Learning-Based Reflection-Aware Virtual Point Removal for Large-Scale 3D Point Clouds

Oggyu Lee, Kyungdon Joo, and Jae-Young Sim, *Member, IEEE*

***Abstract*—3D point clouds are widely used for robot perception and navigation. LiDAR sensors can provide large scale 3D point clouds (LS3DPC) with a certain level of accuracy in common environment. However, they often generate virtual points as reflection artifacts associated with reflective surfaces like glass planes, which may degrade the performance of various robot applications. In this letter, we propose a novel learning-based framework to remove such virtual points from LS3DPCs. We first project 3D point clouds onto 2D image domain to investigate the distribution of the LiDAR's echo pulses, which is then used as an input to the glass probability estimation network. Moreover, the 3D feature similarity estimation network exploits the deep features to compare the symmetry and geometric similarity between real and virtual points with respect to the estimated glass plane. We provide a LS3DPC dataset with synthetically generated reflection artifacts to train the proposed network. Experimental results show that the proposed method achieves the better performance qualitatively and quantitatively compared with the existing state-of-the-art methods of 3D reflection removal.**

***Index Terms*—Deep learning for visual perception, computer vision for automation, data sets for robotic vision.**

I. Introduction

LIDAR sensors basically emit light pulses to environment, and then by measuring the response time, they can accurately estimate the distance to the surrounding scene. By virtue of their accuracy, LiDAR sensors, as a way of acquiring 3D point clouds, have become a popular and essential choice in robotics. Concretely, intelligent agents (e.g., robots and autonomous vehicles) highly depend on the acquired 3D point clouds by LiDAR sensors for various 3D perception tasks, such as mapping [1], [2], [3] and 3D object detection [4], [5]. In most cases, LiDAR sensors guarantee large-scale 3D point clouds (LS3DPCs) with cm-level accuracy, but they could generate undesirable point clouds due to reflective environment, such as buildings with glass, that may significantly influence downstream robot tasks. In this letter, we call these undesirable and physically inexistent point clouds *virtual points* and aim to remove such virtual points for given LS3DPCs.

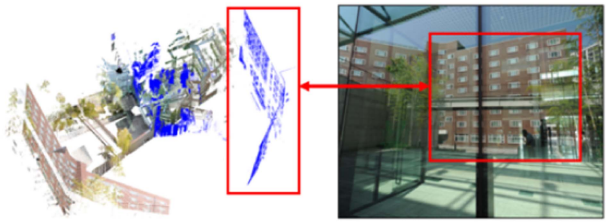


Fig. 1. Reflection artifacts in LS3DPC. *Left:* Captured point clouds including virtual points (blue). *Right:* Captured reference image corresponding to reflection artifacts.

The virtual points occur as the reflection of light pulses. Specifically, a light pulse emitted from LiDAR sensor is reflected on a reflective or specular surface, such as mirrors or glasses, and travels toward another direction to collide with another object, resulting in a virtual point. Therefore, the virtual points inherently appear in a symmetric form against the corresponding actual points with respect to the reflective surface (see Fig. 1).

Several methods have been proposed to address the problem of virtual points via reflection removal, where they exploit the inherent properties of symmetry and geometric similarity between the virtual points and their corresponding real points. In these methods [6], [7], [8], comparing the geometric shapes of point clouds and estimating the glass regions are essential for virtual point detection. They extracted hand-crafted features [9] to compute the geometric similarity between point clouds, and used the multi-echo property of LiDAR to estimate the glass regions. However, the hand-crafted features do not consider the density difference between the virtual point cloud and the real point cloud in reflective environment. Moreover, the previous methods often fail to find accurate glass regions if the points are not sampled on the glass and/or if a real object is located close behind the glass surface.

In this letter, we propose a new systematic framework that removes the virtual points by taking advantage of inherent geometric properties of LS3DPC. In particular, we focus on devising a simple yet effective learning-based method where the network pipelines for both 2D and 3D representations are employed, respectively. The network for 2D image domain takes an input called *count map*, the projection image of a 3D point cloud, and estimates the glass region distingushed from other ones, such as trees and far-away building windows, to assign

Manuscript received 17 June 2023; accepted 13 October 2023. Date of publication 1 November 2023; date of current version 13 November 2023. This letter was recommended for publication by Associate Editor L. Gan and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the National Research Foundation of Korea within the Ministry of Science and ICT (MSIT) under Grant 2020R1A2B5B01002725 and in part by Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korea Government (MSIT); Artificial Intelligence Innovation Hub under Grant 2021-0-02068 and Artificial Intelligence Graduate School Program (UNIST) under Grant 2020-0-01336. *(Corresponding author: Jae-Young Sim.)*

Oggyu Lee is with the Department of Electrical Engineering, UNIST, Ulsan 44919, Republic of Korea, and also with the Hyundai-Steel Company, Dangjin 31719, Republic of Korea (e-mail: oglee@unist.ac.kr).

Kyungdon Joo and Jae-Young Sim are with the Graduate School of Artificial Intelligence, UNIST, Ulsan 44919, Republic of Korea (e-mail: kyungdon@unist.ac.kr; jysim@unist.ac.kr).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3329365, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3329365

the glass probability values to point clouds. On the other hand, the network for 3D domain extracts deep features based on a voxel-based structure to compare the symmetry and geometric similarity between point clouds at once even with different densities and different orientations. Finally, the resulting glass probability and the similarity score are combined together to determine the virtual points.

The contributions of this letter are as follows:

1) We first proposed a simple yet effective learning-based framework that removes virtual points in LS3DPCs, which demonstrated the superior performance both qualitatively and quantitatively compared with the existing methods.
2) We made use of both the 3D point cloud as well as the projected 2D image to estimate the glass region and compute the symmetry and geometric similarity between real and virtual points.
3) We collected 10 ordinary LS3DPC models and applied various augmentation schemes to make realistic synthetic scenes with virtual points for training. Furthermore, we manually labeled 6 scenes from UNIST LS3DPC dataset [6] for quantitative experiments.

II. Related Work

A. Representations of Point Clouds

Point clouds have a set of unordered and unstructured 3D points that straightforwardly convey 3D information. We can use raw point clouds themselves [10], [11], [12], [13], as well as transform point clouds into various domains, such as projection images with the front view [14], [15] or bird-eye view [5], [16], voxels [17], [18], [19], and connected graphs [20], [21]. Each representation has its own pros and cons. We briefly discuss the representations in the following.

Point-based methods are widely used in various fields such as object detection [13] and segmentation [12]. They have been mainly developed based on PointNet [10] and its variants. Charles et al. [10] directly used the raw point cloud as input and processed the points with multi-layer perceptrons (MLPs). Charles et al. [11] improved [10] by adding local grouping which allows the network to look at neighboring points utilizing hierarchical structures.

Projection-based methods [5], [14], [15], [16] have the advantage of converting point clouds to other structured data, but they may place the points with far distance from each other to be close in the projection domain. For example, we can acquire a depth map by projecting the 3D point clouds toward the center of scanner. Then the depth map can be used with the corresponding color images by sensor fusion for various tasks such as detection [14] and noise removal [15]. The bird-eye view projection methods [5], [16], [22] convert the point clouds into the view seen from the top, that are widely used in the detection task due to the benefit of localizing objects.

Voxel-based methods [4], [17], [18], [19], [23], [24] contain 3D information into well structured regular grids. However, they suffer from the limitations of high computation and large memory space with increasing the voxel resolution. To alleviate the computational complexity, sparse convolution-based methods [4], [23] have been proposed that employ 3D convolution on voxel representation. The voxel representation has been actively used in the fields of object detection [4], [17], [18], [19], segmentation [23], and registration [24].

Graph-based methods [20], [21] connect neighboring points and use their connection along with the position in training. They take advantage of using neighboring points but the neighbor computation is more expensive compared to handling raw point clouds itself. Yang et al. [20] applied the graph convolution to train the autoencoder for classification purposes. Wang et al. [21] applied the dynamic graph convolution, which dynamically updates its graph information to connect neighboring points in the features space.

In this work, we used the voxel representation to extract 3D features from LS3DPCs, that can handle empty 3D spaces and different density of point clouds to reliably compute the symmetry and geometric similarity.

B. 3D Reflection Removal

Depending on the principle of LiDAR, 3D reflection removal is divided into two categories: single-echo and multi-echo based approaches. In the single-echo based reflection removal [25], [26], LiDAR sensors capture a single echo pulse only for each emitted light. The positions of virtual points generated by reflective objects vary according to the location of LiDAR sensors, and therefore we additionally require the registration information between the captured LS3DPC models in multiple positions. Gao et al. [25] projected the LS3DPC models toward the origin of scanner to obtain the intensity and range images. By applying the sliding windows to the projected images, they estimated the change of pixel distribution and detected the reflective regions. The virtual points are removed by comparing the reflective regions of the captured scenes at various locations. Gao et al. [26] also improved the previous method by using the transformers to estimate the reflective regions. However, the single-echo based methods require multiple captures as well as the additional intensity data of point clouds. Moreover, these methods can only consider the existence of points not exploiting the geometric properties.

In contrary to the single-echo based methods, the multi-echo based reflection removal [6], [7], [8] can assess multiple echo signals from an emitted light pulse of LiDAR, which are then used to estimate the glass regions without additional information of the range and intensity. Therefore, the multi-echo based methods analyze the symmetry and geometric similarity between the points within a single 3D point cloud model while not requiring multiple scans and their registration. Yun and Sim [7] were the first to propose a solution for the multi-echo based 3D reflection removal. They first estimated the glass regions by using the distribution of the number of echo pulses, and detected the virtual points by comparing the symmetry and geometric similarity of points. They also improved the glass region estimation method in [7] by applying the superpixel method [27] to cluster the glass regions in the panoramic images. The initial method [7] was generzlied in [6], where multiple glass planes are estimated, respectively, and the multiple trajectories of light reflection were investigated.

Whereas the existing methods remove the virtual points using the hand-crafted features, the proposed method detects the virtual points by exploiting deep features of LS3DPCs based on voxel representation. Thus the proposed method successfully works with exceptional cases which are not handled in the existing methods. For example, the proposed method yields reliable performance even with the density difference and locally empty spaces of point clouds, and furthermore, learns the features that

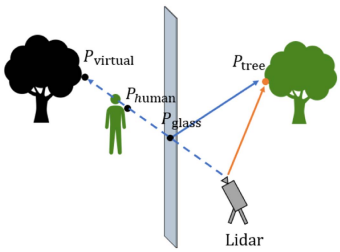


Fig. 2. Principle of virtual point generation. The real and virtual objects are colored in green and black, respectively.

can distinguish the objects with similar shapes but different orientations.

III. Problem Statement

While capturing a real-world scene by using LiDAR scanners, virtual points are generated due to the reflection of light on reflective or specular surfaces such as glass. Fig. 2 illustrates the principle of virtual point generation with multi-echo LiDAR scanners. A light ray (orange), emitted from LiDAR, hits the tree and returns to LiDAR generating a real 3D point P_{tree}. Another light ray (blue dashed) first hits a glass plane and generates a real 3D point P_{glass} on the glass. However, the light is then transmitted and reflected on the glass simultaneously, where the transmitted light additionally generates a real 3D point P_{human} on the human behind the glass plane. On the other hand, the reflected light (blue solid) hits the tree in front of the glass plane that generates a virtual 3D point P_{virtual} behind the glass in a symmetric location to the real point P_{tree} with respect to the glass plane, since the LiDAR scanner is unaware of the existence of glass. We aim to detect and remove the virtual points from a given input point clouds and maintain real points only.

IV. Proposed Method

We propose a learning-based virtual point removal method composed of glass probability estimation network and 3D feature similarity estimation network that are trained independently in a two-step manner. Fig. 3 shows the overall inference process of the proposed method. In the glass probability estimation module, we assign glass probabilities to the pixels of 360 ° image by extracting the deep features from the distribution of the number of echo pulses. Then, based on the voxel representation in 3D space, we iteratively compute the feature similarity between the corresponding points in symmetric positions with respect to the estimated glass plane. Finally, we detect the virtual points by thresholding the result of multiplication between the estimated glass probability and the feature similarity at each point.

A. Glass Probability Estimation

As discussed in Section III, multiple 3D points can be generated from a single light ray hitting the glass surface. For example, the emitted light (blue) in Fig. 2 generates the three points of P_{glass}, P_{human}, and P_{virtual}. We exploit this property to estimate the glass region in 2D image domain. Specifically, as shown in Fig. 4, we project the generated points of an input LS3DPC model onto the surface of the unit sphere and visualize the distribution of the number of points, called *count map*, by unfolding the spherical grids into 2D image domain. We consider a 3×3 surface patch as each pixel in the count map that corresponds to the frustum in 3D space shown in Fig. 4(a). We see that the glass regions are associated with relatively large numbers of points than non-glass regions. However, far-away background objects often exhibit large numbers of points as well, and no points can be sampled on some glass regions where real objects are attached directly behind the glass plane.

To detect the glass regions more reliably and accurately, we extract the deep features from the 2D count map by using a ResNet32 [28] based segmentation network. The extracted features are further refined by using the channel attention and spatial attention based on the dual attention scheme [29]. In the last layer, we apply the sigmoid function to represent the glass probability. Since we have a binary classification task between glass and non-glass regions, the glass probability estimation network is trained by using the binary cross-entropy loss

$$L_{\text{c}}(\mathbf{X}, \mathbf{Y}) = -\frac{1}{N}\sum_{i=1}^{N}\left\{\mathbf{Y}_i \log(\mathbf{X}_i) + (1-\mathbf{Y}_i)\log(1-\mathbf{X}_i)\right\}, \tag{1}$$

where $\mathbf{X}$ is a synthetically generated count map as training data and $\mathbf{Y}$ is the corresponding ground truth map of glass regions. $\mathbf{X}_i$ and $\mathbf{Y}_i$ denote the values at the i-th pixels in $\mathbf{X}$ and $\mathbf{Y}$, respectively, and N is the number of total pixels.

The count map and the resulting probability map are shown in Fig. 4(b) and (c), respectively, where we see that the resulting map captures the glass regions faithfully. We consider the pixels having probability values higher than a threshold of 0.7 as candidates, and then select the point closest to the LiDAR location at each pixel. We finally estimate the glass plane in 3D space by applying the plane RANSAC [30] to the selected points over an entire image. Note that the selected points are highly likely sampled on the glass plane since the objects behind the glass are always farther than the glass plane from the location of LiDAR scanner.

B. 3D Feature Similarity Estimation

The glass plane divides the 3D space into the front space Ω_{front}, where the LiDAR location belongs, and the back space Ω_{back}. Note that the virtual points appear in Ω_{back} only. Therefore, for each point $P \in \Omega_{\text{back}}$, we determine whether it is a virtual point or not. To this end, we compare the symmetry and geometric similarity between $P \in \Omega_{\text{back}}$ and its corresponding real point in Ω_{front}.

However, there are challenges to find the corresponding real points to given virtual points. The virtual points exhibit relatively lower density than that of their corresponding real points. Moreover, some virtual points in Ω_{back} may have no corresponding real points in symmetric positions in Ω_{front} due to the occlusion. To deal with these characteristics, we first voxelize the points and then apply the 3D convolution and max-pooling consecutively to the voxelized points in the proposed 3D feature similarity estimation network. We intuitively assume that, when humans search for virtual points, they usually focus on the global features first compared to the local geometry of points. Thus we employ the multi-scale voxel representation with a bigger scale in the point cloud featuring process, and apply the convolution and downsampling to grasp the global features. We measure the 3D feature similarity between the feature vectors of the voxel of

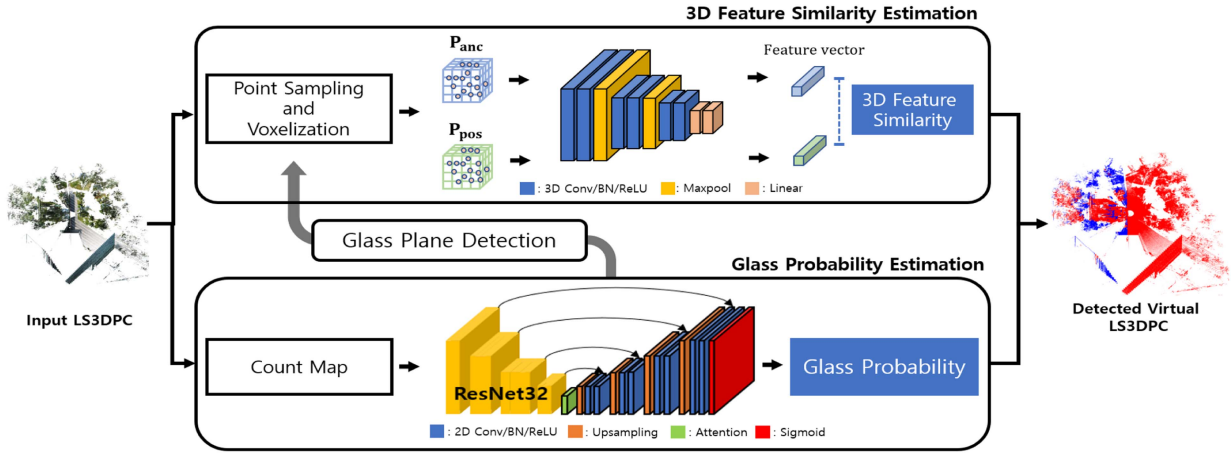


Fig. 3. Overall framework of the proposed 3D reflection removal method.

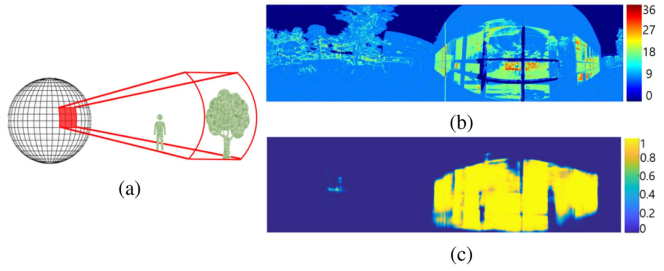


Fig. 4. Count map and glass probability map. (a) Illustration of the surface patch composed of 3×3 spherical grids and the frustum space covered by the surface patch. (b) The count map showing the distribution of the number of projected points at each surface patch. (c) The glass probabilty map estimated from the count map.

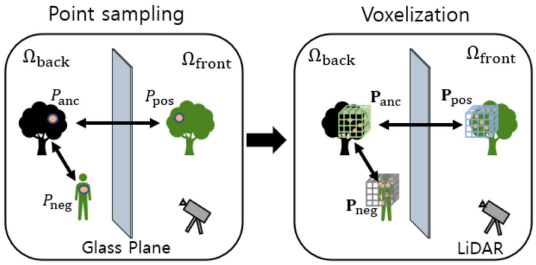


Fig. 5. Point sampling and voxelization. For a given virtual point P_{anc}, we select its positive sample $P_{\text{pos}} \in \Omega_{\text{front}}$ and negative sample $P_{\text{neg}} \in \Omega_{\text{back}}$, respectively. Then we voxelize the neighboring points centered on P_{anc}, P_{pos}, and P_{neg}, respectively.

a query point in Ω_{back} and that of the symmetric real point in Ω_{front}. Then we assign the same similarity score to all the points inside the same voxel. This voxel selection is repeated in a voting manner [31] until the similarity estimation network computes the similarity scores for all the points.

For training the 3D feature similarity estimation network, we employ the triplet loss. Fig. 5 shows the selection of the positive and negative samples inspired by [24], [32]. For a given query point $P_{\text{anc}} \in \Omega_{\text{back}}$, we find the positive sample $P_{\text{pos}} \in \Omega_{\text{front}}$ by using the householder matrix [6], which is corresponding to

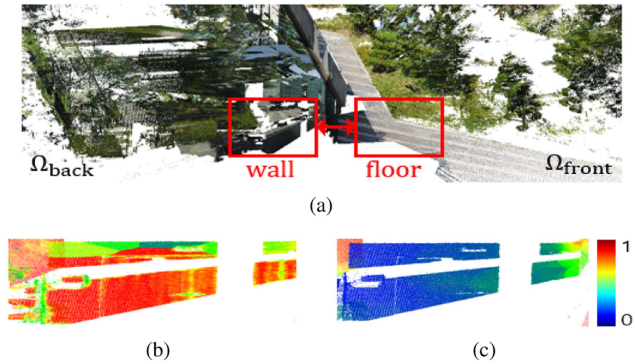


Fig. 6. 3D feature similarity estimation. (a) A 3D scene where both of the wall in Ω_{back} and the floor in Ω_{front} have planar shapes. The similarity scores obtained by using (b) the previous method [6] and (c) the proposed method, respectively.

P_{anc} in a symmetric location with respect to the glass plane. We also select a negative sample P_{neg} from the real points in Ω_{back}, where P_{neg} has a FPFH [9] feature similarity to P_{anc} lower than a threshold value such that the geometric shape of P_{neg} is different enough from that of P_{pos}. Then we create the voxels $\mathbf{P}_{\text{anc}}$, $\mathbf{P}_{\text{pos}}$, and $\mathbf{P}_{\text{neg}}$, with a pre-defined size, centered on P_{anc}, P_{pos}, and P_{neg}, respectively. The points in $\mathbf{P}_{\text{pos}}$ are set to be positive because $\mathbf{P}_{\text{anc}}$ is the reflection of $\mathbf{P}_{\text{pos}}$ and thus should have similar shapes unless occluded. However, different objects in symmetric positions with respect to the glass plane may have similar shapes, for example, both of the wall and floor have planar shapes but they have different orientations from each other, as shown in Fig. 6. In such a case, these objects should be distinguished from each other, and therefore, we augment the point clouds with random rotations before voxelization to guide the network to learn the shapes in various orientations. Moreover, to consider the occlusion, we do not consider P_{anc} in virtual point detection when the corresponding $\mathbf{P}_{\text{pos}}$ is empty, where the voxels containing less than 10 points are considered to be empty.

The 3D feature similarity estimation network is trained such that the feature vectors of the virtual points are forced to be close to the feature vectors of their corresponding positive points,

while being pushed away from the feature vectors of the negative points. In practice, we train the network by using the triplet margin loss given by

$$L_{\text{triplet}}(\mathbf{P}_{\text{anc}}, \mathbf{P}_{\text{pos}}, \mathbf{P}_{\text{neg}}) = \max\left\{||\Phi(\mathbf{P}_{\text{anc}}) - \Phi(\mathbf{P}_{\text{pos}})|| - ||\Phi(\mathbf{P}_{\text{anc}}) - \Phi(\mathbf{P}_{\text{neg}})\| + 1, 0\right\}, \tag{2}$$

where $\Phi(\cdot)$ refers to the feature vectors in the 3D feature similarity estimation network.

V. Experimental Results

We evaluate the performance of the proposed learning-based framework compared with the existing methods [6], [8] by using the synthetically generated dataset and manually labeled real dataset. Note that various hand-crafted features were employed to find accurate glass regions [6], and RGB color values of points were additionally used to cluster the point clouds [8]. 3D visualization results are shown in the supplementary material.

A. Datasets

Synthetic Training Dataset: Collecting a large number of LS3DPC models with reflection artifacts by using terrestrial LiDAR scanners is labor-intensive and time-consuming task. Therefore, to obtain the training data, we add synthetically generated reflection artifacts to real LS3DPC models. We first used a terrestrial LiDAR scanner [33] to capture 10 LS3DPC scenes without reflection artifacts. For each scene, we synthetically placed 100 glass planes with arbitrarily selected sizes ranging from 6∼12 meters in width and 3∼5 meters in height, respectively, and we generated 1000 LS3DPC models with reflection artifacts in total. Specifically, we reconstructed mesh surfaces for LS3DPC models, and performed the ray casting to the mesh surfaces by using the Open3D API [34] with the same resolution of LiDAR scanning to [7]. To generate more realistic LS3DPC models considering the weak intensity of echo pulses in real-world environment, we randomly removed one-third of the rays transmitted through a glass plane and removed one-half of the points sampled on the glass plane and the virtual objects, respectively, during the ray casting. Also, we generated the count maps accordingly by projecting 3D points into 2D image domain. Fig. 7 compares (top) a real LS3DPC model with real reflection artifacts and (bottom) a LS3DPC model with synthetically generated reflection artifacts that is modified from (middle) a real LS3DPC model without reflection. We see that the overall characteristics of the synthetic model are very close to that of the real model in terms of the glass shape and the count map.

Real Test Dataset: To evaluate the performance of the reflection removal on real data with the ground truth labeling, we manually annotated 6 scenes with single glass plane selected from UNIST LS3DPC dataset [7]. We initially assigned the labels of virtual and real points by using [6], and then refined the labels manually using a 3D visualization tool [35]. We excluded the points sampled near the glass plane, within 20 cm from the glass plane, due to the ambiguity caused by the sampling noise and the error of glass plane estimation in [7].

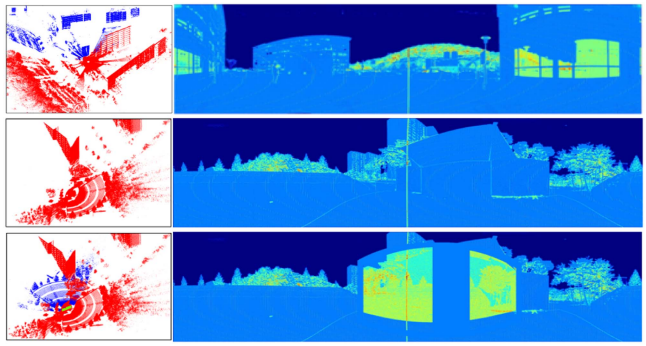


Fig. 7. Synthetically generated reflection artifacts: *Top:* A real scene with real reflection artifacts. *Middle:* A real scene without reflection artifacts. *Bottom:* A real scene modified from the middle one with synthetically generated reflection artifacts where an arbitrarily placed glass plane is colored in green. The real and virtual points are colored in red and blue, respectively.

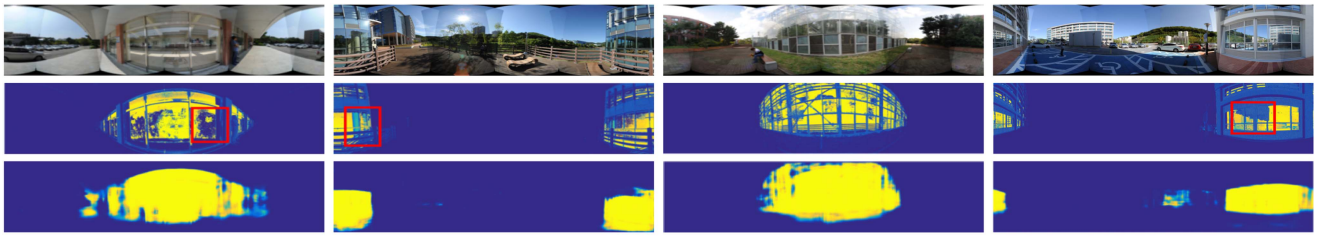
B. Training Details

Glass Probability Estimation Network: Among the count maps of the 1000 generated LS3DPC models with synthetic reflection artifacts, we used 800 count maps to train the glass probability estimation network and used the other 200 maps for testing. The glass probability estimation network was trained for 200 epochs. We resized the input count maps to the size of 256×1024, and applied the augmentation schemes of shifting and translation to avoid overfitting. We set the batch size to 32, and the learning rate to 0.0001. We used the Adam optimizer [36].

3D Feature Similarity Estimation Network: Each LS3DPC model has nearly 5 millions of points causing high computational complexity in 3D feature estimation, and therefore, we used 17 models of good quality among the 1000 LS3DPC models with synthetic reflection artifacts. However, to make the 3D feature similarity estimation network learn various shapes, we additionally included 12 more scenes from Semantic3D dataset [37], which has European-style buildings and vegetation. Among the 29 models, we used 25 models for training and the remaining 4 models for testing, respectively. We trained the network for 60 epochs where each epoch consists of 200 iterations. We took the space with the size of 3.2 meters for voxel sampling, which is then divided into $32 \times 32 \times 32$ voxels with the size of 0.1 meters, considering that the scale of a single LS3DPC model is usually larger than 100 meters. We also applied the rotation and scale augmentation to the points. We set the batch size to 16 and the learning rate to 0.0001. We used the Adam optimizer.

C. Qualitative Performance

Glass Probability Estimation: We first evaluate the performance of the proposed glass probability estimation method compared with the state-of-the-art method [6]. Fig. 8 compares the estimated probability maps. As depicted in the red boxes, the existing method often fails to capture the entire glass areas completely. For example, no points are sampled in the missing regions on the glass plane due to the weak intensity of echo pulses in 'Gymnasium' scene, and multiple echo pulses are not returned in the missing regions in 'Terrace' and 'Natural science building', where some objects, such as curtains and pillars, are



(a) Gymnasium (b) Terrace (c) Botanical garden (d) Natural science building

Fig. 8. Comparison of the estimated glass probability maps. The first row shows the panoramic images including glass regions. The second and third rows show the resulting probability maps estimated by using the existing method [6] and the proposed method, respectively.

TABLE I
QUANTITATIVE PERFORMANCE COMPARISON IN TERMS OF THE OVERALL F1 SCORE EVALUATED ON THE SIX TEST SCENES FROM [7]

Method	(a)	(b)	(c)	(d)	(e)	(f)	Average
[8]	0.615	0.585	0.729	0.499	0.880	0.520	0.638
[6]	0.694	0.822	0.627	**0.706**	0.777	0.379	0.668
Proposed	**0.766**	**0.862**	**0.781**	0.625	**0.924**	**0.862**	**0.803**

The best scores are highlighted in bold. (a) Architecture building. (b) Botanical garden. (c) Engineering building. (d) Gymnasium. (e) Natural science building. (f) Terrace.

closely attached to the glass planes. Then the virtual points associated with the false negatives in the detected glass regions are not considered in the reflection removal process. In contrary, the proposed method grasps such challenging regions reliably, and furthermore, fills in locally missing glass regions faithfully compared with the existing method. The proposed method often provides some false positives in the glass detection results, however the proposed feature similarity estimation module alleviates the effect of such false positives by enforcing the conditions of symmetry and geometric similarity.

Reflection Removal: We evaluate the performance of the reflection removal by showing the detected virtual points in Fig. 9. The glass planes are visualized in yellow in Fig. 9(a), and the real and virtual points are colored in red and blue, respectively. As shown in Fig. 9(c), the existing method [6] provides degraded performance to separate the virtual points from the real points compared to the ground truth. Moreover, it often fails to detect the virtual points of far away structures as shown in 'Architecture building' and 'Terrace.' However, as shown in Fig. 9(d), the proposed method faithfully detects the virtual points, even when they are mixed together with the real points, outperforming the compared existing method while capturing the far away structures successfully.

D. Quantitative Performance

In Table I, we compared the quantitative performance of the proposed method with that of [8] and [6] by using the manually labeled 6 real scenes. For all the test models except 'Gymnasium,' the proposed method outperforms the previous methods by large margins in terms of the average F1 score. Note that relatively many points are sampled on the floor inside the building in the 'Gymnasium' scene, where both the floor and the ground outside the building exhibit planar shapes and have a symmetric relation to each other. Therefore, the proposed method captures such symmetric and geometrically similar relations of points faithfully, and detected the real floor as virtual yielding quantitatively worse performance.

E. Ablation Study

Table II shows the effect of the proposed modules of the 3D feature similarity estimation and the glass probability estimation, respectively. We employ [6], the current state-of-the-art method in multi-echo based 3D reflection removal, as our baseline. We see that, when applying the proposed method to the baseline [6], the reflection removal performance is increased by 0.138 on average, in terms of the F1 score. Especially, we have a significant gain of 0.451 on 'Terrace,' that coincides with the qualitative results compared in the last row in Fig. 9. However, the 3D feature similarity estimation module does not have a gain on 'Botanical garden' due to the irregularly shaped vegetation both inside and outside of the building. In this case, it becomes quite tricky to reliably classify the mixed real and virtual points since the real points associated with the inside vegetation can be detected as the virtual points corresponding to the real points associated with the outside vegetation. On the other hand, the proposed glass probability estimation method also degrades the performance on 'Gymnasium' that has the floors at the symmetric locations inside and outside of the building. The glass probability map obtained by the baseline fails to completely capture the entire glass areas, and therefore lots of real points on the floor inside the building are not detected as virtual due to inaccurate symmetry relation. However, the proposed method accurately estimates the glass plane and hence the real points on the indoor floor are detected as virtual points corresponding to the real points sampled on the floor outside of the building due to accurately computed symmetry relation and geometric similarity. Consequently, correct estimation of the glass regions in the proposed method degrades the performance of reflection removal on the 'Gymnasium' scene ironically.

VI. DISCUSSION

The proposed method was designed assuming planar glasses only, and hence fails to work with curved glasses due to the difficulty in symmetry computation between point clouds. Moreover, the reflection artifacts also occur with various non-glass reflective surfaces such as water surface and marble floors in buildings. Fig. 10 shows an example where we see the virtual points of the person and trees below the wet ground in a rainy day. However, in such a challenging case, the reflection artifacts often appear in different shapes and densities from that of the

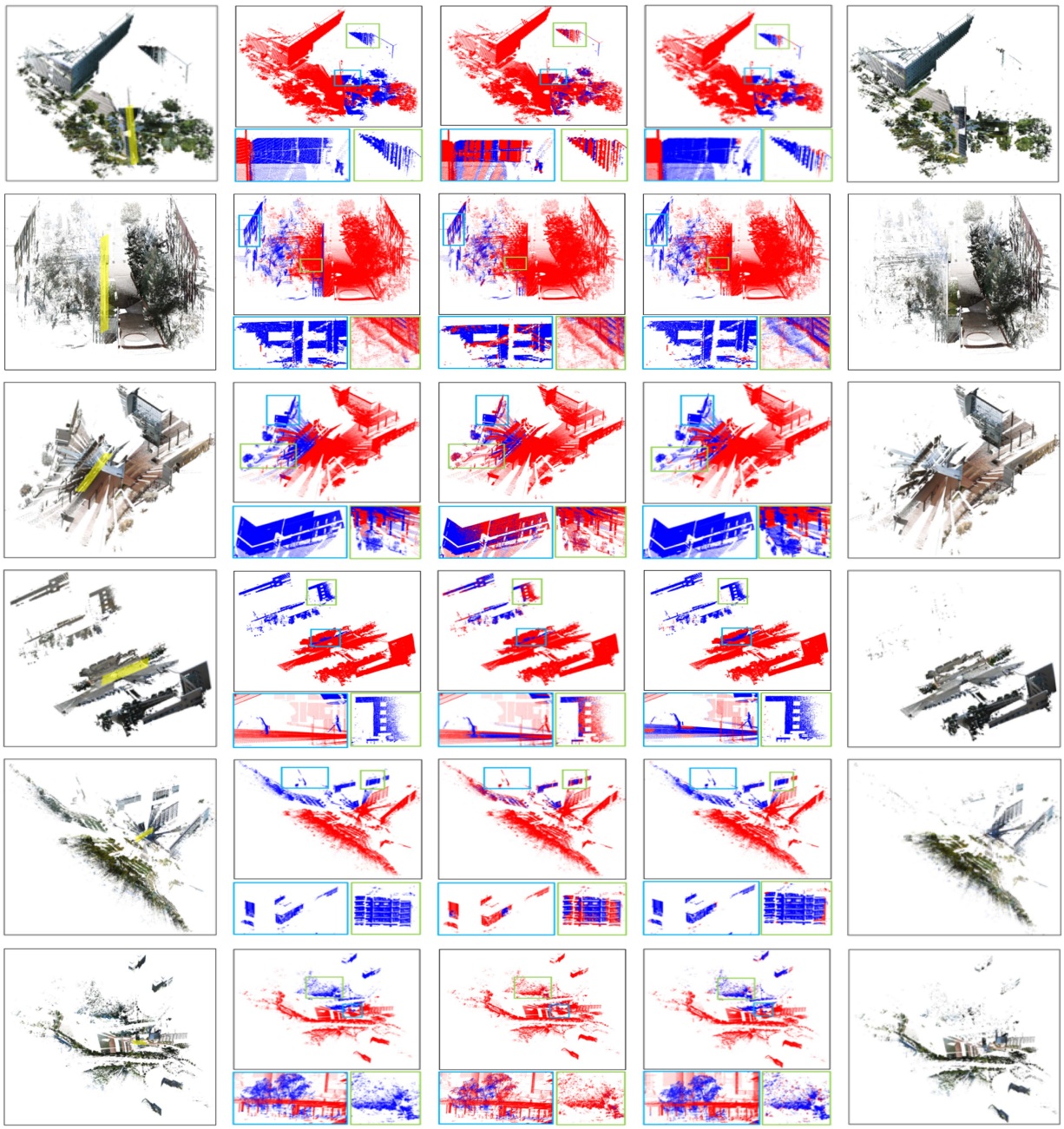


Fig. 9. Comparison of the virtual point detection results. (a) Input LS3DPC models where the glass planes are visualized in yellow. (b) The ground truth labeling of real and virtual points. (c) The virtual point detection results of the existing method [6]. (d) The virtual point detection results and (e) the refined LS3DPC models obtained by using the proposed method. The real and virtual points are colored in red and blue, respectively. From top to bottom, 'Architecture building,' 'Botanical garden,' 'Engineering building,' 'Gymnasium,' 'Natural science building,' and 'Terrace.'.

TABLE II
EFFECT OF THE PROPOSED MODULES OF THE 3D FEATURE SIMILARITY ESTIMATION (FSE) AND THE GLASS PROBABILITY ESTIMATION (GPE)

Method	3D FSE	GPE	(a)	(b)	(c)	(d)	(e)	(f)	Average
Baseline [6]			0.694	**0.822**	0.627	0.706	0.777	0.379	0.668
Proposed	✓		0.745	0.787	0.634	**0.751**	0.893	0.666	0.746
	✓	✓	**0.820**	0.821	**0.805**	0.609	**0.954**	**0.830**	**0.806**

The best scores are highlighted in bold. (a) Architecture building. (b) Botanical garden. (c) Engineering building. (d) Gymnasium. (e) Natural science building. (f) Terrace.

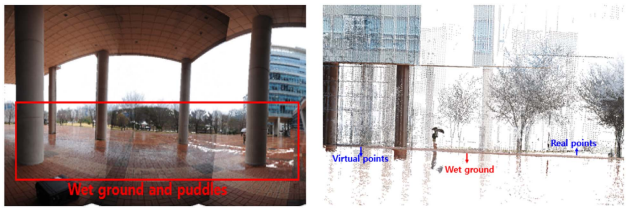


Fig. 10. Reflection on non-glass surface. The virtual points are generated by the reflection on the puddles and wet ground in a rainy day.

original objects. Moreover, the multiple echo property of LiDAR may not hold that makes it hard to apply the proposed method. It can be a future research topic to develop a more generalized reflection removal method to handle the curved glasses and non-glass reflective surfaces.

VII. Conclusion

We proposed a novel learning-based framework for reflection removal in LS3DPCs. We first designed the glass probability estimation network that investigates the distribution of the LiDAR's echo pulses on 2D image domain. Also, we devised the 3D feature similarity estimation network that extracts deep features of 3D points based on the voxel representation which are then used to grasp the symmetry relation with geometric similarity between real and virtual points. We trained the proposed network using LS3DPC models with synthetically generated reflection artifacts, and tested it on real datasets with manually annotated ground truth labels. Experimental results demonstrated that the proposed method significantly outperforms the state-of-the-art methods qualitatively and quantitatively.

References

[1] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst. Conf.*, 2014, pp. 1–9.
[2] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
[3] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.
[4] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, 2018, Art. no. 3337.
[5] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2018, pp. 7652–7660.
[6] J.-S. Yun and J.-Y. Sim, "Virtual point removal for large-scale 3D point clouds with multiple glass planes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 729–744, Feb. 2021.
[7] J.-S. Yun and J.-Y. Sim, "Reflection removal for large-scale 3D point clouds," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2018, pp. 4597-4605.
[8] J.-S. Yun and J.-Y. Sim, "Cluster-wise removal of reflection artifacts in large-scale 3D point clouds using superpixel-based glass region estimation," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 1780–1784.
[9] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3212–3217.
[10] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2017, pp. 652–660.
[11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.
[12] Q. Hu et al., "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2020, pp. 11108–11117.
[13] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2018, pp. 918–927.
[14] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2017, pp. 1907–1915.
[15] R. Heinzler, F. Piewak, P. Schindler, and W. Stork, "CNN-based lidar point cloud de-noising in adverse weather," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2514–2521, Apr. 2020.
[16] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D lidar using fully convolutional network," in *Proc. Robot.: Sci. Syst. Conf.*, 2016.
[17] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 922–928.
[18] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.
[19] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2020, pp. 10529–10538.
[20] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2018, pp. 206–215.
[21] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
[22] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 641–656.
[23] B. Graham, M. Engelcke, and L. v. d. Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2018, pp. 9224–9232.
[24] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2017, pp. 1802–1811.
[25] R. Gao, J. Park, X. Hu, S. Yang, and K. Cho, "Reflective noise filtering of large-scale point cloud using multi-position LiDAR sensing data," *Remote Sens.*, vol. 13, 2021, Art. no. 3058.
[26] R. Gao, M. Li, S.-J. Yang, and K. Cho, "Reflective noise filtering of large-scale point cloud using transformer," *Remote Sens.*, vol. 14, 2022, Art. no. 577.
[27] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[29] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2019, pp. 3146–3154.
[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *ACM Commun.*, vol. 24, no. 6, pp. 381–395, 1981.
[31] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6411–6420.
[32] Z. Gojcic, C. Zhou, J. D. Wegner, and W. Andreas, "The perfect match: 3D point cloud matching with smoothed densities," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2019, pp. 5545–5554.
[33] "Vz-400 terrestrial lidar scanner." [Online]. Available: http://www.riegl.com/nc/products/terrestrial-scanning/produktdetail/product/scanner/48/
[34] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.
[35] "Cloudcompare [GPL software]," 2021. [Online]. Available: http://www.cloudcompare.org/
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[37] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "SEMANTIC3D.NET: A new large-scale point cloud classification benchmark," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 91–98, 2017.