Self-Annotated 3D Geometric Learning for Smeared Points Removal

Miaowei Wang
University of Edinburgh
m.wang-123@sms.ed.ac.uk

Daniel Morris
Michigan State University
dmorris@msu.edu

Abstract

There has been significant progress in improving the accuracy and quality of consumer-level dense depth sensors. Nevertheless, there remains a common depth pixel artifact which we call smeared points. These are points not on any 3D surface and typically occur as interpolations between foreground and background objects. As they cause fictitious surfaces, these points have the potential to harm applications dependent on the depth maps. Statistical outlier removal methods fare poorly in removing these points as they tend also to remove actual surface points. Trained network-based point removal faces difficulty in obtaining sufficient annotated data. To address this, we propose a fully self-annotated method to train a smeared point removal classifier. Our approach relies on gathering 3D geometric evidence from multiple perspectives to automatically detect and annotate smeared points and valid points. To validate the effectiveness of our method, we present a new benchmark dataset: the Real Azure-Kinect dataset. Experimental results and ablation studies show that our method outperforms traditional filters and other self-annotated methods. Our work is publicly available at https://github.com/wangmiaowei/wacv2024_smearedremover.git.

1. Introduction

While dense depth sensors have led to dramatic improvements in 3D computer vision tasks, including alignment [5], classification [45], and reconstruction [22], they nevertheless still suffer from depth artifacts which can harm performance. Factors including scene complexity [39], hardware device conditions [17], and sensor motion [35] can adversely impact depth. Fortunately, consumer-level depth sensors have improved over the years [44], with long-standing problems such as Gaussian noise, shot noise, and multi-path interference being alleviated. However, there continues to exist an important class of invalid depth points at the boundaries of objects, as shown in Fig. 1. These points often interpolate between objects across depth discontinuities, and so we call them **smeared** points, in contrast to other outliers or random noise. Our primary goal is to eliminate smeared points without harming other depth points, especially valid boundary details.

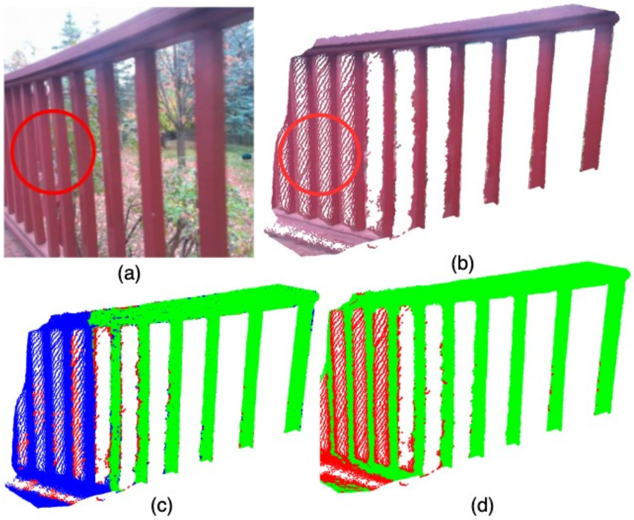


Figure 1. Example scene recorded by an Azure Kinect sensor with smeared points. The cropped color is shown in (a) and a colorized side view of the 3D point cloud is in (b). Significant smearing can be seen between the vertical columns in the red circles. In subplot (c), our method uses multiple viewpoints to automatically annotate smeared points (red) from valid points (green) and left uncertain points (blue). Once trained, our method classifies pixels in a single frame as smeared or valid in subplot (d).

A primary cause of smeared points is multi-path reflections. Pixels on or adjacent to edge discontinuities can receive two or more infrared signal reflections; one from the foreground object and one from the background. Depending on the sensor circuitry, these multiple returns can result in a variety of artifacts and typically are interpolated between the foreground and background object. Common depth noise has a small bias compared to variance and low dependence on 3D shapes. In contrast, smeared point noise, caused by multi-path interference, depends strongly on 3D scenes with one-sided distributions at object boundaries, see

Fig. 1. These smeared points can be problematic for applications that use depth maps as they result in false surfaces in virtual worlds, blurring of fine 3D structures, and degraded alignments between point clouds. These harms are compounded when multiple point clouds each having different artifacts are combined into an overall blurred point cloud.

Now, improvements in sensor processing have given modern sensors the ability to remove some of these smeared points, particularly when there is a large gap between the foreground and background objects. Nevertheless, smearing across smaller depth discontinuities is not solved due to the difficulty in distinguishing occlusion effects from complex shape effects, and as a consequence smeared points continue to plague the otherwise high-quality depth images, shown in Fig. 1. A variety of hand-crafted filters [13,14,43] can be used to reduce noise in-depth maps, but we find that they perform poorly in removing smeared points or else result in overly smoothed surfaces. A data-driven approach would be preferable, but these face the difficulty of acquiring sufficient ground truth which is expensive and time-consuming to obtain. More importantly, it should be pointed out that smeared points extensively exist in current famous RGB-D datasets such as LaMAR [37], NYU Depth V2 [34], and ScanNet [12]. Thus the smeared point is not a niche problem. And there are still smeared points in their provided well-reconstructed ground truth 3D models shown in Fig. 2, which prevents getting clean depth maps from large-scale off-the-shelf datasets.

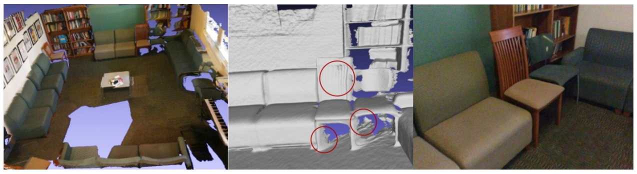


Figure 2. A well-reconstructed 3D model example of ScanNet [12] (left) contains smeared points in the red circles (middle), and the color image (right) is provided for comparison.

Another approach is to create synthetic datasets [32] with known ground truth, but these are limited by how well they model both the sensing and the environment. Unsupervised domain adaption [1, 38] can address this to some extent. However, approaches using multiple different frequencies [3] from the same position, or using multiple cameras [42] create significant overhead in acquisition.

The goal of this paper is to overcome the difficulty in acquiring ground truth data for hand-held depth sensors by developing a novel self-annotated method for eliminating smeared points. This avoids the need for building complex optical sensing models, and it also eliminates the need for expensive manual annotations of data. Instead, our approach leverages the dense depth sensing capabilities of these sensors, along with a multi-view consistency model to automatically self-annotate points. In this way, data can be rapidly acquired without human annotation and used to train a smeared-point remover.

In order to evaluate this method, fifty different real scenes both indoors and outdoors have been collected. Comprehensive experiments on these datasets and ablation studies further demonstrate the core idea in this paper that multi-frame self-annotation can effectively train a smeared point remover. In summary, our contributions are:

- To our knowledge, we propose the first self-annotation technique for smeared points detection that applies geometric consistency across multiple frames.
- By combining self-annotated labels with a pixel-level discriminator, we create a self-annotated smeared point detector.
- We introduce a new real smeared points dataset (AzureKinect) using the Azure Kinect sensor as a benchmark.
- We validate our design choices with several ablations.

2. Related Work

Obtaining noise-free, dense depth from raw, low-quality measurements has received significant attention. Before the rise of data-driven techniques, especially deep learning, numerous hand-crafted filters were designed to alleviate noise by referencing neighboring pixels, such as median filter [14], Gaussian filter [13], Bilateral filter [43], etc. Early work to remove outliers introduced density-based and statistical methods [9,15,41], while geometric and photometric consistency between depth maps and color images [26, 46] was also used to detect outliers. As for time-of-flight multipath interference (MPI), multiple different modulation frequency measurements [7, 8] of the same scene are collected to improve depth quality. In contrast to these methods requiring multiple measurements at different frequencies, our method requires only a single-frequency depth map.

Even before deep learning techniques were widely adopted, convolution and deconvolution techniques [24] were proposed to recover time profiles only using one modulation frequency. DeepToF [32] uses an autoencoder to correct measurements based on the observation that image space can provide most of the sources for MPI. Continuing the classical multi-frequency method, a multi-frequency ToF camera [3] is integrated into the network design to preserve small details based on two sub-networks. RADU [38] updates depth values iteratively along the camera rays by projecting depth pixels into a latent 3D space. These supervised learning methods heavily rely on synthetic datasets generated by a physically-based, time-resolved renderer [23] that uses bidirectional ray tracing which is much more time-consuming to render one realistic depth map. To shrink the gap between real and synthetic

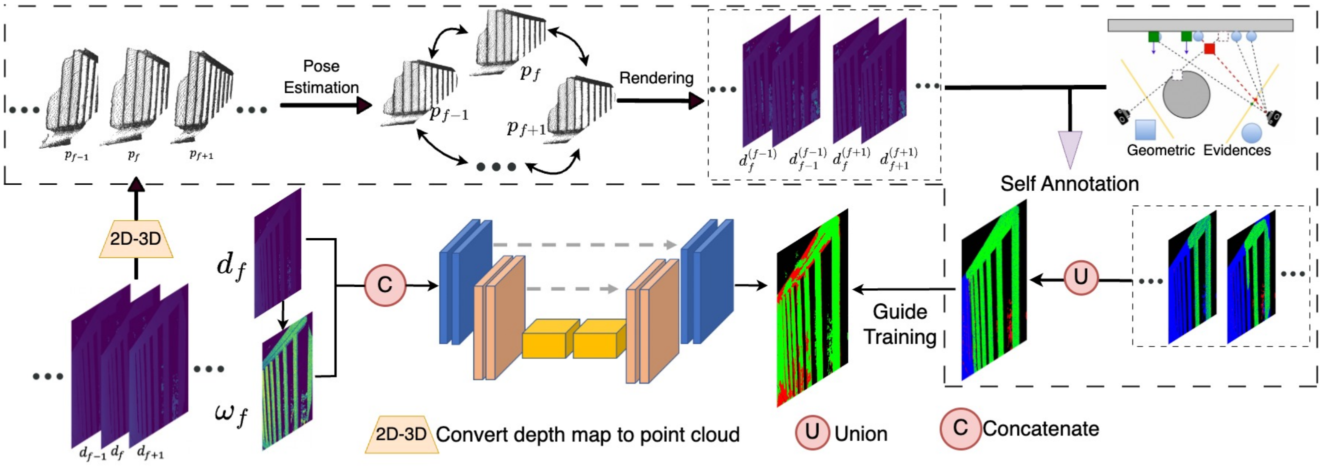


Figure 3. Our self-annotated architecture for smeared point removal. Training scenes are recorded with a hand-held sensor. Multi-frame pose estimation aligns these frames. Then geometric consistency is used to annotate smeared(red), and valid(green) pixels for all frames with left points as unknown(black). Then a U-Net-based classifier is trained to identify smeared points in each frame.

datasets, DeepToF [32] learns real data's statistical knowledge by auto-learning encoders while RADU [38] applies unsupervised domain adaptation by investigating a cyclic self-training procedure derived from existing self-training methods for other tasks [27, 29, 30]. Additionally, an adversarial network framework can be used to perform unsupervised domain adaptation from synthetic to real data [2]. All these methods depend on the reliability of the simulated dataset. Moreover, current self-supervised methods either require a setup of multiple sensors placed in precomputed different positions based on photometric consistency and geometric priors [42] or build noise models by assuming noises follow some random distribution around normal points [16,21] which leads to low availability when processing real scenes. In contrast to these approaches, our method operates in a self-annotated manner directly on real scene data without relying on complex scene formation models or specific noise models, or synthetic datasets.

3. Method

3.1. Approach Summary

This paper divides the smeared point removal into two distinct components: (1) a pixel annotator and (2) a pixel classifier, which are illustrated in Fig. 3. Advances in correcting depth offsets [21, 32, 38, 42] lead to high-quality depth estimates for the majority of depth pixels, leaving a typically small fraction of invalid or smeared pixels. With these pixels often having large errors, our approach is to identify them for removal rather than correct their depth. Thus smeared point removal is a classic semantic segmentation problem and if we had sufficient annotated data, then a supervised classifier could be trained to perform this task. The challenge is how to obtain sufficient annotated data, as manual annotation is time-intensive and expensive.

In this section first, we describe two types of evidence for classifying pixels as either smeared or valid. By accumulating this evidence from multiple scene views, we create an automated smeared-pixel and valid-pixel annotation method. We then use these annotations to train a supervised single-frame smeared pixel classifier

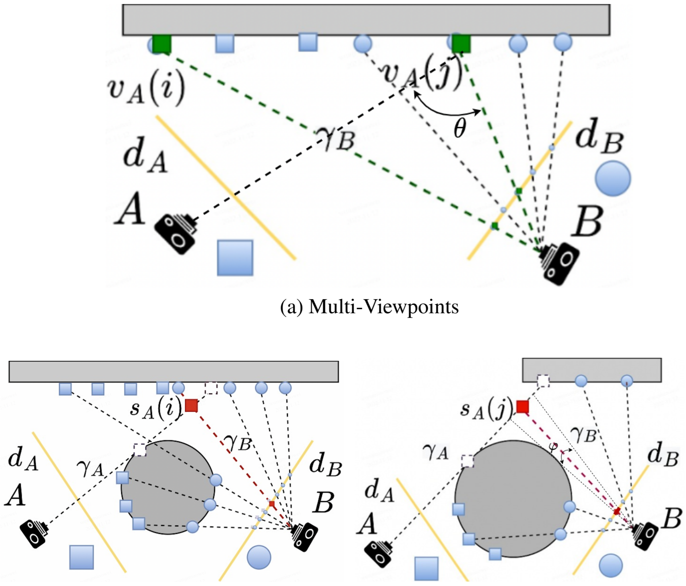
3.2. Multi-View Annotation

Typically smeared pixels occur between objects along rays that graze the foreground object. Now, as the viewpoint changes, these grazing rays change orientation and the resulting location of any interpolated points along these rays will also change. On the other hand, 3D points on objects will remain consistent, or at least overlap, between differing viewpoints. Thus we conclude if a pixel has been observed from multiple viewpoints with differing rays, the pixel must be a valid surface pixel and not a smeared point.

An example of **multi-viewpoint evidence** is shown in Fig. 4a. Points $v_A(i)$ and $v_A(j)$ are observed from separate viewpoints A and B and thus determined to be valid points. Now if the distance between viewpoints is small or the distance to the pixels is large, smeared pixels can coincide spatially. To avoid this, we use the angle θ, always less than 90°, between the viewing rays of coincident points as a confidence measure in a point that is valid, and the confidence score c can be modeled as Eq. (1)

$$c = \sin^2(\theta) \tag{1}$$

The normalization is applied to the confidence score c to be in the range between 0 and 1. Tab. 3 validates this design.



(a) Multi-Viewpoints

(b) See-Through Behind

(c) See-Through Empty

Figure 4. Geometric evidence used for annotating our depth maps. Multi-Viewpoints evidence for valid points (green) is shown in the top row. Two cases of See-Through evidence for smeared points (red) are shown in the bottom two rows.

3.3. Space Carving Annotation

The second category of evidence we gather has to do with space carving. Smeared points, by definition, float off the surface of objects. Now if a ray measuring a depth pixel passes through the location of a 3D point, then this is evidence that that pixel is not actually at that location but is most likely a smeared pixel.

We divide **see-through evidence** for smeared points into a case of positive evidence(See-through Behind) in Fig. 4b and negative evidence(See-through Empty) in Fig. 4c. In both cases, a point is concluded to be a smeared point if another viewpoint can see through it. In the first case, Fig. 4b, a ray γ_B from the camera at location B passes through a point $s_A(i)$, observed from location A, and measures a point behind $s_A(i)$, from which we conclude $s_A(i)$ is a smeared point. In the second case Fig. 4c, a point $s_A(j)$ observed by A should be visible to viewpoint B, and yet there is no measurement along the ray γ_B, either closer or farther than $s_A(j)$. To conclude from this negative evidence that $s_A(j)$ is a smeared point we expand the ray γ_B between the sensor and $s_A(j)$ to a conical section with angle φ and require no points are observed from B within this, which eliminates the case of grazing rays being blocked and incorrectly inferring a smeared point behind them. The conical section angle φ is a regularization term in See-Through Empty and larger values mean fewer detected smeared points with higher confidence. A naive quick equivalent implementation of φ is applying a sliding window in the depth map. No reference points around the detected smeared point in a larger window size mean higher φ. In our experiment, the sliding window with size 3×3 is used to filter unconfident self-annotated smeared labels in See-through Empty.

Multi-Viewpoints $v_f = 1$	See-Through Behind $b_f = 1$	See-Through Empty $e_f = 1$	Inference
✓	–	–	Valid
–	✓	–	Smeared
–	–	✓	Smeared
–	–	–	Unknown

Table 1. Find valid and smeared points using the multi-view consistency and ray-tracing model respectively.

3.4. Geometric Label Generation

Automated pixel annotation involves combining the geometric evidence for valid and smeared points to a sequence of depth images. We note that pixels for which none of the two pieces of evidence apply will have an unknown categorization. To convert geometric evidence among multiple frames to geometric labels trained for the network, we assume that a depth sensor is moved around a rigid scene, typically by hand, and gathers depth frames $\{d_{f-m//2}, \cdots, d_{f+m//2}\}$ from totally $m+1$ consecutive viewpoints, and from which 3D point clouds $\{p_{f-m//2}, \cdots, p_{f+m//2}\}$ are created. Then the first step is to align all viewpoints, which is achieved by multi-frame Iterative Closest Point (ICP) [11]. The result of this alignment is an array of sensor viewpoints and a single-point cloud with each point having a viewing ray to the sensor from which it was gathered. To determine the point visibility we use ray-tracing through rendering as described next.

Pixel Rendering Applying our geometric evidence requires visibility reasoning for all pixels, which is performed using rendering. We denote a pixel observed in frame f as p_f with coordinates (u_f, v_f) and depth d_f. Since we know all camera poses, the pixel can be projected into any other frame f', represented as $p_f^{(f')}$ with coordinates $(u_f^{(f')}, v_f^{(f')})$ and depth $d_f^{(f')}$. This defines a mapping from original pixel coordinates to coordinates in any other camera:

$$I : (u_f, v_f) \rightarrow (u_f^{(f')}, v_f^{(f')}) \qquad (2)$$

Additionally, due to different parameter settings and depth-buffering mechanisms between our renderer and the actual depth sensor, point cloud $p_{f'}$ should also be reprojected to the depth map $d_{f'}^{(f')}$ with the same renderer of $d_f^{(f')}$ when applying our geometry evidence.

The geometric evidence can be gathered into three binary variables for each pixel $\{v_f, b_f, e_f\}$ with each taking values $[0, 1]$. Here $v_f = 1$ indicates valid pixel evidence as it is viewed in multiple frames as in Fig. 4a, while $b_f = 1$ indicates smeared pixel evidence due to See-Through Behind in Fig. 4b, and $e_f = 1$ indicates smeared pixel evidence due to See-Through Empty as in Fig. 4c. These are summarized in Tab. 1. Then, our algorithm to use this evidence to label pixels is shown in Algorithm 1.

Algorithm 1 Algorithm to automatically generate geometric labels for each pixel. Small constants ϵ and δ are set according to pixel depth noise.

```
For target frame $f$, initialize: $b_f, e_f, v_f = 0$
for each $(u_f, v_f)$ in $d_f$ do
    for each $f' \in [f - m//2, f - 1]$ and $[f + 1, f + m//2]$ do
        Rendering new maps $d_f^{(f')}$ and $d_{f'}^{(f')}$
        Index buffer $I : (u_f, v_f) \rightarrow (u_f^{(f')}, v_f^{(f')})$
        if $(u_f^{(f')}, v_f^{(f')})$ is inside frame $f'$ then
            $k = d_f^{(f')} - d_{f'}^{f'}$
            if $|k| < \epsilon$ then
                $v_f(u_f, v_f) = 1$                ▷ Valid pixel
            else if $k < -\delta$ then
                $b_f(u_f, v_f) = 1$                ▷ See-Through Behind
            else if $k = d_f^{(f')}$ then
                $e_f(u_f, v_f) = 1$                ▷ See-Through Empty
            else
                continue;                          ▷ Unknown category
```

In this algorithm, pixels observed in a target frame f are labeled as valid or smeared by doing a pairwise comparison of rendered depths, $(d_f^{(f')}, d_{f'}^{(f')})$, in each of the other reference frames, f'. The number of used reference frames per sequence, m, can be varied, although here we used **m=4** which enabled good multi-frame alignment.

Why train a model rather than directly applying such heuristics? We note that while a multi-frame annotation can be used on its own to remove smeared points, it leaves a significant fraction of points unlabeled (**85**% in our AzureKinect training sets). Relying on this also requires static frames and camera motion, and creates latency. Thus, we use the annotation to train a single-frame network to do the eventual smeared point detection.

3.5. Depth Normals

We anticipate that surface normals will provide useful cues to pixel classification. In particular, smeared pixel normals are often orthogonal to the viewing ray. Surface normals can be computed efficiently and directly from depth maps [33]. We will specify the normal vector $n(u, v)$ at a pixel location (u, v) in the depth map d. This normal can be specified as the perpendicular to a facet connecting the 3D pixel $p(u, v)$, and its neighbor pixel location. Following [47], we define $\omega(u, v)$ to be the inner product of the viewing ray unit vector and the normal unit vector:

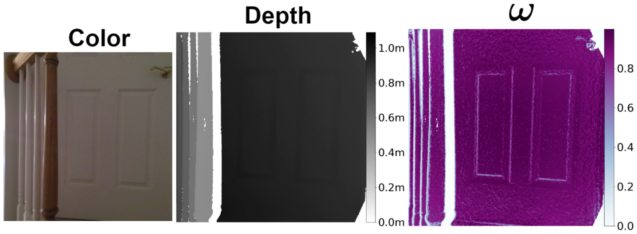


Figure 5. Visualization of the normal view on an indoor scene and values of the boundary are lower compared to non-boundary areas. NOTE: Missing values in the corners of the depth map are directly related to the field-of-view(FoV) [28] of the depth camera.

$$\omega(u, v) = n(u, v)^T \frac{p(u, v)}{||p(u, v||} \tag{3}$$

As shown in Fig. 5, an ω of 1 indicates a surface perpendicular to the viewing ray, while an ω of 0 indicates an orthogonal surface.

3.6. Smeared Classifier and Loss Function

Some off-the-shelf 2D-based segmentation network is adapted here as our smeared classifier rather than a 3D segmentation backbone for three considerations: (1) it is lightweight and fast, (2) depth maps are directly obtained by the sensor when processing raw IR map, and (3) the smeared points generally deviate along the viewing ray, i.e. z-axis which indicates using a z-buffer is sufficient. Our smeared classifier Ψ maps an input $\phi = \{d, \omega\}$ consisting of a depth map and corresponding ray inner products, to an output consisting of the smeared probability p as:

$$\Psi : \phi \rightarrow p \tag{4}$$

We use a binary cross-entropy loss function with the above self-generated geometric labels:

$$CE = -(b + e) \cdot \log p - v \cdot \log(1 - p) \tag{5}$$

To balance both smeared and valid points, weights based on geometric label results are used here as Eq. (6)

$$w_k = 1 - \frac{\|k\|_0}{\|v\|_0 + \|b\|_0 + \|e\|_0}, k \in \{b, e, v\} \tag{6}$$

Besides, the confidence score c for the valid label is also considered to improve robustness as Eq. (7)

$$L = -\alpha \cdot (w_b b + w_e e) \log p - \beta \cdot c w_v v \log (1 - p) \tag{7}$$

In the above final loss equation Eq. (7), α and β are two hyper-parameters for fine-tuning in experiment sections.

Dataset	Type	GT	Size	Resolution	Pose
S1 [3]	Syn	Yes	54	320 × 240	No
S2 [1]	Real	No	96	320 × 239	No
S3 [1]	Real	Yes	8	320 × 239	No
S4 [1]	Real	Yes	8	320 × 239	No
S5 [3]	Real	Yes	8	320 × 239	No
FLAT [19]	Syn	Yes	1200	424 × 512	No
Cornell-Box [38]	Syn	Yes	21300	600 × 600	No
Zaragoza [32]	Syn	Yes	1050	256 × 256	No
AzureKinect	Real	No	1920	1920 × 1080*	Yes
AzureKinect(GT)	Real	Yes	11	1920 × 1080*	No

Table 2. Properties comparisons of related datasets. GT refers to Ground Truth, while the size is the total number of frames. *AzureKinect dataset provides pairs of color and depth maps sharing the same resolution 1920 × 1080, also with raw depth map (640 × 576 resolution) provided.

4. AzureKinect Dataset

To validate the effectiveness of our methods, the real scene datasets using Azure Kinect were collected: we captured a total of 50 indoor and outdoor scenes using the Azure Kinect sensor, one of the state-of-the-art consumer-level cameras in the market. For each scene, we shoot 5 to 10 seconds with the hand-held camera moving without any speed or direction restraint under 5HZ operation frequency. And then a total of 1936 pairs of depth and color frames of real scenes are captured. Like some published datasets such as NYU Depth V2 [34], AVD [4], GMU Kitchen [18], etc, our dataset provides pairs of color and depth information sharing the same resolution(1920 × 1080), as shown in Tab. 2, by transforming depth image to the color camera and doesn't hurt raw frame contents. And we also provide raw depth maps with resolution 640 × 576. Since there are currently no depth sensors on the market that can effectively avoid smeared points, we resort to manually annotating 11 typical frames for 11 different scenes respectively to get ground truth. To ensure the accuracy of the annotation, human annotators are required to carefully observe the whole video clip for each test scene and modify GT labels several times repeatedly, which results in a single depth frame costing a human annotator about 6 hours. To our knowledge, our AzureKinect dataset exceeds existing published real ToF datasets in both size and resolution, see Tab. 2, and is the only dataset provided with pose information for different views of the same scene. Therefore, our dataset lays a good foundation for future work on this new problem though the test set is admittedly small in size.

5. Experiments

Deep learning models from similar tasks: multi-path interference removal (DeepToF [32]), image semantic segmentation (UNet [36], DeepLabV3+ [10], Segformer [48]), are used as the removal backbones based on our self-annotated framework. The self-annotated method DeepDD [42] for removing regular point cloud noises is adapted to this task by replacing pre-calibrated 4 cameras with every 4 consecutive frames with known pre-computed poses. Besides, 5 × 5 median filter based on the depth map and statistical filter [6] based on point cloud are also included in our experiments. We evaluate those models and methods based on the Mean Average Precision where the smeared class is considered positive and the valid point is set as negative. For qualitative comparisons different from others, the predicted results are converted to the point cloud using an intrinsic matrix where smeared points are colored red while the valid points are colored green.

Implementation Details: As mentioned, the geometric labels are first built when joining the off-the-shelf semantic segmentation network. A softmax layer is added to adapt to our segmentation task and we use ResNet-34 [20] as the backbone for UNet [36], DeepLabV3+ [10], Segformer [48]. All codes are implemented by Pytorch and all input frames and labels are cropped and resampled to 512 × 512 for computational needs by using nearest-neighbor interpolation to avoid creating artifacts. Augmentation is performed through random cropping to 128 × 128 with random rotation. We use the mini-batch Adam [25] optimization algorithm, with a weight decay 1e-7, and run 200 epochs with a batch size 32. The initial learning rate is set at 1e-4 and reduced by 10 times after every 25 epochs with a 100-step cosine annealing schedule [31]. We set $\alpha = 0.3, \beta = 0.7, \epsilon = 4mm, \delta = 15mm$ in our experiments. The used adjacent reference frame number is $m = 4$.

5.1. Quantitative and Qualitative Results

Method	Inputs	Features	mAP
Median Filter	d	Hand-crafted	0.231
Statistical Filter [6]	p	Hand-crafted	0.407
DeepDD [42]	(d, ω)	self-annotated	0.103
DeepToF [32]	(d, ω)	self-annotated	0.742
DeepLabV3+ [10]	(d, ω)	self-annotated	0.766
Segformer [48]	(d, ω)	self-annotated	0.729
UNet [36]	(d, ω)	self-annotated	**0.775**
*UNet [36]	(d, ω)	self-annotated	0.771

Table 3. Results of various methods on our AzureKinect datasets. Each row reports the mean average precision of the smeared points with ground truth. * denotes uniform weighting ($c = 1$) in Eq. (7).

To obtain pose information, multiview ICP [40] with five-neighboring point clouds automatically aligns points

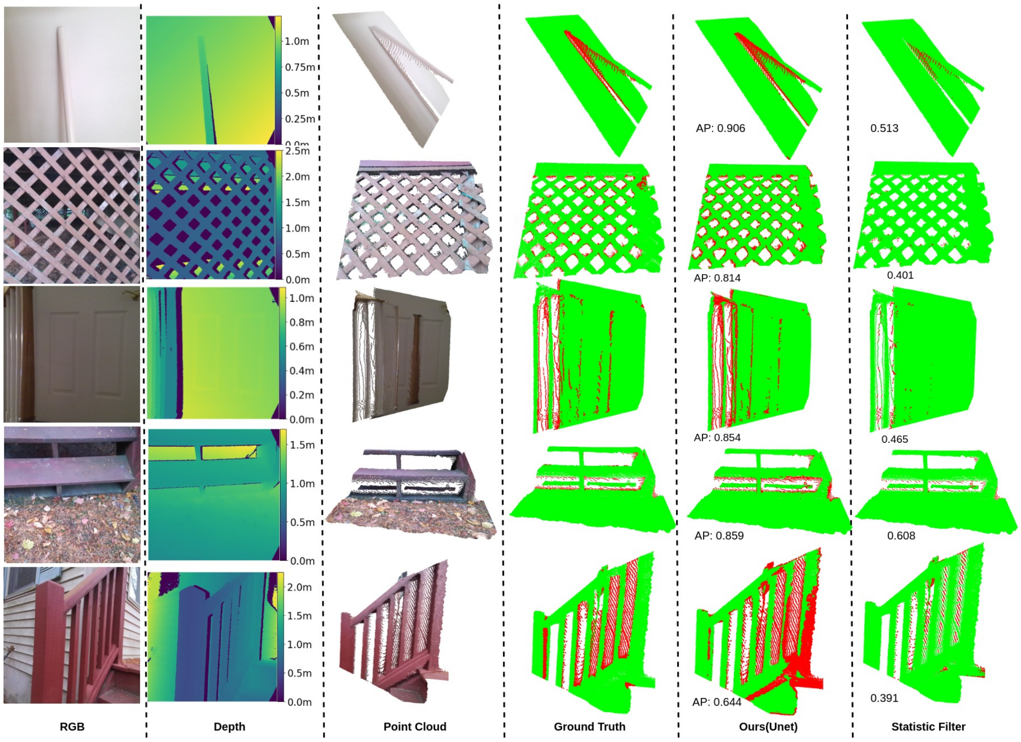


Figure 6. Predicted results with AP on our AzureKinect dataset of our self-annotated learning method using UNet and statistical filter. Smeared points are colored red while valid points are colored green. The areas without any point are masked in white.

and determines camera poses. For the DeepDD [42] model which is a regression model compared to our segmentation task, we apply the threshold standard to get evaluation scores by computing abstract differences between the restored depth and raw depth. If the difference is smaller than the Azure Kinect's systematic error threshold ($11mm + 0.1\%d$), then the depth pixel location is predicted valid, otherwise (larger than that threshold) smeared. Five cases of the test dataset are shown in Fig. 6, where the self-annotated UNet can detect most of the smeared points than the statistical filter though more valid points are misclassified as the distance increase and it is also challenging for a deep learning remover to detect these smeared points which share the similar structures as valid points, observed in the last row of Fig. 6. We evaluate 11 different depth maps from 11 different scenes, where the model using UNet achieves the highest mAP compared to other methods, see Tab. 3. Besides, using uniform weighting ($c = 1$) for multi-view annotation reduces the mAP by 4% than our confidence score design in Eq. (1). The failure of the self-supervised method DeepDD [42] is also noticed in our experiment, where both the consecutive frames with close viewings, and similar color information among the same observed structures impede this method's effectiveness (please refer to our supplementary materials for more qualitative analysis).

5.2. Ablation

To identify the optimal number of consecutive reference frames required, we repeat experiments with different self-annotated labels for partial points, each derived from different numbers of reference frames. We also generate such labels for the test set to ascertain the accuracy of our geometry annotation. Both evaluations on multi-frame geometric classification and our single-frame trained classification are concluded as in Fig. 7. Geometry labels for partial points exhibit $12\% - 15\%$ **higher** mAP than UNet for all points, affirming the precision of our self-annotated labels for partial points. Moreover, using more frames doesn't feed better labels back since the pose estimation is less accurate for long-distance frames and the contradictory information from different frames stands out which further prevents predicted improvements when using more frames.

To validate our selection for input modality ϕ, we replace our remover's input with multiple different combinations of color, depth, and normal-view map ω and evaluate it after 100 training epochs (all convergence guaranteed). For a fair comparison, we conduct a hyperparameter search for each kind of input modality ϕ and report results in Tab. 4 which show that the ω map helps detect smeared points both for depth map and color map with a large increase. Besides, indicated by the drop in performance, we think color im-

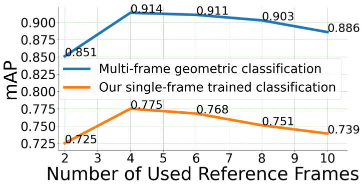


Figure 7. Results of geometric labels generated from different numbers of nearby frames. We report the mean average precision on the AzureKinect data both for multi-frame geometric classification and our single-frame trained classification.

Depth	Color	ω	AP
✓	–	✓	**0.775**
✓	–	–	0.670
–	✓	–	0.567
–	✓	✓	0.629
✓	✓	–	0.613
✓	✓	✓	0.691

Table 4. Results of UNet (ours) with different input types. We report the Average Precision (AP) on the AzureKinect data after hyperparameter optimization.

ages contain some invalid information from similar visual features and produce disturbances.

To validate our choice for the sliding window size $\varphi = 3 \times 3$ in reducing unconfident self-annotated smeared labels in See-Through Empty, different kernel sizes are applied as shown in Fig. 8 for the qualitative comparisons. When $\varphi = 1 \times 1$, it is equivalent not to filter any self-annotated smeared points from See-through Empty. Both 3×3 and 5×5 effectively avoid some misclassifications, but the sliding window with size 3×3 can keep more confident smeared labels than that of 5×5. With $\varphi > 5 \times 5$, too few smeared points are expected to be detected. Therefore, our selection for the sliding window is based on a trade-off assessment of self-annotated label quality and quantity.

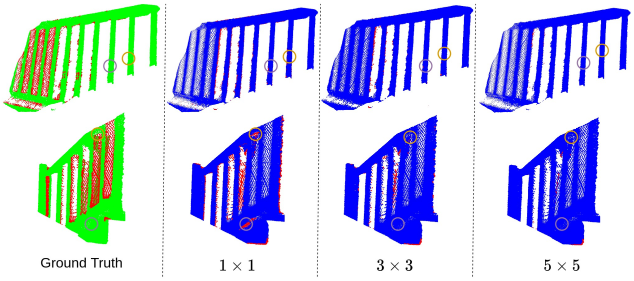


Figure 8. Qualitative comparison among different sliding window sizes for reducing unconfident labels from See-through Empty. The remaining smeared points are colored red with left blue. Misclassifications are reduced and can be seen in small circles.

5.3. Application: 3D Reconstruction

It is always a major challenge to reconstruct objects with sophisticated fine-grained structures using consumer-level cameras. A related experiment in Fig. 9 aligns 15 consecutive frames under the 5HZ work frequency of an Azure Kinect depth sensor and uses down-sampling to make a number of point clouds consistent with three different preprocesses: without any filtering, adding a statistic outliers filter, or using trained UNet model as a preprocessor. Qualitative result in Fig. 9 shows that our trained removal better helps align and keep high-fidelity 3D point clouds relieved of smeared points when placed as a preprocessor.

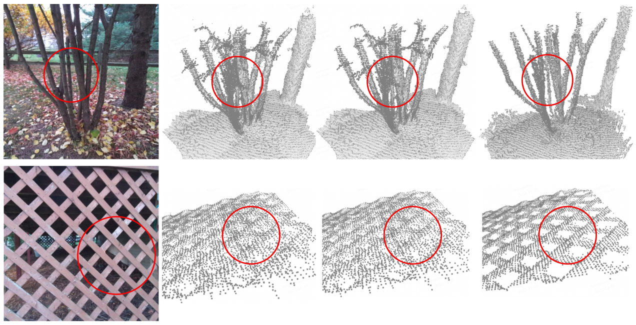


Figure 9. Results of multiple frames alignments using the trained network. From the left column to the right, the second column is the aligned point cloud without any filtering; the third column is the aligned point cloud adding an outlier filter; the last column is using our network as a preprocessor for the raw depth map.

5.4. Limitations

Our pipeline still has several limitations. First, the scenes for training, although not inference, must be static which reduces our data selection especially outside. Second, mechanisms encouraging models to connect and attribute predictions among similar 3D geometry structures need to be further investigated since self-annotated labels are partial and not enough. Finally, incorrect pose estimation due to smeared points can lead to errors. An experiment is performed, where we repeat pose estimation again only using detected valid points (from our initially trained filter), regenerate pseudo-labels, and then retrain our remover from scratch. Results show APs of generated pseudo labels for partial points and predicted scores for all points are raised by 1.5% and 0.8% respectively.

6. Conclusion

In this work, we present a new self-annotated architecture to detect smeared points and then remove this harmful artifact from consumer depth sensors. Visibility-based evidence is automatically gathered from multiple viewpoints of a hand-held sensor to annotate depth pixels as smeared valid or unknown. These annotations are used to train our smeared point detector with no need for manual supervision. Being self-annotated avoids the need for costly human annotation while enabling simple data collection and training of widely varied scenes. As a low-computational network, it can be used as a preprocessor for every single raw frame to improve the quality of 3D reconstruction.

References

[1] Gianluca Agresti, Henrik Schaefer, Piergiorgio Sartor, and Pietro Zanuttigh. Unsupervised domain adaptation for tof data denoising with adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 6

[2] Gianluca Agresti, Henrik Schaefer, Piergiorgio Sartor, and Pietro Zanuttigh. Unsupervised domain adaptation for tof data denoising with adversarial learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5586, 2019. 3

[3] Gianluca Agresti and Pietro Zanuttigh. Deep learning for multi-path error removal in tof sensors. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018. 2, 6

[4] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Kosecka, and Alexander C. Berg. A dataset for developing and benchmarking active vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 6

[5] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987. 1

[6] Haris Balta, Jasmin Velagic, Walter Bosschaerts, Geert De Cubber, and Bruno Siciliano. Fast statistical outlier removal based method for large 3d point clouds of outdoor environments. *IFAC-PapersOnLine*, 51(22):348–353, 2018. 6

[7] Ayush Bhandari, Micha Feigin, Shahram Izadi, Christoph Rhemann, Mirko Schmidt, and Ramesh Raskar. Resolving multipath interference in kinect: An inverse problem approach. In *SENSORS, 2014 IEEE*, pages 614–617. IEEE, 2014. 2

[8] Ayush Bhandari, Achuta Kadambi, Refael Whyte, Christopher Barsi, Micha Feigin, Adrian Dorrington, and Ramesh Raskar. Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization. *Opt. Lett.*, 39(6):1705–1708, Mar 2014. 2

[9] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, may 2000. 2

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 6

[11] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565, 2015. 4

[12] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2

[13] G. Deng and L.W. Cahill. An adaptive gaussian filter for noise reduction and edge detection. In *1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*, pages 1615–1619 vol.3, 1993. 2

[14] Suyog D. Deshpande, Meng Hwa Er, Ronda Venkateswarlu, and Philip Chan. Max-mean and max-median filters for detection of small targets. In Oliver E. Drummond, editor, *Signal and Data Processing of Small Targets 1999*, volume 3809, pages 74 – 83. International Society for Optics and Photonics, SPIE, 1999. 2

[15] AM Fahim, G Saake, AM Salem, FA Torkey, and MA Ramadan. Dcbor: a density clustering based on outlier removal. *International Journal of Computer and Information Engineering*, 2(9):2917–2922, 2008. 2

[16] Lei Fan, Yunxuan Li, Chen Jiang, and Ying Wu. Unsupervised depth completion and denoising for rgb-d sensors. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8734–8740, 2022. 3

[17] Péter Fankhauser, Michael Bloesch, Diego Rodriguez, Ralf Kaestner, Marco Hutter, and Roland Siegwart. Kinect v2 for mobile robot navigation: Evaluation and modeling. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 388–394, 2015. 1

[18] Georgios Georgakis, Md Alimoor Reza, Arsalan Mousavian, Phi-Hung Le, and Jana Košecká. Multiview rgb-d dataset for object instance detection. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 426–434. IEEE, 2016. 6

[19] Qi Guo, Iuri Frosio, Orazio Gallo, Todd Zickler, and Jan Kautz. Tackling 3d tof artifacts through learning and the flat dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 368–383, 2018. 6

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[21] P. Hermosilla, Ritschel, and T. Ropinski. Total denoising: Unsupervised learning of 3d point cloud cleaning. *International Conference on Computer Vision 2019 (ICCV19)*, 2019. 3

[22] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 559–568, New York, NY, USA, 2011. Association for Computing Machinery. 1

[23] Adrian Jarabo, Julio Marco, Adolfo Muñoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. A framework for transient rendering. *ACM Transactions on Graphics (SIGGRAPH Asia 2014)*, 33(6), 2014. 2

[24] Achuta Kadambi, Refael Whyte, Ayush Bhandari, Lee Streeter, Christopher Barsi, Adrian Dorrington, and Ramesh Raskar. Coded time of flight cameras: Sparse deconvolution to address multipath interference and recover time profiles. *ACM Trans. Graph.*, 32(6), nov 2013. 2

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun,

editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6

[26] Kalin Kolev, Petri Tanskanen, Pablo Speciale, and Marc Pollefeys. Turning mobile phones into 3d scanners. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[27] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5468–5479. PMLR, 13–18 Jul 2020. 3

[28] Gregorij Kurillo, Evan Hemingway, Mu-Lin Cheng, and Louis Cheng. Evaluating the accuracy of the azure kinect and kinect v2. *Sensors*, 22(7):2469, 2022. 5

[29] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013. 3

[30] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021. 3

[31] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 6

[32] Julio Marco, Quercus Hernandez, Adolfo Muñoz, Yue Dong, Adrian Jarabo, Min H. Kim, Xin Tong, and Diego Gutierrez. Deeptof: Off-the-shelf real-time correction of multipath interference in time-of-flight imaging. *ACM Trans. Graph.*, 36(6), nov 2017. 2, 3, 6

[33] Yosuke Nakagawa, Hideaki Uchiyama, Hajime Nagahara, and Rin-Ichiro Taniguchi. Estimating surface normals with depth image gradients for fast and accurate registration. In *2015 International Conference on 3D Vision*, pages 640–647, 2015. 5

[34] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2, 6

[35] Chuong V. Nguyen, Shahram Izadi, and David Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 524–530, 2012. 1

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 6

[37] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In *ECCV*, 2022. 2

[38] Michael Schelling, Pedro Hermosilla, and Timo Ropinski. RADU - ray-aligned depth update convolutions for ToF data denoising. In *Conference on Computer Vision and Patter Recognition (CVPR)*, 2022. 2, 3, 6

[39] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2007. 1

[40] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009. 6

[41] Soheil Sotoodeh. Outlier detection in laser scanner point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 297–302, 2006. 2

[42] Vladimiros Sterzentsenko, Leonidas Saroglou, Anargyros Chatzitofis, Spyridon Thermos, Nikolaos Zioulis, Alexandros Doumanoglou, Dimitrios Zarpalas, and Petros Daras. Self-supervised deep depth denoising. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1242–1251, 2019. 2, 3, 6, 7

[43] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, 1998. 2

[44] Michal Tölgyessy, Martin Dekan, Ľuboš Chovanec, and Peter Hubinský. Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2. *Sensors*, 21(2), 2021. 1

[45] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1

[46] Katja Wolff, Changil Kim, Henning Zimmer, Christopher Schroers, Mario Botsch, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. Point cloud noise and outlier removal for image-based 3d reconstruction. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 118–127, 2016. 2

[47] Katja Wolff, Changil Kim, Henning Zimmer, Christopher Schroers, Mario Botsch, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. Point cloud noise and outlier removal for image-based 3d reconstruction. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 118–127, 2016. 5

[48] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 6