational time required for the main and internal iteration until the system returns the optimal transformation parameters or termination criterion. Our proposed method often obtains the optimal result at the first main iteration, as seen in our experimentation result in section 5.4. This is the case because the source data points (except the outliers) are a subset of the target data points; there is a high possibility that the correct matching base is obtained among the first selected target congruent column sets. Below we elaborate using the above example illustrated in Fig 4.5.

Once a pair of base is randomly sampled from the 3% candidate bases (4 pairs as illustrated in Fig. 4.6a), the 'congruent column sets' executes an internal iteration for all possible congruent column bases from the target data points (e.g., 170 pairs as seen in Fig.4.6b) to estimates 2D transformation parameters (rotation and translation). The iteration terminates whenever the criteria are met. The criteria are the maximum number of inliers that supports the estimated transformation and the RMSE. Unless otherwise, the method randomly samples another base from the stored 3% source data point candidates (among four pairs) and continues selecting another set of target data point bases based on the congruent distance. So, considering only 3% of the source data point for minimum set sampling and the correspondence search in the target data point based on the congruent distance helps our method converge quickly.

iv. 3D Transformation after the 'congruent column sets'

The 3D similarity transformation has a total of six parameters (assuming unity scale). The 'congruent column sets' gives us the three parameters: rotation along the Z direction and translation along the X and Y direction through 2D similarity transformation using 'congruent column sets'. Only three are left: the translation along the Z direction and two rotations along the X and Y direction. As discussed above, the vertical Z direction for the BIM model and the as-built point cloud is orthogonal to the X-Y plane. Hence, the rotation angles along the X and Y direction are assumed to be zero. Then, the translation along the Z direction will be the only unknown to be determined. The 3D rotation matrix R is composed using Euler angle $\omega, \varphi, \mathrm{k}$, and the translation vector is estimated using the general transformation relation as:

$$\mathrm{T} = \overline{\mathrm{X}}_{\mathrm{t}} - \mathrm{R}\overline{\mathrm{X}}_{\mathrm{s}} \qquad (4.7)$$

Where $\overline{\mathrm{X}}_{\mathrm{t}}$ and $\overline{\mathrm{X}}_{\mathrm{s}}$ are the center of mass of the detected columns of the target and source data points , respectively. The columns used for the estimation of the vertical (Z) axis translation are the inliers (the column centroids) obtained using our 2D alignment approach.

Where R is a $(3\mathrm{x}3)$ matrix that denotes the rotation, and T is a translation vector. The translation along the Z direction(T_{z}) is the only unknown in T since the translation along X-axis (T_x), and Y-axis (T_y) have already been solved in the 2D transformation using the 'congruent column sets' algorithm. Finally, the as-built point cloud is transformed based on the estimated 3D transformation parameters. The acquired result can be considered as an initial alignment (coarse registration). Then, we apply the principal component (PCA) (C. Kim et al., 2011) as a refinement registration on top of our developed coarse registration to get a more accurate result.

v. 3D transformation using Principal Component analysis

We applied the PCA on top of our developed approach as a fine registration equivalent. The inliers (the column centroids) of both target and source data points obtained using our 2D 'congruent column sets' are used as input for the estimation of 3D transformation parameters using PCA. This gives us more accurate 3D transformation parameters than the one achieved using our 'congruent column sets' approach. The PCA is based on the principal axis of both data points to give the 3D rigid transformation. This method computes the rotation matrix by analyzing the eigenvectors of the covariance matrices of the point cloud and BIM model, and the translation vector is calculated using the center of mass of both datasets after transforming the point cloud in the reference frame of the target data point. For the PCA to give an optimal rotation matrix and translation vector:

- The principal axes (X, Y, Z) of both datasets should be distinct from one another, and outliers and clutters from the built environment in the point cloud must be distributed uniformly along with these directions.
- The center of mass of both datasets must correspond to each other.

In our case, we assume that the point cloud and BIM model column centroid inliers are obtained using our developed 'congruent column sets' approach. So, most parts of the source and target data points are overlapping. Considering a large percentage overlap and the distinct principal axis for both datasets, PCA gives the more accurate result, as shown in the experiments in section 5. A detailed explanation of the estimation of 3D transformation parameters using the PCA is given in the following section.

Given the inliers of source and target column centroids from our 'congruent column sets' approach, the principal axes are defined by the eigenvectors of covariance matrices of three-dimensional coordinates of the data points (Dorai et al., 1994).
Given N data points $\mathrm{X_i} = (\mathrm{X_i}, \mathrm{Y_i}, \mathrm{Z_i})^T$, $\mathrm{i} = 1, \ldots, \mathrm{N}$, the covariance matrix K is defined as:

$$K = \frac{1}{N}\sum_{i=1}^{N}\{(X_i - \bar{X})(X_i - \bar{X})^T\} \tag{4.8}$$

Where the center of mass of points can be calculated as:

$$\bar{X} = \left(\mu_x \mu_y \mu_z\right)^T = \frac{1}{N}\sum\nolimits_{1}^{N} X_i. \tag{4.9}$$

Let $\mathrm{V_1}, \mathrm{V_2}, \mathrm{V_3}$ be eigenvectors and $\lambda_1, \lambda_2, \lambda_3$ Eigenvalues of K, respectively. Then, K can be factorized as:

$$K = V\Delta V^T \tag{4.10}$$

Where $V = \{V_1, V_2, V_3\}\ and\ \Delta = diag\{\lambda_1, \lambda_2, \lambda_3\}$.
The rotation R is determined from the product of eigenvector matrices:

$$R = V_t V_s^{-1} \tag{4.11}$$

Wher $\mathrm{V_t}$ and $\mathrm{V_s}$ are Eigenvectors of target and source data points, respectively.
The translation T is determined by the distance between the center of mass of the target and source data points using equation (4.7). The acquired result can be considered an optimal registration result for construction progress monitoring. We compare the PCA result with the standard ICP (Besl and McKay, 1992) in the experiments section 5.

4.6. Performance of proposed registration approach

The proposed coarse-registration approach was evaluated based on two criteria:-`

1. **Registration speed**: Computational time required to perform the registration.
    - **Number of iterations**: Number of iterations needed until the optimal transformation parameters selecting process reaches the termination criterion
    - **Run time**: the time required until the optimal transformation parameters are obtained
2. **Registration quality:** The quality of our proposed registration method is assessed based on:

- **The number of inliers**: the number of correctly placed point cloud columns that support the optimal transformation (within a threshold distance).
- **RMSE:** the root mean square error of the distances between the column centeroid inliers of the point cloud and the BIM model is computed as seen in equation (4.12), where N is the number of inliers and d_i is the distance between inliers after transformation. We compute the 2D and 3D RMSE.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i)^2} \tag{4.12}$$

- **The invariance to rotational errors**: the point cloud is deliberately rotated with respect to the ground truth, and the rotation error is calculated after transformation using our proposed approach.
- **Cloud to cloud distance**: After registering the point cloud columns to the reference BIM model columns using PCA and ICP, we compute the distance between the corresponding points using the Cloud-to-Cloud distance computation tool. Note that we use the corresponding inliers (detected column point cloud) for the cloud to cloud distance computation, but not the column centroids. This tool computes the distances between two point clouds using the 'nearest neighbor distance': for each point in the compared cloud, the algorithm searches the reference cloud for the nearest point and computes the (euclidean) distance between them (see Fig.4.7). In our case, the reference cloud is the BIM model point cloud (column inliers), and the compared cloud is the point cloud column inliers.

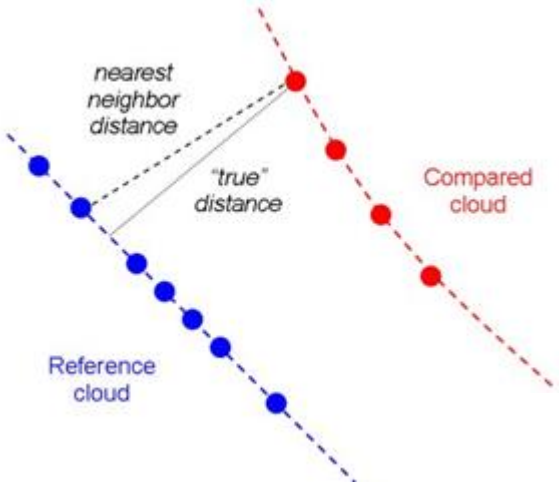


Figure 4.7: The cloud to cloud distance computation mechanism in CloudCompare (Girardeau-Montaut, 2015)

5. EXPERIMENTS AND DISCUSSION

5.1. Point cloud column detection using deep learning

5.1.1. Model training and parameter tuning

The selected pre-trained KPConv model was trained on the large S3DIS standard benchmark dataset. As summarized in Table 4.1, in the original paper, it was mentioned that it had achieved an average accuracy (IoU) of 60% with Torch and 65.5% with TensorFlow library on the S3DIS dataset. The dataset contains thirteen (13) indoor building components and furniture classes. The classes used were ceiling, floor, wall, beam, column, window, door, table, chair, sofa, bookcase, board, and clutter. Our custom dataset is labeled to five (5) classes of our interest (see Table 4.3); all of our interest classes are available in the S3DIS dataset classes(column, beam, ceiling/floor, and wall). In Table 4.2, the KPConv model developers also presented the model prediction accuracy for these common classes in terms of IoU score as 16.5%, 81.4%, 97.3%, and 0% for column, wall, floor, and beam respectively. This shows a very low performance of their trained model for predicting columns and beams relative to the other common classes (ceiling and wall).

The pre-trained model was loaded, and the inference was run on our custom point cloud data from the construction site to see if the pre-trained model could generalize good enough on our custom dataset. The prediction results are visualized in Fig.5.1. As seen in the Figure, from the visual inspection, all the column elements are misclassified as a wall or window, and beams and walls are misclassified as clutter class. However, the floor class is relatively well-segmented.

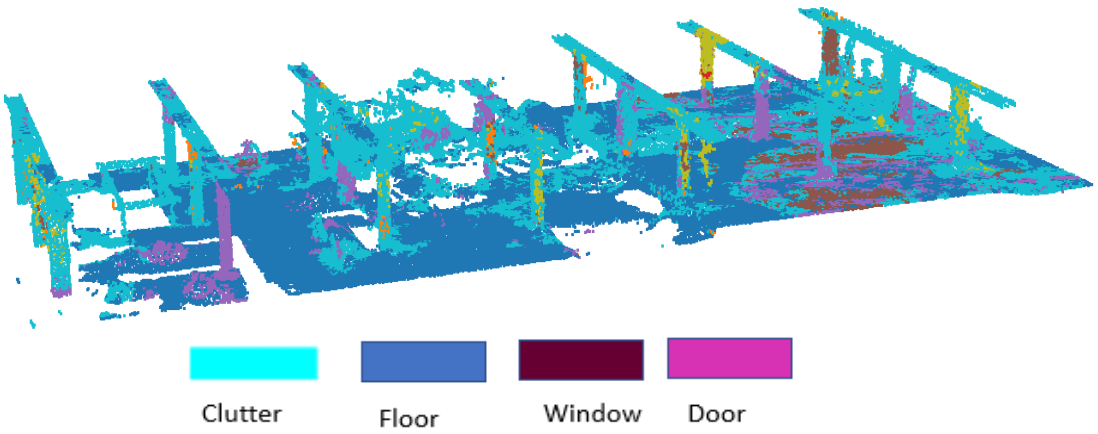


Figure 5.1: The inference result of the pre-trained KPConv semantic segmentation model on the ITC UAV point cloud

Generally, this shows the generalization capability of the pre-trained KPConv model on our custom dataset is significantly poor. As a result, the decision was made to apply a fine-tuning process of transfer learning (as discussed in section 4.2.1), where the model was trained on our prepared training dataset by unfreezing the entire pre-trained model to improve the model segmentation accuracy.

The KPConv model was loaded, and the last output layer was customized from thirteen (13) to five (5) to match the number of our output classes, then trained the entire model from scratch on our prepared training dataset mentioned in Table 4.3. During the training phase, the model hyperparameters, such as a learning rate, the number of epochs, the optimizer, and the regularization parameters, were adjusted to achieve the best possible results. Finally, the KPConv model was trained with parameters summarized in Table 5.1.

Table 5.1:Model training parameter settings

KPConv training parameter tuning	
Optimizer	Adam
Initial learning rate	0.01
Exponential learning rate decay	0.985
Maximum epoch	400
Momentum	0.98
Batch size	4
Batch limit (points)	20000
Kernel points	15
Number of KPConv layers	5

5.1.2. Quantitative analysis

Since there is a significant class imbalance in our prepared test dataset, the IoU metrics were used as appropriate metrics to evaluate the prediction accuracy of the semantic segmentation model (as discussed in section 2.3.2c). The trained model was evaluated on the test dataset and has achieved 73% accuracy in overall classes using the mean intersection over union(mIoU) and column segmentation accuracy of 69%. The test result of the trained model is summarized in Table 5.2 in terms of the confusion matrix of intersection over union (IoU).

Table 5.2: Confusion Matrix of IoU metrics for semantic segmentation results from the KPConv model

	Clutter	Column	Beam	Slab	Wall
Clutter	0.31	0.10	0.01	0.08	0.48
Column	0.00	0.69	0.05	0.00	0.00
Beam	0.00	0.04	0.89	0.01	0.00
Slab	0.00	0.00	0.00	0.98	0.00
Wall	0.00	0.18	0.01	0.03	0.76

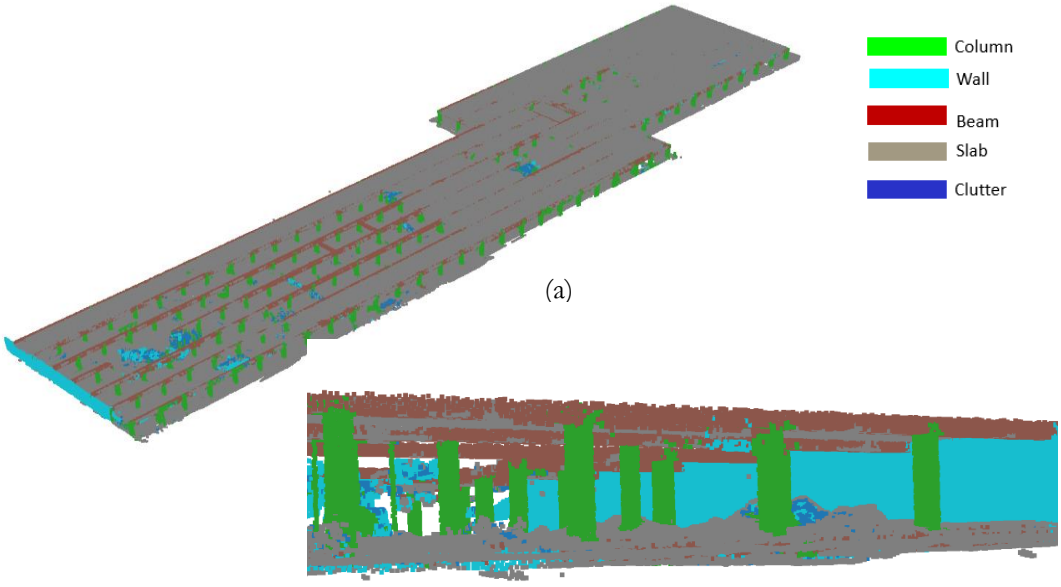
As shown in Table 5.2, the trained model segmentation accuracy of column, beam, wall, slab, and clutter classes was 69%, 89%, 76%, 98%, and 31%, respectively. This shows a significant improvement in the segmentation accuracy of the KPConv model, particularly the column detection accuracy, which was 16.5%, as reported in the original paper (Thomas et al., 2019a) presented in Table 4.2. This improvement was made possible by training the model from scratch by preparing new training data as described in section 4.2. In addition to that, we have extracted the normal vector component for each point in the training data and integrated it as an additional attribute to the geometric data. However, the confusion matrix shows that there is still some confusion between classes in the classification result. From Table 5.2, it can be observed that the points belonging to the wall class have significant confusion with the column class. About 18% of wall class points were misclassified as column classes. The confusion between column and wall class could mainly be caused due to the class similarity as both classes are vertical elements and possess similar normal vector components. The slab category achieved the highest accuracy, followed by the beam and wall class. The clutter class has achieved the lowest accuracy when compared to the other classes. The main reason behind the very low prediction accuracy on the clutter class might be due to the limited quantity of training data for the clutter class (as seen in Table 4.3).

5.1.3. Qualitative analysis

The inference was made on our construction site point cloud (both the TLS and UAV), and the semantic segmentation results are visualized in Fig.5.2, Fig.5.3, and Fig.5.4. From the visualization of the segmentation result for both the UAV and TLS point clouds, it can be observed that the trained model has labeled the points in the point cloud correctly according to their corresponding category. However, in the regions where columns are embedded in the wall, there is significant confusion in the segmentation result between the two classes (see Fig.5.4b). Intuitively, even for a human expert, it would extremely be

difficult to distinguish between the two classes if the columns were embedded in the walls, which is often the case in the AEC/FM context. Also, as illustrated in Fig.5.3b, the points which belong to the clutter class (safety fences) were misclassified as a wall class.

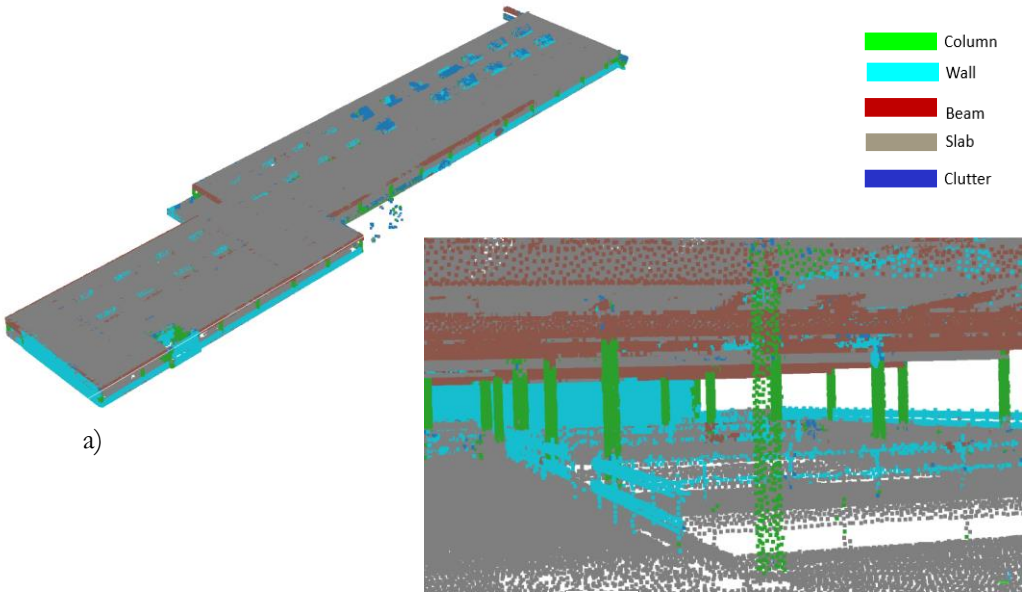
.



(a)

(b)

Figure 5.2: Results obtained from semantic segmentation of the TLS point cloud ground floor: a) Outside view, b) Inside view



a)

b)

Figure 5.3: Results obtained from semantic segmentation of the TLS point cloud-first floor: a) Outside view, b) Inside view

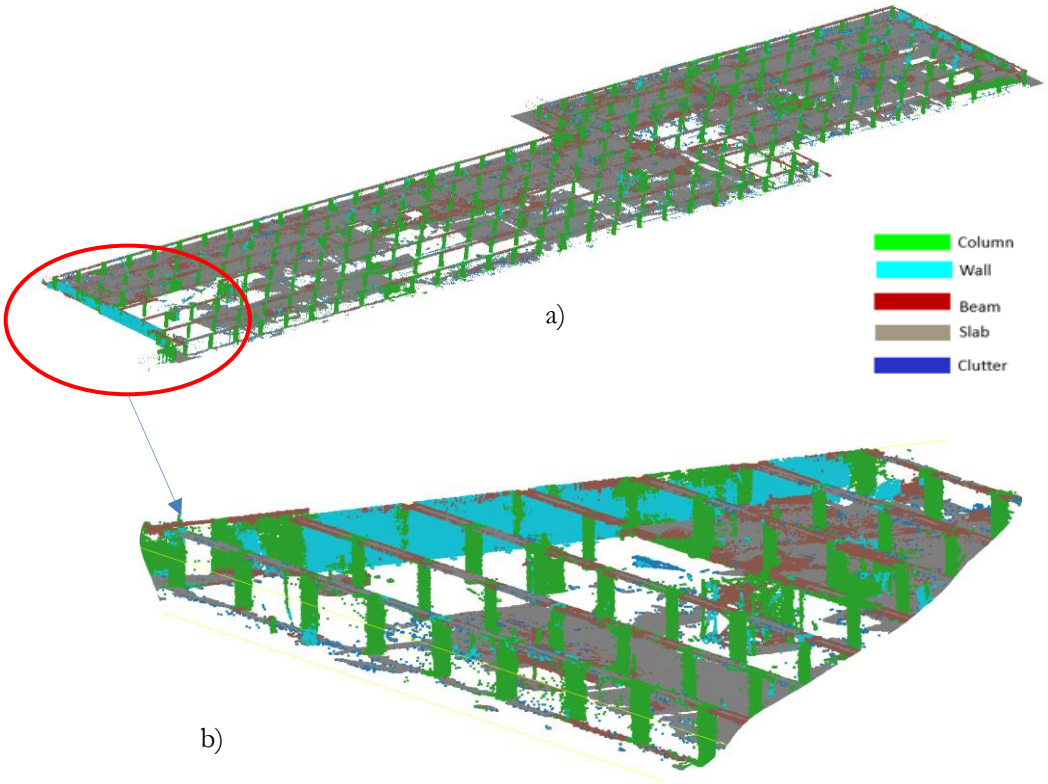


Figure 5.4: Results obtained from semantic segmentation of the UAV point cloud-first floor: a) ground floor, b) the detailed view of the outlined region

The resulting extracted columns for both TLS and UAV point clouds are shown in Fig.5.5. From the qualitative analysis of both datasets, the quality of the detected TLS columns is better than the UAV ones. This was expected as the UAV point clouds are noisier compared to the TLS due to the low-quality consumer-grade sensor used for the UAV data acquisition.

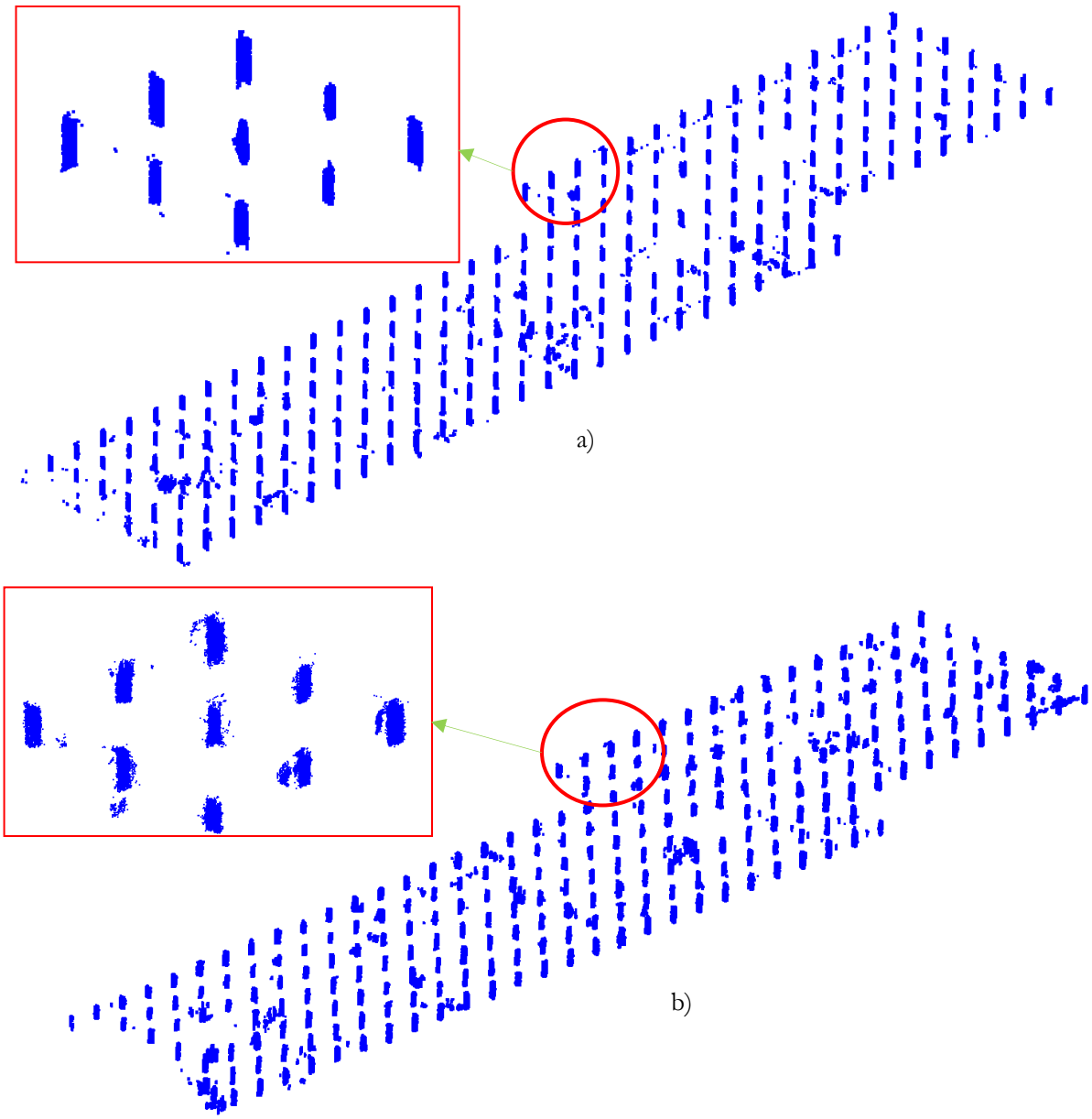


Figure 5.5: The ground floor points detected as columns by KPConv semantic segmentation model: a) TLS point cloud columns, b) UAV point cloud columns

5.2. BIM model column detection

As described in methodology section 4.3, the BIM model in IFC format was imported into the Revit software, and columns were retrieved from the available semantic information in the BIM. The columns were filtered from the structural section of the BIM model and exported as a mesh in STL format. Then, we sampled uniform resolution point clouds from the BIM column mesh. Fig.5.6 illustrates the ground floor columns of the BIM model in a point cloud format.

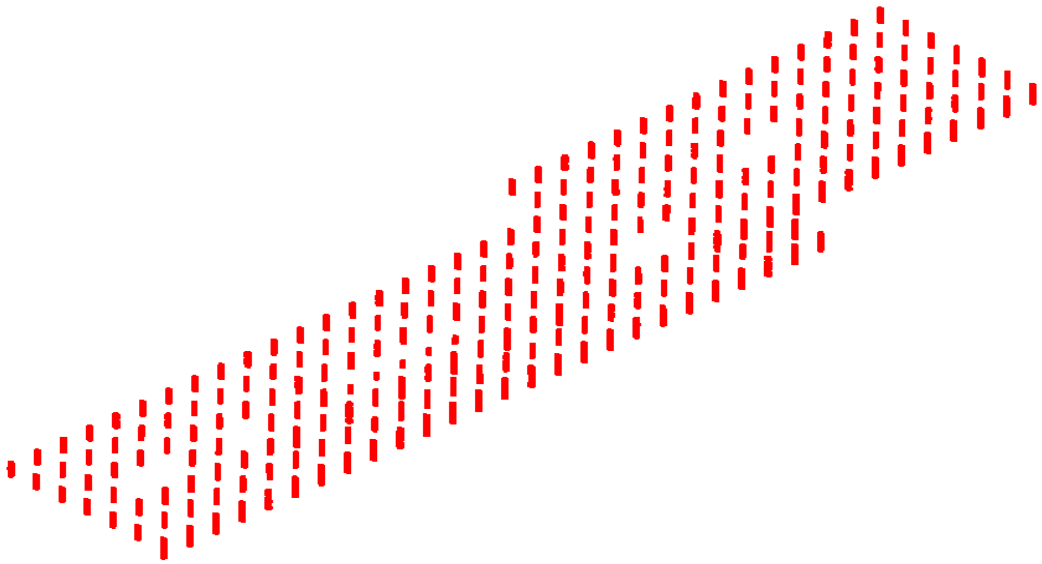


Figure 5.6: The ground floor BIM model columns in a point cloud format

5.3. Preprocessing the detected columns

We used the detected ground floor columns from both datasets to test our developed registration approach. The ground floor columns have a rectangular cross-section with a height that extends from the top of the ground floor to the ceiling of the first floor. The following preprocessing techniques were applied to both datasets before loading the detected columns into our developed registration method:

i. Filtering BIM model columns

The ground floor BIM model columns exported from the Revit software has varying height, as illustrated in Fig.5.7. It was observed from the BIM model that the height of the ground floor columns extends from the top of the ground floor to the top of the first floor, which is not the case with the point cloud columns. This has affected the accuracy of the vertical translation parameter in the transformation, which will be discussed in section 5.4.1.

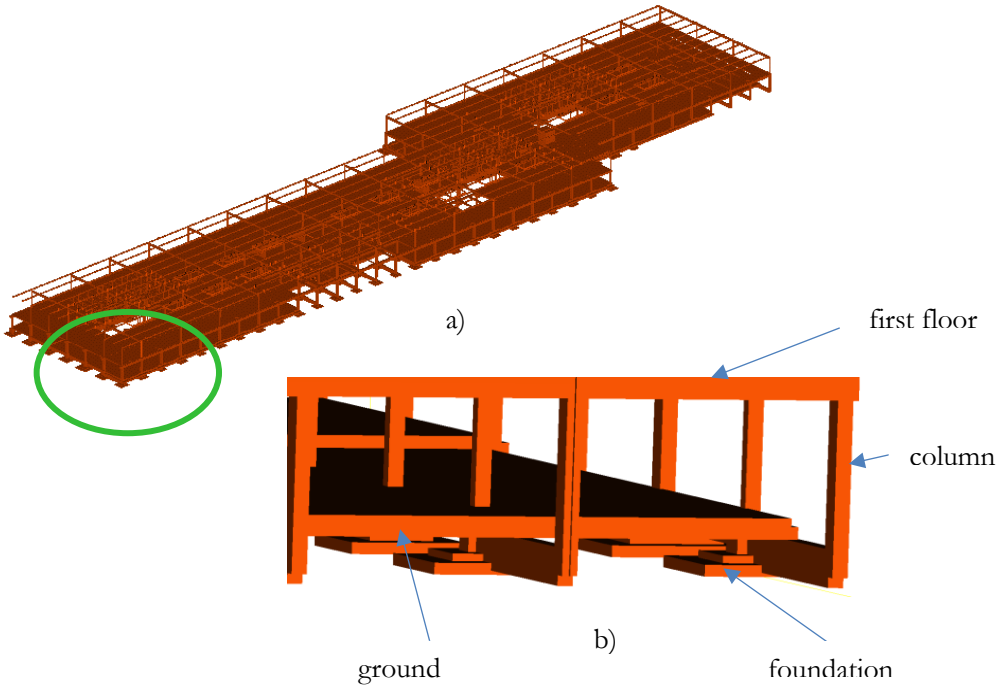


Figure 5.7: BIM model ground column height variation: a) the BIM model structural component, b) detailed view of the ground floor column height variation

As seen in Fig.5.7, some of the ground floor columns extend below the ground floor to the foundation part. The column points above the ground floor have a positive vertical Z-value, whereas those below have a negative value. We have removed those ground floor column points with negative Z-value because our counterpart point cloud columns contain points above the ground level.

ii. Cleaning the detected point cloud columns

The TLS and UAV point cloud columns segmented from the deep learning semantic segmentation model, as illustrated in Fig.5.5, contains some outliers (misclassified points). Those points possibly contribute to the deviation in the computed geometric centroid of the column clusters in the later stage, which on the other hand, affects the registration accuracy of the developed method. As a result, we have applied the outlier removal technique to filter out points far away from the cluster of points belonging to the column category. For this purpose, the detected point cloud columns were imported into the CloudCompare software, and the 'Statistical Outlier Removal' tool was used. As explained in the methodology section 4.4, after some experiments, we set the 'number of points ' to 20 points and 'standard deviation multiplier' $\boldsymbol{n}$ to 1.5 to estimate the threshold distance for the outlier removal.

iii. The detected columns clustering result

The extracted columns from both datasets were subjected to the clustering algorithm to get the column instances of each dataset. We used the "Label Connected comp." segmentation tool which is available in the CloudCompare software, as explained in the methodology section 4.4. After some experiments, we tuned the 'minimum points per component' to 20 points and the 'octree level' to 8 (grid step = 0.886117). Then, we removed the segmented clusters with less than 20 points. Finally, we got 252, 256, and 274 column clusters for the BIM, TLS, and UAV datasets, respectively.

iv. Column centroid computation result

Once the column clusters were obtained, we computed the 3D geometric centroid of each cluster of the detected columns to use it as an input for our developed registration algorithm. We used the ground floor columns for both datasets. The computed centroid points for each dataset are shown in Fig.5.8.

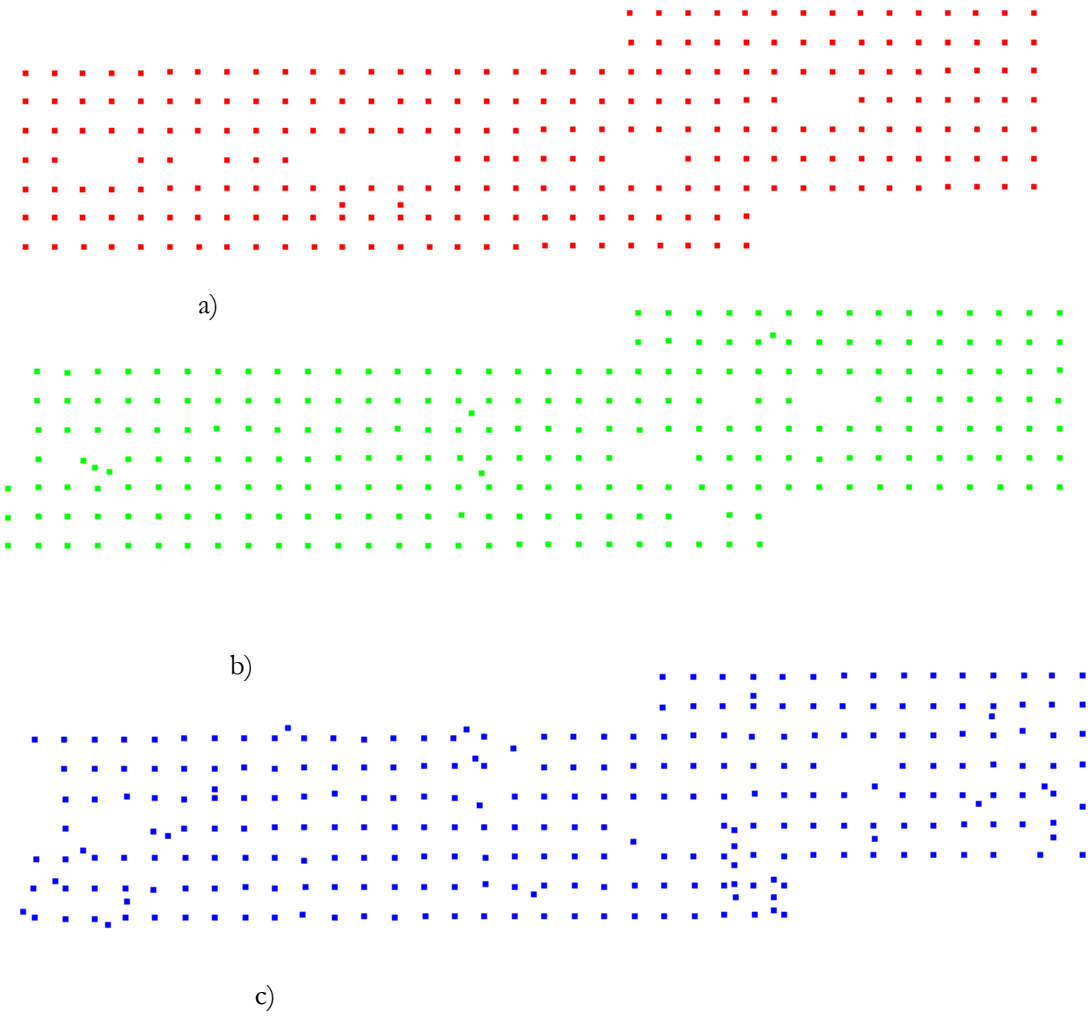


Figure 5.8: The centroids of the detected columns: a) BIM model, b) TLS Point cloud, c) UAV point cloud.

5.4. Experimental setup for coarse registration

The experimental setup has been done to test the robustness of our proposed registration approach in section 4.5. We tested the proposed method using both the TLS and UAV dataset for the following different scenarios:

- Registration before/after rotating the source data point to test for rotation invariance.
- Registration using a section of the source data points to test for symmetry and self-similarities in the dataset.

For both scenarios, we configure the computed column centroids of both TLS and UAV dataset, as illustrated in Fig. 5.9 and Fig.5.10. Note that in all the experimentation scenarios, the target data point pose remains unmodified.

a. Dataset configuration for the rotation invariance test

In this case, we have tested the robustness of our proposed alignment approach by imposing various rotational angles (along Z-axis) on the source data point (bot the TLS and UAV) with respect to the initial orientation, as illustrated in Fig.5.9. The initial orientation of the BIM and point cloud dataset is

illustrated in Fig.5.9b and a, respectively. Fig.5.9bc represents the rotated position with respect to the initial position after deliberately applying rotation angles: $30^0, 45^0, 60^0$, and 75^0.

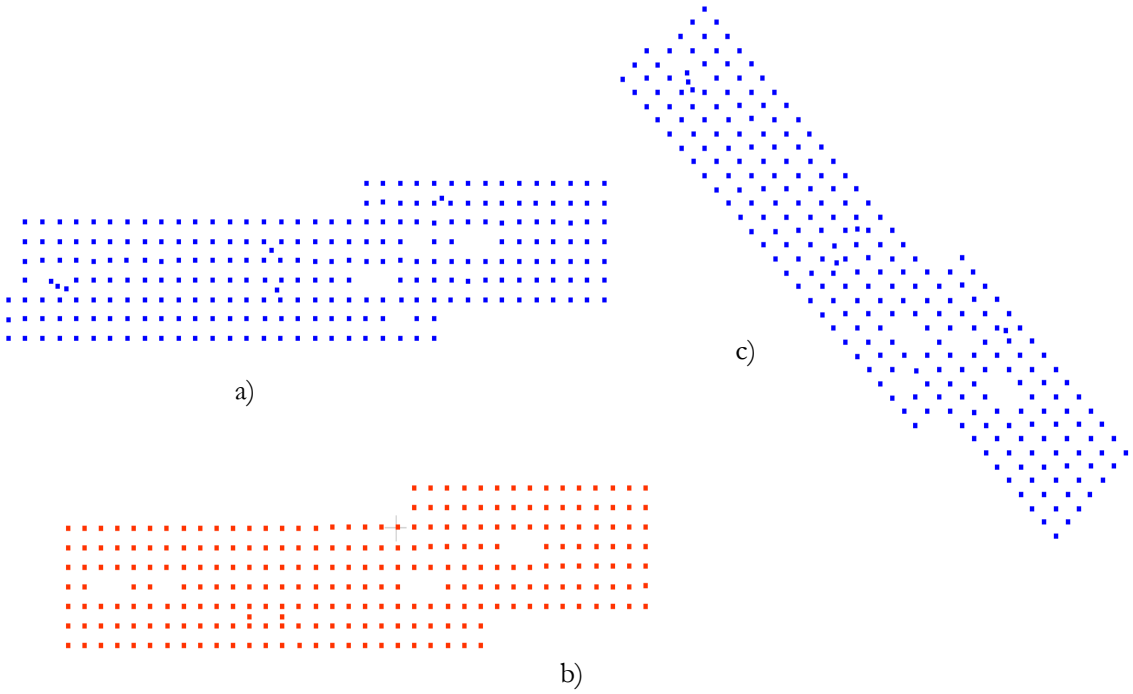


Figure 5.9: Experimental dataset configuration for rotation invariance test: a) actual orientation of the source point dataset, b) actual orientation of the BIM data set, c) oriented source datapoint after applying different rotation angle

b. Dataset configuration for the symmetry test

Usually, the as-built point cloud from the construction site might contain only a part of the building, and only a limited number of columns could be obtained. Due to significant symmetry and self-similarity (the column pattern is uniform) in our test dataset, it was expected that several transformations would have very high inliers, although only one of them is the correct one. This often poses a challenge in the process of estimating the correct 3D transformation parameters. To test for this problem using our proposed method, we selected some parts of the point cloud columns from the source data point to align with the target data points. Our reference/target data points are used entirely (see Fig. 5.8a) in all the test cases. The part of point cloud columns used for both the TLS and UAV datasets for the symmetry test is shown in Fig.5.10, and the results are presented in the respective section for each dataset.

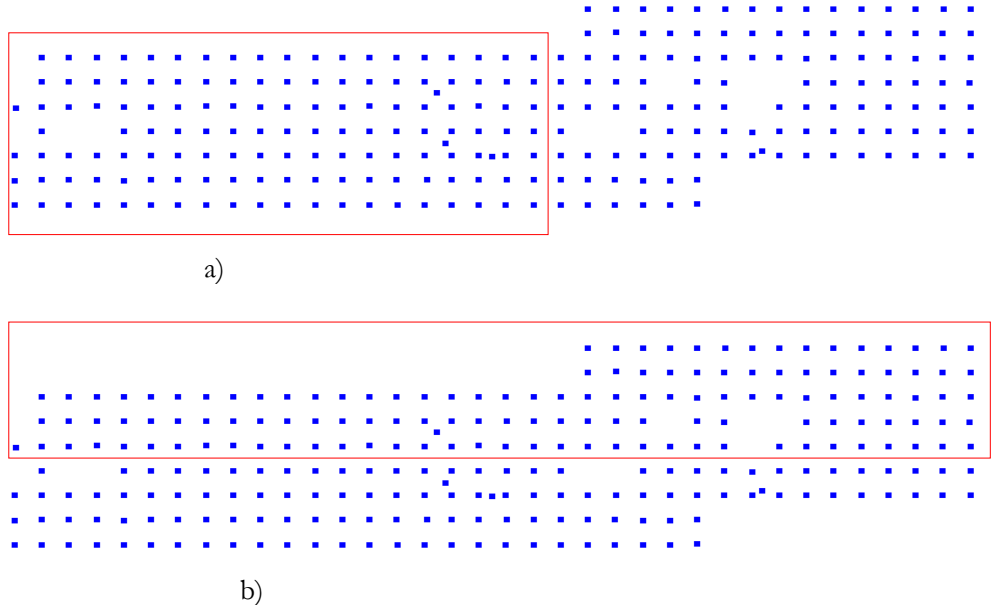


Figure 5.10: Experimental dataset configuration for symmetry test using a different section of the source data points as outlined by the rectangular red line: a) left section, b) top section

In our experimentation with both datasets, we summarize our 'congruent column sets' registration results in terms of quality and computational speed metrics explained in the methodology section 4.6. The RMSE was calculated both in 2D and 3D coordinate frames to show the effect of vertical translation error on the RMSE. For the registration computations, we used "Intel(R) Core (TM) i5-4200M CPU @ 2.50GHz" with RAM storage of 16GB laptop.

As explained in section 4.5, we have applied automated 3D registration using the PCA on top of the inliers (3D column centroids) obtained using our developed 'congruent column sets' algorithm for a more accurate result. Then, we computed the RMSE both in 2D and 3D coordinate frames after PCA and summarized the result for both datasets. Also, we applied the classical ICP fine registration on top of the coarse alignment obtained using our 'congruent column sets' algorithm for only one case (the actual source data point before applying rotation), and the results are summarized for both datasets. Note that, as mentioned in the methodology section 4.5, for comparison, we apply both PCA and ICP on top of the corresponding 3D column centroids of both datasets, which are identified as inliers by our 'congruent column sets' algorithm. We used the classical point-to-point ICP with the 100% overlap ratio of both datasets since we have equal coverage of inlier centroids from both datasets.
In the following section, we load the computed column centroids into our proposed method and discuss the experimentation results of both the TLS and UAV point cloud separately.

5.4.1. Registration of the TLS point cloud to the BIM model

As described in methodology section 4.5, the computed TLS and BIM column centroids, as illustrated in Fig.5.8, were loaded into our developed 'congruent column sets' algorithm, and the experimental results are discussed in the following section for both scenarios. The total number of the detected TLS point cloud column centroids is 256, out of which about 13 (5%) centroids are outliers (non-columns). Note that we estimated the outlier ratio by visual inspection since the inlier column centroids are often aligned in a linear fashion in X and Y directions. Our proposed method considers only 3% of the pair of source data points with the maximum distance between them, as discussed in section 4.5. For both test scenarios of the TLS dataset, we set the following parameter for our proposed algorithm:

- Maximum main iteration= 100

- Inlier ratio= 90%
- threshold distance= 30cm

The proposed registration method has been run three times for each case, and the average results are summarized in the following sections.

i. Test for rotation invariance

As illustrated in Fig.5.9, we have tested the robustness of our proposed alignment approach by imposing various vertical rotational angles (along Z-axis) on the TLS dataset. Then, we summarized the registration result in Table 5.3 for all the initial orientations with/ without applying rotation.

Table 5.3: Summary of the transformation results for the various orientation of the TLS source data points

Initial Angle(degree)		0^0	30^0	45^0	60^0	75^0
Rotation Error(degree)		0	0.02	0.02	0.02	0.03
Inliers		238	238	238	239	239
Total iteration		57	63	30	54	49
Run time(sec)		52	67	42	48	52
'congruent column sets' RMSE(m)	2D	0.06	0.05	0.06	0.05	0.06
	3D	0.18	0.17	0.18	0.17	0.18
PCA RMSE(m)	2D	0.02	0.03	0.03	0.03	0.03
	3D	0.12	0.12	0.12	0.12	0.12
ICP RMSE (m)	3D	0.13	-	-	-	-

As observed from Table 5.3, our developed registration approach has correctly aligned source data points to the reference target data points in all initially imposed rotation angles with an average rotation error of 0.02 degrees. This shows the proposed method is invariant to rotation. The method has achieved the average RMSE of 5cm in 2D (XY plane) and about 17cm in 3D via our 'congruent column sets'The performance was also evaluated in terms of computational speed, and the algorithm requires, on average, 50 iterations within 55 sec to achieve the optimal result. This high speed in computation is achieved by considering only the top 3% of pairs of source data points with the maximum distance between them for 'congruent column sets' random sampling. This would take many hundred iterations if the basic RANSAC algorithm were applied by considering entire source data points. Fig.5.11 illustrates the overlaying of the point cloud columns to the corresponding BIM model columns after transformation using 'congruent column sets'.

After our 'congruent column sets' as fine registration equivalent, we applied PCA for a more accurate result and achieved an average RMSE result of 3cm in 2D (XY plane) and 12cm in 3D. Then, we compared the PCA result with the standard ICP fine registration. Using ICP fine registration on top of the coarse alignment obtained using our 'congruent column sets', we have achieved the 3D RMSE of 13cm for the case without applying rotation on the source data point. This shows with PCA, we have achieved better accuracy than the refinement registration using standard ICP. The low accuracy of ICP with respect to PCA might be caused due to the minimization problem of the point-to-point metric. Fig.5.13 illustrates the overlaying of the point cloud columns to the corresponding BIM model columns after transformation using both PCA and ICP.

The RMSE achieved in 2D is significantly lower than that achieved in 3D. This might be caused by the difference in the vertical component of the computed 3D centroid of detected columns for both datasets. This difference was expected because the vertical height of the detected point cloud columns is less than the corresponding BIM model columns (see Fig.5.11c). It is observed from the BIM model that the average height of the BIM model columns retrieved from the Revit software is 3.4m, whereas that of the detected point cloud columns was measured as 2.8m. This is due to the BIM model columns being designed to extend from the top of the ground floor to the top of the first floor. The height of the

ground floor column of the BIM model includes the clear height from the top of the ground floor to the bottom of the beam (2.8m), the depth of the beam (about 40cm) just above it, and the thickness of the floor (about 20cm). In addition to this, the detected point cloud columns have varying height distribution which might be caused by occlusion, clutters, noise in the point cloud, and the semantic segmentation model detection accuracy. Consequently, this has significantly increased the vertical translation error (along Z-axis) between both datasets, which in turn contributes to the 3D RMSE, as seen in Table 3.2. In Fig. 5.12, we have illustrated the distribution of the distance error between the inliers of the point cloud and the BIM model column centroids after registration using 'congruent column sets' and PCA in both 2D and 3D to show the effect of column height difference in both datasets on RMSE computation.

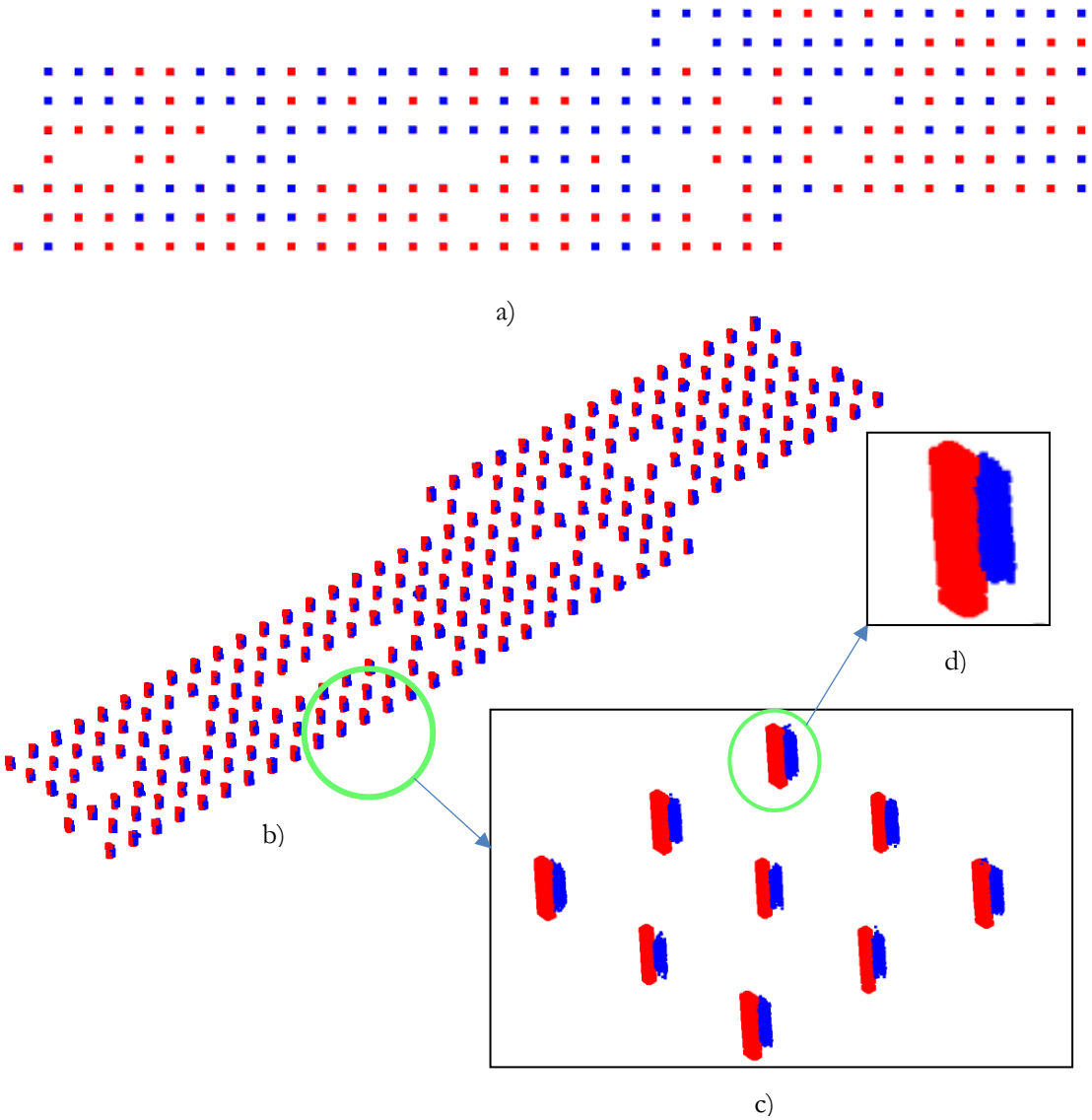
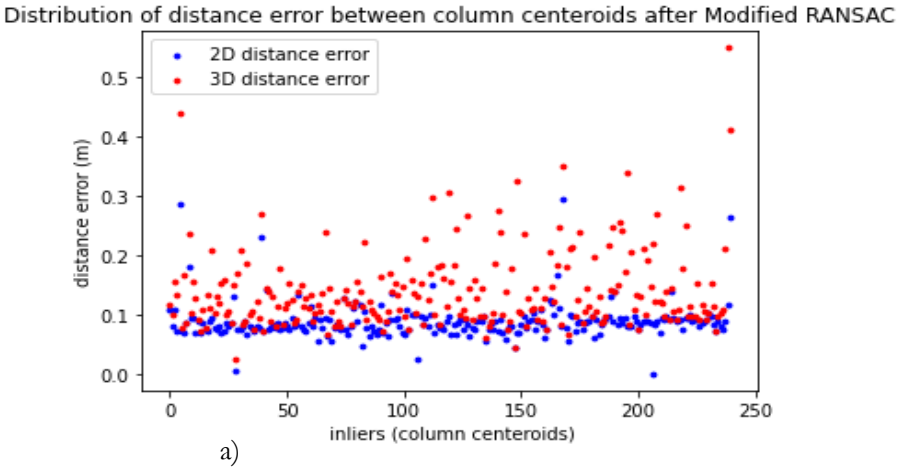
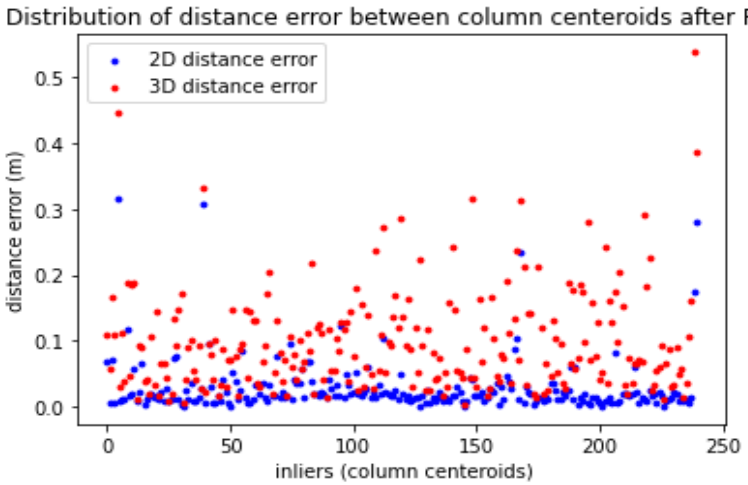


Figure 5.11: Overlay of the TLS cloud columns(blue color) with the BIM model columns(red color) after coarse alignment using our 'congruent column sets' algorithm: a) based on 3D centroids of the detected columns, b) based on entire column inliers of the corresponding dataset, c) & d) illustrates the vertical height difference between the detected point cloud columns and BIM model columns.



a)



b)

Figure 5.12: The 2D and 3D distance error between the inliers of TLS point cloud and BIM model column centroids after transformation; a) using the 'congruent column sets' algorithm, b) after the PCA

We have applied the refinement registration using the standard ICP algorithm on top of the 'congruent column sets'-based coarse registration, and the result is illustrated in Fig.5.13.

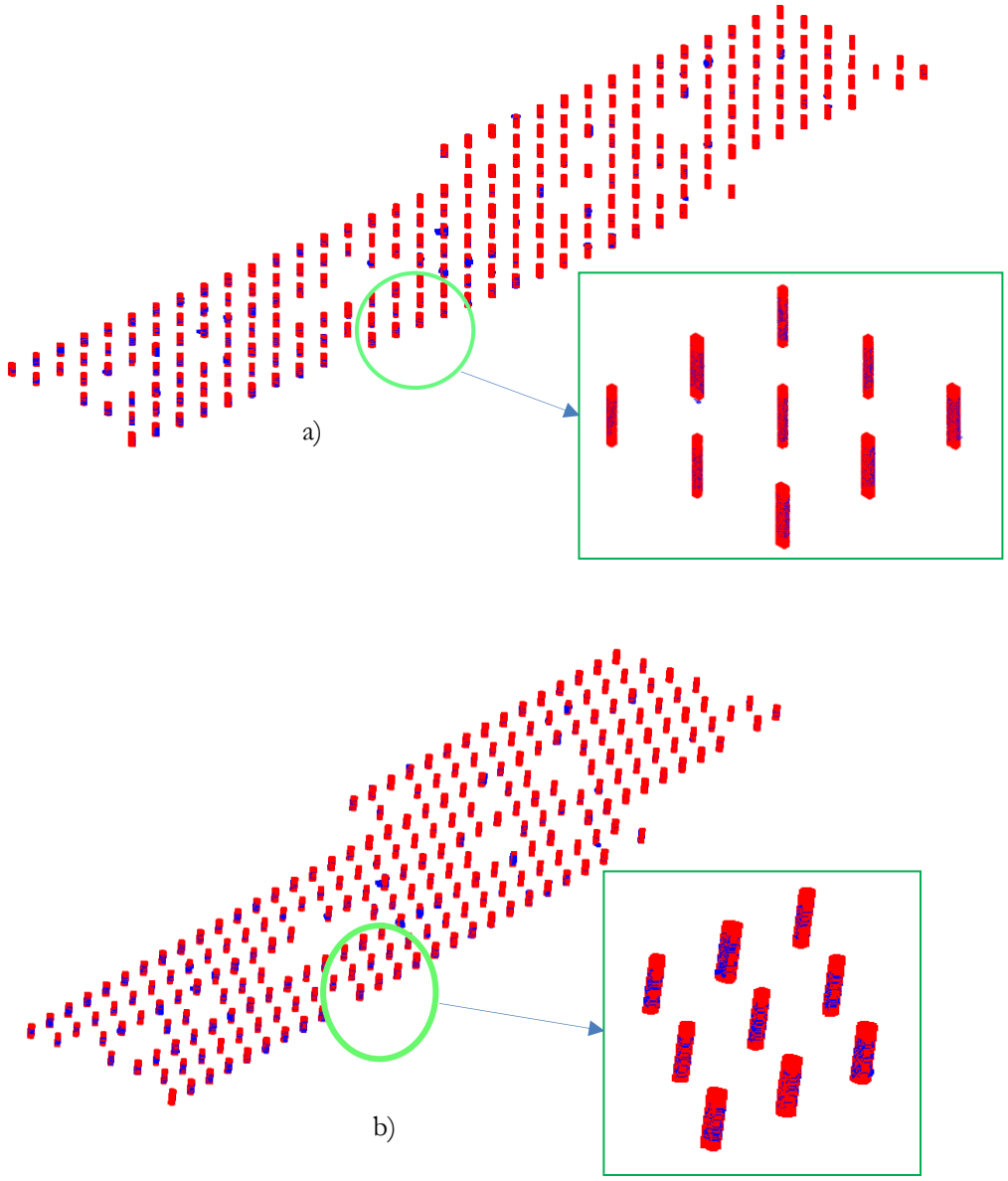


Figure 5.13: Overlay of the inlier columns of TLS and BIM datasets; a) after transformation using PCA, b) after fine registration using ICP

As described in methodology section 4.6, we have also computed the cloud-to-cloud distance between the transformed point cloud columns (the inliers) and the reference BIM model columns, as illustrated in Fig.5.14 and Fig.5.15. We have achieved the average mean distance of 3.1cm ± 4.4cm after registration using both PCA and refinement registration using the ICP. This shows that the point cloud alignment using PCA followed by the 'congruent column sets' already gives us a comparable result as refinement registration using ICP followed by 'congruent column sets'. The red points in Fig.5.14 and Fig.5.15 are the outliers in the detected columns that cause the discrepancy in the clod to cloud distance.

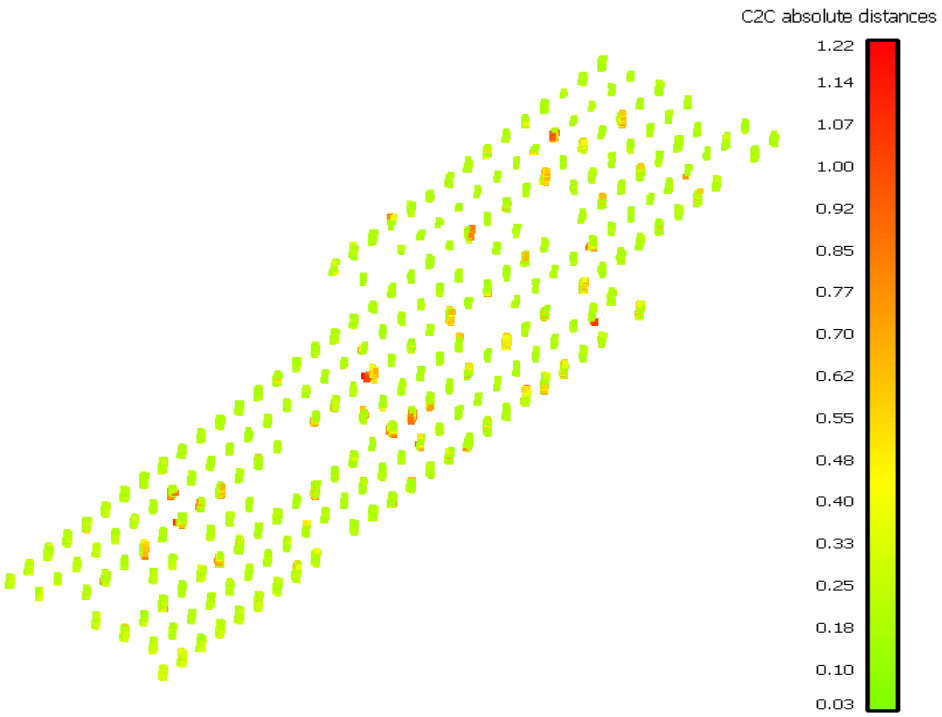


Figure 5.14: Cloud to cloud distance between the TLS Point cloud inlier columns to the corresponding BIM model columns(in m) after registration using PCA. Colors are relative to the minimum and maximum distance value

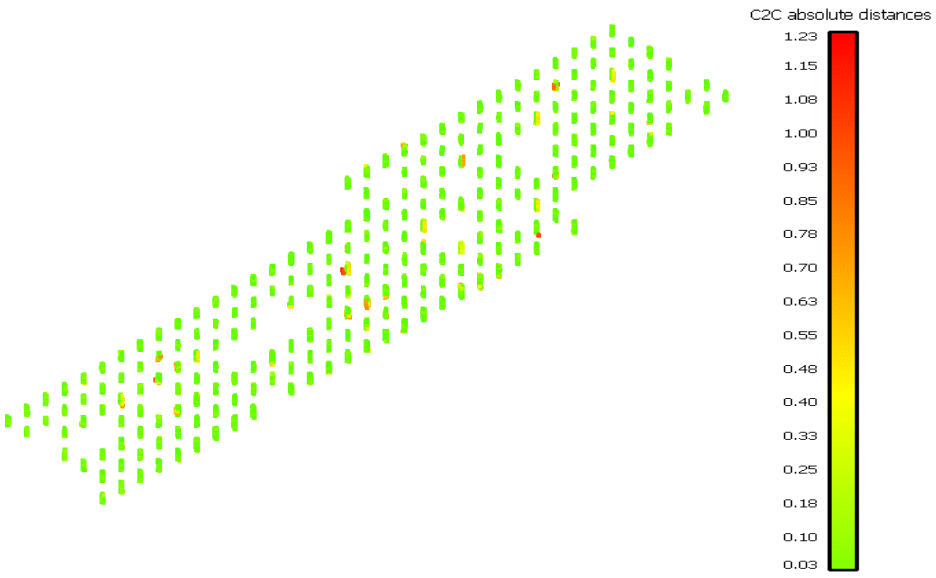
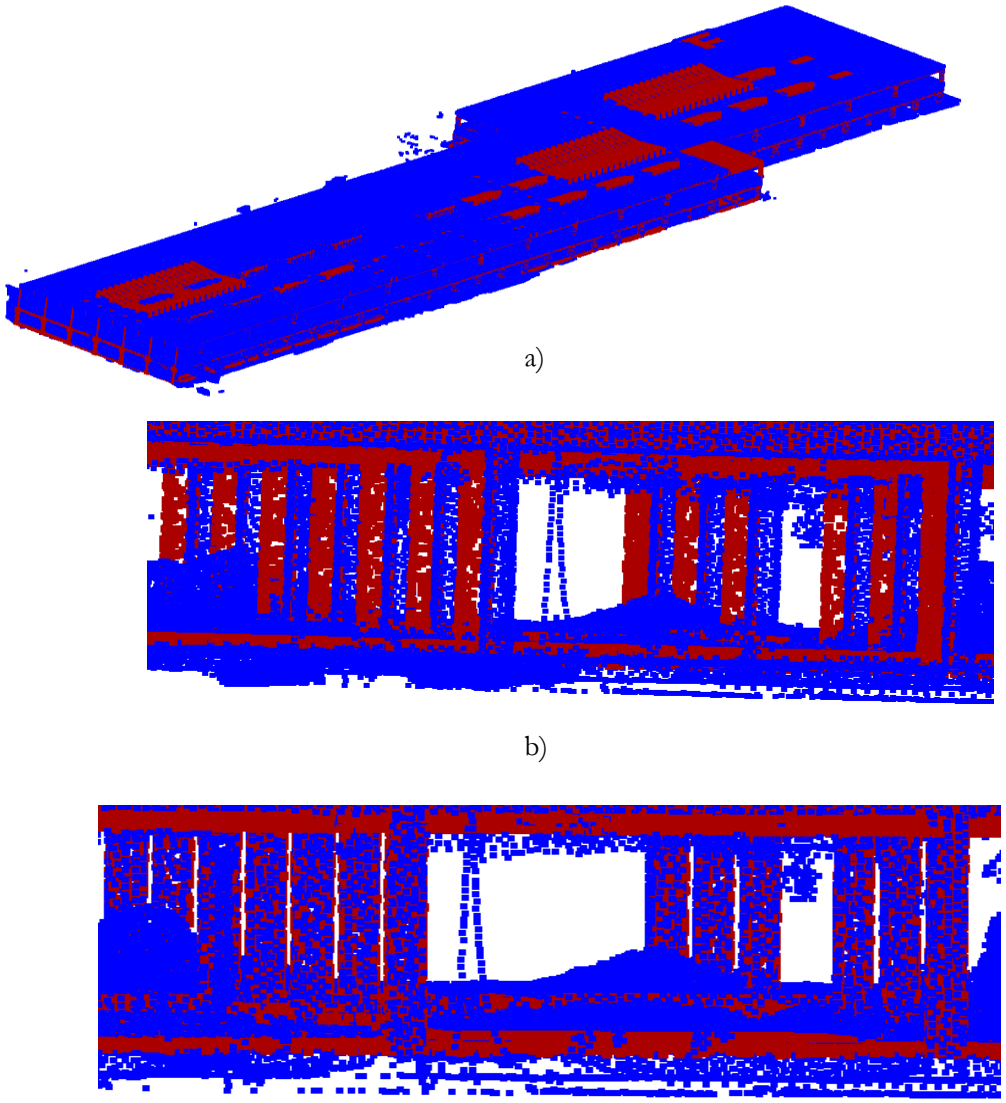


Figure 5.15: Cloud to cloud distance between the TLS Point cloud inlier columns to the corresponding BIM model columns(in m) after fine registration using ICP. Colors are relative to the minimum and maximum values.

We have applied the 3D transformation obtained using the 'congruent column sets' and PCA on the entire TLS point cloud and illustrated the result in Fig.5.16.



a)

b)

c)

r

Figure 5.16: Overlay of the TLS-point cloud (colored blue) with the BIM model point cloud (colored red) after transformation: a) entire transformed building front view, b) interior view after transformation using 'congruent column sets', b) interior view after transformation using PCA

ii. Test for symmetry and self-similarities in the TLS dataset

For this test, we used the TLS dataset in a similar configuration, as illustrated in Fig.5.10. The input parameters used for our developed 'congruent column sets' are:

- Maximum iteration= 100
- inlier ratio= 90%
- threshold distance= 30cm

Case I: Registration using a left section of source data points as illustrated in Fig.5.10a. The selected source data points contain 139 column centroids. The outlier ratio in the selected dataset is about 3%. Note that in this case, our proposed method selects the 3% candidate bases only for the left part illustrated in Fig.5.10a (149 column centroids), as discussed in more detail in the methodology section 4.5. The selected

source data points and the entire target data points illustrated in Fig.5.8a were loaded into the registration algorithm, and the results are summarized in Table 5.4.

Table 5.4: Summary of the registration result for the left part TLS source data point

'congruent column sets' RMSE (m)		PCA RMSE (m)		Inliers within 30cm threshold & 90% inlier ratio	Total iteration taken	Iteration time (sec)
2D	3D	2D	3D			
0.04	0.42	0.024	0.12	126	164	212

As seen from Table 5.4, when compared to the case for the entire point cloud column registration result in Table5.3, the average iteration and computational time needed to get the best transformation is higher. This might be happened because of many congruent pairs of column centroids in the target data points due to the repetitive column pattern for the selected 3% candidate bases of column centroids in the source data points. The algorithm could get the correct transformation parameters at each execution of the code, as illustrated in Fig.5.17.

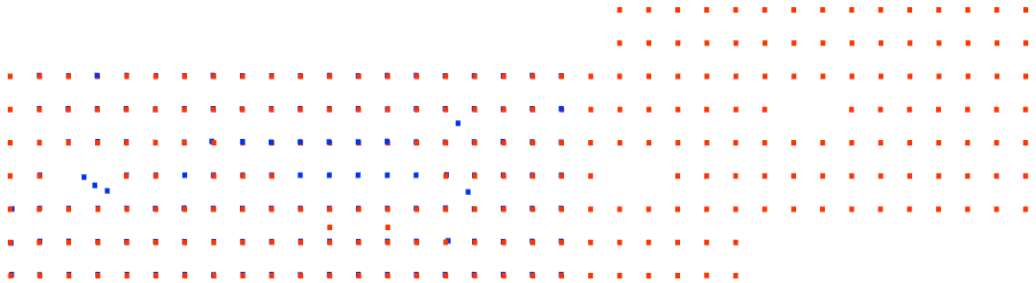


Figure 5.17: coarse registration result using 'congruent column sets' algorithm: the blue colored dots represent the source data point, and the red ones represent the target data points

Case III: Coarse registration using a top part of source data points (see Fig.5.10b).

The selected source data point contains 134 column centroids, and the outlier ratio in the selected dataset is less than 3%. During the algorithm's execution with all the given parameters, the selected source data points were not correctly aligned due to the symmetry and self-similarity (a repetitive pattern of columns) in the dataset (see Figure 18). The column centroids in the green circle are the misaligned point cloud column centroids. The rotation and translation along the y-axis are correctly estimated, but the translation along x- the axis is wrong. In this situation, the correct transformation requires more support (inliers) from the source data point.

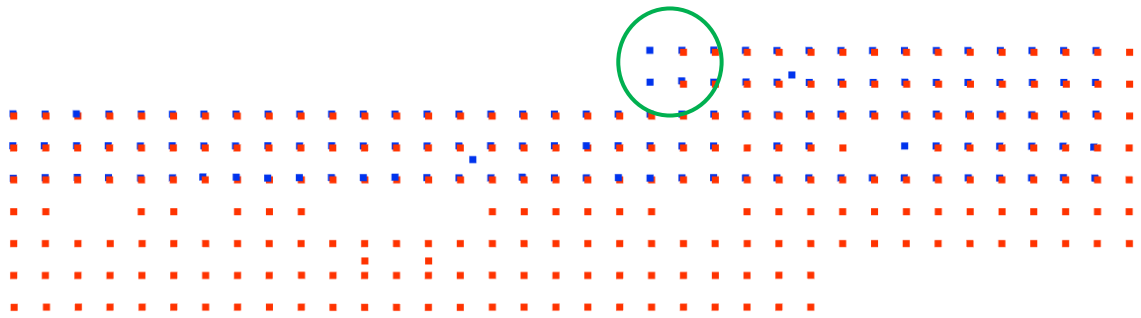


Figure 5.18: Wrongly aligned source data point (blue color) to the target data point(red color) due to a significant symmetry and repetitive column pattern in the dataset.

For this limitation, we suggest that the user should use a more strict parameter setting for the inlier ratio at the cost of computational time. Since, in our case, the source data point contains about a 3% outlier ratio, we achieved the correct transformation by increasing the inlier ratio from 90% to 95%, keeping the other parameters fixed, and the correct transformation was achieved, as illustrated in Fig.5.19.

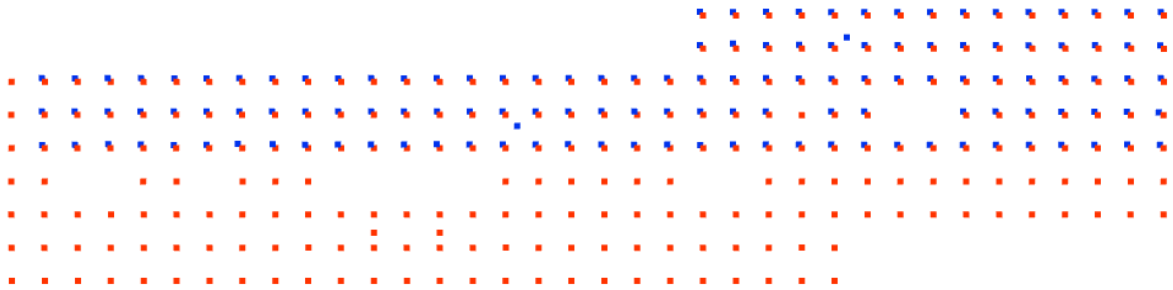


Figure 5.19:The correct transformation result with the 95% inlier ratio to avoid the problem of high symmetry in the dataset. The blue dots in the figure represent the source data points

The results for the correct transformation are summarized in Table 5.5.

Table 5.5: Summary of the registration result for the top part TLS source data point

'congruentcolumn sets' RMSE (m)		PCA RMSE (m)		Inliers within 30cm threshold & 95% inlier ratio	Total iteration taken	Iteration time (sec)
2D	3D	2D	3D			
0.04	0.43	0.026	0.12	130	57	63

5.4.2. Registration of the UAV point cloud to the BIM model

In section 5.1, we have illustrated the UAV point cloud columns detected using semantic segmentation, then applied a post-processing technique to filter the outliers, and finally computed the centroids as illustrated in Fig.5.8c. The total number of computed column centroids is 274, with about 15% outliers. The detected UAV columns are noisier than that of the TLS ones due to the low-quality consumer-grade sensors used in UAVs compared to the TLS (see Fig.5.20). This might affect the detected column centroid relative to the TLS. For this reason, the parameters used in our 'congruent column sets' algorithm for the TLS point cloud alignment don't work for the UAV ones. Consequently, we relaxed some of the parameters for our 'congruent column sets' algorithm to consider the effect of an outlier in the detected UAV columns. We used an 80% inlier ratio and a 50cm threshold distance, which was 90% and 30cm, respectively, in the case of TLS.

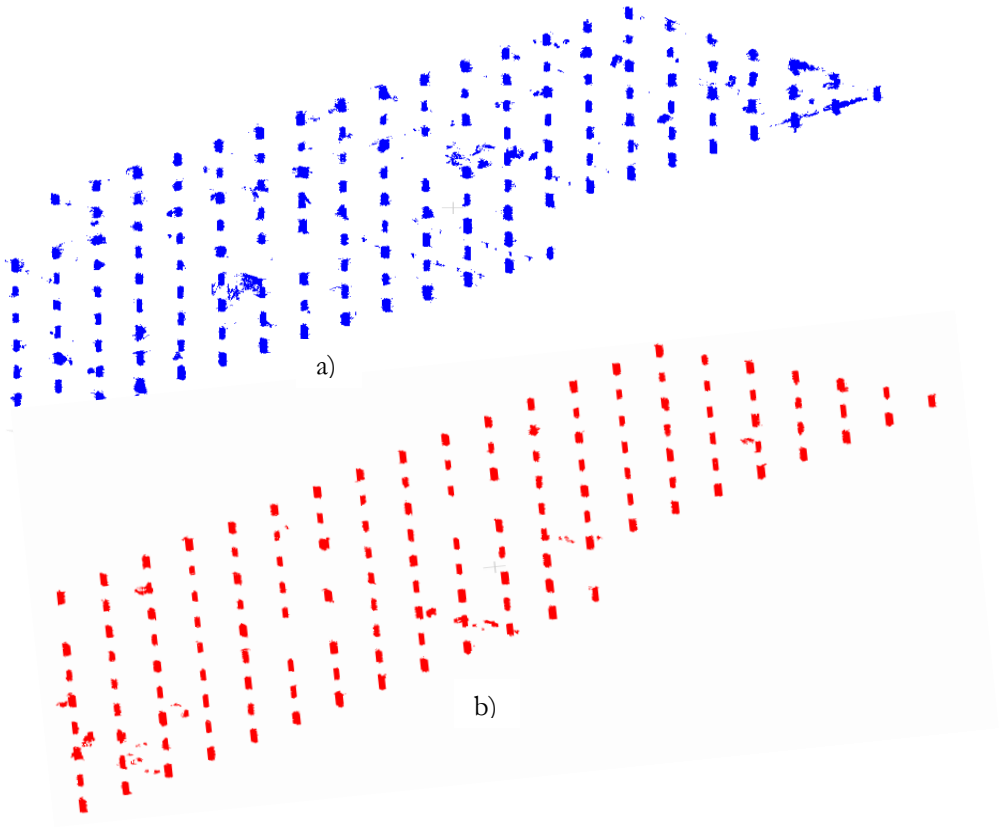


Figure 5.20: Part of the detected columns from the deep learning model : (a) UAV dataset, (b) TLS dataset.

We have applied registration experimentation to the UAV dataset using our proposed method in the same fashion as we did for the TLS dataset.

i. Test for rotation invariance

We applied the rotation invariance test for the UAV dataset in the same fashion as we did for the TLS point cloud in section 5.4.1 and summarized the results in Table 5.6. The parameters used for the 'congruent column sets' algorithm are:

- Maximum iteration=100
- inlier ratio= 80%
- threshold distance= 50cm

Table 5.6: Summary of the transformation results for the various orientation of the UAV source data points

Initial Angle (degree)		0^0	30^0	45^0	60^0	75^0
Rotation Error (degree)		0	0.03	0.03	0.04	0.03
Inliers		229	229	229	229	229
Total iteration		176	120	170	175	192
Run time (sec)		97	62	76	86	119
'congruent column sets' RMSE (m)	2D	0.082	0.095	0.096	0.09	0.094
RMSE (m)	3D	0.21	0.21	0.20	0.19	0.21
PCA RMSE (m)	2D	0.068	0.072	0.064	0.075	0.073
	3D	0.17	0.17	0.17	0.17	0.18
ICP RMSE (m)	3D	0.18	-	-	-	-

As illustrated in Table 5.6 and Fig.5.21, with all the initially applied orientations, the developed alignment approach correctly registers the UAV point cloud in the coordinate frame of the BIM model with an average rotation error of 0.03. The developed approach has achieved the average RMSE of 9cm in 2D and 20cm in 3D using the 'congruent column sets' and 7cm and 17cm, respectively, using the PCA. Also, we have achieved an RMSE of 18cm in 3D using ICP on top of the 'congruent column sets'. The registration algorithm takes 166 iterations and 86 sec to achieve the optimal result. This shows the UAV dataset costs higher computational time compared to the TLS dataset. This might be due to the higher fraction of outliers in the UAV dataset than in the TLS ones.

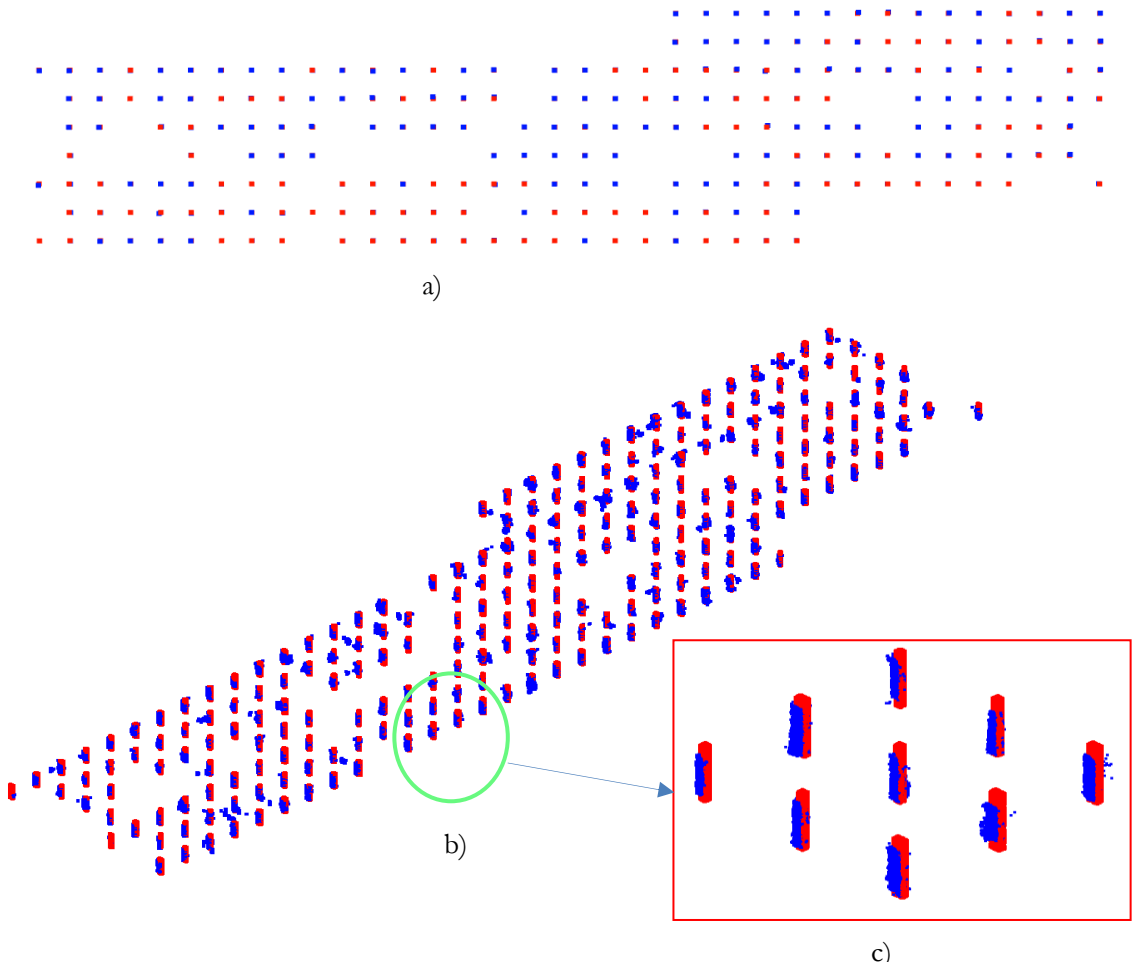
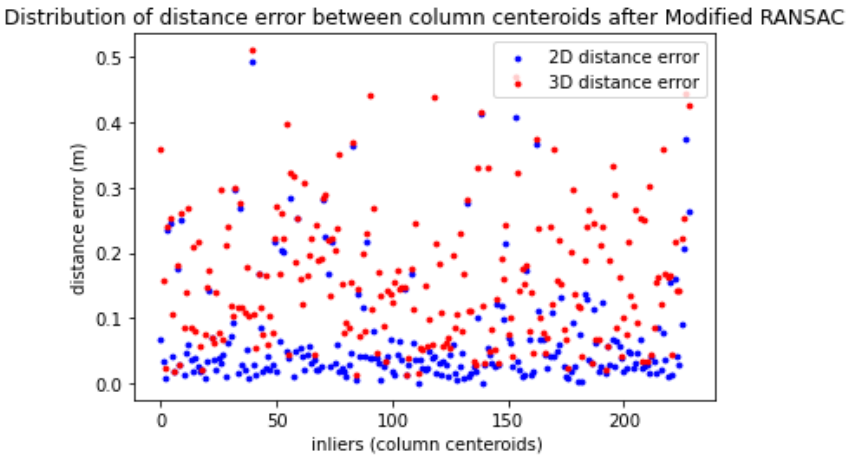
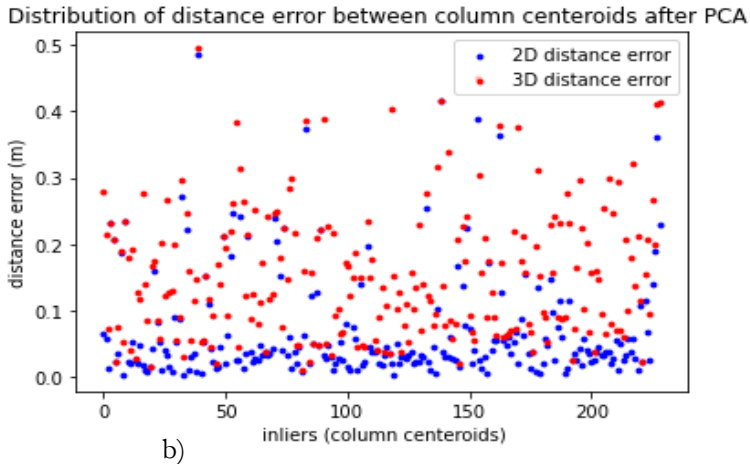


Figure 5.21: Overlay of columns detected from UAV-based point cloud (blue colored) and the BIM (red-colored) after registration using the 'congruent column sets': a) overlay based on inliers of 3D centroids of the detected columns, b) overlay based on the inliers of detected entire columns from both dataset

As explained in section 5.4.1 with the TLS dataset case, the rise in 3D RMSE is caused by the biased centroid estimation due to the variation in the detected column height of both datasets. We have illustrated the distribution of the distance between the corresponding column centroids after alignment in both 2D and 3D in Fig.5.22.



a)



b)

Figure 5.22: The 2D and 3D distance error between the inlier of the UAV and BIM model columns after transformation; a) using the 'congruent column sets' algorithm, b) after the PCA

As summarized in Table 5.6, we have achieved the RMSE of about 18cm using the classical ICP and 17cm using PCA on top of the coarse registration using 'congruent column sets'. Fig.5.23 illustrates the overlay of inlier columns detected from the UAV-based point cloud after applying registration using both PCA and ICP on top of the 'congruent column sets'-based coarse registration.

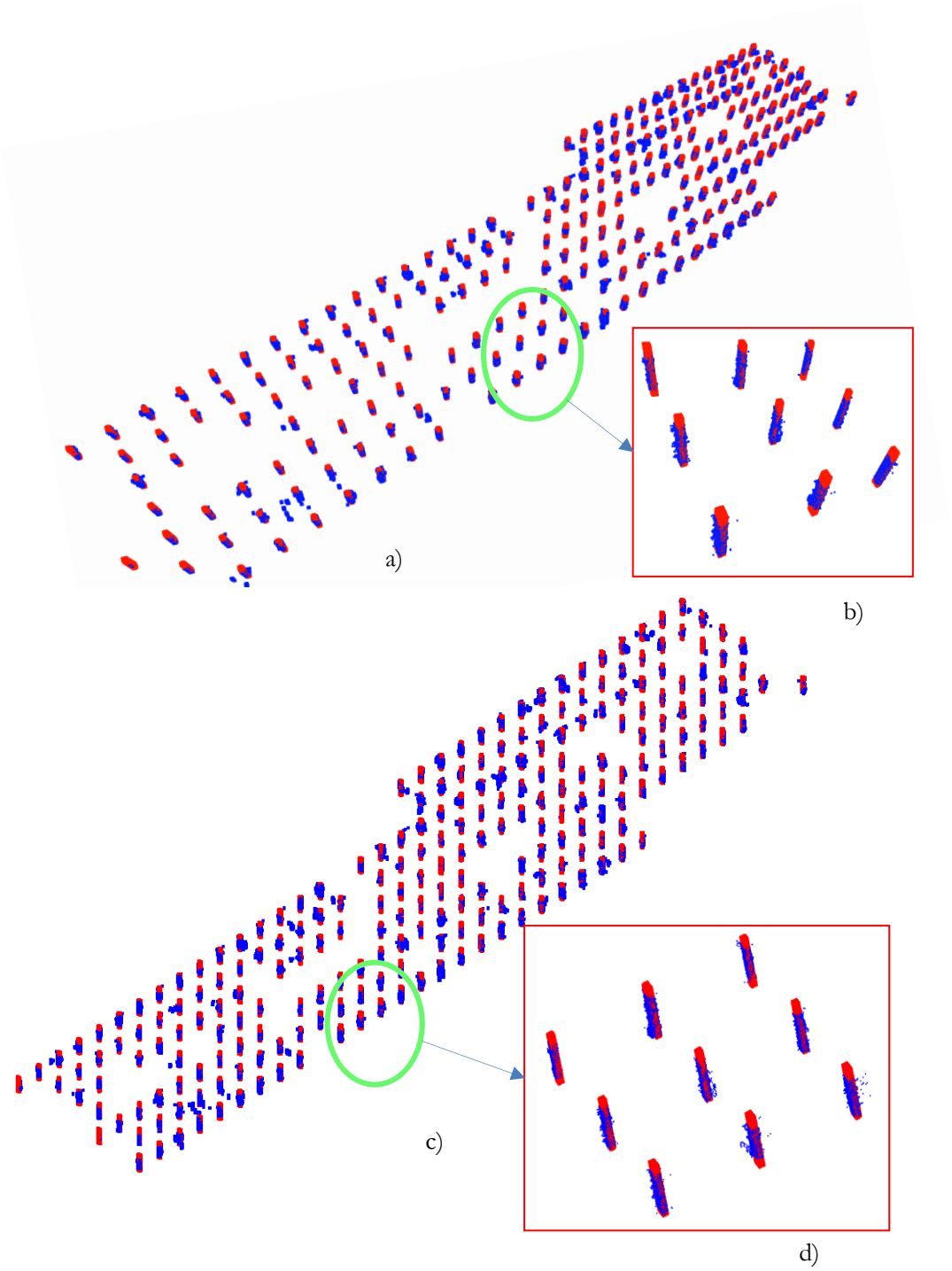


Figure 5.23: Overlay of columns detected from UAV-based point cloud (blue colored) and the BIM model (red-colored) after registration: a) using PCA, b) fine registration using ICP

We have computed the cloud-to-cloud distance between the transformed UAV point cloud columns (the inliers) and the reference BIM model columns, as illustrated in Fig.5.24. We have achieved the average mean distance of 10.8cm ± 22.5cm after registration using PCA and 10cm ± 22cm after refinement registration using ICP, followed by 'congruent column sets'. This is a lower registration accuracy when compared to the result achieved with the TLS one in section 5.4.1 (3cm ± 4cm). This result was expected as the UAV-based point clouds are noisier than the TLS ones. In addition to this, the registration using the PCA followed by the 'congruent column sets' approach has achieved almost comparable cloud to model distance as with the refinement registration using ICP followed by 'congruent column sets'. As

illustrated in Fig.5.24, the points with red color are the outliers in the point cloud, which were misclassified as as column points during the semantic segmentation. So, they have biased the column centroid computation, which contributed to the rise in point cloud-BIM model distance results.

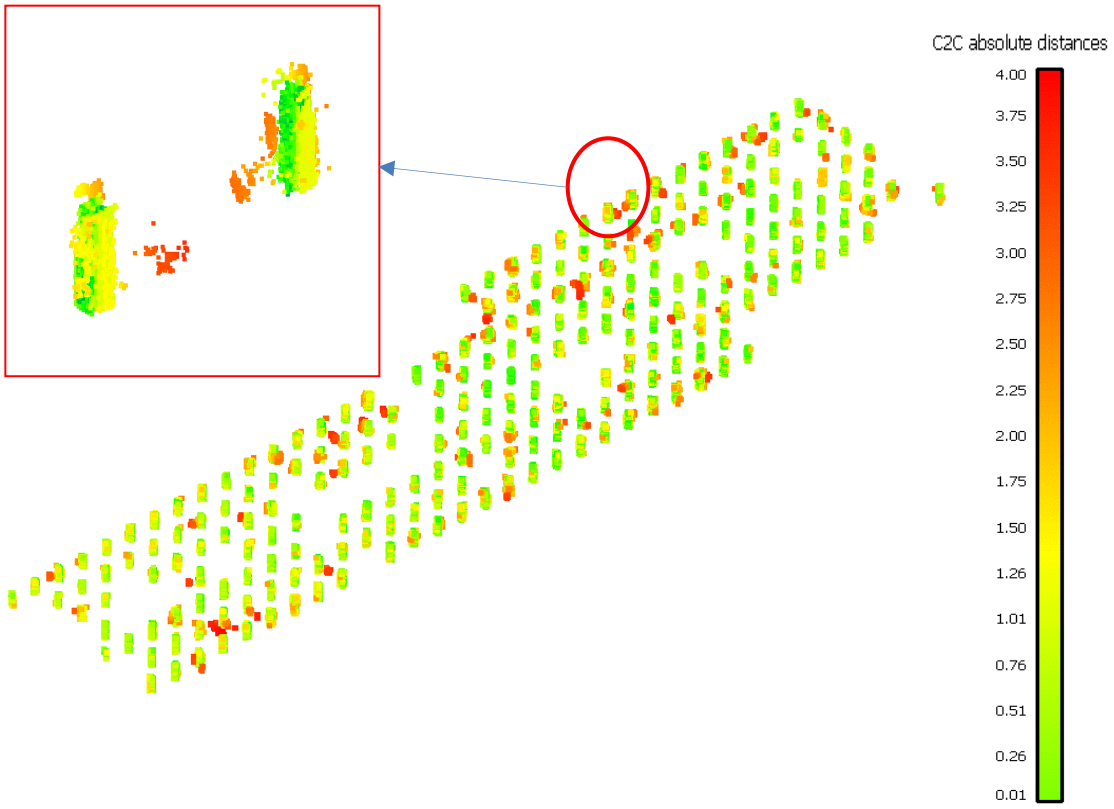
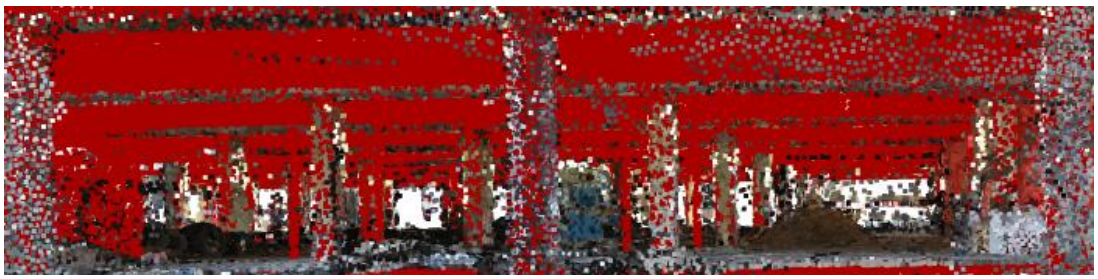


Figure 5.24:Cloud to cloud distance between the UAV Point cloud inlier columns to the corresponding BIM model columns(in m) after registration using PCA. Colors are relative to the minimum and maximum value

We have applied the 3D transformation obtained using both the 'congruent column sets' and PCA on the entire ground floor UAV point cloud to overlay with the BIM model point cloud, as illustrated in Fig.5.25.

a)



b)

c)

Figure 5.25: Overlay of the ground floor of the UAV-point cloud (true color) with the BIM model point cloud (red color) after transformation using the proposed method: a) the top view after registration with the ground floor of the UAV point cloud, b) the interior view after registration using 'congruent column sets', c) the interior view after registration using PCA

ii. Test for symmetry and self-similarities in the UAV dataset

In this case, we applied similar experimentation as we did for the TLS point cloud using the UAV dataset as the configuration illustrated in Fig.5.10. The parameters of 'congruent column sets' used for this test are:

- Maximum iteration= 100
- inlier ratio= 80%
- threshold distance= 50cm

Case I: Coarse registration using a left part of the UAV source data points (Fig.5.10a)
The total number of column centroids in the selected part is 145, out of which about 12% are outliers. The results are summarized in Table 5.7. As the case with the TLS data points, the algorithm takes many iterations to converge in this case because there are many possible congruent candidate bases due to the repetitive pattern of columns in the dataset. The higher outlier ratio in this case also increases the run time when compared to the TLS ones.

Table 5.7: Summary of the registration result for the left part UAV source data point

'congruent column sets' RMSE(m)		PCA RMSE(m)		Inliers within 50cm threshold & 80% inlier ratio	Total iteration	Iteration time(sec)
2D	3D	2D	3D			
0.06	0.16	0.05	0.15	118	279	223

Unfortunately, the transformation results in the wrong alignment for the given algorithm parameters due to symmetry in the target data points (see Fig 5.26). Due to symmetry in the target data, the UAV point cloud columns are shifted to the right side as encircled by a green outline in Fig 5.26. As discussed in section 5.4.1 with the TLS dataset, the misalignment can be improved by using more strict parameters (increasing the inlier ratio) to get more support for the correct transformation parameters, which might increase the computational time needed.

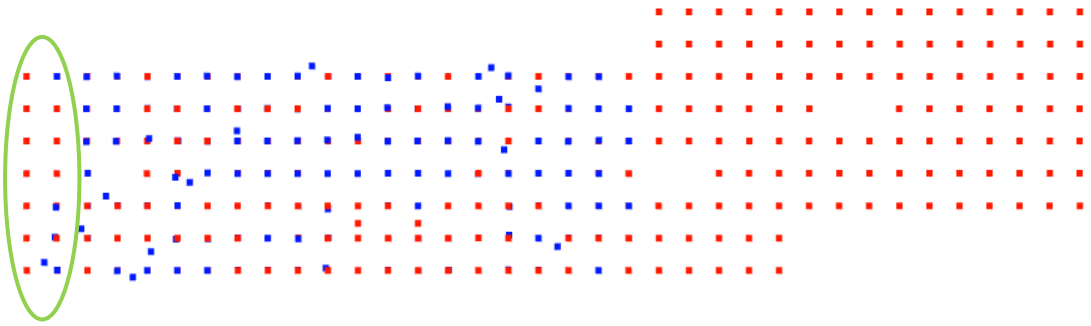


Figure 5.26: Overlay of the left part of the UAV source data point column centroids with the target data point columns after 'congruent column sets'.

Case III: Coarse registration using a top part of the UAV source data points (Fig.5.10b).
Similar to the case I, for the given parameters, our proposed registration approach resulted in the wrong alignment of the selected source data point due to the symmetry in the dataset. As discussed, this limitation could be improved by increasing the inlier ratio parameter of the algorithm to get more support from the source data point.

6. CONCLUSION AND FUTURE WORK

The rapidly growing 3D sensing technologies, such as TLS and UAV in the AEC/FM industry, have become a novel technique to provide effective ways of acquiring the as-built status of the construction site. This helps to monitor the work progress by comparing the as-built data with the as-planned status recorded in the 3D BIM model of the project. The comparison between the as-built data with the as-planned BIM model for progress monitoring requires aligning the two datasets in a common coordinate frame. Even though there is a well-developed automatic method for the fine registration of the 3D point cloud, the existing methods for the coarse registration are immature in the AEC/FM sector.

This research proposed a column-based automatic coarse registration method for the alignment of the as-built point cloud in the reference frame of the 3D as-planned BIM model for construction progress monitoring. The developed method estimates the 3D rigid transformation parameters by automatically extracting columns from the as-built point cloud to match with the corresponding columns in the BIM model of the project. Columns are chosen as the basis for registration because they are the dominant structural components of a building and are constructed at the early stages of construction. So, it is very likely that part of the structural columns is present in the construction site point cloud acquired at the early stage of structural construction (walls should not cover the columns). The proposed method extracts columns from the point cloud by semantic segmentation using deep learning. The corresponding BIM model columns are easily extracted from the structural component module. The corresponding columns are then automatically matched to estimate the rigid transformation parameter. We proposed an automated coarse registration method called 'congruent column sets' which is motivated by the basic RANSAC algorithm for column matching that is robust to the outliers in the detected point cloud columns. The research also applies PCA on top of the 'congruent column sets' to achieve accurate registration for progress monitoring.

The method uses the point-based KPConv deep learning model to semantically segment the main structural construction component (Column, Beam, Slab, Wall). A 'clutter' class was included to categorize the rest of the points that do not belong to the main structural components. The inputs to the model are an unstructured set of points (X, Y, Z) from indoor construction site point clouds and the extracted normal vector component ($n_x, n_y,$ and n_z). We replaced the color attribute in the original work with a normal vector because the normal vector components of the point cloud can represent our interest classes better than the color attribute. The model is trained and tested with publicly available indoor point clouds, point clouds from the construction site provided by the University of Edinburgh, and simulated point cloud. The trained model was evaluated on the test dataset and has achieved the mean intersection over union (mIoU) of 73% of overall classes and column segmentation accuracy of 69%. The achieved column detection result showed a significant improvement over the 16% mIoU in the developer's original work. However, the semantic segmentation model showed to have higher confusion between the wall, clutter, and column categories. The confusion between the classes was expected due to the training data imbalance and similarity between classes.

The performance of the proposed coarse registration approach was evaluated with experiments carried out on the TLS and UAV real construction site point cloud. We segmented both point clouds by the trained model, and the columns were extracted automatically. The corresponding columns from the BIM model are retrieved from the BIM's structural family using BIM software. Both the detected point cloud and BIM model columns were clustered, and the centroids were computed and loaded into the proposed coarse registration approach. Our experimentation result shows that the proposed alignment approach has reliably detected the best portion of the inlier columns that supports the estimated optimal transformation parameters. Results show that column-based registration achieved a rotation error of 0.02 degrees and an RMSE of 0.12 meters for the TLS dataset and 0.03 degrees and 0.17meters for the UAV dataset. As discussed in this study, the elevated RMSE is caused by the column height difference between the corresponding columns. The computational time required is 55 seconds for the TLS dataset and 86

seconds for the UAV datasets. Based on our experimental result, our proposed registration approach using PCA on top of the 'congruent column sets' has achieved almost a comparable result with the refinement registration using ICP. Further experiments, however, demonstrated the limitations of our proposed coarse registration method due to symmetry and self-similarities constraints in the AEC/FM context. In such cases, more strict parameter settings are required, which substantially slows down the registration process. Furthermore, this could be improved by considering more features or integrating extra parameters.

Based on the conducted experimentation on both the UAV and TLS point cloud, the quantitative and qualitative registration results show that the proposed approach can align the point cloud very reliably even without the ICP refinement. As a result, we conclude that our proposed method contributes to automating the registration between the as-built point cloud and the as-planned BIM model to monitor the construction progress (by comparing the TLS and UAV as-built point cloud with the as-planned BIM model). However, future work should address the following issues:

- Although the proposed registration approach works well with the achieved column detection accuracy of the KPConv model, the misclassified points (outliers) have biased the column centroid estimation. The result shows that the trained model has confused the temporary construction site point clouds, e.g., formworks, safety fences, and moving objects, with the columns. Further work can be done to improve semantic segmentation accuracy by adding much more diverse training and test data from the construction site (particularly temporary construction site point clouds). Also, integrating more attributes to normal vectors, such as neighborhood surface roughness and neighborhood maximum and minimum curvature. In addition to this, future work can be done by using deep learning models with state-of-the-art performance on semantic segmentation of indoor construction site point clouds.
- For obtaining accurate alignment in scenarios with abundant symmetry and self-similarity in the dataset, improvements to the matching and scoring algorithm can be made. This might be accomplished by using additional matching and acceptance criteria or considering extra distinct features from both datasets.
- It is shown in the experiments that the vertical height variation between the detected BIM model and point cloud columns significantly biased the column centroid estimation, which in turn resulted in an increased error in the vertical translation. For future work, the vertical translation will be tested, including additional semantic segmentation objects (floors and beams in both datasets.
- Our proposed method is tested on the as-built point clouds where the columns have not been covered by walls. However, the column detection using semantic segmentation fails if the columns are embedded in the walls. Therefore, in such a case, other surface feature extraction and matching approaches can provide a better automated coarse registration result for progress monitoring.

LIST OF REFERENCES

Adan, A., & Huber, D. (2011). 3D reconstruction of interior wall surfaces under occlusion and clutter. *Proceedings - 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2011*, 275–281. https://doi.org/10.1109/3DIMPVT.2011.42

Aiger, D., Mitra, N. J., & Cohen-Or, D. (2008). 4-Points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics*, *27*(3). https://doi.org/10.1145/1360612.1360684

Akca, D. (2003). *Full automatic registration of laser scanner point clouds*. https://doi.org/10.3929/ethz-a-004656666

Alsadik, B., & Nex, F. (2021a). The Rise in UAV Inspections for Civil Infrastructure. In *GIM International*. https://www.researchgate.net/publication/352910991

Alsadik, B., & Nex, F. (2021b). The Rise in UAV Inspections for Civil Infrastructure | GIM International. *GIM International*, *June*. https://www.gim-international.com/content/article/the-rise-in-uav-inspections-for-civil-infrastructure

Anil, E. B., Akinci, B., & Huber, D. (2011). Representation requirements of as-is building information models generated from laser scanned point cloud data. *Proceedings of the 28th International Symposium on Automation and Robotics in Construction, ISARC 2011*, 355–360. https://doi.org/10.22260/isarc2011/0063

Arditi, D., Arditi, D., & Gunaydin, H. M. (2015). Total quality management in the construction process Total quality management in the construction process. *International Journal of Project Management*, *7863*(AUGUST 1997), 235–243.

Arditi, D., & Gunaydin, H. M. (1997). Total quality management in the construction process. *International Journal of Project Management*, *15*(4), 235–243. https://doi.org/10.1016/S0263-7863(96)00076-2

Armeni, I., Sax, S., Zamir, A. R., & Savarese, S. (2017). *Joint 2D-3D-Semantic Data for Indoor Scene Understanding*. http://arxiv.org/abs/1702.01105

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016). *3D Semantic Parsing of Large-Scale Indoor Spaces (a) Raw Point Cloud (b) Space Parsing and Alignment in Canonical 3D Space (c) Building Element Detection Enclosed Spaces*. http://buildingparser.stanford.edu/

Arun, K. S., Huang, T. S., & Blostein, S. D. (1987). Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-9*(5), 698–700. https://doi.org/10.1109/TPAMI.1987.4767965

Bae, K. H. (2009). Evaluation of the convergence region of an automated registration method for 3D laser scanner point clouds. *Sensors*, *9*(1), 355–375. https://doi.org/10.3390/s90100355

Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-D shapes. In P. S. Schenker (Ed.), *Sensor fusion IV: control paradigms and data structures* (pp. 586–606). https://doi.org/10.1117/12.57955

Bognot, J. R., Candido, C. G., Blanco, A. C., & Montelibano, J. R. Y. (2018). BUILDING CONSTRUCTION PROGRESS MONITORING USING UNMANNED AERIAL SYSTEM (UAS), LOW-COST PHOTOGRAMMETRY, and GEOGRAPHIC INFORMATION SYSTEM (GIS). *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(2), 41–47. https://doi.org/10.5194/isprs-annals-IV-2-41-2018

Bosché, F. (2012). Plane-based registration of construction laser scans with 3D/4D building models. *Advanced Engineering Informatics*, *26*(1), 90–102. https://doi.org/10.1016/j.aei.2011.08.009

Bosche, F., Haas, C. T., & Murray, P. (2008). Performance of automated project progress tracking with 3D data fusion. *Proceedings, Annual Conference - Canadian Society for Civil Engineering*, *1*, 349–358.

Bueno, M., Bosché, F., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2018a). 4-Plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models. *Automation in Construction*, *89*, 120–134. https://doi.org/10.1016/j.autcon.2018.01.014

Bueno, M., Bosché, F., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2018b). 4-Plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models. *Automation in Construction*, *89*, 120–134. https://doi.org/10.1016/j.autcon.2018.01.014

Camarillo, M. K., Basha, E., & Khan, M. S. (2020). Integration of unmanned aerial vehicles and aerial photogrammetry into a civil engineering course to enhance technology competency. *ASEE Annual Conference and Exposition, Conference Proceedings*, *2020-June*. https://doi.org/10.18260/1-2--34855

Chen, J., & Cho, Y. K. (2018). Point-to-point Comparison Method for Automated Scan-vs-BIM Deviation Detection. {*Proceedings of the 2018 17th International Conference on Computing in Civil and Building Engineering, Tampere, Finland*, 5–7. https://www.academia.edu/download/56812721/JingdaoChen_full_paper2_icccbe2018.pdf

Chen, J., Cho, Y. K., Kim, K., & Student, P. D. (2018). Region Proposal Mechanism for Building Element Recognition for Advanced Scan-to-BIM Process. *Construction Research Congress 2018*, 221–231. https://doi.org/10.1061/9780784481264.022

Chen, J., Kira, Z., & Cho, Y. K. (2019). Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction. *Journal of Computing in Civil Engineering*, *33*(4). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000842

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, *10*(3), 145–155. https://doi.org/10.1016/0262-8856(92)90066-C

Cho, Y. K., Asce, A. M., & Gai, M. (2014). Projection-Recognition-Projection Method for Automatic Object Recognition and Registration for Dynamic Heavy Equipment Operations. *American Society of Civil Engineers.*, 1–9. https://doi.org/10.1061/(ASCE)CP

Dai, F., Rashidi, A., Brilakis, I., & Vela, P. (2013). Comparison of Image-Based and Time-of-Flight-Based Technologies for Three-Dimensional Reconstruction of Infrastructure. *Journal of Construction Engineering and Management*, *139*(1), 69–79. https://doi.org/10.1061/(asce)co.1943-7862.0000565

Dimitrov, A., & Golparvar-Fard, M. (2015). Segmentation of building point cloud models including detailed architectural/structural features and MEP systems. *Automation in Construction*, *51*(C), 32–45. https://doi.org/10.1016/J.AUTCON.2014.12.015

Dold, C., & Brenner, C. (2006). REGISTRATION OF TERRESTRIAL LASER SCANNING DATA USING PLANAR PATCHES AND IMAGE DATA. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives 36 (2006)*, *36*, 78–83. https://doi.org/https://doi.org/10.15488/3750

Dorai, C., Weng, J., & Jain, A. K. (1994). Optimal registration of multiple range views. *Proceedings of 12th International Conference on Pattern Recognition*, *1*, 569–571. https://doi.org/10.1109/ICPR.1994.576362

El-Omari, S., & Moselhi, O. (2009). Data acquisition from construction sites for tracking purposes. *Engineering, Construction and Architectural Management*, *16*(5). https://doi.org/10.1108/09699980910988384

Engelmann, F., Kontogianni, T., Hermans, A., & Leibe, B. (2017). *Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds*.

Fischler, M. A., & Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with. *Communications of the ACM*, *24*(6).

Girardeau-Montaut, D. (2015). CloudCompare: 3D point cloud and mesh processing software. *Webpage: Http://Www.Cloudcompare.Org*.

Golparvar-Fard, M., Peña-Mora, F., Gutgsell Endowed Professor, J., Savarese, S., & Professor, A. (2009). D 4 AR-A 4-DIMENSIONAL AUGMENTED REALITY MODEL FOR AUTOMATING CONSTRUCTION PROGRESS MONITORING DATA COLLECTION, PROCESSING AND COMMUNICATION Application of D 4 AR-A 4-Dimensional augmented reality model for automating construction progress monitoring data collection. In *Journal of Information Technology in Construction (ITcon)* (Vol. 14). http://www.itcon.org/2009/13

Greenwood, W. W., Lynch, J. P., & Zekkos, D. (2019). Applications of UAVs in Civil Infrastructure. *Journal of Infrastructure Systems*, *25*(2), 04019002. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000464

GSA. (2009). GSA BIM Guide Series 05 - BIM Guide for Energy Performance. *Energy*, *February*.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2020). Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/tpami.2020.3005434

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2021). Deep Learning for 3D Point Clouds: A Survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Vol. 43, Issue 12, pp. 4338–4364). IEEE Computer Society. https://doi.org/10.1109/TPAMI.2020.3005434

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., & Pollefeys, M. (2017). *Semantic3D.net: A new Large-scale Point Cloud Classification Benchmark*. http://arxiv.org/abs/1704.03847

Hackel, T., Wegner, J. D., & Schindler, K. (2016). Contour Detection in Unstructured 3D Point Clouds. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *2016-Decem*, 1610–1618. https://doi.org/10.1109/CVPR.2016.178

Han, X.-F., Sun, S.-J., Song, X.-Y., & Xiao, G.-Q. (2018). *3D Point Cloud Descriptors in Hand-crafted and Deep Learning Age: State-of-the-Art*. http://arxiv.org/abs/1802.02297

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. http://image-net.org/challenges/LSVRC/2015/

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Josa a*, *4*(4).

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2019). *RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds*. http://arxiv.org/abs/1911.11236

Hua, B.-S., Tran, M.-K., & Yeung, S.-K. (2018). Pointwise Convolutional Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 984–993.

Huang, J., & You, S. (2013). Detecting objects in scene point cloud: A combinational approach. *Proceedings - 2013 International Conference on 3D Vision, 3DV 2013*, 175–182. https://doi.org/10.1109/3DV.2013.31

Ip, C. Y., & Gupta, S. K. (2007a). Retrieving Matching CAD Models by Using Partial 3D Point Clouds. *Computer-Aided Design and Applications*, *4*(5), 629–638. https://doi.org/10.1080/16864360.2007.10738497

Ip, C. Y., & Gupta, S. K. (2007b). Retrieving Matching CAD Models by Using Partial 3D Point Clouds. *Computer-Aided Design and Applications*, *4*(5), 629–638. https://doi.org/10.1080/16864360.2007.10738497

Jacob-Loyola, N., Muñoz-La Rivera, F., Herrera, R. F., & Atencio, E. (2021). Unmanned aerial vehicles (Uavs) for physical progress monitoring of construction. *Sensors*, *21*(12). https://doi.org/10.3390/s21124227

Jaw, J., & Chuang, T. (2008). Feature-based registration of terrestrial lidar point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS)*, *XXXVII*(Part B3b).

Jaw, J. J., & Chuang, T. Y. (2008a). FEATURE-BASED REGISTRATION OF TERRESTRIAL LIDAR POINT CLOUDS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *37*, 303–308.

Jaw, J. J., & Chuang, T. Y. (2008b). Registration of ground-based LiDAR point clouds by means of 3D line features. *Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers,Series A/Chung-Kuo Kung Ch'eng Hsuch K'an*, *31*(6), 1031–1045. https://doi.org/10.1080/02533839.2008.9671456

Jaw, J. J., & Chuang, T. Y. (2008c). Registration of ground-based LiDAR point clouds by means of 3D line features. *Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers,Series A/Chung-Kuo Kung Ch'eng Hsuch K'an*, *31*(6). https://doi.org/10.1080/02533839.2008.9671456

Johnson, A. E., & Hebert, M. (1998). Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, *16*(9–10). https://doi.org/10.1016/s0262-8856(98)00074-2

Jung, J., Hong, S., Yoon, S., Kim, J., & Heo, J. (2016). Automated 3D Wireframe Modeling of Indoor Structures from Point Clouds Using Constrained Least-Squares Adjustment for As-Built BIM. *Journal of Computing in Civil Engineering*, *30*(4), 04015074. https://doi.org/10.1061/(asce)cp.1943-5487.0000556

Kaiming, H., Xiangyu, Z., Shaoqing, R., & Jian, S. (2016). Deep Residual Learning for Image Recognition Kaiming. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. http://image-net.org/challenges/LSVRC/2015/

Kaiser, T., Clemen, C., & Maas, H.-G. (2022). Automatic co-registration of photogrammetric point clouds with digital building models. *Automation in Construction*, *134*, 104098. https://doi.org/10.1016/j.autcon.2021.104098

Kim, C., Kim, C., & Son, H. (2013). Automated construction progress measurement using a 4D building information model and 3D data. *Automation in Construction*, *31*, 75–82. https://doi.org/10.1016/j.autcon.2012.11.041

Kim, C., Lee, J., Cho, M., & Kim, C. (2011). Fully automated registration of 3D CAD model with point cloud from construction site. *Proceedings of the 28th International Symposium on Automation and Robotics in Construction, ISARC 2011*. https://doi.org/10.22260/isarc2011/0169

Kim, S., Kim, S., & Lee, D. E. (2020). Sustainable application of hybrid point cloud and BIM method for tracking construction progress. *Sustainability (Switzerland)*, *12*(10), 1–16. https://doi.org/10.3390/su12104106

Kiziltas, S., & Akinci, B. (2005). The need for prompt schedule update by utilizing reality capture technologies: A case study. *Construction Research Congress 2005: Broadening Perspectives - Proceedings of the Congress*, 321–330. https://doi.org/10.1061/40754(183)32

Landrieu, L., & Simonovsky, M. (2017). *Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs*. http://arxiv.org/abs/1711.09869

Li, J., Huang, H., Du, L., & Zhang, X. (2018). 3D Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation Xiaoqing Ye 1,2[0000−0003−3268−880X]. *European Conf. on Computer Vision (ECCV)*, 415-430.

Lin, J. J., Han, K. K., & Golparvar-Fard, M. (2015). A framework for model-driven acquisition and analytics of visual data using UAVs for automated construction progress monitoring. *Congress on*

Computing in Civil Engineering, Proceedings, *2015-January*(January), 156–164. https://doi.org/10.1061/9780784479247.020

Low, K.-L. (2004). Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration. *Chapel Hill, University of North Carolina*, *4*(10), 1–3. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.116.7292&rep=rep1&type=pdf

M Jiang, Y Wu, T Zhao, Z Zhao, & C Lu. (2018). PointSIFT-A SIFT-like Network Module for 3D Point Cloud SemanticSegmentation. *ArXiv Preprint ArXiv:1807.00652*.

Mahmood, B., & Han, S. (2019). 3D Registration of Indoor Point Clouds for Augmented Reality. *Computing in Civil Engineering 2019*, 1–8. https://doi.org/10.1061/9780784482421.001

Mahmood, B., Han, S. U., & Lee, D. E. (2020). BIM-based registration and localization of 3D point clouds of indoor scenes using geometric features for augmented reality. *Remote Sensing*, *12*(14). https://doi.org/10.3390/RS12142302

McCabe, B. Y., Hamledari, H., Shahi, A., Zangeneh, P., & Azar, E. R. (2017). Roles, Benefits, and Challenges of Using UAVs for Indoor Smart Construction Applications. *Congress on Computing in Civil Engineering, Proceedings*, *2017-June*, 349–357. https://doi.org/10.1061/9780784480830.043

Muja, M., & Lowe, D. G. (2009). FAST APPROXIMATE NEAREST NEIGHBORS WITH AUTOMATIC ALGORITHM CONFIGURATION. *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, *1*, 331–340. https://doi.org/10.5220/0001787803310340

Nahangi, M., Heins, A., McCabe, B., & Schoellig, A. (2018). Automated localization of UAVs in GPS-denied indoor construction environments using fiducial markers. *ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things*. https://doi.org/10.22260/isarc2018/0012

Nguyen, C. H. P., & Choi, Y. (2018a). Parametric comparing for local inspection of industrial plants by using as-built model acquired from laser scan data. *Computer-Aided Design and Applications*, *15*(2), 238–246. https://doi.org/10.1080/16864360.2017.1375675

Nguyen, C. H. P., & Choi, Y. (2018b). Comparison of point cloud data and 3D CAD data for on-site dimensional inspection of industrial plant piping systems. *Automation in Construction*, *91*, 44–52. https://doi.org/10.1016/j.autcon.2018.03.008

Omar, T., & Nehdi, M. L. (2016a). Data acquisition technologies for construction progress tracking. *Automation in Construction*, *70*, 143–155. https://doi.org/10.1016/j.autcon.2016.06.016

Omar, T., & Nehdi, M. L. (2016b). Data acquisition technologies for construction progress tracking. In *Automation in Construction* (Vol. 70, pp. 143–155). Elsevier B.V. https://doi.org/10.1016/j.autcon.2016.06.016

Perez-Perez, Y., Golparvar-Fard, M., & El-Rayes, K. (2021a). Scan2BIM-NET: Deep Learning Method for Segmentation of Point Clouds for Scan-to-BIM. *Journal of Construction Engineering and Management*, *147*(9). https://doi.org/10.1061/(asce)co.1943-7862.0002132

Perez-Perez, Y., Golparvar-Fard, M., & El-Rayes, K. (2021b). Segmentation of point clouds via joint semantic and geometric features for 3D modeling of the built environment. *Automation in Construction*, *125*(March 2020), 103584. https://doi.org/10.1016/j.autcon.2021.103584

Pətrəucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., & Haas, C. (2015). State of research in automatic as-built modelling. *Advanced Engineering Informatics*, *29*(2), 162–171. https://doi.org/10.1016/j.aei.2015.01.001

Pottmann, H., Huang, Q. X., Yang, Y. L., & Hu, S. M. (2006). Geometry and convergence analysis of algorithms for registration of 3D shapes. In *International Journal of Computer Vision* (Vol. 67, Issue 3, pp. 277–296). https://doi.org/10.1007/s11263-006-5167-2

Previtali, M., Barazzetti, L., Brumana, R., & Scaioni, M. (2014). Towards automatic indoor reconstruction of cluttered building rooms from point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(5), 281–288. https://doi.org/10.5194/isprsannals-II-5-281-2014

Pu, S., & Vosselman, G. (2009). Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *64*(6), 575–584. https://doi.org/10.1016/j.isprsjprs.2009.04.001

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017a). PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*. https://doi.org/10.1109/CVPR.2017.16

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017b). PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 77–85. https://doi.org/10.1109/CVPR.2017.16

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. http://arxiv.org/abs/1706.02413

Romero-Jarén, R., & Arranz, J. J. (2021). Automatic segmentation and classification of BIM elements from point clouds. *Automation in Construction*, *124*(January). https://doi.org/10.1016/j.autcon.2021.103576

Rusinkiewicz, S. (2019). A symmetric objective function for ICP. *ACM Transactions on Graphics*, *38*(4). https://doi.org/10.1145/3306346.3323037

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 145–152. https://doi.org/10.1109/IM.2001.924423

Rusu, R. B., Blodow, N., & Beetz, M. (2009). Fast Point Feature Histograms (FPFH) for 3D registration. *2009 IEEE International Conference on Robotics and Automation*, 3212–3217. https://doi.org/10.1109/ROBOT.2009.5152473

Ruwen Schnabel, D.-I., Klein, R., & Gumhold, S. (2010). *Efficient Point-Cloud Processing with Primitive Shapes*.

Scott, S., & Assadi, S. (1999). A survey of the site records kept by construction supervisors. *Construction Management and Economics*, *17*(3), 375–382. https://doi.org/10.1080/014461999371574

Segal, A. v, Haehnel, D., & Thrun, S. (2009). Generalized-ICP. *Robotics: Science and Systems*, 435.

Son, H., Bosché, F., & Kim, C. (2015). *As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey*. https://doi.org/10.1016/j.aei.2015.01.009

Swetha Koppula, H., Anand, A., Joachims, T., & Saxena, A. (2011). *Semantic Labeling of 3D Point Clouds for Indoor Scenes*. 1–9. http://pr.cs.cornell.edu/sceneunderstanding

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. https://doi.org/10.1016/j.autcon.2010.06.007

Targ, S., Almeida, D., & Enlitic, K. L. (2016). *Workshop track-ICLR 2016 RESNET IN RESNET: GENERALIZING RESIDUAL ARCHITECTURES*.

Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., & Guibas, L. (2019a). KPConv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-October*. https://doi.org/10.1109/ICCV.2019.00651

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019b). *KPConv: Flexible and Deformable Convolution for Point Clouds*. http://arxiv.org/abs/1904.08889

Torrey, L., & Shavlik, J. (2010). *Transfer Learning*.

Truong-Hong, L., & Lindenbergh, R. (2022). Extracting structural components of concrete buildings from laser scanning point clouds from construction sites. *Advanced Engineering Informatics*, *51*. https://doi.org/10.1016/j.aei.2021.101490

Turkan, Y., Bosche, F., Haas, C. T., & Haas, R. (2012a). Automated progress tracking using 4D schedule and 3D sensing technologies. *Automation in Construction*, *22*, 414–421. https://doi.org/10.1016/j.autcon.2011.10.003

Turkan, Y., Bosche, F., Haas, C. T., & Haas, R. (2012b). Automated progress tracking using 4D schedule and 3D sensing technologies. *Automation in Construction*, *22*, 414–421. https://doi.org/10.1016/j.autcon.2011.10.003

U.S. GSA. (2009). GSA,"GSA BIM Guide For 3D Imaging, version 1.0". *U.S. General Services Administration (GSA)*, *3*. http://www.gsa.gov/graphics/pbs/GSA_BIM_Guide_Series_03.pdf

Wang, C., Cho, Y. K., & Kim, C. (2015). Automatic BIM component extraction from point clouds of existing buildings for sustainability applications. *Automation in Construction*, *56*, 1–13. https://doi.org/10.1016/j.autcon.2015.04.001

Wang, Q., Guo, J., & Kim, M. K. (2019). An application oriented scan-to-bim framework. *Remote Sensing*, *11*(3). https://doi.org/10.3390/rs11030365

Weinmann, M., Jutzi, B., Hinz, S., & Mallet, C. (2015). Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*, 286–304. https://doi.org/10.1016/j.isprsjprs.2015.01.016

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). *How transferable are features in deep neural networks?* http://arxiv.org/abs/1411.1792

Zhao, H., Jiang, L., Jia, J., Torr, P., & Koltun, V. (2021). Point Transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16259–16268.

Zhou, Q.-Y., Park, J., & Koltun, V. (2018). *Open3D: A Modern Library for 3D Data Processing*. http://arxiv.org/abs/1801.09847