ReVISOR: ResUNets with visibility and intensity for structured outlier removal

Maxime Kirgo, Guillaume Terrasse, Guillaume Thibault, Maks Ovsjanikov

ReVISOR: ResUNets with visibility and intensity for structured outlier removal

Maxime Kirgo [a,b,*], Guillaume Terrasse [b], Guillaume Thibault [b], Maks Ovsjanikov [a]

[a] École polytechnique, IPParis, 1 rue Honoré d'Estienne d'Orves, Palaiseau, 91120, Essonne, France
[b] EDF R&D, 7 boulevard Gaspard Monge, Palaiseau, 91120, Essonne, France

A B S T R A C T

Laser acquisition of large-scale point clouds is prone to several categories of measurement errors, which can lead, in particular, to the presence of undesirable *outlier* points. Existing outlier detection techniques are primarily based on analyzing *local properties* of point distributions to distinguish "clean" from "noisy" data. In contrast, real-world acquisition often has to deal with *reflective surfaces*, which can give rise to structured outliers that can be indistinguishable from clean geometry through purely local analysis. We make several contributions to address the problem of reflection-induced outlier detection. First, to overcome the scarcity of annotated data, we introduce a new dataset tailored for this task. Second, to capture *non-local* dependencies, we study and demonstrate, for the first time, the utility of deep learning based semantic segmentation architectures for reflection-induced outlier detection. By doing so, we bring together the fields of shape denoising/repair and semantic segmentation. Third, we demonstrate that additional non-local cues in the form of laser intensity and a computed visibility signal help boost the performance considerably. We denote our pipeline as ResUNets with Visibility and Intensity for Structured Outlier Removal, or **ReVISOR**, and demonstrate its superior performance against existing baselines on real-world data.

1. Introduction

Laser acquisition of large-scale point clouds is prone to several categories of measurement errors. Detecting real-world outliers, such as the artifacts caused by *reflections*, is a particularly challenging task. Unlike unstructured, e.g., Gaussian noise or uniformly distributed outliers, reflections can lead to wrong acquisitions that closely resemble large parts of actual geometry, located *far away* from the real surfaces. This non-local behavior severely reduces the applicability of local statistical approaches, and more broadly the vast majority of previous work on this topic. Unfortunately, non-local outlier detection has rarely been studied in previous works, first because real-world acquisition data is lacking, and, second, because labeling is very difficult. Indeed, it takes about 8 hours for an expert annotator to segment a single laser scan (Hullo et al., 2015), which translates into months of work for annotating complex industrial plants.

The presence of structured outliers can hinder many downstream tasks, from scene reconstruction to performing correct measurements. For example, if an operator uses the raw 3D point cloud, containing outliers, to measure a distance between a piping and the viewer as illustrated in Fig. 2, picking an outlier point instead of a point from the actual geometry can lead to a vastly wrong measurement.

In this paper, we propose the first comprehensive investigation of automated removal of reflection-induced outlier points in complex industrial 3D scenes. Contrary to the majority of previous works on outlier removal, this problem setting requires detecting acquisition artifacts caused by real-world reflective surfaces that can be highly irregular and non-planar. This setup severely reduces the utility of both axiomatic approaches, and, as we demonstrate below, existing purely local learning-based methods.

Specifically, the problem that we consider is challenging for three main reasons. First, a successful method should adapt to any size of acquired point cloud for outlier detection in *large-scale* scenes. Second, local approaches are not applicable since reflection-induced outliers can be highly *structured* and resemble the underlying geometry (see Fig. 4). Third, only a very limited number of scenes with annotated ground truth is available, making large-scale learning difficult.

In this context, we demonstrate empirically that previous outlier detection methods fail on this task, in particular, because they fail to exploit scene-level contextual information.

* Corresponding author at: École polytechnique, IPParis, 1 rue Honoré d'Estienne d'Orves, Palaiseau, 91120, Essonne, France.
E-mail addresses: maximekirgo@gmail.com (M. Kirgo), guillaume.terrasse@edf.fr (G. Terrasse), guillaume.thibault@edf.fr (G. Thibault), maks@lix.polytechnique.fr (M. Ovsjanikov).

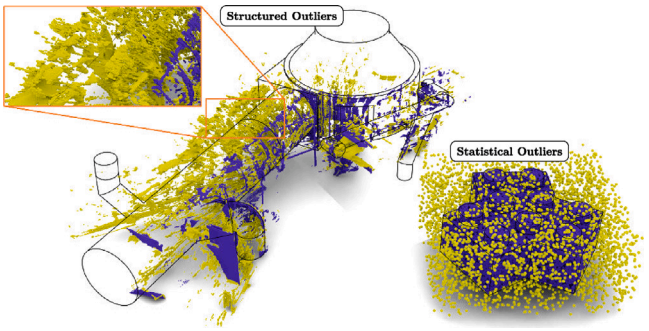

Fig. 1. In this paper, we present the problem of detecting reflection-induced *structured outliers*, arising in real-world acquisition (*left*), compared to more commonly-studied statistical outliers (*right*). Clean points are marked in blue, whereas outliers are shown in yellow. Note that clusters of reflection-induced outliers can closely resemble patches of clean geometry, rendering purely local approaches ineffective and requiring more global integrated scene analysis. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To address this challenge, we propose to formulate outlier detection as a semantic segmentation problem, unlike a pure labeling problem as done in most prior works. This reformulation, together with using an adapted semantic segmentation architecture, enables non-local interaction between outlier point decisions, which is crucial for our task.

Lastly, we demonstrate that a *visibility* signal, provided in addition to the standard 3D geometry and laser intensity information, both helps to regularize the training and significantly improves the accuracy of reflection-induced outlier detection, by injecting additional scene-level semantic cues to the learning process.

In summary, our contributions are:

- A new dataset for structured outlier detection in industrial scenes;
- An in-depth study highlighting that existing local, patch-based learning approaches have limited accuracy for reflection-induced outlier detection;
- A novel approach adapting a semantic segmentation architecture for reflection-induced outlier detection. Our method, notably, is capable of capturing long-range information within point clouds, which is crucial when dealing with real-world structured outliers;
- A further improvement demonstrating that computed visibility as an input signal allows to regularize structured outlier detection.

Both our code and complete dataset of labeled data (valid points and reflection-induced outliers) will be publicly released upon final acceptance of this paper.

The remainder of this paper is organized as follows. We first present an overview of related work and background on semantic segmentation of large-scale 3D point clouds (Section 2). We then present our general approach, including a description of the architecture design and network input in Section 3. Section 4 describes our dataset and compares it to the datasets for statistical outlier detection. Section 5 is dedicated to extensive experimental results and comparison of our approach to baselines, while Section 6 concludes the paper.

2. Related work

Reflection Detection. The detection of reflections in 3D scans has mostly been addressed from an axiomatic point of view. Yun and Sim (2018) propose an efficient method for detecting reflections caused by glass surfaces in architectural scenes, that they evaluate qualitatively on a dedicated dataset containing eleven scans. Initially limited to a single reflective plane per scene, this work was extended (Yun and Sim, 2019) to detect multiple glass planes. The main limitation of this method is its reliance on a careful parameter setup. Moreover, only glass *planes* are considered, which limits its applicability to more general reflective surfaces. Finally, their proposed dataset does not contain ground-truth annotations of the outlier points, preventing its use for supervised learning and for quantitative evaluations. A few recent works tackle the problem of detecting (planar) mirror surfaces in RGB-D interior scenes (Tan et al., 2021; Mei et al., 2021). Both approaches introduce new datasets that contain precise segmentation of planar mirrors in interior RGB-D images. More closely related to our approach, the detection of ghost targets (i.e. acquisition artifacts that resemble objects to detect) produced by reflections in scans of road scenes is addressed in two recent works by leveraging a PointNet-like (Chamseddine et al., 2021) and a Transformer (Wang et al., 2021; Gao et al., 2022) architectures. However, these approaches rely on the combination of multi-modal sensor data to detect ghost targets, whereas our approach focuses solely on the 3D geometry and laser scanner's intensity.

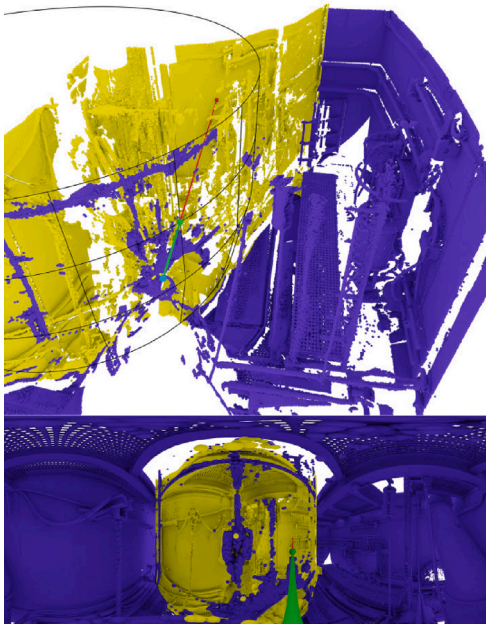

Fig. 2. Illustration of a measurement error induced by structured outliers (*yellow*) between the center of acquisition (*light blue dot*) and a point on a piping (*black lines*). The correct distance corresponds to the *green line segment* and the erroneous distance to the *red line segment*. Correctly acquired points are indicated in *purple*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Outlier Removal and denoising. Our task is closely linked to outlier detection and point cloud denoising. The topic of outlier detection has been commonly treated from a statistical perspective (Barnett and T., 1995; Maimon and Rokach, 2005; Rousseeuw and Hubert, 2011) by developing robust distribution analysis approaches with strong theoretical guaranties. Specific methods for 3D point clouds have been constructed in the past using adapted axiomatic methods (Fleishman et al., 2005; Cazals and Pouget, 2005, 2008). More recently, deep learning-based methods for both denoising and outlier removal have been formulated first using supervised training over local point cloud neighborhoods, as in PointCleanNet (Rakotosaona et al., 2020), and extended in follow-up works, using unsupervised learning (Hermosilla et al., 2019), manifold reconstruction (Luo and Hu, 2020), exploiting graph structures (Pistilli et al., 2020; Irfan and Magli, 2021), non-local information (Huang et al., 2020), encoder–decoder models (Zhang et al., 2020b) or score-based approaches (Luo and Hu, 2021). Unfortunately, the previously-mentioned approaches are tailored for statistical noise removal, arising close to the underlying surface. In contrast, our application scenario involves *structured* noise, that closely resembles real surfaces.

Semantic Segmentation. We rely on a 3D point cloud semantic segmentation network to solve the binary classification of inlier versus outlier points. Following the work of Qi et al. on PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b), a wealth of methods have been proposed to produce per-point labeling using raw point coordinates and features as input. These methods are commonly designated as *point-based* methods (Zhao et al., 2019; Duan et al., 2019; Yang et al., 2019; Yan et al., 2020; Hu et al., 2020). Some approaches leverage a graph structure computed from the point cloud that allows to take advantage of graph processing techniques and graph convolution (Wang et al., 2018a, 2019a; Feng et al., 2020; Bazazian and Nahata, 2020; Xu et al., 2020). Others make use of 2D projections, such as a perspective (Kundu et al., 2020), a spherical projection (Milioto et al., 2019) or both (Alnaggar et al., 2021). Point clouds can be encoded within a regular structure, such as a voxel grid (Wang et al., 2017; Zhou and Tuzel, 2018; Choy et al., 2019) or a custom layout of points (Wu et al., 2019; Zhang et al., 2019; Komarichev et al., 2019; Thomas et al., 2019; Boulch, 2020; Xu et al., 2020, 2021). Hybrid approaches, entangling 2D and 3D information (Jaritz et al., 2019; Robert et al., 2022) or fusing voxel and point-based approaches (Zhang et al., 2020a) have also been developed. Finally, Transformer-like architectures recently highlighted the interest of transposing self-attention mechanisms to point cloud segmentation (Lai et al., 2022; Guo et al., 2021; Zhao et al., 2021; Mazur and Lempitsky, 2021). The main contribution in all these work lies in the local point operator, acting on neighboring points, used in the segmentation architecture. Concerning the architecture itself, encoder–decoder, such as U-Nets (Ronneberger et al., 2015) or DeepLab (Chen et al., 2017, 2018), and HRNet-like (Wang et al., 2020) are the two main designs for semantic segmentation in the image domain. More recently, Vision-Transformers (Dosovitskiy et al., 2020; Zheng et al., 2021) have been proposed to leverage the ability of Transformer networks (Vaswani et al., 2017) to capture long-range relationships. In the point cloud processing community, the encoder–decoder strategy, advocated in PointNet++, represents the go-to architecture in the vast majority of recent semantic segmentation networks. We therefore chose to focus our study on this type of architecture and use a U-Net design. A notorious exception to the trend of using encoder–decoder architectures for semantic segmentation is DGCNN (Wang et al., 2019b), where an EdgeConv layer recomputes a nearest-neighbor graph on the input point cloud at each stage to build rich geometric features. DGCNN is not suited for processing point clouds with more than around a thousand points due to the expensiveness of the EdgeConv layer. All of these architectures have so far been used for labeling into semantically meaningful classes, rather than for outlier detection. One goal of our study is to highlight that semantic segmentation architectures are well suited for structured outlier removal.

Statistical outliers/noise datasets. The authors of PointCleanNet (Rakotosaona et al., 2020) propose a dataset with point clouds containing statistical outliers that were synthetically generated from the ground truth surfaces. We highlight the difference of our task to detecting such statistical noise by showing that a state-of-the-art denoising architecture, namely ScoreDenoise (Luo and Hu, 2021), fails at segmenting outliers (see Fig. 6).

Glass pane outliers datasets. The public benchmark proposed by the authors of Yun and Sim (2018) contains a collection of exterior large-scale point clouds containing outliers caused by the reflection of objects on planar glass panes. This setup is related to the problem targeted by our dataset, with three main differences. First, their dataset does not contain ground truth annotation, which prevents quantitative evaluations. Second, the reflective surfaces considered are in glass, a material that is not opaque as in our scenario. This difference is important because a transparent reflective material implies that (i) correctly acquired points can be found after the reflective interface and (ii) the intensity of the reflection is less pronounced in transparent material because a large portion of the laser energy actually gets through the glass interface (Yun and Sim, 2018). Finally, the reflective surfaces are exclusively planar, in strong contrast with our setup, where the reflective surfaces consist of piping with highly varying shapes.

Perfect mirror datasets. Our benchmark is also related to the datasets constructed for the detection of perfect mirrors, as proposed for RGB-D acquisitions in recent works (Tan et al., 2021; Mei et al., 2021). To illustrate the inefficiency of architectures targeting the task of detecting reflections on perfect mirrors, we compare our approach to a state-of-the-art method, namely the PDNet architecture (Mei et al., 2021) in Appendix C.

3. Our approach: ReVISOR

As mentioned above, our main goal is to develop a fully automatic approach for reflection-induced outlier removal. Moreover, as hinted earlier, and as we show extensively in our results (Section 5), to address this problem it is important to develop a data-driven solution capable of exploiting non-local cues for successful structured outlier detection.

Our method is based on three key building blocks: first, we propose to use a semantic segmentation architecture for outlier detection. Secondly, we introduce *estimated* visibility and laser intensity as additional input signals to the network. Third, we introduce a novel carefully curated dataset, that enables supervised training and evaluation for reflection-induced outlier removal. In Section 3.1 we introduce the problem setting and provide a general motivation and overview of our approach. In Section 3.2 we present our proposed architecture design, whereas Section 3.3 describes the input to the network, including the point cloud decomposition and our proposed visibility signal as an additional feature input.

3.1. Motivation and overview

Given a point cloud P consisting of an unordered set of point $P = \{p_i\}_{i=[1..n]}$, where $p_i \in \mathbb{R}^3$, the goal of outlier detection is to label each point p_i as either belonging to the real underlying geometry or being an *outlier*. A common learning-based approach for outlier detection, as introduced in Rakotosaona et al. (2020), is to consider the local neighborhood $\mathcal{N}_i$ of every point p_i within the point cloud P, and to make a prediction of whether p_i is an outlier, by feeding $\mathcal{N}_i$ to some trainable predictor $\mathcal{F}_\Theta(\mathcal{N}_i)$, where Θ are the parameters of the network $\mathcal{F}$. For example, in Rakotosaona et al. (2020) the authors used PointNet (Qi et al., 2017a) as the backbone network $\mathcal{F}$ and $\mathcal{N}_i$ was built via $\mathcal{N}_i = \{p_j \in P, \text{ s.t. } \|p_j - p_i\| < r\}$, i.e., all points p_j within P that are less than some fixed distance r away from p_i. Intuitively, by analyzing the distribution of points in the neighborhood of p_i it should be possible to decide whether p_i is an outlier. Unfortunately, despite its simplicity, this method has two fundamental limitations: first, the predictions at different points are done independently, since each point is simply considered as the centroid of its patch. This both can lead to discontinuous predictions within the larger point cloud, and furthermore can make detecting *structured* outliers difficult, as such decisions might depend on correlations between different points. The second limitation of this approach is the limited receptive field size of $\mathcal{F}$, which only considers a *small local patch* around each point. As we demonstrate below, this leads to very poor prediction accuracy of reflection-induced outlier detection.

To address these issues, we propose to replace the local classification-based approach introduced in Rakotosaona et al. (2020) with a semantic segmentation network. Thus, we propose to decompose the input point cloud that represents an industrial scene into a set of *large overlapping regions*. We then feed each region R_m into a *semantic segmentation network* $\mathcal{G}$ so that its output $f = \mathcal{G}(R_m)$ gives a prediction score $f(p_i)$ for every point p_i in R_m of whether it is an outlier. The key difference from the approach described above is that the predictions for all the points within R_m are made *jointly*, and furthermore, we use a significantly larger receptive field size by decomposing a scene into a small set of overlapping regions rather than associating a local patch

to each point in the point cloud. Finally, at test time, if $\{R_k\}_{k\in[0,K]}$ regions overlap at a given point $p_{overlap}$, we compute the final prediction $f(p_{overlap})$ by averaging over the predictions in each individual region $\{f_k\}_{k\in[0,K]}$: $f(p_{overlap}) = \frac{1}{k}\sum_{k=0}^{K} f_k(p_{overlap})$.

3.2. Architecture design

As highlighted above, we propose to use a semantic segmentation architecture to perform the classification of the input point clouds into inliers and outliers.

Our network architecture, illustrated in Fig. 3, closely follows the design recently advocated in Liu et al. (2020). It consists in a Residual U-Net network with the following building blocks:

- A down-sampling block: we combine a strided residual block with two residual blocks, both leveraging the chosen convolution operator in their middle layer (see the light-blue box on the right of Fig. 3). The down-sampling is achieved by performing a grid sub-sampling.
- An up-sampling block: we use a 1-nearest neighbor upsampling to project the low-resolution features on the points of the next resolution, followed by a simple multi-layered perceptron.

A key design decision within our architecture is the choice of local convolution operator. In our default implementation we use the pseudo-grid kernel point convolution (Thomas et al., 2019) (*Grid.*). This operation is obtained by first placing a kernel of points with fixed position at each input point. After this, neighboring features, weighted by their relative distance to the closest kernel point, are summed to produce a new feature value at each kernel point location. The contribution of each kernel location is multiplied by a weight to produce the output feature at each point location. The fact that the layout of the kernel points is regular, as if the points were lying on a grid explains the name of this approach.

We have also performed extensive experiments to compare this choice of convolution operation with alternative approaches, including more recent ones based on sparse convolution with the MinkowskiEngine (Choy et al., 2019), PosPool (Liu et al., 2020), adaptive weights (Wang et al., 2018b), multi-layered perceptrons and Point Transformer (Zhao et al., 2021). In Appendix A, we present a comparison of our default implementation with different convolution operations. Interestingly, most of the conclusions that we draw (e.g., on the role of the receptive field size) remain independent of the choice of the convolution operation. However, pseudo-grid kernel point convolution produces the best results in practice, which is why we adopt it throughout our approach.

Layer setup. All layers are followed by batch-normalization with momentum 0.9, followed by ReLu non-linearity. At each down sampling stage, the radius and the grid subsampling size of the previous step are multiplied by 2. Conversely, at each up-sampling stage, the radius and grid subsampling size are divided by 2. The base radius size is set to 1/32th of the input radius.

Number of points. The number of sampled points depend on the characteristics of the dataset considered. In our study, we focus on a dataset with statistical outliers and a real-world dataset featuring structured outliers. **For statistical outliers**, the input number of points is set to 500 for all receptive fields, following the choice of the authors of PointCleanNet (Rakotosaona et al., 2020). **For structured outliers**, the input number of points is set to 15000 for the largest receptive field radii (starting at radii $\approx$ 0.5 m), following the parameter setup of the authors of PosPool (Liu et al., 2020). For smaller radii, the number of points is linearly decreased down to a minimum number of points of 1024. For instance, a patch radius of 0.3 m uses 7500 input points.

Spatial subsampling. To reduce the computational footprint of our method (see Appendix F for our computation hardware details), we subsample all our scenes to a 5 mm spatial subsampling.

Hierarchical down-sampling and up-sampling. The ResUNet architecture design that we employ leverages a U-Net-like structure (Ronneberger et al., 2015), that requires to down-sample the input patch of points progressively in the left branch of the "U" and to upsample it back in its right branch. Different choices are possible to downsample an input point cloud. The most common options (Hu et al., 2020) are farthest point sampling (FPS), random sampling and grid subsampling. For the experiments presented in this paper, we select the latter because it allows a more natural comparison with voxel-based methods.

3.3. Network input

Point cloud decomposition The input of our default network consists of the 3D XYZ coordinates of the points contained within a spherical neighborhood of the full point cloud (i.e. the coordinates are centered around the neighborhood's center). This spherical neighborhood is the *receptive field* or, as mentioned in Section 3.1, the input region R_m of our network.

During training, we randomly select region centers among "inlier" and "outlier" points from all training scenes, so that each batch contains an equal number of "inlier" and "outlier" patch centers. This selection ensures that the network is equally exposed to patches containing both classes of points, even when considering highly imbalanced datasets.

At test time, we subsample each point cloud using a subsampling distance of half the diameter of the network's receptive field and use the corresponding points as the centers of the patches for which the network will perform a prediction. As described in Section 3.1, we average the overlapping predictions as they occur at test time.

Point Visibility as an Additional Input Feature Our setup differs from standard statistical outlier removal mainly because *local* information is insufficient to decide whether a patch of points was taken from actual geometry or belongs to a cluster of structured outliers that resemble the underlying geometry (see Fig. 2 top). In addition to proposing a novel adapted semantic segmentation architecture, as described above, we also introduce *additional non-local features*, as input to the network, that are dependent on the long-range interaction between points in the 3D point cloud.

Visibility is a non-local property: given a point of view, a point can be occluded with respect to this point of view by another point, that can be arbitrarily far away. We observe that in the context of *reflection-induced outliers*, visibility is particularly meaningful since the vast majority of the outlier points should be occluded by points correctly acquired on the reflective surfaces and are henceforth much more to likely be tagged as "invisible" than inlier points. See for example the inset of Fig. 1. Interestingly, we note (and demonstrate empirically in Section 5) that using a classical non-learning based method to estimate visibility and using it as an additional signal to the network boosts the overall performance of our pipeline significantly. **Visibility Computation** A widely used approach for determining the visibility of points in the input point cloud has been introduced by Katz et al. (2007). In our framework, we exploit it as an additional guiding input signal for non-local outlier detection.

Indeed, since the visibility computation is conducted on a *complete scene*, it leverages *non-local information* in the sense that more information than a point cloud patch is available at the time of its computation. The approach proposed in Katz et al. (2007) for computing visibility consists in flipping all points in an input point cloud on the exterior of a virtual sphere, with fixed radius, centered at a viewpoint and encompassing the entire point cloud. The points lying on the convex hull of the resulting point cloud represent the points visible from the viewpoint considered for the given radius of the sphere. This radius is the unique hyperparameter of the algorithm. For completeness, in Appendix D we provide additional details on the visibility feature computation and parameterization.

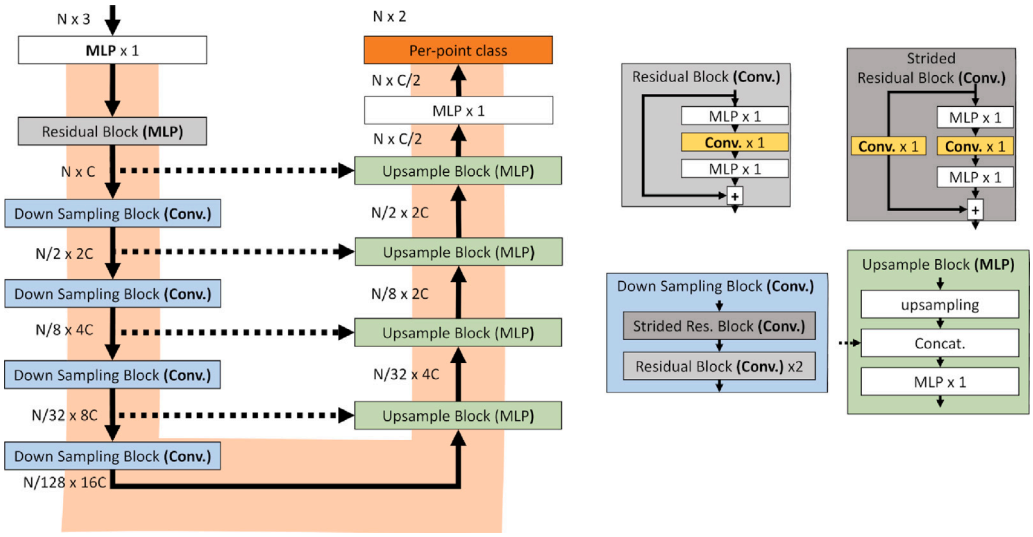

Fig. 3. Illustration of our network architecture following the design adopted in Liu et al. (2020). *N* and *C* indicate respectively the number of points and the feature dimensionality, *MLP* stands for multi-layered perceptron, *Conv.* designates the convolution operator employed and *Res.* stands for residual.

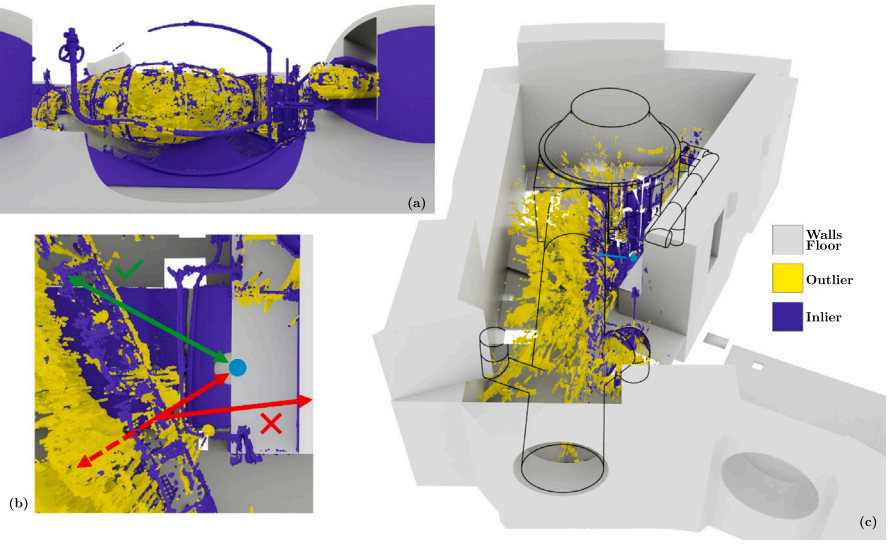

Fig. 4. Illustration of a scanned room, shown from different viewpoints. Figure *(a)* shows the panoramic view of the acquired scene (i.e. the viewpoint of the acquisition device). Notice the large quantity of outlier points (*yellow*). Figure *(b)*, an orthographic view of the room near the scanning device (*cyan point*) with the trajectory of the laser beam depicted with arrows. The *green solid arrow* corresponds to the trajectory of a correctly acquired point, the *red solid arrow* shows the trajectory of a reflected ray and the *red dashed arrow* represents the erroneous trajectory inferred by the acquisition device based on the laser signal it received back. Figure *(c)* illustrates a global view of the point cloud with *color-coded* ground truth annotation, a reconstructed 3D model of walls and floor *in gray* to provide semantic context, and the real surface of the reflective piping in *solid black line*. The location of the acquisition center is shown by a *cyan point*, and the viewing direction by a *cyan arrow*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Datasets

4.1. A real-world structured outlier dataset

As mentioned above, obtaining labeled data for real-world structured non-local outlier detection is difficult. Therefore, little effort has been done to assess the quantitative performance of different approaches for this task. To the best of our knowledge, our dataset is the first to be specifically designed for benchmarking outlier detection in an industrial context, that contains reflection-induced artifacts. We first present the content of the dataset before highlighting the novelty of our proposed downstream task with respect to existing ones.

4.1.1. Overview

Our dataset is composed of a collection of rooms from industrial power-plants acquired via a fixed TLS (Terrestrial Laser scanning) device mounted on a tripod that produces large-scale point clouds (around 45 Million points per scene). These rooms contain piping that has the property of being heat-insulated with a highly reflective material. The main task is to detect the outliers produced by the deflection of the TLS system's laser beam in order to remove them. This setup is especially challenging compared to traditional indoor or outdoor scenes (Behley et al., 2019; Hackel et al., 2017; Dai et al., 2017; Armeni et al., 2017; Song et al., 2015) because industrial facilities are environments in which objects are highly clustered, as illustrated in Fig. 5. We adopt the

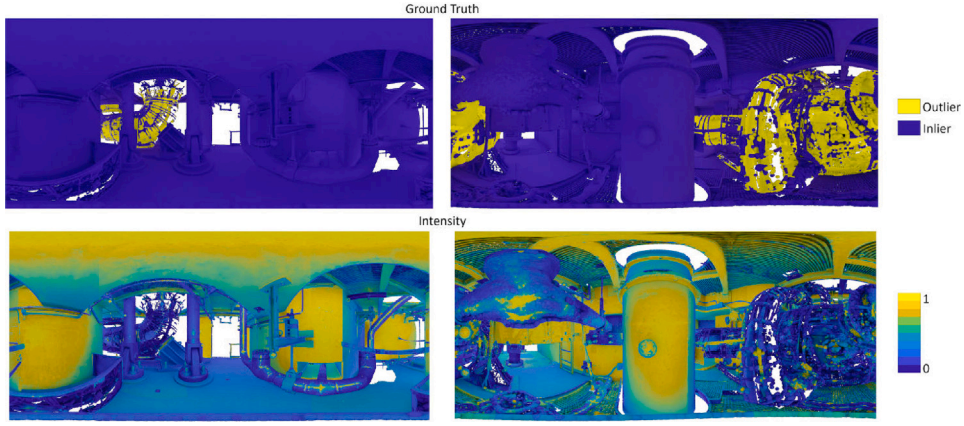

Fig. 5. Examples of two rooms from our dataset. The top row depicts the ground truth segmentation between Inliers and Outliers. The bottom row presents the normalized color-coded intensity signal.

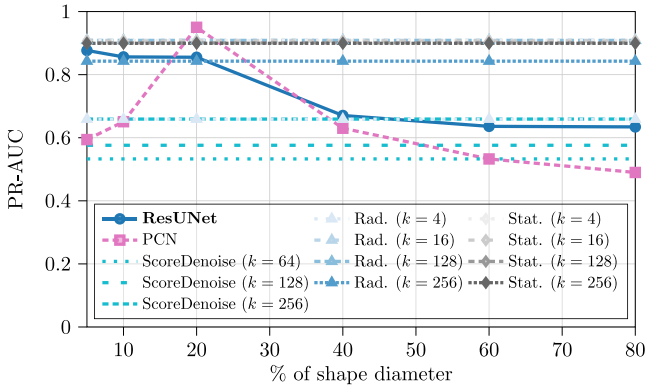

Fig. 6. Performance of our approach, two non-learning local outlier removal techniques *Rad.* and *Stat.*, PointCleanNet (*PCN*) and ScoreDenoise, evaluated at increasing input patch radii for the semantic segmentation network. The evaluation is conducted on the shapes of the FAMOUSTHINGI test set. The radius is expressed as a percentage of the shape diameter. *PR-AUC* stands for precision–recall area under the curve.

following labeling convention: the negative class (0) represents **inliers**, while the positive class (1) encodes **outliers**.

Content. The dataset consists in 21 scans stemming from 14 stations of 3 different facilities of Électricité de France. The point clouds were acquired using a Leica Z+F 5010C and a Leica HDS 7000 laser scanners. Both are high quality laser scanners, and produce point clouds with a density of at least one point every 3 mm at 5 m of the acquisition center. The local noise on standard materials is at most of 1 mm in terms of RMS.

The labeling procedure was conducted by a trained field expert using the 3D point cloud processing software RealWorks by Trimble. Additional external cues including panoramic images of the environments and floor plans of the industrial plants were used for context during the labeling procedure. In ambiguous cases, the "inlier" label was assigned by default. Each scan is equipped with estimated normals, the normalized laser intensity signal and the annotation of the inlier and outlier points provided by the expert. For our testing set, the data was cross-validated by a rigorous internal process, that also uses ground truth floor plans and 2D image views.

Train/Test split: Throughout all comparisons we used 13 scans from two facilities for training and validation, while the remaining 8 are used to evaluate the method at test time. Note that the test set contains stations from an entirely different facility than those used during training. This makes the problem particularly challenging as a successful method must not only be able to exploit long-range correlations within the input point cloud, but also be sufficiently skilled to generalize to entirely new, unseen scenes from different industrial facilities.

Table 1
Statistics on our dataset. It consists in 21 scans of 3 facilities of Électricité de France, across 14 scenes.

Quantity	Train	Validate	Test	Total
# points (sum)	15 048 663	6 872 918	2 693 886	38 130 522
# points (mean/scan)	1 504 866	2 290 973	2 026 118	1 815 739
% outliers (mean/scan)	34.6	20.9	20.5	27.2

During preprocessing, each scan is first downsampled to a spatial resolution of 5 mm. A manual distance thresholding was then performed along the vertical axis to roughly trim the points lying on different floors, as well as a distance thresholding, so that points lie within a 5 m sphere around the acquisition device. Table 1 summarizes statistics on the number of points and the percentage of outliers present in the dataset.

4.1.2. Novelty of our problem setup

As highlighted above, our setup is novel and challenging because we consider *structured non-local* outliers caused by *arbitrary* and *irregular* reflective surfaces, which is neither standard (statistical, non-structured) outlier detection that involves artifacts lying nearby real geometry, nor reflection detection on planar glass panes or perfect mirrors. In Section 4.2, we give an overview of the most closely related dataset to ours, and highlight the difference of our setup.

Although our setup is related to mirror detection in RGB-D data via 2.5D techniques (i.e. using 2D convolutional neural networks with depth as an additional channel), our problem is both different and more challenging because: (i) we have a limited amount of data at our disposal, as highlighted in Appendix C; (ii) the reflective surfaces that we consider are not planar and rectangular, but can have a great variety of configurations (see the illustrations of our dataset in Appendix B); (iii) we focus on unorganized 3D point clouds, making the resulting method versatile and independent of the acquisition method.

4.2. Statistical outlier datasets

When considering statistical outliers, we train different methods on the POINTCLEANNET dataset for outlier detection introduced in Rakotosaona et al. (2020) and evaluate on the dataset FAMOUSTHINGI (Rakotosaona et al., 2021). The first dataset presents a collection of 28 shapes, sampled in distinct point clouds, each containing 140K points with 40% of outliers. The second dataset consists of 31 shapes, from which we sample 50K surface points with the same percentage of outliers. Both datasets share the property of containing exclusively man-made

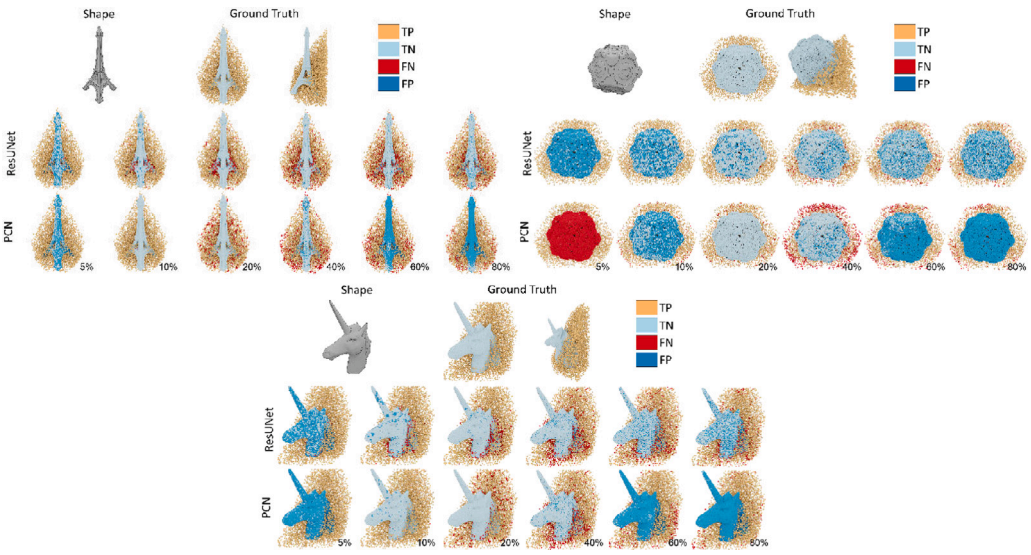

Fig. 7. Outlier detection on our statistical outliers test set (at the top left, top right and bottom are respectively the "Eiffel_Tower_mini", "companion-dodec" and "unicorn" shapes), using either a semantic segmentation ResNet (*ResUNet*) or PointCleanNet (*PCN*). *TP*, *TN*, *FP* and *TN* respectively designate true positives, true negatives, false positives and false negatives.

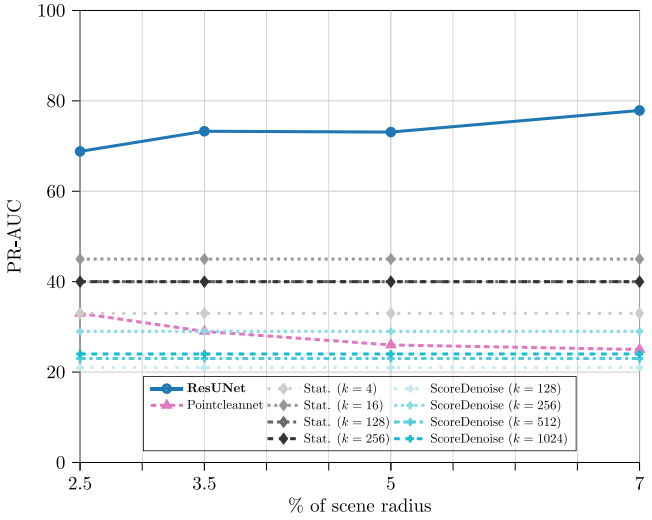

Fig. 8. Performance in terms of area under the precision–recall curve (PR-AUC), of our approach using only raw XYZ coordinates at increasing input patch radii compared to local denoising approaches on our dataset. The radius is expressed as a percentage of the maximal scene radius (5 m).

shapes, and only contain *synthetic outlier points*, unlike our dataset which is composed of real-world data. The outlier points are generated by adding points with random coordinates uniformly distributed in the unit cube that contains each shape. This setting is of course vastly different from point clouds produced by real-world acquisitions. This setup is nevertheless common to train and evaluate a denoising pipeline. We choose to evaluate on FamousThingi instead of the test set of the PointCleanNet dataset in order to have more evaluation shapes than this test set contains (10 shapes) and to achieve a greater variety between the training and testing shapes. Indeed, both datasets have no shape in common.

4.3. Evaluation metrics

To study the effect of changing the receptive field size and the type of convolution operator, we leverage the area under the precision–recall curve (PR-AUC) metric. This metric allows to summarize the performance of each tested model under varying decision thresholds. Moreover, the inlier-outlier distribution is imbalanced (there are roughly 80% inliers for 20% outliers in the dataset) and this metric is especially suited for imbalanced class distributions (Davis and Goadrich, 2006).

We also evaluate different designs using standard evaluation metrics, namely the accuracy, the precision, recall and mean intersection over union.

5. Results

5.1. Baselines

In the experiments that we carry on in this section, we consider the following baselines.

- **PointCleanNet** (Rakotosaona et al., 2020) **(*PCN*).** PointCleanNet proposes a two-step denoising pipeline: first, the outlier points are detected by a network that outputs an outlier score, trained with the L_1 regression loss; second, the remaining "inlier" points are denoised by estimating an offset vector. Note that both networks take a patch of points as input and output a score/denoising offset *for the center point of the patch only*. For this study, we only consider the outlier detection part that we retrained using the same setup as for the other networks.
- **ScoreDenoise** (Luo and Hu, 2021)**.** This network is a state-of-the-art denoising architecture that leverages EdgeConv-like (Wang et al., 2019b) convolution operations. The network predicts a gradient score that allows to displace noisy points back to their original position. For our comparison, we retrain the network to output a per-point probability for the inlier and outlier class, trained with the cross-entropy loss.
- **Statistical outlier removal (*Stat.*).** Points that are further away from their neighbors than the average distance for the full point cloud are labeled as outliers. We use Open3D's implementation (Zhou et al., 2018) and consider a neighbor number of 4, 16, 128 and 256.
- **Radius outlier removal (*Rad.*).** Points that have a number of neighbors within a spherical neighborhood smaller than a given threshold are considered as outliers. We also use Open3D's implementation for this method with the same number of neighbors.

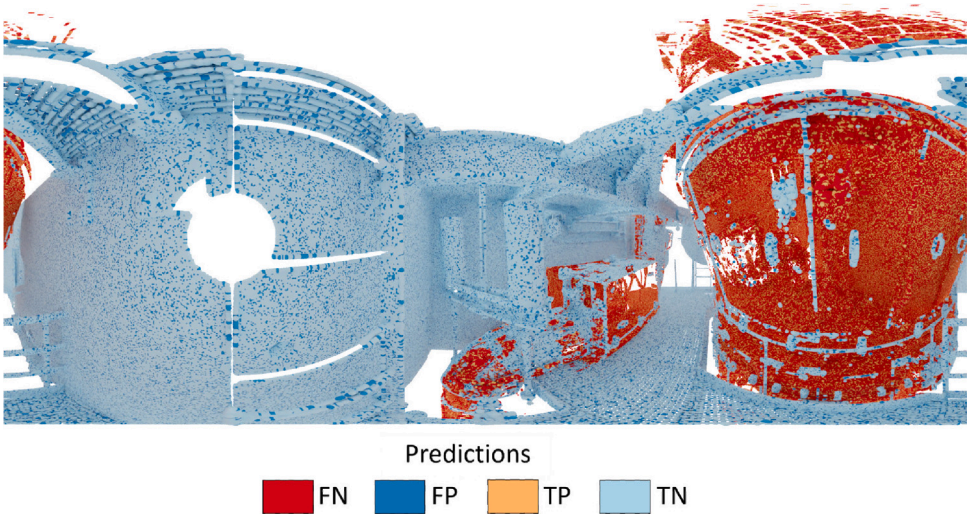

Fig. 9. Qualitative illustration of the predictions obtained with ScoreDenoise ($k = 256$). We color-code the true negative (*TN*), true positive (*TP*), false positive (*FP*) and false negative (*FN*). Note how bad the outlier class is segmented (high rate of false negatives).

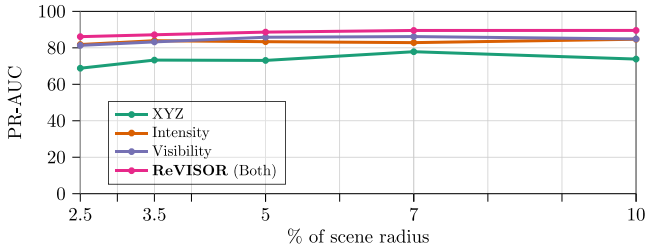

Fig. 10. AUC under the precision–recall curve, expressed in percent, of our network using different sets of features as a function of the patch radius size, expressed in percent of the maximal scene radius (5 m). The "Grid" local point operator is used. Note how the combination of visibility and intensity yields the best results.

Note that the two last methods are non-learning and are extremely close in spirit. Therefore, we only display results using the statistical outlier removal baseline in our experiments on the real-world dataset (Section 5.3).

5.2. Pilot study

As a first experiment, we use the statistical outlier dataset proposed by the authors of Rakotosaona et al. (2020) to train a ResNet with pseudo-grid convolution operator, compared to our baselines (see Section 5.1).

The evaluation is performed on 31 test shapes of the FAMOUSTHINGI dataset (Rakotosaona et al., 2021), from which we sampled $50K$ points, with a proportion of 40% outliers, randomly sampled within the unit cube.

Fig. 6 depicts the outcome of this experiment. First, we note that the two non-learning approaches *Stat.* and *Rad.* are competitive with the learning-based approaches *when selecting the proper k parameter value*. Second, the ResNet consistently outperforms ScoreDenoise in terms of precision–recall AUC (PR-AUC) and is comparable to PCN for radius values of 20% of shape diameter. Third, the AUC decays for a radius greater than 20% of the shape's diameter for both PCN and the ResNet, which indicates that adding long-range context is unimportant for statistical denoising. Finally, PCN requires a careful tuning of its receptive field size to obtain maximal performance, and given such an appropriately-chosen receptive field size, this architecture does provide the best overall performance. This problem is thus indeed solvable using PCN, as claimed by the authors of that work. This last conclusion is in strong contrast to the reflection-induced outliers problem as we will see in the next section.

Fig. 7 qualitatively illustrates the prediction obtained with PCN and the ResNet on the evaluation dataset. We note that the ResNet's predictions are *smoother* than those of PCN and that the decay in PR-AUC also translates to worse predictions for both architectures.

5.3. Results on the real-world dataset

5.3.1. Impact of the receptive field size

In order to compare our proposed approach to competing baselines, we first train and evaluate the different models on our dataset at increasing receptive fields, up to 7% of the total scene radius (5 m). The result of this experiment is shown in Fig. 8 in terms of PR-AUC.

We note that the non-learning approach LOF performs poorly, due to its local and solely density-driven nature. Moreover, the evaluation clearly highlights the ineffectiveness of traditional denoising pipelines: neither PCN nor ScoreDenoise provide meaningful predictions in this context.

On the contrary, our approach vastly outperforms the aforementioned baseline approaches and we observe an increase in performance as the receptive field of the network is increased: at 7% of the scene diameter, the performance of our model is maximal with a PR-AUC of 77.87%. The additional challenge that represent our new dataset can be seen when comparing this maximal value with the best performing method of the pilot study (90% of PR-AUC). Fig. 9 provides a qualitative illustration of ScoreDenoise's performance on a test scene with $k = 256$, the best performing version of this network on our experiments. The prediction is biased towards the inlier class and fails to accurately predict the outlier class. PCN provides predictions of a similar quality.

In general, the results obtained on our real-world dataset are in strong contrast with the outcome of our pilot study (Section 5.2).

Finally, Table 2 compares our method to PCN and ScoreDenoise when computing an inference on a point cloud with around 2.3M points. We use the same setting for all networks as for our other experiments. The comparison shows that our method is faster to compute than our competitors by orders of magnitude. This is due to the fact that we perform predictions on a full patch of points. On the contrary, PCN, processes each point in a point cloud individually. Similarly, ScoreDenoise can only infer a few points per patch due to its construction of a local graph structure.

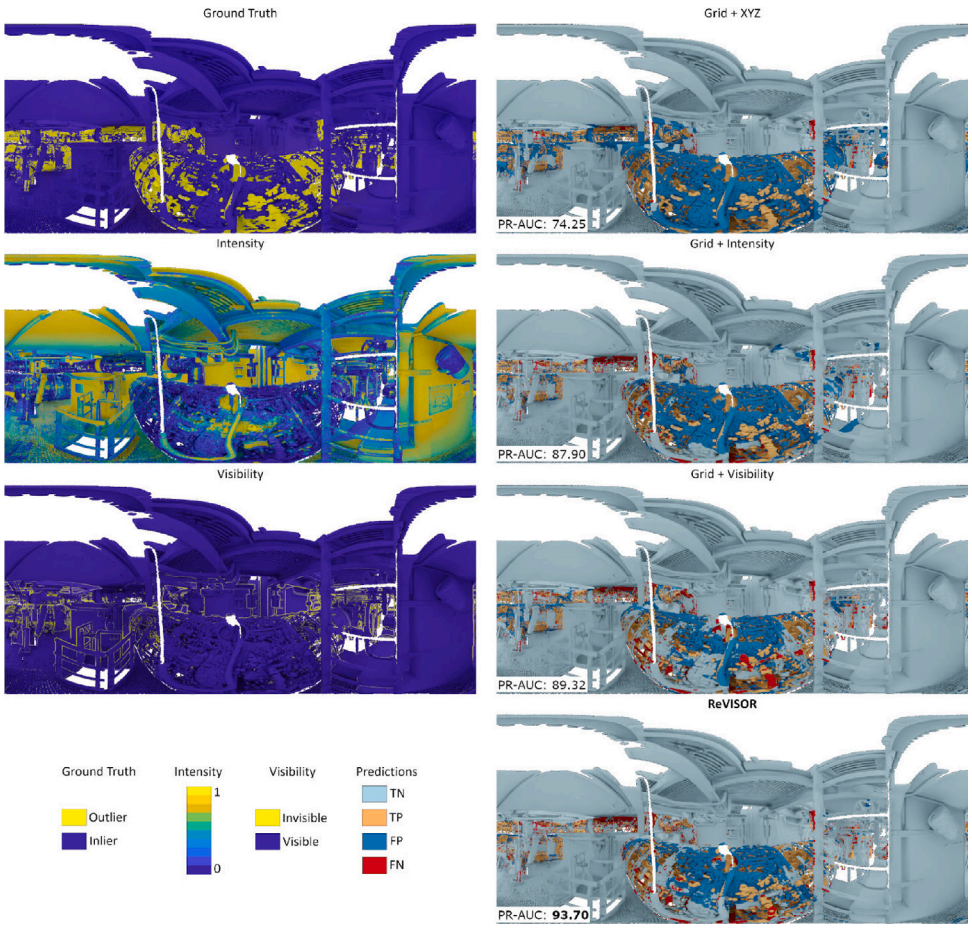

Fig. 11. Qualitative illustration of the interest of adding features on a "standard" scene from our test set. In the left column, we display from top to bottom the ground truth, the intensity signal and the computed visibility feature. In the right column, we show the true negative (*TN*), true positive (*TP*), false positive (*FP*) and false negative (*FN*) when using the different feature sets and the *Grid* local point operator. The positive class designates the outlier points, and the negative class the inlier points.

Table 2
Inference computation time of PCN, ScoreDenoise and our approach on a point cloud with around 2.3M points.

Network	PCN	ScoreDenoise	Ours
Execution time	2 days, 34 min 22 s	19 min 31 s	**2 min 26 s**

5.3.2. Visibility and intensity as input feature: a quantitative study

Input features are paramount to efficiently solve a semantic segmentation task. In our setup, instead of the traditional per-point RGB color information, the intensity value of the laser acquisition is available at each point. Our study aims at highlighting that a joint use of intensity and visibility reaches optimal performance. This result is largely independent of the local point operator used.

The interest of using a combination of visibility and intensity feature appears clearly when computing the PR-AUC, as can be seen in Fig. 10. We provide more experimental results on the interest of jointly using intensity and visibility in Appendix E.

First, we note that in all cases, the PR-AUC value decays after attaining its maximum and reaches values that are smaller than the ones observed at 7 % of the scene radius.

Second, we note that the visibility and intensity signals have very similar behaviors. They perform significantly better than raw coordinates for all receptive field sizes. We also observe that ReVISOR roughly *maintains* the same PR-AUC across large receptive fields and that it stays the best performing design compared to the sole use of intensity or visibility.

Our main hypothesis to explain the interest of using both intensity and visibility for the segmentation of structured outliers lies in the joint characteristics of these features. More specifically, both features behave in a "complementary manner" in terms of segmentation performance when employing a simple thresholding of their value. This is due to the fact that (i) intensity is low, whereas visibility is high in structured outlier regions and (ii) intensity is sensitive to physical properties of the acquired surfaces whereas visibility is not. We provide further analysis on this matter in Appendix E.

Figs. 11–13 illustrates the use of different input features on the same test scans of our dataset.

Adding intensity marginally reduces the number of false positives, at the cost of more false negatives in Fig. 11. The incorporation of visibility provides fewer false negatives in areas far from the acquisition center and on slanted surfaces.

We further note in Fig. A.16 that adding intensity alone does not allow to correctly label as an inlier region the sphere-shaped device located in the middle of the image, whereas the visibility-enhanced architectures label this equipment correctly.

As an additional comparison, we also evaluated a classifier that labels as outliers **all** points in the scene displayed on Fig. 12: the resulting PR-AUC is 23.21%, which is significantly lower than the models that we display.

In Fig. 13, the addition of features does not improve the detection performance of the large reflective area on the right hand-side of the image. Nevertheless, the small piping located in the middle of the image present significantly less false positives and false negatives when using

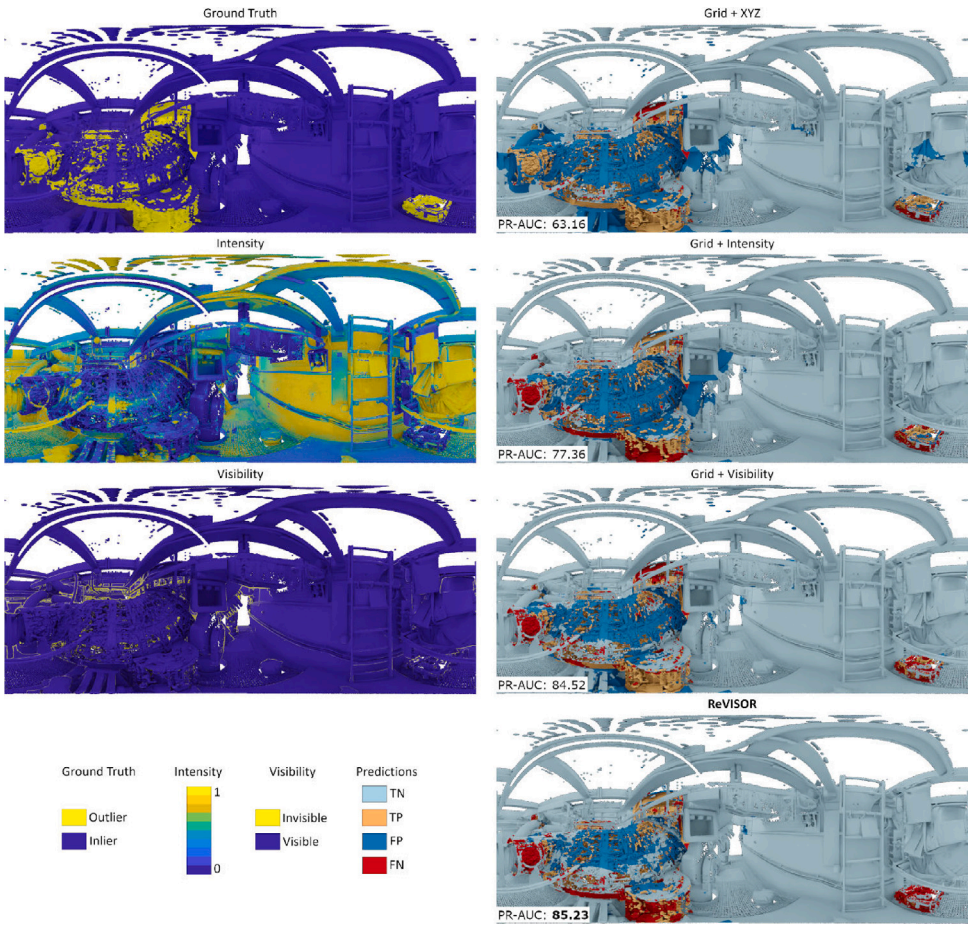

Fig. 12. Qualitative illustration of the interest of adding features on a "standard" scene from our test set. In the left column, we display from top to bottom the ground truth, the intensity signal and the computed visibility feature. In the right column, we show the true negative (*TN*), true positive (*TP*), false positive (*FP*) and false negative (*FN*) when using the different feature sets and the *Grid* local point operator. The positive class designates the outlier points, and the negative class the inlier points.

our ReVISOR feature setup, rather than raw coordinates, intensity alone or visibility alone. Fig. 13 is a "difficult" setup: the strong reflection on the right side of the acquisition is associated with a high intensity value. Moreover, since this region is close to the acquisition device and since the parts of the piping that were correctly acquired have a small area, the visibility signal is less effective.

6. Conclusion

In this paper, we studied two setups for outlier detection: (i) "statistical" outliers generated on synthetic shapes via random *local* 3D displacements and (ii) *structured*, *non-local* outliers present in real-world acquisitions of industrial power-plants.

Our study on statistical outliers, that our research community has mostly focused on, highlighted that learning-based approaches with *small receptive fields* are key to produce better results than non-learning techniques that remain competitive in this setup. A careful choice of receptive field size is required to obtain maximal performance and the quality of predictions decays rapidly for large receptive field sizes.

For the second class of outliers, we presented a new dataset, specifically designed for segmenting reflection-induced outliers in large-scale 3D point clouds. The dataset contains industrial TLS stations with highly reflective piping, which provides a challenging, real-world use case scenario for structured outlier detection. The dataset proposes 21 diverse industrial TLS stations, corrupted with structured noise. It is annotated by experts with many years of experience in this task, and cross-validated by a rigorous internal process that uses ground truth floor plans and 2D image views for our testing set.

We propose the first effective baseline approach for this problem, based on a semantic segmentation network with adapted features. It strongly outperforms existing methods for our structured outlier detection problem. We investigate the role of the *receptive field size* of different architectures, and highlight the importance in our context of medium to large patches, since locally many outlier patches resemble clean geometry. We demonstrate the utility of *visibility* features, which help boost the performance, again by providing cues about the non-local configuration of objects. The best performance is obtained when using the laser intensity and the point visibility as input signal, regardless of the convolution operator employed. Hence, we denote the proposed pipeline as ResUNets with Visibility and Intensity for Structured Outlier Removal (**ReVISOR**).

The main limitation of our framework is its supervised nature. Labeled data is hard to obtain for this class of problem and therefore, the size of our dataset is limited compared to other 3D point cloud datasets. New mobile hand-held laser scanning devices are likely to introduce other types of structured noise, which constitutes an opportunity to enrich our dataset and apply our methodology to a more general task, where the acquisition center is not clearly identified anymore.

Another perspective would be to better exploit long-range dependencies without relying on large patches. Making the visibility computation differentiable would be of interest in this regard, as it would allow to adjust the computation of visibility so that outlier points are considered as "invisible" in more ambiguous or difficult cases. With an increased amount of data, other architecture designs than Residual U-Nets could be considered to efficiently encode large-scale information. Graph-based approaches with super-points (Landrieu and Simonovsky,

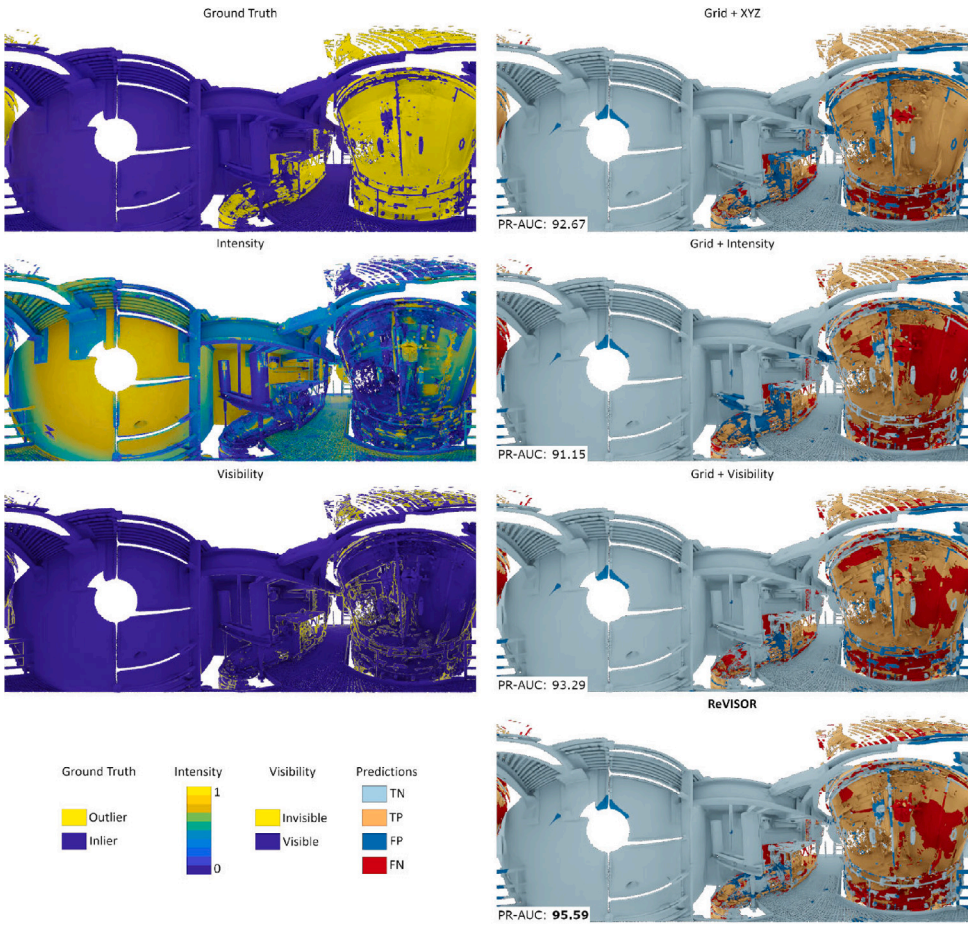

Fig. 13. Qualitative illustration of the interest of adding features on a "difficult" scene from our test set. In the left column, we display from top to bottom the ground truth, the intensity signal and the computed visibility feature. In the right column, we show the true negative (*TN*), true positive (*TP*), false positive (*FP*) and false negative (*FN*) when using the different feature sets and the *Grid* local point operator. The positive class designates the outlier points, and the negative class the inlier points.

2018) or Transformer networks (Yu et al., 2022) for example could be trained on a larger dataset. We see these extensions of our approach as an exciting direction for future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by the ERC Starting Grant No. 758800 (EXPROTEA), the ANR AI Chair AIGRETTE and the ANRT CIFRE Convention Nº 2019/0433.

Appendix A. Comparison of different convolution operations

We have compared a range of different backbone convolution approaches when implementing our residual U-Net architecture. Specifically we have compared the pseudo-grid kernel point convolution (Thomas et al., 2019) used in our default implementation with sparse convolution using the MinkowskiEngine (Choy et al., 2019), PosPool (Liu et al., 2020), adaptive weights (Wang et al., 2018b), multi-layered perceptrons and point transformer (Zhao et al., 2021):

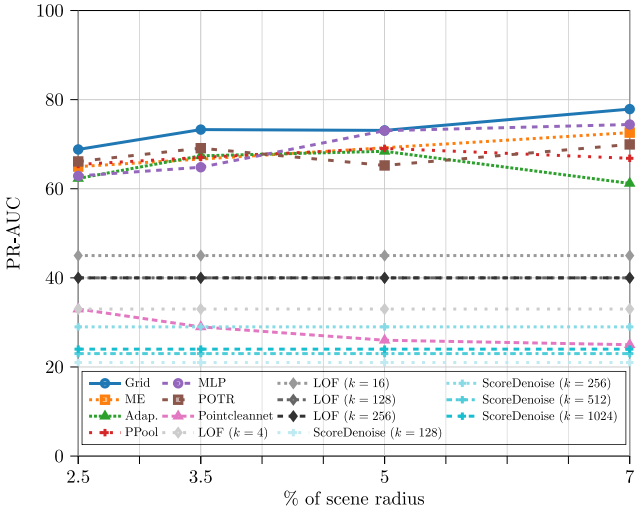

Fig. A.14. Performance in terms of area under the precision–recall curve (PR-AUC), of a semantic segmentation ResUNet using various local point operator at increasing input patch radii compared to local denoising approaches on our dataset. The radius is expressed as a percentage of the maximal scene radius (5 m).

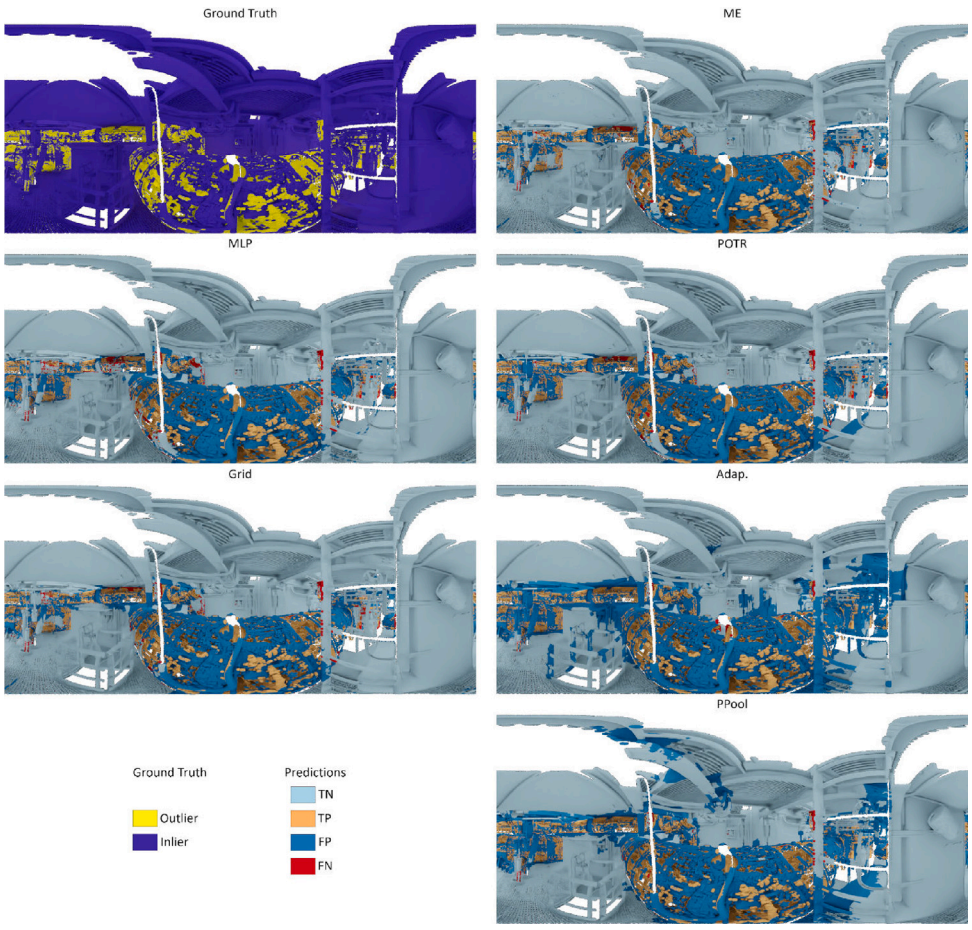

Fig. A.15. Qualitative evaluation of different local point operators, without features. The ground truth (*GT*) is provided in the top left cell, while each following cell depicts the true negative (*TN*), true positive (*TP*), false positive (*FP*) and false negative (*FN*) when using MinkowskiEngine (*ME*), PosPool (*PPool*), Adaptive weights (*Adap.*), Pseudo-grid (*Grid.*) or multi-layered perceptrons (*MLP*).

- MinkowskiEngine (Choy et al., 2019) (*ME*): the input point cloud is first voxelized. The output feature at each voxel is computed via a convolution between the features of nearby voxels and sparse 3D grid kernels.
- Adaptive Weights (Wang et al., 2018b) (*Adap.*): a stack of fully-connected layers processes the relative point coordinates of each input point. The output of this operation is multiplied by the relative point features in a point-wise fashion and summed to obtain the new feature value.
- Pseudo-grid kernel point convolution (Thomas et al., 2019) (*Grid.*) : a kernel of points with fixed positions is placed at each input point. Closeby neighboring features, weighted by their relative distance to the closest kernel point, are summed to produce a new feature value at each kernel point location. The contribution of each kernel location is multiplied by a weight to produce the output feature at each point location. The fact that the layout of the kernel points is regular, as if the points were lying on a grid explains the name of this approach.
- Multi-layered perceptrons (*MLP*): at each point, fully connected layers are applied to the concatenated relative point positions and neighboring features, followed by a max-pooling operation. This design is equivalent to Pointnet++ (Qi et al., 2017b).
- Point Transformer (Zhao et al., 2021) (*POTR*): a local attention mechanism, based on the relative position of neighboring points is employed to weight the contribution of nearby points. The key, query and value embeddings are computed from the relative positions via fully connected layers.

Note that the approaches *Adap.* and *MLP* are not defining a convolution operation explicitly, but learn a deep function to aggregate neighboring point representations. Except for *ME*, where neighboring voxel positions are used, all relative neighborhoods consist in a sphere neighborhood with fixed radius.

To choose the local point operator that is best suited to our task, we train and evaluate these different approaches on our dataset at increasing input patch radii, *with local 3D coordinates as sole input*. Our results are summarized in Fig. A.14.

We observe that all operators behave similarly and attain their best performance for a patch diameter of 7% of the scene radius (5 m), that is a patch radius of 0.35 m. Nevertheless, the best PR-AUC score obtained in this configuration is 0.78, and is attained by the (pseudo-)*Grid* local point operator. It empirically justifies our choice of the *Grid* local point operator for the ResUNet part of our ReVISOR framework.

Figs. A.15–A.17 provide a qualitative illustration of the different operators using this optimal parameter setup. We observe that the tested operators output results that are very close. The outlier points are well detected for medium-sized pipings, such as in Figs. A.15 and A.16, and cause more difficulty for extreme reflection cases, such as in Fig. A.17, where many outliers are missed no the right hand-side of

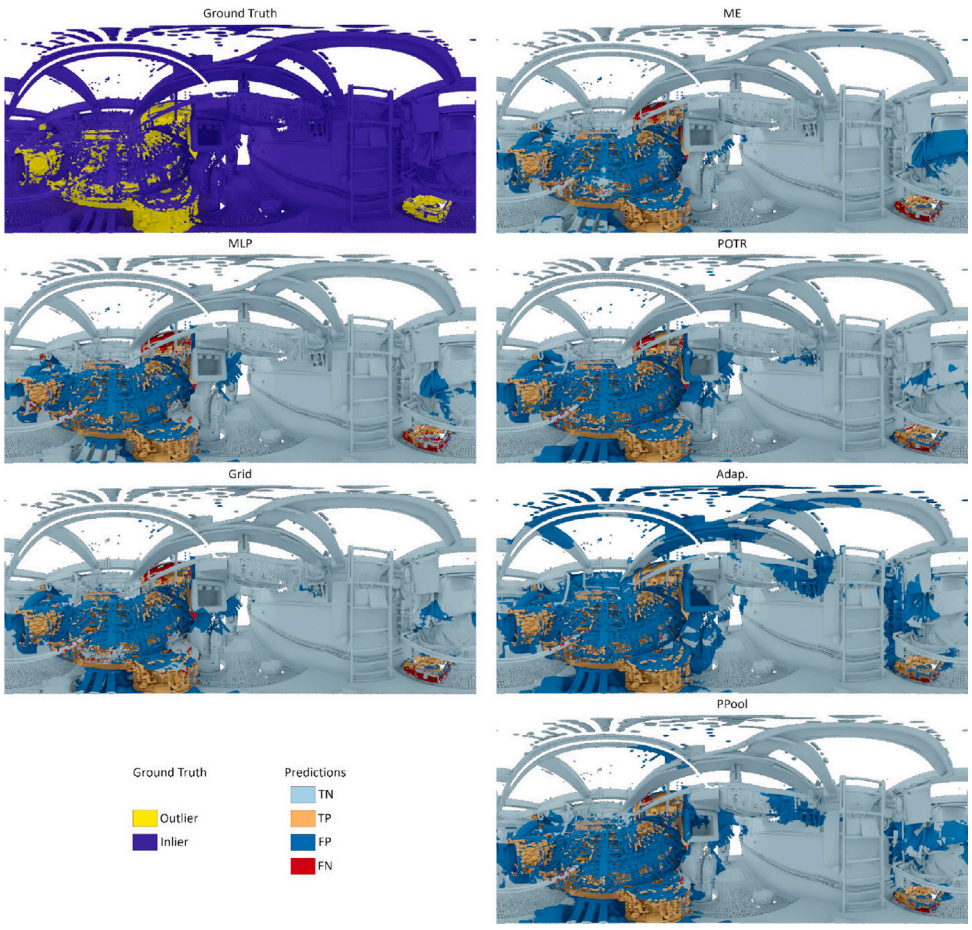

Fig. A.16. Qualitative evaluation of different local point operators, without features. The ground truth (*GT*) is provided in the top left cell, while each following cell depicts the true negative (*TN*), true positive (*TP*), false positive (*FP*) and false negative (*FN*) when using MinkowskiEngine (*ME*), PosPool (*PPool*), Adaptive weights (*Adap.*), Pseudo-grid (*Grid.*) or multi-layered perceptrons (*MLP*).

the image and at the bottom. We note nevertheless that the "POTR", "Grid" and "PPool" operators display fewer segmentation errors.

Appendix B. Illustration of our full dataset

To highlight the diversity of the scenes that we use to train and evaluate our architecture, we illustrate our complete dataset with our training scenes presented in Figs. B.18 and B.19 and our test scenes in Fig. B.20. Notice the variety of shapes and configurations present in both scenes and the differences between the training set and the evaluation set.

Appendix C. Comparison with mirror segmentation architectures

Table C.3 provides the quantitative comparison between PDNet (Mei et al., 2021) and our approach. It highlights that state-of-the-art RGBD-based mirror segmentation approaches are not successful on our dataset. We hypothesize that the networks leveraged by these approaches require more training examples than what we provide. Moreover, they target *planar* mirrors, a setting that is less general than ours: the reflective surfaces are irregular piping in our dataset.

PDNet is trained on our dataset, with each scan converted via a spherical projection to an RGB image and a depth map of size 3000×1500 pixels, for 50 epochs. Each input RGB image is a patch of size 416×416 from the full-resolution image, with the laser intensity signal in grayscale. Intensity represents indeed the feature closest to color in our setup. To ensure continuous depth and RGB maps, the points are rendered as spherical splats of radius $0.005 \times \sqrt{3}$, 0.005 m being the spatial sampling rate of the point cloud. The output prediction is projected back to the original point cloud to compare to the other approaches.

To conclude, we deduce from this experiment that image-based mirror detection architectures cannot be leveraged on our data, probably

Table C.3
Evaluation of various convolution operators (*Conv.*) and a recent mirror segmentation method (*Meth.*), namely PDNet (Mei et al., 2021). Since we use the laser intensity to obtain grayscale images as input for PDNet, we compare to the semantic segmentation approaches with the intensity as input feature. *PPool* stands for PosPool (Liu et al., 2020), *Adap.* for Adaptive weights (Wang et al., 2018b), *Grid* for pseudo-grid kernel-point convolution (Thomas et al., 2019) and *MLP* for Multi-Layered Perceptron. All quantities are expressed in percents.

Conv./Meth.	Features	Acc.	mIoU
Mirror Segmentation			
PDNet (Mei et al., 2021)	intensity	79.34	0.16
Semantic segmentation ResUNet			
PPool	intensity	85.67	51.74
Adap.	intensity	85.69	50.51
Grid	intensity	88.39	57.47
MLP	intensity	89.10	59.82

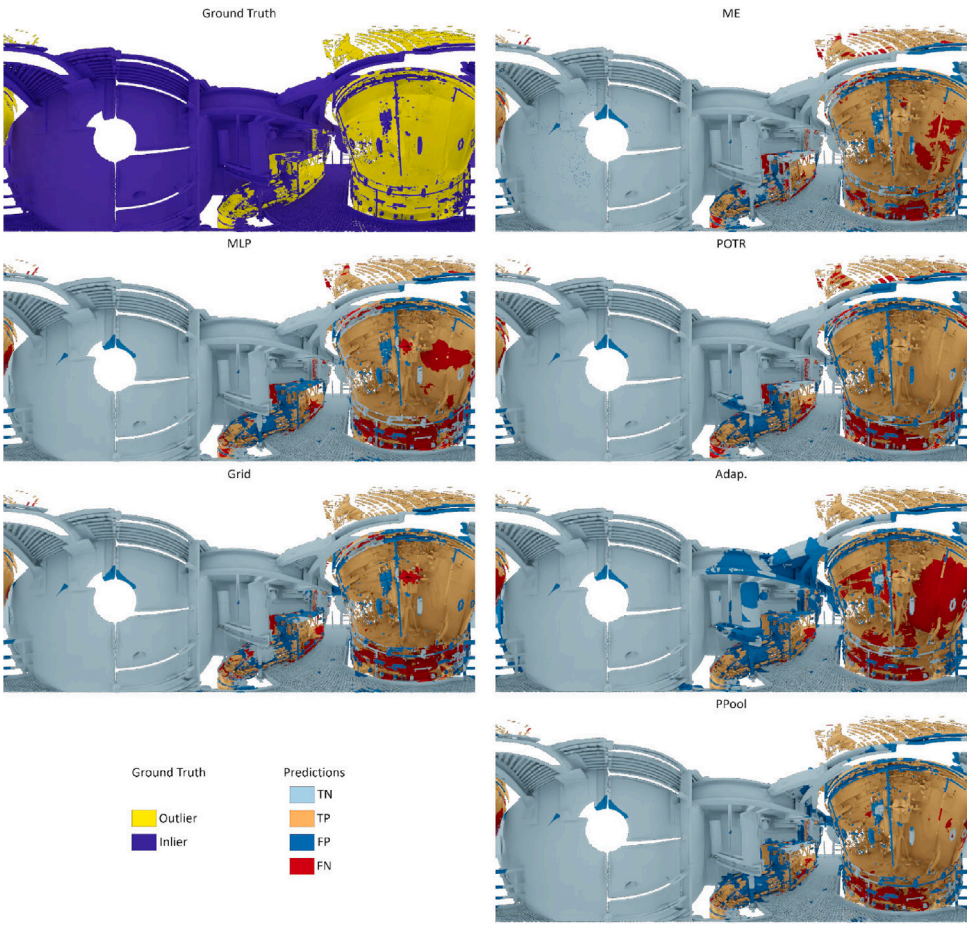

Fig. A.17. Qualitative evaluation of different local point operators, without features. The ground truth (*GT*) is provided in the top left cell, while each following cell depicts the true negative (*TN*), true positive (*TP*), false positive (*FP*) and false negative (*FN*) when using MinkowskiEngine (*ME*), PosPool (*PPool*), Adaptive weights (*Adap.*), Pseudo-grid (*Grid.*) or multi-layered perceptrons (*MLP*).

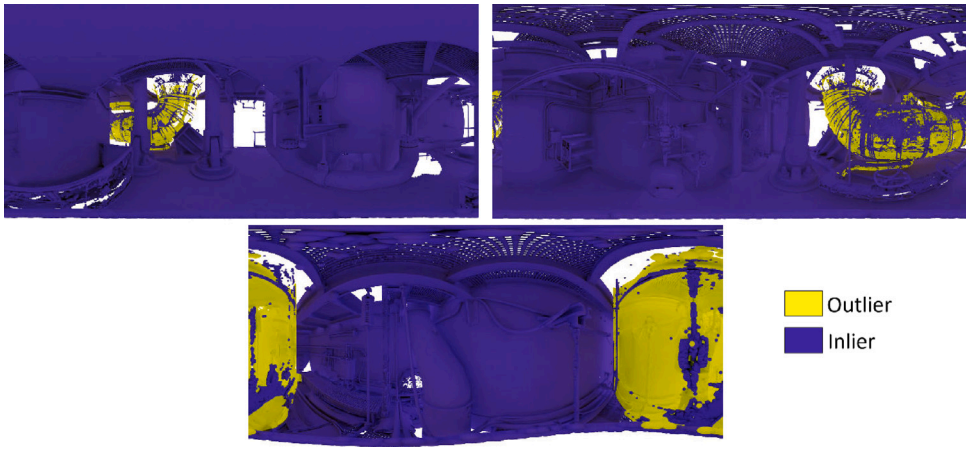

Fig. B.18. Illustration of our training scenes, with color-coded ground truth.

mostly because of the unavailability of a reliable RGB input channel (we only have access to an intensity signal) and the little amount of data (the mirror segmentation dataset proposed in Mei et al. (2021) consists in hundreds of scenes whereas we only have 13 scenes).

Appendix D. Visibility computation and parameterization

In this appendix, we provide additional details on the computation of the visibility signal that we use in our method.

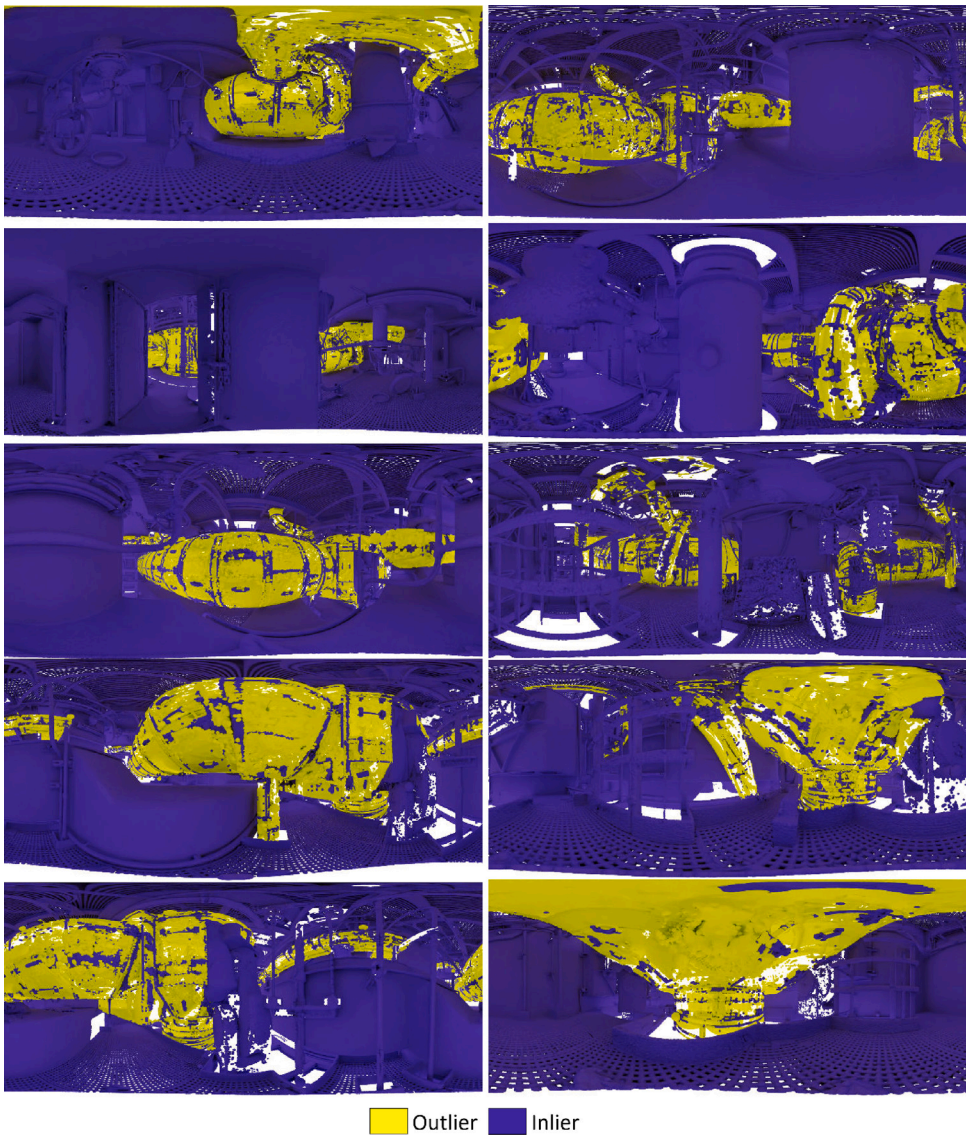

Fig. B.19. Illustration of our training scenes, with color-coded ground truth.

D.1. Computation

Given a point cloud $\mathcal{P} = \{p_i\}_{i\in[1,N]}$ with N points and a viewpoint C (the acquisition device location in our case), the algorithm developed in Katz et al. (2007) assigns to each point a label: 0 if the point is visible and 1 if the point is invisible from C. The core of this approach consists in the *hidden point removal* (HPR) operator, that processes the point cloud in two steps (see Fig. D.21).

1. **Spherical inversion.** Given $\mathcal{P}$ and a sphere that contains all the points of $\mathcal{P}$, the spherical inversion consists in reflecting all $p_i \in \mathcal{P}$ with respect to the sphere. The reflection of p_i is denoted as $\hat{p}_i$ and is computed as follows:

$$\mathbf{\hat{p}_i} = \mathbf{p_i} + 2\left(R - \|\mathbf{p_i}\|\right) \cdot \frac{\mathbf{p_i}}{\|\mathbf{p_i}\|}.$$

2. **Convex hull regression.** Given the set of all reflected points $\hat{\mathcal{P}} = \{\mathbf{\hat{p}_i}\}_{i\in[1,N]}$, compute the convex hull of $\hat{\mathcal{P}} \cup \{C\}$.

D.1.1. Parameterization

The computation of the visibility feature relies on a hyper-parameter, namely the radius of the inversion sphere R. To observe significant changes on the visibility, an exponential change of R is required. We thus reparameterize R with γ as such: $R := 10^{\gamma}$.

Inspired by the approach of the authors of the visibility computation method (Katz et al., 2007), we determine γ as a trade-off between precision and recall when using the raw feature value as the outlier segmentation on the training and testing sets of our dataset. Fig. D.22 (*purple lines*) shows the output of this computation. Intuitively, since $R := 10^{\gamma}$, where R is the radius of the inversion sphere, R has the same variations as γ. Moreover, as displayed on Fig. D.21, small values for R, i.e. small values for γ, yield a larger number of invisible points. Conversely, large values for R/γ correspond to a small number of invisible points. Now, recall that invisible points are likely to be reflected outliers, occluded by inlier points and that we are classifying invisible points as outliers in this experiment. The decay of the recall corresponds to a decay in the number of visible points. Extreme values for γ are not informative since they correspond to either a "all points are outliers" or a "all points are inliers" segmentation. The optimal parameter has to be chosen "in the middle" of these extreme values. Since the intersection of the precision and recall curves occurs for $\gamma = 3.2$, we select this value as our "middle value" and use it to compute the visibility feature in the remaining of this paper.

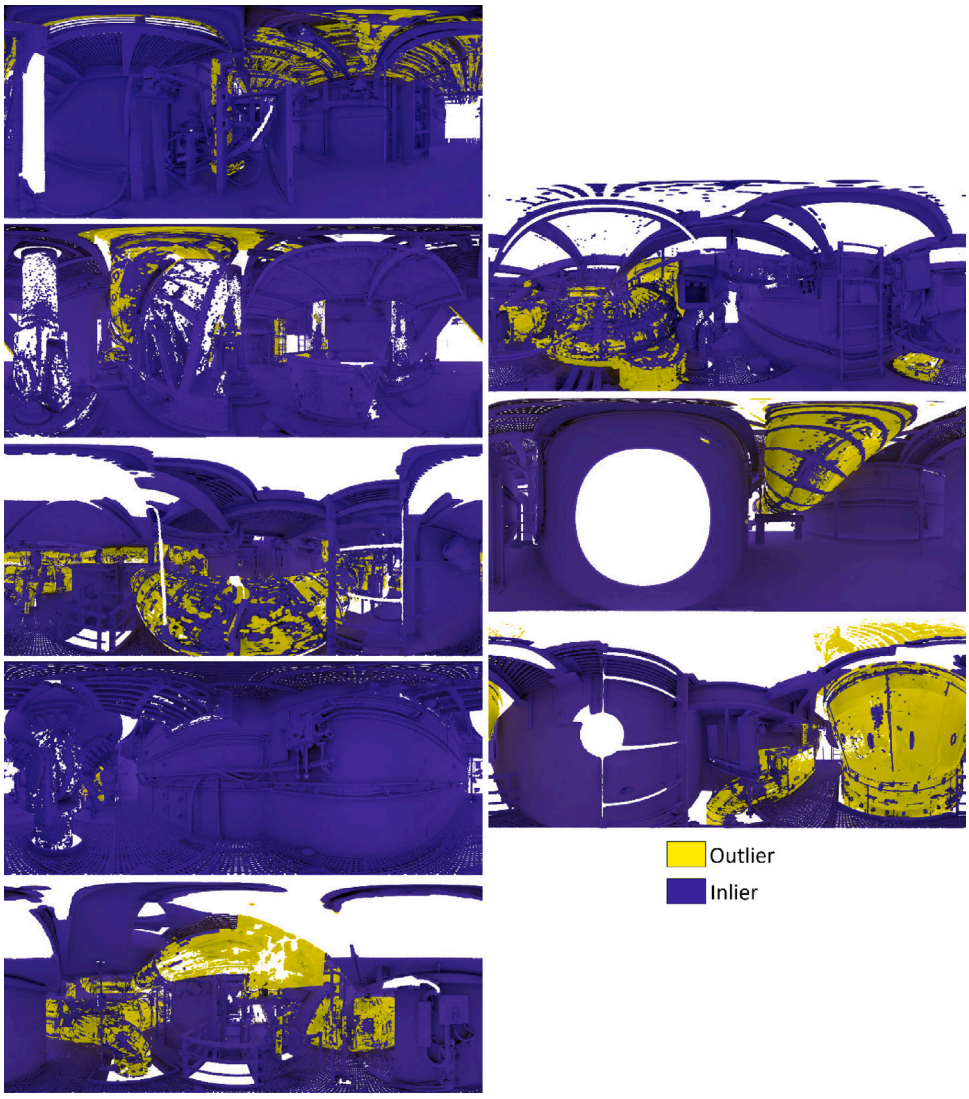

Fig. B.20. Illustration of our evaluation scenes, with color-coded ground truth.

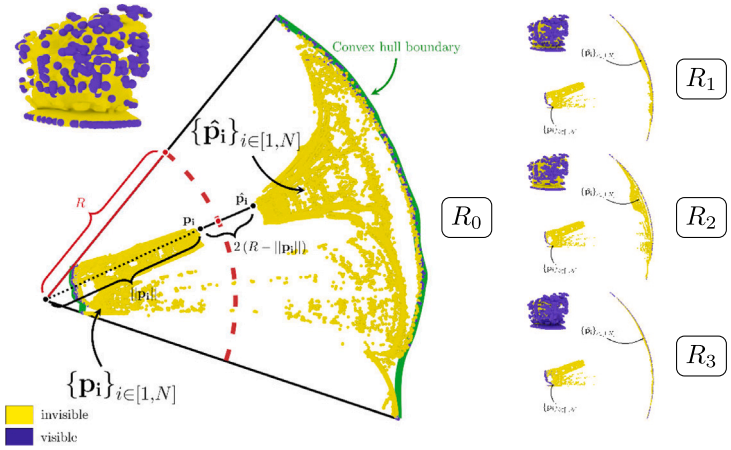

Fig. D.21. A schematic depiction of the spherical inversion with increasing inversion radii $\{R_i\}_{i\in[0,3]}$. The inset figure shows the appearance of the point cloud from the inversion center and the main figure shows the same scene projected on the (x, z) plane. We show four different inversion radii to illustrate how the visible areas vary when the inversion radius varies: greater radii increase the number of visible points because more inverted points get "squeezed" on the convex hull of visible points. At very large radii, the convex hull is a portion of a sphere, with all points lying on it, i.e. all points are marked as visible. Conversely, at very small radii, the inverted points are "dragged" towards the acquisition center and only a few number of points lie on the convex hull.

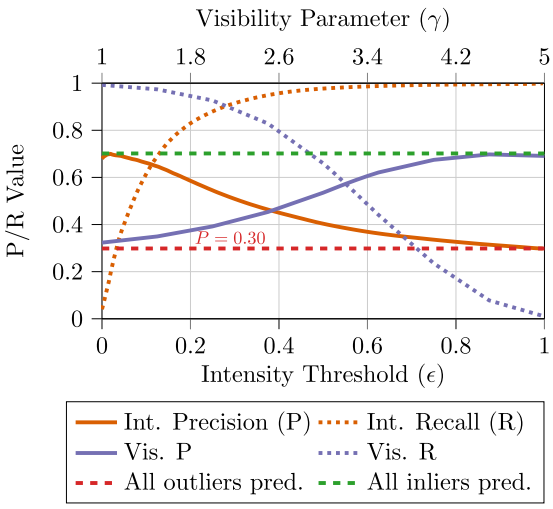

Fig. D.22. Precision/Recall curves using the thresholded intensity or visibility feature solely, computed on our training dataset. Both features vary in opposition, which reinforces their joint use.

Appendix E. Visibility and intensity features with various local point operators

In order to analyze the different local point operators that behave best at the optimal receptive field size (40% of the scene diameter), we feed the intensity and/or the visibility signal in addition to the raw point coordinate to the network. Fig. E.23 presents the resulting evaluation in terms of precision/recall and Fig. E.24 in terms of mean accuracy and mean intersection over union.

The main insight of our study is the similar performance across the different local point operators: all roughly perform identically, with a slight advantage for the (pseudo-)*Grid* local point. Employing intensity or visibility alone performs better than using raw 3D coordinates. Leveraging both features simultaneously leads to the best results for all operators but *ME* and *POTR*. For these two last local point operator, using intensity or visibility alone or both leads to similar results.

Appendix F. Implementation details and experimental setup

In this section, we provide more details on the implementation and the experimental setup of the different networks that we use.

For the *Adap.*, *Grid*, *MLP*, *Pool* and *POTR* local point operators, a binary mask indicates to the network which points are actual points and which are padding points, added to the input to constitute a batch with a fixed number of input points (see below). The input feature dimension is fixed to 3. By default, the input features are set to a tensor filled with ones. When an actual feature channel is used, a column of the input feature tensor is filled with the feature values. Since we employ at most two features simultaneously, the dimensionality of the architecture does not change. In the case where no input features are used, we use the local 3D coordinates as features.

All our ResUNet models are trained using the (binary) cross-entropy loss on 70 epochs, with 2000 input patches per epoch. The starting learning rate is set at 0.01, with a decay of 0.92 every 10 epoch. A warm-up of 10 epochs is used with *no features* to initiate the training procedure. We use the stochastic gradient descent optimizer, with 0.01 weight decay to optimize the weights of the network.

For the evaluation, we consider a subsampling of $\frac{R_{patch}}{4}$, where R_{patch} designates the radius of the patch. During the training, we monitor the mean validation loss at each epoch and select the weights at the epoch where the validation loss is the lowest. The validation set consists in 320 randomly selected patches, that come from 4 different scenes.

To train and evaluate all our models but the models with a *POTR* local operator or *ME* convolutions, we use a computer with 187 GB of RAM, a processor with 4 cores at 3.6 GHz and a single GPU with 16 GB of memory, running CUDA version 10.1. For the models using the *POTR* local operator or *ME*, we employ a computer with 376 GB of RAM, a processor with 4 cores at 2.4 GHz and a single GPU with 32 GB of memory, running CUDA version 11.0.

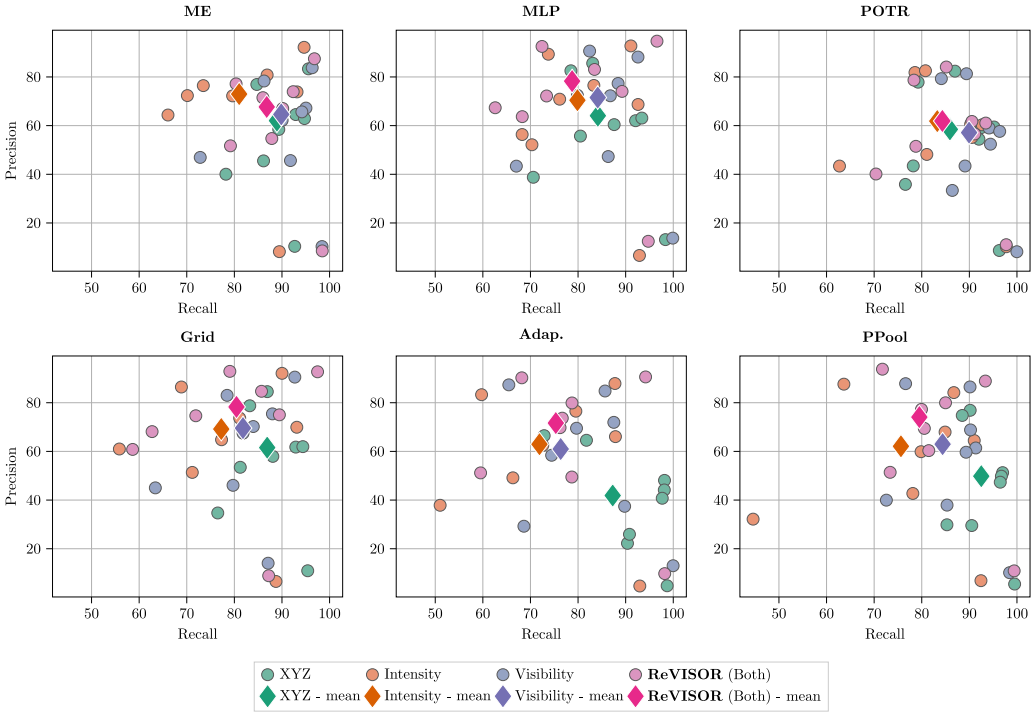

Fig. E.23. Benchmark of different input feature combinations in terms of precision and recall, for all tested local point operators. Each *circle* marker corresponds to a test scene and each *diamond* marker to the average over all test scenes.

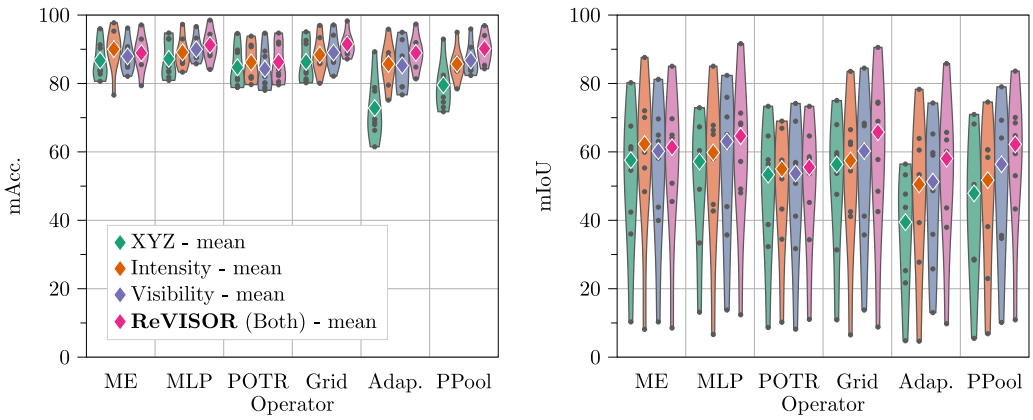

Fig. E.24. Benchmark of different input feature combination in terms of mean accuracy (*mAcc.*) and mean intersection over union (*mIoU*), for all tested local point operators.

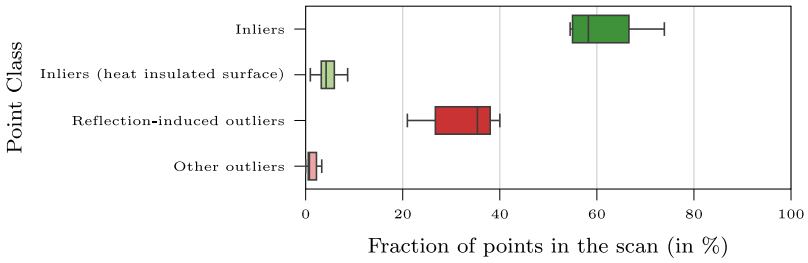

Fig. G.25. Distribution of the proportion of points belonging to inliers and outliers. We distinguish between: (i) inlier points that lie on the surface of a heat insulated piping (*light green*) and other inliers (*dark green*), and (ii) outlier points that are caused by reflections (*dark red*) on heat-insulated piping and other outliers (*light red*). The vast majority of points belong to inliers that are not related to heat-insulated surfaces and to outliers that are induced by reflections of the laser beam on heat-insulated surfaces. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Appendix G. Supplementary material on our dataset

G.1. Distribution of points depending on their class

To supplement the presentation of our dataset (Section 4.1), we show in Fig. G.25 the percentage of points in each scan that fall into each classification of points (inliers and outliers) and visualize the distribution across all scans of the dataset. We differentiate between reflection-induced outliers and other types of outliers and also distinguish inliers associated with a heat insulator from other types of inliers. The two dominant classes are inliers not related to heat insulated piping (all scans have more than 50% of their points falling in this category) and reflection-induced outliers (more than 20% of points in each scan). It justifies our choice of using only these two classes and discarding the other two, that are both considered as "inliers" (i.e. points that are not reflection-induced outliers) in our experiments.

G.2. Point cloud acquisition quality

In order to illustrate the acquisition quality of our scans, we performed the following experiment on a scan from our training dataset (top left scan in Fig. B.18, see also Fig. G.27 (right) for a photography of the heat insulator of this room). We reconstructed a piece of the heat insulated piping and four cylindrical shapes present in the scene, as illustrated in Fig. G.26. Despite their geometrical similarity (all regions are cylindrical) and their shared weak intensity (the selected regions have less than 20% of the maximal intensity recorded by the acquisition device), we observe that the points acquired in the heat-insulated section are more distant from the actual surface to which they belong compared to points in the other selected regions. Fig. G.27 (left) illustrates quantitatively this qualitative assessment with the histogram of the distances of each point to the surface to which they belong in logarithmic scale.

G.3. Reflection-induced outliers and intensity feature correlation

The intensity signal represents an important cue for our reflection-induced outlier detection model. However, intensity alone is insufficient to accurately segment the acquisitions. A weak intensity signal can indeed result from two main causes: (i) the laser beam hit a reflected material and got scattered in the scene ; (ii) the laser beam hit a dark material and the laser beam got absorbed. Moreover, the laser beam does not necessarily loose a significant portion of its energy when hitting a reflective surface (see for instance the intensity map in Fig. 11, middle section of the heat insulated piping in the foreground).

To further illustrate this limitation, we conduct the following experiment on the acquisitions of our training dataset. We threshold the intensity signal between 0 and a threshold value. The resulting binary classification is then used as the reflection-induced outlier prediction. The experiment is repeated for all scans individually and for 25 threshold values, evenly placed between 0 and 1 (both excluded). We then compute the accuracy of the resulting segmentations for all scans and all threshold values. Fig. G.28 shows the value of the best threshold for each scan. The optimal threshold value varies randomly between scans and therefore cannot be defined *a priori*.

Moreover, we consider in Fig. G.29 the Pearson correlation between the segmentation obtained for each scan at each threshold value. It highlights that all threshold values lead to a weak to moderate correlation with the reflection-induced outliers, the weak correlation being dominant.

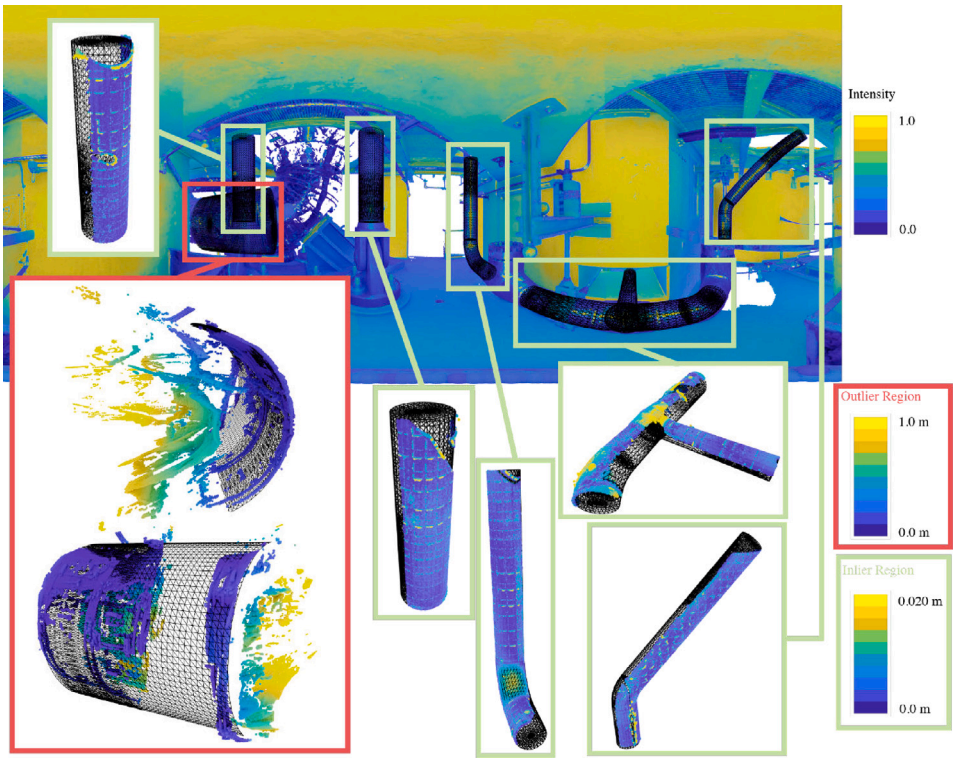

Fig. G.26. Illustration of the quality of our laser acquisitions. The top image corresponds to the intensity field as viewed from the acquisition device of the top left scan of Fig. B.18. The reconstructed shapes of piping are superimposed in black. We consider a horizontal section for the heat-insulated piping (*red box*) and four cylindrical sections across the scene (*green boxes*). All selected areas share a weak intensity signal. For all areas, we display a magnified view of the reconstructed area from a different perspective and visualize the absolute distance to the reconstructed mesh with a gradient of colors. For the outlier region (*red box*), the distances ranges from 0.0 m to 1.0 m. For the inlier regions (*green boxes*), the distances are displayed between 0.0 m and 0.020 m. The distribution of distances for both regions is illustrated in Fig. G.27 (left). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

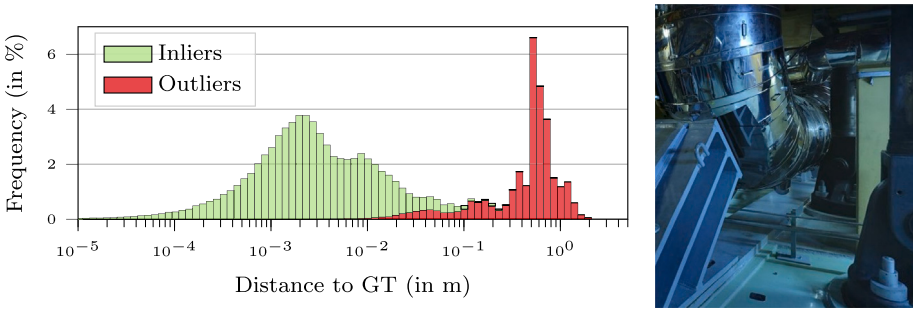

Fig. G.27. Left: Histograms of the distance to reconstructed geometry for points associated to a heat insulator section (i.e. mostly outliers, in *red*) and to piping that is not heat insulated (i.e. only inliers, in *green*). When the bars of both histograms overlap, they are stacked. Note how the vast majority of points belonging to "regular" piping lies close to the reconstructed geometry (distance to mesh below 0.025 m), while points belonging to a heat insulated piping are further away (distance to mesh greater than 0.25 m). Right: Photography of the heat insulator in Fig. G.26. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

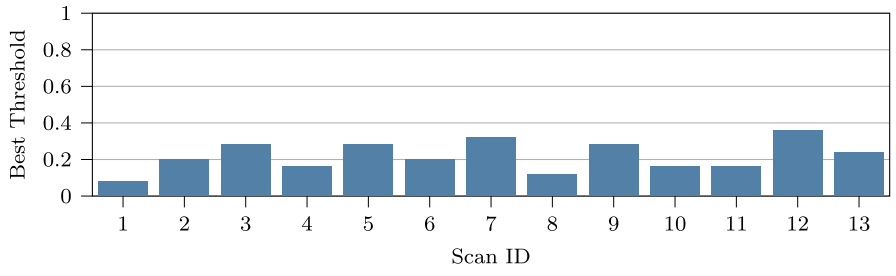

Fig. G.28. Best threshold value of the intensity signal for each scan in our dataset, when using the intensity signal to segment reflection-induced outliers. The optimal threshold value is random across scans.

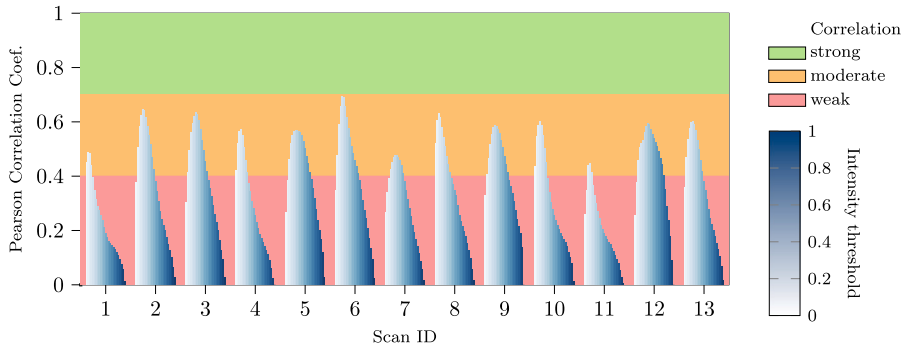

Fig. G.29. Pearson correlation coefficient between the thresholded intensity and the ground truth reflection-induced outliers for each scan of our training set. Weak intensity values are moderately correlated with the reflection-induced outliers. Most threshold values lead to a weak correlation with the reflection-induced outliers.

References

Alnaggar, Y.A., Afifi, M., Amer, K., ElHelw, M., 2021. Multi projection fusion for real-time semantic segmentation of 3D lidar point clouds. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1800–1809.

Armeni, I., Sax, S., Zamir, A.R., Savarese, S., 2017. Joint 2D-3D-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105.

Barnett, V., T., L., 1995. Outliers in Statistical Data. Wiley Online Library.

Bazazian, D., Nahata, D., 2020. DCG-Net: Dynamic capsule graph convolutional network for point clouds. IEEE Access 8, 188056–188067.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J., 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9297–9307.

Boulch, A., 2020. ConvPoint: Continuous convolutions for point cloud processing. Comput. Graph. 88, 24–34.

Cazals, F., Pouget, M., 2005. Estimating differential quantities using polynomial fitting of osculating jets. Comput. Aided Geom. Design 22 (2), 121–146.

Cazals, F., Pouget, M., 2008. Jet fitting 3: A generic C++ package for estimating the differential properties on sampled surfaces via polynomial fitting. ACM Trans. Math. Software 35 (3).

Chamseddine, M., Rambach, J., Stricker, D., Wasenmuller, O., 2021. Ghost target detection in 3D radar data using point cloud based deep neural network. In: 2020 25th International Conference on Pattern Recognition. ICPR, IEEE, pp. 10398–10403.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2017. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. 40 (4), 834–848.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 801–818.

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

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

Davis, J., Goadrich, M., 2006. The relationship between Precision-Recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 233–240.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Duan, Y., Zheng, Y., Lu, J., Zhou, J., Tian, Q., 2019. Structural relational reasoning of point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 949–958.

Feng, M., Gilani, S.Z., Wang, Y., Zhang, L., Mian, A., 2020. Relation graph network for 3D object detection in point clouds. IEEE Trans. Image Process. 30, 92–107.

Fleishman, S., Cohen-Or, D., Silva, C.T., 2005. Robust moving least-squares fitting with sharp features. ACM Trans. Graph. 24 (3), 544–552.

Gao, R., Li, M., Yang, S.-J., Cho, K., 2022. Reflective noise filtering of large-scale point cloud using transformer. Remote Sens. 14 (3), 577.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R.R., Hu, S.-M., 2021. PCT: Point cloud transformer. Comput. Vis. Media 7 (2), 187–199.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M., 2017. Semantic3D. net: A new large-scale point cloud classification benchmark. arXiv preprint arXiv:1704.03847.

Hermosilla, P., Ritschel, T., Ropinski, T., 2019. Total denoising: Unsupervised learning of 3D point cloud cleaning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 52–60.

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

Huang, C., Li, R., Li, X., Fu, C.-W., 2020. Non-local part-aware point cloud denoising. arXiv preprint arXiv:2003.06631.

Hullo, J.-F., Thibault, G., Boucheny, C., Dory, F., Mas, A., 2015. Multi-sensor as-built models of complex industrial architectures. Remote Sens. 7 (12), 16339–16362.

Irfan, M.A., Magli, E., 2021. 3D point cloud denoising using a joint geometry and color k-NN graph. In: 2020 28th European Signal Processing Conference. EUSIPCO, IEEE, pp. 585–589.

Jaritz, M., Gu, J., Su, H., 2019. Multi-view PointNet for 3D scene understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.

Katz, S., Tal, A., Basri, R., 2007. Direct visibility of point sets. In: ACM SIGGRAPH 2007 Papers. pp. 24–es.

Komarichev, A., Zhong, Z., Hua, J., 2019. A-CNN: Annularly convolutional neural networks on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7421–7430.

Kundu, A., Yin, X., Fathi, A., Ross, D., Brewington, B., Funkhouser, T., Pantofaru, C., 2020. Virtual multi-view fusion for 3D semantic segmentation. In: European Conference on Computer Vision. Springer, pp. 518–535.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J., 2022. Stratified transformer for 3D point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8500–8509.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4558–4567.

Liu, Z., Hu, H., Cao, Y., Zhang, Z., Tong, X., 2020. A closer look at local aggregation operators in point cloud analysis. In: European Conference on Computer Vision. Springer, pp. 326–342.

Luo, S., Hu, W., 2020. Differentiable manifold reconstruction for point cloud denoising. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1330–1338.

Luo, S., Hu, W., 2021. Score-based point cloud denoising. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4583–4592.

Maimon, O., Rokach, L., 2005. Data Mining and Knowledge Discovery Handbook. Springer.

Mazur, K., Lempitsky, V., 2021. Cloud transformers: A universal approach to point cloud processing tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10715–10724.

Mei, H., Dong, B., Dong, W., Peers, P., Yang, X., Zhang, Q., Wei, X., 2021. Depth-aware mirror segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3044–3053.

Milioto, A., Vizzo, I., Behley, J., Stachniss, C., 2019. RangeNet++: Fast and accurate lidar semantic segmentation. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 4213–4220.

Pistilli, F., Fracastoro, G., Valsesia, D., Magli, E., 2020. Learning graph-convolutional representations for point cloud denoising. In: European Conference on Computer Vision. Springer, pp. 103–118.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413.

Rakotosaona, M.-J., Guerrero, P., Aigerman, N., Mitra, N.J., Ovsjanikov, M., 2021. Learning delaunay surface elements for mesh reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22–31.

Rakotosaona, M.-J., La Barbera, V., Guerrero, P., Mitra, N.J., Ovsjanikov, M., 2020. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In: Computer Graphics Forum. Vol. 39. Wiley Online Library, pp. 185–203.

Robert, D., Vallet, B., Landrieu, L., 2022. Learning multi-view aggregation in the wild for large-scale 3D semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5575–5584.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Rousseeuw, P.J., Hubert, M., 2011. Robust statistics for outlier detection. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 1 (1), 73–79.

Song, S., Lichtenberg, S.P., Xiao, J., 2015. SUN RGB-D: A RGB-D scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 567–576.

Tan, J., Lin, W., Chang, A.X., Savva, M., 2021. Mirror3D: Depth refinement for mirror surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15990–15999.

Thomas, H., Qi, C.R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. KPConv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6411–6420.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wang, L., Giebenhain, S., Anklam, C., Goldluecke, B., 2021. Radar ghost target detection via multimodal transformers. IEEE Robot. Autom. Lett. 6 (4), 7758–7765.

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

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., Tong, X., 2017. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. ACM Trans. Graph. 36 (4), 1–11.

Wang, C., Samari, B., Siddiqi, K., 2018a. Local spectral graph convolution for point set feature learning. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 52–66.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al., 2020. Deep high-resolution representation learning for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 43 (10), 3349–3364.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019b. Dynamic graph CNN for learning on point clouds. ACM Trans. Graph..

Wang, S., Suo, S., Ma, W.-C., Pokrovsky, A., Urtasun, R., 2018b. Deep parametric continuous convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2589–2597.

Wu, W., Qi, Z., Fuxin, L., 2019. Pointconv: Deep convolutional networks on 3D point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9621–9630.

Xu, M., Ding, R., Zhao, H., Qi, X., 2021. PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds. arXiv preprint arXiv:2103.14635.

Xu, Q., Sun, X., Wu, C.-Y., Wang, P., Neumann, U., 2020. Grid-GCN for fast and scalable point cloud learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5661–5670.

Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S., 2020. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5589–5598.

Yang, J., Zhang, Q., Ni, B., Li, L., Liu, J., Zhou, M., Tian, Q., 2019. Modeling point clouds with self-attention and gumbel subset sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3323–3332.

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J., 2022. Point-BERT: Pre-training 3D point cloud transformers with masked point modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19313–19322.

Yun, J.-S., Sim, J.-Y., 2018. Reflection removal for large-scale 3D point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4597–4605.

Yun, J.-S., Sim, J.-Y., 2019. Virtual point removal for large-scale 3D point clouds with multiple glass planes. IEEE Trans. Pattern Anal. Mach. Intell..

Zhang, F., Fang, J., Wah, B., Torr, P., 2020a. Deep fusionnet for point cloud semantic segmentation. In: European Conference on Computer Vision. Springer, pp. 644–663.

Zhang, Z., Hua, B.-S., Yeung, S.-K., 2019. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1607–1616.

Zhang, D., Lu, X., Qin, H., He, Y., 2020b. Pointfilter: Point cloud filtering via encoder-decoder modeling. IEEE Trans. Vis. Comput. Graphics 27 (3), 2015–2027.

Zhao, H., Jiang, L., Fu, C.-W., Jia, J., 2019. Pointweb: Enhancing local neighborhood features for point cloud processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5565–5573.

Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V., 2021. Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16259–16268.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al., 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6881–6890.

Zhou, Q.-Y., Park, J., Koltun, V., 2018. Open3D: A modern library for 3D data processing. arXiv:1801.09847.

Zhou, Y., Tuzel, O., 2018. VoxelNet: End-to-end learning for point cloud based 3D object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499.