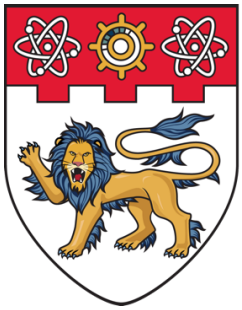

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

AS-BUILT PIPE BUILDING INFORMATION MODELLING (BIM) AUTOMATION

Qualifying Examination Report

Doctor of Philosophy

Submitted by
Xie Yuan(G1702946L)

School of Mechanical and Aerospace Engineering (MAE)

Supervisor: Associate Professor Dr. Cai Yiyu (MAE)

2020

Abstract

Building Information Modelling (BIM) is recognized and promoted as the de facto standard for new construction industry recently due to the great benefits it can bring by bridging communication and technical gaps among different stakeholders throughout the building life cycle. It is imperative to have as-built BIM during the entire life cycle of a building, particularly for the longest stage of the operation and maintenance. Among different components of a building, mechanical, electrical, and plumbing (MEP) system, requires regular maintenance and thus facility management. However, for buildings and facilities that are built before the existence of BIM or even before the era of computer-aided design (CAD), documenting their as-built conditions is still practiced in a very manual and tedious way with substantial time and efforts needed even for a skilled modeler. This research aims to develop a framework automatically reconstructing BIM of MEP system from imaging data in multi-modalities. The proposed method can streamline the as-built MEP BIM modelling work and help to digitize existing MEP facilities in a much efficient and low-cost manner.

Specifically, we limit our targeted objects to the piping components in MEP, which are the most common and dominant types found in MEP. The piping system is divided into three parts to facilitate the reconstruction automation, namely exposed pipeline, hidden pipeline, and pipe connectors.

The exposed cylindrical pipeline reconstruction, as a main part to this research, has been extensively studied and state-of-the-art methods have been carefully reviewed. The exposed MEP objects are captured in point cloud format using 3D imaging technologies such as LiDAR, or laser scanning, and photogrammetry, Inspired by prior works and the emerging hot research topics of deep learning application in 3D data from 2D images, a novel deep learning network called PipeNet and geometry combined method is proposed. The PipeNet is able to predict the cylinder centerline points and radius associatively through the feature mapping, and the post-network geometric processing further summarizes the prediction based on the properties of BIM, and finally outputs the reconstructed parametric pipe models. Preliminary results using both synthesized and real scanned data show a comparable detection rate with good accuracy,

and with less execution time and user inputs compared with state-of-the-art methods.

Ideas on the hidden pipeline and pipe connector reconstruction are discussed to direct future works. As-built pipe connector reconstruction will be formatted as a model retrieval problem using a BIM model library. For the hidden MEP objects, which are usually embedded or hidden in or after partition walls or suspension ceilings, their accurate survey is still a challenging task under wide research, let alone 3D reconstruction. We will capture those hidden MEP objects using non-destructive imaging technologies such as ground penetrating radar (GPR). Our proposed method will take in the generated volumetric data as input and, together with the exposed systems, output the as-built BIM of the complete piping system.

Acknowledgements

I would like to give my sincere thanks to my supervisor, Associate Professor Dr. Cai Yiyu (MAE) for the continuous support, constructive guidance and valuable insights to help me identify the research topic and develop a systematic approach to the problem.

I also want to express my appreciation towards my colleagues in the Surbana Jurong (SJ)-NTU Corp Lab, who have always provided me valuable help and feedback related to my research work. I would like to thank Mr. Liu Tianrui for many insightful discussions on how to approach the problem. I am also thankful to Mr. Srivathsan, who has helped me with the training dataset simulation. I want to thank Mr. Wu Xiangyu, Mr. Huang Lihui, Ms. Joanne for the on-site data collection so that I can test my algorithm. Last but not least, my appreciation goes to Mr. Charles and Mr. Srinivasa, who have supported me with guidance on Building Information Modelling (BIM) and project management.

I am very grateful to all the support from my family and my friends for their constant company in all times.

Xie Yuan

Matriculation No. G1702946L

School of Mechanical and Aerospace Engineering

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Documentation for buildings and facilities is a tedious and laborious task in the past when all components are represented in two-dimensional schematic drawings. Hand-drawing makes it hard for timely and regular updates especially in the construction and maintenance stages. With the development of CAD technology, documentation becomes fast, and revision or updates are easy to share among different stakeholders. CAD further lays the foundation of BIM, which stores not only geometric shapes and dimensions but all other metadata for example material, structural bearing usage, manufacturer and cost etc. BIM brings opportunities to bridge the gaps among stakeholders across disciplines, and to enhance the communication and collaborations throughout the building or facility life cycle including design, construction and maintenance. However, all the advantages do not come free. New equipment, operation routines, skillset trainings, and government regulations need to be adopted in order to properly implement BIM as the new industry standard.

While BIM is gradually and steadily implemented for new buildings or facilities, converting the existing ones to BIM produces challenges. Currently, such work is generally divided into data collection or survey and data processing which are both largely manual in the traditional way: the surveyors visit the site to measure and record the size and location information, and the modelers create the models from scratch using the survey notes and photographs as references. The whole process, which is error prone, typically takes months or even years pending on the complexity of existing building. Now with the emergence of laser-scanning technology, the

survey work can be done much faster and more accurately. Point cloud data is generated from laser scanning which is comprised of usually millions of points each encoded with the relative location to the scanner of the contact point of the laser and the objects and sometimes the object color or laser reflection intensity values. Several BIM software is available in the market that can assist the modelers draw the model by importing the point cloud data and some can even semi-automatically extract primitive shapes such as planes, spheres and cylinders from the point cloud. However, the modelers still need to connect the primitive shapes and assign metadata manually to make it a complete BIM. There is still a lack of a fully automation tool to reconstruct BIM from point cloud.

Building components can be broadly divided into two categories: the structural/architectural (S/A) components and other facilities such as MEP components and furniture. Comparing to S/A components, MEP systems are considered more complex due to their smaller sizes, relatively free-formed shapes and usually cluttered, occluded and/or concealed locations. Though there are numerous methods investigated in this field, fully automated reconstruction of 3D as-built MEP models remains as an open problem.

1.2 Motivation

Given the advantages BIM can bring to the construction industry and built environment and the development of survey technologies, as-built BIM is no doubt the trend that the industry will advance in the years to come. With MEP as the main and important facility system in every building or industrial plants, reconstruction of as-built MEP BIM is unavoidable, and its achievement is certainly beneficial to building management for example facility maintenance and energy consumption control. While a few commercial software is available for semi-automatic pipeline modelling from point cloud, manual inputs such as thresholds setting, connectivity reconstruction and semantic label assignment are required and deemed as the bottleneck of the whole modeling processes [1].

1.3 Problem statement

This research aims to develop a pipeline that reconstructs as-built BIM of MEP components in buildings automatically. There are two sources of inputs to the pipeline: one is exposed MEP

point cloud data scanned using LiDAR, or 3D laser scanner, or generated using photogrammetry, and the other is volumetric data of hidden MEP components obtained using GPR scanner. The pipeline output is a complete BIM model.

It should be noted that this research is part of a scan-to-BIM project aiming to automate the BIM reconstruction of a whole building including S/A and MEP components and furniture. The S/A component reconstruction part will output a semantically labelled point cloud with all non-S/A components, including MEP, labelled as clutter. Considering that the MEP components in buildings, if exposed, are always either near the wall or beneath the ceilings, clutter points at those locations will be cropped out and used as input of the proposed pipeline.

In many cases, MEP components in buildings are concealed behind false ceilings or walls for better appearance and protection. This is also a distinctive feature of MEP in residential buildings compared to those in industrial plants where most components are exposed for easy maintenance. We assume the information of these hidden components are collected using UWB scanners and the two sources of input are registered and with enough data resolution and completeness prior to feeding into the pipeline.

The MEP components considered in this research comprise cylindrical pipes such as sanitary, plumbing, venting pipes, gas and fire protection pipes, rectangular ducts such as ventilation ducts and cable trays, and pipe connectors such as elbows, Tee junctions and so on.

1.4 Objectives

This research tries to solve the problem with the following objectives:

1. Develop a novel exposed pipeline reconstruction algorithm with maximized automation. The algorithm should be intelligent to require no or minimal manual inputs and reconstruct both cylindrical pipes and rectangular ducts. The algorithm should accept raw registered point cloud data that include potential targeted MEP components as input and output the reconstructed piping system in BIM format.

2. Develop a hidden pipeline reconstruction algorithm with maximized automation. The algorithm should take in volumetric data output generated by GPR scanner, comprehend the data, reconstruct MEP objects out of the data and output in BIM format.

3. Develop a library model retrieval algorithm to reconstruct as-built pipe connectors. The

algorithm should extract pipe connectors from the input point cloud and match them to a BIM model library to retrieve the best-match.

1.5 Organization of the report

This chapter introduces the background, motivation, research gaps, and objectives of this research work. The rest of the report is organized as follows:

Chapter 2 presents the literature review conducted in fields relevant to this research work. As there is no literature on reconstructing the general MEP systems into BIM, reconstruction methods of cylindrical pipes, rectangular pipes, and pipe fittings and fixtures are reviewed separately. Point feature estimation which is usually adopted along with existing geometric model reconstruction methods is briefly described. Finally, reconstruction of hidden objects is touched on as research in this area is still at infancy stage.

Chapter 3 details the proposed cylindrical MEP component reconstruction method. With the point set abstraction, the PipeNet is carefully designed and illustrated how the post-network processing can output the complete pipe components. The MEP model datasets simulation and point cloud synthesis prepared for the PipeNet training, as well as the algorithm testing results using both synthesized and real scanned data are also discussed in this Chapter, followed by a thorough analysis.

Chapter 4 presents the ongoing research of the rectangular ducts reconstruction by modifying the method proposed for cylindrical pipes. The modified training dataset simulation and synthesis and the preliminary testing results on scanned point cloud are presented in this Chapter.

Chapter 5 draws a conclusion for the report, highlighting the contributions. Ideas for future work are also introduced.

Chapter 2

Literature review

MEP elements can be broadly divided into cylindrical (pipes, conduits etc.) and rectangular (ventilation ducts, cable trays etc.) shapes and other connectors and fittings with more complicated geometries (elbows, valves, flanges etc.). Literature review was conducted for cylindrical and rectangular shape detection in point cloud and pipe connector model reconstruction.

2.1 Cylindrical pipe reconstruction

2.1.1 Point cloud preprocessing

2.1.1.1 Point feature extraction

Point normal is the most used feature in 3D geometry processing algorithms. The common way of computing point normal is by extracting geometric properties of the query point neighborhood through various methods for example fitting flat plane [2] or curved surface [3], PCA analysis [4–7] and supervised SVM clustering [8]. The traditional methods often face the problem of neighborhood size selection which a critical parameter for normal estimation especially in cluttered scenes with curved surfaces because too large a neighborhood may lead to unreliable local geometric properties while too small a neighborhood may be overwhelmed by noises. Zhao et.al [9] used top-down octree method together with local plane fitting to identify the largest consistent planar neighborhoods about query points and assign the plane normal to all points in the neighborhood. Khaloo and Lattanzi [10] presented a deterministic MM-estimator approach to filter inlier neighbor points followed by the classic PCA to estimate

robust normals.

More recently, with the extension of deep learning from 2D image processing to 3D, deep Neural Network (DNN) methods [11, 12] and methods combining handcraft geometric feature descriptors or prior knowledge and DNN were proposed for normal prediction. Lu et.al [13] proposed a PCA and deep-learning combined approach to estimate normals. The query points together with neighbor points were projected to the plane defined by the two principal eigenvectors to generate a heatmap with prefixed resolution. The heatmaps were learnt through a Convolutional Neural Network (CNN) to classify as feature or non-feature points i.e. edge or non-edge points, followed by another CNN for normal prediction. Lenssen et.al [14] proposed an iterative normal estimation method using Graph NN to refine the weight of neighbor points and iterative re-weighting least square plane fitting was implemented for normal estimation. The author noted that the point cloud noisy level could affect the selection of point neighborhood size. Boulch and Marlet [15] improved the Hough space voting in [16] by replacing the voting step with CNN processing. Three neighbor points were randomly selected and the normal of the plane they defined was computed as a hypothesis and transferred to the Hough accumulator. The filled hemispherical accumulator was projected to the base plane to form a heatmap that was fed into a 2D CNN for final normal selection. A multi-scale variant was also proposed by generating multi-channel Hough heatmap. Zhou et.al [17] proposed a novel multi-scale point descriptor comprising the multi-scale bilateral filtered suboptimal normals and associated height maps and apply CNN to obtain the fine-tuned normals. The reported average angle error ranged from 4.5 degrees to 10.8 degrees and timing ranged from 23s to 312s.

In [18], Zhou et.al presented a scale-selective deep learning network for normal estimation. The network local feature learning was supervised by a local plane classification task where the network branch predicted if the neighbor points lied on the same plane as the input patch center point defined by the center point normal. A multi-scale selection subnetwork was adopted that a scale weight was predicted for each sub-scale patch global features extracted and the normal of the scale with the highest weight was output as final. The difference between our proposed network and this work is that we do not use multiple single-scale networks for the scale feature extraction, instead, we use a single network for all scales based on the hypothesis that the best scale that used for the final prediction should share similar geometric shape and so as the network kernels.

Curvature is the other most used feature in describing point cloud for flat and curved surface

points classification. As curvature is directional in 3D space, the maximum and minimum, or principal values are most useful in cylinder detection-related algorithms because a point on cylindrical surface has its maximum curvature equal to the reciprocal of the cylinder radius and the minimum curvature equal to zero and direction along the center axis. Son et.al [19] estimated the point curvature by fitting a Non-uniform rational basis spline (NURBS) surface patch to the neighborhood and the principal curvatures were calculated from the surface parameters. Seibert et.al [20] transformed the curvature estimation problem to osculating circle fitting on planes containing both the query points and point normal rotating around the normal direction at discrete angles. The circles with maximum and minimum radii were selected for the determination of principal curvatures and directions. This method was applied by Dimitrov et.al [5] in the computation of multi-scale feature descriptor used for robust normal estimation as part of their work of segmenting MEP components from structural/architectural components in building point clouds.

2.1.1.2 Point cloud segmentation

Existing MEP detection or reconstruction methods often include a segmentation step that extracts MEP points out of the scene before the cylinder detection process. While the classification of curved and flat surface points was usually performed using curvature threshold, the segmentation was achieved by clustering points in proximity with similar point features for example normal-based [3, 21–24] and roughness-based [5] region growing algorithms.

Deep learning is also recently applied in point cloud segmentation and scene comprehension, the objective of most of those works is to segment point cloud into different objects for example walls, slabs and furniture through training on similar datasets. Sonntag and Morgenshtern [25] proposed a deep learning and convex-optimization combined method to segment random objects into flat faces. The point cloud was first divided into local patches using k-nearest neighbor (k-NN) and each patch was voxelized. A CNN was then trained to predict probability of two patches being from the same region followed by convex optimization denoising the probability matrix and the final clusters iteratively generated by a rounding step on the probability matrix.

2.1.2 Cylinder detection

The most common methods used in geometric object detection are RANdom SAmple Consensus (RANSAC) [26] and Hough transform (HT) [27]. As cylinder is the most common

geometric shape in MEP and most existing works focused on cylindrical pipe detection, this section reviews these methods and others proposed particularly for cylinder detection from point clouds.

RANSAC is an iterative method that fits parametric models to the minimum required number of data points sampled randomly from the entire data. The number of inlier points that comply with the fitted model within a predefined threshold is recorded as the score of the model. The model with the highest score is selected whenever the maximum number of iterations is reached or when the number of inlier points is satisfied.

Schnabel et.al [28] proposed a method to estimate cylinder shape using only two data points together with their normals. The direction of the cylinder center axis was determined by the cross product of the point normals. Projecting the two points and normals to the plane with normal direction equal to the center axis direction, if the two points were on the surface of a cylinder, the extended lines along the point normal directions and passing through the two points would intersect at the center point of the projected circle of the cylinder. The radius of the cylinder was assigned as the distance between the circle center point and the sample point. This estimated cylinder was infinitely long. To derive the start and end positions of the cylinder, all inlier points were extracted from the scene and projected to the center axis for the bounding box extraction. However, many false positives could be generated if applying RANSAC method directly to the whole scene data samples. Moritani et.al [29] applied this method to extract cylinder segments and pruned false positives using several thresholds including number of inliers, length and radius of the cylinder, and the angular coverage of inliers. Nguyen and Choi [23] applied this method to extract parameters of straight cylinder segments after segmenting the raw scene point cloud into individual components by comparing with the as-designed CAD model.

Besides using RANSAC directly for cylinder detection, combination of RANSAC plane or circle detection ang Gaussian sphere was also often used for cylinder detection [3,30,31] because of the property that normals of points on cylindrical surfaces mapped on Gaussian spheres will distribute in circle pattern. Chaperon et.al [31] proposed to detect cylinders in two steps: first using RANSAC plane detection on the Gaussian sphere to find the cylinder axes directions, and next using RANSAC cylinder detection with the direction determined from the previous step to extract the cylinder radius and positions. Liu et.al [32] proposed to detect global common directions of the pipes in the scene based on the assumption that global orientation similarities

were more stable than the individual instances in the noisy point cloud input. The cylinder axis direction was estimated by applying RANSAC plane detection on the Gaussian sphere based on the similar idea proposed in [28] where the plane normal direction was computed using the cross products of random two point normals. The resulted cylinder direction spherical map was used for global direction estimation by using mean-shift clustering and the cylinder instances with the same orientation were separated in the orthogonal plane projections by circle fitting. In [33] a structure detection and decomposition method was proposed to detect cylinders in large-scale plant point clouds. Based on the assumption that the cylinder directions were either perpendicular or parallel to the ground, the author first extracted the points on cylinders perpendicular to the ground by comparing point normal to the ground normal and projected them to the ground plane. The remaining points were classified as points on cylinders parallel to the ground and the cylinder axes directions were estimated using Gaussian sphere histogram. Points were then projected along the detected directions and RANSAC circle detection was applied on both the ground plane and projected planes.

HT method detects primitive shapes or geometries by parameter voting mechanism. Each data point casts votes in the Hough parameter space for all possible shape of primitive object it may belong to. The shapes with the highest voted parameter sets were selected. It is successfully implemented in 2D image processing for line and circle detection [34]. In practice, the Hough space vote accumulator is discretized in a pre-defined granularity which is the main factor affecting the algorithm's computational space and time complexities. Another factor is the dimensionality of the accumulator, i.e. the number of parameters that represent the target object. Right circular cylinder has five degrees of freedom which makes it prohibitive to directly apply HT for detection. The large size of input point cloud produced by terrestrial laser scanner which often comprising millions of points makes the problem even more demanding. Rabbani et.al [35] proposed a 2-step sequential HT to reduce the Hough space dimensionality by dividing the five parameters into two groups: one group including two parameters for axis orientation, the other including one parameter for radius, and two parameters for cylinder location in the local coordinate system. The first step transformed the Gaussian sphere of the point cloud to a 3D Hough Gaussian sphere to estimate the direction of cylinder axis. Then the second step projects the point cloud onto a 2D plane with a normal direction equal to the cylinder axis direction and to detect circle in the plane using a 2D Hough space with user-specified radii to derive the cylinder radius and position.

Due to the discretization procedure in real implementation, HT method often poses a

dilemma that more cells in the Hough space give more accurate estimates with the sacrifice of computation time and space complexities increase; while fewer cells boost the computation speed and require less memory space, the result accuracy is jeopardized. To solve this problem, a few modifications on the sequential HT were proposed. Su et.al [36] presented a coarse-to-fine approach in the first step by iteratively increasing the number of accumulators at each cell in the Hough space that the great circles passed through after generating the initial hypothesis in a coarsely discretized Hough space. Patil et.al [21] also proposed an adaptive discretization method. A coarse axis direction in a Gaussian sphere with sparse cells was firstly estimated, and then the cells in the neighborhood of the initially most voted cells in the Hough space were resampled to a higher density. Figueiredo et.al [6] proposed an orientation voting Hough accumulator cell sampling method. The cells were randomly generated from Gaussian Mixture Model chosen according to prior knowledge of possible cylinder orientations. The vote was not binary but computed in a continuous form that more weights would be given to cells with directions orthogonal to both the point normal and principal curvature directions and points with larger curvatures. Ahmed et.al [37] assumed that the pipes are built along the orthogonal directions of the main axes of a building and thus applied 2D HT for detection of circles with discrete presumed-radii in the cross sectional slices cut along the three directions at predefined distance intervals. The consecutive circles were then connected, and cylinder axes were fitted among the circles' centers. It however requires the user's input of Hough space granularity and prior knowledge of the radii range of the cylinders in the scene. The author suggested using low resolution Hough space for large pipes and increasing the granularity in additional runs for small pipes.

There are also other methods proposed for cylinder detection besides RANSAC and HT. Son et.al [22] classified pipe points by checking curvature according to an existing piping and instrumentation diagrams (P&IDs), grouped cylinder instances using connectivity- and normal-based region growing method proposed by Rabbani et al [24] and merged close instances to form connected pipelines. The centerlines of the pipeline were estimated at the center of the circles with the smallest radii which correspond to the maximum curvatures at some seed points sampled at predefined intervals along the surface. The centerline nodes were later connected, and the connectivity topology of the centerline nodes were used to classify pipe type (straight, elbow, reducer, and Tee). Lee et.al [38] combined the Voronoi diagram-based method [39] and topological thinning method [40] to estimate the pipe centerline nodes and generate complete skeletons for the pipelines. The pipe radii were computed by the average of median distances

from each skeletal node to k-nearest points on the surface. The elbow and tee connectors were first modelled using the radii and pre-specified sleeve lengths and the straight pipe segments were modelled by connecting the ends of connectors.

Nurunnabi et.al [41] proposed a robust PCA (RPCA) approach to estimate the cylinder axis direction. The RPCA introduced an outlyingness score to filter out outliers before conducting the normal PCA. The radius and positions of the cylinder was derived by applying Least Trimmed Squares (LTS) on the projection of the points along the central axis direction. Jin and Lee [42] proposed to use RANSAC sphere fitting at neighborhoods of sample points to estimate the centerline nodes of the cylinder. The centerline candidates were then refined by local averaging and connected by eigenvector angular similarity check. The centerlines at elbow sections were reconstructed by cubic spline interpolation. This method however required prior knowledge on the cylinder diameter range to adjust the neighborhood size used in the first step. A large neighborhood would increase the computation time while a small neighborhood could not detect pipes with large radii because they would be potentially miss classified as planes.

Tran et.al [43] proposed an iterative cylinder detection method based on hierarchical clustering. An initial cylinder axis direction was estimated at a seed point using eigenvalue analysis and the remaining points were projected along this direction onto a plane where algebraic circle fitting was performed. The cylinder inlier points were updated by filtering the points on the fitted circle using normal and distance conditions. In [44] Araújo and Oliveira presented a connectivity-based method to extract cylinder segments. The input points were first projected onto planes with the normal directions approximately perpendicular to point normals. The connected clusters were then detected in the projections by searching in the 3D k-NN graph. After the clusters were extracted, the cylinder axes directions were refined by applying PCA to the point normal. A novel quadtree-based circle detection algorithm was proposed based on the properties that the reverse extended normals of points on a circle should intersect at the circle center and that the point normal histogram should be uniformly distributed. If projected circles were detected, least-square fitting of cylinders to the clusters was performed to extract the cylinder parameters.

The performance metrics of some current works are summarized in Table Table 2.1, and some major application limitations are recognized as follow:

1. Prior assumptions e.g. pipe directions and prior knowledge e.g. pipe radius ranges and P&IDs are applied. However, in many use cases the prior knowledge is not available, or

the prior assumption may not always hold for example some sewage pipes in buildings are not built to the main axes of the building but typically designed with a mild slope in order to ensure the fluid mobility. The algorithm feasibility is thus jeopardized.

2. Handcrafted point features are used e.g. point normal and curvatures, and therefore preprocessing of the scanned point cloud is required. This process is potential to bring in errors that are propagated through the later processing, and it also increases the computation time as well.

3. User intervention or inputs are required, e.g. the granularity for Hough space, neighborhood size for k-NN or other connectivity graph generation. Different parameter sets can also impact the algorithm accuracy and execution time. These important parameters are usually empirically set, and tuning may be required for different datasets.

2.1.3 Duct reconstruction

Automatic reconstruction of ducts with rectangular cross-sectional shapes from point clouds is not widely studied yet. While there is commercial software [45] that can assist users extract ducts in point clouds, the process is not fully automated and still requires manual interventions. Narumi et.al [46] proposed to reconstruct ducts using rectangle detection method on the point projections on the cross sectional planes perpendicular to the three orthogonal main axes of the building at regular intervals. The reconstructed duct straight segments were further connected by examining the normal of the end faces and the distance between the ends. In order to overcome the incomplete scan problems, the author also extended the duct segments by examining the number of point cloud near the end face vertices. Guo et.al [4] used RANSAC plane detection to reconstruct the cable trays and ventilation ducts as part of their work of geometric quality inspection comparing with the as-designed models.

2.1.4 Connector reconstruction

After the main parts of the pipelines, i.e. the straight cylinder segments are detected and reconstructed, the remaining work is to build the connectivity topology and reconstruct the pipe connectors. While most prior pipeline reconstruction methods did not consider this step but used simple default connector models in place of the as-built models, there are research on model

Table 2.1: Current work performance metrics

Works	Method used	Recall	Precision	F1 score	Mean radius error (mm)	Limitation
Lee et.al [38]	Voronoi diagram-based skeleton estimation	• Straight 0.97 • Elbow 0.94 • Tee 0.92	Not provided	Not available	2.79	Long implementation time.
Son et.al [22]	Curvature-based skeleton estimation	• Straight 0.92 • Elbow 0.97 • Tee 1	Not provided	Not available	3.4	Point curvature and P&ID are required.
Qiu et.al [32]	HT	0.95	0.49	0.65	Not provided	Point normal is required.
Patil et.al [21]	Normal-based region growing and HT	0.6	0.89	0.72	3.81	Point normal is required.
Kawashima et.al [3]	Normal-based region growing and RANSAC circle fitting	• Straight 0.86 • Elbow 0.88 • Tee 0.71	• Straight 0.58 • Elbow 0.59 • Tee 0.58	• Straight 0.69 • Elbow 0.71 • Tee 0.64	Not provided	Point normal is required.
Liu et.al [33]	RANSAC circle fitting	0.62	0.55	0.58	Not provided	No connectivity reconstruction. Assumption of pipe direction is applied.
Araújo and Oliveira [44]	k-NN clustering, PCA on point normals, and circle fitting	0.52	0.68	0.57	Not provided	No connectivity reconstruction. Point normal is required.
Jin and Lee [42]	Sphere RANSAC-based skeleton estimation	0.78	0.86	0.82	Not provided	Range of radius is required to set global k-NN parameter.

retrieval and reconstruction using exiting model catalogues or libraries. Son e.al [47] proposed to reconstruct the industrial instrumentation objects by exploiting information from the existing P&IDs. The topologic information containing the adjacency relationships between instrumentations and pipelines and the as-designed geometric information containing radii of pipelines were derived from the existing P&IDs. A graph representation of the topologic relationships among pieces of objects was generated and used to reconstruct the graph representation of the scene point cloud.

Other methods worked by comparing carefully crafted features of the model libraries or as-designed model and those of point cloud data. Czerniawski et.al [48] proposed a bag-of-features approach to recognize pipe spools in point cloud for progress tracking and quality check. The principal curvatures of all points in the scene were computed and used as feature descriptors to compare with those of surface points sampled on the as-designed objects. The most similar locations were remained as hypothesis space candidates and for each candidate the bivariant histograms of the curvature descriptors was compared with that of the as-designed model to find the most similar candidate. In [49] Sharif et.al proposed to generate an offline model library using 3D hash table. The feature set stored in the table was composed of the angles formed by surface normals of two randomly sampled point with a fixed distance d and the line segment connecting them. The online matching process was done in a RANSAC-based search manner. Random points were uniformly sampled in the scene and all points in distance d from them were selected to form potential point pairs whose feature sets were computed and compared with those stored in the hash table. Transformation matrices of the target model were computed based on the pairs of points matched and the model was registered to the scene for point compliance check. Pang et.al [50, 51] proposed to recognize the pipe fittings using exhaustive window-search method. The training samples were first voxelized and converted to 3D summed area table from where the 3D Haar features were computed, and a detector was trained using Adaboost training method which computed the weighted sum of all local 3D Harr features. However, to detect varying sized objects, multiple times of search were required and since the 3D Haar feature was not orientation-invariant, object rotation to principal direction using PCA was required each time before the library matching evaluation.

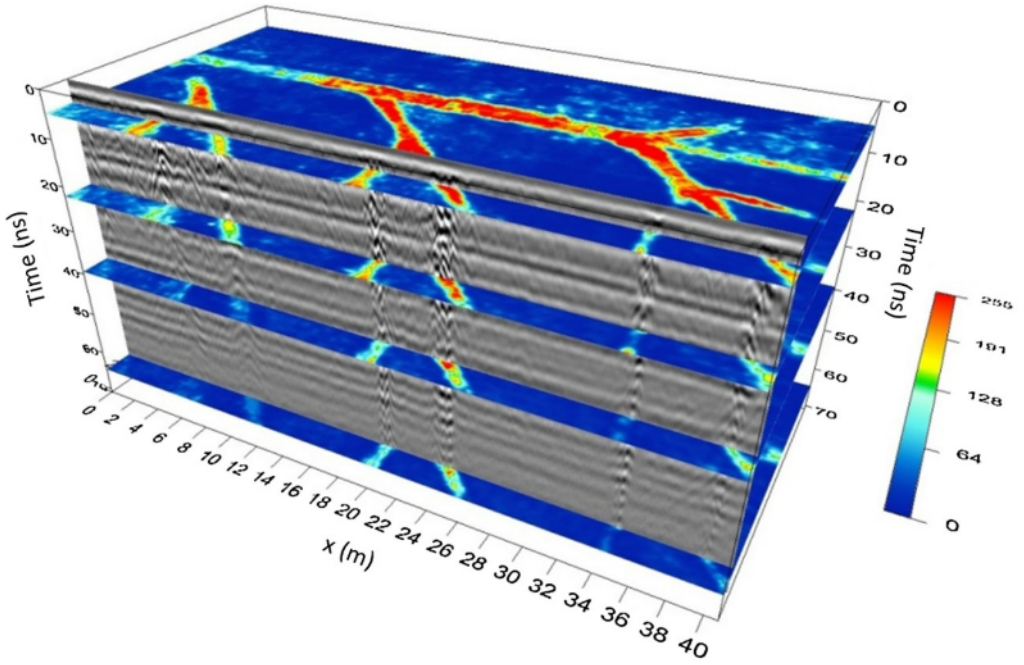

Figure 2.1: Example radargram output from GPR scanner

2.1.5 Hidden MEP object reconstruction

Most MEP components in commercial buildings are designed to be embedded inside walls or hidden after partition wall or suspension ceiling. The laser scanners or other imaging technologies are not capable of detecting them. Nevertheless, Ground Penetrating Radar (GPR) technology has been extensively used for buried-object detection and structure health diagnosis [52,53]. The integration of GPR with other positioning devices, such as a Global Positioning System (GPS) and an Inertial Measurement Unit (IMU), can significantly improve the detection accuracy. And the combination of ultra-wideband (UWB) with GPR can further boost the detection capability [54]. The output from GPR scanners is in the form of image, or radargram, stacks with each pixel represents the signal strength received by the antenna and the depth axis represents the receiving time of the signal. An example output data is shown in Figure 2.1 [52]. Comprehension of these data requires expert knowledge as the data can vary with different wall materials, internal structures, the hidden object layouts and even the air moisture rate. Hopefully, these factors can be explicitly learnt leveraging on the strong generalization capabilities of DNN and the objects reconstruction can thus be carried out thereafter.

Chapter 3

Cylindrical pipe reconstruction

In this Chapter the proposed cylindrical pipeline reconstruction method is presented, which combines the robustness of traditional geometric processing and adaptation capability to noises of deep learning and minimizes the dependence on prior knowledge about objects in the scene from users.

3.1 Overview

The proposed cylindrical pipe reconstruction method is inspired by the skeleton-based methods [22,38] and recent deep learning network research [11,55]. We believe that the skeleton-based cylinder reconstruction is a rather intuitive method considering how the pipes are designed by engineers and how they are presented in both 2D and 3D drawings. A complete pipe is intrinsically constructed by a centerline together with radii parameters. Therefore, we propose a two-step cylindrical pipe reconstruction method via collaboration of a deep network called PipeNet and post-network geometric processing. First, we propose a multi-scale pipe centerline prediction deep network PipeNet that consumes the raw point cloud and for each point predicts its corresponding point on the centerline and associated radius. Then the ultimate centerlines are fitted and refined among the predictions. The final complete pipes are reconstructed by merging centerlines and the connectors are reconstructed by extracting the point cloud of connection points and matching with the connector database.

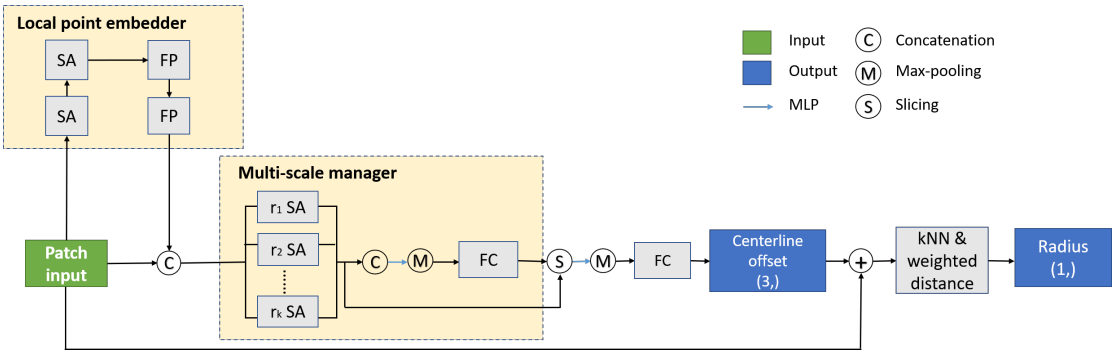

Figure 3.1: PipeNet architecture

3.2 PipeNet architecture

The PipeNet is composed of two modules namely local point embedder and multi-scale manager. Essentially the associated cylinder parameters for each point concern the local neighborhood features only, thus the input to the PipeNet is chosen as the local patch point cloud centered at the query point and the output is the translation vector from the query point to the predicted centerline point and the radius of its associated cylinder. The overall architecture is shown in Figure 3.1.

3.2.1 Point set convolution

Convolution is commonly adopted in deep learning for image processing. This operation effectively abstracts local neighborhood features through shared kernel weights which perfectly fits parallel procesing. By stacking of convolution layers, the input data is abstracted into fewer points with each point embeded with more information of larger receptive field. These features can be used for classification or segmentation followed by de-convolution or feature propogation layers. The pre-requisite of the traditional convolution is that the input data is ordered or structured in space, such as images which are basically groups of fixed number of pixels arranged in fixed order. The convolution kernel shape can then be easily defined according to the input data structure so that the kernel weights are applied to the unique values of the corresponding pixels. However, this is not the case for point cloud which consists of millions of unstructed points with varying density and spacing across the captured 3D space. Though it is possible to convert point cloud into voxels before further processing, the compression inevitably will result in information loss, the extent of which depends on the voxel size and the conversion

operation used, let along the size of the resulted volume data can go up to millions voxels easily and the processing space and time go up exponentially. Moreover, for targeted MEP objects whose sizes can vary drastically, the decision of voxel size itself is not trivial without prior knowledge of the scene. Nevertheless, the point set convolution can be defined as [56]:

$$f(x_0, x_1, ..., x_n) = S(Gh(x_0, x_1), ..., h(x_0, x_n)) \tag{3.1}$$

where $x_i = (p_i, f_i)$ is neighbor point about x_0 with location p_i and supplementary features f_i for example the color. $f : R^N \rightarrow R$ is the abstracted feature, $h : R^N \rightarrow R^K$ is the point-wise kernel function, for example the point location offset Δp_i , feature difference Δf_i and/or embedded with another feature mapping function. $G : R^K \rightarrow R^M$ is the aggregation function that generalize the set features, and $S : \{R^M, R^M, ...\} \rightarrow R$ is a symmetric function such as max, summation, or average pooling that ensures the feature learned is invariant to the point order. Because point cloud does not possess a fixed structure, stride operation as done in the image processing can not be directly imported. Alternatively, sampling operation such as furthest point sampling, grid downsampling, random sampling and inverse density sampling is often adopted as the first operation of each convolution layer.

3.2.2 Local point embedder

The patch input is fed to the local point embedder first. The design motivation is that a key difference of the application case of the developed network to the testing datasets used by current point local feature estimation works [11–13,17] is that the input scenes of MEP components can be more complex than single objects datasets for example PartNet and PCPNet dataset, and the indoor scenes with only S/A components and furnitures for example S3DIS, ShapeNet and NYU depth dataset, in a way that the sizes of targeted MEP objects can vary drastically and the objects can be very close to each other or even intertwined. As a result, it is inevitable that the patch input will include neighbor points from different objects than what the query point belongs to, especially when the prior knowledge of the dimension range of targeted objects is not available and the input patch radius must be set to cater to the largest object in the scene. And different to traditional cylinder detection methods which take in handcrafted features such as point normal or curvature, we leverage deep learning to extract the local context feature of each point. The local point embedder is thus designed to provide the subsequent layers

more point-wise local neighborhood context. The best-fit sub-patches for smaller objects will be learned and selected by the multi-scale manager discussed in the next section. While our network is not restricted to any point feature learning network, we use PointNet++ [57] as the backbone in this module due to its simplicity and demonstrated success on various tasks such as normal estimation [12], object detection [55] and large-scene segmentation [58]. Two set abstraction (SA) layers and two feature propogation (FP) layers are stacked to embed each neighbor point in the patch of its own neighborhood features. Within each SA layer, input to the point-wise kernel function is the location offset from each neighbor point to the center query point concatenated with the feature embedding from last layer. FP layer propogates the feature embedding of the sampled points back to the unsampled patch by using the inverse distance weighted average based on k nearest neighbors. The final propogated point local feature is concatenated with the location and pass to the next module.

3.2.3 Multi-scale manager

Multi-scale manager module takes in the original patch input with the point local feature embeded and outputs a sub-patch point set that can describe the shape of the cylinder the query point belongs to. Sub-patches of different radii are sampled using furthest point sampling and each fed to a set abstraction layer that encode the sub-patch global features. After which, a MLP classifier is trained to select the best sub-scale.

The selected sub-patch is later abstracted again using PointNet layer followed the centerline point offset and radius regression. Specifically, the regression is realized with a MLP network with fully connected layers, ReLU, batch normalization and drop out layers. The MLP takes the selected scale global feature $f_i \in R_C$ and outputs the Euclidian space offset $\Delta P_i \in R^3$ such that the predicted cylinder centerline point has $C_i = x_i + \Delta P_i$. The radius r_i is computed as the inversed distance weighted average of the distance from C_i to x_i and its k-NN points to mitigate noise effects. Though the multi-scale manager concept is used in prior works namely NestiNet [11] and the multi-scale normal estimation network (msNE) in [18], our method is different and is specifically designed to the cylinder detection application. There are mainly two key differences. Firstly, both NestiNet and msNE adopt parallel structure of the normal regression and the scale selection module. NestiNet used a mixture-of-export (MoE) architecture, where several normal regression exports are trained each for a different scale or a combination of scales and the scale manager takes in all the sub-patch raw features and output a vector of scores of

each exports. The msNE runs multiple single-scale normal estimation networks in parallel and each generates a global feature which is both used to output the normal prediction and taken as input to the scale selection module. Because only the normal estimation of the highest scored sub-scale is output as final, the parallel structure of the regression and scale selection processes essentially result in unnecessary computation and memory space. Comparing to them, our network selects the best scale first followed by regression using the selected sub-scale data only. Secondly, instead of training a separate export or sub-network for each scale, we use a *shared* PointNet layer for all scales. This design is based on the idea that the finally selected patch should exhibit a similar pattern – a sub-patch extracted purely from the hosting cylinder of the query point. And the weights sharing also help increase the network computation efficiency further. Note that another difference to msNE of our approach is that after the best scale is selected we specifically train another set abstraction layer for the output regression instead of using the same global feature for both the scale selection and output regression.

We train the network to minimize the Mean Squared Error loss of L2 distance between the estimated centerline points and radii and the ground truth:

$$L = \frac{1}{N}\sum_{i=1}^{N}(\|c_i - c_{i_{GT}}\| + \|r_i - r_{i_{GT}}\|) \tag{3.2}$$

3.3 Post-network processing

We further extract centerlines from the PipeNet output centerline point sets C: (ci, ri) through line candidate generation, refinement and intersecting steps. We use straight line representation instead of spline or Bezier based on the observation that in BIM pipe routes are always designed and represented by straight lines and connectors, and that the straight segments usually dominate a complete pipe. It is worth mentioning that unlike most current cylinder detection methods, our method does not require the entire input data to be processed. During implementation we randomly sample a subset input points to reduce the execution time.

3.3.1 Line candidate generation

The prediction set C is first segmented by principal eigenvector-based region growing method followed by RANSAC line fitting. Although RANSAC line fitting can be directly applied to the whole prediction set to generate the line candidates, we observe that more noise candidates

would be produced in that way which results in more workload for the next refinement step, and applying RANSAC to the whole scene also takes more computation time. The criteria used for region growing method include both the spatial connectivity and eigenvector similarity. For each point, an initial normalized principal eigenvector e corresponding to the largest eigenvalue is computed using PCA in $N(c_i, r_i)$ which is a set of neighboring points contained in the sphere of the radius r_i, First, a seed point $c_s \in C$ having the greatest number of neighbors is chosen and its corresponding e_s is set at the principal eigenvector of this region e_r. Then, the set of points $\{c_i | c_i \in C, c_i \in N(c_s, r_i), \langle e_i, e_r \rangle < \tau\}$ are added to the region, where $\langle \cdot \rangle$ means the angle formed by the two vectors. The region principal eigenvector is updated using all added points. Each of the added points is then chosen as new seed points, and the other points satisfying the same condition are iteratively added to the region. The above steps are iterated until any point satisfying the condition resides in the neighborhood of the seed points. Then, for each segmented region, a line segment candidate is generated using RANSAC line fitting. For simplicity, we refer the produced line segment as line in the following, which is defined by the normalized direction d and endpoints $\{E_1, E_2\}$ and is supported by an inlier point set P_{in} and an averaged $R_i = \frac{1}{N(P_{in})} \sum_{i=1}^{N(P_{in})} r_i$, where $N(P_{in})$ is the number of points in P_{in}.

3.3.2 Line candidate refinement

Due to the presence of noise, a few undesired lines may be produced, and due to the varying density of the prediction set C, a ground truth line may be divided into a few short segments. We observed that the undesired lines usually have poor point support or short length. We refine the initial line candidates L using a region-growing-alike line refinement algorithm. Specifically, we select the longest line candidate as the seed line $l_s \in L$. The set of lines $\{l_i | l_i \in L, \langle d_i, d_s \rangle < \alpha, D(l_i, l_s) < \beta, \frac{|R_i - R_s|}{R_s} < \gamma\}$ are added to the group, where $D(\cdot)$ is the distance function between the two lines and $|R_i - R_s|$ is the absolute value of the radius difference. The added lines are chosen as new seeds and the step is iterated until all lines satisfying the conditions are added to the group.

The distance function $D(l_i, l_s)$ is computed as follow and is illustrated in Figure 3.2:

1. If the two lines overlap, i.e. the projection of any endpoint of a line to the other line falls within the line segment, the line distance is computed as the minimum distance from the

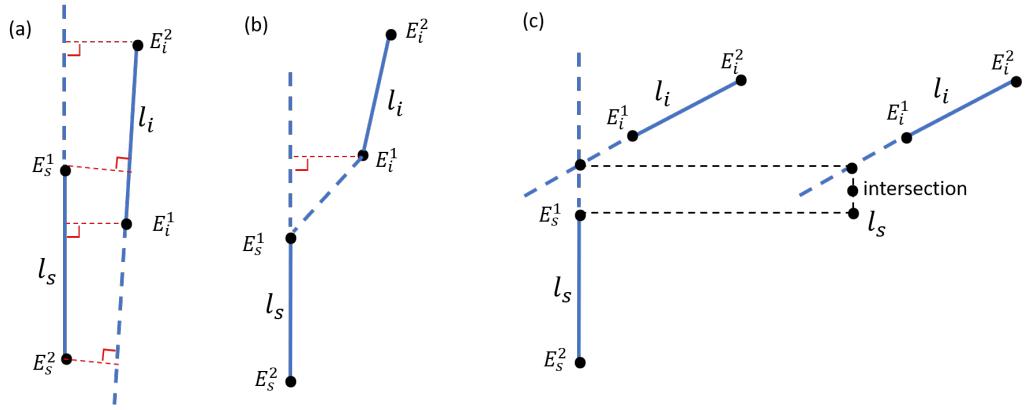

Figure 3.2: Line distance for (a) overlapping line pair, (b) non-overlapping line pair. (c) Skew line intersection side view (left) and top view (right).

endpoints to the other line:

$$D(l_i, l_s) = \min\{d(E_k, l_j) \mid E_k \in \{E_i, E_s\}, k \neq j\} \tag{3.3}$$

where $d(p, l)$ is the vertical distance from point p to line l.

2. If the two lines do not overlap, the nearest endpoint pair is first selected, and the line distance matrix is computed by two parts. The first part is the distance from the nearest endpoint on the testing line to the seed line, i.e. the vertical distance of the nearest endpoint to the seed line. The second is the projected distance between the two nearest endpoints on the seed line.

$$D(l_i, l_s) = \underset{\|E_i^j E_s^k\|}{\arg\min}\{d(E_i^j, l_s)\,,\ \|E_i^j E_s^k\| \cdot \cos\langle \widehat{E_i^j E_s^k}, d_s\rangle \mid E_i^j \in \{E_i\}, E_s^k \in \{E_s\}\} \tag{3.4}$$

All the distance thresholds are defined in terms of R_s, i.e. $\alpha, \beta, \gamma = \mu R_s, \mu \in R$, which is to help resolve the issue of pipe radius variation in the scene and to enhance the adaptability of the algorithm in scenes of pipes with a large radius diversity. A refined line is fitted to all the inlier points of the line members in each group. The refined line set is further filtered by line length and inlier density defined as number of inlier points per unit length.

3.3.3 Line intersecting and connector extraction

Because only straight pipe segments are extracted in the above steps, their connection relationships are recovered in this section. As the generated lines of two connected pipe segments are not guaranteed to be co-planar, their intersection is computed in two steps. First both lines are extended and the nearest points on one line to the other line are found. The middle of the two nearest points is defined as the skew line intersection. The grouping approach is similar to that in the line refinement, in a region-growing-alike manner with modified growing criteria $\{l_i | l_i \in L, D'(l_i, l_s) < \beta\}$. The distance function $D'(\cdot)$ is changed to the maximum distance from the intersection to the nearest endpoints for end-joint lines, i.e. connected by elbows or reducers, and the minimum distance from the intersection to the nearest endpoints for middle-joint lines, i.e. connected by tee connectors. After the lines are grouped into complete pipes, their directions are fine-tuned to ensure pairwise co-planarity which is a requirement for BIM.

Since the pipe connector locations are determined, their corresponding raw point cloud data will be extracted with cropping box centered at the intersection points and with sleeve direction and length catered to the connector type. The extracted connector point clouds will be later matched with the BIM model library and the retrieved BIM models will be assembled accordingly to form the complete reconstructed pipes.

3.4 Datasets

3.4.1 Simulated training datasets

The network training dataset is synthesized in-house since open-source datasets comprising MEP category is currently not available. Since the training dataset required by the network includes both point cloud and ground truth i.e. the centerlines and radii parameters, manual labelling real scanned data is apparently not practical. In this case, the simulation of point cloud data has proven to be the ideal workaround.

The dataset simulation is consisted of two steps: MEP model simulation and scan point cloud data synthesis. Both steps are implemented using Blender software. The MEP models are simulated using a modified pipe-nightmare add-on [59]. Rounded pipe elbows with different curvatures and multiple pipe in parallel arrangement are added in order to emulate the real MEP systems, and export of centerline is also added for ground truth generation. The radii

Figure 3.3: Point cloud synthesis workflow

range is decided as 2cm to 22cm according to the MEP specifications in public residential flats in Singapore, where 81% population resides [60], regulated by Housing and Development Board (HDB). For simplicity, we refer these flats as HDB flat in the following. It is worth mentioning that the simulated MEP models now include cylindrical pipes only without any pipe fittings or fixtures due to the limitation of the add-on.

Point cloud data synthesis for the models is implemented using BlenSor add-on [61]. Figure 3.3 shows the synthesis workflow. After the groups of models are generated, scanner locations are added as key frames. For each model group, nine scanner locations are set at one side 1.5m from the model to emulate the real scan process. Then the individual scans at each ordered location is collected by automating the scans for all model groups. Since scanner locations are known, the scan registration is done by combining the individual scans of the same model group, and then the registered scans are down sampled using 3mm voxel grid. The ground truth is generated by sampling points along the centerlines at 2mm interval and assigning the nearest centerline point together with the associated radius parameter to each scan point. Some examples of point cloud and corresponding ground truth are shown in Figure 3.4.

3.4.2 Scanned test datasets

Real scan data is collected in HDB flats using Faro laser scanner and in SJ-NTU Corp lab using Riegl laser scanner, shown in Figure 3.5. Because by the time of scanning the Corp lab was not renovated, a rather complete MEP system was exposed resulting in a good quantity of testing data. On the other hand, MEP components inside the HDB flat were less exposed so only pipes in the kitchen and in the risers outside the flat were identified and tested. The MEP point clouds are manually segmented out from the scenes for module testing purpose. Since ground truth is not available and manual assigning ground truth is not feasible, visual inspection of testing results was conducted.

Drone is also adopted for data collection at restricted area of HDB ground level, as shown in Figure 3.6. Because HDB ground level is often open for public and most MEP components

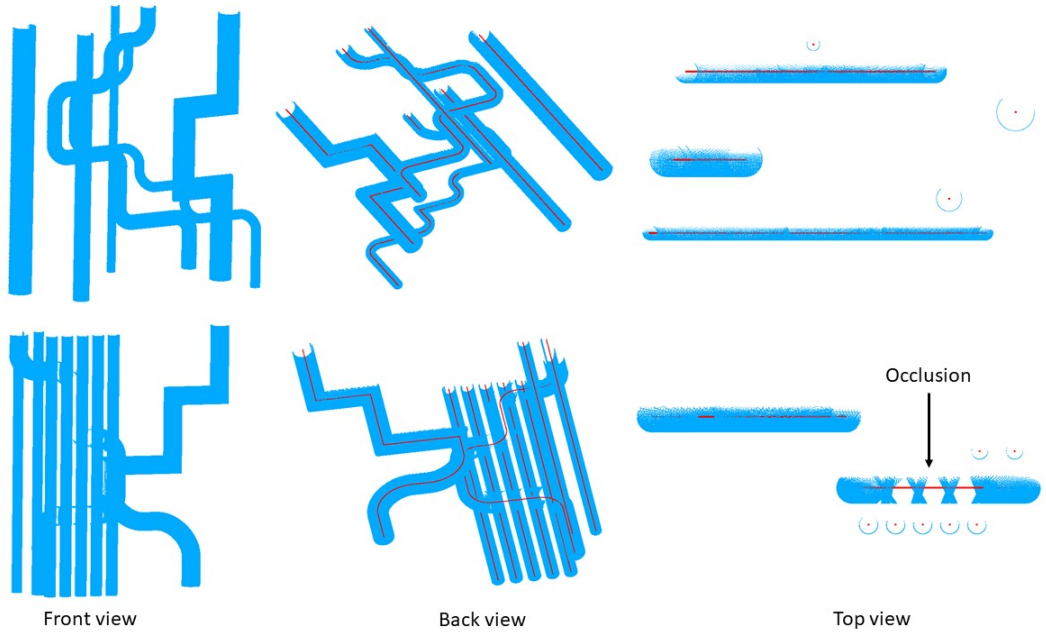

Figure 3.4: Example of MEP point cloud synthesis. Synthesized point cloud (blue), and centerline ground truth (red).

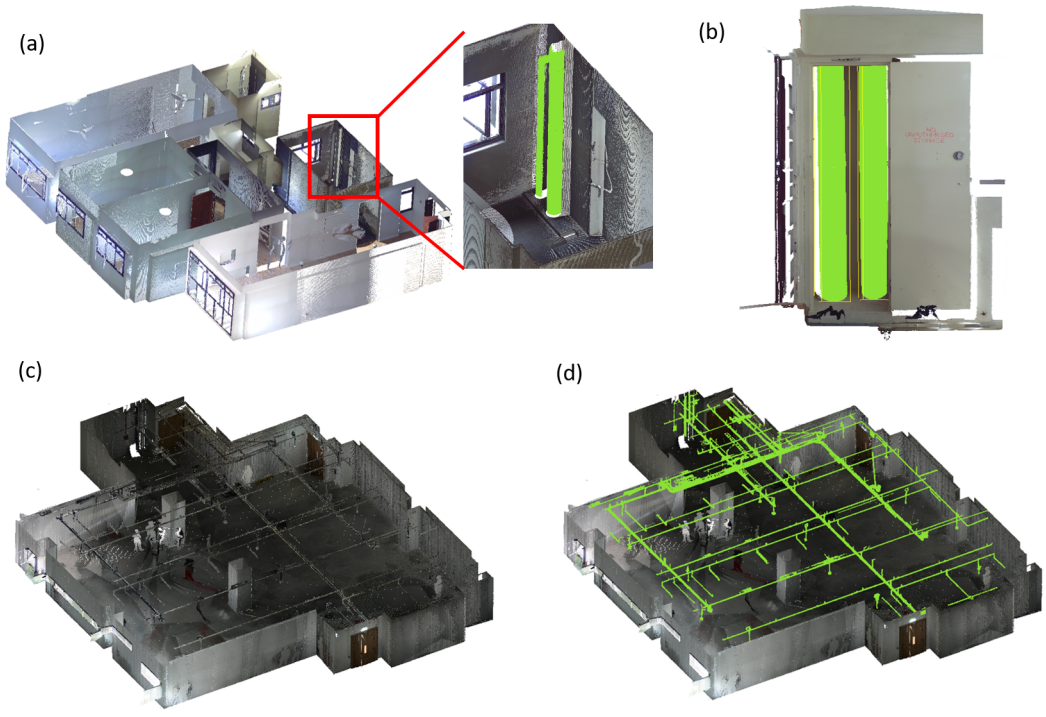

Figure 3.5: Point cloud scanned (a) inside HDB flat, (b) in riser outside HDB flat and (c-d) in SJ-NTU Corp Lab with ceiling point cloud removed. Targeted MEP components are highlighted in green.

Figure 3.6: Drone scanned MEP point cloud

there are exposed, we used Dji Mavic Air, a drone with camera, to fly around targeted pipe components and at the same time took videos of them. Images extracted from the videos were used to reconstructed mesh models using Autodesk Recap and point clouds were sampled on the mesh models. Note that the bumpy edge surfaces of the reconstructed mesh models are resulted by occlusion and lack of features due to the same white color of the pipes and the background walls or ceilings.

3.5 Experiment

3.5.1 Training details

The PipeNet is trained using the simulated dataset. The SA module in the local point embedder has radius (0.02, 0.04, 0.05) meter and (0.04,0.08) meter, and number of sample points (16, 32, 64) and (32,64) respectively. In the multi-scale manager module, the sub-scales are set with sphere radii (0.03, 0.05, 0.08, 0.12, 0.15) meter and 256 sample points. All sample points are gathered using farthest point sampling. 160 model groups are simulated, and the

model is trained following the fifth fold validation method, i.e. 32 groups are used only for testing. Each model group is randomly sampled 2048 patches at each epoch. Each patch has 1024 neighbor points within a sphere of radius 0.15 meter. For neighborhoods with more than 1024 points, we perform random sampling, and for those with fewer points we duplicate a random subset points to meet the same number. During training we augment the input patch on-the-fly by randomly rotating the object and jitter the position of each points by a random Gaussian noise with zero mean and (0.001,0.002,0.003,0.004) standard deviation and clipped at two standard deviation. The model is trained for 60 epochs with batch size 64 using Adam optimizer. The initial learning rate is 0.001 and is decayed at rate 0.7 for every 3 epochs, capped at 10^{-6}.

3.5.2 Quantitative result

Each test model group is randomly sampled 10240 points for the PipeNet testing and we record the mean Euclidean distance between the prediction and the ground truth. In order to examine the noise robustness, we augment the test dataset with noise ratio σ 0.067% - 0.2% based on the size of the model bounding box. The prediction mean error (mErr) and mean standard deviation (mStd) are summarized in Table 3.1. The unit of distance error is millimeter. Figure 3.7 visualizes some predictions compared to the ground truth. While in Table 3.1 the errors grow slowly with the noise increases, it can be seen from Figure 3.7 that the predicted centerline points distribute evenly around the ground truth. We further continue the reconstruction with the post-network processing and evaluate the radius error and centerline angle error between the reconstructed pipes and the ground truth using $\sigma = 0$ and 0.167%. The results are tabulated in Table 3.3. The unit of angle error is degree. It shows that after the post-network processing, our radius estimation error on noise datasets is comparable with the current works and the centerline angle error is fairly robust to noises.

We also use the synthetic data to test robustness to different coverage angles from 45° to 180°, shown in Table 3.2. The unit of angle error is degree. Although a higher angle coverage produces more accurate radius prediction as expected, the centreline angle errors remain low for all angle coverage cases and our post-network processing is shown to be effective in reducing the reconstruction error.

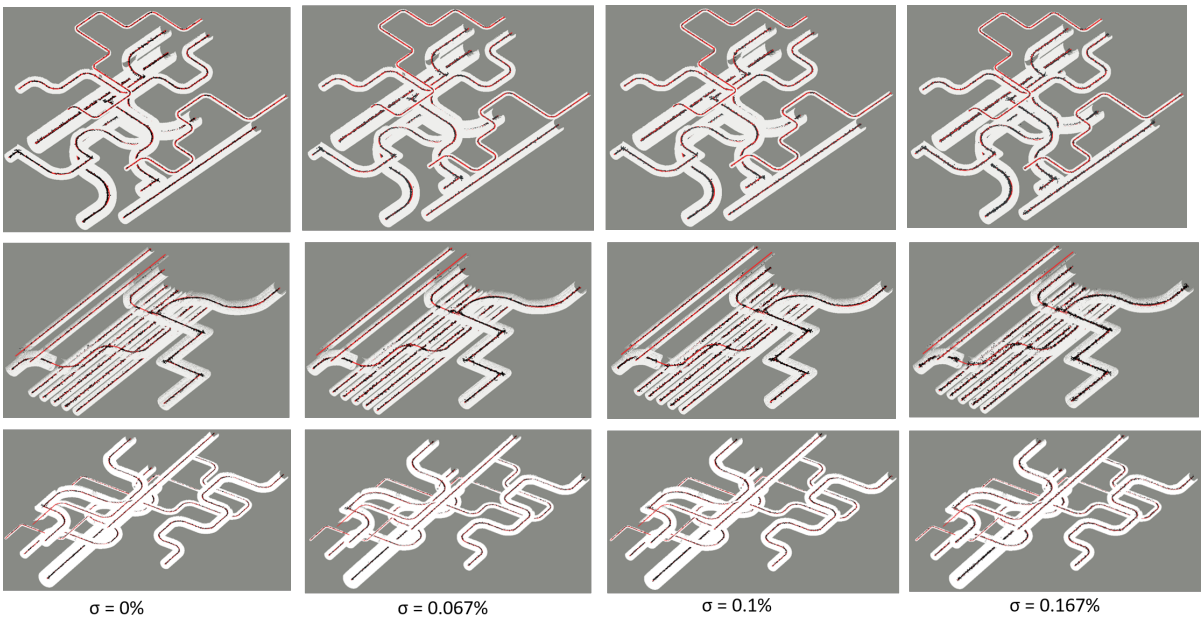

Figure 3.7: PipeNet centerline point prediction results. Pipe model (white), centreline points prediction (black), and centreline ground truth (red).

Table 3.1: PipeNet results on simulated dataset varying noise ratio

σ	**Centerline**		**Radius**	
	mErr	mStd	mErr	mStd
0	7.00	12.19	3.77	6.37
0.067%	7.71	12.75	3.90	6.46
0.100%	8.61	13.56	4.143	6.735
0.133%	9.77	14.581	4.597	7.226
0.167%	11.64	16.674	5.70	8.61
0.200%	15.70	22.58	8.88	12.57

Table 3.2: PipeNet and reconstruction results varying surface coverage angle

Coverage angle	**Centerline**			**Radius**		
	mErr	mStd	Angle mErr after recon	mErr	mStd	mErr after recon
45	16.02	16.77	0.01	13.82	14.82	13.60
60	10.46	11.60	0.02	7.51	8.95	6.39
90	7.65	9.09	0.02	5.08	6.52	3.98
120	6.44	7.69	0.01	4.34	5.93	2.89
180	5.89	7.12	0.01	3.50	4.83	1.51
> 180	5.35	6.25	0.01	3.33	4.56	0.49

Table 3.3: Reconstruction error varying noise ratio

σ	**Centerline**	**Radius**	
	Angle mErr	mErr	mStd
0	1.00	1.46	0.70
0.167%	1.01	3.45	2.36

Figure 3.8: Reconstruction of drone scanned data

3.5.3 Qualitative result

Qualitative test results on real scan data are shown in Figure 3.8 and 3.9. In scan of SJ-NTU Corp lab, out of 92 pipe segments, 88 are recognized and reconstructed, giving a recall rate 95.7% which outperforms the state-of-the-art. Even in high noisy area as shown in Figure 10 (c), our method can still recognize the pipes with good accuracy. However, 55 false positives are recognized, resulting a precision of 61.5%. The main source of false positives come from the supporting cables of the pipes, electricity wires and ducts as shown in Figure 10 (d). Our F1 score is thus 0.75, 0.07 lower than [42]. Nevertheless, our precision is expected to be improved with the generalized PipeNet, which will predict the semantic classes of the MEP components, i.e. pipes and ducts. It can also help to prune points of cables and electricity wires based on the predicted confidence of the points belonging to the pipe class.

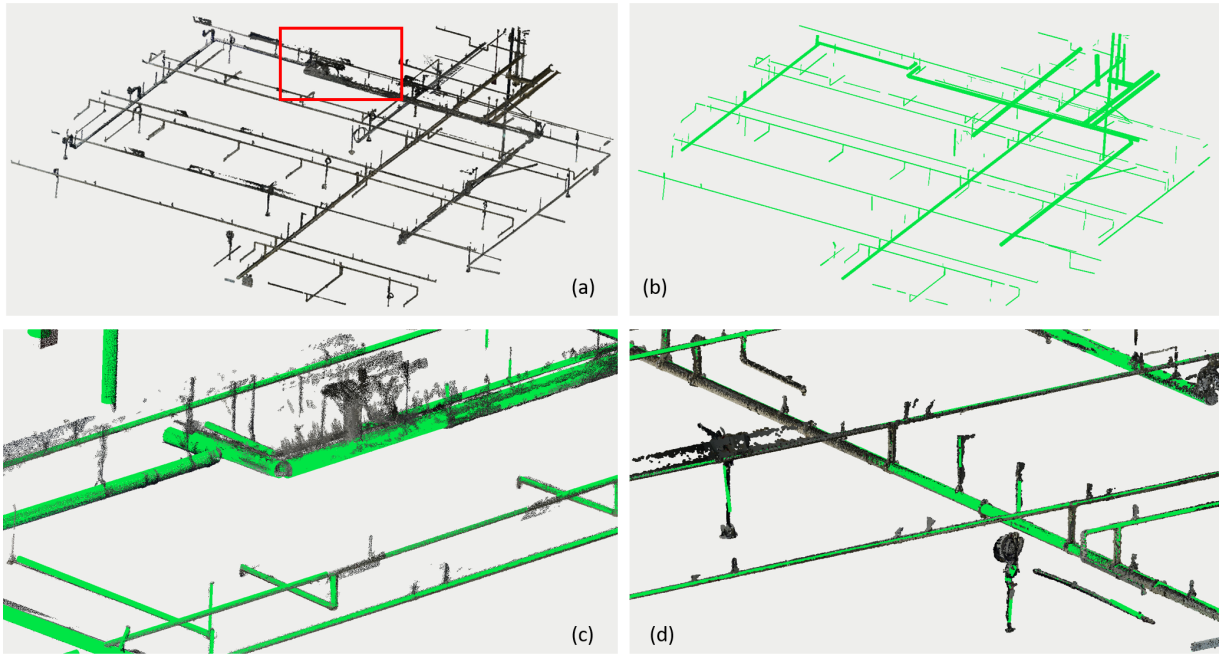

Figure 3.9: Reconstruction of laser scanned data. (a) scan point cloud, (b) pipe reconstruction, (c) close view of the noisy area highlighted in red box in (a), and (d) false positives at pipe supporting cables and electricity wires.

Chapter 4

Generalized pipe reconstruction

4.1 Modification details for generalization

MEP components include not only cylindrical pipes but also ducts in either complete or incomplete rectangular cross-sectional shapes for example the ventilation ducts and cable trays. For simplicity, we refer pipes in cylindrical shape as pipe and rectangular shape as duct in the following. We observe that ducts are designed in a similar way in BIM software as pipes. As such, we expect ducts to be reconstructed in a similar way as the proposed PipeNet and geometric processing combined method except that ducts are defined by more parameters, i.e. the centerline points, widths, and heights.

The MEP model simulation algorithm is modified to generate both pipe and duct models, as shown in Figure 4.1. The point cloud synthesis and ground truth generation steps remain the same, except that the ground truth now include one more semantic label column to differentiate the points between the two shape categories. A semantic classifier branch with the same structure as the centerline offset regression layers is added to the PipeNet after the multi-scale manager, as shown in Figure 4.2. The semantic classifier branch is supervised by the classical cross entropy loss, and the loss function for the regression branch is modified as:

$$L = \frac{1}{N(p \in pipe)} \sum_{i=1}^{N} (\|c_i - c_{i_{GT}}\| + \|r_i - r_{i_{GT}}\|) \cdot s(p_i \in pipe), \tag{4.1}$$

where $s(p_i \in pipe)$ is whether the label of the input point is pipe, N is the total number of query points and $N(p \in pipe)$ is the number of points belonging to pipes.

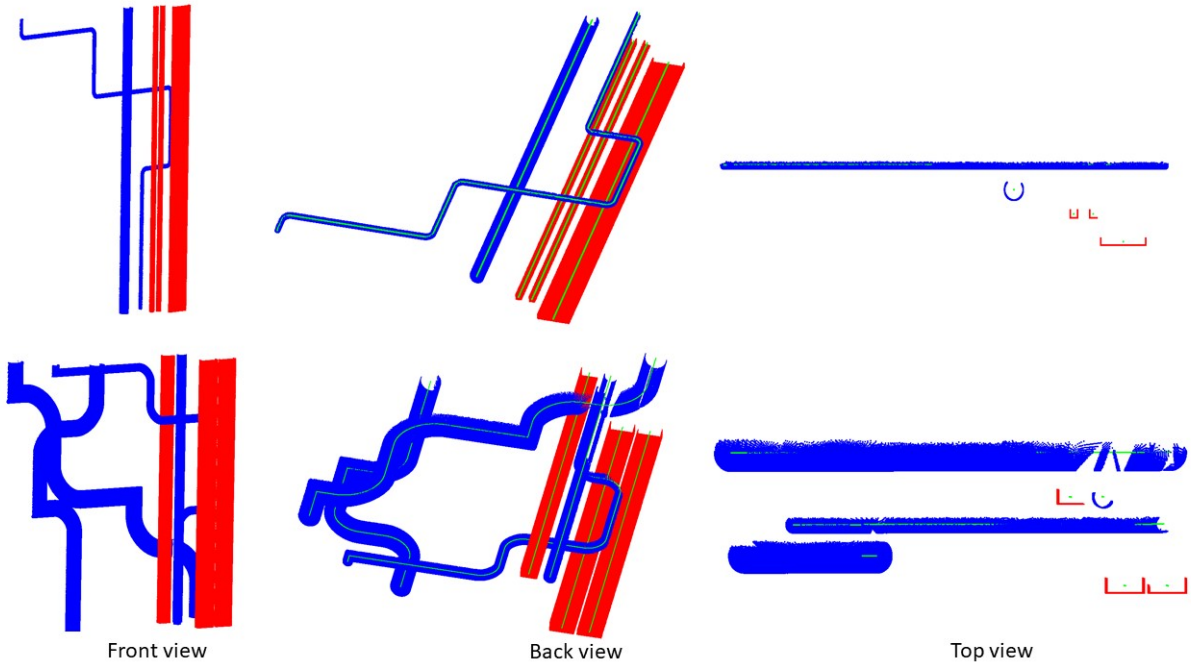

Figure 4.1: Simulated generalized pipe model point cloud. Cylindrical pipes (blue) , rectanglur ducts (red), and centerline ground truth (green).

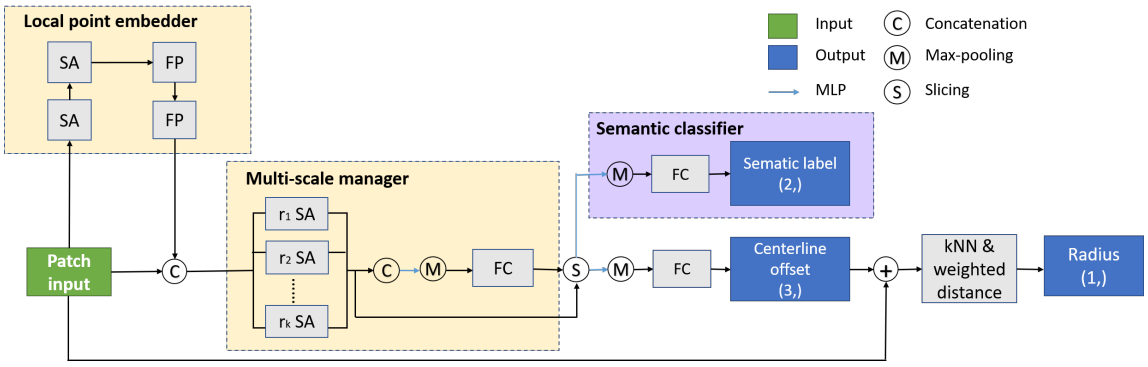

Figure 4.2: PipeNet with semantic classifier branch

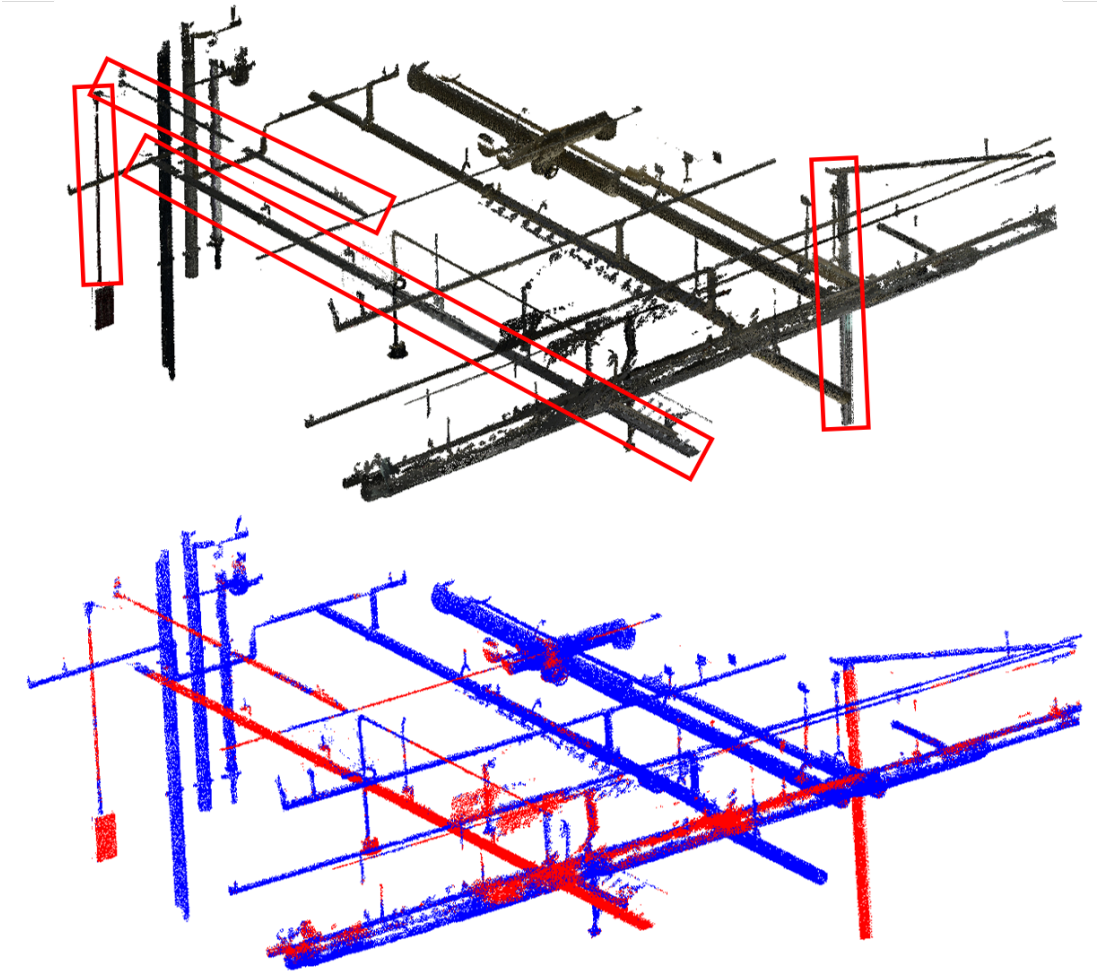

Figure 4.3: PipeNet with semantic classifier branch

4.2 Preliminary results

The network structure is the same as used for the cylindrical pipes and the training details also remain. 90 model groups are simulated and 72 are used for training following the fifth fold validation method. We tested the classification task on the SJ-NTU Corp lab partial scan data where both pipes and cable trays exist. As shown in Figure 4.3, all four ducts are classified correctly, while some noises on a pipe are wrong classified as ducts. The preliminary result shows that the PipeNet has the potential to be expanded to deal with both classes as expected.

Chapter 5

Conclusion and Future Work

5.1 Contribution

The main contribution of this work is the proposal of automatic as-built MEP BIM reconstruction, specifically:

1. We proposed a novel deep-learning and geometric processing combined cylindrical pipe reconstruction method. It is intelligent with maximized automation to consume the raw point cloud directly requiring no pre-processing such as handcrafted point feature calculation and scene segmentation, and no prior assumption or knowledge such as pipe direction, approximate radius range, etc. The test results are shown to be comparable to the state-of-art.

2. The reconstruction method can be expanded to the generalized pipes in both cylindrical and rectangular cross-sectional shapes so that both classes can be reconstructed simultaneously. The preliminary results are shown promising to the generalization.

3. The method is also expected to be expanded to the hidden pipe reconstruction, and together with the BIM library matching of the as-built pipe connectors, the complete piping systems can be reconstructed automatically and accurately.

5.2 Limitations and future works

Certain aspects of proposed method can be improved. In this section we discuss about these potential improvement and initial thoughts on the future works.

5.2.1 Potential improvement on PipeNet

The set abstraction layer and the scale global feature encoder structure can be further explored. The currently used PointNet layer focuses more on the whole sample point set abstraction by examining the relationship between each neighbor point and the center query point, however richer information may be learnt through the relationships among the neighbor points. To achieve that, a graph network structure may be adopted as in [62, 63].

We note that the training accuracy of radius is always better than that of centerline points. To improve the centerline accuracy, the regression task could be decomposed into two tasks of reversed (inwards) normal direction and radius regression as essentially the centerline offset equals the reversed normal vector multiplies the radius. The eased normal direction regression task may help better supervise the selected scale feature embedding.

We also find the network memory size is large resulting a small batch size processed during implementation. The balance between network performance and memory occupation needs to be tuned to reduce the algorithm execution time.

5.2.2 Generalized pipe reconstruction

We expect to streamline the reconstruction of pipes and ducts, so the generalized PipeNet need to be further modified. As ducts have more parameters, a separate regression branch or sub-network specific for them should be added. The semantic classifier can be modeled first followed by the separate regression modules for the two types like in [64], as shown in Figure 5.1. The post-network processing for pipes is expected to be applicable for ducts because of their similar design principle.

The synthetic MEP models can be further augmented with some negative objects, i.e. neither pipes or ducts, for example pipe fixtures, so that the semantic prediction can be used to prune non pipe points and reduce the reconstruction false positive rates. We observe that generally the top surfaces of ducts are absent from scanned point cloud data because ducts are usually

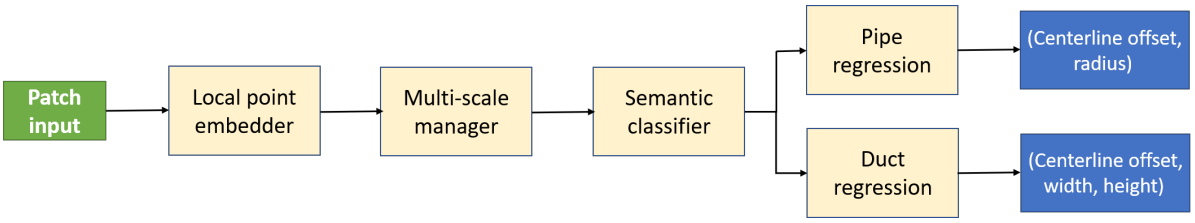

Figure 5.1: PipeNet with semantic classifier branch

installed overhead either below ceilings or hidden above false ceilings. Though data of the other three surfaces are enough to reconstruct the component, it will pose difficulties of classifying the component in BIM types, for example between cable trays and ventilation ducts. Nevertheless, the sizes of these two types of components may vary, the specific BIM types are still possible to be retrieved by matching with the BIM library.

5.2.3 Connector model retrieval with BIM library

Reconstruction of as-built pipe connectors and fixtures is to be achieved by retrieving the best-match model from a BIM library with the extracted point cloud data of targeted objects. As the designed models and input data are in different modalities, a multi-modal embedding space can be established to conduct the matching. The multi-modal embedding requires a trained encoder and point cloud data as input.

Firstly, point cloud of the BIM models can be synthesized using the same virtual scanner as we used for MEP point cloud synthesis. However, because BIM models are parametric that allows them to adapt to the different dimensions and to other constraints while keeping their properties [65], an exhaustive BIM model library is not practical. For example, the radius of a pipe elbow model can be changed freely in BIM software so that the model instances can be used to connect pipes with different radii. We will build the BIM library with each model in several commonly used dimensions according to the regulations of HDB and catalogues from MEP parts suppliers. The models with different dimensions will be trained in group in the encoder for a joint embedding to achieve dimension invariance. The final dimension used for the as-built models will be determined according to the connected pipe dimension.

We observe that due to occlusion, most pipe fittings are scanned on maximally half of the surface during implementation and the incompleteness often takes place in a symmetric fashion. This may pose problems to the matching accuracy. Two ways can be experimented

to tackle this issue. The first is to augment the sampled BIM model point cloud with partial incompleteness during training, while the other is to complete the scanned point cloud by using an encoder-decoder module.

The encoder will take in the synthesized model point cloud and output feature vector in fixed length and stored in the BIM library offline. During online implementation, the extracted scan point cloud is fed to the same encoder and the generated feature vector is used to match in the library. The best match model will be selected, oriented according to the scan data, and assembled to the connected pipes. The other BIM properties associated with the model such as materials will be assigned to the connected pipes to complete the reconstructed BIM.

5.2.4 Hidden MEP reconstruction

Proceq GP8000 [66] is a portable UWB scanner designed to detect buried objects inside concrete and diagnose concrete structure health conditions, as shown in Figure 5.2. We will use this scanner to scan at potential locations of the hidden MEP components for example the exposed pipes and facilities connected with pipe outlets such as faucets and toilets. Due to the nature of the scanner output data, which is a volumetric data assembled by radargrams, a structured point cloud data can be generated by proper threshold filtering on the received signal strength and time-slicing on the depth axis. The proposed PipeNet can be further modified and trained on simulated radar scans and applied to detect the pipe components based on the same regression concept. Finally, the hidden and the exposed pipes will be connected to generate the complete as-built MEP BIM.

5.3 Future Work Timeline

A tentative timeline for development and implementation of the future works proposed in the previous section is detailed in Figure 5.3.

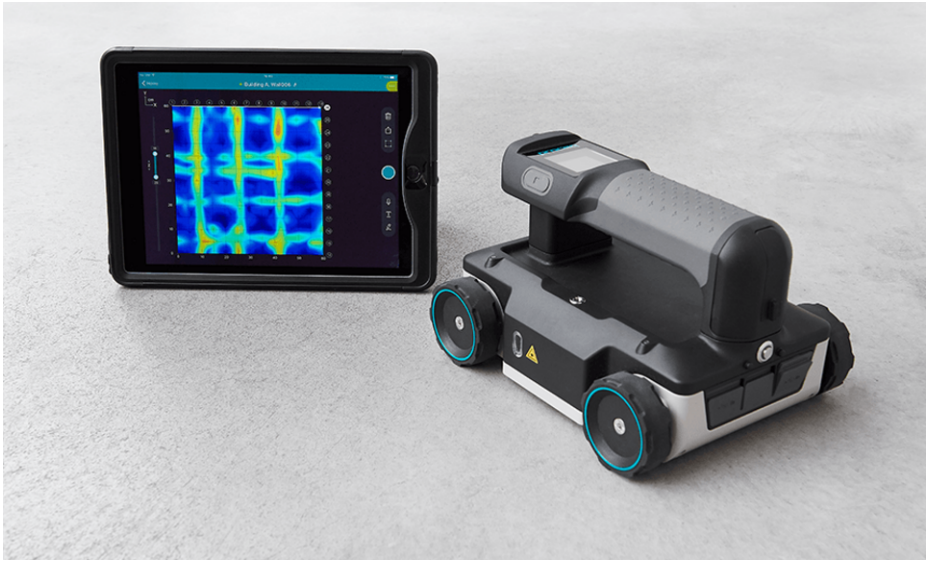

Figure 5.2: PipeNet with semantic classifier branch

Figure 5.3: Proposed timeline for development and implementation of future works.

References

[1] E. Agapaki and I. Brilakis, "State-of-Practice on As-Is Modelling of Industrial Facilities," in *Adv. Comput. Strateg. Eng.*, I. F. C. Smith and B. Domer, Eds. Cham: Springer Verlag, 2018, pp. 103–124. [Online]. Available: http://link.springer.com/10.1007/978-3-319-91635-4_6

[2] M. Nahangi, T. Czerniawski, C. T. Haas, and S. Walbridge, "Pipe radius estimation using Kinect range cameras," *Autom. Constr.*, vol. 99, pp. 197–205, mar 2019.

[3] K. Kawashima, S. Kanai, and H. Date, "As-built modeling of piping system from terrestrial laser-scanned point clouds using normal-based region growing," *J. Comput. Des. Eng.*, vol. 1, no. 1, pp. 13–26, 2014.

[4] J. Guo, Q. Wang, and J. H. Park, "Geometric quality inspection of prefabricated MEP modules with 3D laser scanning," *Autom. Constr.*, vol. 111, mar 2020.

[5] A. Dimitrov and M. Golparvar-Fard, "Segmentation of building point cloud models including detailed architectural/structural features and MEP systems," *Autom. Constr.*, vol. 51, no. C, pp. 32–45, mar 2015.

[6] R. Figueiredo, P. Moreno, and A. Bernardino, "Robust cylinder detection and pose estimation using 3D point cloud information," *2017 IEEE Int. Conf. Auton. Robot Syst. Compet. ICARSC 2017*, pp. 234–239, 2017.

[7] J. Zhang, J. Cao, X. Liu, H. Chen, B. Li, and L. Liu, "Multi-Normal Estimation via Pair Consistency Voting," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 4, pp. 1693–1706, apr 2019.

[8] J. Huang and S. You, "Segmentation and matching: Towards a robust object detection system," in *2014 IEEE Winter Conf. Appl. Comput. Vision, WACV 2014*. IEEE Computer Society, 2014, pp. 325–332.

[9] R. Zhao, M. Pang, C. Liu, and Y. Zhang, "Robust normal estimation for 3D LiDAR point clouds in urban environments," *Sensors (Switzerland)*, vol. 19, no. 5, mar 2019.

[10] A. Khaloo and D. Lattanzi, "Robust normal estimation and region growing segmentation of infrastructure 3D point cloud models," *Adv. Eng. Informatics*, 2017.

[11] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "Nesti-Net: Normal Estimation for Unstructured 3D Point Clouds using Convolutional Neural Networks," in *CVPR*, 2019.

[12] P. Guerrero, Y. Kleiman, M. Ovsjanikov, and N. J. Mitra, "PCPNet Learning Local Shape Properties from Raw Point Clouds," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 75–85, may 2018. [Online]. Available: http://doi.wiley.com/10.1111/cgf.13343

[13] D. Lu, X. Lu, Y. Sun, and J. Wang, "Deep feature-preserving normal estimation for point cloud filtering," *Comput. Des.*, vol. 125, p. 102860, aug 2020. [Online]. Available: https://doi.org/10.1016/j.cad.2020.102860 https://linkinghub.elsevier.com/retrieve/pii/S0010448520300531

[14] J. E. Lenssen, C. Osendorfer, and J. Masci, "Deep Iterative Surface Normal Estimation," in *CVPR*, apr 2020. [Online]. Available: http://arxiv.org/abs/1904.07172

[15] A. Boulch and R. Marlet, "Deep Learning for Robust Normal Estimation in Unstructured Point Clouds," *Comput. Graph. Forum*, vol. 35, no. 5, pp. 281–290, aug 2016. [Online]. Available: http://doi.wiley.com/10.1111/cgf.12983

[16] A. Boulch and R. Marlet, "Fast and robust normal estimation for point clouds with sharp features," *Eurographics Symp. Geom. Process.*, 2012.

[17] H. Zhou, H. Chen, Y. Feng, Q. Wang, J. Qin, H. Xie, F. L. Wang, M. Wei, and J. Wang, "Geometry and Learning Co-supported Normal Estimation for Unstructured Point Cloud," in *CVPR*, 2020.

[18] J. Zhou, H. Huang, B. Liu, and X. Liu, "Normal Estimation for 3D Point Clouds via Local Plane Constraint and Multi-scale Selection," *CAD Comput. Aided Des.*, vol. 129, p. 102916, oct 2019. [Online]. Available: http://arxiv.org/abs/1910.08537

[19] H. Son, C. Kim, and C. Kim, "Fully Automated As-Built 3D Pipeline Extraction Method from Laser-Scanned Data Based on Curvature Computation," *J. Comput. Civ. Eng.*, vol. 29, no. 4, p. B4014003, jul 2015.

[20] H. Seibert, D. Hildenbrand, M. Becker, and A. Kuijper, "Estimation of curvatures in point sets based on geometric algebra," in *VISAPP 2010 - Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2010.

[21] A. K. Patil, P. Holi, S. K. Lee, and Y. H. Chai, "An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds," *Autom. Constr.*, vol. 75, pp. 65–78, 2017.

[22] H. Son and C. Kim, "Automatic segmentation and 3D modeling of pipelines into constituent parts from laser-scan data of the built environment," *Autom. Constr.*, vol. 68, pp. 203–211, 2016.

[23] C. H. P. Nguyen and Y. Choi, "Parametric comparing for local inspection of industrial plants by using as-built model acquired from laser scan data," *Comput. Aided. Des. Appl.*, vol. 15, no. 2, pp. 238–246, mar 2018.

[24] T. Rabbani, F. van den Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraints," *Int. Soc. Photogramm. Remote Sens.*, pp. 248–253, 2006. [Online]. Available: https://research.utwente.nl/en/publications/segmentation-of-point-clouds-using-smoothness-constraints

[25] M. Sonntag and V. I. Morgenshtern, "Region segmentation via deep learning and convex optimization," in *CVPR*, nov 2019. [Online]. Available: http://arxiv.org/abs/1911.12870

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, 1981.

[27] P. V. C. Hough, "Method and means for recognizing complex patterns," *US Pat. 3,069,654*, 1962.

[28] Ruwen Schnabel, Roland Wahl, and Reinhard Klein, "Efficient RANSAC for Point-Cloud Shape Detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[29] R. Moritani, S. Kanai, H. Date, M. Watanabe, T. Nakano, and Y. Yamauchi, "Cylinder-based efficient and robust registration and model fitting of laser-scanned point clouds for as-built modeling of piping systems," *Comput. Aided. Des. Appl.*, vol. 16, no. 3, pp. 396–412, 2019.

[30] K. Kawashima, S. Kanai, and H. Date, "Automatic recognition of piping system from laser scanned point clouds using normal-based region growing," in *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, 2013.

[31] T. Chaperon and F. Goulette, "Extracting cylinders in full 3D data using a random sampling method and the Gaussian image," in *VMV 2001*, 2001.

[32] R. Qiu, Q.-y. Zhou, and U. Neumann, "Pipe-Run Extraction and Reconstruction," *ECCV*, no. Springer International Publishing Switzerland, pp. 17–30, 2014.

[33] Y. J. J. Liu, J.-b. B. Zhang, J.-c. C. Hou, J.-c. C. Ren, and W.-q. Q. Tang, "Cylinder Detection in Large-Scale Point Cloud of Pipeline Plant," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 10, pp. 1700–1707, 2013.

[34] J. Illingworth and J. Kittler, "The Adaptive Hough Transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 690–698, 1987.

[35] T. Rabbani and F. V. D. Heuvel, "Efficient Hough transform for automatic detection of cylinder in point clouds," *ISPRS WG III/3, III/4, V/5 Work. "Laser scanning 2005"*, pp. 60–65, 2005.

[36] Y. T. Su and J. Bethel, "Detection and robust estimation of cylinder features in point clouds," in *Am. Soc. Photogramm. Remote Sens. Annu. Conf.*, 2010.

[37] M. F. Ahmed, C. T. Haas, and R. Haas, "Automatic detection of cylindrical objects in built facilities," *J. Comput. Civ. Eng.*, 2014.

[38] J. Lee, H. Son, C. Kim, and C. Kim, "Skeleton-based 3D reconstruction of as-built pipelines from laser-scan data," *Autom. Constr.*, vol. 35, pp. 199–207, 2013.

[39] T. K. Dey and W. Zho, "Approximate medial axis as a Voronoi subcomplex," in *CAD Comput. Aided Des.*, vol. 36, no. 2. Elsevier, feb 2004, pp. 195–202.

[40] J. Cao, A. Tagliasacchiy, M. Olsony, H. Zhangy, and Z. Su, "Point cloud skeletons via Laplacian-based contraction," in *SMI 2010 - Int. Conf. Shape Model. Appl. Proc.*, 2010, pp. 187–197.

[41] A. Nurunnabi, Y. Sadahiro, and R. Lindenbergh, "Robust cylinder fitting in three-dimensional point cloud data," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 42, no. 1W1, pp. 63–70, 2017.

[42] Y.-H. Jin and W.-H. Lee, "Fast Cylinder Shape Matching Using Random Sample Consensus in Large Scale Point Cloud," *Appl. Sci.*, vol. 9, no. 5, p. 974, mar 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/5/974

[43] T. T. Tran, V. T. Cao, and D. Laurendeau, "Extraction of cylinders and estimation of their parameters from point clouds," *Comput. Graph.*, vol. 46, pp. 345–357, feb 2015.

[44] A. M. Araújo and M. M. Oliveira, "Connectivity-based cylinder detection in unorganized point clouds," *Pattern Recognit.*, vol. 100, p. 107161, apr 2020.

[45] "EdgeWise for MEP Modeling – ClearEdge3D." [Online]. Available: https://new.clearedge3d.com/edgewise/mep-modeling/

[46] M. Narumi, S. Kanai, H. Date, E. Wakisaka, S. Sakamoto, and A. Prof, "Laser-scanned as-built 3D modeling of air-conditioning ducts based on Manhattan world assumption," in *15th Int. Conf. Comput. Civ. Build. Eng.*, 2016.

[47] H. Son, C. C. C. C. Kim, and C. C. C. C. Kim, "3D reconstruction of as-built industrial instrumentation models from laser-scan data and a 3D CAD database based on prior knowledge," *Autom. Constr.*, vol. 49, pp. 193–200, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.autcon.2014.08.007

[48] T. Czerniawski, M. Nahangi, C. Haas, and S. Walbridge, "Pipe spool recognition in cluttered point clouds using a curvature-based shape descriptor," *Autom. Constr.*, vol. 71, no. Part 2, pp. 346–358, nov 2016.

[49] M.-M. Sharif, M. Nahangi, C. Haas, and J. West, "Automated Model-Based Finding of 3D Objects in Cluttered Construction Point Cloud Models," *Comput. Civ. Infrastruct. Eng.*, vol. 32, no. 11, pp. 893–908, nov 2017. [Online]. Available: http://doi.wiley.com/10.1111/mice.12306

[50] G. Pang and U. Neumann, "Training-based object recognition in cluttered 3D point clouds," in *Proc. - 2013 Int. Conf. 3D Vision, 3DV 2013*, 2013, pp. 87–94.

[51] G. Pang, R. Qiu, J. Huang, S. You, and U. Neumann, "Automatic 3D industrial point cloud modeling and recognition," in *Proc. 14th IAPR Int. Conf. Mach. Vis. Appl. MVA 2015*. Institute of Electrical and Electronics Engineers Inc., jul 2015, pp. 22–25.

[52] X. Núñez-Nieto, M. Solla, A. Novo, and H. Lorenzo, "Three-dimensional ground-penetrating radar methodologies for the characterization and volumetric reconstruction of underground tunneling," *Constr. Build. Mater.*, vol. 71, pp. 551–560, nov 2014. [Online]. Available: https://www-sciencedirect-com.ezlibproxy1.ntu.edu.sg/science/article/pii/S0950061814009957

[53] F. M. Fernandes and J. C. Pais, "Laboratory observation of cracks in road pavements with GPR," *Constr. Build. Mater.*, vol. 154, pp. 1130–1138, nov 2017. [Online]. Available: https://www-sciencedirect-com.ezlibproxy1.ntu.edu.sg/science/article/pii/S0950061817316148

[54] J. Ali, N. Abdullah, M. Yusof, E. Mohd, and S. Mohd, "Ultra-Wideband Antenna Design for GPR Applications: A Review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 7, 2017. [Online]. Available: http://thesai.org/Publications/ViewPaper?Volume=8&Issue=7&Code=ijacsa&SerialNo=53

[55] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep Hough Voting for 3D Object Detection in Point Clouds," 2019. [Online]. Available: http://arxiv.org/abs/1904.09664

[56] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *arXiv preprint arXiv:1612.00593*, 2016.

[57] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017.

[58] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively Segmenting Instances and Semantics in Point Clouds," in *CVPR*, 2019. [Online]. Available: https://github.com/WXinlong/ASIS.

[59] "Pipe Nightmare [0.3.33] - Coding / Released Scripts and Themes - Blender Artists Community." [Online]. Available: https://blenderartists.org/t/pipe-nightmare-0-3-33/682448

[60] "Singapore HDB Statistics, facts & figures - The world of Teoalida." [Online]. Available: https://www.teoalida.com/singapore/hdbstatistics/

[61] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "BlenSor: Blender sensor simulation toolbox," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6939 LNCS, no. PART 2, 2011, pp. 199–208.

[62] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing," in *CVPR*, 2019. [Online]. Available: https://github.com/hszhao/PointWeb

[63] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph Attention Convolution for Point Cloud Semantic Segmentation," in *CVPR*, 2019.

[64] L. Li, M. Sung, A. Dubrovina, L. Yi, and L. Guibas, "Supervised Fitting of Geometric Primitives to 3D Point Clouds," in *CVPR*, 2019.

[65] H. Son and C. Kim, "Semantic as-built 3D modeling of structural elements of buildings based on local concavity and convexity," *Adv. Eng. Informatics*, vol. 34, no. August, pp. 114–124, 2017. [Online]. Available: https://doi.org/10.1016/j.aei.2017.10.001

[66] "Portable Ground Penetrating Radar — Proceq GP8000." [Online]. Available: https://www.proceq.com/product/proceq-gp8000/