CLOI: AN AUTOMATED BENCHMARK FRAMEWORK FOR GENERATING GEOMETRIC DIGITAL TWINS OF INDUSTRIAL FACILITIES

Eva Agapaki[1] and Ioannis Brilakis[2]

[1]Senior Software Developer, PTC Inc.,U.S.A. Email: agapakieva@gmail.com

[2]Laing O'Rourke Reader, Department of Engineering, University of Cambridge, CB2 1PZ, U.K.

ABSTRACT

This paper devises, implements and benchmarks a novel framework, named *CLOI*, that can accurately generate individual labelled point clusters of the most important shapes of existing industrial facilities with minimal manual effort in a generic point-level format. *CLOI* employs a combination of deep learning and geometric methods to segment the points into classes and individual instances. The current geometric digital twin generation from point cloud data in commercial software is a tedious, manual process. Experiments with our *CLOI* framework reveal that the method can reliably segment complex and incomplete point clouds of industrial facilities, yielding 82% class segmentation accuracy. Compared to the current state-of-practice, the proposed framework can realize estimated time-savings of 30% on average. *CLOI* is the first framework of its kind to have achieved geometric digital twinning for the most important objects of industrial factories. It provides the foundation for further research on the generation of semantically enriched digital twins of the built environment.

INTRODUCTION

The industrial sector and especially the oil and gas is an industry with the highest potential growth in terms of worker productivity and economic value of the sector within the next couple of years. The Global Infrastructure Initiative forecasts that heavy industrial buildings and the oil and gas sector are among the construction sectors with the highest potential for investments with an average Compound Annual Growth Rate (CAGR) of 3.4% (McKinsey Global Institute 2015). Therefore, it is crucial that the industrial sector is properly maintained given the high value of the industrial assets for our economies.

Maintenance, safety management and retrofitting are vital operations in the life-cycle of existing industrial facilities. Corrective or poor maintenance incurs unplanned downtime costs, which are estimated to be $50 billion per year (National Institute of Standards and Technology 2018). The primary reasons for these incidents are ineffective and inefficient facility management and poor mapping of the existing industrial equipment. Faster digital industrial documentation is urgently required to reduce unscheduled equipment downtimes and boost the Overall Equipment Effectiveness (OEE) of a factory, which is currently estimated to be between 5 to 20% (PECI 1999).

There are limits on the acceptable shut down duration that will not impede production. These limits cannot be violated without incurring extra costs. This is why adoption of Digital Twins (DTs) is crucial for the industrial sector. The greatest value of using DTs is that they are projected to save substantial costs for facility managers by automating the preventive maintenance process which will enable accurate positioning of each industrial object and timely maintenance decisions. For example, DTs can help to keep records of the inventory, processes, historical data and additional equipment. This allows owners to identify inefficiencies and ways to address them.

There are four maintenance strategies that a factory owner can follow to prevent damages. These are reactive, planned, proactive and predictive maintenance (Coleman et al. 2017). Each maintenance strategy is measured with the OEE metric with the highest OEE being for a predictive maintenance strategy, which indicates an effective strategy. OEE is low for reactive maintenance, since potential damage caused to machines can deteriorate the machines' condition, hence maintenance costs will be higher and unplanned downtime of a factory will affect performance. Planned maintenance can also have increased replacement costs over time and there is an implied need of storing additional spare parts in the factory's inventory. Proactive maintenance treats the root cause of the problem, ultimately reducing costs without impeding production. Predictive maintenance uses historical data of equipment and production units to predict when they are likely to fail. These measures can reduce machine downtime by 30% to 50% and increase the machine life by 20 to 40% (Dilda et al. 2018).

Predictive maintenance is where a DT would be most helpful for predicting failures using real-time factory space and sensor data. DTs have the potential to automated the preventive maintenance process which will enable accurate positioning of each industrial object and timely maintenance decisions. Studies show that the wider adoption of DTs will unlock 15-25% savings to the global infrastructure market by 2025 (Barbosa et al. 2017; Gerbert et al. 2016).

The concept of DTs is not new. NASA first generated the term "twin" when building two identical space vehicles for its Apollo program (Glaessgen and Stargel 2012). The modern terminology of a "digital twin" has been attributed to Dr Michael Grieves as part of his research in Product Lifecycle Management (PLM) (Grieves 2014). Reports based on the digitization index have shown that the oil and gas industry has been highly digitised as compared to the construction industry, which is in the bottom of the list (Agarwal et al. 2016). Despite the high value DTs have in the industrial sector, yet, industrial facilities do not have DTs for existing industrial factories due to the high perceived cost which outweighs their benefits (West and Blackburn 2017).

The generation of a geometric Digital Twin (gDT) is the core and first step in the DT generation (Borrmann and Berkhahn 2018). The inputs for the generation of gDTs are usually point clouds scanned with Terrestrial Laser Scanners (TLS) (Marshall 2016). 90% of the gDT generation cost is spent on converting point cloud data to 3D models due to the sheer number of objects of each industrial facility (Fumarola and Poelman 2011; Hullo et al. 2015). Hence, cost reduction is only possible by automating the generation of gDTs. However, automatically classifying millions of objects is a very hard classification problem due to the very large number of classes and the strong similarities between them. We provided in our previous work (Agapaki et al. 2018) a comprehensive technical assessment and viable evaluation of existing state-of-the-art software tools available. In the following paragraphs, we summarize the state-of-practice based on this evaluation.

State-of-practice

In our previous work (Agapaki et al. 2018), we identified the most frequent and laborious to model object types, which are cylindrical objects (straight pipes, electrical conduit and circular hollow sections), valves, elbows, I-beams, angles, channels and flanges. Cylinders require 80% of the total modelling time of the ten most important object types in EdgeWise (ClearEdge 2019) and represent 45.5% of the total number of objects in an industrial plant on average. EdgeWise was selected compared to other state-of-the-art software, because it is the only commercially available tool that attempts to automatically extract cylinders from the point cloud of an industrial plant without significant user assistance. EdgeWise has significantly accelerated 3D modelling of industrial plants according to the findings discussed above. However, it has some limitations, which can be summarized as follows:

1. Structural elements (I-beams, angles, channels) should be manually modelled and their location in the point cloud is roughly defined based on the modeler's discretion.
2. Segmentation of cylinders has been partially achieved with detection rates being 75% recall and 62% precision on average (Agapaki et al. 2018). The same metrics for cylindrical objects labelled as pipes are 58% and 47% respectively. It is also important to note that EdgeWise erroneously includes points that do not belong to a geometric shape. This is due to fitting errors, which occur since primitive shapes are perfect shapes, whereas the scanned, physical objects are imperfect (e.g. a cylindrical pipe may be bent).
3. EdgeWise is not designed to output geometric shapes in an open and generic format. As such, modelers cannot easily exchange data between different operational-phase gDT platforms due to data inconsistency between them.

Therefore, the evaluation of EdgeWise uncovered (a) the substantial performance of this software in detecting cylinders with its pitfalls, (b) the inability of the software to (i) further classify cylinders into conduit or pipes or CHSs and (ii) detect and further classify I-beams, channels, flanges, valves and angles in spite of their high frequency in an industrial facility.

This performance of EdgeWise has substantial room for improvement and this paper intends to address the above-mentioned limitations in order to automatically generate gDTs of industrial facilities and assist the tedious current practice. We propose a geometric twinning framework for existing industrial facilities and bench-mark it with the current state of practice. The goal of this research is to devise, implement and benchmark a novel framework that can accurately generate individual labeled point clusters of the most important shapes of existing industrial facilities with minimal effort in a generic point-level format. In the following section, the state-of-the-art research methods related to the above-mentioned limitations are presented. We then outline the framework in the proposed solution, which is followed by the experiments and results. The conclusions are then derived in the last section.

BACKGROUND

We first investigated gDT generation strategies in the literature and grouped them into **S1** and **S2** strategies in Figure 1. The **S1** strategy is composed of (a) primitive industrial shape detection and (b) fitting. The **S2** strategy includes (a) class segmentation, (b) instance segmentation and (c) fitting. Class segmentation describes the task of partitioning a set of measurements in the 3D point cloud space into smaller, coherent and connected subsets, which are called classes (Li et al. 2019a). Classes represent objects with common geometric characteristics such as cylinders, elbows, I-beams, valves, flanges, angles and channels. Instance segmentation assigns a label per point based on the individual object that the point belongs to. Shape detection is the procedure of identifying the location and geometric properties of instances of semantic objects that belong to a certain class. Detected individual objects are usually represented by a bounding box containing the object. 3D object fitting is the process of assigning instance point clusters to geometric representations.

The main reasons for selecting the **S2** gDT strategy in this paper are:

1. Class segmentation directly processes and labels the TLS point clouds without converting 3D points into other geometric representations such as bounding boxes. On the other hand, **S1** detection methods need to convert 3D points to geometric primitive shapes.

2. Primitive fitting RANSAC-based methods cannot properly detect samples with points that are closely located to each other (Liang et al. 2018; Li et al. 2019a; Agapaki and Brilakis 2020a).
3. Another merit of the **S2** gDT strategy is that classes are easily separable for further processing and information is not lost due to data conversions. **S1** strategy methods rely on point cloud fitting to generate standardized geometric representations. **S1** methods similar to those of (Jin and Lee 2019; Patil et al. 2017; Rabbani 2006; Liu et al. 2013; Kawashima et al. 2014; Schnabel et al. 2007; Son and Kim 2016) are out of the scope of this work.

Therefore, the literature review is elaborating on: (a) **S2** class segmentation methods and (b) **S2** instance segmentation methods. Fitting methods are not discussed, since they are out of scope of this paper.

Class segmentation

Class segmentation methods applied on industrial shapes have been widely investigated. We categorize them into three groups: (a) attribute based methods, (b) machine learning and (c) deep learning methods. A comprehensive review of class segmentation methods based on hand-crafted features is provided by (Agapaki and Nahangi 2020) and some of the most important methods are explained in the paragraphs that follow.

Attribute-based Attribute-based methods are bottom-up approaches that cluster base elements to generate complex systems in successive higher levels until a top-level system is formed (e.g. bridge, facility) (Borenstein and Ullman 2008). These methods cluster points with similar attributes into subsets. An n-dimensional attribute space is created to extract the attributes in the parameter domain, where n represents the estimated number of attributes. These methods process a point cloud starting from point-wise features and generate higher-level features, such as surface normals (Rusu et al. 2009; Sampath and Shan 2010), mesh (Marton et al. 2009) or patches (Vosselman 2009; Zhang et al. 2015). Attribute-based methods group points based on the similarity of low-level features such as color, curvatures, roughness, density or surface normal vectors. The estimated attributes are clustered and extracted in the parameter domain. Attribute based methods can be divided in two broad categories based on the shape descriptors they use: global or local. Local descriptors allow for partial matching of features, therefore are preferred for occluded scenes compared to global descriptors. Global descriptors describe the scene as a whole. For instance, local descriptors of a cylinder are curvature and normal vectors, whereas global descriptors are its length and diameter, which correspond to properties for the whole cylinder. Curvature has been extensively used as a local feature for industrial piping segmentation (Dimitrov and Golparvar-Fard 2015; Perez-Perez et al. 2016). However, substantial manual segmentation is needed to pre-process the input TLS data. (Xiong et al. 2013) uses local geometric features as well as contextual relationships between point clusters to segment planar segments as wall, floor and ceiling. However, their method cannot be applied to more complex shapes such as industrial shapes.

Machine learning We review one of the most widely used parametric supervised machine learning methods in the class segmentation literature, which is **Support Vector Machines (SVMs)**. (Li et al. 2016) used SVMs on TLS urban point clouds and then a multi-classification graph-cut algorithm to optimize the initial segmentation result. Similarly, (Zhang et al. 2013) used a region-growing algorithm before applying an SVM for urban point cloud segmentation. (Huang and You 2013) and (Armeni et al. 2016) use SVM classifiers with local features to segment cylindrical and indoor space objects. The use of SVMs in these approaches though has inherently two limitations: (1) SVM is not designed for imbalanced classes. Weights inversely proportional to the class frequency are applied to the imbalanced classes. Industrial facility datasets are highly imbalanced with respect to the most important object types they have, since their distribution follows the Zipf's law as proved in (Agapaki et al. 2018). For this reason, the application of SVMs on TLS industrial facility data is not preferred, unless one oversamples the object types that appear less frequently. (2) the success of SVMs depends on the selection of hand-crafted features, the type of kernel function and the parameters to the kernel function. Improper selection of features can result in misclassifications, whereas application of different kernel functions for a dataset gives different results.

Other popular methods that are widely used for class segmentation of the built environment are Conditional Random Fields (CRF) and Markov Random Fields (MRF). (Perez-Perez et al. 2016) uses a MRF method together with a CRF to distinguish geometric and semantic attributes of point cloud clusters for ceiling, floor, wall and pipe categories. Pipes are segmented at 79% precision and 3% recall. Low recall rates and manual pre-processing of the point cloud data are the main limitations of this study. (Bassier et al. 2019) and (Perez-Perez et al. 2021) used the combination of geometric (AdaBoost classifier) and contextual features (SVM classifier) to segment floors, ceilings, roofs, beams, walls and clutter of indoor buildings.

3D Class Segmentation Deep Learning methods CNNs have been widely used for a variety of tasks in image segmentation (Krizhevsky et al. 2012; LeCun et al. 2008; Taha and Hanbury 2015; Pang et al. 2012; Wang et al. 2018a; Teichmann et al. 2018). We group these methods in three main categories as suggested by (Wang et al. 2019b): (**DLa**) view-based (Su et al. 2015; Kalogerakis et al. 2017; Wei et al. 2016), (**DLb**) volumetric (Maturana and Scherer 2015; Wu et al. 2015; Zhou and Tuzel 2017; Klokov and Lempitsky 2017; Tatarchenko et al. 2017) and (**DLc**) geometric deep learning methods (Qi et al. 2017b; Qi et al. 2017a; Wang et al. 2019b).

DLa methods represent objects in 3D space as a collection of 2D views. These 2D projected views are then processed by applying standard Convolutional Neural Networks (CNNs). A CNN is applied to each 2D view and then the features are aggregated using max pooling (Su et al. 2015). Recently, their results were refined by aggregating the predicted 2D projections of 2D onto the 3D shapes through a CRF (Kalogerakis et al. 2017). This technique is more useful for data acquired by RGB-D sensors since a single view can be processed at a time (Wei et al. 2016) rather than TLS point clouds. The main drawback of these methods is that they cannot be applied if the input is noisy and/or incomplete. Therefore, these methods are not further investigated for TLS point cloud industrial applications.

A key difference of 3D data compared to image data is that adjacent 3D points in the Euclidean space are not necessarily correlated. There are two widely used volumetric CNN methods that can accommodate for this. These are voxelization and octree based CNN methods. *Voxelization* is a technique to convert the unstructured geometric data to a regular 3D grid, so that standard CNN operations can be applied (Maturana and Scherer 2015; Wu et al. 2015). The main drawbacks of voxelization is the complexity of the network that does not take into account the sparsity of data. For instance, the 3DShapeNET (Wu et al. 2015) for a small 3D voxel input of 30x30x30 has $30x30x30 = 27,000$ parameters and becomes prohibitively large for larger voxel inputs. A solution to this is anisotropic probing that only selects parts of images to train and results in a very low number of parameters and low computational cost (Su et al. 2015). Another solution would be to only store the occupied voxels and constrain the computations near the surface of the 3D object. Voxelization produces a sparse grid, so the volumetric representation is not often useful and results in loss of data. Voxel based object detection (Zhou and Tuzel 2017) and class segmentation has the limitation of high memory usage which is dependent on the 3D voxel resolution. 3D space partitioning methods (k-d trees or octrees) (Klokov and Lempitsky 2017; Tatarchenko et al. 2017) do not consider local geometry.

The second set of volumetric methods uses *octrees* which recursively partition the 3D space and label each voxel according to object occupancy (Meagher 1980) with internal nodes having exactly eight children. Octree based methods do not compute features per individual 3D point. Rather, they process cubes of data based on a voxel data structure. The convolutions on octrees are performed using hash tables that only search around neighbourhoods (Shao et al. 2018). Octree CNN methods tend to be memory efficient in comparison to voxel-based CNN methods, but they still require substantial time to train. Yet, octree size determination highly depends on the TLS point density and the desired level of detail (LOD) of output point clusters. Therefore, these methods are not suitable for the class segmentation of industrial facilities and are not further analyzed.

The unstructured nature of point clouds hinders the use of convolutions between 3D points, unlike images where 2D convolutions can be applied on pixels. **DLc** methods solve this challenge by directly processing point cloud data in 3D space. Geometric deep learning methods are chosen as the most suitable for class segmentation as explained by (Agapaki and Brilakis 2020a), since they address the following challenges that TLS industrial point cloud processing has: (1) irregularity in the TLS data structure, (2) TLS data sparsity, noise, presence of outliers and occlusions as well as density variations especially in industrial settings and (3) differences in industrial object scales, rotation and translation variant objects as well as geometric similarities between objects of the same class. PointNETs (Qi et al. 2017b; Qi et al. 2017a) and their derivatives (Wang et al. 2019b; Wang et al. 2018b; Landrieu and Simonovsky 2018; Thomas et al. 2019) have solved these challenges by applying permutation invariant functions as well as local 3D filters in their network architectures. PointNET networks concatenate global and local features into point feature vectors based on which class labels are predicted. PointNET++ improves the PointNET architecture by adding local neighbourhood geometric features.

These networks and their derivatives ave been extensively used to automatically extract geometric features from point clouds and segment indoor objects, openings (e.g., doors and windows), and structural components (e.g., beams, ceilings, columns, floors, and walls) (Komori and Hotta 2019; Wang et al. 2019a; Liang et al. 2019; Peyghambarzadeh et al. 2020; Lu et al. 2020; Li et al. 2019b; Ma et al. 2020).

The advantages and limitations of each category of methods discussed above are summarized in Figure 2.

Instance Segmentation

3D instance segmentation is based on 3D geometric class segmentation networks. These methods can be grouped into shape-based (top-down), shape-free (bottom-up) or class-agnostic (bottom-up). Our readers can refer to (Agapaki and Brilakis 2020b) for a comprehensive literature review of each of these methods. We elaborate on the state-of-the-art literature on shape-free methods, since these are more suitable for the generation of gDTs from TLS industrial data (Agapaki and Brilakis 2020b).

Shape-free methods are based on deep learning networks, which aggregate features per point and output instance labels per point given a similarity matrix between pairs of points (Wang et al.

2018b; Wang et al. 2019b) or embedding another network measuring point-wise distances (Pham et al. 2019). PointNET (Qi et al. 2017b) and PointNET++ (Qi et al. 2017a) are the backbone networks for these methods, meaning that they achieve class segmentation as well. Although these networks take into consideration the local neighbourhoods of points, they cannot explicitly define the boundaries of complex industrial shapes. Object boundaries can be taken into account by considering the class and instance segmentation labels. (Chen et al. 2019) use a graph-based instance segmentation method in combination with PointNET to classify points. They addressed oversegmentation by using a component merging approach based on the object classes, normals of each point and contextual relationships of certain objects such as walls. Their method is specific to indoor building scenes and cannot be generalized to industrial objects. (Liu et al. 2021) use a CNN that learns to correlate geometric and color information in order to determine instance boundaries. However, color does not assist segmentation in industrial settings (Agapaki and Brilakis 2020a). The readers can refer to (Xie et al. 2019) for a detailed review of all the instance segmentation methods.

Another category of recent bottom-up instance segmentation methods are class-agnostic methods. (Chen et al. 2021) propose an instance segmentation method (LRGNet) and validate it on popular indoor and outdoor point cloud datasets (S3DIS, ScanNet and KITTI dataset). They use a deep learning network to optimize their region growing method without assigning class labels to points. LRGNet is agnostic to objects of any class and geometry, however it lacks to generate the class and instance segmentation labels per point cluster. This translates to either additional labor hours or additional processing of the segmented clusters (e.g. training a classifier) to assign those labels that are needed for the gDT generation.

PROPOSED SOLUTION

We target to solve the problem of the generation of gDTs of existing industrial facilities with respect to cost and modelling time reduction. The main objective of this paper is to develop a benchmark framework as the foundation for future research.

Overview

The proposed framework consists of two major parts. Specifically, these parts are **(1) class segmentation** and **(2) instance segmentation** that intend to answer the research questions as outlined in the Background section and aim to outperform the existing state of practice and research in the industrial modelling space.

We propose a novel hybrid framework which develops deep learning networks and leverages their detected outputs with industrial engineering knowledge, in order to automatically extract labelled point clusters corresponding to industrial shape components without generating surface primitives (**class point clusters**) and then to efficiently detect individual industrial shapes from the labelled point clusters (**instance point clusters**).

Real-world industrial environments are more challenging than buildings that have been extensively studied and scanned in previous research efforts as mentioned in the Background section. Industrial components do not comply with a universal colour scheme, rather colours depend on each manufacturer's specifications (Agapaki and Brilakis 2020a). There are significant challenges to be addressed when segmenting industrial objects. Previous class and instance segmentation methods rely on color information to segment building components, however industrial object colors change based on each manufacturer. This makes color information an inconsistent feature to use when processing industrial point clouds. The dimensions of industrial facilities as well as lack of contextual rules between shapes differentiate them from indoor buildings, which are structured based on rooms. In other words, the relative location of a cylinder in a facility does not imply that the locations of these objects should comply to specific spatial rules, however the relative location of an elbow and a pipe are strongly correlated. We propose a 3D-slicing facility window method, CLOI-NET-class based networks and CLOI-Instance graph-connectivity algorithms to tackle these challenges. The 3D windows are used to segment the TLS dataset in non-overlapping parts, so that a portion of these windows will be used for training. These windows should be non-overlapping, so that the training and test set are disjoint. These algorithms are the core foundation of the methods built upon them to enhance the segmentation and detection results. The proposed algorithms can deal with the challenges outlined above and can accurately detect the majority of *CLOI* industrial classes.

The outputs of the CLOI framework are both class labels (e.g. cylinder, elbow, valve) and instance labels (e.g. cylinder 1, cylinder 2, valve 3). Subclassification of cylinders to pipes, circular hollow sections, handrails and electrical conduit is beyond the scope of this work. The proposed algorithms address scale variance. The algorithms are scale invariant, since we feed them with objects at different scales from a few centimeters to some meters and there are intra-class variations. For instance, there are many types of valves as expressed above, which are grouped in one class and the proposed algorithms should be able to segment valves of all the above mentioned categories.

We illustrate the developed hybrid framework in Figure 3. It consists of two major processes: **Process 1**, class segmentation of *CLOI* industrial point clusters, and **Process 2**, instance segmentation of *CLOI* industrial geometric shapes from point clusters.

The proposed framework starts with a raw, laser-scanned, Point Cloud Dataset (PCD) of an existing industrial facility (data format: points in .pcd, .txt, .las, .xyz). External noise such as vegetation, adjacent buildings is removed using commercial software as explained in (Agapaki and Brilakis 2020a). The industrial PCD contains *CLOI* geometric shapes and any other industrial shapes inside a factory (data format: points in .pcd, .txt, .las, .xyz). The first step of the framework is to automatically split the PCD facility in 3D windows and the 3D windows in "3D blocks". Then, the 3D blocks are aligned in the global coordinate system. As such, the outputs of this step are 3D block PCDs (data format: points in .pcd, .txt, .las or .xyz). Then, we manually annotate industrial facilities to generate a benchmark dataset and the outputs of this step are class and instance segmentation labels and points. It is important to note that this is an essential offline step needed for training purposes and serves as the ground truth for the validation of the framework.

Next, we propose a three-step class segmentation method (Process 1) to segment the *CLOI* point clusters from the 3D blocks. The final outputs of this process are seven industrial shapes, namely cylinders, elbows, channels, I-beams, angles, flanges and valves, in the form of labelled point clusters (data format: points in .pcd, .txt, .las, .xyz). Then, we suggest an optimal manual annotation (if the users select it) to remove the erroneous point clusters maintained from Process 1 followed by proposing an efficient instance segmentation method (Process 2) through which the seven *CLOI* classes (in point cluster format) can be directly segmented to individual shapes. The final outputs of this process are point data corresponding to the points, class and instance labels per point. We elaborate on each process in the following sections.

We validate Process 1 on the *CLOI* benchmark dataset (Agapaki et al. 2019), which is composed of four laser scanned industrial facilities. The original number of laser scanned points, the number of instances, the area and the manual labor hours to manually annotate (with class and instance labels) each facility are documented in Figure 4.

Process 1: CLOI-NET-Class segmentation

The methods of Process 1 bypass the stage of surface generation altogether and directly output segmented and labelled point clusters. The 3D window parsing method breaks down the whole industrial facility into subset windows for more efficient processing. The key insight behind Process 1 is to formulate a high dimensional feature space to automatically assign labels per point so that the target point clusters can be quickly located in the point cloud.

The inputs of Process 1 are the 3D coordinates of the TLS point cloud. The outputs are segmented point clusters that are labelled based on the class they belong to. The main steps of the method include (1) 3D space parsing into smaller blocks, (2) geometric deep learning segmentation with PointNET++ SFR and (3) further refinement of class labels using contextual rules. **Step 2** allows the user to use passive (no test data used in training) or active (test data used in training) learning with the goal to minimize the manual annotation time. **Step 3** is composed of three processes. These are: (a) a cylinder classifier to segment cylinder with diameter greater or equal to 1m using curvatures, (b) steel shape (angles, I-beams, channels) segmentation based on computation of normals and (c) class label confidence adaptation to correct misclassified class labels from previous steps. Details of our methodology, named *CLOI-NET-Class*, can be found in (Agapaki and Brilakis 2020a).

Process 2: CLOI-Ins instance segmentation

The inputs of Process 2 are the predicted point clusters from the CLOI-NET-Class method for the evaluation of the proposed framework. The same 3D block generation method from Process 1 is used for segmenting the input data. The outputs of this process are point-wise instance labels (individual point clusters of *CLOI* classes).

Process 2 consists of two major steps: Step 1 predicts an instance label per point by using a graph-based method, namely Breadth First Search (BFS) that was originally introduced by (Bauer and Wössner 1972). Step 2 is a boundary segmentation method that is used to enhance the instance segmentation results of Step 1. An assumption of the method is that the initial TLS industrial data is partitioned in 3D non-overlapping sliding windows with overlapping 3D blocks. The outputs of Step 1 are connected components based on connectivity relationships in order to segment the instances as output. The boundary segmentation method in Step 2 outputs binary labels on whether a point is a boundary point or not. These instance point clusters present industrial shapes at Level of Detail (LOD) 300.

The novelty of Process 2 in isolation is two-fold:

1. the efficiency of the BFS algorithm by applying it on the entire PCD and connectivity between points
2. the intelligence of the boundary segmentation method to account for boundary points and robustly process points in small regions.

Readers can refer to (Agapaki and Brilakis 2020b) for details of the CLOI-Ins instance segmentation process.

In our previous work, we have validated the performance of the framework's processes in isolation, to ensure that they are likely to contribute to the performance of the CLOI framework as a whole. The CLOI-NET-Class and CLOI-Ins segmentation processes are integrated together and evaluated as a complete framework in this paper. The novelty of the CLOI framework is to prove that it requires competitively less manual segmentation time compared to current state-of-practice.

The novelties of this work are: (1) the experimental evaluation of the integrated CLOI-NET Class and CLOI-Ins segmentation processes, (2) showing how the parameters of the instance segmentation method change when predicted class labels are used instead of ground truth class labels, (3) proving that instance segmentation performance is correlated with class segmentation performance or in other words that good class segmentation performance is crucial for achieving good instance segmentation performance and (4) evaluating the time-savings of the complete CLOI-framework and comparing it to the current state-of-practice.

EXPERIMENTS AND RESULTS

Implementation

We validate the CLOI framework on the first industrial dataset of class and instance labelled point clusters, *CLOI*, (Agapaki et al. 2019). Figure 4 summarizes the four industrial datasets used for validation. We tested our framework on indoor and outdoor industrial scenes which include four industrial facility types: a warehouse, a petrochemical plant, an oil refinery and a processing unit. We used the class and instance labels of this dataset as ground truth annotations. *CLOI* consists of 12,497 shapes and 7.1 billion labelled points. Detailed statistics and scanner specifications of the data can be found in (Agapaki et al. 2019).

Two research platforms were developed for the framework validation; one capable of high computing for training deep neural networks and one for visualisations of large scale TLS industrial datasets. Training of the CLOI-NET-Class method was performed on Google Cloud instances. We developed a proof of concept prototype and implemented the CLOI framework on Tensorflow 2.0. We used a Google Cloud VM with NVIDIA Tesla P100 GPUs to run our experiments and visualized the point clouds and segmentation outputs of the CLOI framework on the CLOI platform that we developed. This platform is built on the Potree Viewer (http://potree.org/) and a demo is available on Youtube (https://youtu.be/K3rnBctMYAU). Potree is built upon ThreeJS and allows for rendering of large point clouds in a WebGL web browser (Schuetz 2016; Devaux et al. 2012). We created the user interface to select the TLS dataset of a *CLOI* facility, then segment the *CLOI* classes and validate with the ground truth class labels. The user can also select a point and only view the points associated with that *CLOI* class. Further details about the implementation of Process 1 and Process 2 can be found in (Agapaki and Brilakis 2020a) and (Agapaki and Brilakis 2020b) respectively.

Manual annotation

The *CLOI* dataset was generated by manually annotating the four industrial facilities. The Ground Truth (GT) datasets are the desired outputs to compare against those generated by the proposed methodology and also used for training. The following GT datasets were created for the *CLOI* dataset validation.

GT class: A given industrial facility, TLS scanned, point cloud input is segmented into the eight *CLOI* classes. Each individual point was assigned a class point-wise label. Figure 4 shows each *CLOI* facility coloured with one of the eight class labels and the manual annotation time involved to generate the GT per facility. The number of shapes (instances), original number of 3D points and the area per facility are also provided. One can distinguish that even if a small facility area is scanned, the density of the scans may be so high that the number of points is much higher compared to a sparsely scanned facility. For instance, the *oil refinery* is only $300m^2$, making it the smallest facility of the dataset, but it has the largest number of surveyed 3D points.

GT instance: A given point cloud input is assigned to an individual instance point cluster.

GT boundary: A given point is classified as a "boundary" point if there is more than one instance in a neighbourhood of radius $4cm$ around it. The data structure used to define the neighbourhoods around each point is a kDTree.

Experiments

The performance of the framework was evaluated based on the performance of the *CLOI-NET-Class* segmentation method (Process 1) and the *CLOI-Instance* segmentation method (Process 2).

We evaluate our CLOI-NET class and CLOI-Ins segmentation methods on each of the four CLOI datasets using a k-fold cross validation strategy. Therefore, each facility is evaluated separately and the trained facilities are disjoint from the test facility. For example, in order to evaluate performance on the oil refinery dataset, we trained on the petrochemical facility, the warehouse and the processing unit point cloud datasets.

The inputs of the proposed framework are the class segmented clusters of Process 1. The average accuracy and mean Intersection over Union (mIoU) of the class segmentation experiments from (Agapaki and Brilakis 2020a) was 79.8% and 44.65% respectively. The training and test sets are disjoint. In other words, we trained on all the *CLOI* facilities (three facilities used for training on every experiment) except the one of interest to segment that is used for validation. We validated the theoretical active learning model as outlined in (Agapaki and Brilakis 2020a). Results showed that the total cost annotation function and the validation accuracy follow the theoretical model and the optimal data pre-annotation percentage that minimized the total annotation cost is between 20±10%. The CLOI-NET-Class performance following the active learning approach had on average 15% higher accuracy than the passive learning approach.

The performance of Process 2 (CLOI-Ins segmentation) was 73% mPrec and 71% mRec on all *CLOI* facilities using the ground truth class labels as inputs (Agapaki and Brilakis 2020b). For the evaluation of the framework, we compared the state-of-the-art instance segmentation networks (SGPN (Wang et al. 2018b; Wang et al. 2019b)), the BFS algorithm and the proposed *CLOI* framework in Table 1. The results illustrated in Table 1 show that SGPN has very low performance on the oil refinery data with the ASIS network performing better in all efficiency metrics. The oil refinery is used to compare the state-of-the-art deep learning instance segmentation networks, the BFS algorithm and the *CLOI* framework methodology. For the application of the BFS algorithm, the minimum instance size was selected for the predicted *CLOI* class point clusters based on performance. Therefore, the author conducted experiments to determine the minimum instance size based on the performance in terms of precision and recall on the *CLOI* datasets. The performance (precision and recall) was measured after running the BFS algorithm for a different minimum instance size. The minimum instance sizes tested are 0, 100, 200, 300, 400 and 500 and the results are presented in the curves of Figure 5. The results in Figure 5 illustrate that the optimal trade-off between precision and recall is for minimum instance size 200 points instead of the minimum instance size of 20 points that was computed based on the ground truth class segmentation labels (Agapaki and Brilakis 2020b). This is attributed to noisy predicted class labels compared to the ground truth class labels used to evaluate Process 2 independently. There is an exception for the minimum instance size (μ) and the minimum neighbourhood size (E) for the case of cylinders. The results indicate to set the instance size at 50 points and the minimum neighbourhood size (E) at 3cm (instead of 4cm) only for cylinder instance point clusters due to the observation that cylinders have higher class segmentation label predictions and the CLOI-Instance methodology benefits from that. We also observe in Table 1 a 10% increase in precision due to the *class boundary* constraint on the BFS algorithm for a minimum neighbourhood of 4cm.

Given a set of predicted instances and a set of ground truth instances, the performance of instance segmentation is measured in the following way, which is standard in instance segmentation literature: For any predicted instance $pred_i$ and ground truth instance gt_j we say that they are matched if $(pred_i, gt_j) >= 0.25$, where IoU is the number of common points in $pred_i$ and gt_j, divided by the total number of points in $pred_i$ and gt_j. Then, precision is defined as the percentage of predicted instances of some class c, that are matched to some ground truth instance of the same class. Similarly, recall is defined as the percentage of ground truth instances of some class c, that are matched to some predicted instance of the same class.

We present these results for the oil refinery dataset as an example for comparison of the best performing existing instance segmentation methods and the proposed *CLOI* framework. The illustrated results in Table 1 and 2 demonstrate that the CLOI-Instance methodology clearly outperforms the current state-of-the-art research.

Another important note is that the CLOI framework results are calculated assuming that the users pre-annotate X% of the test facility with X% being the value from Table 3 depending on the facility. These percentages are based on the active learning curves of (Agapaki and Brilakis 2020a). Then, we present the precision and recall per *CLOI* class and the average precision and recall curves in Figure 6 as a reference. The results for the other three facilities are included in the Appendix. It is evident that for all datasets the recall metric of all the *CLOI* classes outperforms the precision metric for all the IoU threshold values. The greater difference between the mean precision and mean recall is for the oil refinery (Figure 6(c)), which is attributed to the high complexity of this dataset, the large number of highly occluded conduits and the large number of connected I-beams. This leads to reduced performance for all classes. Although the CLOI-Instance proposed methodology has promising results compared to the state-of-the-art methods for the instance segmentation task, the results demonstrate that the predicted class labels significantly reduce the precision and recall metrics compared to the same results presented given the ground truth class labels (Agapaki and Brilakis 2020b).

We observed the following based on Table 4 and Figures 6, 7, 8 and 9. Missed instances are attributed to the large number of conduits that are placed inside cable trays and also the large number of pipe junctions where the boundaries cannot be clearly defined between instances of the same class (i.e. cylinders) for all facility types. Another interesting observation is that the recall of valves in the petrochemical facility (91.7%) is due to instances being separated from each other and the likelihood of encountering two valves adjacent to each other in the petrochemical dataset being very low. The most frequently encountered types of valves in this dataset are hand-wheel ball valves and check valves or the sequence of a ball valve and a gate valve are grouped together in the same instance. A reason for the reduced precision of the warehouse valves (29.4%) is the over-segmentation of the hand-wheel parts of gate valves. This can be attributed to occluded connections between the hand wheel and the body of the valves or the steam and the hand wheel part. The performance of flanges in the processing unit dataset is very low compared to the other facilities (14.3% Prec and 0.5% Rec), which is due to the prevalence of weld neck flanges that have not clearly defined boundaries with pipes. This results in grouping part of the pipe and flange in one instance. A similar case is for connections between flanges with threaded rods and pipes. Also, there are cases where there are instances directly connected to concrete slabs and the floor, such as pumps or other equipment. For those cases, all points are grouped in one instance point cluster. Also, there are cases where there are instances directly connected to concrete slabs and the floor, such as pumps or other equipment in all facilities. For those cases, all points are grouped in one instance point cluster, leading to under-segmentation. A possible improvement of the method could be to obtain more accurate point clouds in areas close to object connections (i.e. I-beam connections, pipe to flange or valve connections).

The CLOI framework performance of cylinders is relatively high across the *CLOI* facilities given their high class segmentation performance (Agapaki and Brilakis 2020a) for all the IoU threshold values. We remind the reader that the cylinder class segmentation performance was 81.25% precision, 81.75% recall and 68.25% IoU on average. There are though some cases where the cylinder instance point clusters are over- or under-segmented. These cases are the **Cyl** cases presented in (Agapaki and Brilakis 2020b). The results of the *CLOI* framework show an additional pain point. This is the uncertainty of the CLOI-NET-Class segmentation on predicting the class labels of the points. This leads to erroneous instance label predictions and mostly impacts the *CLOI* classes that have low class segmentation performance (the reader can refer to (Agapaki and Brilakis 2020a) for a detailed discussion). Figure 11(e) shows an example where the highlighted I-beam is correctly segmented when ground truth labels are used. However, the evaluation of the *CLOI* framework shows that incorrect prediction of the class labels of the I-beam point cluster (Figure 11(d)) leads to incorrect instance segmentation of the same I-beam (Figure 11(f)).

The class and instance segmentation recall were plotted in Figure 12 in order to investigate the impact of low class segmentation recall on instance segmentation. Figure 12 shows that class and instance segmentation for all CLOI classes are highly correlated; the higher the class segmentation recall, the higher instance segmentation recall is. Channels of the warehouse dataset are an exception, since the recall rate of channels on instance segmentation is low (34.6%) compared to their class segmentation recall (91%). In other words, the instance segmentation performance is not explained by the linear correlation plot. This can be attributed to channels being in close proximity (parts of stairs or roof steel members) in the warehouse dataset, which impacts the BFS algorithm and subsequently their instance segmentation recall. Another reason of the low recall metric is the connectivity of strut channels to conduit. In this case, cylinders and channels are erroneously predicted as one instance.

Another achievement of the *CLOI* framework is that it correctly segments sub-instances of an instance point cluster that has the "other" class label and even outperforms the manual instance segmentation in cases where a ground truth instance is under-segmented (Figure 10(a) and Figure 10(b)). This particularly applies for instances close to the floor or roof of a facility. The superior performance of the *CLOI* framework is attributed to the connectivity information that the BFS algorithm uses to segment instances. Another case where the *CLOI* framework outperforms the manual instance segmentation is for sequences of pipe components that have different radii. An example of that is Figure 10(c) where the *CLOI* framework correctly segmented the cylinder from a pump and a flange with steel rods.

We then recommend to use the 25% IoU threshold that gives slightly improved results (50% mPrec and 35.3% mRec for all the *CLOI* facilities). The *CLOI* classes that have significantly higher metrics are those with higher class segmentation results as explained above. These are cylinders (53.6% mPrec and 44% mRec), elbows (66.8% mPrec) and I-beams (63% mPrec and 64.3% mRec).

Time savings in Geometric Digital Twinning

One of the main goals of Process 2 was to prove that the CLOI-Instance method requires competitively less manual segmentation time compared to the current practice. We validated this hypothesis for the overall framework given that the class segmentation labels are predicted from the CLOI-NET-Class method (Process 1). We use the percentage of *CLOI* classes that the CLOI-Instance method correctly predicts as a proxy to approximate the number of manual labour hours that are still needed in order to achieve an accurate gDT generation. The results are summarized for each *CLOI* dataset in Table 5. A comparison of the manual instance segmentation time for the *CLOI* benchmark dataset generation and the *CLOI* overall framework segmentation time is presented in Figure 13. The total number of man hours needed when deploying the overall framework is calculated as follows. The number of manually segmented *CLOI* classes is computed as the product of the number of shapes that are missed by the framework ($1-recall$) and the average time it takes a modeller to manually segment a given shape. An assumption for the simplification of the calculation here that each *CLOI* class takes the same time regardless of its complexity. The results illustrate that 35% of the manual labour hours are saved on average. The oil refinery dataset is one of the most complex *CLOI* datasets and this is reflected in reduced savings in labour hours for instance segmentation. It is noteworthy that for all *CLOI* facilities, the cylinder *CLOI* class has relatively low recall ($\approx 40\%$) which is attributed to the large number of conduit that are clustered together in one instance.

We evaluated in (Agapaki et al. 2018) the state-of-the-art commercial software that semi-automatically segments cylinders from TLS industrial datasets, however a direct comparison cannot be made since the total number of cylinders considered in that evaluation does not match the number of cylinders in the *CLOI* dataset. However, the number of cylinders correctly detected by EdgeWise can be compared with the number of cylinders segmented by the proposed framework. The results in Table 6 demonstrate that the proposed framework correctly segments more cylinders than those detected by EdgeWise. The proposed framework is designed to better segment conduits and even with the discussed limitations, Table 6 illustrates its superiority to EdgeWise which is mostly in the correctly predicted conduits that EdgeWise does not identify.

The performance of the proposed framework is then compared directly with EdgeWise assuming that the modeling of *CLOI* classes will be manually performed in EdgeWise. Therefore, the average modeling labour time per object is taken from (Agapaki et al. 2018) and multiplied with the number of objects that are not automatically segmented. The output in labour hours in shown in Figure 14 and compared with the manual labour hours for the objects that EdgeWise cannot automatically detect (a fraction of cylinders and the rest of *CLOI* classes). Figure 14 shows that 21% and 39% more time savings are achieved when the proposed framework is utilized for the warehouse and petrochemical plant respectively.

The warehouse and the petrochemical plant datasets are then used as a proxy to estimate the average percentage of labour hour reduction of the CLOI framework compared to EdgeWise per *CLOI* class. The average percentage per class is shown in Table 7. An assumption was made that the modeling time of all cylindrical shapes is the same, since our framework detects cylinders and not their sub-classes, i.e. pipes. Then, the *CLOI* framework is directly compared with EdgeWise for the petrochemical plant with 240,687 objects that was used for manual modeling in (Agapaki et al. 2018). The same assumptions are used here for consistency of the results. The results in Figure 15 reveal that 12 person-months are needed when using the *CLOI* framework instead of the 17 person months that are needed when using EdgeWise. In particular, *CLOI* saves 10% more man-hours for cylinder modeling, which is translated in 773 labour hours saved. Although there is still time required for manual cylinder extraction, the proposed framework clearly outperforms the commercial software EdgeWise.

CONCLUSIONS

This paper presents *CLOI*, an automated benchmarking framework for generating gDTs of existing industrial facilities from point cloud data. This work focuses on the generation of instance point clusters in a cost-effective approach compared to the current practice. The framework consists of two main processes: the *CLOI-NET-Class* segmentation (Process 1), which generates the ten most important industrial objects in the format of class point clusters and *CLOI-Ins* segmentation (Process 2), which segments the class point clusters into individual point clusters. The *CLOI* framework was experimentally validated on the largest published industrial point cloud dataset, which consists of four TLS industrial point clouds. The consistent results on the *CLOI* dataset demonstrate that the proposed framework can reduce the onerous, repetitive manual work of segmenting industrial shapes and therefore reduce the modelling time of the resulting models. The proposed framework provides the foundation for other researchers to cost-effectively segment industrial factories by realizing estimated time-savings of 30% on average and can be used to generate efficiently a gDT of the facility. In the following paragraphs, we present the contributions (**Con**) and limitations (**Lim**) of the *CLOI* framework in detail.

Con 1 This is the first framework of its kind to achieve significantly high and reliable performance (50% mPrec and 35.3% mRec) compared to current state-of-the-art research and commercially available software. It is the first framework to provide significant improvements on cylinder segmentation (53.6% mPrec and 44% mRec) and the first to segment the rest of the *CLOI* classes. It, therefore, provides a solid foundation for future work in generating DTs of industrial facilities.

Con 2 This research moves forward the state of automated class and instance segmentation from TLS point cloud datasets as well as promotes the value of adding "intelligence" to the PCD data. The interpretation of the results strongly suggest that the performance of both the CLOI-NET-Class and the CLOI-Instance methods are significantly improved by using the optimal amount of data during training (≈ 30%) and contextual enforcement rules to accurately segment the *CLOI* classes. **Con 3** It is the first framework of its kind to significantly reduce the manual labour hours (by at least 33%) compared to the state of practice, EdgeWise. It also has 21% and 39% more time savings when segmenting the warehouse and the petrochemical facility dataset compared to EdgeWise. **Con 4** The connectivity of pipe components or members of steel frames assist the modeller in identifying all the connected components of a pipe spool or steel frame when using the outputs of this framework. Figure 16 shows characteristic examples from the warehouse and the oil refinery datasets. The confidence level of the predicted class labels from the CLOI-NET-Class method is also an indicator of whether the performance of the instance segmentation under-segments instances. Figure 16(aiii) shows that the elbows of the pipe spool were predicted with uncertainty (confidence level score ≤ 80%) and this performance led to the under-segmentation of the pipe spool into cylinder and elbow instances. In this case, under-segmentation can be helpful for the modellers since segmentation of the pipe spool into its parts will be an easier task to achieve.

Lim 1 The *CLOI* dataset, although the largest available dataset of TLS industrial point clouds, is not enough to fully validate the proposed framework. More industrial facility point clouds with various configurations are needed to enhance the statistical validity of the framework with an increased confidence level and decrease the bias between facilities especially for the *CLOI* classes that are underrepresented in the dataset. As demonstrated in (Agapaki and Brilakis 2020a) more data is not always beneficial, so careful experimental set-up should be conducted to alleviate from negatively impacting the performance. **Lim 2** Manual annotation of TLS industrial point clouds according to the data preparation explained in the experiments section is an onerous task. In these efforts, an automated segmentation interface should be adopted to enable for easy generation of labelled class and instance point clusters. **Lim 3** Finally, the framework is not designed to segment objects of the same geometric group, for instance pipes, conduits and circular hollow sections or further object types within the same class i.e. globe valves, gate valves. This could be an interesting direction for future research.

There are several gaps in knowledge around industrial gDT in research that follows based on the findings of this work and would benefit from further research, to extend and further enhance the developed framework. Direct future research includes: (a) improved point cloud parsing that is used as input to the CLOI framework and (b) enhanced instance segmentation methods. Future research can focus on improved active learning selection methods based on influence, diversity and uncertainty similar to approaches for active image segmentation (Jain and Grauman 2016). For the CLOI-ins instance segmentation part, a graph-cut based method could be used to improve the instance results instead of the BFS method. The stability of segmentation of instances can be further investigated in future research especially in noisy TLS industrial datasets.

Use classification of cylinders is another interesting research direction to pursue, since in this line of research we assumed that all cylindrical shapes belong to the same "cylinder" class. However, these shapes can be either pipes, conduits, circular hollow sections (i.e. parts of staircases or columns) or even vessels. Sub-classification of cylinders could be achieved by adding further contextual relationships in the CLOI-NET class segmentation method.

Another research direction is fluid recognition in the pipelines that run in industrial facilities and especially in an oil and gas refinery, which is critical for the production of the unit. Photogrammetry data sources and/or P&IDs can be correlated to the laser scanned data and inference suggestions on the material type can be made. For example, features used for material recognition can be either the colour, texture, micro-texture, outline shape (i.e. curvature) or reflectance-based features and CNN networks have recently been used for material recognition from images (Schwartz and Nishino 2019). Longer term directions of this research include optimized generative design of structural shapes based on their automatically generated gDTs.

DATA AVAILABILITY

Some or all data, models, or code used during the study were provided by a third party. Direct requests for these materials may be made to the provider as indicated in the Acknowledgements.

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 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

Agapaki, E. and Brilakis, I. (2020a). "Cloi-net: Class segmentation of industrial facilities' point cloud datasets." *Advanced Engineering Informatics*, 45, 101121.

Agapaki, E. and Brilakis, I. (2020b). "Instance segmentation of industrial point cloud data.

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

Agapaki, E., Miatt, G., and Brilakis, I. (2018). "Prioritizing object types for modelling existing industrial facilities." *Automation in Construction*.

Agapaki, E. and Nahangi, M. (2020). "Chapter 3 - Scene understanding and model generation." *Infrastructure Computer Vision*, I. Brilakis and C. Haas, eds., Elsevier, 1 edition, Chapter 3.

Agarwal, R., Chandrasekaran, S., and Sridhar, M. (2016). "The digital future of construction, <https://www.globalinfrastructureinitiative.com/sites/default/files/pdf/The-digital-future-of-construction-Oct-2016.pdf>.

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

Barbosa, F., Woetzel, J., Mischke, J., Joao Ribeirinho, M., Sridhar, M., Parsons, M., and Brown, S. (2017). "Reinventing Construction: A Route to Higher Productivity, <https://www.mckinsey.com/ /media/McKinsey/Industries/Capital Projects and Infrastructure/Our Insights/Reinventing construction through a productivity revolution/MGI-Reinventing-construction-A-route-to-higher-productivity-Full-report.ashx>.

Bassier, M., Van Genechten, B., and Vergauwen, M. (2019). "Classification of sensor independent point cloud data of building objects using random forests." *Journal of Building Engineering*, 21, 468–477.

Bauer, F. L. and Wössner, H. (1972). "The "Plankalkül" of Konrad Zuse: A Forerunner of Today's Programming Languages." *Communications of the ACM*.

Borenstein, E. and Ullman, S. (2008). "Combined top-down/bottom-up segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12), 2109–2125.

Borrmann, A. and Berkhahn, V. (2018). "Principles of geometric modeling." *Building Information Modeling*, Springer, 27–41.

Chen, J., Kira, Z., and Cho, Y. K. (2019). "Deep learning approach to point cloud scene understanding for automated scan to 3d reconstruction." *Journal of Computing in Civil Engineering*, 33(4), 04019027.

Chen, J., Kira, Z., and Cho, Y. K. (2021). "Lrgnet: Learnable region growing for class-agnostic point cloud segmentation." *IEEE Robotics and Automation Letters*, 6(2), 2799–2806.

ClearEdge (2019). "Plant Modeling Capabilities, <https://www.clearedge3d.com/products/edgewise-plant/>.

Coleman, C., Chandramouli, M., Damodaran, S., and Deuel, E. (2017). "Making maintenance smarter, <https://www2.deloitte.com/us/en/insights/focus/industry-4-0/using-predictive-technologies-for-asset-maintenance.html>.

Devaux, A., Br??dif, M., and Paparoditis, N. (2012). "A web-based 3D mapping application using WebGL allowing interaction with images, point clouds and models." *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*.

Dilda, V., Mori, L., Noterdaeme, O., and Niel, J. V. (2018). "Using advanced analytics to boost productivity and profitability in chemical manufacturing, <https://www.mckinsey.com/industries/chemicals/our-insights/using-advanced-analytics-to-boost-productivity-and-profitability-in-chemical-manufacturing>.

Dimitrov, A. and Golparvar-Fard, M. (2015). "Segmentation of building point cloud models including detailed architectural/structural features and MEP systems." *Automation in Construction*, 51(C), 32–45.

Fumarola, M. and Poelman, R. (2011). "Generating virtual environments of real world facilities: Discussing four different approaches." *Automation in Construction*, Vol. 20, 263–269.

Gerbert, P., Castagnino, S., Rothballer, C., Renz, A., and Filitz, R. (2016). "Digital in Engineering and Construction, <http://futureofconstruction.org/content/uploads/2016/09/BCG-Digital-in-Engineering-and-Construction-Mar-2016.pdf>.

Glaessgen, E. H. and Stargel, D. S. (2012). "The digital twin paradigm for future NASA and U.S. Air force vehicles." *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*.

Grieves, M. (2014). "Digital Twin: Manufacturing Excellence Through Virtual Factory Replication." *Nc-Race 18*.

Huang, J. and You, S. (2013). "Detecting objects in scene point cloud: A combinational approach." *Proceedings - 2013 International Conference on 3D Vision, 3DV 2013*, 175–182.

Hullo, J.-F., Thibault, G., Boucheny, C., Dory, F., and Mas, A. (2015). "Multi-Sensor As-Built Models of Complex Industrial Architectures." *Remote Sensing*, 7(12), 16339–16362.

Jain, S. D. and Grauman, K. (2016). "Active Image Segmentation Propagation." *Proceedings of* *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Jin, Y. H. and Lee, W. H. (2019). "Fast cylinder shape matching using random sample consensus in large scale point cloud." *Applied Sciences (Switzerland)*.

Kalogerakis, E., Averkiou, M., Maji, S., and Chaudhuri, S. (2017). "3D Shape segmentation with projective convolutional networks." *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*.

Kawashima, K., Kanai, S., and Date, H. (2014). "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.

Klokov, R. and Lempitsky, V. (2017). "Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models." *Proceedings of the IEEE International Conference on Computer Vision*.

Komori, J. and Hotta, K. (2019). "Ab-pointnet for 3d point cloud recognition." *2019 Digital Image Computing: Techniques and Applications (DICTA)*, IEEE, 1–6.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." *Advances In Neural Information Processing Systems*.

Landrieu, L. and Simonovsky, M. (2018). "Large-scale point cloud semantic segmentation with superpoint graphs.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (2008). "Backpropagation Applied to Handwritten Zip Code Recognition." *Neural Computation*.

Li, B., Shi, Y., Qi, Z., and Chen, Z. (2019a). "A survey on semantic segmentation." *IEEE International Conference on Data Mining Workshops, ICDMW*.

Li, Y., Ma, L., Zhong, Z., Cao, D., and Li, J. (2019b). "Tgnet: Geometric graph cnn on 3-d point cloud segmentation." *IEEE Transactions on Geoscience and Remote Sensing*, 58(5), 3588–3600.

Li, Z., Zhang, L., Tong, X., Du, B., Wang, Y., Zhang, L., Zhang, Z., Liu, H., Mei, J., Xing, X., and Mathiopoulos, P. T. (2016). "A three-step approach for tls point cloud classification." *IEEE Transactions on Geoscience and Remote Sensing*, 54(9), 5412–5424.

Liang, M., Yang, B., Wang, S., and Urtasun, R. (2018). “Deep Continuous Fusion for Multi-sensor 3D Object Detection.” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Liang, Z., Yang, M., Deng, L., Wang, C., and Wang, B. (2019). “Hierarchical depthwise graph convolutional neural network for 3d semantic segmentation of point clouds.” *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 8152–8158.

Liu, T., Cai, Y., Zheng, J., and Thalmann, N. M. (2021). “Beacon: a boundary embedded attentional convolution network for point cloud instance segmentation.” *The Visual Computer*, 1–11.

Liu, Y. J., Zhang, J. B., Hou, J. C., Ren, J. C., and Tang, W. Q. (2013). “Cylinder detection in large-scale point cloud of pipeline plant.” *IEEE Transactions on Visualization and Computer Graphics*, 19(10), 1700–1707.

Lu, Q., Chen, C., Xie, W., and Luo, Y. (2020). “Pointngcnn: Deep convolutional networks on 3d point clouds with neighborhood graph filters.” *Computers & Graphics*, 86, 42–51.

Ma, J. W., Czerniawski, T., and Leite, F. (2020). “Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic bim-based point clouds.” *Automation in Construction*, 113, 103144.

Marshall, G. F. (2016). *Handbook of Optical and Laser Scanning*.

Marton, Z. C., Rusu, R. B., and Beetz, M. (2009). “On fast surface reconstruction methods for large and noisy point clouds.” *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference on*, 3218–3223.

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

McKinsey Global Institute (2015). “Digital America: A tale of the Haves and Have-Mores.” *Report no.*, <https://www.mckinsey.com/~/media/McKinsey/Industries/High Tech/Our Insights/Digital America A tale of the haves and have mores/MGI Digital America_Executive Summary_December 2015.ashx>.

Meagher, D. (1980). *Octree encoding: a new technique for the representation, manipulation and* *display of arbitrary 3-D objects by computer*. Rensselaer Polytechnic Institute.

National Institute of Standards and Technology (2018). “The Costs and Benefits of Advanced Maintenance in Manufacturing.” *Report no.*, U.S. Department of Commerce, <https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.100-18.pdf>.

Pang, Y., Li, L., Hu, W., Peng, Y., Liu, L., and Shao, Y. (2012). “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*.

Patil, A. K., Holi, P., Lee, S. K., and Chai, Y. H. (2017). “An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds.” *Automation in Construction*, 75, 65–78.

PECI (1999). “Portable Data Loggers Diagnostic Tools for Energy-Efficient Building Operations.” *Report no.*, Prepared for the U.S. Environmental Protection Agency and U.S. Department of Energy by Portland Energy Conservation, Incorporated, Portland, Oregon, <https://www.pnnl.gov/main/publications/external/technical_reports/PNNL19634.pdf>.

Perez-Perez, Y., Golparvar-Fard, M., and El-Rayes, K. (2016). “Semantic and Geometric Labeling for Enhanced 3D Point Cloud Segmentation.” *Construction Research Congress 2016*, 2542–2552.

Perez-Perez, Y., Golparvar-Fard, M., and El-Rayes, K. (2021). “Segmentation of point clouds via joint semantic and geometric features for 3d modeling of the built environment.” *Automation in Construction*, 125, 103584.

Peyghambarzadeh, S. M., Azizmalayeri, F., Khotanlou, H., and Salarpour, A. (2020). “Point-planenet: Plane kernel based convolutional neural network for point clouds analysis.” *Digital Signal Processing*, 98, 102633.

Pham, Q., Nguyen, D. T., Hua, B., Roig, G., and Yeung, S. (2019). “JSIS3D: Joint Semantic-Instance Segmentation of 3D Point Clouds with Multi-Task Pointwise Networks and Multi-Value Conditional Random Fields.” *CVPR*.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017a). “PointNet++: Deep Hierarchical Feature

Learning on Point Sets in a Metric Space." *Computer Vision and Pattern Recognition (CVPR)*.

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

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

Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M. (2009). "Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments." *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, 1–6.

Sampath, A. and Shan, J. (2010). "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds." *IEEE Transactions on Geoscience and Remote Sensing*, 48(3 PART2), 1554–1567.

Schnabel, R., Wahl, R., and Klein, R. (2007). "Efficient RANSAC for Point-Cloud Shape Detection." *Computer Graphics Forum*, 26(2), 214–226.

Schuetz, M. (2016). "Potree: Rendering Large Point Cloud in Web Browsers." Ph.D. thesis, University of TU Wien, , <https://pdfs.semanticscholar.org/0d9d/db7335331d28a4a23e086e960396fd4e1b65.pdf>.

Schwartz, G. and Nishino, K. (2019). "Recognizing Material Properties from Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shao, T., Yang, Y., Weng, Y., Hou, Q., and Zhou, K. (2018). "H-CNN: Spatial Hashing Based CNN for 3D Shape Analysis." *IEEE Transactions on Visualization and Computer Graphics*.

Son, H. and Kim, C. (2016). "Automatic segmentation and 3D modeling of pipelines into constituent parts from laser-scan data of the built environment." *Automation in Construction*.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). "Multi-view convolutional neural networks for 3D shape recognition." *Proceedings of the IEEE International Conference on Computer Vision*.

Taha, A. A. and Hanbury, A. (2015). "Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool." *BMC Medical Imaging*.

Tatarchenko, M., Dosovitskiy, A., and Brox, T. (2017). “Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs.” *Proceedings of the IEEE International Conference on Computer Vision*.

Teichmann, M., Weber, M., Zöllner, M., Cipolla, R., and Urtasun, R. (2018). “MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving.” *IEEE Intelligent Vehicles Symposium, Proceedings*.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L. J. (2019). “Kpconv: Flexible and deformable convolution for point clouds.

Vosselman (2009). “Advanced Point Cloud Processing.” *In Photogrammetric Week ’09*, 137–146.

Wang, L., Huang, Y., Hou, Y., Zhang, S., and Shan, J. (2019a). “Graph attention convolution for point cloud semantic segmentation.” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10296–10305.

Wang, S., Suo, S., Ma, W. C., Pokrovsky, A., and Urtasun, R. (2018a). “Deep Parametric Continuous Convolutional Neural Networks.” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

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

Wang, X., Shen, X., Shen, C., and Jia, J. (2019b). “Associatively Segmenting Instances and Semantics in Point Clouds.” *CVPR*.

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

West, T. and Blackburn, M. (2017). “Is Digital Thread/Digital Twin Affordable? A Systemic Assessmet of the Cost of DoD’s Latest Manhattan Project.” *Procedia Computer Science*, 114, 47–56.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). “3D ShapeNets: A deep representation for volumetric shapes.” *Proceedings of the IEEE Computer Society Confer-*

ence on Computer Vision and Pattern Recognition, Vol. 07-12-June, 1912–1920.

Xie, Y., Tian, J., and Zhu, X. X. (2019). “A review of point cloud semantic segmentation.” *arXiv preprint arXiv:1908.08854*.

Xiong, X., Adan, A., Akinci, B., and Huber, D. (2013). “Automatic creation of semantically rich 3D building models from laser scanner data.” *Automation in Construction*, 31, 325–337.

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

Zhang, J., Lin, X., and Ning, X. (2013). “Svm-based classification of segmented airborne lidar point clouds in urban areas.” *Remote Sensing*, 5(8), 3749–3775.

Zhou, Y. and Tuzel, O. (2017). “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection.” *arXiv*.

List of Tables

TABLE 1. CLOI framework performance for the oil refinery dataset

Method	mPrec (%)	mRec (%)
SGPN (Wang et al. 2018b)	5.3	6.5
ASIS (Wang et al. 2019b)	16.7	4.5
CLOI-Framework (without boundary)	20.6	19.9
CLOI-Framework	31.1	21.0

TABLE 2. Performance of instance segmentation networks per *CLOI* shape in the oil refinery dataset

Prec (%)	**Angles**	**Channels**	**Cylinders**	**Elbows**	**I-beams**	**Valves**	**Flanges**
ASIS	0	0	27.2	25	41.5	6.3	0
SGPN	3.8	4.2	3.5	7.6	8.6	5.3	14
BFS	15.3	5.3	33.7	36.6	30	10.2	13.5
CLOI-Instan e	29.7	17.1	28.2	54.3	45.6	15.1	28
Rec (%)	**Angles**	**Channels**	**Cylinders**	**Elbows**	**I-beams**	**Valves**	**Flanges**
ASIS	0	0	4.6	0.1	25.5	1.5	0
SGPN	2.8	3.5	4.2	3.6	5.9	15.2	4.6
BFS	18.1	8.8	23.2	15	39.3	25.7	9.3
CLOI-Instan e	17.7	11.7	28.8	15.7	39.0	25.3	8.8

TABLE 3. Optimal class segmentation pre-annotation percentage of test facility data for active learning

Test facility	Optimal pre-annotated data (%)
Warehouse	30
Processing unit	30
Oil refinery	25
Petrochemical	20

TABLE 4. Performance of the CLOI-Instance method per *CLOI* shape for all the *CLOI* datasets (IoU=25%)

Oil refinery	**Angles**	**Channels**	**Cylinders**	**Elbows**	**I-beams**	**Valves**	**Flanges**
Prec (%)	43.9	27.1	49.6	70.2	57.4	21.3	34.7
Rec (%)	26.1	18.6	43.1	20.4	49.1	35.9	10.8
Warehouse	**Angles**	**Channels**	**Cylinders**	**Elbows**	**I-beams**	**Valves**	**Flanges**
Prec (%)	56	67.1	64.7	76.9	44.4	29.4	30.8
Rec (%)	16.5	34.6	49.1	18.6	100	41.7	28.6
Petrochemical	**Angles**	**Channels**	**Cylinders**	**Elbows**	**I-beams**	**Valves**	**Flanges**
Prec (%)	50	52.6	51.1	70	77.8	29.7	40
Rec (%)	35	46.2	48.2	20	61.8	91.7	8.3
Processing unit	**Angles**	**Channels**	**Cylinders**	**Elbows**	**I-beams**	**Valves**	**Flanges**
Prec (%)	36.8	39.1	48.8	50	72.3	41.4	14.3
Rec (%)	8.7	23.7	35.5	9.1	46.4	43.5	0.5

TABLE 5. Manual labour hours and total segmentation savings of the overall framework per *CLOI* facility.

Oil refinery	Angles	Channels	Cylinders	Elbows	I-beams	Valves	Flanges	Other
Recall (%)	26	19	43	20	49	36	11	25
Total # of shapes	211	2347	94	121	723	215	202	563
Manually segmented # of shapes	156	1910	54	96	368	138	180	425
Total # of man hours				173				
Total savings (%)				**26**				
Warehouse	Angles	Channels	Cylinders	Elbows	I-beams	Valves	Flanges	Other
Recall (%)	16.5	34.6	56	18.6	100	41.7	28.6	27.9
Total # of shapes	111	168	910	258	12	85	21	195
Manually segmented # of shapes	93	110	400	210	0	50	15	141
Total # of man hours				67				
Total savings (%)				**42**				
Petrochemical	Angles	Channels	Cylinders	Elbows	I-beams	Valves	Flanges	Other
Recall (%)	35	46.2	41.8	20	61.8	91.7	8.3	29
Total # of shapes	60	264	1489	376	140	53	130	828
Manually segmented # of shapes	39	142	866	301	54	4	119	588
Total # of man hours				74				
Total savings (%)				**37**				
Processing unit	Angles	Channels	Cylinders	Elbows	I-beams	Valves	Flanges	Other
Recall (%)	8.7	23.7	35.5	9.1	46.4	43.5	0.4	25.1
Total number of shapes	188	34	1100	382	274	341	229	370
Manually segmented # of shapes	172	26	710	347	147	193	228	277
Total # of man hours				117				
Total savings (%)				**28**				

TABLE 6. Correctly predicted cylinders of the petrochemical plant and warehouse point clouds using EdgeWise and our framework.

# of cylinders correctly predicted	**Warehouse**	**Petrochemical**
EdgeWise	468	164
Proposed framework	510	623

TABLE 7. Percentage (%) of the reduction of the labour hours of the *CLOI* framework compared to EdgeWise per class.

CLOI class	% of labour hour reduction
Cylinders	22.3
Channels	40.4
I-beams	81
Valves	67
Elbows	19.3
Flanges	18.5
Angles	25.7

List of Figures

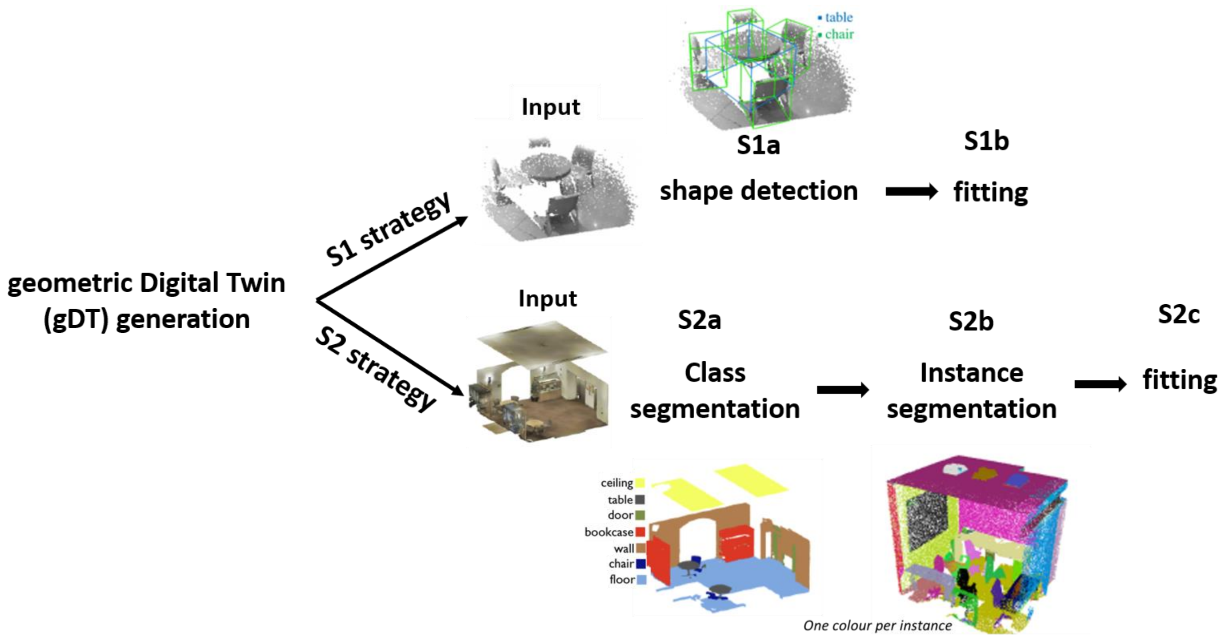


Fig. 1. Automated geometric Digital Twinning strategies.

Methods	Advantages	Limitations
Attribute based	• no need to train • potentially faster	• difficult to capture complicated features • prior knowledge needed • might not be robust to noisy data
Machine learning	• automatically learn features and best combinations of features • data driven • combine with hand-crafted features	• large training dataset might be needed, • might overfit if training data not diverse enough

Fig. 2. Advantages and limitations of segmentation methods discussed.

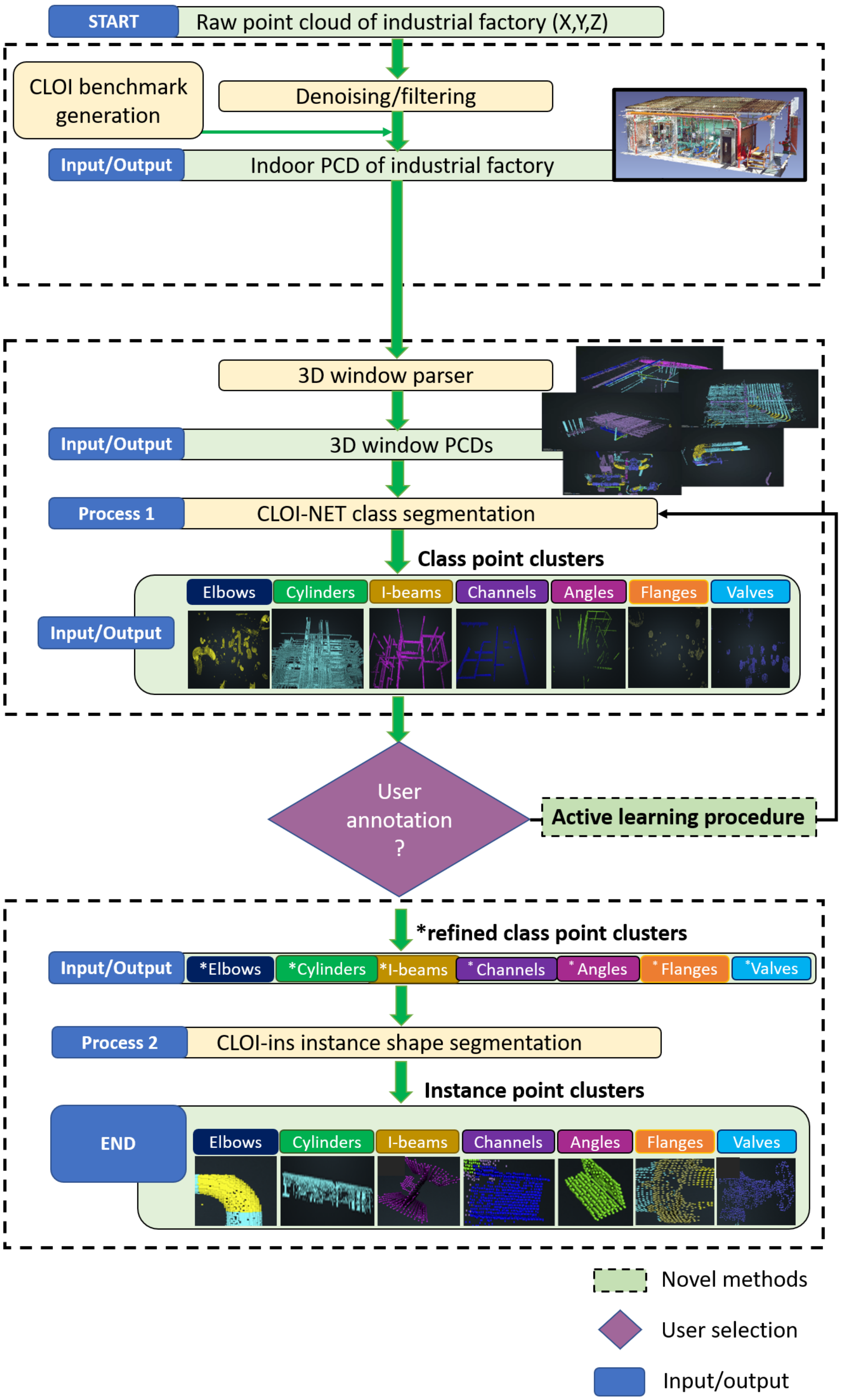


Fig. 3. Workflow of the proposed *CLOI* framework.

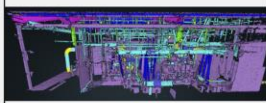
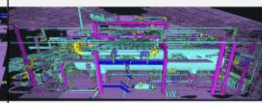
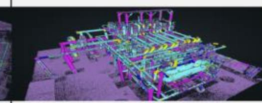
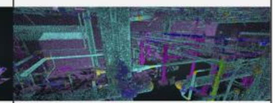
Facility name	Warehouse	Oil refinery	Processing unit	Petrochemical plant
Class annotated Ibeam Angle Elbow Flange Pipe Channel Valve Other				
Original number of points (in millions)	129	6,005	340	675
Number of shapes	1,760	4,476	2,918	3,340
Facility area (m²)	396	300	536	1,379
Labor hours (h)	115.5	233	162.5	117.5

Fig. 4. CLOI benchmark dataset specifications.

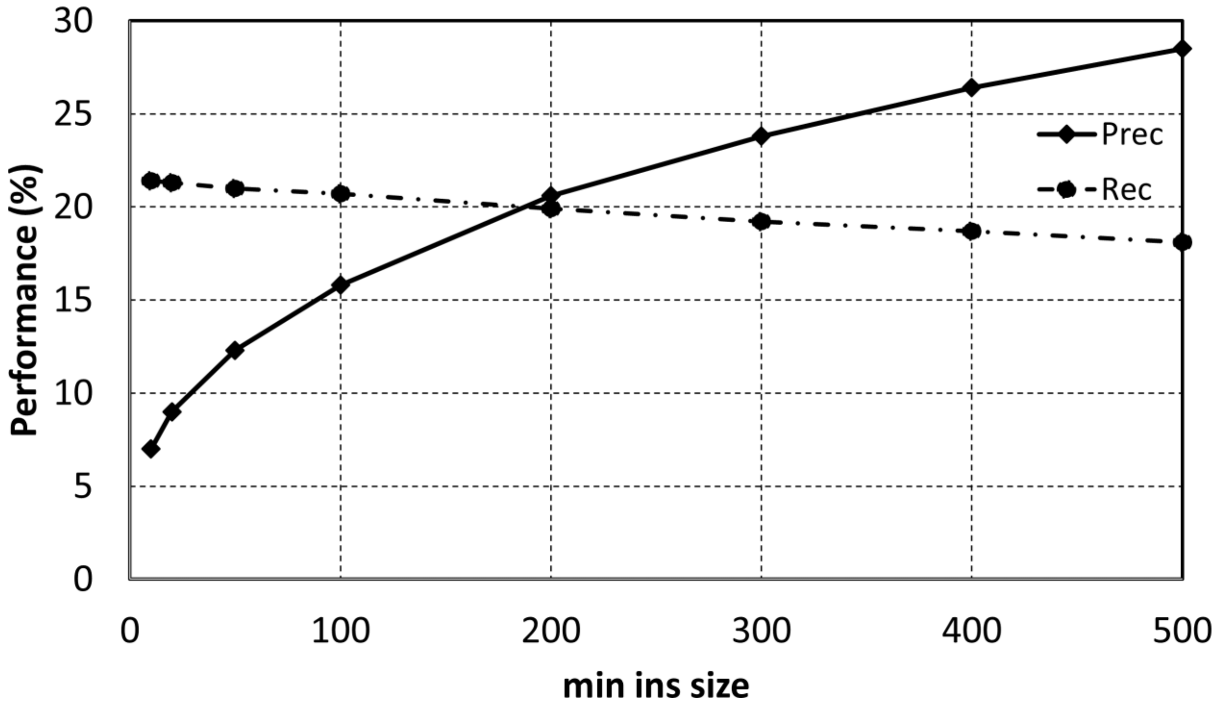


Fig. 5. Performance of the BFS algorithm with respect to the minimum instance size (mu) for IoU=50% and epsilon=4 cm. Test on the oil refinery facility.

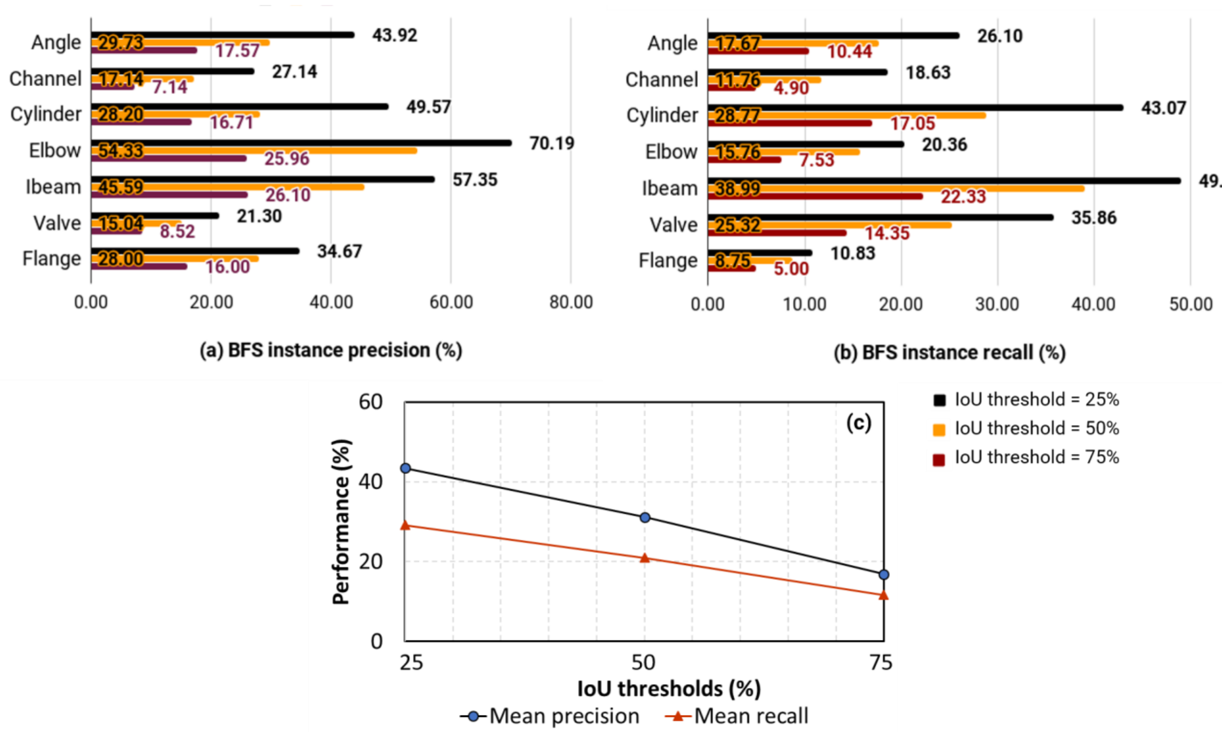


Fig. 6. (a) CLOI framework precision and (b) recall per CLOI class and (c) mean precision and recall for different IoU thresholds for the oil refinery facility.

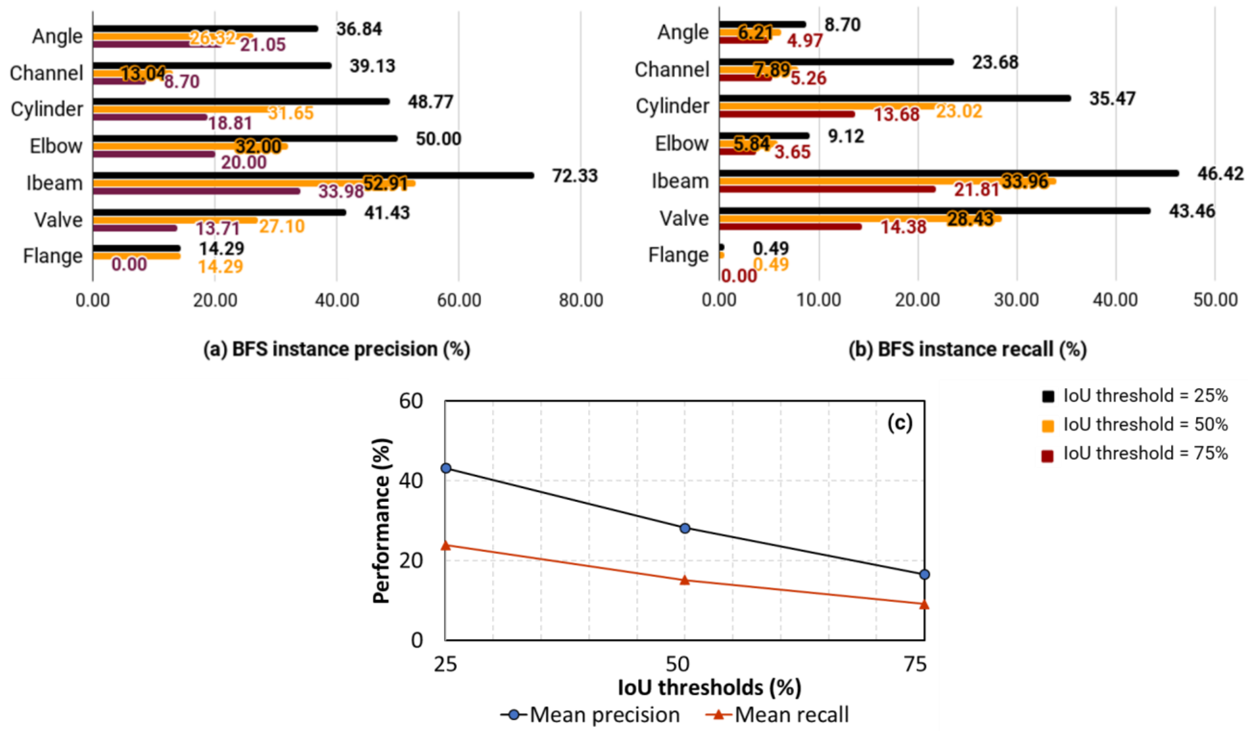


Fig. 7. (a) CLOI framework precision and (b) recall per CLOI class and (c) mean precision and recall for different IoU thresholds for the processing unit facility.

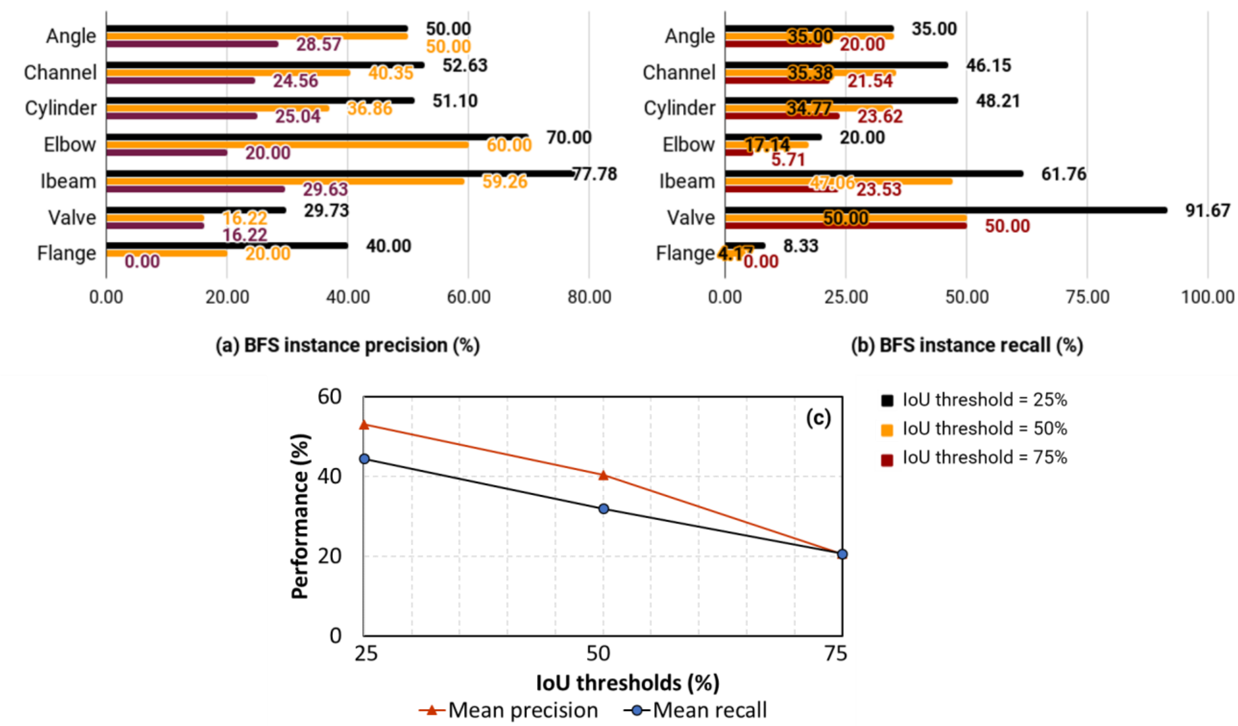


Fig. 8. (a) CLOI framework precision and (b) recall per CLOI class and (c) mean precision and recall for different IoU thresholds for the petrochemical plant facility.

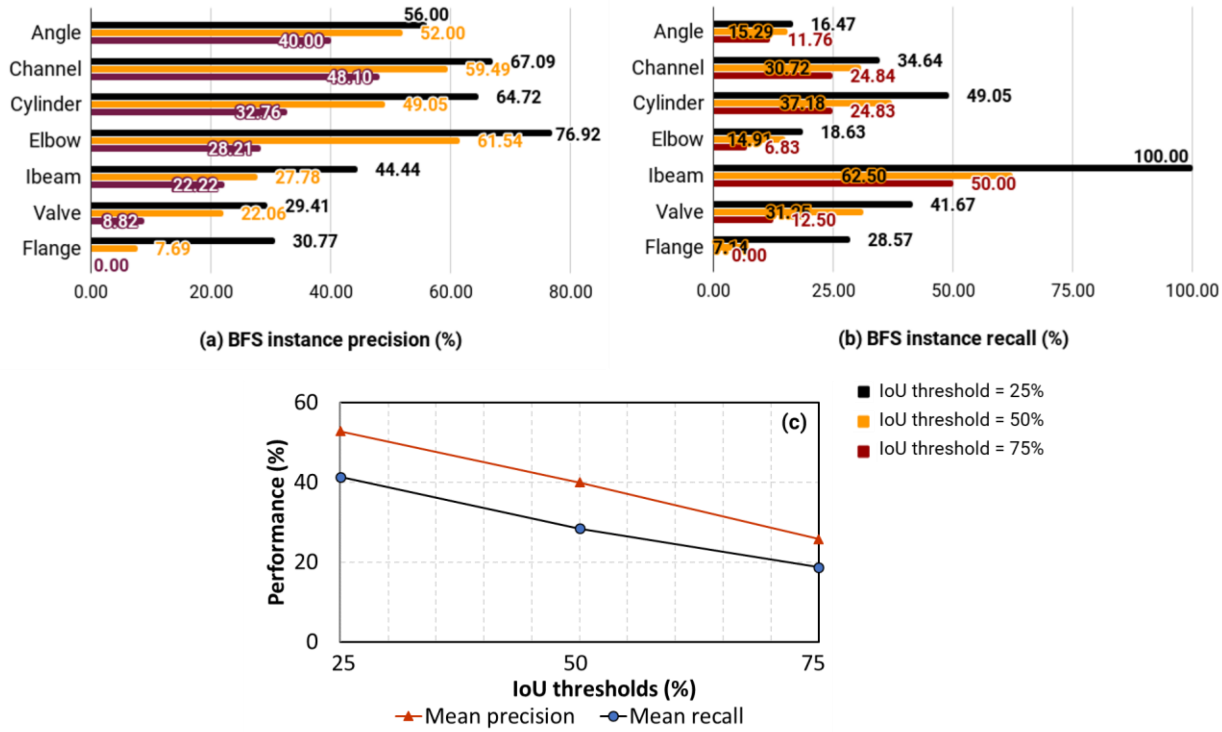


Fig. 9. (a) CLOI framework precision and (b) recall per CLOI class and (c) mean precision and recall for different IoU thresholds for the warehouse facility.

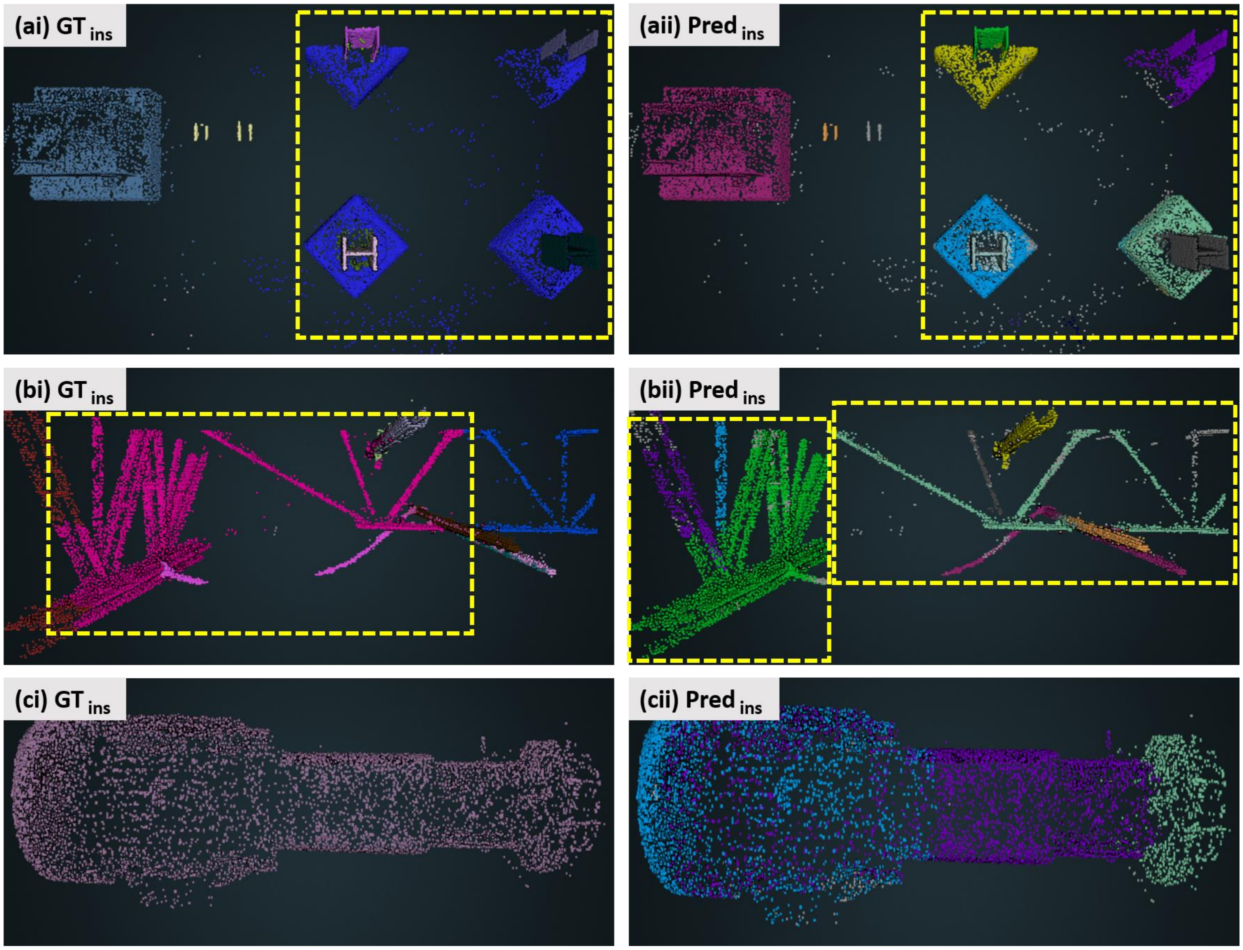


Fig. 10. Examples where the CLOI framework outperforms the manual instance segmentation. (i) refers to ground truth instances and (ii) refers to predicted instances with the CLOI framework.

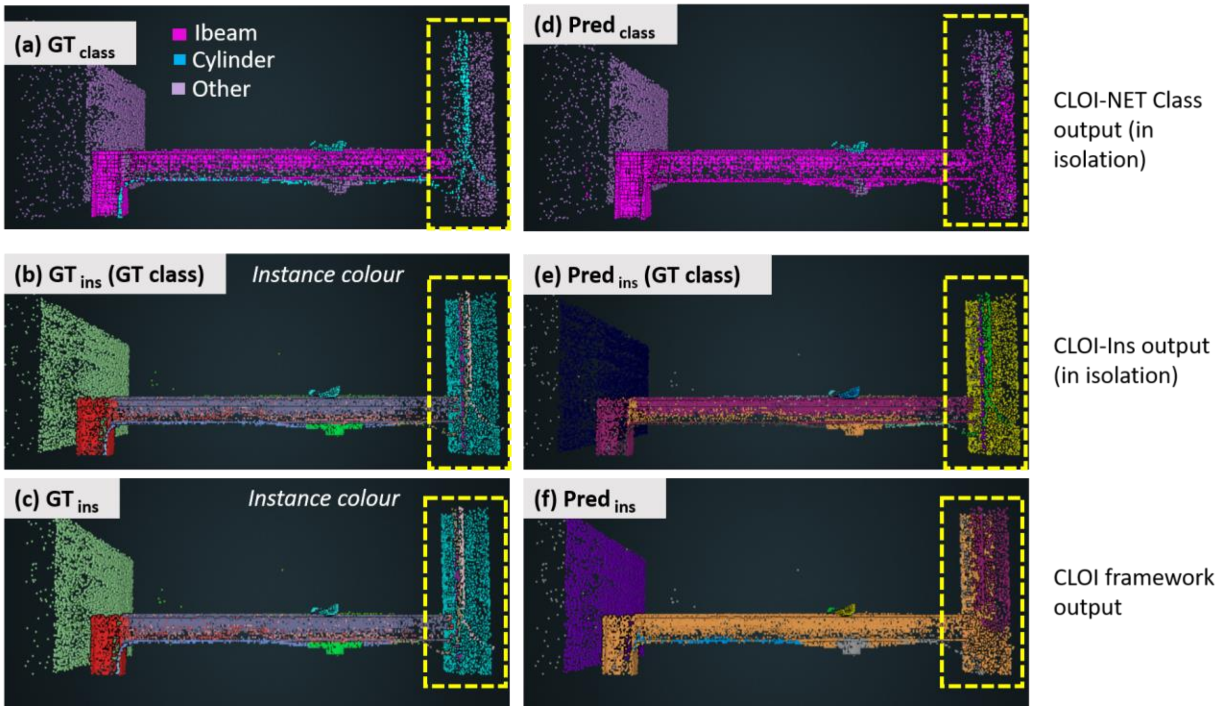


Fig. 11. (a) Class segmented ground truth and predicted point clusters (CLOI-NET outputs), (b) instance segmented ground truth and predicted point clusters (CLOI-Ins outputs) and (c) instance segmented ground truth and predicted point clusters (CLOI framework outputs).

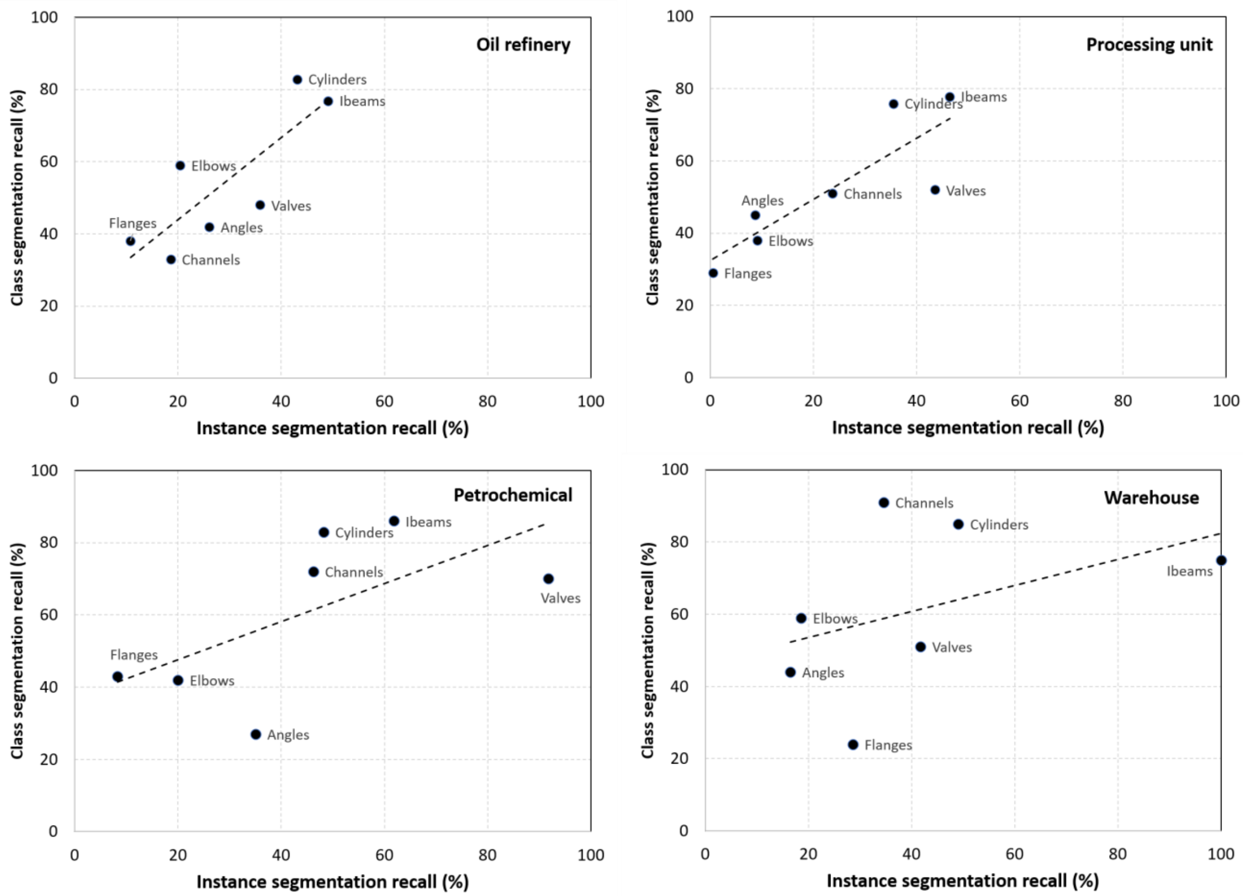


Fig. 12. Class and instance segmentation recall per CLOI class for the oil refinery, processing unit, petrochemical and warehouse datasets.

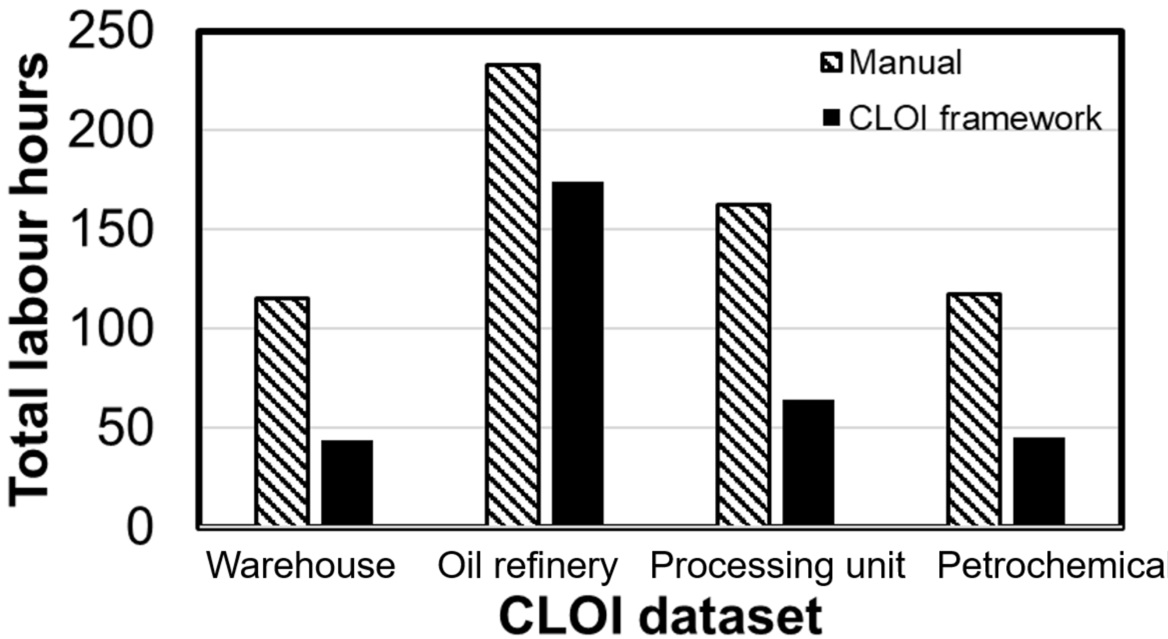


Fig. 13. Manual and our framework's total labor hours per CLOI dataset.

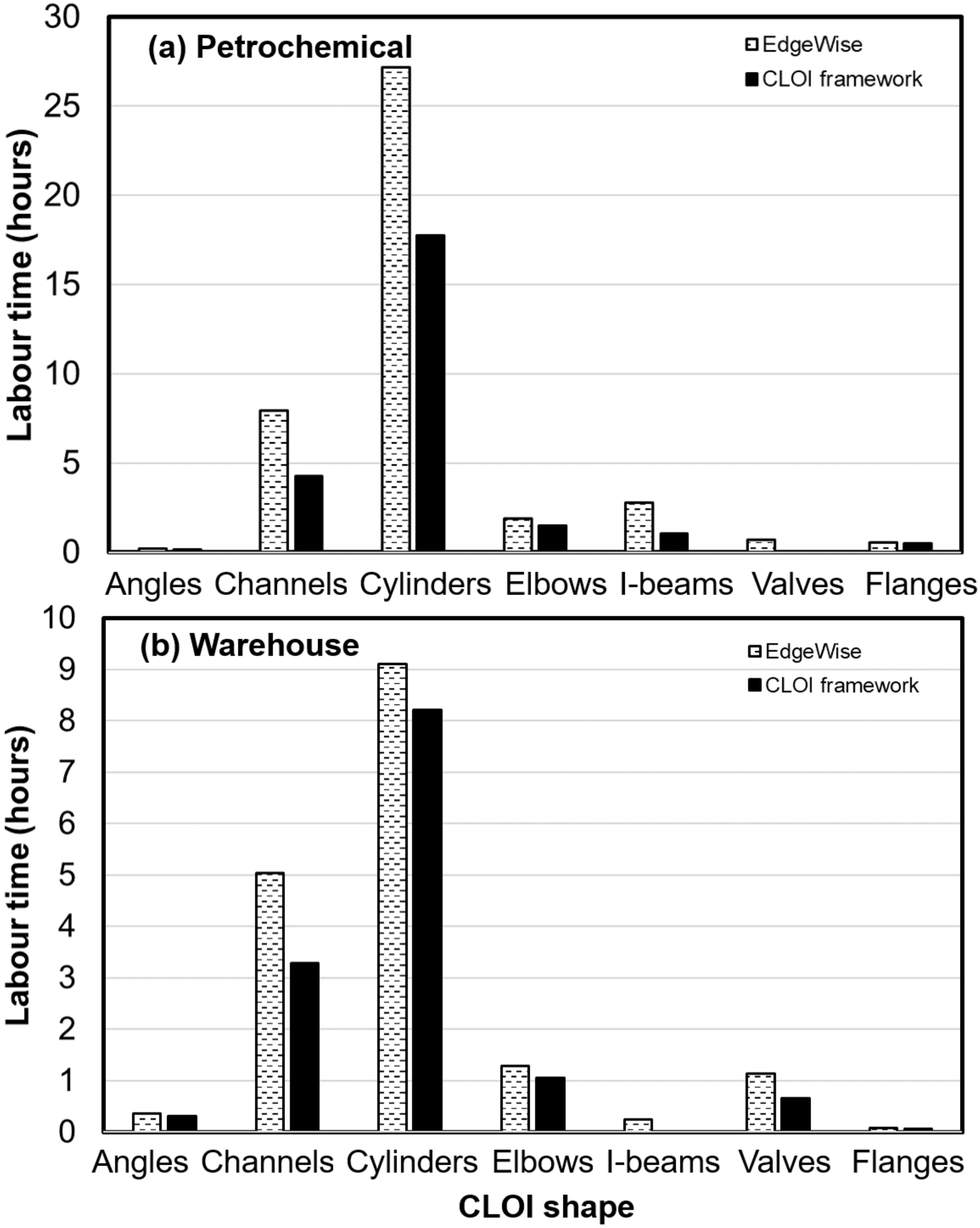


Fig. 14. Comparison of EdgeWise and our framework with respect to manual labour hours per CLOI shape for the (a) petrochemical and (b) the warehouse dataset.

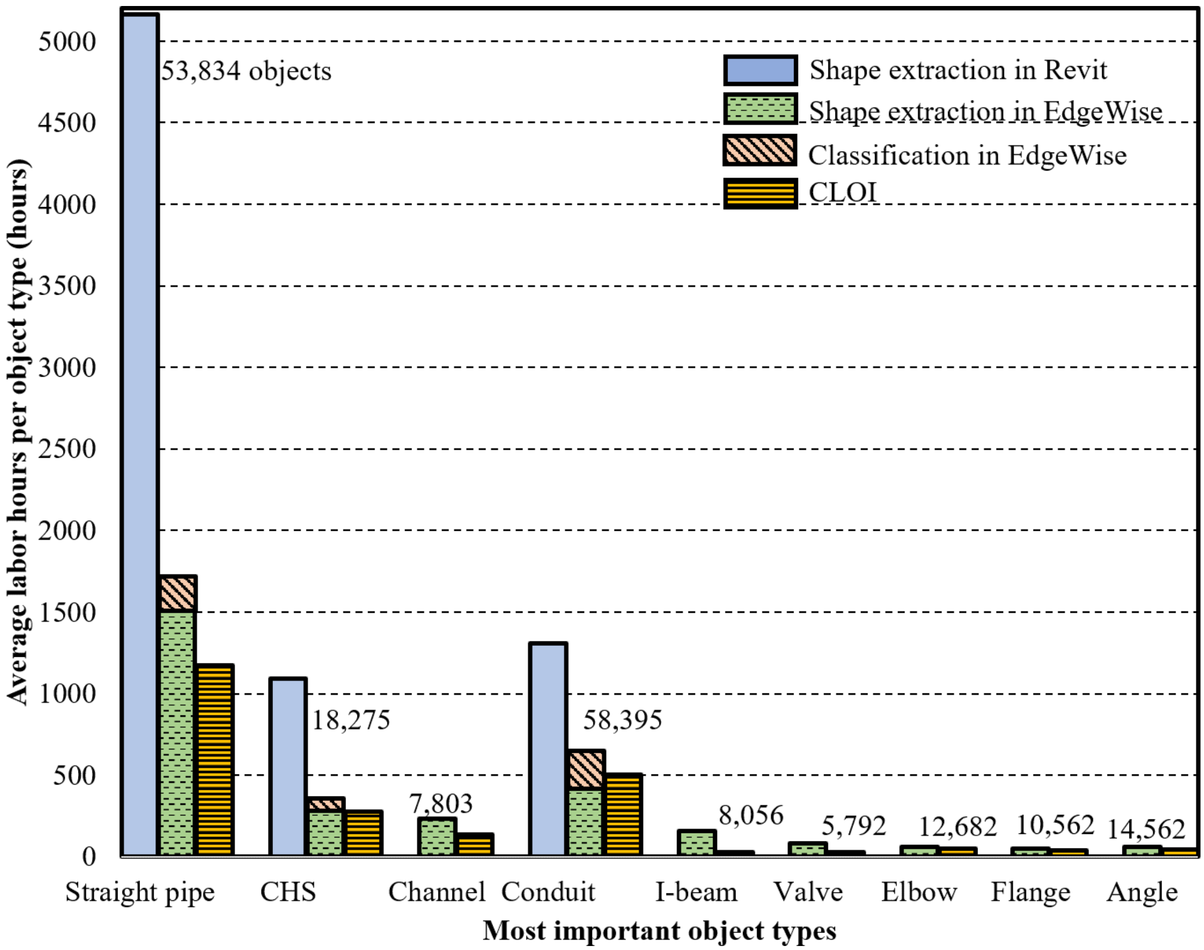


Fig. 15. Average modelling labor hours per object type for the most important objects of a sample facility with shown numbers of objects.

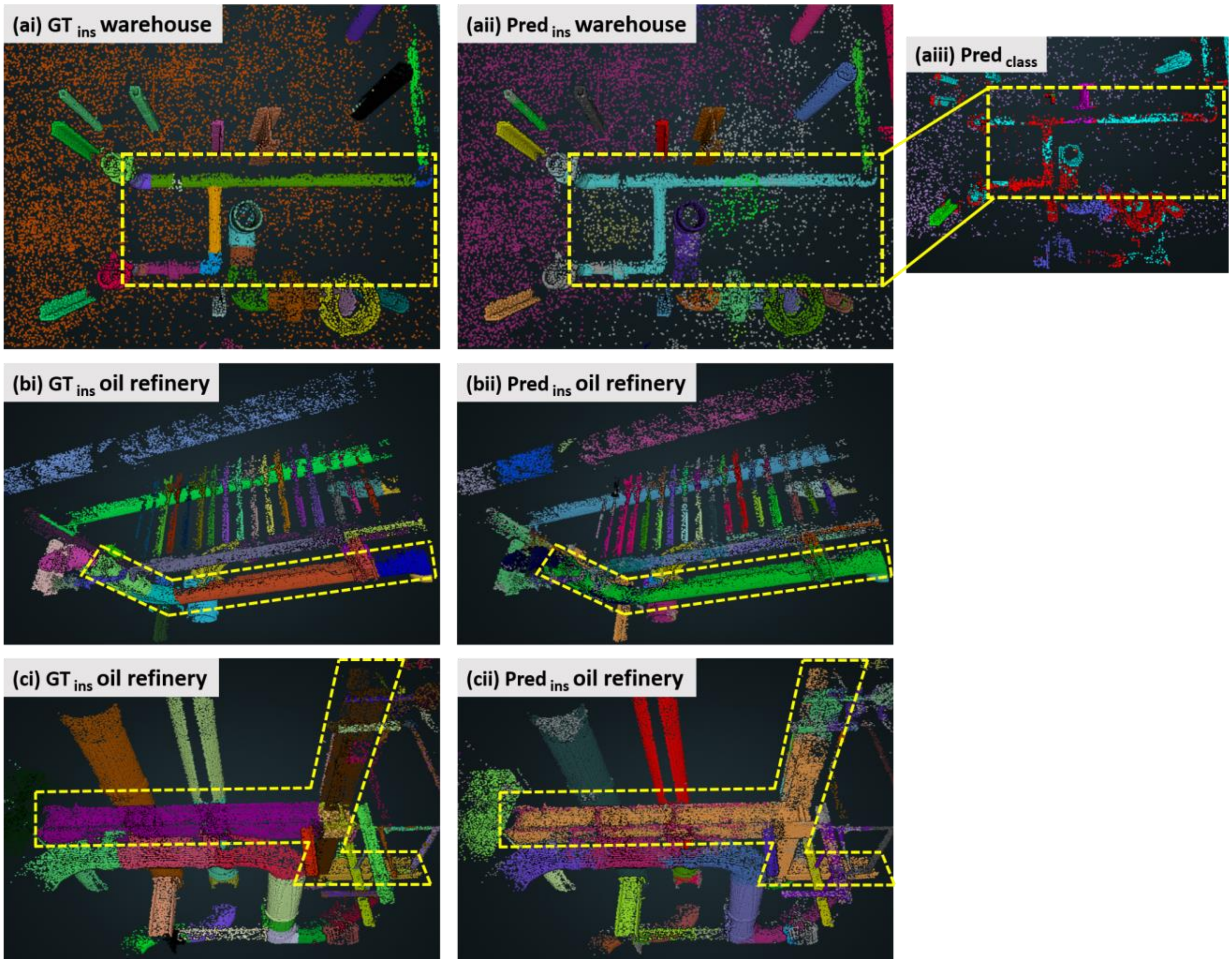


Fig. 16. Examples of ground truth and predicted instances of piping elements (a,b) and (c) I-beams of a steel frame. (aiii) Predicted class label predictions (predictions with <= 80% confidence score coloured in red).