CLOI-NET: CLASS SEGMENTATION OF INDUSTRIAL FACILITIES' POINT CLOUD DATASETS

Eva Agapaki[a,*], Ioannis Brilakis[a]

[a]*Department of Engineering, University of Cambridge, CB2 1PZ, U.K.*

Abstract

Shape segmentation from point cloud data is a core step of the digital twinning process for industrial facilities. However, it is also a very labor intensive step, which counteracts the perceived value of the resulting model. The state-of-the-art method for automating cylinder detection can detect cylinders with 62% precision and 70% recall, while other shapes must then be segmented manually and shape segmentation is not achieved. This performance is promising, but it is far from drastically eliminating the manual labor cost. We argue that the use of class segmentation deep learning algorithms has the theoretical potential to perform better in terms of per point accuracy and less manual segmentation time needed. However, such algorithms could not be used so far due to the lack of a pre-trained dataset of laser scanned industrial shapes as well as the lack of appropriate geometric features in order to learn these shapes. In this paper, we tackle both problems in three steps. First, we parse the industrial point cloud through a novel class segmentation solution (CLOI-NET) that consists of an optimized PointNET++ based deep learning network and post-processing algorithms that enforce stronger contextual relationships per point. We then allow the user to choose the optimal manual annotation of a test facility by means of active learning to further improve the results. We achieve the first step by clustering points in meaningful spatial 3D windows based on their location. Then, we apply a class segmentation deep network, and output a probability distribution of all label categories per point and improve the predicted labels by enforcing post-processing rules. We finally optimize the results by finding the optimal amount of data to be used for training experiments. We validate our method on the largest richly annotated dataset of the most important to model industrial shapes (*CLOI*) and yield 82% average accuracy per point, 95.6% average AUC among all

*Corresponding author
Email address: `ea437@cam.ac.uk` (Eva Agapaki)

classes and estimated 70% labor hour savings in class segmentation. This proves that it is the first to automatically segment industrial point cloud shapes with no prior knowledge at commercially viable performance and is the foundation for efficient industrial shape modeling in cluttered point clouds.

Keywords: class segmentation, industrial facilities, point cloud processing, CLOI

1. INTRODUCTION

This paper focuses on class segmentation of the most important industrial shapes from point cloud data generated by Terrestrial Laser Scanners (TLS). We choose the most labor intensive industrial object shapes (classes) to model as defined in our previous work (Agapaki et al., 2018). These are, in descending order of labor intensiveness: electrical conduits, straight pipes, circular hollow sections (CHSs), elbows, channels, solid bars, I-beams, angles, flanges and valves. We introduce a new point cloud dataset called *CLOI* that consists of those shapes. The abbreviation *CLOI* is defined by the initials of the geometric shapes of the most important industrial classes, namely C-shapes, L-shapes, O-shapes and I-shapes, and their combinations. We focus on all potential types of manufacturing/industrial facilities as defined by the North American Industry Classification System (NAICS) (United States Census Bureau, 2012) on the condition that the *CLOI* classes are present. We define class segmentation as a partitioning of the TLS point cloud dataset to clusters of points with class labels assigned per point. This is different from detection that refers to object localization by determining the orientation and location of an object without necessarily associating class labels to points. The challenge that our research addresses is how to efficiently minimize the cost and manual labor of automatically generating object oriented Industrial geometric Digital Twins (IgDTs), such that their benefits outweigh the initial investment made to generate these models. This challenge is of utmost importance due to the potential value IgDTs are expected to bring to the industrial sector in terms of preventive maintenance and unplanned shutdowns.

Improper maintenance of aging industrial facilities is a growing concern for the manufacturing industry given its significant and potentially irreversible impacts on both the natural and human environments. The United States Pipeline and Hazardous Materials Safety Administration reported more than 10,000 failures in oil and gas pipelines across the U.S. which incurred financial losses of around $6 billion in the form of property damage, production losses, environmental impacts and human casualties (U.S. Department of Transportation, 2013). Maintenance concerns are growing given that an estimated 72% of the existing 300,000 U.S. factories are more than 20 years old (The American Institute of Architects and Rocky Mountain Institute, 2013). The oil and gas industry is more prone to improper maintenance since more than half of the world's oil rigs will be more than 30 years old over the next 5 years (Phillips, 2017). As an example, the Deepwater Horizon Spill, one of the largest marine oil spills in history was caused due to poor maintenance of a drill pipe in the gulf of Mexico and caused damages of $17.2 billion across the Gulf coast (Office of Maritime Administrator, 2011). Unplanned shutdowns due to corrective or poor maintenance are estimated to cost $50 billions per year in the U.S. with 44% of all unscheduled equipment downtimes resulting from aging equipment (National Institute of Standards and Technology, 2018). Poor preventive maintenance also decreases the Overall Equipment Effectiveness (OEE) of a factory between 5 to 20% (PECI, 1999). These issues are mostly linked to inefficient and ineffective facility management and proper documentation of the existing conditions that lead to maintenance actions well after the problems have occurred. These have generated a market demand for a quicker and more efficient maintenance scheme of existing industrial facilities. Recent studies have shown that refurbishment and preventive maintenance of industrial assets will prevent the above-mentioned issues. For instance, the Chartered Institute of Building (Edwards and Townsend, 2011) have shown that the need for refurbishing and retrofitting 93% of existing industrial facilities will be a major focus in the U.K. construction industry by 2050. Another example of the perceived value of preventive maintenance proposed by the Association of Swedish Engineering Industries (Bokrantz et al., 2016) is the strategy to eliminate production shutdowns in Sweden by 2030. We argue that these market demands establish the need to generate and maintain up-to-date IgDTs. Yet most facilities do not have usable IgDTs. This occurs because the perceived cost of generating and maintaining the DT greatly counteracts the perceived benefits of the DT. The main reason for that is partly due to the high ratio of manual labor cost while generating the DT to data collection (laser scanning), which is roughly ten (Lu and Brilakis, 2017, Fumarola and Poelman, 2011, Hullo et al., 2015). This explains why there is an urgent need to generate less labor-intensive industrial modeling techniques that can improve the productivity of industrial assets and their maintenance. In this paper we address a core step of generating IgDTs, i.e. class segmentation of CLOI shapes.

Class segmentation is the foundation for many reverse engineering applications. It particularly facilitates clash detection analysis that managers of aging industrial facilities are confronted with (Akponeware and Adamu, 2017). This is only achieved by segmenting the class point clusters of interest and providing those to the Engineering Procurement Construction (EPC) engineers. For example, piping engineers would only be interested in inspecting the piping system. Structural engineers, on the contrast, would only be focused on the structural integrity of the industrial facility. Segmentation of all the points of primary (load-bearing) steel shapes will be helpful for stress analysis, Finite Element Analysis (FEA) (Song et al., 2018) and structural health monitoring of the steel frames (Park et al., 2007). Direct segmentation of points rather than generating a segmented IgDT will result in further cost savings. Therefore, segmenting the piping, structural and other important industrial objects from the TLS data is of paramount importance. Improving the effectiveness of class segmentation algorithms that take TLS data as input remains a challenge towards high level scene understanding solutions for industrial environments.

Leading 3D CAD vendors (Autodesk, AVEVA, Bentley, FARO and ClearEdge3D) have developed software containing a variety of 3D modeling functions that enable modeling from point cloud data, however none of those outputs class segmented TLS data. Geometric modeling using current software packages entails (a) primitive shape detection, (b) semantic classification of detected shapes and (c) fitting. Firstly, primitive shapes are detected (e.g., cylinders, tori, planes) and classified (e.g., pipes, elbows, I-beams). Afterwards, the primitives are fitted to known solid shapes to obtain their geometric parameters. A limited number of software achieve semi-automated modeling. We evaluated in our previous work (Agapaki et al., 2018) state-of-the-art commercial packages and demonstrated that EdgeWise (ClearEdge, 2019) provides to-date the most advanced semi-automated 3D modeling tool. The modeling of pipelines is summarized in three basic steps: (a) automated detection and fitting of cylinders, (b) semantic classification of cylinders and (c) manual extraction and editing of pipes. Structural sections are manually modeled. Fitting of user-selected primitives (e.g., circular hollow sections, cuboids, tori etc.) is performed automatically. EdgeWise automatically detects cylinders with 62% precision and 75.6% recall on average.

We also showed that semi-automatically modeling cylinders with EdgeWise reduces man-hours needed for modeling those by 64%. However, this means that for a petrochemical plant with 240,687 objects and 53,834 pipes, 2,382 manual labor hours are still needed to model these cylinders (Agapaki et al., 2018). EdgeWise does not generate cylinder class labels per point, it directly extracts cylindrical objects. Therefore, EdgeWise is not designed to uniquely assign class labels to points, but rather could fit multiple standardized cylindrical shapes to an individual point cluster. This assumption does not necessarily reflect the existing conditions of facilities, since cylindrical objects are either covered (anti-corrosion coating layer) or insulated, which means they are not straight cylinders. Another limitation is that, although EdgeWise is promising, it is far from a robust solution since there is high variability of the cylinder detection rates (standard deviation of 20.4% and 28.6% in recall and precision respectively) as proved by Agapaki et al. (2018). Cylinder detection in EdgeWise is also dependent on parameter selection by the modeler. These parameters are the maximum number of points to detect a cylinder and the distance tolerance which explains how far away from the cylinder a 3D point can be, so that it is not excluded from the extraction algorithms (Agapaki et al., 2018). As such, the state-of-the-art 3D modeling practice has three main limitations: (a) the modelers should segment the structural elements manually or roughly select regions of interest using clipping polygons to fit standardized structural steel shapes, (b) detection of cylinders has only been partially solved and is dependent on user defined parameters and (c) EdgeWise does not enrich the point cloud data with semantic class labels but only fits 3D solid standardized shapes. It is easily distinguishable that the current practice still needs substantial manual efforts and is not designed to offset the high costs of IgDT generation. This necessitates the need to redesign the procedure of IgDT generation.

We argue that cost reduction of IgDT generation will be achieved by automating the following steps: (a) class segmentation, (b) instance segmentation and (c) fitting. (a) describes the procedure to associate each 3D point of a laser scanned factory with a class label (such as cylinder, elbow, I-beam, valve, flange, angle and channel) (Li et al., 2019). Instance segmentation adds an instance label to the cluster of points (e.g. cylinder #2), but is beyond the scope of this paper. This paper is the first to automatically generate class segmented TLS industrial data. We present our novel automated CLOI-NET methodology in three parts: (a) a deep learning PointNET++ based geometric shape/class segmentation network, (b) optimization of the PointNET++ based network to boost class segmentation cost savings, should the user select it and (c) inference-rule segmentation enrichment for fine-grained class level predictions. We evaluate our CLOI-NET on our *CLOI* dataset. This is the first benchmark labeled dataset for industrial facilities that enables the use of supervised segmentation deep learning algorithms. We discuss the current state of research in Section 2 and we outline our proposed methodology in Section 3. We then elaborate on our research methodology and experiments in Section 4. Finally, we present our conclusions in Section 5.

2. BACKGROUND

There are two distinct IgDT generation strategies investigated in the literature. The first one (**S1**) involves two steps: (a) primitive industrial shape detection and (b) fitting. The second one (**S2**) has three steps: (a) class segmentation, (b) instance segmentation and (c) fitting. Therefore, we elaborate the current state of research in three parts: (a) industrial shape detection methods, (b) industrial shape class segmentation methods and (c) class segmentation deep learning methods with an overview of available TLS benchmark datasets. We discuss both detection and class segmentation methods in order to investigate the suitability of each for our industrial space application. We focus on the most important *CLOI* classes, namely: (a) cylinders, (b) structural steel shapes and (c) piping elements.

*2.1. Industrial shape detection (***S1***)*

2.1.1. Industrial cylinders

State-of-the-art research work has partially solved the cylinder detection problem and achieved similar performance compared to commercially available software packages like EdgeWise (Agapaki et al., 2018). Research studies do give us an idea of the methods that are likely used by EdgeWise given the similarity in performance (Jin and Lee, 2019, Ahmed et al., 2014, Patil et al., 2017, Sharif et al., 2017, Liu et al., 2013, Lee et al., 2013, Kawashima et al., 2014, Qiu et al., 2014, Bey et al., 2011, Rabbani et al., 2006, Su and Bethel, 2010). Research efforts so far have focused on automated cylinder detection by defining the five parameters that describe cylinder orientation, position and radius using a variety of methods. Most of the methods use pre-knowledge to detect cylinders: (a) cylinders in orthogonal directions (Liu et al., 2013, Kawashima et al., 2014, Ahmed et al., 2014, Qiu et al., 2014), (b) a priori CAD models (Bey et al., 2011) or (c) Piping and Instrumentation Diagram (P&ID) (Son et al., 2013).

Industrial cylinder detection methods are model driven. The most commonly used methods are based on RANdom SAmple Consensus (RANSAC) (Fischler and Bolles, 1981) and Hough Transform (Hough, 1959). The main limitation of RANSAC methods are their computational inefficiency in large TLS datasets with multiple cylinders given the large number of point selection needed. Hough Transform methods are limited for detection of cylinders with similar directional orientation in TLS data with multiple cylinders (Rabbani et al., 2006, Patil et al., 2017, Ahmed et al., 2014). The method proposed by Ahmed et al. (2014) has two additional limitations: (a) they only detect cylinders in orthogonal directions along the main axes of a facility and (b) the number of cylinders and diameters of cylinders are pre-defined to assist the detection procedure. Their assumption is that typical pipe diameters are within the range of 0.0508 and $0.1016m$ (2 and $4in$). Patil et al. (2017) recently developed a cylinder detection method that depends on threshold values for radius and normal estimation. Their cylinder radius range is $0.0254m$ - $0.762m$ and the normal deviation is $5°$. Their RANSAC and updated Hough Transform based on work by Rabbani et al. (2006) detects cylinders in two sample datasets with 60% recall and 89% precision. Our previous work (Agapaki and Brilakis, 2017) investigated the range of pipe radii being from $0.0075m$ to $0.525m$. Sharif et al. (2017) propose a model-based (RANSAC-based) cylindrical and structural object detection method by matching features of the acquired point cloud data with those of library generated point cloud models. However, the experiments are limited to a small-scale pipe spool and a structural frame and they are also dependent on manual effort needed to manually generate the library of point cloud models. Likewise, Liu et al. (2013) detect cylinders by detecting circles using RANSAC in projected planes in two orthogonal directions (parallel and perpendicular to the ground plane of an industrial facility). However, their main limitation is they cannot detect cylinders in arbitrary orientations. Recently, Jin and Lee (2019) proposed a RANSAC-based method to detect cylinders. They fitted spheres, connected their traces and then a RANSAC technique was applied to determine the axes of cylinders. There were several preprocessing steps required for plane removal and filtering. This method is promising (77% recall and 86.5% precision on average). As stated in their paper, a downside of the method is that the cylinder modelling performance is highly dependent on the sphere increments.

Other cylinder detection methods are highly dependent on user defined parameters, prior knowledge provided by the users or manual cropping of the initial TLS dataset. Lee et al. (2013) proposed a method to detect straight pipes, elbows and junctions from points in the piping system using a Voronoi diagram. However, the input point cloud only includes pipe elements and other parts of the piping system or industrial shapes such as flanges or valves and other parts of an industrial facility are manually segmented. This method makes the inherent assumption that it is comprised only of straight pipes, elbows and tees. If other objects exist, their method cannot distinguish them. For example, it could detect an I-beam as a straight pipe. This means that their method requires significant manual cropping to detect pipe elements in industrial **Point Cloud Datasets (PCDs)**. Kawashima et al. (2014) propose an automated method to detect straight cylinders, elbows and tees by using a normal-based, region growing method. Then, they estimate the positions and orientations of straight cylinders by calculating the eigenvalues and surface-normal vectors of their 3D points. The main limitation of this method is that their results are highly dependent on the parameters used in the detection method. Recall rates range from 60% to 94% depending on the parameters selected in their experiments. Son et al. (2013) and Son and Kim (2016) use P&IDs to assist the detection of straight cylinders, elbows, reducers and tees. The average overall recall of their method is 95%. However, as-is P&IDs are often not available as prior knowledge in industrial plants, thus they do not reflect the modifications a plant undergoes through its life. Li and Feng (2019) proposed the BAGSFit method that automatically segments boundaries with a CNN and then fits primitives (e.g. spheres, cones, cylinders and planes) from simulated and real-world RGB-D images. Similarly, Figueiredo et al. (2019) extract cylindrical shapes based on curvature and a-priori sampling of orientations and then extract 2D bounding boxes using a CNN network achieving performance of 72% precision and 63% recall on average. The above-mentioned methods detect match points to pre-defined cylinder models. Their detected points are then used to fit standardized cylinders.

A comparative study of the state-of-the-art research methods that have investigated cylinder detection is summarized in Table 1. The performance metrics used are precision and recall defined as follows (Powers, 2011),

$$Prec_c = \frac{|pred_c \cap gt_c|}{|pred_c|} = \frac{TP_c}{TP_c + FP_c} \tag{1}$$

$$Rec_c = \frac{|pred_c \cap gt_c|}{|gt_c|} = \frac{TP_c}{TP_c + FN_c} \tag{2}$$

where TP_c, TN_c, FP_c and FN_c correspond to the number of the true positive, true negative, false positive and false negative predictions per point for class c. $pred_c$ and gt_c correspond to the set of points predicted as class c and set of ground truth points that belong to class c respectively.

Table 1. Comparison of state-of-the-art research methods on cylinder detection

Method	**Performance**	
	Precision (%)	**Recall (%)**
Fast RANSAC (Jin and Lee, 2019)	77	86.5
Area-adaptive Hough Transform (Patil et al., 2017)	60.15	89.2
Hough Transform (Rabbani, 2006)	59.7	82.95
RANSAC on projected slices (Liu et al., 2013)	54.4	61.5
RANSAC (Schnabel et al., 2007)	50.7	26.3
Region growing (Kawashima et al., 2014)	50.1	88.9
P&ID (Son and Kim, 2016)	-	92.3

The performance of these primitive-based methods is rather low and cannot be generalized to large scale TLS industrial facilities. Another reason these methods are likely unsuitable for industrial cylinder modeling is the high relative ratio of the total number of TLS points in a dataset to the number of per cylinder points. Liu et al. (2013) demonstrated that RANSAC methods cannot be used on TLS data with cylinders that have significant variation in the number of their points. The suitability of RANSAC methods will also be investigated on our *CLOI* data in Section 3.3 of the proposed solution.

2.1.2. *Industrial structural steel shapes*

Detection of structural steel members in industrial TLS data is a challenging task that requires substantial manual modelling effort, since the methods that have been developed either only work on specific cases or have mediocre performance. Anil et al. (2012) investigate four manual methods to detect structural steel components: (1) point to point, (2) distance between edges, (3) distance between plane to plane intersection lines and (4) cross-section tracing. When compared to **American Institute of Steel Construction (AISC)** sections, their best method only achieves 18.75% accuracy for columns and 39.68% accuracy for beams. Yeung et al. (2014a) compare the Hough Transform method and a clustering method based on normal vectors to detect structural steel sections (I-beams), by slicing the point cloud in all orthogonal directions. Yeung et al. (2014b) use binary images and a predefined library to find the best match pixels of a standard steel section and the image. However, cross-section errors vary significantly (-41% to +15%). Laefer and Truong-Hong (2017) use a non-parametric, kernel density estimation method to detect the primary surfaces of structural steel members, which appear as local maximum peaks of probability density curves. They detect steel columns and I-beams with 85.7% recall and 77.8% precision. However, this method is only applicable to gridded structural members and these members are manually segmented from the noisy TLS point cloud data. Cabaleiro et al. (2014) use a Hough Transform method to automatically extract the web and flange lines of steel frame connections using 2.5D images and manually complete the steel frame using the software Solidworks 2012. The main limitation of these methods is that they only recognize members that are orthogonal to one of the slicing planes and are not applicable to occluded regions. Circular and rectangular columns have been successfully detected from rasterized images (Díaz-Vilariño et al., 2015) in partially occluded indoor environments. However, the main limitation of their method is that its success is dependent on data completeness. This means that if the positions of the laser scanner changed, this would greatly affect the columns detected using this method. Detection of structural steel shapes depends on matching the primitive shape with pre-defined steel profiles, which are again RANSAC- and Hough Transform-based. These methods are not further investigated given these methods' limitation on the relative number of points of the extracted shape to the total number of points and the manual user input.

The methods discussed in this section reveal that automated class detection of steel sections from industrial TLS data when fitted to standardized steel profile shapes, give accurate results for deformation modeling, stress analysis and **Finite Element Analysis (FEA)** has not been achieved yet.

2.1.3. Industrial piping elements

Elbows are curved joints, which connect two cylinders of the same radii to allow change of direction. Detection of elbows is based on the dot product of the vectors of the axis of connected cylinders (Kawashima et al., 2014). Although their method detects some types of elbows (45-, 90-degree elbows) with 58.1% precision and 90.85% recall, it does not recognize 180- and 120-degree elbows (return elbows) due to the assumption made that directions of the intersecting pipes should not exceed 90 degrees. Son and Kim (2016) only detect 45-,90- and 120-degree elbows with 97% recall. The main limitation of their method is that it relies on existing Piping and Instrumentation Diagrams (P&IDs) in order to determine curvature at points on the surface of pipes based on the radii of pipes.

Machine learning methods have been used for valve and flange detection. Pang and Neumann (2016) concatenate multiple Convolutional Neural Networks (CNNs) in projected 2D images generated by an exhaustive scanning window search. Their method allows for detection of valves with 77% recall and 88% precision. The advantage of this method is that it unifies the detection for multiple object classes with a multi-class CNN and uniform-size training samples without requiring prior segmentation of the scene. This method is promising, however it has a limitation. It is not designed for direct segmentation of TLS data, it rather requires to detect 2D shapes on projected depth images and then reprojects them in 3D. As a result, the detection of occluded shapes or shapes that are too close to each other is limited to the visibility of shapes on the projected views that are processed by the CNN. For instance, industrial spaces are highly congested and specifically many industrial shapes are closely located even overlapping each other such as electrical conduit that will not be visible in 2D projected views. Huang and You (2013) used Support Vector Machine (SVM) and local descriptor classifiers, Fast Point Feature Histograms (FPFH) and 3D Self-Similarity (3D SSIM) descriptors, to detect pipes, planes, parts of valve and elbow assemblies based on normal vector similarity. Then, they match the detected bounding boxes with the ground truth ones using a rigid body transformation with RANSAC. They achieve 87% precision and 62.5% on flange detection and 41.5% and 68% on valve detection. The method is limited to matching pipes, a specific type of valves (hand-wheel valves) and elbows with a library of pre-existing shapes. Another limitation is that their method does not detect the large tanks with small curvatures as cylinders but rather as planes.

The above-mentioned methods focus on detecting valves, flanges and elbows in synthetically or simplified industrial scenes, however none of these methods is designed to detect these shapes in real settings of multiple industrial facilities. Another limitation is that these methods are mostly focused on detecting individual shapes and not on segmenting all the points of specific classes.

*2.2. Industrial shape class segmentation (***S2***)*

In this section, we investigate the class segmentation methods applied on industrial shapes. Local descriptors have been used to segment cylinders in industrial scenes. Curvature based descriptors demonstrate superior performance compared to other local shape descriptors (Heider et al., 2012, Nagase et al., 2013). Dimitrov and Golparvar-Fard (2015) use a region growing method and principal curvatures as features to segment Mechanical, Electrical and Plumbing (MEP) systems in TLS point clouds. This method takes point cloud density, surface roughness, curvature and clutter into consideration. Although their main limitations are (a) over segmentation especially for highly occluded scenes and (b) lack of contextual inter-connectivity relationships to connect shapes, principal curvature is a local feature that can describe the 3D structure of points in occluded scenes and we will investigate using it in Section 3.5.1 of the proposed solution. Perez-Perez et al. (2016) use the segmentation method proposed by Dimitrov and Golparvar-Fard (2015) and refine class labels of indoor point cloud data using an SVM classifier and an Adaboost classifier. Then, they combine the semantic labels (wall, ceiling, floor, cylinder) and the geometric category labels (horizontal, vertical, cylindrical) learned into a Conditional Random Field (CRF) formulation to incorporate neighborhood context and their last step is to use a Markov Random Field (MRF) to enforce coherence between semantic (class) and geometric labels. Their results indicate 79% precision and 93% recall for pipe/cylinder segmentation. However, their main limitation is that their method is tested on simplified datasets since they manually segment their TLS data to only represent wall, floor, ceiling and cylinder components. This means that substantial manual effort is needed to pre-process the data. Huang and You (2013) segment four categories; planes, cylinders, edges and thin-cylinders (cylinders with diameter less than $5cm$). They use an SVM local descriptor classifier with point normals as local features (FPFH - Fast Point Feature Histograms and 3D-SSIM - 3D Self-Similarity). However, their method has mostly been tested on virtual point clouds and partial real-world industrial scenes, having less than 200,000 points. The above-mentioned research efforts on cylinder segmentation from TLS scanned data that constitute 50% of the objects of an industrial facility on average (Agapaki et al., 2018) indicate that this problem remains an unsolved challenge.

There are even fewer methods that assign a class label per point of steel shapes. Armeni et al. (2016) segment concrete beams and columns and other indoor object classes from TLS data using a 3D sliding window and an SVM classifier to learn local (occupancy, ratio, color, normals and curvature) and global features (3D position and size) in each 3D window. Their precision is 66.67% and 91.77% for beams and columns respectively. Steel structural components have not been investigated in their study. It is, therefore, evident that class segmentation of industrial shapes has not been solved in the literature. In the next section, we will investigate class segmentation methods using techniques applied in related fields.

2.3. Class Segmentation Deep Learning methods

In computer vision problems, image segmentation (referred to as semantic segmentation in the computer vision community) using hand-crafted features achieved a plateau in performance. CNNs are extensively used in image segmentation (Krizhevsky et al., 2012), text classification (LeCun et al., 2008), medical imaging (Taha and Hanbury, 2015, Pang et al., 2012) and self-driving vehicles (Wang et al., 2018a, Teichmann et al., 2018). A basic CNN architecture is using a deep neural network that combines convolutional and pooling layers to aggregate local information per pixel/letter in images/text respectively. Wang et al. (2019a) groups the existing 3D deep learning methods in three main groups: (a) view-based (Su et al., 2015b, Kalogerakis et al., 2017, Wei et al., 2016), (b) volumetric (Maturana and Scherer, 2015, Wu et al., 2015, Zhou and Tuzel, 2017, Klokov and Lempitsky, 2017, Tatarchenko et al., 2017) and (c) geometric deep learning methods (Qi et al., 2017b,a, Wang et al., 2019a).

There are three challenges that need to be addressed for the application of these techniques in real-world, TLS point clouds of industrial facilities:

1. TLS data is irregular (unstructured) and needs to be permutation invariant. This means that if the order of the points changes, this should not affect the result.

2. TLS data is noisy, sparse, with outliers, occlusions, density variations and especially for industrial settings, large-scale.

3. TLS industrial shapes can have different scales, objects in these point clouds may have the same shape but can be translated or rotated to their principal axes. Therefore, the selected method should be rotation and translation invariant. Objects of the same class can even have substantially different geometric shapes, e.g. valves (European Commission, 2010).

Geometric deep learning methods address all three challenges by using conventional building blocks like convolution and pooling to directly process 3D points. Henceforth, these methods will be further analyzed due to the scope of this paper that is focused on directly assigning class labels to points rather than converting the 3D points to other representations in order to process them.

2.3.1. Geometric Deep Learning Methods

The key difference between geometric deep learning methods and traditional approaches is that the former are feature-agnostic, i.e. they have to learn the shape features instead of hand-crafting them. Geometric deep learning has become a core technique for class segmentation tasks (Qi et al., 2017a,b). Prior to deep neural nets, class segmentation of images and point clouds was traditionally solved using feature extractors (such as spin Images (Johnson and Hebert, 1999)) combining classical classifiers such as SVMs (Agrawal et al., 2009)), semantic hashing (Behley et al., 2010)) or Conditional Random Fields (CRFs) to enable label consistency in neighboring points (Munoz et al., 2009, 2008, Triebel et al., 2006). A comprehensive overview of hand-designed point features is out of the scope of this paper, but our readers can refer to Biasotti et al. (2016), Guo et al. (2014), Patraucean et al. (2015) and Grilli et al. (2017).

Laser-scanned point clouds are massive datasets, where, unlike images, convolution operations between 3D points cannot be performed since point clouds are unstructured and this prohibits the use of 3D CNNs. For this reason, PointNETs were developed. PointNETs are a special class of network architectures that process point cloud data in 3D space. Their key operation is a symmetric function applied to 3D coordinates so that they are invariant to permutations. Qi et al. (2017b) developed a deep neural network (PointNET) that takes point clouds as inputs and outputs segmented labeled point clusters. PointNET was trained on the ShapeNet data set (Su et al., 2015a) and the Stanford3D indoor dataset (Armeni et al., 2016). However, PointNET is not designed to capture spatial relationships between features. In other words, PointNET processes fixed-size blocks separately, and uses fully connected neural network layers for the points of each block. However, this implies that it treats local information the same way as global information during the learning process, something that impairs the learning procedure. Also, the learned features are sensitive to the global transformation and rotational transformations of point clouds due to loss of neighboring information per point.

PointNET++ (Qi et al., 2017a) solved this problem by applying individual PointNETs to local neighborhoods of points and combined their outputs by using a hierarchical approach. As such, PointNET++ captures both local and global contextual information. PointNET++ has been widely used in buildings (Chen et al., 2019) and urban scenes (Behley et al., 2019). Chen et al. (2019) use a graph-based method to represent the connectivity between objects and segment them using PointNET++ (77.9% accuracy). A similar approach was studied by Shen et al. (2018) and exploited local high-dimensional feature vectors based on a nearest-neighbor-graph, which is constructed from the locations of 3D points. Another limitation of CNNs when applied on TLS data is that CNNs cannot adjust to point density variations (Li et al., 2016), since they process structured data (in a grid). For this reason, other techniques like projecting the point cloud to a voxel grid, tracking non-empty voxels using a hash table and then performing sparse convolution were used (Choy et al., 2019) or TLS data points were spherically projected to an image (Wu et al., 2018). The former is ideal for data of video sequences since it allows an extra spatial dimension (time), thus creating networks with 4-dimensional convolutions. The latter takes into account the geometry of a rotating LIDAR sensor and after application of a CNN, results are smoothed using a CRF. A recent approach used local information between pairs of neighborhoods of points and propagated this information by using EdgeConv layers (Wang et al., 2019b).

Most of these deep learning networks are designed to use both spatial coordinates and RGB information per point. However, we argue that the latter is not suitable for industrial environments, because color does not give unique information to distinguish shapes in industrial spaces. For instance, cylinders can have the same color as structural elements and color is dependent on the manufacturers' specifications. ANSI/ASME A13.1 (American Society of Mechanical Engineers (ASME), 2015) is the most commonly used general purpose color coded scheme. Some industries require an even more detailed and customized standard coding scheme. For instance, water treatment pipes follow the Ten States Standards (Lakes, 2004), which depends on the fluid carried on the pipelines. Other systems are so specialized that color coding should even distinguish between pipes carrying the same material using the phase of the material as color identifier (International Institute of Ammonia Refrigeration, 2014). Fig. 1 compares three widely applied piping color coding schemes, the ASME A13.1 standards (American Society of Mechanical Engineers (ASME), 2015), the British standards (BS 1710:2014, 2014) and the ANSI-APWA **(American Public Works Association)** standards (ANSI Z535.1, 2017). We observe that all pipes except the ones used for fire purposes are painted with different colors based on the color coding scheme. This explains why we argue that there is no universal and widely applied color scheme that can be used as a unique feature for a geometric deep learning network on the class segmentation of industrial shapes.

Type/Material properties	ANSI A13.1	BS 1710	APWA
Water	Green	Blue	Blue
Steam	User defined	Silver/Grey	Yellow
Oils	Brown	Brown	Yellow
Gases (except air)	Brown	Yellow ochre	Yellow
Air	Blue	Light Blue	Yellow
Other liquids	Orange	Black	Green
Electrical & ventilation	User defined	Orange	Red
Fire	Red	Red	Red

Fig. 1. Comparison of color between pipe color coding schemes.

We also explored the suitability of intensity as a feature for deep learning networks in industrial TLS datasets. Our experiments were based on the same network architecture as shown in Table 7, however we observed significant overfitting. We attributed that to surface reflectivity and roughness that did not facilitate the learning process. As the learning process in this work is geometry

driven, we have not investigated experimentally the reasons behind this issue, or ways to amend it.

We provide here potential limitations of using intensity, as documented in the literature. It is known that LiDAR intensity values are greatly affected by factors related to data acquisition geometry (i.e. distance between the laser scanner and the target object or the angle between the emitted laser beam and the target surface normal) (Korpela et al., 2010, Yan et al., 2012, Kukko et al., 2008, Pfeifer et al., 2007, Vain et al., 2009, Krooks et al., 2013, Coren and Sterzai, 2006, Ding et al., 2013, Höfle and Pfeifer, 2007, Jutzi and Gross, 2009, Kaasalainen et al., 2011). Most LiDAR scanners use near-infrared lasers that are sensitive to environmental effects and weather conditions (e.g. temperature of surfaces, moisture, fog or rain), which impacts the intensity (Yan et al., 2012, Höfle and Pfeifer, 2007, Kashani et al., 2015, Shin et al., 2019, Ijaz et al., 2013, Filgueira et al., 2017). Solar exposure can also impact intensity (Gatziolis and Andersen, 2008).

In summary, intensity values are primarily affected by two environmental factors:

1. diverse scene settings in point cloud datasets (outdoor and indoor settings). Three of the datasets we used were indoor scenes with the fourth one being an outdoor facility,
2. variation of scene settings in the same facility. A few examples of those are hot surfaces versus cold surfaces. The intensity of a pipe that contains hot liquids is completely different from a pipe that is not operational and therefore its surface is cold.

Further research is needed to address the limitations due to laser reflectivity on surfaces, which is beyond the scope of this paper. Our proposed solution and scope are geometry driven, so the use of intensity will not be further analyzed. Future work could explicitly analyze the effects of adding intensity information on class segmentation of industrial TLS data by pre-processing intensity data (Alkadri et al., 2020) and proposing alternative network architectures that include intensity values as additional features.

Geometric deep learning methods address all three challenges of real-world, TLS data that were presented earlier. Henceforth, these methods will be further investigated due to the scope of this paper on class segmentation that is focused on directly assigning class labels to points rather than converting the 3D points to other representation in order to process them.

Applications of the above-mentioned networks range from indoor to urban scenes. However, none of them is implemented on industrial facility data. The Stanford 3D Indoor spaces dataset (Armeni et al., 2016) is extensively used to validate these methods on indoor TLS datasets. Spatial coordinates, color information and relative position of each point within room settings are used as learnable features to apply the geometric deep learning networks mentioned above. It can be easily understood that these methods cannot be directly applied on industrial environments since these spaces present three main challenges: (**C1**) there is no universal color scheme that is followed across different facilities as proved above, (**C2**) industrial spaces are typically large and semi-structured with shapes that may span across their entire length/width and (**C3**) they are heterogeneous spaces where there are usually no direct contextual rules between shapes that belong in separate systems (piping, structural, electrical) and only the components that belong to the same system are internally connected with strong context. In other words, the relative location of a cylinder in a facility cannot imply derivation of contextual rules for the position of an I-beam. A prerequisite to apply a class segmentation deep learning network is the availability of a benchmark TLS dataset. Hence, we investigate in the next section the requirements in terms of the size and techniques to generate a TLS dataset on industrial spaces.

Benchmark datasets. Manual extraction of thousands of point clusters from point clouds for a segmentation algorithm is a tedious process that prohibits training for the application of deep learning algorithms. The collected point clouds need to be annotated in order to allow the use of supervised learning multi-classifiers. There are several benchmark datasets of indoor scenes, which are generated by RGB-D cameras or are synthetically generated (Armeni et al., 2017, Dai et al., 2017, 2018, Hua et al., 2016, Li et al., 2018, McCormac et al., 2017, Zhang et al., 2015, Silberman et al., 2012). A lot of work has also been done on road scenes from images (Ros et al., 2016, Chen et al., 2016, Song et al., 2015, Xiang et al., 2015, Zeeshan Zia et al., 2013, Zia et al., 2014) and recently promising work using voxelization techniques in neural network architectures for TLS point cloud data was conducted by (Zhou et al., 2017). TLS benchmark datasets of urban scenes for self-driving car applications have recently been developed such as the Oakland3d, Freiburg, Wachtberg, Semantic3d, Paris-Lille-3D, Zhang et al. and KITTI (Behley et al., 2012, Hackel et al., 2017, Munoz et al., 2009, Roynard et al., 2018, Steder et al., 2010, Geiger et al., 2012). Current benchmark datasets are summarized in Table 2. Henceforth, there is no benchmark dataset to date that captures TLS point clouds of industrial facilities. As such, there is an imperative need to generate a dataset to use supervised methods like deep learning. The benchmark datasets in Table 2 give us intuition on the acceptable number of shapes/3D points to target for our dataset generation.

Table 2. Overview of point cloud datasets with class annotations. [1]refers to the number of points in millions of each dataset, [2]refers to the number of classes used for evaluation and number of classes annotated is in brackets

	#points[1]	*#classes*[2]	**Sensor**	**Annotation**	**Use**
Semantic3D	4,009	8 (8)	Terrestrial 3D Laser Scanner	Point-wise	Urban scenes
KITTI	1,799	3	Velodyne HDL-64E	Bounding box	Urban scenes
Stanford 3D	273	13 (13)	Matterport 3D scanner	Point-wise	Buildings
Paris-Lille-3D	143	9 (50)	Velodyne HDL-32E	Point-wise	Urban scenes
Zhang et al.	32	10 (10)	Velodyne HDL-64E	Point-wise	Indoor scenes
Oakland3D	1.6	5 (44)	2D laser scanner (SICK LMS)	Point-wise	Outdoor
Freiburg	1.1	4 (11)	2D laser scanner (SICK LMS)	Point-wise	People/bicycles
Wachtberg	0.4	5 (5)	Velodyne HDL-64E	Point-wise	Urban scenes

A technique of hand labeling through crowd sourcing has emerged for images (Silberman et al., 2012, Song and Chandraker, 2015). For this purpose, crowd sourcing platforms like Amazon Mechanical Turk (Amazon Mechanical Turk, 2018) or LabelMe (Russell et al., 2008) have been developed. However, it is more difficult to accomplish this task for TLS point clouds due to noise, occlusions and difficulty to interpret cluttered 3D scenes for untrained users. Industrial scenes are a significant example of complex scenes with thousands of object categories that make hand-labelling even more time-consuming. Henceforth, another annotation needs to be used to generate a TLS benchmark of industrial facilities.

2.4. Gaps in knowledge, objectives and research questions

Considering the state of practice and body of research reviewed above, existing works that attempted to segment industrial scenes using **S2** methods only focus on cylindrical or piping components. Research efforts attempted cylinder detection using **S1** methods, however these methods are parameter dependent and not extensively tested on large-scale facilities. Therefore, semantic information from point cloud data is lost and standardized shapes do not usually capture existing geometric shapes. Cylinder detection **S1** methods that have been widely researched do not provide information per point, which can lead to mislabeled points and erroneous IgDT generation in highly occluded industrial scenes. For example, points of sagging pipes may be excluded when a bounding box is fitted in the cylinder. Although EdgeWise saves up to 64% of the cylinder modeling time (Agapaki et al., 2018), there is still substantially high manual modeling time involved resulting in high modeling cost. **Gap 1** No method on cylinder segmentation has effectively reduced the substantial manual modeling time required for cylinder segmentation from TLS industrial point cloud data and it is still unclear whether **S1** or **S2** methodology benefits the IgDT generation.

Structural steel shapes (channels, I-beams and angles) and other piping components (elbows, flanges and valves) are detected manually or substantial manual cropping of industrial shapes that are not of interest to detect is involved in **S1** methods. **Gap 2**. Class segmentation of industrial steel shapes and piping elements has not been solved. All the above-mentioned methods measured their success solely by taking into consideration the detection **S1** or segmentation **S2** performance. **Gap 3**. No method has optimized both time and performance, which both affect the cost of IgDT generation.

Therefore, we argue that a method that satisfies all the user requirements in IgDT generation is missing in the literature. We therefore contend that the problem of automatically generating IgDTs at a low cost has yet to be solved and is conditional on accurately segmenting *CLOI* class point clusters from industrial TLS data.

The objectives of this work are to:

- *Objective 1*: Automatically segment cylinders from TLS data with robust performance. This will be achieved by answering the following research questions; **RQ1a**: Which cylinder gDT methodology is more efficient for cylinder segmentation: **S1** or **S2**? A subsequent question that then needs to be answered is **RQ1b**: How to automatically segment cylinders without relying on prior knowledge such as 3D models and user defined cylinder geometry?

- *Objective 2*: Automatically segment the most important *CLOI* industrial shapes from the TLS data without manual cropping of irrelevant point clusters. This will be tackled by answering the following research questions; **RQ2a**: How to automatically segment industrial steel shapes given their highly occluded and noisy profiles? And **RQ2b**: How to automatically segment with robust performance the *CLOI* shapes other than cylinders and steel shapes from TLS data with varying point densities, occlusions and outliers?

- *Objective 3*: Automatically segment all the *CLOI* industrial shapes with optimal trade-off between manual effort needed and segmentation performance. This will be done by answering the research question; **RQ3**: How to optimize both the manual labor costs and the segmentation performance?

3. PROPOSED SOLUTION

3.1. Scope

We focus on the class segmentation of the most important industrial *CLOI* shapes as identified in our previous work (Agapaki et al., 2018), since these shapes constitute 75% of industrial facilities on average. We also group all cylindrical shapes in one category, namely "cylinders". The prioritized shapes that we focus on are: cylinders, elbows, channels, I-beams, angles, flanges and valves. Most of the *CLOI* shapes match one to one to a component class, (i.e. the shape is unique to this component), but for cylinders the shape is not unique. So we are segmenting the *CLOI* shapes, and by default, we also segment their component classes except for cylinders. Segmentation of the subcategories of cylindrical shapes (i.e. pipes, circular hollow sections, handrails, electrical conduit) is beyond the scope of this research. TLS scanned datasets typically have (1) cylinders with diverse sizes and (2) total number of TLS points being a lot more than the number of points of a cylinder. In this paper, we only focus on the class segmentation of *CLOI* shapes and not on the instance segmentation of those.

3.2. *Overview*

Fig. 2 presents the workflow of our proposed methodology. The inputs of our method are the spatial coordinates of TLS points and the outputs are labeled, segmented point clusters with confidence levels of the predictions. Here we define segmented point clusters as all the points that belong to one class i.e. all cylinder points is one class point cluster. The method consists of three major steps: **Step 1** partitions each facility into smaller spaces using a 3D sliding window/block approach and prepares the data for training, **Step 2** predicts a class label per point using a modified version (SFR) of a geometric deep learning network for point cloud segmentation (PointNET++) with the goal to accurately segment the *CLOI* shapes. **The name SFR stands for Smaller and Fewer neighbourhoods with smaller Radius.These choices will be explained in detail in this section.** In **Step 2**, the user has two options on how to train the network, either training with no data from the test facility or manually annotating data of the test facility and including those for training. The latter is based on the assumption that, inevitably, any class segmentation algorithm will have errors, which will have to be manually corrected eventually. Therefore the goal is to minimize the total manual annotation time. **Step 3** refines the predicted class labels by improving class level predictions with stronger contextual relationships. We name our methodology *CLOI-NET*.

Step 2 is further divided in two sub-steps depending on whether the user intends to annotate part of the test facility. **Step 2a** will focus on the class segmentation network without user annotation, whereas **Step 2b** will involve user annotation. **Step 3** is partitioned into three sub-steps that are implemented regardless of the decision on user annotation. These are a cylinder classifier (**Step3a**), a steel shape segmentation algorithm (**Step 3b**) and a class label confidence adaptation method (**Step 3c**). The sections that follow describe each step of the proposed solution in detail in order to answer the research questions presented in Section 2.4.

3.3. *Step 1: 3D building block generation*

We first evaluate the applicability of cylinder detection methods (**S1** methods) and particularly, RANSAC. We follow the same assumption with Liu et al. (2013) to determine the number of uniformly random 3D point selections per cylinder needed: each cylinder will be

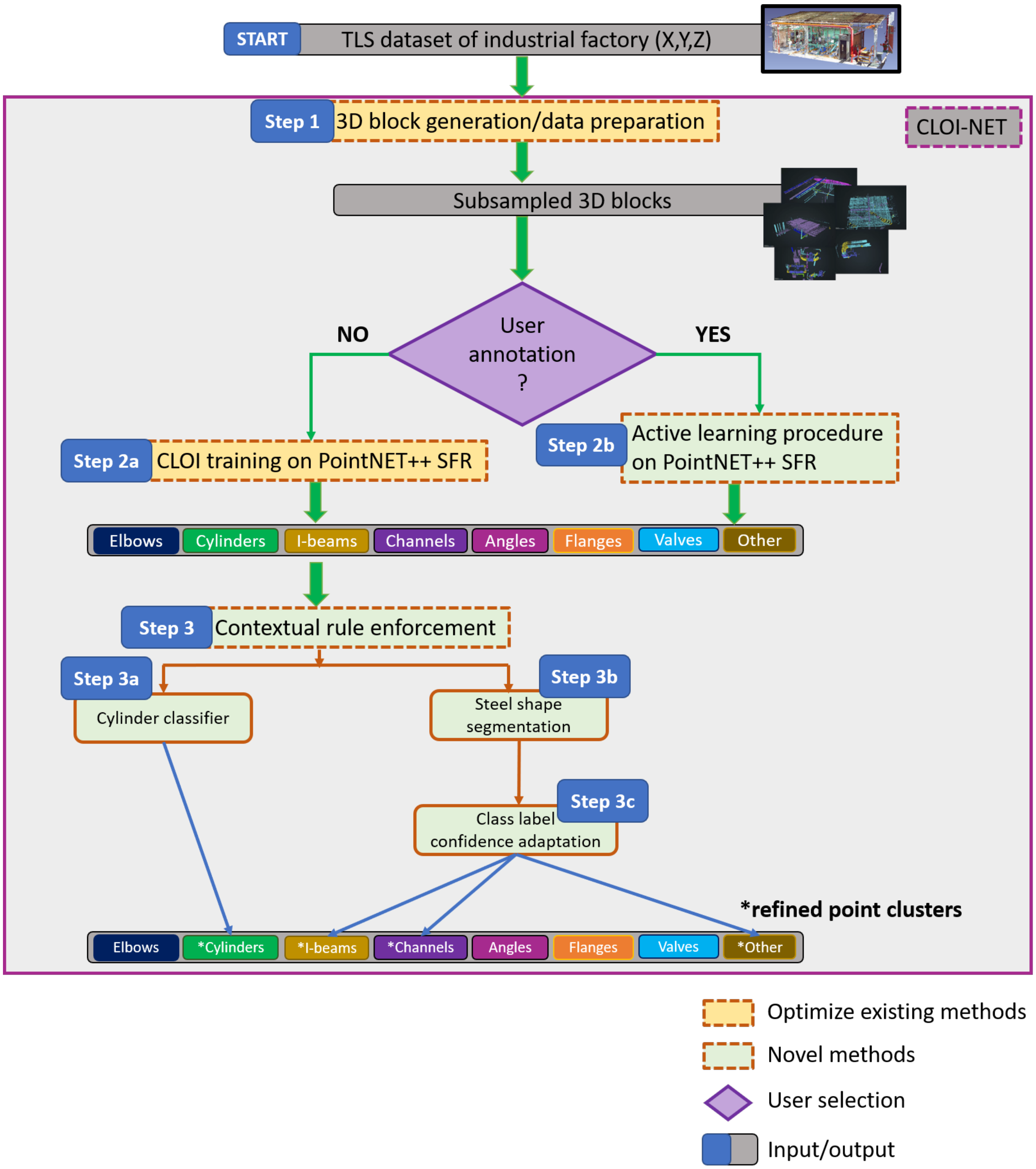

Fig. 2. Proposed *CLOI-NET* methodology

detected with the probability of at least 90%. This is achieved by repeatedly selecting five uniformly random points per iteration from a given TLS point cloud of a facility according to the RANSAC algorithm proposed by Devillers et al. (2003). Therefore, the number of random selections (k_i) of 3D points for each cylinder is given by:

$$k_i \approx \frac{\log 0.1}{\log(1 - X_i^5)} \tag{3}$$

where X_i is the probability of a point to belong to a cylinder i.

Table 3. Cylinder data summary for RANSAC evaluation

Facility	**Number of cylinders**	**Point Number of cylinders**	**Percentage in total points (%)**	**Selection number in RANSAC**
Oil refinery	1	$2.15 * 10^8$	6.63	$1.8 * 10^6$
	1	$2.15 * 10^8$	6.6	$1.83 * 10^6$
	1	$1.37 * 10^8$	4.2	$1.73 * 10^7$
	1	$5 * 10^7$	1.52	$2.82 * 10^9$
	1	$3.8 * 10^7$	1.19	$9.54 * 10^9$
	2341	$<2 * 10^7$	<0.5	$>3.6 * 10^{11}$
Warehouse	1	$5.9 * 10^5$	0.71	$1.22 * 10^{11}$
	1	$5.73 * 10^5$	0.7	$1.38 * 10^{11}$
	1	$5.46 * 10^5$	0.66	$1.75 * 10^{11}$
	1	$4.62 * 10^5$	0.56	$4.08 * 10^{11}$
	1	$4.15 * 10^5$	0.5	$6.94 * 10^{11}$
	899	$<4.1 * 10^5$	<0.5	$>7.7 * 10^{11}$
Petrochemical plant	1	$3 * 10^5$	0.82	$6.34 * 10^{10}$
	1	$3 * 10^5$	0.82	$2.36 * 10^{11}$
	1	$2.31 * 10^5$	0.63	$3.65 * 10^{11}$
	1	$4.62 * 10^5$	0.58	$4.91 * 10^{11}$
	1	$4.15 * 10^5$	0.54	$9.4 * 10^{11}$
	1483	$<1.8 * 10^5$	<0.5	$>1.1 * 10^{12}$
Processing unit	1	$7.7 * 10^6$	2.18	$4.67 * 10^8$
	1	$4.29 * 10^6$	1.21	$8.76 * 10^9$
	1	$2.31 * 10^5$	0.63	$8.26 * 10^{10}$
	1	$4.62 * 10^5$	0.58	$9.09 * 10^{10}$
	1	$4.15 * 10^5$	0.54	$1.34 * 10^{11}$
	1	$4.15 * 10^5$	0.54	$1.85 * 10^{11}$
	1094	$<2.2 * 10^6$	<0.5	$>2.64 * 10^{11}$

The huge number of uniformly random selections of points per cylinder i in Table 3 demonstrate that RANSAC cannot be directly applied in TLS industrial data, since it is computationally intractable ($> 3*10^{11}$ points selected per iteration in every industrial facility tested). The majority of cylinders have very few points relative to the total number of points in the TLS dataset ($< 0.5\%$). This leads to a very high number of uniformly random point selections per cylinder. The data statistics and results in Table 3 confirm the observations discussed by Liu et al. (2013) and answer the **RQ1a** that modeling of cylinders should not be considered as a detection problem and then fitting cylinder primitives. Rather, it should be solved with **S2** detection methods as a class segmentation and instance segmentation problem. The rest of *CLOI* shapes have even fewer points relative to the total number of points in the TLS dataset and a model-based method would require an even higher number of parameters to detect them. There is also an increasing trend in the computer vision community to shift away from traditional model-based methods and apply deep learning methods due to their reliable performance and scalability (Qi et al., 2017a,b, Wang et al., 2019b, Lecun et al., 2015, Santhanam et al., 2019, Zhang et al., 2019). In addition to that,

- *R1*: class segmentation gives representation power to the TLS point clouds as mentioned by Wang et al. (2019b). This is because it embeds topological information directly on the real point cloud representation, without the need to introduce bounding boxes or shape primitives into the representation, as was previously required when using **S1** object detection methods.

- *R2*: the robustness of primitive fitting RANSAC-based method is highly dependent on the spatial distribution of samples (Liang et al., 2018). In other words, samples with points that are closely located to each other usually cannot be properly detected (Liang et al., 2018, Li and Feng, 2019).

Therefore, we propose the following methodology to solve the class segmentation problem of cylinders and the rest of *CLOI* classes to answer research questions **RQ1b**, **RQ2a** and **RQ2b**.

We use a 3D block parser to slice the facility into smaller pieces that will then be used for training a geometric deep learning network. We follow the four conventions proposed by Qi et al. (2017b): (a) the 3D blocks are overlapping, (b) the 3D blocks are subsampled, (c) we define the horizontal plane dimensions as the XY-plane for each 3D block of a facility and (d) the height dimension of each block to be parallel to the Z-axis of the facility. We use principal component analysis (PCA) to align a TLS industrial point cloud such that the XY-plane of each 3D block is positioned roughly parallel to the global XY-plane. The 3D blocks are used for training in Step 2 ("learnable" blocks).

We first determine the 3D "learnable" block dimensions by investigating industrial shape dimensions. We conduct a statistical analysis on the dimensions of industrial piping shapes in existing as-designed facilities. The Outer Diameter (OD) of piping elements as defined in Agapaki and Brilakis (2017) shows that the OD ranges between 10 mm and 1050 mm. We determine the dimensions of structural steel industrial members based on British Standards. The profiles of I-beam and channel steel sections can be characterized by the following independent parameters: (a) width of the section (B), (b) depth between fillets (d), (c) thickness of the web (t_c) and (d) thickness of the flanges (t_f). We compute the mean and standard deviation of these parameters for steel profiles based on the British Steel Manuals (BS EN 10365:2017, 2017) and these are summarized in Table 4. These parameters can be generalized based on steel shape dimensions on other specification catalogues (AISC, 2016, CISC, 2015, European Standard, 2005). Table 4 shows that the dimensions of the steel sections are within the range of 1 m^2, with maximum steel section dimension that of $0.4m$ for the I-beam sections. This means that the selection of $1m$ side of 3D blocks is reasonable as proved by our analysis. The cubic blocks are the units used for training the facilities in Step 2. We then present in the next section the class segmentation network architecture and parameters chosen.

3.4. Step 2a: Class Segmentation Network

We use a deep learning network for an initial class segmentation of our TLS point cloud data. We experimentally prove the applicability of the PointNET++ network for our *CLOI-NET* method, by testing the three state-of-the-art point cloud segmentation networks for our pointwise application: PointNET (Qi et al., 2017b), DGCNN (Wang et al., 2019b) and PointNET++ (Qi et al., 2017a). We use precision Eq.1, recall Eq.2, accuracy and the commonly used mean Jaccard Index or mean intersection-over-union ($mIoU$) metric (Everingham et al., 2014) to measure the performance of the above-mentioned networks. We

Table 4. Shape parameters of I-beams, Channels and Angles

Shape parameter	Mean (mm)			Standard deviation (mm)		
	I-beam	Channel	Angle	I-beam	Channel	Angle
Width of section (B)	239.5	83.1	171	86.1	13.4	33.8
Depth between fillets (d)	387.2	178.6	146	211	79.6	34
Thickness of the web (t_c)	13.3	7.2	14.4	7	1	4
Thickness of the flanges (t_f)	20.4	13.2	-	11.7	2.8	-

define accuracy and mIoU as;

$$\text{accuracy} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c} \tag{4}$$

$$\text{mIoU} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FP_c + FN_c} \tag{5}$$

where TP_c, TN_c, FP_c and FN_c correspond to the number of the true positive, true negative, false positive and false negative predictions per point for class c and C is the total number of classes.

We measure the success of a deep learning segmentation network based on the mIoU metric, since precision and recall do not sufficiently explain the prediction results. Class segmentation errors occur due to two main reasons. Assuming that we have a binary classification problem, 100% precision does not imply sufficient performance since the algorithm may only correctly predict a small part of the TLS data and incorrectly predict the rest of the point cloud. Mathematically, this can be expressed as $pred_c \cap gt_c = pred_c$ or equivalently the predicted points are a subset of the ground truth points. As such, 100% precision can be associated to a very low recall. Similarly, recall cannot solely describe a good classifier since the classifier may consider all the TLS dataset and predict that all points belong to a single class. Mathematically, this means that $pred_c \cap gt_c = gt_c$, which in other words means that the ground truth points are a subset of the predictions, therefore although recall is 100% precision is very low. Therefore, we need to use another metric that does not reward recall or precision for successful implementation of our segmentation networks. This is mIoU, which synthesizes precision and recall (Eq.5). The mIoU metric has also been used for class segmentation of indoor 3D spaces (46.67% mIoU in Qi et al. (2017b) and 56.1% in Wang et al. (2019b)). As presented in Table 5, PointNET++ outperforms the other two networks in all efficiency measures (accuracy, precision, recall and mIoU) and especially mIoU, as such we choose it as a baseline to our *CLOI-NET* methodology. Although promising, the class segmentation rates still have room for improvement (32% mIoU). Hence, we need to fine-tune the PointNET++ network to address the challenges of TLS industrial point cloud data. We validate these experiments on an oil refinery dataset (part of the *CLOI* dataset).

Table 5. Performance of class segmentation deep learning networks for the oil refinery dataset

Network	Accuracy (%)	Precision (%)	Recall (%)	mIoU (%)
DGCNN (Wang et al., 2019b)	66	36	31	22
PointNET (Qi et al., 2017b)	50	21	19	12
PointNET++ (Qi et al., 2017a)	**68**	**46**	**41**	**32**

PointNET++ receives as input a cluster of points and outputs a category prediction among the 8 *CLOI* classes. Industrial TLS data have three challenges (**C1, C2** and **C3**) as discussed in Section 2.3.1. An additional challenge (**C4**) for the application of deep learning networks is that industrial TLS data are imbalanced datasets in terms of the number of points per class to the total number of TLS points as proved by Agapaki et al. (2018). We exclude RGB data from our input due to **C1**. The original version of PointNET++ is based on relative (with respect to the 3D block) spatial coordinates, RGB data and normalized absolute coordinates in the range $[0, 1]$. Normalized absolute coordinates are not relevant in the industrial settings, since these were used to obtain features related to the position of the 3D block within a building room, which is not applicable in our case since instead of rooms we have large, unstructured spaces and shapes not directly connected with contextual rules (challenge **C2** and **C3**). Therefore, we only use relative spatial coordinates to train our fine-tuned PointNET++. Although the speed of convergence will not be our main concern in this work, it is noteworthy that when one balances the training classes by oversampling blocks that have the least frequent classes to address the challenge **C4**, training converges around 60% faster. We achieve class balancing by selecting equal number of blocks of each class in each training epoch of the network in Algorithm 1. Also, a small number of points is insufficient for accurate predictions of shapes, even for a human observer. We discard blocks that have less than 100 points to overcome this issue as proposed by Qi et al. (2017a). We present the tunable parameters of PointNET++ in the next paragraphs.

Algorithm 1 Class balancing algorithm

procedure UNIFORM CLASS BALANCING BETWEEN BLOCKS
    for epoch $k = 1 \ldots N$ **do**
        for batch $i = 1 \ldots X$ **do**
            $cur\_batch \leftarrow \emptyset$
            for block $b = 1 \ldots B$ **do**
                pick class $j \in [1, \ldots, 8]$ uniformly at random
                sample block cur_block that contains ≥ 1 point of class j uniformly at random
                $cur\_batch \leftarrow cur\_batch \cup \{cur\_block\}$
            Train on cur_batch

We group the tunable parameters of PointNET++ into two distinct groups: (1) geometric hyper-parameters, which depend on each neighborhood scale and (2) network-related hyper-parameters as presented in Table 6. These parameters are essential for the key building block of PointNET++, which is its sampling module. This module aggregates features from each neighborhood like a CNN would do for pixels in image segmentation problems. We fine-tune the geometric parameters of this sampling module to better fit the intricacies of industrial shape data. The search radius, r_i of the neighborhood ball (a) and the number of neighbors (b), denoted as $N(q_i)$, where q_i is the center point of each neighborhood define the neighborhoods from which features are extracted and their estimation is presented as follows. Parameter (c) is the number of neighborhood centers denoted as (q_i) for which the neighborhood information is aggregated. Parameters (d) and (e) directly influence the architecture of the neural network in each scale. (d) is applied to all points of a specific neighborhood and (e) is the size of the neural network applied after the backwards feature extrapolation. The dropout rate (g) is adjusted to avoid overfitting and the learning rate (h) influences the convergence speed and capability of the network to generalize. The network parameters are discussed in the next section at the method implementation.

Table 6. Fine-tuned parameters of PointNET++ SFR

Geometric parameters	(a) search radius
	(b) number of neighbors
	(c) number of points to subsample
	(d) size of the MLP representing h
	(e) size of the MLP for extrapolation
Network hyper-parameters	(g) dropout rate
	(h) learning rate

For the geometric parameters, we create six network architectures containing different neighborhood criteria. These architectures are listed in Table 7. The parameters are chosen heuristically based on the amount of information present in neighborhoods of different sizes and measured by a metric derived by the authors (neighborhood rate). The sizes of the MLPs are motivated by the ones used by Qi et al. (2017a), whereas the neighborhood sizes are motivated by the optimal neighborhoods derived from the dimensions of *CLOI* shapes as explained below.

In particular, industrial shapes (especially steel shapes) have very fine details as summarized in Table 4. As such, we need to capture fine-grained regions within each "learnable" unit block. The search radius (r_i) at the largest scale is adapted to be less than $0.8m$ per neighborhood and choices of radii per each scale are based on PointNET++ (Qi et al., 2017a). We carry out a random parameter search from a manually selected pool of parameters and the experimental results of the six PointNET++ SFR architectures are presented in Table 8. For this parameter search, we only test on the oil refinery dataset, since it is a representative facility of our *CLOI* dataset. We observe that the capacity of the network slightly changes with the choice of parameters. This is quantified by an achieved overall accuracy ranging between 69 and 72%, mIoU between 34 and 38%, precision and recall changes of $\pm7\%$ and $\pm6\%$ respectively. SFR_3 generates optimized results by applying the following changes from the original PointNET++ parameters: (a) reduction of the numbers of the neighborhoods (i) in each block from 1024 (PointNET++) to 512 points, (b) reduction of the max radius (r_i) of each neighborhood from $0.1m$ (PointNET++) to $0.05m$ and (c) increase of the maximum number of samples from 32 (PointNET++) to 64 to select more neighborhood points within each center point (q_i). The selected PointNET++ SFR_3 architecture is presented in Fig. 4. The recall of PointNET++ SFR is 16% higher than that of the original PointNET++ version with increases in all the other performance metrics as well. The naming convention SFR stems from Smaller and Fewer neighborhoods, more points selected per neighborhood, which characterizes our PointNET++ version applied on industrial TLS data.

We further validate the optimal proposed network by quantifying the neighborhood information in a novel metric defined by the authors. This definition stems from the fact that the first sampling layer of PointNET++ is the one that captures the finest details in the point cloud. It is natural to assume that each feature produced by the first layer should depend only on points of one specific class, in order to be able to capture characteristic features of one particular class. If many of the neighborhoods processed by the first layer contain points of more than one class, one can expect a suboptimal learning performance. The primary reason for that is the network will be trying to learn some features that are not specific to one shape but to a combination of neighboring shapes which makes learning a harder task. Henceforth, we define a metric named neighborhood rate (Nh_{rate}) to account for neighborhoods that have 3D points belonging to more than one *CLOI* class:

Table 7. Network architecture parameters for PointNET++ SFR

SFR ID	Parameter	Scale 1	Scale 2	Scale 3	Scale 4
1	(a) search radius, r_i in (m)	0.025	0.05	0.1	0.2
	(b) number of neighbors, i	64	64	64	64
	(c) number of points to subsample, $N(q_i)$	1024	256	64	16
	(d) size of the MLP representing h	[32,32,64]	[64,64,128]	[128,128,256]	[256,256,512]
	(e) size of the MLP for extrapolation	[256,256]	[256,256]	[256,128]	[128,128,128]
ID	**Parameter**	**Scale 1**	**Scale 2**	**Scale 3**	**Scale 4**
2	(a) search radius, r_i in (m)	0.025	0.05	0.1	0.2
	(b) number of neighbors, i	64	64	64	64
	(c) number of points to subsample, $N(q_i)$	1024	256	64	16
	(d) size of the MLP representing h	[64,64,128]	[128,128,256]	[128,128,256]	[256,256,512]
	(e) size of the MLP for extrapolation	[128,64]	[256,256]	[256,256]	[256,256,256]
ID	**Parameter**	**Scale 1**	**Scale 2**	**Scale 3**	**Scale 4**
3	(a) search radius, r_i in (m)	0.05	0.1	0.2	0.4
	(b) number of neighbors, i	64	64	64	64
	(c) number of points to subsample, $N(q_i)$	512	128	32	8
	(d) size of the MLP representing h	[32,32,64]	[64,64,128]	[128,128,256]	[256,256,512]
	(e) size of the MLP for extrapolation	[256,256]	[256,256]	[256,128]	[128,128,128]
ID	**Parameter**	**Scale 1**	**Scale 2**	**Scale 3**	**Scale 4**
4	(a) search radius, r_i in (m)	0.05	0.1	0.2	0.4
	(b) number of neighbors, i	64	64	64	64
	(c) number of points to subsample, $N(q_i)$	512	128	32	8
	(d) size of the MLP representing h	[64,64,128]	[128,128,256]	[128,128,256]	[256,256,512]
	(e) size of the MLP for extrapolation	[256,256]	[256,256]	[256,128]	[128,128,128]
ID	**Parameter**	**Scale 1**	**Scale 2**	**Scale 3**	**Scale 4**
5	(a) search radius, r_i in (m)	0.1	0.2	0.4	0.8
	(b) number of neighbors, i	32	32	32	32
	(c) number of points to subsample, $N(q_i)$	1024	256	64	16
	(d) size of the MLP representing h	[32,32,64]	[64,64,128]	[128,128,256]	[256,256,512]
	(e) size of the MLP for extrapolation	[256,256]	[256,256]	[256,128]	[128,128,128]
ID	**Parameter**	**Scale 1**	**Scale 2**	**Scale 3**	**Scale 4**
6	(a) search radius, r_i in (m)	0.1	0.2	0.4	0.8
	(b) number of neighbors, i	64	64	64	64
	(c) number of points to subsample, $N(q_i)$	1024	256	64	16
	(d) size of the MLP representing h	[64,64,128]	[128,128,256]	[128,128,256]	[256,256,512]
	(e) size of the MLP for extrapolation	[256,256]	[256,256]	[256,128]	[128,128,128]

Table 8. Performance of class segmentation deep learning networks for the oil refinery dataset

Network	Accuracy (%)	Precision (%)	Recall (%)	mIoU (%)
SFR_1	69	49	56	37
SFR_2	70	49	54	36
SFR_3	**72**	50	**57**	**38**
SFR_4	71	51	56	38
SFR_5	67	56	52	34
SFR_6	70	51	51	36

$$Nh_{rate} = \frac{\#\ neighborhoods\ with\ \geq 2\ CLOI\ classes}{total\ \#\ neighborhoods} \tag{6}$$

where the number of neighborhoods in all blocks is defined as the number of blocks per point cloud dataset multiplied by the number of neighborhoods per block.

We present our results for the proposed SFR networks in Fig. 3. We observe that the Nh_{rate} is increasing with the neighborhood radius and the best performing SFR experiment is SFR_1, with $r = 0.025m$ and small increase in Nh_{rate} between $r = [0.025 - 0.05m]$ for all *CLOI* facilities. However, the increase in Nh_{rate} between $r = [0.025 - 0.05m]$ is smaller compared to the increase between $r = [0.05 - 0.1m]$. As such, we choose the SFR_3 network architecture which is the best performing network validated by our previous experiments. It is clear that PointNET++ SFR captures distinctive features of specific neighborhoods, since there are fewer classes in each neighborhood to associate features with specific classes. When there are many classes in one neighborhood, features are not distinctive of a particular class. In other words, the number of neighborhoods chosen in PointNET++ with points belonging to more than one classes is higher than those neighborhoods chosen in PointNET++ SFR. This means that we justifiably expect PointNET++ lagging in performance compared to our PointNET++ SFR.

We note that a two-fold reduction in the radius (r_i) leads to around 50% decrease in the number of neighborhoods (i) that contain multiple classes. Therefore, according to

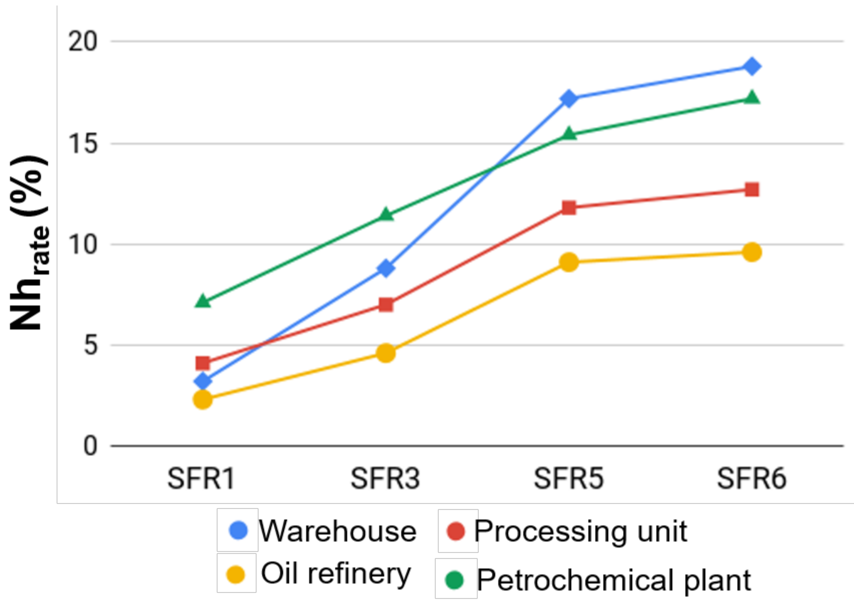

Fig. 3. Neighborhood rate comparison

intuition, this should improve the learning process. We present the selected PointNET++ SFR_3 architecture in Fig. 4. Step 2 partly answers the research question **RQ1** and **RQ2**.

The neighbourhood rate (Nh_{rate}) is also a measure of the occlusions a TLS dataset can have. This is attributed to the fact that if two points belonging to different $CLOI$ classes are closely located to each other, the more likely it is that these shapes are occluded. We then explore the impact of occlusions on performance with respect to the neighbourhood rate (Nh_{rate}). Fig. 5 shows the $mIoU$ performance of the selected PointNET++ SFR_3 network per $CLOI$ facility with respect to the Nh_{rate}. The results show that the higher the Nh_{rate} is, the smaller the $mIoU$ performance is. This highlights the impact of occlusion on performance. A more detailed analysis of this impact is not within the scope of this paper but is an interesting direction for further research.

3.5. Step 3: Contextual rule enforcement

The performance of PointNET++ SFR is still fairly satisfactory to answer the research questions **RQ1b**, **RQ2a** and **RQ2b**. We further refine the point label predictions from our PointNET++ SFR network using the following three stages of post-processing inference rules. More specifically, we propose the following techniques:

(a) The parameters of PointNET++ SFR are fine-tuned to capture local neighborhood

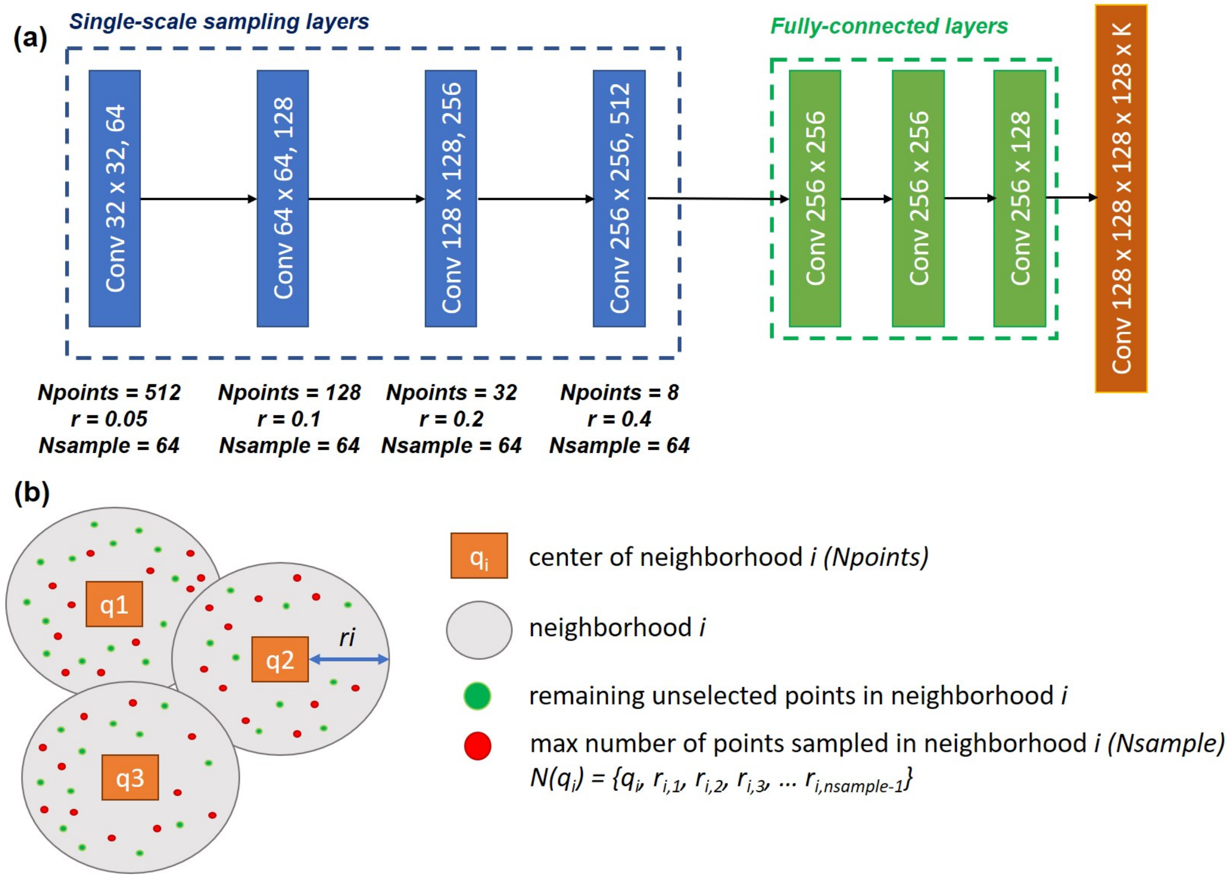

Fig. 4. (a) PointNET++ SFR architecture and (b) illustration of parameters used for each network layer at the block-level.

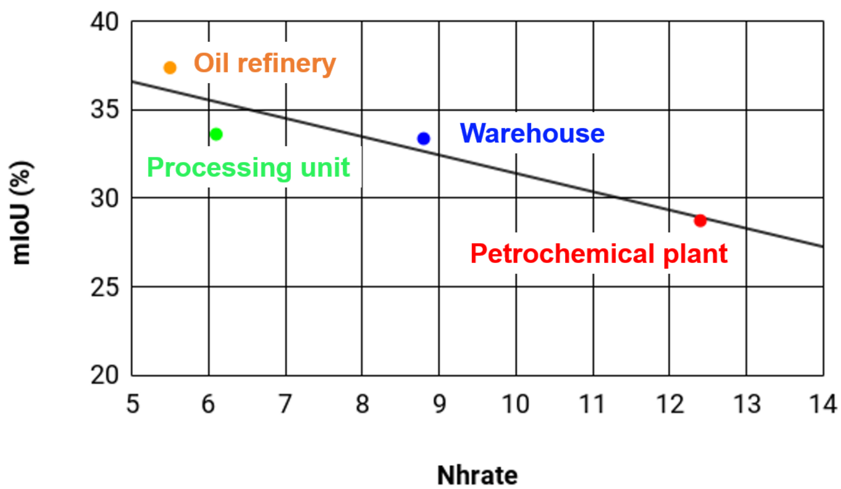

Fig. 5. PointNET++ SFR performance with respect to Nh_{rate}

information in dimensions of a cubic block. This limitation does not permit segmentation of shapes larger in dimensions than a cubic block. This affects mostly cylinders that can have diameters larger than $1m$. We segment cylinders with larger than $1m$ diameters by computing curvatures to answer the research question **RQ1b**.

(b) Other challenging shapes are secondary steel shapes (channels, angles and some I-beams). These shapes have low frequency of appearance in industrial facilities and prediction of their labels is a complex task. We follow an approach tailored to the peculiarities of these shapes for segmenting those shapes to answer the research question **RQ2a**.

(c) We observe that predictions with low confidence level given by the neural network have a much higher chance of being incorrect. We replace these low confidence level predictions of PointNET++ SFR with higher-confidence level predictions based on (b) and the confidence level of PointNET++ SFR predictions **RQ2b**.

3.5.1. Step 3a: Cylinder classifier

Large enough shapes (cylinders with diameter greater or equal to $1m$) are not captured as discussed earlier. Therefore, we develop a method to distinguish cylindrical shapes from other shapes in industrial settings focusing on a curvature-based analysis.

We compute the mean and Gaussian curvatures (principal curvatures) upon calculation of surface normals on each point (x_0, y_0). We first compute the surface normals at point (x_0, y_0) as follows. We define a $kDTree$ structure and find the closest points of (x_0, y_0) within a fixed distance (r). The definition of r is based on the bias-variance trade-off of noisy neighborhoods of points. Small radius results in high variance of the curve, whereas larger radius results in high bias. The choice of $r = 0.1m$ gave us a balance on the trade-off of bias-variance.

We then shift the center of mass of the closest points to the origin $(0, 0, 0)$ and compute the normal as the min eigenvector of the covariance matrix. We rotate the points so that the normal is on direction Z. We follow the approach by Har'el (1995) that locally approximates the surface of neighboring points by a quadratic polynomial, and computes the curvature of that surface. The approximate surface is given by:

$$z = ax + by + cxy + d + ex^2 + fy^2 \tag{7}$$

The principal curvatures are the eigenvalues of the paraboloid surface. Henceforth, the mean (H) and Gaussian (K) curvature of paraboloid surfaces at its vertex can then be approximated by finding the trace and determinant of an associated matrix.

We compute our principal curvatures as:

$$\text{principal curvatures} = \text{eigenvalues}\left(\frac{1}{1+f_x^2+f_y^2}\begin{pmatrix}2e & c\\ c & 2f\end{pmatrix}\right) \tag{8}$$

where

$$\begin{pmatrix}f_x\\ f_y\end{pmatrix} = \begin{pmatrix}a\\ b\end{pmatrix} + \begin{pmatrix}2e & c\\ c & 2f\end{pmatrix}\begin{pmatrix}x_0\\ y_0\end{pmatrix}$$

in order to calculate curvatures on point x_0, y_0.

To smoothen the curvature computations and remove outliers, we assign as the (gaussian or mean) curvature of each point the median of the respective curvatures of points within distance 0.2m. Then, our algorithm predicts a point as cylinder if the following three conditions ($G1 - G3$) are met:

G1. $K \leq 0.1m^{-1}$

G2. $H \geq 0.3m^{-1}$

G3. $H \leq 3m^{-1}$

These parameters are experimentally verified on the *CLOI* dataset. For an ideal cylinder, $K = 0$ and the mean curvature is $H = 1/D$, where D is the diameter of the cylinder.

3.5.2. Step 3b: Steel shape Segmentation

We develop a procedure to segment channels, angles and I-beams based on corner detection. Our hypothesis is that these structural steel sections are composed of two perpendicular planes (two sets of perpendicular planes for the case of I-beams and channels) and we need to automatically detect these two planes. We consider channels as being composed of two L-shape corners, whereas I-beams as having two T-shape corners. We first compute the normals of each point as the minimum eigenvector of the covariance matrix of the neighborhood of each point at a fixed radius of $0.3m$ based on the mean dimensions defined in British Steel Standards (BS EN 10365:2017, 2017) for these shapes. These normals are unoriented, as such we correct their orientation. We do so by picking a viewpoint, which is a vector v, such that when the inner product of the viewpoint and our normal is positive we keep the same orientation. Otherwise, when $\langle n, v\rangle < 0$, we change the direction of the normal vector n. We then define neighborhoods with fixed radius of $0.5m$ to check whether they are corners. For each neighborhood, we cluster the normals based on k-means clustering, where in our case $k = 2$ for two types of normals. We enforce that the angle between the two planes should be $> 60°$ instead of $= 90°$ as being angle edges due to noise. After finding the planes, we fit rectangles by assigning the points to either plane.

An illustration of the L- and T-shapes with the directions of their normal vectors (n_1 and n_2) is given in Fig. 6 and an L-shape and T-shape type distinction based on geometric parameters is presented in Fig. 7.

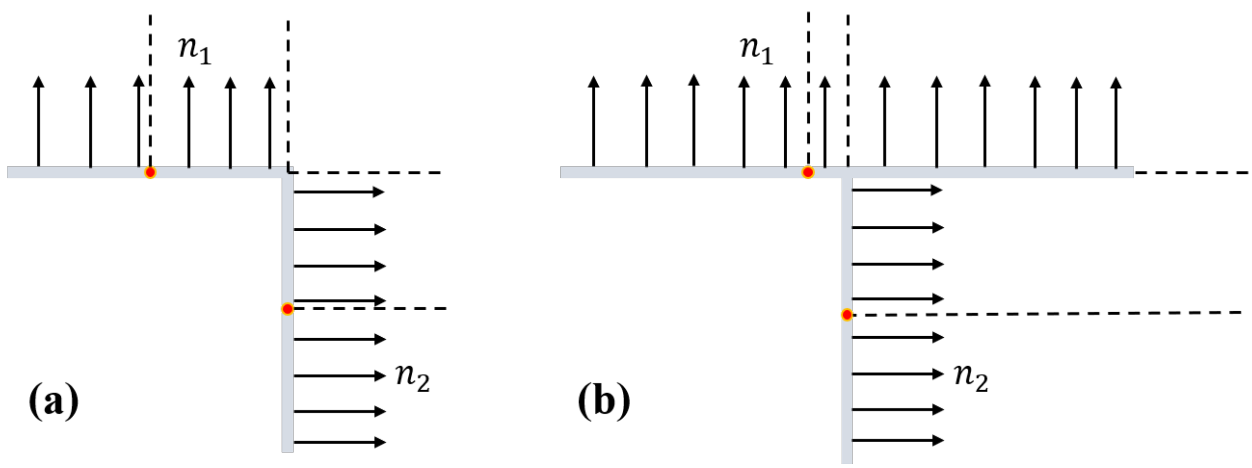

Fig. 6. Illustrative representation of normals in (a) L-shape and (b) T-shape structural steel profiles.

The condition for a structural steel profile to have an L-shape is that

$$\frac{r_1}{S_1} \approx \frac{r_2}{S_2} \tag{9}$$

whereas the condition for a structural steel profile to have a T-shape is that

$$\frac{r_1}{S_1} \ll \frac{r_2}{S_2} \tag{10}$$

where r_1, r_2 are the distances from the center of each rectangle to the intersection of the planes and S_1, S_2 are the lengths of each rectangle of the L-, T-shape respectively.

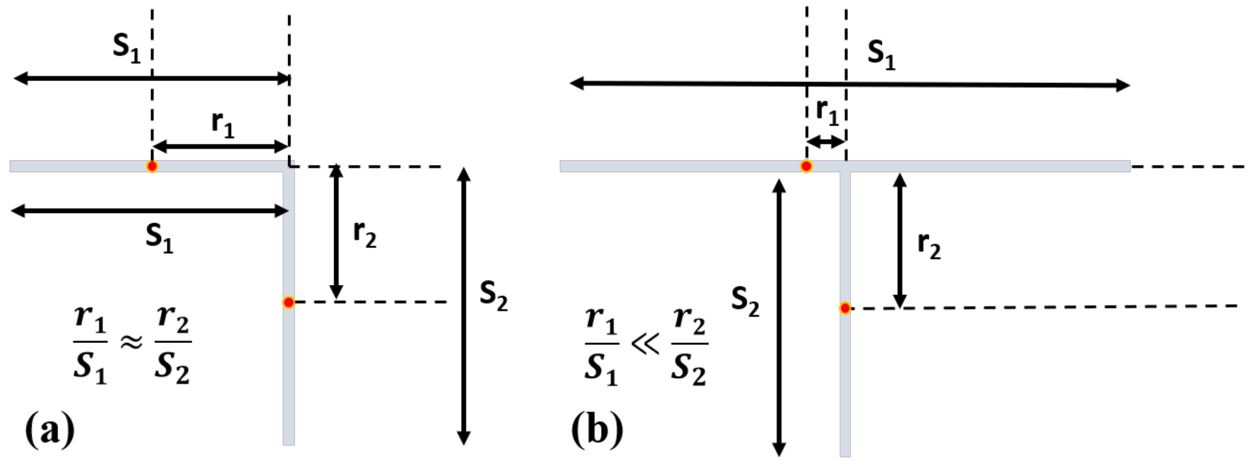

Fig. 7. Illustration of (a) L-shape and (b) T-shape structural steel profiles.

The normals of angles and channels are clustered in the same category given that they have the same L-shape. In contrast, I-beams form a T-shape.

3.5.3. Step 3c: Confidence level adaptation

The next step in our CLOI-NET methodology acknowledges the fact that low confidence predictions by the PointNET++ SFR network (Section 3.4) are more likely to be erroneous than high confidence ones. Therefore it is legitimate to determine a methodology to improve the predictions of the classes that are misclassified, and prioritizing those with low confidence. We present in Fig. 8 the confusion matrices of precision and recall for all the eight *CLOI* classes. **We used heatmaps to show the precision and recall metrics in the confusion matrices. Dark blue colors indicate high precision/recall, whereas yellowish colors represent smaller precision/recall values.** We show that angle and channel points are misclassified as "other" (17% and 30% probability respectively). Predicted "channels" are actually "other" (55% probability) or "angles" (9.7% probability). There is a similar trend for predicted "angles" being "channels" (6.4% probability), "cylinders" (14% probability) or "other" (36% probability). Our PointNET++ SFR network also confuses I-beam points with angle (23% probability) and channel points (34% probability). We develop the following two-step method to correct these misclassifications:

- *Step 1*: If a point is predicted as "channel" and is not close ($> 0.1m$) to either an L-shape or a T-shape corner, we convert its label to "other". Definitions and determination of whether a point is close to an L-shape are provided in Section 3.5.2.

- *Step 2*: If a point is predicted as "I-beam" with low confidence (<80%) and it is close to an L-shape corner, we classify it as "channel".

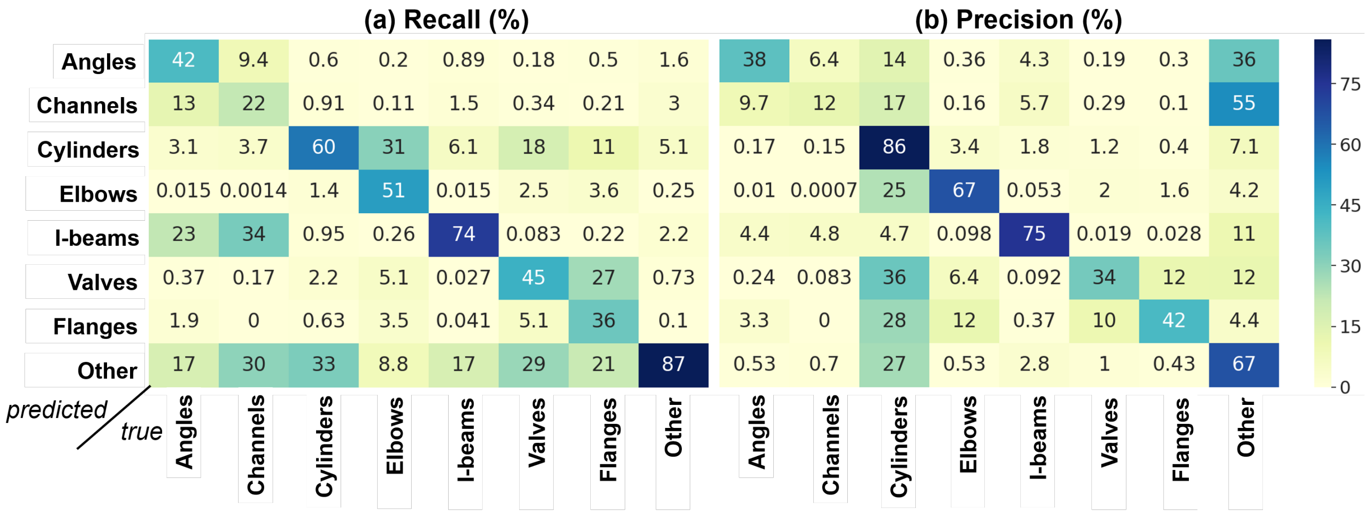

Fig. 8. Confusion matrices with (a) recall and (b) precision of network trained on PointNET++ SFR_3 for the oil refinery dataset.

Valves and flanges are also classes that are more often misclassified, however some of these predictions cannot be corrected since they stem from ground truth errors. Some "cylinders" and "valves" are also predicted as "other", however we cannot revert these cases since these can be parts of equipment with similar shapes and consistent geometric rules cannot be generated to correct these misclassifications. Angles are misclassified as "I-beams" (23% probability) and "Other" (17% probability). However their performance was fairly satisfactory and their frequency is relatively low (2%) compared to the other *CLOI* classes (Agapaki et al., 2018). Therefore, our CLOI-NET method on Step 3c focuses on improving the predictions of channels.

3.6. Step 2b: Annotation Cost Optimization

Our pipeline so far takes as input facilities that are annotated for training and first performs the test on PointNET++ SFR with the unlabeled facility and secondly post-processes these results using shape-specific rules for fine-grained per point segmentation. In this section we make the observation that the complex and noisy nature of the class segmentation problem for industrial data ensures that any algorithm will be approximate, and thus some of its predictions will be erroneous. For the process of generating IgDTs, however, these errors will, inevitably have to be manually corrected. Therefore any practical analysis of Digital Twin generation should focus on minimizing the *manual annotation cost* to address the research question **RQ3**. Indeed, in this section we use a simple model for the annotation time to demonstrate that manual *pre-annotation* of parts of the test dataset can greatly improve the accuracy of the predictions, thus significantly minimizing manual annotation time. We propose a two-stage annotation procedure with the goal to minimize the annotation cost, should the user choose the option to manually annotate part of the test facility. Our motivation stems from a recent research area called "active learning". Researchers use active learning for image annotation (Jain and Grauman, 2016, Mahapatra et al., 2018) and exploit the most valuable images to manually annotate and then include them in the training set. As such, we follow a two-step procedure: (a) we apply the PointNET++ SFR training model that has no annotated windows from the test facility and post-process the test windows and (b) we manually annotate an x fraction of the windows from the test facility using the predictions of PointNET++ SFR to help us during annotation. We assume that this annotation step is performed using any manual annotation tool, i.e. CloudCompare (Cloudcompare, 2016) or the LFM Software (AVEVA, 2019). We then apply the PointNET++ SFR model with the manually annotated windows during training and post-process the remaining test windows. We denote the approach described in Section 3.4 a passive learning approach, since no data from the test facility is included while training. A comparison of the steps followed for the active and passive learning approach is presented in Fig. 9. For the application of the active learning procedure on the pipeline of PointNET++ SFR, we parse the 3D TLS data in disjoint "windows" and then slice the facility into smaller pieces that will then further subdivide into cubic blocks for further processing. Therefore, we enforce uniqueness of the 3D blocks during training and testing splits.

We introduce a simple model in which we use the percentage of incorrect predictions ($1 -$ accuracy) as a proxy for the manual annotation time. Our assumption is based on similar work conducted for active learning on clinical concept extraction in medical tasks (Kholghi et al., 2017). Intuitively, we assume that the manual annotation time is proportional to the percentage of incorrect predictions. However, our analysis is agnostic of the actual evaluation metric used and could have been carried in terms of other metrics, e.g. Precision, Recall, mIoU.

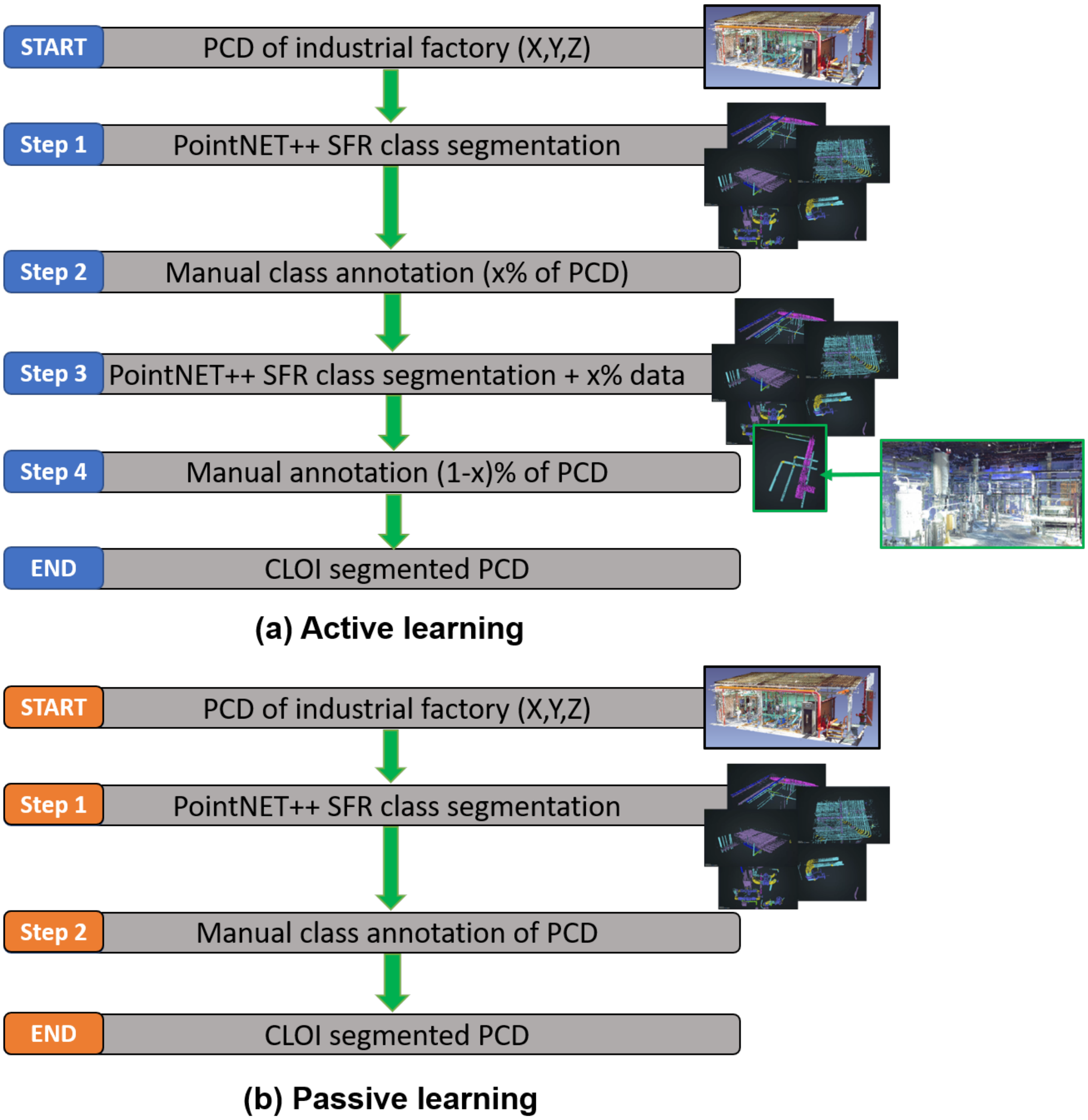

Fig. 9. (a) Active and (b) passive learning methodologies.

We define the total annotation cost $c(x)$ as the cost of the two annotation phases separately, which are Steps 2 and 4 in Fig. 9(a). We propose that $c(x)$ is a function of the fraction x of test windows that were manually pre-annotated. We assume that the annotation cost: (1) is proportional to the time to manually annotate the points of the whole facility, and (2) is proportional to the fraction of incorrectly classified points $(1 - \text{accuracy})$. So, we define the total annotation cost $c(x)$ as:

$$c(x) = f(x) + g(x) \tag{11}$$

where

$$f(x) = \lambda x(1 - a(0)) \tag{12}$$

$$g(x) = \lambda(1 - x)(1 - a(x)) \tag{13}$$

and $a(x) : [0, 1] \rightarrow \mathbb{R}$ is the accuracy on the remaining point cloud data after training on x fraction of the test set, λ is the time to manually annotate the entire point cloud facility, $a(0)$ is the accuracy of the first annotation phase with $x = 0$ annotated windows, $f(x)$ is the time to pre-annotate x fraction and $g(x)$ is the time to annotate $1 - x$ fraction after active learning is performed on our data.

We then determine the optimal amount of data that need to be annotated to minimize this cost ($c(x)$). We make two further natural assumptions: (a) we assume that as the pre-annotated data of the test facility increases, the training accuracy increases, or equivalently that $a'(x) \geq 0$, and (b) we assume that the accuracy learning curve is concave, i.e. $a''(x) \leq 0$. We base the latter assumption on recent active learning experiments (Jain and Grauman, 2016). This means that the more data we provide for training, the rate of accuracy increase decreases.

We inspect its first and second derivatives of the cost function $c(x)$. We have:

$$c'(x) = -\lambda a(0) + \lambda a(x) - \lambda(1 - x)a'(x) \tag{14}$$

and:

$$c''(x) = \lambda(2a'(x) - (1 - x)a''(x)) \tag{15}$$

where $a''(x) \leq 0$ from assumption (b) and $a'(x) \geq 0$ from assumption (a), which means that $c''(x) \geq 0$ i.e. the cost function $c(x)$ is convex. In other words, $c(x)$ only has one global minimum that we find by setting:

$$-a(0) + a(x) - (1 - x)a'(x) = 0 \tag{16}$$

where x is the annotation percentage that minimizes the total cost.

We first prove that the optimal manual annotation percentage is always at most 50%. According to the mean value theorem, there exists an annotation percentage ξ that:

$$a(x) - a(0) = a'(\xi)x \tag{17}$$

where $0 \leq \xi \leq x$. As $a(x)$ is a concave function (as we increase x, accuracy increases at a slower rate), we have:

$$a'(\xi) \geq a'(x) \tag{18}$$

Applying Eq. 18 to Eq. 17, we get:

$$a(x) - a(0) \geq a'(x)x \tag{19}$$

Combining Eq. 16 with Eq. 19, we find the following equation for the maximum pre-annotation percentage:

$$a'(x)(2x - 1) \leq 0 \tag{20}$$

Given that $a'(x) \geq 0$, we have $x \leq 0.5$. This means that it is never advantageous to pre-annotate more than 50% of the TLS data of a facility. A qualitative illustration of the accuracy and annotation cost curves with respect to the annotation percentage used for active learning is presented in Fig. 10. We demonstrate that the better the quality of learning is, the less the annotation cost, hence the manual pre-annotation percentage x of a test facility needed for training is smaller. In order words, the higher the accuracy curve is (optimal annotation percentage x to the top left of the plot), we achieve better quality and faster learning for the same pre-annotation percentage x. It is important to note that the annotation cost in Fig. 10(b) is the cost after applying the active learning approach as a percentage of the total manual annotation cost of the passive learning approach.

We measure the success of our pipeline not by maximizing the point-wise accuracy of our method, rather by minimizing the cost that it incurs to the modelers when using it. Our

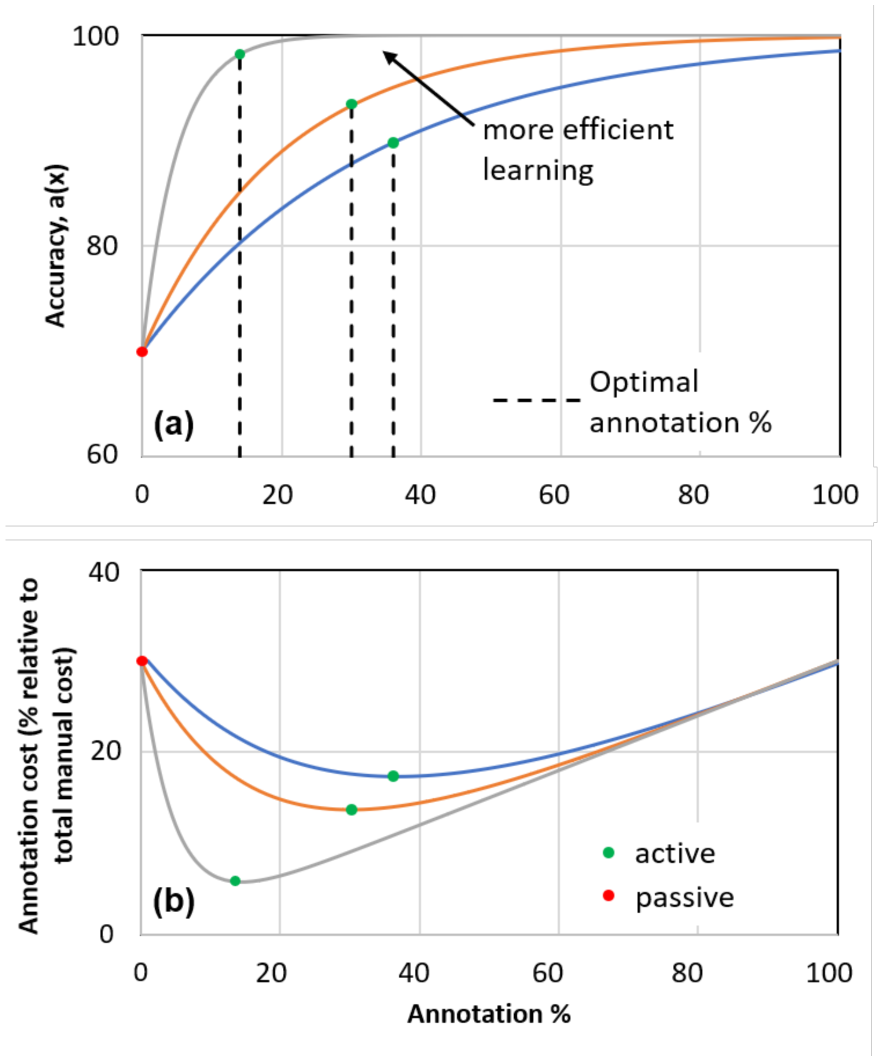

Fig. 10. (a) Accuracy of training and (b) annotation cost with respect to the pre-annotated percentage (%) of a facility used for training.

novel method leverages the advances in point cloud deep learning segmentation, contextual shape specific attributes and active learning in order to accurately predict point-wise class labels with no significant difference in performance for different industrial environments. A critical part of our method's novel design is the stage-wise annotation, which permits both human-annotated and automatically annotated points to influence the system's view of what needs the most human attention next.

The **hypothesis** of this paper is that class segmentation is (i) efficient and reliable, (ii) scalable when exploiting deep learning methods in a sensible manner tailored to industrial spaces and (iii) there is no significant bias in the segmentation performance for different industrial facilities. We evaluate our hypothesis experimentally.

4. RESEARCH METHODOLOGY

The research design that we followed was to validate each process outlined in Section 3. Therefore, we propose the following research activities to validate the automated segmentation of class point clusters:

1. generation of the *CLOI* benchmark dataset class labels (Section 4.2) and data preparation (Section 4.3.1) in order to run the training experiments of the CLOI-NET proposed solution,
2. implementation of the CLOI-NET proposed solution (Section 4.3.2) and
3. measuring the class segmentation performance to validate the hypothesis (Section 4.4).

Then, Section 4.5 follows with a discussion of the performance of the CLOI-NET class segmentation solution in two levels: (a) overall performance (**Section 4.5.1**) and (b) class component performance (Section 4.5.2). For the overall performance of the CLOI-NET methodology, we first investigated the robustness of the proposed methodology by determining the facility bias. Then, we measured the cost savings by implementing the proposed CLOI-NET active learning approach. The second part of Section 4.5 focuses on the discussion of the proposed method's performance on class component level.

We first state the assumptions of our research methodology in the section that follows.

4.1. Assumptions

A *CLOI* facility satisfies the following conditions:

C1. Cylindrical shapes are grouped into one *CLOI* class. This category includes the most important cylindrical shapes from the piping, electrical and structural categories.

C2. Cylinders have diverse sizes.

C3. The number of points of the TLS scanned datasets is much larger than the number of points per cylinder.

C4. Noise and clutter outside the industrial facilities is properly removed manually. For instance, the removed points can be either vegetation or irrelevant points for outdoor facilities or points reflected outside the scanned area of indoor facilities.

According to the PointNET++ Qi et al. (2017a) implementation and standards of industrial facilities (Agapaki and Brilakis, 2017, BS EN 10365:2017, 2017), we assume the proposed CLOI-NET class segmentation method is feasible in the context of either indoor or outdoor TLS scanned industrial factories under the following conditions, which are also confirmed by our experiments:

A1. The registration quality of the TLS industrial data is performed in commercial software and is not part of the research methodology of this paper. In other words, it is assumed to be of high quality to conduct post-processing, since data is collected from professional laser surveys. Specifications of the laser surveys are given in Table 9.

A2. The proposed framework is independent of the laser scanner surveying parameters.

A3. The PointNET++ SFR network learns point features in cubic meter 3D blocks following the initial PointNET++ implementation Qi et al. (2017b).

A4a. The PointNET++ SFR sampling layer generates neighborhoods around point centers with the condition that the number of points belonging to different *CLOI* classes in these neighborhoods is minimized.

A4b. The PointNET++ SFR sampling layer parameters are optimized based on the network's performance.

A5. Cylinders with diameters greater than $1m$ cannot be classified in our PointNET++ SFR network.

A6. The PointNET++ SFR confidence level of a *CLOI* class prediction is positively correlated with the prediction that this point is correct.

A7. The performance of individual *CLOI* classes in the PointNET++ SFR is dependent on the prior distribution of *CLOI* classes. Dominant classes are expected to have higher prediction rates.

A8a. The user annotation time during the active learning methodology is a fraction of the incorrect predictions of the PointNET++ SFR network.

A8b. The pre-annotation cost in the active learning procedure is proportional to the time to manually pre-annotate.

A8c. The performance of training a class segmentation framework improves the more pre-annotated data of a test facility one uses during training.

In particular, A1-A7 are validated experimentally in Section 4.4 whereas A8a, A8b and A8c are validated in Section 4.5.1.

4.2. Ground Truth Data

To test our **hypothesis**, we generate the first dataset of class labeled point clusters of industrial facilities named *CLOI* (Agapaki et al., 2019). *CLOI* consists of 10 classes that cover a wide range of industrial scenes (both indoor and outdoor). We use the TLS datasets of four laser scanned industrial facilities for the generation of *CLOI* as shown in Fig. 11. One facility is a warehouse, one is a petrochemical plant, one an oil refinery and the fourth a processing unit. These facilities are anonymized since rights are reserved by AVEVA Group Plc. and British Petroleum. All datasets are obtained using static terrestrial laser scanners. We provide the (to the best of our knowledge) hitherto largest collection of terrestrial laser scans of industrial facilities with point-level (a) class and (b) instance ground truth annotations. (A) refers to one of the ten *CLOI* classes and (b) is an index number that refers to a specific individual shape and is not further used in this work. In total, it consists of 12,497 shapes and 4.3 billion points with their class and instance labels for each point. To this end, this paper provides *CLOI*, the largest annotated dataset based on already existing datasets presented in Table 2 and the only dataset of industrial environments that is captured with more than one sensor. This means that processing *CLOI* point cloud data is independent of the data capturing system that was used to generate the data. *CLOI* is also the only dataset available for processing industrial environments. Below we investigate the metadata of *CLOI*; the frequency of appearance of each class and the scanner specifications of each TLS dataset.

The frequency of appearance of each class across the four industrial facilities is shown in Fig. 12. We observe that there is variation in the frequency of appearance of channels and cylinders ($\sim 10-25\%$) across the four *CLOI* facilities. This is attributed to the specific use of each industrial plant.

We acquire each dataset with the scanner specifications demonstrated in Table 9. There is variability in the linearity error of the TLS scanners used to scan the *CLOI* facilities. Each facility was scanned with a different TLS scanner and the oil refinery facility was surveyed with the most accurate scanner that is designed to operate in industrial environments (Surphaser, 2015). The petrochemical plant was surveyed in greyscale.

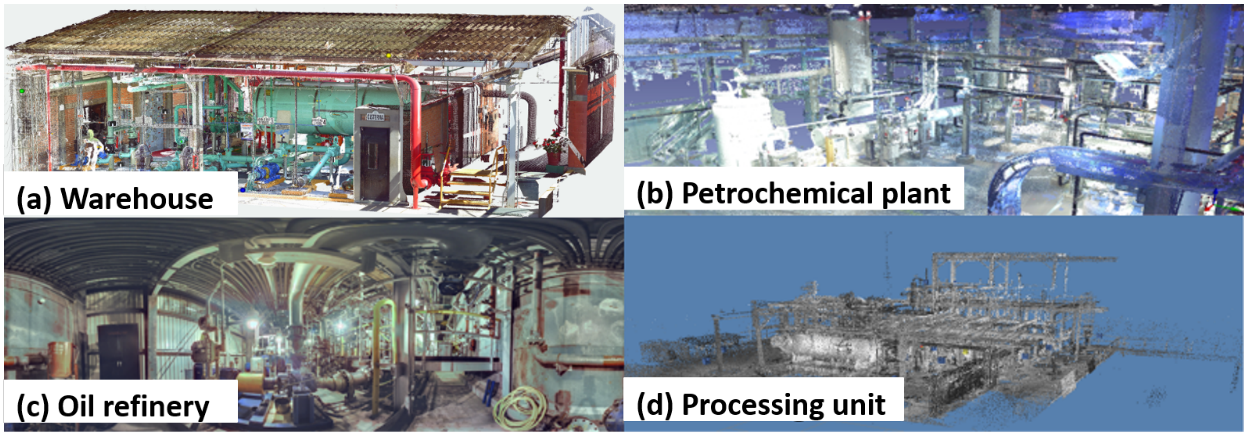

Fig. 11. (a) warehouse, (b) petrochemical plant, (c) oil refinery and (d) processing unit.

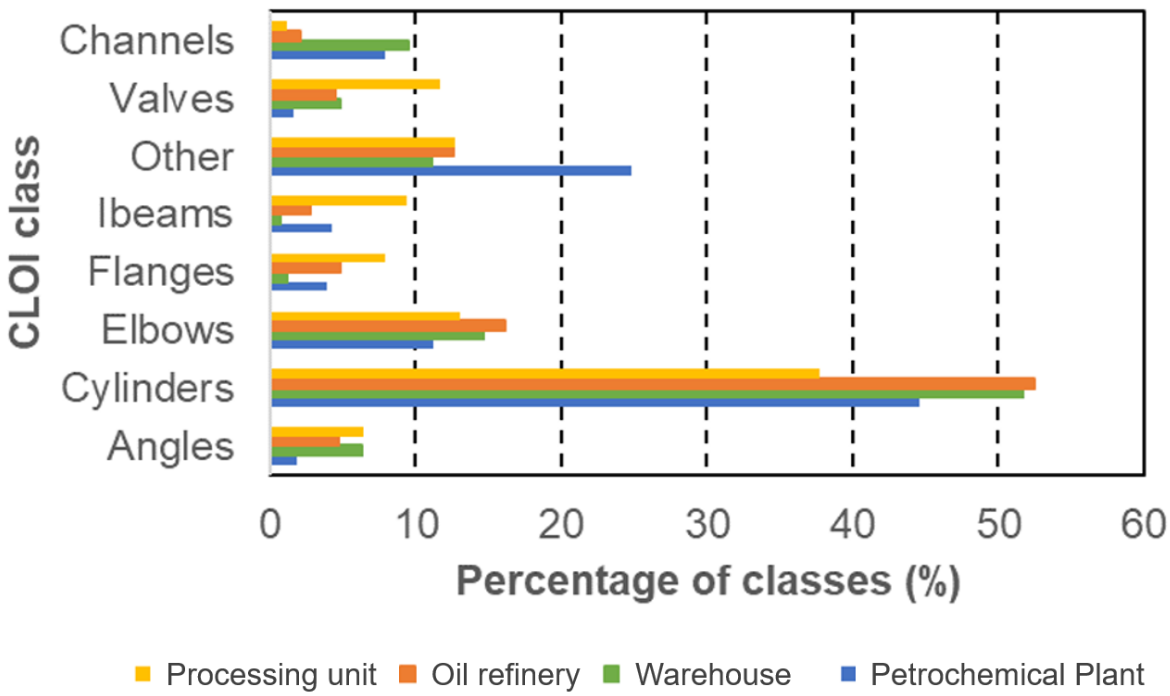

Fig. 12. Frequency of appearance of the *CLOI* labeled classes.

1070 We provide the definitions of the *CLOI* shapes below. We define angles as roll-formed steel angles that have an L-shape. The legs of the L-shape have equal or unequal length. Channels refer mostly to steel beams with a C-shape. Cylinders include the following three sub-categories: (a) circular hollow sections, which refer to cylinders that support pipes, cylindrical structural columns and handrails, (b) conduits that refer to the tubes that protect

Table 9. Metadata of *CLOI* dataset

Metadata	Warehouse	Oil refinery	Petrochemical plant	Processing Unit
Scans	10	57	44	27
Original size	74,264,368	2,911,602,008	346,748,967	340,349,857
Scanner	FARO X330	Surphaser 105HSX	Z+F Imager 5010C	Z+F Imager 5003 scanner
Resolution range @10m	0.3mm	0.21mm	0.1mm	1.6mm
Vertical resolution	0.009°	0.0003°	0.001°	0.018°
Measurement range	±5°	±0.004°	±0.5°	-
Linearity error	±2mm	<0.7mm	$\leq$ 1mm	±3mm

electric wiring and (c) pipes which are tubes that carry fluids and gases. Elbows are tubes that connect piping elements or conduits. Flanges refer to plates or rings at the end of pipes. We define I-beams as the structural steel beams that have an I-shape. Valves refer to all the devices that control the flow of liquids through the pipelines. We cover all types of valves across our datasets (globe, ball, gate, butterfly, diaphragm, plug, check, needle, pinch valve). "Other" refers to any other point clusters that do not belong to the above-mentioned classes. It is important to note that *CLOI* classes are more fine-grained and challenging to distinguish than many of the existing indoor and outdoor segmentation datasets (Armeni et al., 2016, Roynard et al., 2018, Hackel et al., 2017).

The first step in our pipeline is to prepare and register the laser scanned point clouds, so that we can annotate them in the commercial manual labeling platform for industrial plants, LFM (AVEVA, 2019). The readers can refer to Agapaki et al. (2019) for details on the *CLOI* dataset generation.

4.3. Experiments

4.3.1. Data preparation

We subdivide each facility in overlapping 3D cubic blocks as explained in Section 3.3. We show examples of regions from *CLOI* facilities and an illustration of slicing a facility into 3D cubic blocks in Fig. 13. These examples are taken from our *CLOI* oil refinery facility. We use 0.5m stride to overlap the 3D cubic blocks as proposed by Qi et al. (2017b). We use the Farthest Point Sampling (FPS) technique to sub-sample points (Qi et al., 2017a) within these 3D cubic blocks. This technique is used for density-invariant subsampling and leads to a more uniformly distributed point cloud. We start with an empty set of points S and we progressively add a point x such that x has the maximum distance from the points in S. The distance between S and x is defined as $\min_{S_i \in S} d(x, S_i)$. We use this sampling method, since it covers the entire point cloud as opposed to random sampling that can be restricted to dense parts of it with the same number of output points. At training time, we sample 4096 points in each block on-the-fly. At test time, we test on all the points of a cubic block. The cubic blocks are then shifted to the global axis origin $[0, 0, 0]$ and aligned to the principal global coordinate system axes both for training and testing.

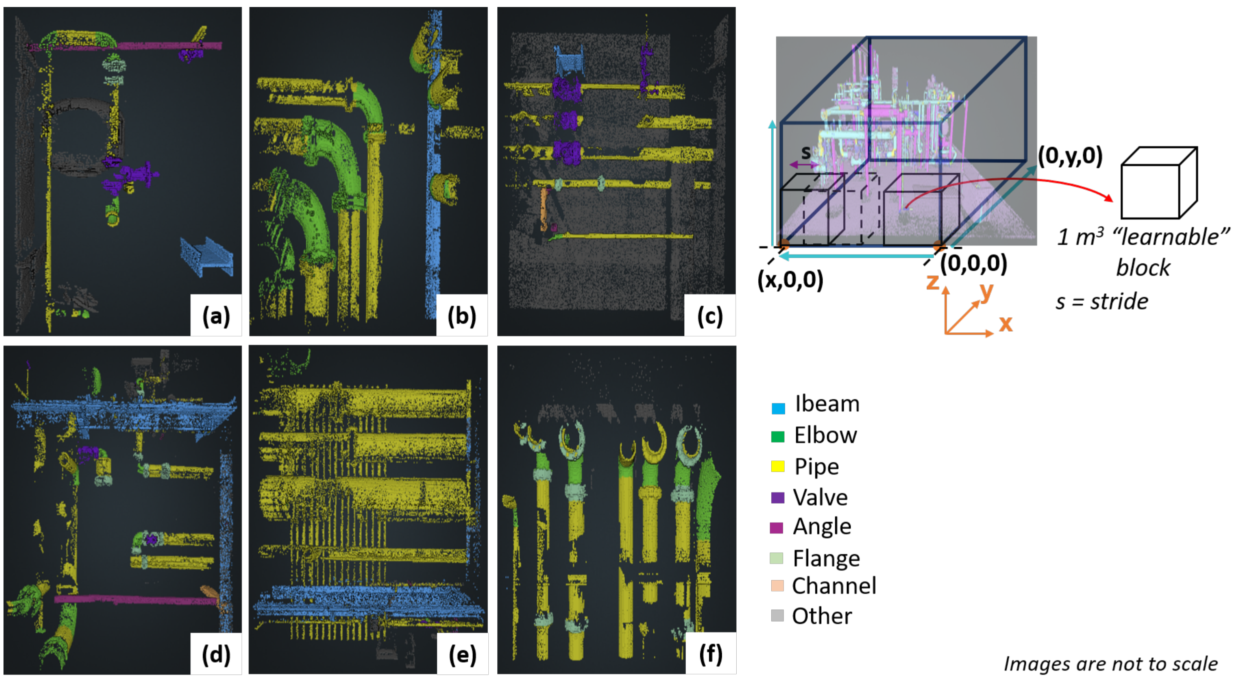

Fig. 13. 3D block generation examples from the oil refinery facility.

4.3.2. Implementation

We implement our solution on Tensorflow 2.0 as a proof of concept and execute our experiments on Google Cloud (Deep Learning VM image) with NVIDIA Tesla P100 GPUs. Visualizations of our point clouds and segmentation results are implemented on Potree Viewer (http://potree.org/) in JavaScript, which is built upon ThreeJS. A pre-trained model on industrial facility point cloud data does not exist. As such, we train the network from scratch for 250 epochs or about 100 000 steps using a batch size of 24, an initial learning rate of 0.01 and a learning rate decay factor of 0.5. We choose this set of hyperparameters so that the loss function converges during training as proposed by PointNET++. We experimentally identify the optimized network parameters of PointNET++ SFR. We use a randomized parameter search with a fixed list of options for each network parameter, dropout and learning rate as listed in Table 6. We conduct experiments to optimize the dropout rate (g) of the last training layer ranging from 0.2 to 0.6 and the learning rate (h) from 0.1 to 0.001, based on the parameters used in Qi et al. (2017a). We find the optimal dropout rate being 0.3, since our experiments suggest that overfitting is ameliorated by doing so. Fig. 4 shows the PointNET++ SFR deep segmentation network that we implement. We adjust the training size for each combination by observing saturation in the training accuracy. We do not further investigate other network hyper-parameter sets as these do have minor impact on the training quality for the scope of this work. They rather control the stability and speed of convergence of the loss function. The training time takes $12 - 15h$ to converge on average with the proposed configuration.

Fig. 12 shows that *CLOI* is an imbalanced dataset. Many learning algorithms have a trend to bias the majority class for imbalanced datasets due to the objective of error minimization (Hanley and McNeil, 1982). Henceforth, we assess the effectiveness of our CLOI-NET methodology in terms of the discrimination measure **Area Under the Curve (AUC)** which is equivalent to the Wilcoxon test in ranking classifiers. The AUC metric was first used by Hanley and McNeil (1982) in diagnostic radiology and later used in validating machine learning algorithms (Bradley, 1997). This metric is defined as the Area Under the Receiver Operating Characteristic (ROC) curve which shows the trade-off between recall (or True Positive Rate - TPR) and False Positive Rate (FPR) as defined below:

$$FPR_c = \frac{FP_c}{FP_c + TN_c} \tag{21}$$

The TPR_c is also known as sensitivity of classification and measures the probability of correct prediction of points of a *CLOI* class c, whereas the FPR_c is known as the probability of false alarm and measures the probability of incorrect predictions among all the points that belong to all other classes other than c. The AUC_c is then given by (Powers, 2011):

$$AUC_c = \frac{TPR_c + TNR_c}{2} \tag{22}$$

where $TNR_c = 1 - FPR_c$.

The AUC_c metric is ideal for predicting probabilities of classes that have a small number of points in the *CLOI* datasets which is an issue that has similarly been tackled in 3D indoor spaces (Armeni et al., 2016) and medical imaging applications due to small and heterogeneous datasets (Hanley and McNeil, 1982).

4.4. Evaluation

We evaluate the CLOI-NET proposed method of our prototype on the optimal hyper-parameters identified in Sections 3.4-3.6. We first evaluate the output of the PointNET++ SFR network. This is a prediction of the class label of each 3D point with a confidence score. This score is interpreted as the likelihood of a 3D point to belong to one of the eight *CLOI* classes. We compare predicted and ground truth labels pointwise and evaluate accuracy, precision, recall and Intersection-over-Union (IoU) scores. We first use the overall accuracy for comparing our sets of experiments. However, since this metric is biased towards dominant classes (classes having a large number of TLS data points), we then use precision, recall and IoU for individual class evaluations. We evaluate our PointNET++ SFR proposed solution on each *CLOI* facility and the details of our experiments are illustrated in Table 10. We train on pre-annotated (i) single facilities, (ii) "all" and (iii) "all but test" *CLOI* facilities. For (i), we test on (a) either the same facility that PointNET++ SFR was trained on or (b) any other *CLOI* facility. We use a k-fold validation strategy such that each facility is a single fold. As such, the training models do not see any part of the test facility.

More specifically, we show that when the same training facility is used for testing (experiments (ia) or (iia)), the test accuracy increases since it is easier for the PointNET++ SFR network to learn from data of a trained facility. It is important to note that in these cases we do not use the same 3D blocks for the training and test experiments, the 3D blocks are completely disjoint. The accuracy of the (ia) experiments is marked in bold and is the maximum per row and column in Table 10. However, for all the *CLOI* facilities, annotating data of the same facility and training on those (experiments (ia)) does not contribute to significantly higher accuracy than learning from annotated data of other *CLOI* facilities (experiments (iia)). For example, the evaluation accuracy when training on the warehouse alone (80% of its data for training) is 84.65%, and the same metric when training on all *CLOI* facilities including 80% of the warehouse data is slightly smaller (79.9%). This means that including more data from other *CLOI* facilities for training does not necessarily assist the learning algorithms and is an indicator of differences between facilities. We further investigate factors for facility differentiation in Section 4.5.1. We also demonstrate that when a single *CLOI* trained facility is not the same as the one used for testing (experiments (ib)), performance is relatively low. This is another case indicating bias between our facilities. As such, we conduct experiments with all the *CLOI* facilities for training except one used for testing ("all but test" - experiments (iiib)). We observe that the petrochemical plant and the processing unit perform better when the former is used for training and the latter for testing (72.7% test accuracy) in comparison to 61.85% when "all but test" *CLOI* facilities are trained. We attribute this to a greater similarity of these facilities and as such we further investigate facility bias in Section 4.5.1. We observe a similar trend between the petrochemical plant and the warehouse respectively. We also conduct experiments with "all" the facility data during training (experiments (ii)). "All" facility data corresponds to an active learning approach where 80% of all *CLOI* facilities (including the test facility) are trained. With this experiment, we show that increasing the amount of training data results in higher accuracy (from 66% to 82% on average). This means that if modelers are willing to annotate 80% of the test facility, this will only increase the validation accuracy by $15 \pm 5\%$. A more detailed analysis on the optimal annotation percentage of the test facility that we include while training follows in Section 4.5.1. If we do not include any data from the test facility for the evaluation ("all but test"), then the accuracy is higher than training on a single *CLOI* facility (experiments (ib)). This indicates that more data during training does improve performance but it should be properly selected. Pre-annotating data from the test facility (experiments (ia)) or training on a single *CLOI* facility other than the test facility (experiments (ib)) is time-consuming to annotate for the performance gain that is achieved. Therefore, the results of the "all but test" experiment are used for further processing.

We also investigate the confidence level of the predictions of our PointNET++ SFR network. Fig. 14 shows the percentage of correctly predicted points per facility (accuracy) with respect to the confidence level of the predictions. We observe that for the oil refinery 62% of the points are correctly predicted with confidence level 80% and above. Similarly, 63.4%, 66.7% and 58% of points are correctly predicted with confidence level 80% and above for the warehouse, petrochemical plant and processing unit respectively. Therefore, there is a positive correlation between the the correctly classified points and the confidence level

Table 10. Evaluation accuracy (%)

Training Facility \ Test Facility	Warehouse	Oil Refinery	Processing unit	Petrochemical plant
Warehouse	**84.65**	55.43	47.17	73.33
Oil Refinery	51.54	**92.97**	59.79	62.85
Processing Unit	50.16	58.13	**76.27**	56.44
Petrochemical Plant	60	56.25	72.67	**90.28**
all	79.9	85.73	72.7	89.1
all but test	64.1	68.61	61.85	70

of the predictions for all four *CLOI* facilities. In other words, the higher the confidence level, the more points are correctly predicted. This is an indication that our PointNet++ SFR network outputs correct *CLOI* class labels with high confidence, whereas the outputs of incorrect *CLOI* class labels are given with low confidence. Therefore, we further post-process the incorrect labels that have low confidence to improve the class segmentation performance of our method.

We highlight three main pitfalls of our PointNET++ SFR network that account for misclassified points based on these experiments: (a) shapes with volume larger than a cubic meter cannot be efficiently captured, (b) classes of imbalanced datasets are penalized and (c) the confidence level of predictions is not propagated while learning neighboring geometry. As such, we further investigate the efficiency performance for each *CLOI* class addressing each pitfall in Table 11. We demonstrate that cylinder prediction adaptation increases recall by 14.75% on average, meaning that points belonging to cylinders with large radius are now correctly predicted. Cylinder class adaptation penalizes precision by 2.5%, however the IoU which combines precision and recall is improved by 3.5% on average. Steel shape adaptations improve the performance metrics of channels (highlighted in Table 11). Our CLOI-NET predictions are significantly improved by implementing the active learning approach and confidence level adaptations for cylinders and I-beams. The precision, recall and IoU for

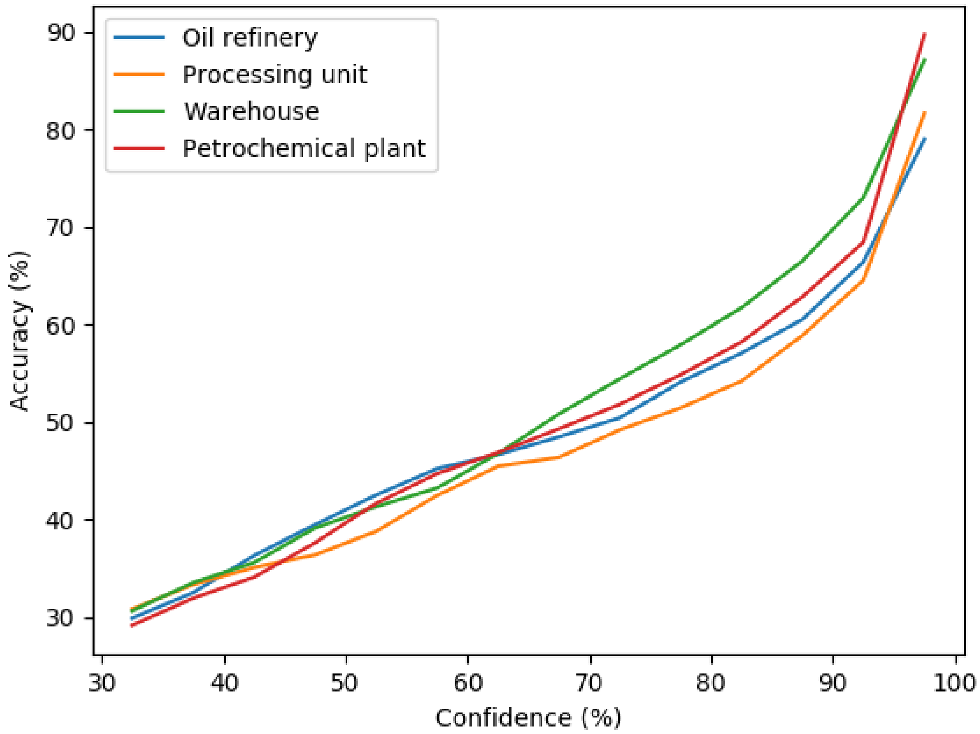

Fig. 14. Confidence level of predictions with respect to accuracy for each CLOI facility.

cylinders are 81.25%, 81.75% and 68.25% respectively. Likewise, the precision, recall and IoU for I-beams are 74.75%, 78.25% and 61.25% respectively. The other classes have lower performance metrics, however they still have non-trivial performance.

We observe a trend that rewards the performance of dominant *CLOI* classes such as cylinders and I-beams in Table 11). Their IoUs are 68.25% and 61.25% on average, whereas angles, valves and flanges have lower performance in our CLOI-NET methodology (26.15%, 28% and 21.25% IoUs respectively). Fig. 15 plots the empirical ROC curves for each facility on our data with minority classes being the angles and flanges. On average, *CLOI* facilities have a very high AUC measure of 95.6%. Micro-averaged metrics are used to aggregate the contributions of all classes (*CLOI* types) to compute the average metric. This metric is ideal for *CLOI* classes due to the class imbalanced *CLOI* datasets. In other words, many more points of cylinders and I-beams exist in the dataset in comparison to the other *CLOI* classes, therefore their metrics are higher.

Table 11. Average segmentation precision, recall and IoU (%) per *CLOI* shape

PointNET++ SFR	Angles	Channels	Cylinders	Elbows	I-beams	Valves	Flanges	Other
Precision	28.5	38.75	**81.75**	50.75	64	36.75	30	73.75
Recall	26.25	28.25	62.25	42.5	67	47.74	22.25	83
IoU	15.5	17.25	55	29.25	49	23.5	13.75	63
Cylinder adaptation	Angles	Channels	Cylinders	Elbows	I-beams	Valves	Flanges	Other
Precision	28.5	38.75	79.25	50.75	64	36.75	30	77
Recall	26.25	28.25	**77**	42.5	67	47.74	22.25	82
IoU	15.5	17.25	**58.5**	29.25	49	23.5	13.75	65
Steel shape/ Confidence level adaptation	Angles	Channels	Cylinders	Elbows	I-beams	Valves	Flanges	Other
Precision	28.5	**42.75**	81.75	50.75	64	36.75	30	74
Recall	26.25	**35.25**	62.25	42.5	67	47.74	22.25	83
IoU	15.5	**20.25**	55	29.25	49	23.5	13.75	63
CLOI-NET	Angles	Channels	Cylinders	Elbows	I-beams	Valves	Flanges	Other
Precision	**45.5**	**49.25**	81.25	54.75	74.75	41.25	39.75	84.5
Recall	**39.25**	**61.75**	81.75	49.25	78.25	55.25	33.5	86.5
IoU	**26.25**	**41.25**	68.25	33.75	61.25	28	21.25	74

While precision measures the probability of a *CLOI* class classified as true to actually be positive, the FPR measures the ratio of false positives within the true negative ("other") points. We expect the FPR metric to be higher for classes that have small number of points in the *CLOI* facilities due to the large number of points belonging to the dominant *CLOI* classes (cylinders, I-beams). We also show that the recall and precision of cylinders is penalizing the less frequent classes and we improve that with the post-processing confidence level adaptation and steel shape label contextual rule enforcement steps.

We further demonstrate the capacity of our CLOI-NET method on class segmentation of industrial shapes by adding the 3D detection results of the commercial software EdgeWise (ClearEdge, 2019) in Table 12. The motivation behind comparing with EdgeWise is that in our previous work (Agapaki et al., 2018), this software package showed superior performance out of all available software and research methods on automatically detecting cylinders. We evaluate the precision and recall of our method for each *CLOI* class and we compare (only

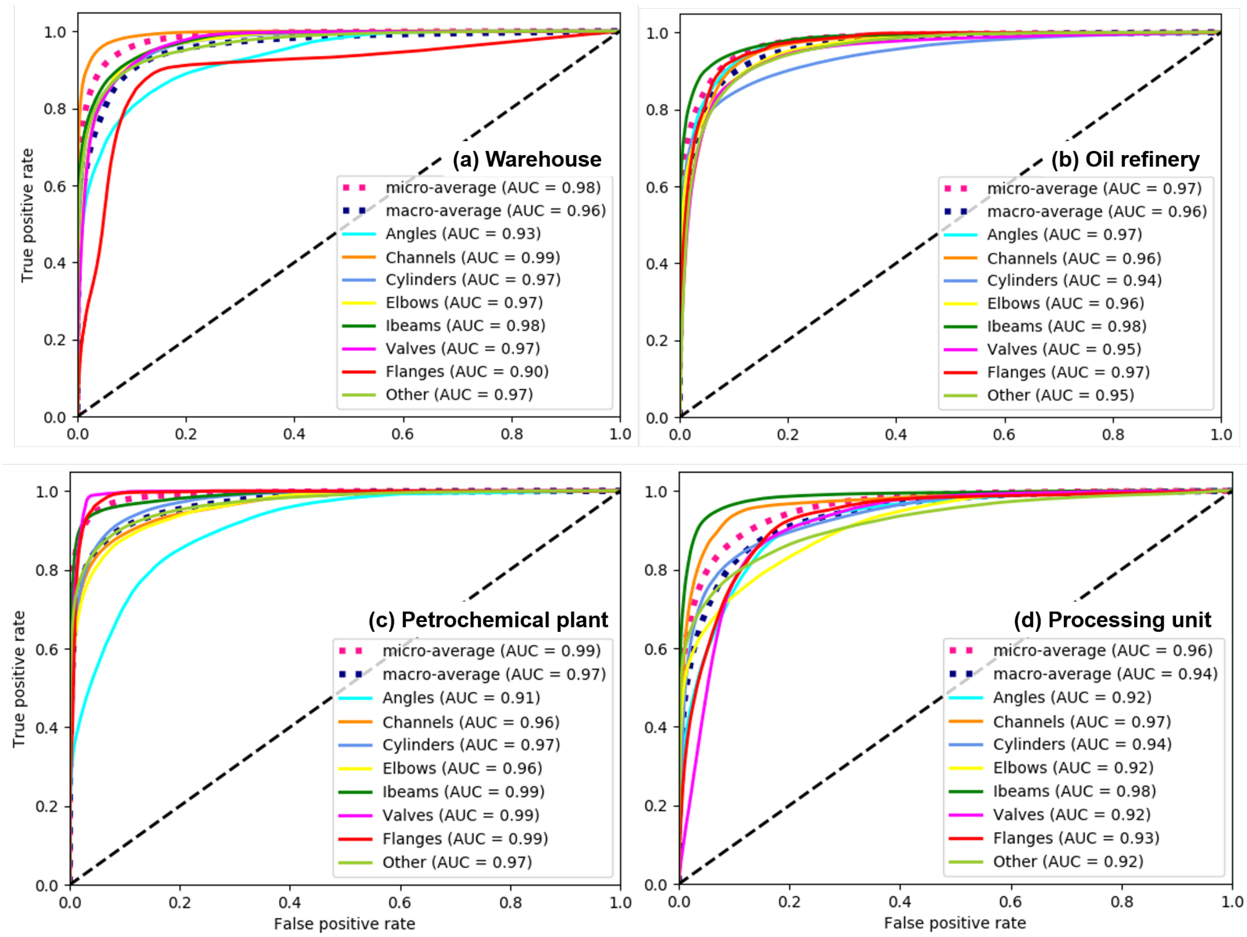

Fig. 15. CLOI-NET ROC curves across *CLOI* facilities.

intuitively and not directly) with the respective metrics with EdgeWise. It is important to note that EdgeWise does not automatically segment structural steel components other than cylinders. This is the reason that only the cylinder segmentation results of EdgeWise are included in Table 12. The difference in the evaluation metrics between our method and EdgeWise (ClearEdge, 2019) is that cylinder segmentation in EdgeWise is measured per fitted cylindrical shapes whereas our calculation of the same metrics is based on points. Once we have instance segmentation results which are out of the scope for this paper, we can make a direct comparison for cylinders.

The goal of **S1** detection methods is not to solve the cylinder segmentation problem (Section 3.3). One of the main merit of our method is that users can separate the points of each class, further process them and then more efficiently and intelligently generate the gDT without losing the point cloud information. Cylinder detection **S1** methods do not provide information per point rather they directly fit cylindrical shapes. While having been widely researched and achieving promising performance, these methods do not comply with the assumptions (Section 4.1) and scope of this work (Section 3.1). It is important to note that the goal of this work is not to compare **S1** and **S2** methods (where applicable) with each other. We rather presented and validated our proposed solution that best addresses the pain points of the current practice as outlined in Section 1. Direct comparison with existing methods of the **S1** object detection literature is out of the scope of this work. This is attributed to the fact that the metrics used in **S1** and **S2** methods are not comparable. The former compare precision and recall metrics on shapes, whereas the latter compare precision and recall on 3D points. Therefore, a comparison of state-of-the-art existing class segmentation methods on TLS datasets is presented as follows.

Table 12. Segmentation precision and recall per shape for the petrochemical plant and warehouse point clouds

Precision/recall (%)	Angles	Channels	Cylinders	Elbows	I-beams	Valves	Flanges
EdgeWise (Petrochemical plant)	-	-	69.3/59.6	-	-	-	-
CLOI-NET (Petrochemical plant)	25/27	63/72	76/83	40/41	75/83	14/70	39/43
EdgeWise (Warehouse)	-	-	41.25/69.8	-	-	-	-
CLOI-NET (Warehouse)	44/44	86/91	79/85	42/59	51/75	57/51	21/24

We compare the performance of state-of-the-art class segmentation methods with our proposed CLOI-NET Class segmentation proposed solution. The results in Table 13 show that our method has the highest performance when tested on the oil refinery *CLOI* facility presenting the first method of its kind to solve the class segmentation task on industrial TLS data.

Table 13. Comparison of state-of-the-art class segmentation methods tested on the oil refinery dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	mIoU (%)
PointNET (Qi et al., 2017b)	50	21	19	12
PointNET++ (Qi et al., 2017a)	68	46	41	32
SGPN (Wang et al., 2018b)	-	12.3	14.5	7.6
ASIS (Wang et al., 2019a)	-	26.5	23.9	14.5
DGCNN (Wang et al., 2019b)	66	36	31	22
CLOI-NET (passive)	**72**	**54.9**	**55.1**	**40.8**
CLOI-NET (active)	**83**	**59**	**59.6**	**45.1**

One can observe the visualization results of the four *CLOI* facilities in Fig. 16 and compare them with the ground truth annotated points. We color points in both ground truth and predicted classes based on the semantic class label they belong to. We also show illustrative examples of predicted and ground truth point clusters of each *CLOI* class in Fig. 17 and Fig. 18.

One can visualize windows where the CLOI-NET predicted labels and ground truth labels are presented in Fig. 17. Fig. 17(a) shows that in some cases our predicted labels depict the existing conditions even better than the ground truth due to annotation errors in the ground truth data. For example, the points of an elbow are correctly predicted, however our ground truth misclassified those. We also observe another case in Fig. 17(b) where flanges are considered as parts of valves. This is also a reasonable near-miss, since flanges are sometimes parts of valves. We also encounter this issue when generating our *CLOI* ground-truth labels, where in cases it would be difficult to separate flanges from valves. Regions where misclassifications are frequently encountered are usually close to the ceiling or floor of the facilities in densely occluded regions as shown in Fig. 17(c) and Fig. 17(d). We observe that even with human eyes, one could not distinguish the shapes close to the roof. If one wants to capture these regions more accurately, a more specialized laser scanning survey should be conducted. As such, low performance of our CLOI-NET algorithms in these

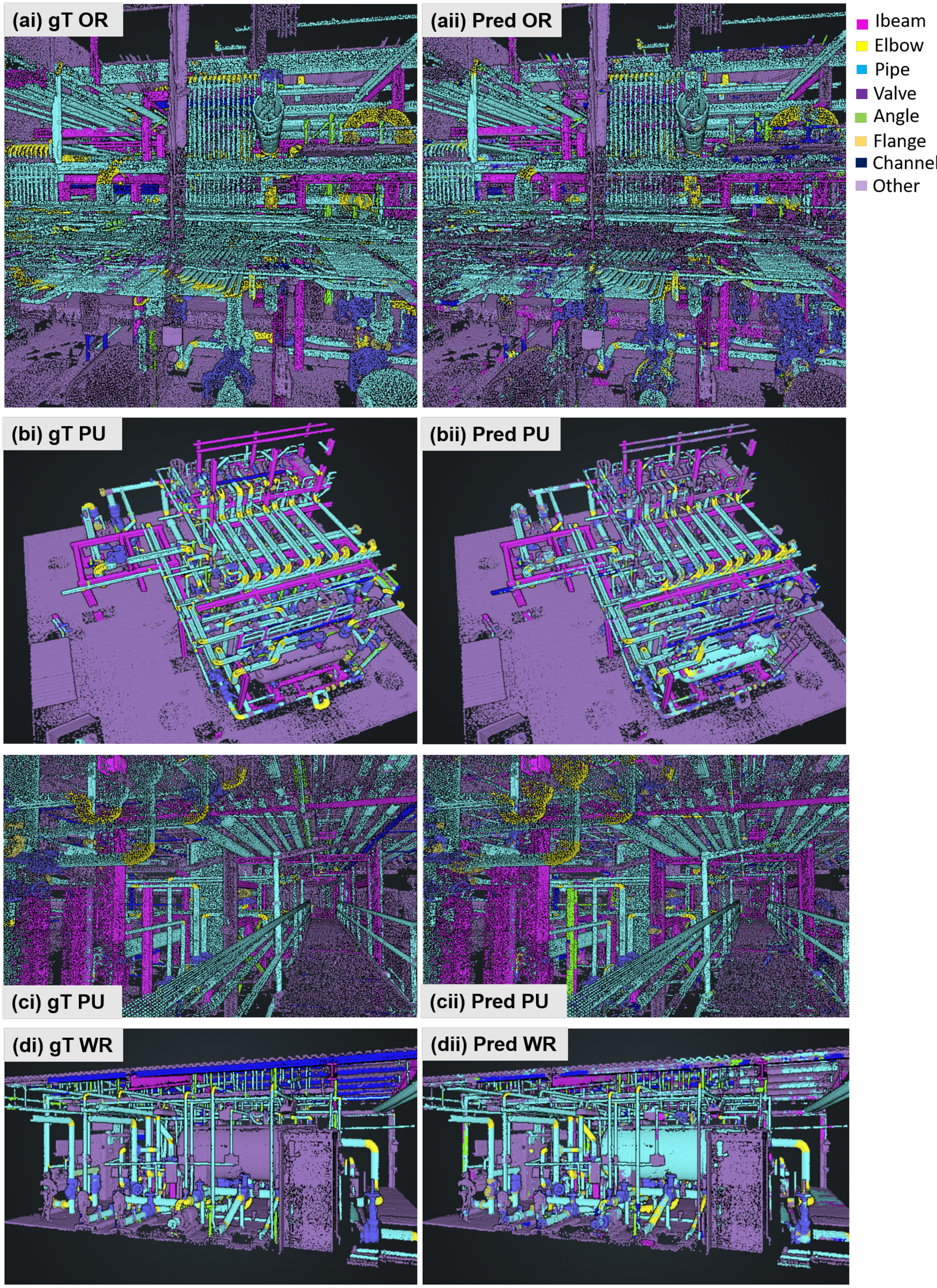

Fig. 16. (i) Ground truth annotated points and (ii) automatically segmented points across all *CLOI* facilities.

regions is reasonable to be expected.

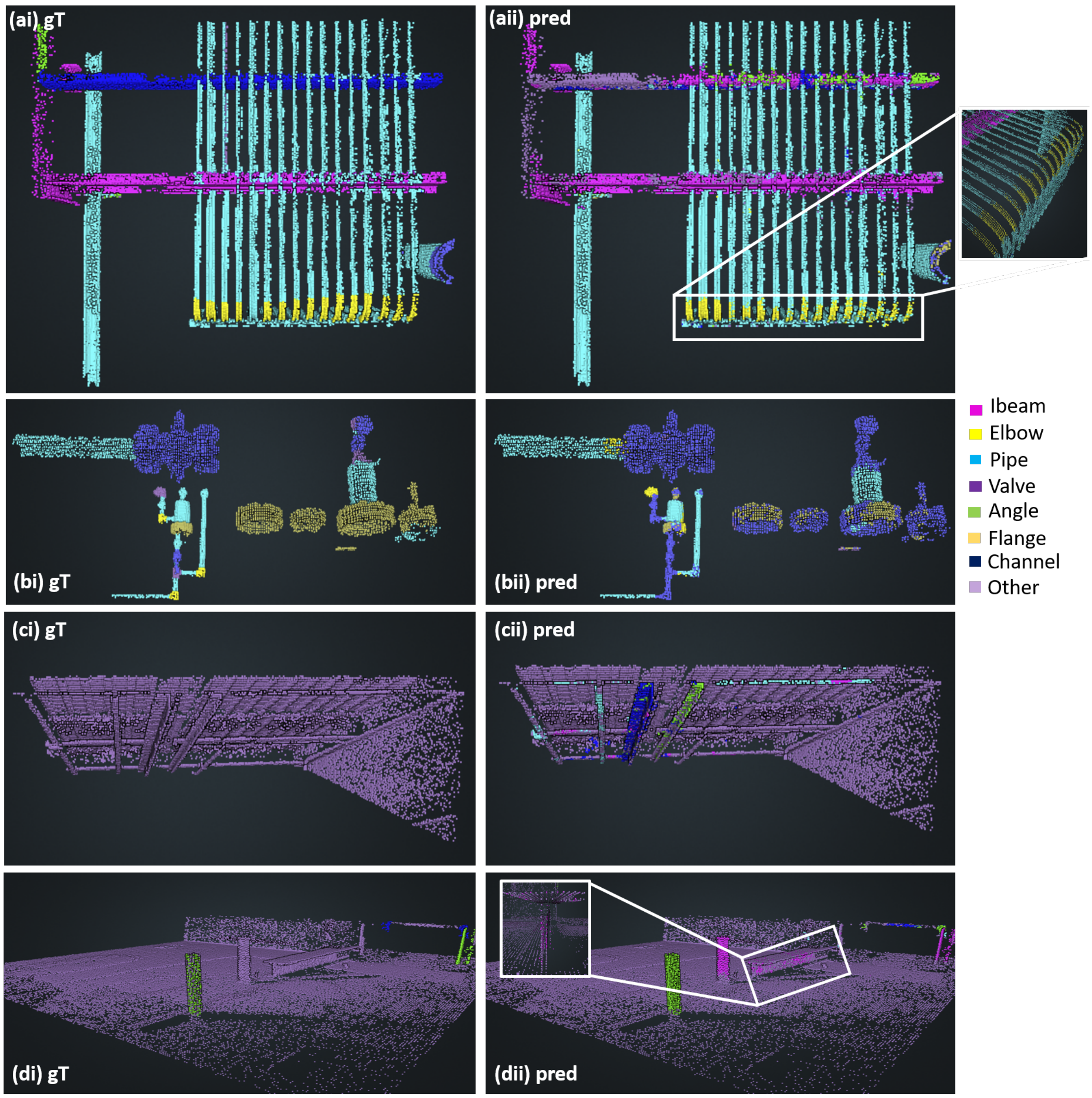

Fig. 17. (i) Ground truth annotated points and (ii) automatically segmented points generated from the oil refinery dataset.

4.5. Discussion

In this section we discuss the performance of CLOI-NET in two levels: (a) overall performance and (b) class component performance. For the overall performance of our methodology, we first investigate the robustness of our method by determining the facility bias. Then, we measure the cost savings by implementing our CLOI-NET active learning approach. The

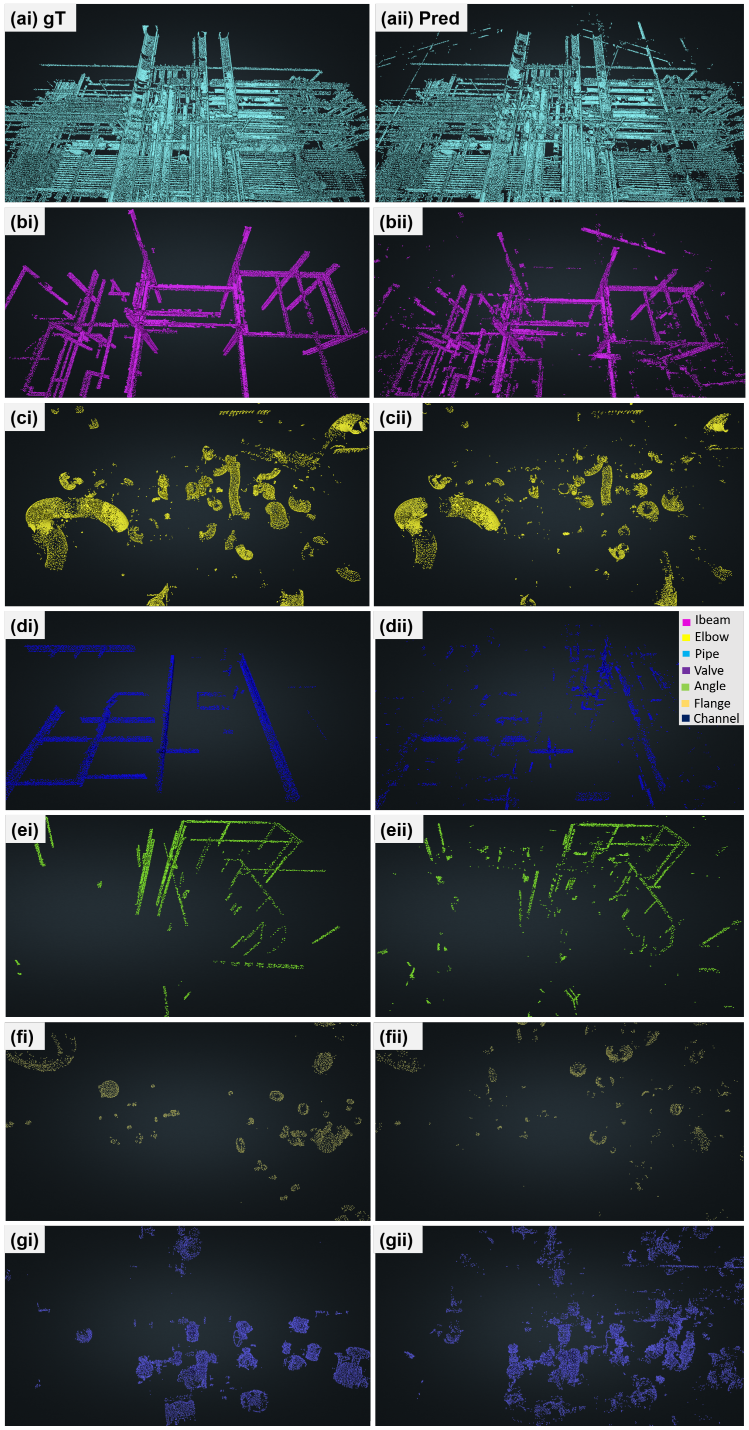

Fig. 18. (i) Ground truth annotated *CLOI* point clusters and (ii) automatically segmented *CLOI* point clusters generated from the oil refinery dataset.

second part of this section focuses on the discussion of our method's performance on class component performance.

4.5.1. CLOI-NET overall performance

The average class segmentation accuracy and mIoU are 66.5% and 44.65% when all the *CLOI* facilities are included for training except the one of interest to segment that is tested ("all but test"). CLOI-NET has been proven to be consistent, reliable and without significant bias, since the class segmentation performance for all *CLOI* facilities has a small standard deviation (3.57% test accuracy). The CLOI-NET performance using the active learning approach ("all") has greater standard deviation (6.12%) and average accuracy of 82%, which may be attributed to the greater difference between the *CLOI* facilities. We investigate two main factors that can account for this bias of our *CLOI* training dataset. These are (a) the point density of each *CLOI* facility and (b) *CLOI* facility diversity.

Point density and diversity of *CLOI* facilities

Fig. 19 shows the normalized point density across all four *CLOI* facilities with their 25% ($Q1$) and 75% ($Q3$) percentiles. We demonstrate that the point density of the oil refinery is at least one order of magnitude greater compared to the other three datasets, meaning that this facility was more densely scanned. This finding is also in line with Table 9, where we observe that this facility has the largest number of scans covering the largest number of points in comparison to the rest of the facilities. Fig. 19 also demonstrates that there is wider dispersion of data across facilities as indicated by the range of $Q1-Q3$ percentiles. We observe that the point density is not normally distributed across *CLOI* facilities. Instead, the point distributions are skewed towards point densities less than 200 000 points per square meter of a facility especially for the petrochemical plant. This facility has larger open spaces compared to the other facilities.

We determine (b) by training a PointNET++ network (Qi et al., 2017a) to predict whether a facility is recognizable by its shapes in order to investigate facility bias. We train a PointNET++ network that has as inputs non-overlapping 3D blocks of all *CLOI* facilities and gives as outputs the predicted facility where a 3D block belongs to. If the network correctly predicts from which facility the 3D block came from, it means the facility is differentiated in comparison to the other *CLOI* facilities. In other words, this means that facilities would not be similar to each other, should the network distinguish them. Fig. 20 demonstrates that all four *CLOI* facilities are distinguishable by coloring the predictions per facility as a heatmap. **Light red colors indicate high precision/recall, whereas darker colors**

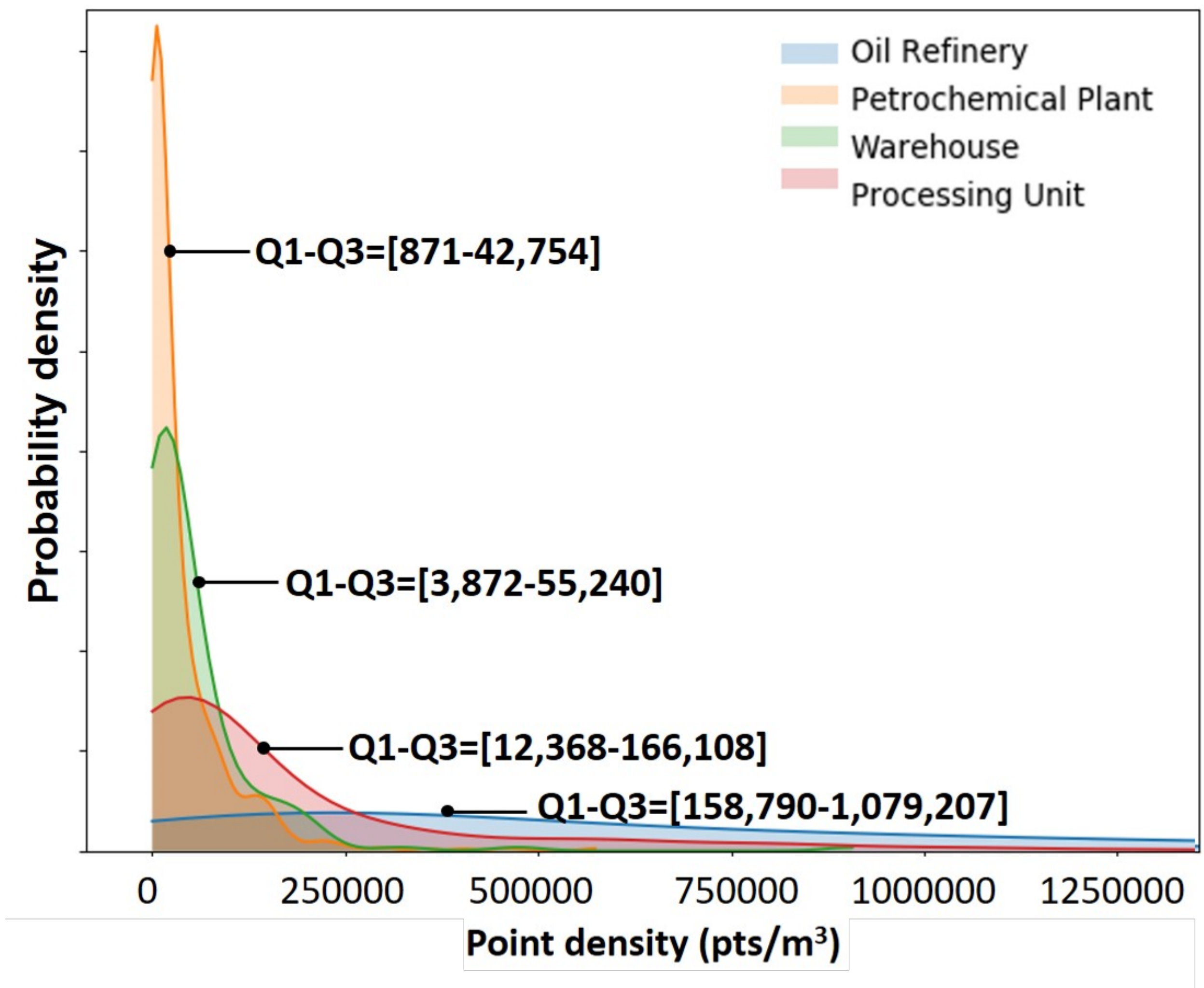

Fig. 19. Normalized point density across all *CLOI* trained facilities with 25% and 75% percentiles.

represent smaller precision/recall values. In particular, the petrochemical plant is the one that according to this experiment is more easily distinguishable (87% precision and 77% recall) compared to the other three *CLOI* facilities. The oil refinery is the facility that PointNET++ has more difficulty to distinguish, since its shapes have 24% likelihood to be incorrectly predicted as shapes of the petrochemical plant. There are two main factors for the difference in the shapes of the oil refinery facility: (a) the point density of the *CLOI* shapes is higher in the oil refinery facility compared to the other three facilities ($Q1-Q3$=$[158,790-1,079,207]$) and (b) the TLS survey accuracy is higher in comparison to the other TLS surveyed *CLOI* facilities ($< 0.7mm$ linearity error). This is reflected in the $mIoU$ performance which is higher for the oil refinery facility 47%, whereas for the processing unit and warehouse the $mIoU$ is 45.125% and 45.5% respectively. These facilities have similar overall performance due to the same factors. Their point densities are similar, $Q1-Q3$ range of $[3,872-55,240]$ and $[12,368-166,108]$ for the warehouse and the processing unit respectively. The wider range in the point density of the processing unit is attributed to the fact that it is the only outdoor facility in the *CLOI* dataset. Technically, outdoor scenes are inherently more occluded and incomplete exhibiting extreme variations in point density (Hackel et al., 2016). These effects are mitigated by the limited size and constrained shape of indoor facilities. The scanner properties are also comparable in the processing unit and the warehouse. For example, the processing unit has a linearity error of $3mm$ as opposed to $2mm$ linearity error of the warehouse (Table 9). This similarity is reflected in their mIoU metrics, which are 45.125% for the processing unit and 45.5% for the warehouse. The petrochemical plant has the lowest $mIoU$ performance of 42.5% due to the different industrial shapes it captures in comparison to the other *CLOI* facilities. For instance, the petrochemical plant has around 25% industrial shapes classified as "other". These shapes are mostly shapes belonging to electrical circuits and other electrical equipment (i.e. transformers, motor control centers). There are also rooms that the other facilities do not have, for instance an exhibition/conference room, resulting to the majority of the "other" shapes. However, the accuracy of the petrochemical plant is rewarded by the high performance of dominant classes like the "other" and "cylinder" with 25% and 45% of the points respectively. Henceforth, the accuracy is 70%.

Active Learning Cost Savings

We then validate our model of total annotation cost as presented in Section 3.6 on *CLOI*. We test our methodology on all *CLOI* facilities and here we use the oil refinery as an example to illustrate our methodology. We randomly select $X\%$ of (non-overlapping) 3D blocks of the oil refinery that we want to test on and include them in the training set with the rest of the *CLOI* facilities. We then measure the accuracy on the 1-X percentage of the 3D blocks of the oil refinery dataset that were not included for training. We further calculate the total annotation cost as a two-stage annotation cost from (a) the manual annotation cost of the X% fraction of the oil refinery using predictions after training *CLOI* with no pre-annotated 3D blocks from the oil refinery and (b) the manual annotation cost of the

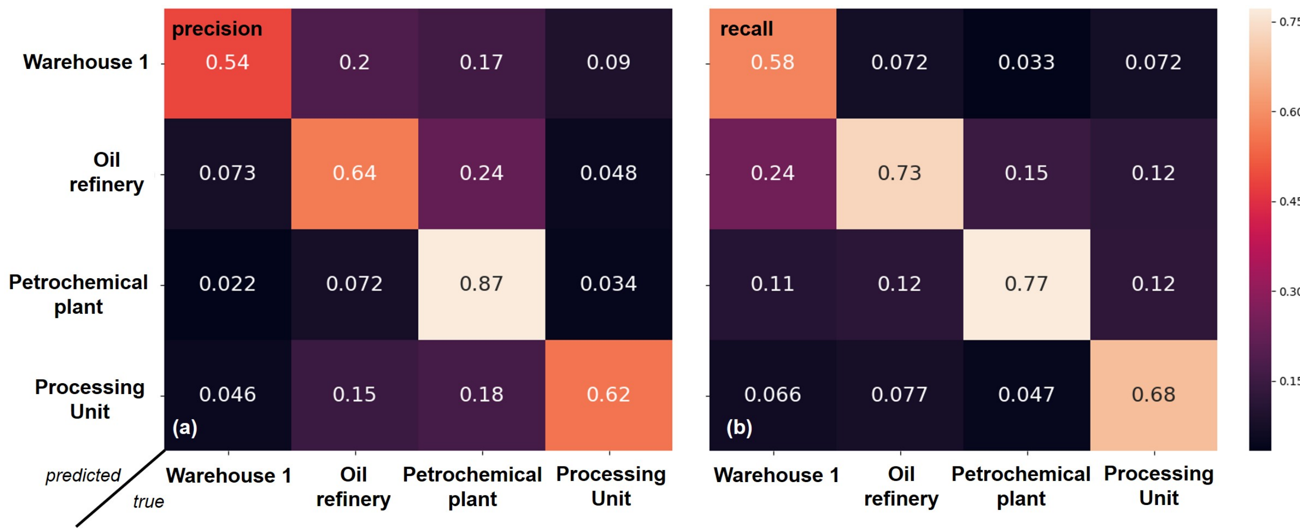

Fig. 20. Confusion matrices with (a) precision and (b) recall of network trained on all *CLOI* industrial shapes.

remaining $(1 - X)$ fraction of 3D blocks after augmenting the training with $X\%$ of the oil refinery 3D blocks. We present the resulting accuracy and total annotation cost curves in Fig. 21. Our analysis in Section 3.6 showed that it is never advantageous to pre-annotate more than 50% of 3D blocks, as such we select annotation percentages in the range of $[0, 50]$. We also try an annotation percentage of 80% of the 3D blocks to validate consistency of our results experimentally. We take four random samples at each annotation percentage in order to reduce variance. The average standard deviation for all our experiments is $\pm 0.4\%$. Experiments were conducted for the mIoU curves as well and since they have a similar trend with the accuracy curves, only the accuracy curves are illustrated in this paper.

We validate our theoretical model as outlined in Section 3.6 and Fig. 10. First, we prove that our training accuracy curve is a concave function with decreasing slope the more data we add during training. Also, we evaluate experimentally in Fig. 21 that the total cost annotation function is a convex function with global minimum at around 25% of annotated data in the oil refinery dataset. The results of the other facilities show that the optimal percentage is between 20 to 30%. This gives us the insight that the optimal window annotation percentage in order to minimize the total annotation cost is between $25 \pm 5\%$. We demonstrate that our accuracy curve in Fig. 21 is roughly (because of finite data) a highly non-linear, concave function, in contrast to the results by Jain and Grauman (2016) where for passive (random user annotation) their curve was a linear concave function. User selective techniques such as selection of 3D windows based on diversity and influence of selection could improve the accuracy rate increase, therefore these techniques can be considered in future work.

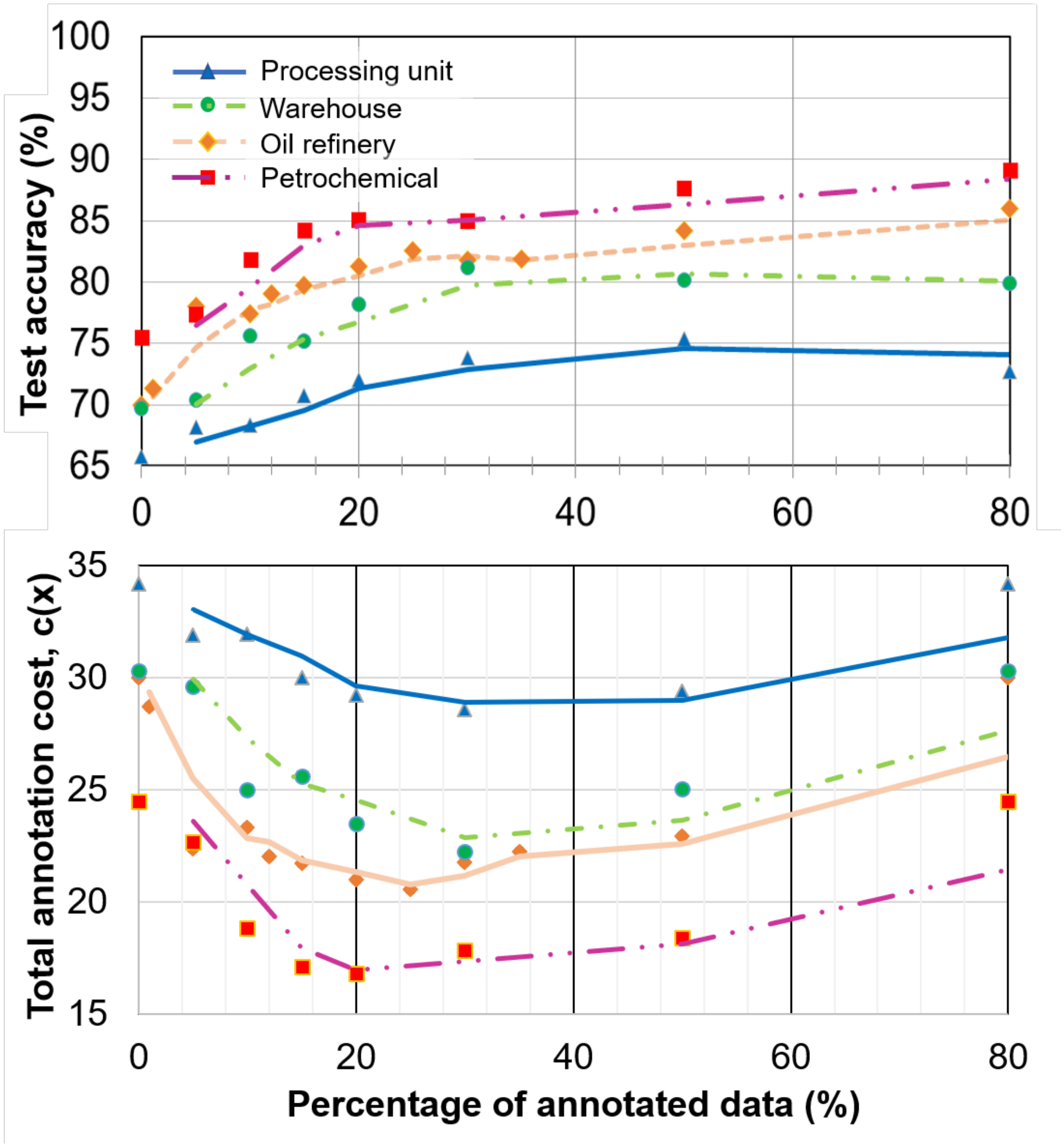

Fig. 21. (a) Test accuracy as a function of the percentage of annotated data included during training and (b) total annotation cost with respect to percentage of annotated data.

We then conducted a separate sensitivity analysis on the PointNET++ SFR network parameters compared to the original PointNET++ with respect to the active learning performance for pre-annotation rates 20%, 30% and 35%, in order to validate whether the selected parameters indeed yield significantly improved performance for cost optimization with active learning. The results of our experiments, which are illustrated in Fig. 22, indicate that, in all cases, the parameters of the PointNET++ SFR_3 network indeed lead to improved performance for the problem of manual labor cost optimization with the active learning network, regardless of the industrial facility tested or the choice of metric (accuracy or mIoU score). This proves the robustness of the PointNET++ SFR network, as its advantage is not specific to passive learning, but rather generalizes to the active learning approach as evaluated by the annotation cost optimization framework.

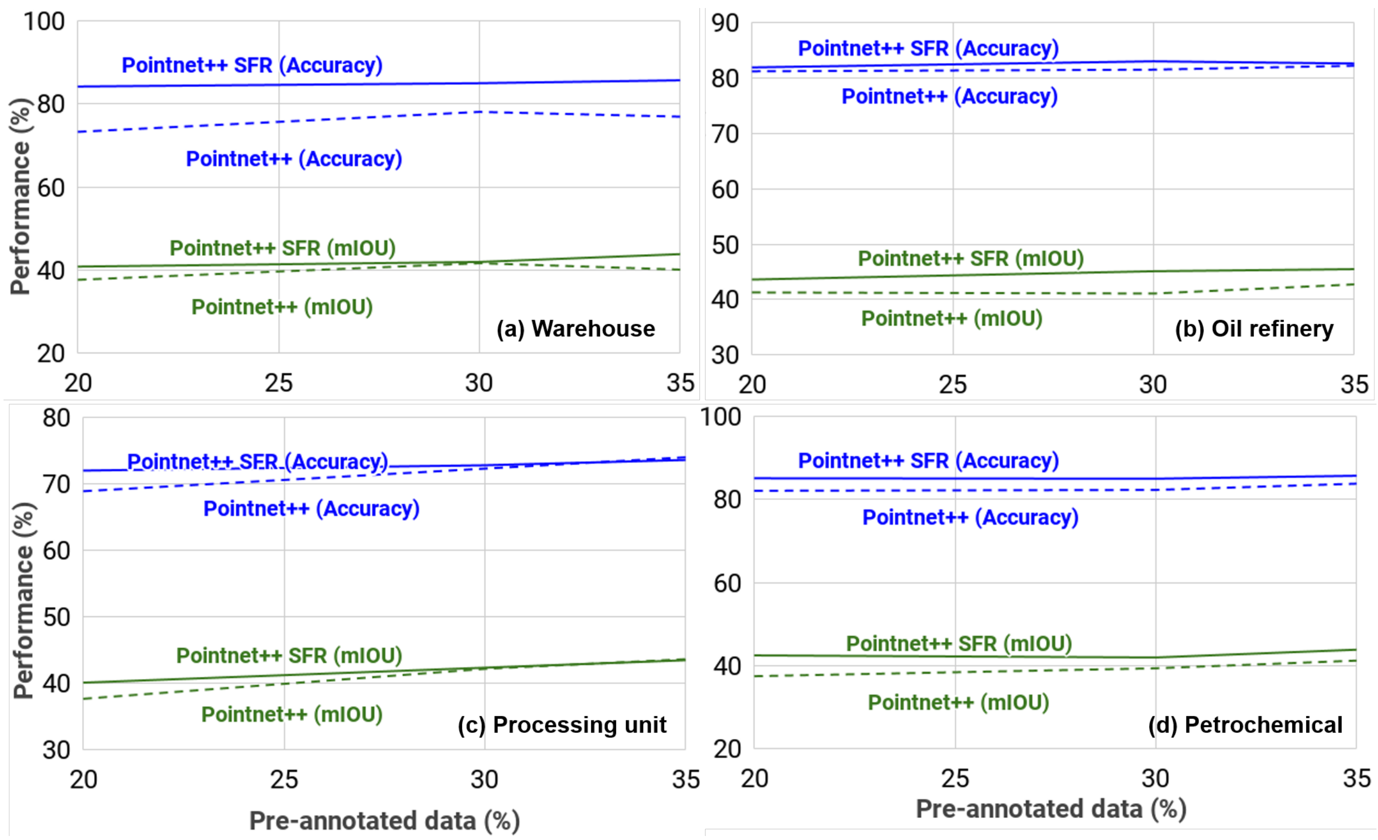

Fig. 22. Performance of the active learning approach with respect to the pre-annotated data percentage on all the *CLOI* facilities

4.5.2. CLOI-NET performance on individual CLOI classes

All *CLOI* facilities had very high micro-average AUC (higher than 90%), specifically the AUC for the warehouse, the oil refinery, the petrochemical plant and the processing unit was 96%, 96%, 96.75% and 93.75% respectively. The AUC for the angles of the warehouse and the petrochemical plant (93% and 91%) have reduced performance compared to those of the other *CLOI* classes. The percentage of angles in the petrochemical plant dataset is the lowest in comparison to the other *CLOI* classes in the same dataset (less than 5%), which means that inherently angles are rare to find in this dataset and were also difficult to distinguish even in the manual ground truth annotation. The angles of the petrochemical plant have also relatively low AUC (71%), which is attributed to the channels being parts of stairs or roof steelwork that is difficult to identify even with the human eye. This is more evident in this facility due to the roof having more steel members to support it than the other facilities. This problem can be addressed if the laser survey specifically targets roof refurbishment and other laser equipment (i.e. drones) could be used to improve the accuracy of the laser survey.

The average precision (PR), recall (R) and IoU were very high for **cylinders** and **I-beams** (above 75%) for most of *CLOI* facilities (Table 11). Particularly, the average PR of cylinders is 81.25% (std=6.3%) and the same metric for I-beams is 74.75% (std=16.8%). The reason for the higher PR standard deviation of I-beams is due to their reduced PR in the warehouse facility (51%). The I-beams of this facility were highly occluded (more than 50% of their shape occluded), as such they were misclassified as channels or "other". The class label adaptation for channels (Step 3.5.2) improved their IoU by 5% on average and corrected the misclassified channel points to I-beams and the reverse (7% increase in R and 5% increase in PR of channels). R is higher for both the warehouse cylinders (81.75% on average, std=3.4%) and I-beams (78.25% on average with std=3.4%). Respectively, IoU is 68.25% on average for the warehouse cylinders (std=5.6%) and 61.25% for I-beams (std=12.4%). The cylinder PR of the warehouse and the petrochemical plant (79% and 76% respectively) is relatively lower than PR of cylinders in the other facilities. For the warehouse, this is attributed to the false positives of cylinders in the corrugated steel profiles of the roof, which is a primary reason for the reduced PR of cylinders (41.25%) using EdgeWise in the same dataset as well (Agapaki et al., 2018). For the petrochemical plant, the PR of cylinders is lower (76%) compared to the other *CLOI* facilities, since it has steel trapezoid profiles in the roof and corrugated steel profiles for wall cladding, which in many cases are misclassified as cylinders. Another reason for the reduced PR of cylinders in the same facility is that the roof is composed of steel tubular roof trusses that in the ground truth were mislabeled as "other". Our CLOI-NET correctly predicted the point clusters of the tubular steel truss as cylinders, however the performance metrics are reduced due to the mislabeled ground truth. The petrochemical plant has rollover cables that our CLOI-NET predicted as false positives due to the sparsity of points in these clusters and highly occluded cables due to twists and congestion of conduits in cable trays. Another reason for the inferior cylinder performance in the petrochemical plant is that this facility has two grip strut safety metal grating walkways and stair treads both with serrated diamond patterns. The complexity of these geometric patterns that in many cases can be similar to cylindrical shapes leads to most of the points of the walkway and stairs being incorrectly classified as cylinder points.

The other *CLOI* classes (angles, channels, elbows, valves and flanges) had lower metrics compared to the dominant classes for all facilities, however they are still significant given that they outperform the current practice and research that do not solve class segmentation of those shapes. The petrochemical facility has initially lower than average performance in **channel** segmentation (6% IoU) as shown in Table 11. However, our CLOI-NET methodology increases IoU to 50% with 20% pre-annotated data (from Fig. 10) of the petrochemical facility in the training dataset. The remaining misclassifications of channel points are partly due to cable organizer side channels that are incorrectly classified as "other" in the ground truth data. Another reason is that the petrochemical plant has many rectangular columns which are misclassified as channels by our CLOI-NET methodology, as a result their PR is 63%. A third reason for the reduced R of channels (72%) is that many channels in the roof are incorrectly classified as cylinders. The PR of **valves** is initially very low (8%) in the petrochemical plant and is not greatly improved (PR=14%) due to two main reasons: (a) in most cases electrical inductors are misclassified as valves and (b) the incorrectly predicted valve points belong to spotlights close to the roof of the petrochemical plant. However, the valves of the warehouse have satisfactory performance (57% PR, 51% R, 36% mIoU), since most of them (37 out of 79 valves) are globe valves with hand wheels and check valves that have distinctive geometric shape. The near-missed points of warehouse valves are mostly misclassified as flanges (Fig. 17) or points of flanged ball valves with maximum face to face dimension of 33cm. We have a similar trend in the other *CLOI* datasets. The **angles** of the petrochemical plant are improved with our CLOI-NET methodology (from IoU=8% to IoU=14%). However, the angle performance in this unit is still the lowest compared to the other *CLOI* facilities since the angle points were mostly parts of steel cross braces that were misclassified as "other".

The performance of our CLOI-NET methodology is significant despite these misclassifications of underrepresented *CLOI* classes as discussed above. Additional methods that address those limitations need to be investigated. Although it is too soon to claim that the proposed method will address all needs in *CLOI* industrial class segmentation, the experiments proved that our method fills some gaps in knowledge and is capable of dealing with complex and diverse industrial spaces. This method can be the foundation to segment other industrial shapes.

5. CONCLUSIONS

Class segmentation in industrial point clouds remains an unsolved problem. In this paper, we presented CLOI-NET, a novel deep learning and geometric method for the segmentation of the most important industrial shapes in TLS point clouds, and tested it in the largest dataset of real-world industrial facilities (*CLOI*), generated by the authors. The validation metrics showed that our CLOI-NET method is reliable, scale and TLS scanning system independent. Our active learning optimization resolves the bias between the annotated *CLOI* facilities and potential facilities that can be added in the future. Therefore, our CLOI-NET method is only based on the registered industrial point cloud itself regardless of the varying point densities. These support our hypothesis. Given the high performance of our method compared to existing research and commercial tools on real world TLS industrial point clouds containing defects such as occlusions and sparseness, we contend that there is virtually minimal manual human intervention needed in our entire pipeline due to the high confidence of our CLOI-NET's predictions. We do not claim that our CLOI-NET method has no manual user intervention, rather we minimize manual modeling by achieving optimal performance when pre-annotating. Given the theoretical model we developed in Section 3.6 and validated experimentally in Section 4.5.1, our CLOI-NET model with passive learning saves on average 66% (std=3.8%) of the manual labor hours needed for class segmentation. The same model with our active learning methodology achieves on average 80% (std=6.1%) of manual labor savings. Further work can theoretically validate the applicability of more complicated models on the correlation between the learning accuracy and the annotation cost.

The contributions of this research are the following:

1. Our method can deal with complex real-world industrial facility settings, such as highly dense industrial spaces (oil refinery dataset). Our CLOI-NET achieved remarkably high performance in all facilities, even on the processing unit, which was surveyed with $3mm$ linearity error scanner and the survey was in gray-scale.
2. Our method automatically segments the *CLOI* shapes and as a by-product of this paper, our method generates the largest annotated dataset in the built environment. Researchers interested in the industrial space are welcomed to contribute to our dataset directly.
3. Our CLOI-NET method is the first to automatically and robustly solve the class segmentation of cylinders, I-beams and valves that have easily distinguishable geometric patterns in the processing unit, warehouse and the oil refinery. It also achieves remarkable performance in the remaining *CLOI* classes and in many cases defers from the ground truth due to manual annotation errors.
4. Our method dramatically reduces the computational costs by applying an active learning method. In this way, the manual annotation time is minimized without sacrificing performance and manual cost.

However, the proposed method does not intend to be a cure-all. It is limited on further cylinder classification into pipes, circular hollow sections and conduits. This can enhance the class segmentation and subsequently add value to the IgDT generation. Further cylinder classification is part of future work that we want to address. Although our method presents lower metrics for shapes with ambiguous and noisy edges like structural steel shapes and flanges, it is important to consider that for these shapes, it is difficult and ambiguous even for the human eyes to recognize them. Still, a more detailed TLS survey should be conducted towards accurate segmentation of industrial steel shapes, should the modelers want higher performance. Further investigation of the TLS survey parameters with the CLOI-NET performance is an interesting direction to be considered in future work.

Our method is designed with several practice implications, i.e. automated segmentation of important industrial shapes, fully automated and ready to test on unlabeled facilities. Therefore, we first expect to save industrial managers' valuable time in data collection and annotation of their facilities, so that they can concentrate their efforts on tackling unprecedented circumstances and solving problems that necessarily demand their expertise. Secondly, it can be applicable to spaces where CLOI classes appear, since it decomposes large industrial open spaces into meaningful smaller windows. Thirdly, our method works directly on the point cloud data and as a result is not dependent of a data capturing technique and system.

Future planned research activities will focus on (1) the overcoming of the above-mentioned limitations and addressing some of the assumptions; (2) instance segmentation of CLOI shapes and use classification of cylinders; and (3) fitting IFC objects to the generated labeled point clusters.

6. ACKNOWLEDGEMENTS

We thank our colleague Graham Miatt, who has provided insight, expertise and data that greatly assisted this research. We also express our gratitude to Bob Flint from BP International Centre for Business and Technology (ICBT), who provided data for evaluation. The research leading to these results has received funding from the Engineering and Physical Sciences Research Council (EPSRC) and the US National Academy of Engineering (NAE). AVEVA Group Plc. and BP International Centre for Business and Technology (ICBT) partially sponsor this research under grant agreements RG83104 and RG90532 respectively. We also want to thank Sara Mandoki and Alex Glyn-Davies, two undergraduate students at the University of Cambridge, who meticulously assisted with the CLOI benchmark dataset generation. We gratefully acknowledge the collaboration of all academic and industrial project partners. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the institutes mentioned above.

References

E. Agapaki and I. Brilakis. Prioritising object types of industrial facilities to reduce as-is modelling time. In *Proceeding of the 33rd Annual ARCOM Conference, 4-6 September 2017*, pages 402–411, Cambridge, U.K., 2017.

E. Agapaki, A. Glyn-Davies, S. Mandoki, and I. Brilakis. CLOI: A Shape Classification Benchmark Dataset for Industrial Facilities. In *2019 ASCE International Conference on Computing in Civil Engineering*, 2019.

Eva Agapaki, Graham Miatt, and Ioannis Brilakis. Prioritizing object types for modelling existing industrial facilities. *Automation in Construction*, 2018. ISSN 09265805. doi: 10.1016/j.autcon.2018.09.011.

Anuraag Agrawal, Atsushi Nakazawa, and Haruo Takemura. MMM-classification of 3D range data. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2009. ISBN 9781424427895. doi: 10.1109/ROBOT.2009.5152539.

Mahmoud Fouad Ahmed, Carl T. Haas, and Ralph Haas. Automatic Detection of Cylindrical Objects in Built Facilities. *Journal of Computing in Civil Engineering*, 28(3): 04014009, 2014. ISSN 0887-3801. doi: 10.1061/(ASCE)CP.1943-5487.0000329. URL `http://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000329`.

AISC. ANSI / AISC 360-16, Specification for Structural Steel Buildings. *American Institute of Steel Construction*, 2016.

Anderson O. Akponeware and Zulfikar A. Adamu. Clash detection or clash avoidance? An investigation into coordination problems in 3D BIM. *Buildings*, 2017. ISSN 20755309. doi: 10.3390/buildings7030075.

Miktha Farid Alkadri, Francesco De Luca, Michela Turrin, and Sevil Sariyildiz. An integrated approach to subtractive solar envelopes based on attribute information from point cloud data. *Renewable and Sustainable Energy Reviews*, 2020. ISSN 18790690. doi: 10.1016/j.rser.2020.109742.

Amazon Mechanical Turk. Amazon Mechanical Turk, 2018. URL `https://www.mturk.com/`.

American Society of Mechanical Engineers (ASME). Scheme for the Identification of Piping Systems. Technical report, 2015. URL `https://www.asme.org/codes-standards/find-codes-standards/a13-1-scheme-identification-piping-systems`.

Engin Burak Anil, Raghuram Sunnam, and Burcu Akinci. Challenges of identifying steel sections for the generation of As-Is BIMs from Laser Scan Data. In *2012 Proceedings of the 29th International Symposium of Automation and Robotics in Construction, ISARC 2012*, 2012.

ANSI Z535.1. American National Standard for Safety Colors. Technical report, 2017. URL `https://www.nema.org/Standards/ComplimentaryDocuments/ANSIZ535_1-2017CONTENTSANDSCOPE.pdf`.

I. Armeni, O. Sener, H. Jiang, M. Fischer, and S. Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

I. Armeni, A. Sax, A. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. Technical report, 2017.

AVEVA. LFM Software, 2019. URL `https://sw.aveva.com/engineer-procure-construct/lfm-server`.

J. Behley, V. Steinhage, and A. Cremers. Performance of Histogram Descriptors for the Classification of 3D Laser Range Data in Urban Environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2012.

Jens Behley, Kristian Kersting, Dirk Schulz, Volker Steinhage, and Armin B. Cremers. Learning to hash logistic regression for fast 3D scan point classification. In *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010. ISBN 9781424466757. doi: 10.1109/IROS.2010.5650093.

Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI. In *ICCV*, 2019.

A. Bey, R. Chaine, R. Marc, G. Thibault, and S. Akkouche. Reconstruction of Consistent

3D CAD Models from Point Cloud Data using a priori CAD models. In *ISPRS Workshop Laser Scanning*, pages 29–31, Calgary, Canada, 2011.

S. Biasotti, A. Cerri, A. Bronstein, and M. Bronstein. Recent Trends, Applications, and Perspectives in 3D Shape Similarity Assessment. *Computer Graphics Forum*, 2016. ISSN 14678659. doi: 10.1111/cgf.12734.

Jon Bokrantz, Anders Skoogh, Torbjörn Ylipää, and Johan Stahre. Handling of production disturbances in the manufacturing industry. *Journal of Manufacturing Technology Management*, 2016. ISSN 1741038X. doi: 10.1108/JMTM-02-2016-0023.

Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 1997. ISSN 00313203. doi: 10.1016/S0031-3203(96)00142-2.

BS 1710:2014. Specification for identification of pipelines and services. Technical report, 2014. URL `https://shop.bsigroup.com/ProductDetail?pid=000000000030283829`.

BS EN 10365:2017. Hot rolled steel channels, I and H sections. Dimensions and masses. Technical report, 2017. URL `https://shop.bsigroup.com/ProductDetail/?pid=000000000030356574`.

M. Cabaleiro, B. Riveiro, P. Arias, J.C. Caamaño, and J.A. Vilán. Automatic 3D modelling of metal frame connections from LiDAR data for structural engineering purposes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 96:47–56, 2014. ISSN 09242716. doi: 10.1016/j.isprsjprs.2014.07.006. URL `http://dx.doi.org/10.1016/j.isprsjprs.2014.07.006%5Cnhttp://linkinghub.elsevier.com/retrieve/pii/S0924271614001804`.

Jingdao Chen, Zsolt Kira, and Yong K. Cho. Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction. *Journal of Computing in Civil Engineering*, 2019. ISSN 08873801. doi: 10.1061/(ASCE)CP.1943-5487.0000842.

Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3D Object Detection for Autonomous Driving. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2147–2156, 2016. ISSN 10636919. doi: 10.1109/CVPR.2016.

236. URL http://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Chen_Monocular_3D_Object_CVPR_2016_paper.html.

C. Choy, J. Gwak, and S. Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

CISC. CISC Code of Standard Practice for Structural Steel. Technical report, Canadian Institute of Steel Construction, 2015. URL https://www.cisc-icca.ca/product/cisc-code-of-standard-practice-for-structural-steel-8th-edition/.

ClearEdge. Plant Modeling Capabilities, 2019. URL https://www.clearedge3d.com/products/edgewise-plant/.

Cloudcompare. Cloudcompare (version 2.9.alpha), 2016. URL http://www.cloudcompare.org/.

Franco Coren and Paolo Sterzai. Radiometric correction in laser scanning. *International Journal of Remote Sensing*, 2006. ISSN 13665901. doi: 10.1080/01431160500217277.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.261.

Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jurgen Sturm, and Matthias Niebner. ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. ISBN 9781538664209. doi: 10.1109/CVPR.2018.00481.

Olivier Devillers, Bernard Mourrain, Franco P. Preparata, and Philippe Trebuchet. Circular cylinders through four or five points in space. *Discrete and Computational Geometry*, 2003. ISSN 01795376. doi: 10.1007/s00454-002-2811-7.

Lucía Díaz-Vilariño, Kourosh Khoshelham, Joaquín Martínez-Sánchez, and Pedro Arias. 3D Modeling of Building Indoor Spaces and Closed Doors from Imagery and Point Clouds. *Sensors*, 15(2):3491–3512, 2 2015. ISSN 1424-8220. doi: 10.3390/s150203491.

Andrey Dimitrov and Mani Golparvar-Fard. Segmentation of building point cloud models including detailed architectural/structural features and MEP systems. *Automation in Construction*, 51(C):32–45, 2015. ISSN 09265805. doi: 10.1016/j.autcon.2014.12.015.

Qiong Ding, Wu Chen, Bruce King, Yanxiong Liu, and Guoxiang Liu. Combination of overlap-driven adjustment and Phong model for LiDAR intensity correction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2013. ISSN 09242716. doi: 10.1016/j.isprsjprs.2012.09.015.

John Edwards and Andrew Townsend. Buildings under Refurbishment and Retrofit, 2011. URL `http://www.carbonaction2050.com/sites/carbonaction2050.com/files/document-attachment/BuildingsunderRefurbandRetrofit.pdf`.

European Commission. European Commission Pressure Equipment Directive (PED), 2010. URL `https://web.archive.org/web/20100815081132/http://ec.europa.eu/enterprise/sectors/pressure-and-gas/documents/ped/`.

European Standard. EN 1993-1-1 Eurocode 3: Design of steel structures. Technical report, 2005.

Mark Everingham, S. M.Ali Eslami, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 2014. ISSN 15731405. doi: 10.1007/s11263-014-0733-5.

Rui Figueiredo, Atabak Dehban, Plinio Moreno, Alexandre Bernardino, José Santos-Victor, and Helder Araújo. A robust and efficient framework for fast cylinder detection. *Robotics and Autonomous Systems*, 2019. ISSN 09218890. doi: 10.1016/j.robot.2019.04.002.

A. Filgueira, H. González-Jorge, S. Lagüela, L. Díaz-Vilariño, and P. Arias. Quantifying the influence of rain in LiDAR performance. *Measurement: Journal of the International Measurement Confederation*, 2017. ISSN 02632241. doi: 10.1016/j.measurement.2016.10.009.

Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395, 6 1981. ISSN 0001-0782. doi: 10.1145/358669.358692.

Michele Fumarola and Ronald Poelman. Generating virtual environments of real world facilities: Discussing four different approaches. In *Automation in Construction*, volume 20, pages 263–269, 2011. ISBN 0926-5805. doi: 10.1016/j.autcon.2010.08.004.

Demetrios Gatziolis and Hans Erik Andersen. A guide to LIDAR data acquisition and processing for the forests of the pacific northwest. *USDA Forest Service - General Technical Report PNW-GTR*, 2008. ISSN 08874840. doi: 10.2737/pnw-gtr-768.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012. ISBN 9781467312264. doi: 10.1109/CVPR.2012.6248074.

E. Grilli, F. Menna, and F. Remondino. A review of point clouds segmentation and classification algorithms. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, volume 42, pages 339–344, 2017. doi: 10.5194/isprs-archives-XLII-2-W3-339-2017.

Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan. 3D object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. ISSN 01628828. doi: 10.1109/TPAMI.2014.2316828.

T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017. doi: 10.5194/isprs-annals-IV-1-W1-91-2017.

Timo Hackel, Jan D. Wegner, and Konrad Schindler. Contour Detection in Unstructured 3D Point Clouds. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1610–1618, 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.178. URL `http://ieeexplore.ieee.org/document/7780547/`.

J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 1982. ISSN 00338419. doi: 10.1148/radiology.143.1.7063747.

Z Har'el. CURVATURE OF CURVES AND SURFACES–A PARABOLIC APPROACH. *Department of Mathematics, Technion–Israel Institute ...*, 1995.

Paul Heider, Alain Pierre-Pierre, Ruosi Li, Rolf Mueller, and Cindy Grimm. Comparing local shape descriptors. *Vis. Comput.*, 28(9):919–929, 2012. ISSN 0178-2789. doi: 10.1007/s00371-012-0725-9. URL `http://link.springer.com/article/10.1007/s00371-012-0725-9`.

Bernhard Höfle and Norbert Pfeifer. Correction of laser scanning intensity data: Data and model-driven approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2007. ISSN 09242716. doi: 10.1016/j.isprsjprs.2007.05.008.

P. V. C. Hough. Machine Analysis Of Bubble Chamber Pictures. C590914:554–558, 1959.

Binh Son Hua, Quang Hieu Pham, Duc Thanh Nguyen, Minh Khoi Tran, Lap Fai Yu, and Sai Kit Yeung. SceneNN: A scene meshes dataset with aNNotations. In *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, 2016. ISBN 9781509054077. doi: 10.1109/3DV.2016.18.

Jing Huang and Suya You. Detecting objects in scene point cloud: A combinational approach. In *Proceedings - 2013 International Conference on 3D Vision, 3DV 2013*, pages 175–182, 2013. ISBN 9780769550671. doi: 10.1109/3DV.2013.31.

Jean-François Hullo, Guillaume Thibault, Christian Boucheny, Fabien Dory, and Arnaud Mas. Multi-Sensor As-Built Models of Complex Industrial Architectures. *Remote Sensing*, 7(12):16339–16362, 2015. ISSN 2072-4292. doi: 10.3390/rs71215827. URL `http://www.mdpi.com/2072-4292/7/12/15827`.

Muhammad Ijaz, Zabih Ghassemlooy, Jiri Pesek, Ondrej Fiser, Hoa Le Minh, and Edward Bentley. Modeling of fog and smoke attenuation in free space optical communications link under controlled laboratory conditions. *Journal of Lightwave Technology*, 2013. ISSN 07338724. doi: 10.1109/JLT.2013.2257683.

International Institute of Ammonia Refrigeration. Guidelines for: Identification of Ammonia Refrigeration Piping and System Components. Technical report, Bulletin No. 114, 2014. URL `http://web.iiar.org/membersonly/PDF/CO/Bulletin_114_REV032014.pdf`.

Suyog Dutt Jain and Kristen Grauman. Active Image Segmentation Propagation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.313.

Young Hoon Jin and Won Hyung Lee. Fast cylinder shape matching using random sample consensus in large scale point cloud. *Applied Sciences (Switzerland)*, 2019. ISSN 20763417. doi: 10.3390/app9050974.

Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999. ISSN 01628828. doi: 10.1109/34.765655.

B. Jutzi and H. Gross. Normalization of lidar intensity data based on range and surface incidence angle. *International Archives of Photogrammetry and Remote Sensing*, 2009. ISSN 1682-1750.

Sanna Kaasalainen, Anttoni Jaakkola, Mikko Kaasalainen, Anssi Krooks, and Antero Kukko. Analysis of incidence angle and distance effects on terrestrial laser scanner intensity: Search for correction methods. *Remote Sensing*, 2011. ISSN 20724292. doi: 10.3390/rs3102207.

Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3D Shape segmentation with projective convolutional networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.702.

Alireza G. Kashani, Michael J. Olsen, Christopher E. Parrish, and Nicholas Wilson. A review of LIDAR radiometric processing: From ad hoc intensity correction to rigorous radiometric calibration, 2015. ISSN 14248220.

Kazuaki Kawashima, Satoshi Kanai, and Hiroaki Date. As-built modeling of piping system from terrestrial laser-scanned point clouds using normal-based region growing. *Journal of Computational Design and Engineering*, 1(1):13–26, 2014. ISSN 22884300. doi: 10.7315/JCDE.2014.002. URL `http://linkinghub.elsevier.com/retrieve/pii/S2288430014500024`.

Mahnoosh Kholghi, Laurianne Sitbon, Guido Zuccon, and Anthony Nguyen. Active learning reduces annotation time for clinical concept extraction. *International Journal of Medical Informatics*, 2017. ISSN 18728243. doi: 10.1016/j.ijmedinf.2017.08.001.

Roman Klokov and Victor Lempitsky. Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. ISBN 9781538610329. doi: 10.1109/ICCV.2017.99.

Ilkka Korpela, Hans Ole Ørka, Juha Hyyppä, Ville Heikkinen, and Timo Tokola. Range and AGC normalization in airborne discrete-return LiDAR intensity data for forest canopies. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2010. ISSN 09242716. doi: 10.1016/j.isprsjprs.2010.04.003.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 2012. ISSN 10495258. doi: http://dx.doi.org/10.1016/j.protcy.2014.09.007.

A. Krooks, S. Kaasalainen, T. Hakala, and O. Nevalainen. Correction of Intensity Incidence Angle Effect in Terrestrial Laser Scanning. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2013. doi: 10.5194/isprsannals-II-5-W2-145-2013.

Antero Kukko, Sanna Kaasalainen, and Paula Litkey. Effect of incidence angle on laser scanner intensity and surface data. *Applied Optics*, 2008. ISSN 15394522. doi: 10.1364/AO.47.000986.

Debra F. Laefer and Linh Truong-Hong. Toward automatic generation of 3D steel structures for building information modelling. *Automation in Construction*, 2017. ISSN 09265805. doi: 10.1016/j.autcon.2016.11.011.

G. Lakes. Recommended Standards for Wastewater Facilities. Technical report, 2004.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 2008. ISSN 0899-7667. doi: 10.1162/neco.1989.1.4.541.

Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning, 2015. ISSN 14764687.

Joohyuk Lee, Hyojoo Son, Changmin Kim, and Changwan Kim. Skeleton-based 3D reconstruction of as-built pipelines from laser-scan data. *Automation in Construction*, 35: 199–207, 11 2013. ISSN 0926-5805. doi: 10.1016/j.autcon.2013.05.009.

Biao Li, Yong Shi, Zhiquan Qi, and Zhensong Chen. A survey on semantic segmentation. In *IEEE International Conference on Data Mining Workshops, ICDMW*, 2019. ISBN 9781538692882. doi: 10.1109/ICDMW.2018.00176.

Duanshun Li and Chen Feng. Primitive fitting using deep geometric segmentation. In *Proceedings of the 36th International Symposium on Automation and Robotics in Construction, ISARC 2019*, 2019. doi: 10.22260/isarc2019/0105.

W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger. Interior-Net: Mega-scale Multi-sensor Photo-realistic Indoor Scenes Dataset. In *Proc. of the British Machine Vision Conference (BMVC)*, 2018.

Y. Li, S. Pirk, H. Su, R. Qi, and L. Guibas. FPNN: Field Probing Neural Networks for 3D data. In *FPNN: Field Probing Neural Networks for 3D data*, Barcelona, Spain, 2016.

Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep Continuous Fusion for Multi-sensor 3D Object Detection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. ISBN 9783030012694. doi: 10.1007/978-3-030-01270-0{_}39.

Yong Jin Liu, Jun Bin Zhang, Ji Chun Hou, Ji Cheng Ren, and Wei Qing Tang. Cylinder detection in large-scale point cloud of pipeline plant. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1700–1707, 2013. ISSN 10772626. doi: 10.1109/TVCG.2013.74.

Ruodan Lu and Ioannis Brilakis. Recursive Segmentation for As-Is Bridge Information Modelling. In *Proceedings of the Joint Conference on Computing in Construction (JC3)*, pages 209–217, Heraklion, Greece, 2017. doi: https://doi.org/10.24928/JC3-2017/0020.

Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean Philippe Thiran, and Mauricio Reyes. Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network. In *Lecture Notes in Computer Science* *(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. ISBN 9783030009335. doi: 10.1007/978-3-030-00934-2{_}65.

D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, 2015.

John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J. Davison. SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation? In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. ISBN 9781538610329. doi: 10.1109/ICCV.2017.292.

D. Munoz, N. Vandapel, and M. Hebert. Directional Associative Markov Network for 3-d Point Cloud Classification. In *Proc. of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 63–70, 2008.

Daniel Munoz, J. Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual classification with functional max-margin markov networks. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009. ISBN 9781424439935. doi: 10.1109/CVPRW.2009.5206590.

M. Nagase, M. Akizuki, and M. Hashimoto. 3D feature point matching for object recognition based on estimation of local shape distinctiveness. In Springer, editor, *Proceedings of the 15th International Conference on Computer Analysis of Images and Patterns*, pages 473–481, York, UK, 2013.

National Institute of Standards and Technology. The Costs and Benefits of Advanced Maintenance in Manufacturing. Technical report, U.S. Department of Commerce, 2018. URL `https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.100-18.pdf`.

Office of Maritime Administrator. Deepwater Horizon Maritime Casualty Investigation Report. Technical report, 2011.

Guan Pang and Ulrich Neumann. 3D point cloud object detection with multi-view convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 585–590, 2016. ISBN 978-1-5090-4847-2. doi: 10.1109/ICPR.2016.7899697. URL `http://ieeexplore.ieee.org/document/7899697/`.

Yachun Pang, Li Li, Wenyong Hu, Yanxia Peng, Lizhi Liu, and Yuanzhi Shao. Computerized segmentation and characterization of breast lesions in dynamic contrast-enhanced MR images using fuzzy c-means clustering and snake algorithm. *Computational and Mathematical Methods in Medicine*, 2012. ISSN 1748670X. doi: 10.1155/2012/634907.

H. S. Park, H. M. Lee, Hojjat Adeli, and I. Lee. A new approach for health monitoring of structures: Terrestrial laser scanning. *Computer-Aided Civil and Infrastructure Engineering*, 2007. ISSN 10939687. doi: 10.1111/j.1467-8667.2006.00466.x.

Ashok Kumar Patil, Pavitra Holi, Sang Keun Lee, and Young Ho Chai. An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds. *Automation in Construction*, 75:65–78, 2017. ISSN 09265805. doi: 10.1016/j.autcon.2016.12.002.

Viorica Patraucean, Iro Armeni, Mohammad Nahangi, Jamie Yeung, Ioannis Brilakis, and Carl Haas. State of research in automatic as-built modelling. *Advanced Engineering Informatics*, 29(2):162–171, 2015. doi: 10.1016/j.aei.2015.01.001.

PECI. Portable Data Loggers Diagnostic Tools for Energy-Efficient Building Operations. Technical report, Prepared for the U.S. Environmental Protection Agency and U.S. Department of Energy by Portland Energy Conservation, Incorporated, Portland, Oregon, 1999. URL `https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-19634.pdf`.

Yeritza Perez-Perez, Mani Golparvar-Fard, and Khaled El-Rayes. Semantic and Geometric Labeling for Enhanced 3D Point Cloud Segmentation. In *Construction Research Congress 2016*, pages 2542–2552, 2016.

Norbert Pfeifer, Peter Dorninger, Alexander Haring, and Hongchao Fan. Investigating terrestrial laser scanning intensity data: Quality and functional relations. *8th Conference on Optical 3-D Measurement Techniques*, 2007.

M. Phillips. How to implement a preventive maintenance strategy, 2017. URL `https://www.processingmagazine.com/how-to-implement-preventive-maintenance-strategy/`.

D.M.W. Powers. Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 2(1):37–63,

2011. ISSN 2229-3981. doi: 10.1.1.214.9232. URL `http://www.bioinfopublication.org/files/articles/2_1_1_JMLT.pdf`.

C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Computer Vision and Pattern Recognition (CVPR)*, 2017a.

R. Qi, H. Su, Mo K., and Guibas L. J. PointNET: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017b.

Rongqi Qiu, Qian Yi Zhou, and Ulrich Neumann. Pipe-run extraction and reconstruction from point clouds. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8691 LNCS, pages 17–30, 2014. ISBN 9783319105772. doi: 10.1007/978-3-319-10578-9{_}2.

T. Rabbani. Automatic reconstruction of industrial installations using point clouds and images. *Publications on Geodesy, 62*, 2006.

T Rabbani, F a van den Heuvel, and G Vosselman. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences - Commission V Symposium 'Image Engineering and Vision Metrology'*, 36(5):248–253, 2006. URL `http://www.isprs.org/proceedings/XXXVI/part5/paper/RABB_639.pdf`.

G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

X. Roynard, J.-E. Deschaud, and F. Goulette. Paris-Lille-3D: A Point Cloud Dataset for Urban Scene Segmentation and classification. In *CVPR 2018*, 2018.

Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 2008. ISSN 09205691. doi: 10.1007/s11263-007-0090-8.

P. Santhanam, E. Farchi, and V. Pankratius. Engineering Reliable Deep Learning Systems. In *AAAI Fall Symposium Series on AI in Government & Public Sector*, Washington, DC, 2019. Computers and Society. URL `https://arxiv.org/abs/1910.12582`.

R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, 6 2007. ISSN 0167-7055, 1467-8659. doi: 10.1111/j.1467-8659.2007.01016.x.

Mohammad Mahdi Sharif, Mohammad Nahangi, Carl Haas, and Jeffrey West. Automated Model-Based Finding of 3D Objects in Cluttered Construction Point Cloud Models. *Computer-Aided Civil and Infrastructure Engineering*, 32(11):893–908, 2017. ISSN 14678667. doi: 10.1111/mice.12306.

Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. ISBN 9781538664209. doi: 10.1109/CVPR.2018.00478.

Jungil Shin, Hyunsuk Park, Taejung Kim, and Sang Hoon Hong. Characteristics of Laser Backscattering Intensity to Detect Frozen and Wet Surfaces on Roads. *Journal of Sensors*, 2019. ISSN 16877268. doi: 10.1155/2019/8973248.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012. ISBN 9783642337147. doi: 10.1007/978-3-642-33715-4{_}54.

H. Son, C. Kim, and C. Kim. Knowledge-based approach for 3D reconstruction of as-built industrial plant models from laser-scan data. *International Symposium on Automation and Robotics in Construction (ISARC)*, 756:885–893, 2013.

Hyojoo Son and Changwan Kim. Automatic segmentation and 3D modeling of pipelines into constituent parts from laser-scan data of the built environment. *Automation in Construction*, 2016. ISSN 09265805. doi: 10.1016/j.autcon.2016.05.010.

Mingming Song, Seyedsina Yousefianmoghadam, Mohammad Ebrahim Mohammadi, Babak Moaveni, Andreas Stavridis, and Richard L. Wood. An application of finite element model updating for damage assessment of a two-story reinforced concrete building and comparison with lidar. *Structural Health Monitoring*, 2018. ISSN 17413168. doi: 10.1177/1475921717737970.

Shiyu Song and Manmohan Chandraker. Joint SFM and detection cues for monocular 3D localization in road scenes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 3734–3742, 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298997.

Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298655.

B. Steder, G. Grisetti, and W. Burgard. Robust Place Recognition for 3D Range Data based on Point Features. In *Proceedings of the IEEE Intl. Conf. on Robotics & Automation ICRA)*, 2010.

H. Su, M. Savva, L. Yi, X. Chang, S. Song, Yu F., Li Z., Xiao J., Huang Q., Savarese S., Funkhouser T., Hanrahan P., and Guibas L. J. ShapeNet: An information-rich 3D model repository, 2015a. URL `http://shapenet.cs.stanford.edu`.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015b. ISBN 9781467383912. doi: 10.1109/ICCV.2015.114.

Yun-Ting Su and James Bethel. Detection and Robust Estimation of Cylinder Features in Point Clouds. In *Asprs*, volume 2, pages 887–893, 2010. ISBN 9781617389160.

Surphaser. 3D Laser Scanners. Technical report, 2015. URL `https://www.laserscanning-europe.com/sites/default/files/redakteur_images/Surphaser105HSX.pdf`.

Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, 2015. ISSN 14712342. doi: 10.1186/s12880-015-0068-x.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. ISBN 9781538610329. doi: 10.1109/ICCV.2017.230.

Marvin Teichmann, Michael Weber, Marius Zöllner, Roberto Cipolla, and Raquel Urtasun. MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving. In *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018. ISBN 9781538644522. doi: 10.1109/IVS.2018.8500504.

The American Institute of Architects and Rocky Mountain Institute. Deep Energy Retrofits: An Emerging Opportunity. Technical report, 2013. URL `http://www3.cec.org/islandora-gb/en/islandora/object/islandora%3A1037/datastream/OBJ-EN/view`.

R. Triebel, K. Kersting, and W. Burgard. Robust 3D Scan Point Classification using Associative Markov Networks. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 2603–2608, 2006.

United States Census Bureau. North American Industry Classification System. Technical report, 2012. URL `https://www.census.gov/cgi-bin/sssd/naics/naicsrch?chart=2012`.

U.S. Department of Transportation. Pipeline Incident 20 Year Trends. Technical report, 2013. URL `https://www.phmsa.dot.gov/data-and-statistics/pipeline/pipeline-incident-20-year-trends`.

Ants Vain, Sanna Kaasalainen, Ulla Pyysalo, Anssi Krooks, and Paula Litkey. Use of Naturally Available Reference Targets to Calibrate Airborne Laser Scanning Intensity Data. *Sensors*, 2009. ISSN 1424-8220. doi: 10.3390/s90402780.

Shenlong Wang, Simon Suo, Wei Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep Parametric Continuous Convolutional Neural Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018a. ISBN 9781538664209. doi: 10.1109/CVPR.2018.00274.

W. Wang, R. Yu, Q. Huang, and U. Neumann. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In *Computer Vision and Pattern Recognition*, 2018b.

X. Wang, X. Shen, C. Shen, and J. Jia. Associatively Segmenting Instances and Semantics in Point Clouds. In *CVPR*, 2019a.

Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, 2019b.

L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li. Dense human body correspondences using convolutional networks. In *Dense human body correspondences using convolutional networks*. CVPR, 2016.

B. Wu, A. Wan, X. Yue, and K. Keutzer. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation*, 2018.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 1912–1920, 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298801.

Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Data-driven 3D Voxel Patterns for object category recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 1903–1911, 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298800.

Wai Yeung Yan, Ahmed Shaker, Ayman Habib, and Ana Paula Kersting. Improving classification accuracy of airborne LiDAR intensity data by geometric calibration and radiometric correction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2012. ISSN 09242716. doi: 10.1016/j.isprsjprs.2011.10.005.

J. Yeung, M. Nahangi, Y. Shahtaheri, and C. Haas. Comparison of Methods Used for Detecting Unknown Structural Elements in Three-dimensional Point Clouds. In *ASCE Construction Research Congress 2014*, 2014a.

J. Yeung, M. Nahangi, S. Walbridge, and C. T. Haas. A preliminary investigation into automated identification of structural steel without a priori knowledge. In *31st International Symposium on Automation and Robotics in Construction and Mining, ISARC 2014 - Proceedings*, 2014b. ISBN 9780646597119.

M. Zeeshan Zia, Michael Stark, Bernt Schiele, and Konrad Schindler. Detailed 3D representations for object recognition and modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2608–2623, 2013. ISSN 01628828. doi: 10.1109/TPAMI.2013.87.

J. Zhang, Q. Huang, and X. Peng. 3D Reconstruction of Indoor Environment Using the Kinect Sensor. In *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, pages 538–541, 9 2015. doi: 10.1109/IMCCC.2015.119.

J. Zhang, X. Zhao, Z. Chen, and L. Zhejun. A Review of Point Cloud Semantic Segmentation. *IEEE Access*, 7:179118–179133, 2019. URL `https://ieeexplore.ieee.org/document/8930503`.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing through ADE20K Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.544.

Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *arXiv*, 2017. URL `http://arxiv.org/abs/1711.06396`.

Muhammad Zeeshan Zia, Michael Stark, and Konrad Schindler. Are cars just 3D boxes? Jointly estimating the 3D shape of multiple objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3678–3685, 2014. ISBN 9781479951178. doi: 10.1109/CVPR.2014.470.