# MilliPCD: Beyond Traditional Vision Indoor Point Cloud Generation via Handheld Millimeter-Wave Devices

PINGPING CAI, University of South Carolina, USA
SANJIB SUR, University of South Carolina, USA

3D Point Cloud Data (PCD) has been used in many research and commercial applications widely, such as autonomous driving, robotics, and VR/AR. But existing PCD generation systems based on RGB-D and LiDARs require robust lighting and an unobstructed field of view of the target scenes. So, they may not work properly under challenging environmental conditions. Recently, millimeter-wave (mmWave) based imaging systems have raised considerable interest due to their ability to work in dark environments. But the resolution and quality of the PCD from these mmWave imaging systems are very poor. To improve the quality of PCD, we design and implement *MilliPCD*, a "beyond traditional vision" PCD generation system for handheld mmWave devices, by integrating traditional signal processing with advanced deep learning based algorithms. We evaluate *MilliPCD* with real mmWave reflected signals collected from large, diverse indoor environments, and the results show improvements in the quality *w.r.t.* the existing algorithms, both quantitatively and qualitatively.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**; • **Computing methodologies → Machine learning approaches**.

Additional Key Words and Phrases: Point Cloud Data, Graph Neural Networks, Millimeter-Wave, Wireless Sensing.

**ACM Reference Format:**
Pingping Cai and Sanjib Sur. 2022. MilliPCD: Beyond Traditional Vision Indoor Point Cloud Generation via Handheld Millimeter-Wave Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 160 (December 2022), 24 pages. https://doi.org/10.1145/3569497

## 1 INTRODUCTION

With the rapid development of applications in robotics [1], autonomous driving [2], drones [3], and virtual or augmented reality (VR/AR) [4], there is an increasing demand for robot systems to understand the 3D environment. To enable this capability, existing systems use many sensors and algorithms to facilitate devices acquiring and interpreting the surrounding 3D information. Besides, to help devices store and process the 3D information, researchers have developed multiple data structures for representation, *e.g.*, Meshes [5], Voxels [6], and Point Clouds [7]. Among them, Point Cloud Data (PCD) is one of the efficient data structures, which can be generated by LiDARs or RGB and Depth (RGB-D) sensors. These sensors can measure the depth of the target scene, which can be used to reconstruct the relative 3D position of the objects and generate a PCD. Although these systems have matured and are used in many research and commercial settings [1, 2, 4, 8, 9], their performance is still limited under challenging environmental conditions. For example, RGB-D sensors rely on good ambient lighting; otherwise, their optical camera might not work in a dark environment. While LiDARs can work under poor lighting conditions, they are expensive and cumbersome, and they could often be impaired by the presence of

Authors' addresses: Pingping Cai, pcai@email.sc.edu, University of South Carolina, USA; Sanjib Sur, sur@cse.sc.edu, University of South Carolina, USA.


2474-9567/2022/12-ART160 $15.00
https://doi.org/10.1145/3569497

airborne obscurants [10]. Thus, designing a low-cost but robust PCD generation system that is lightweight and easily portable, like cameras, and can work efficiently even under poor lighting conditions is of vital importance.

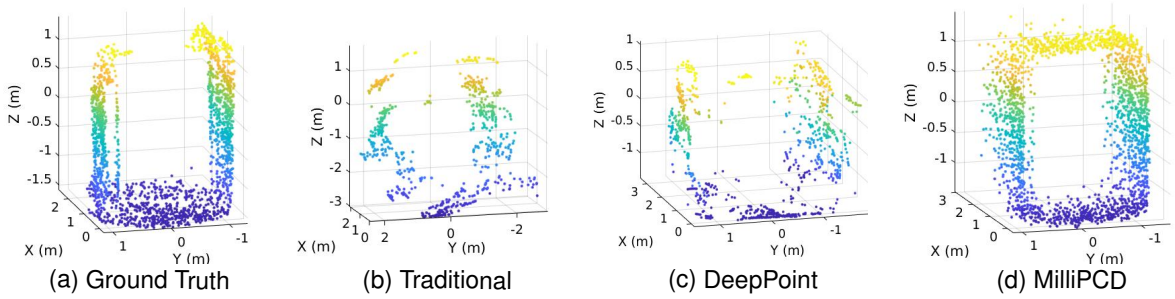


Figure 1. PCD reconstruction results from different methods: (a) Ground truth PCD; (b) PCD from traditional non-coherent mmWave imaging; (c) DeepPoint [11]; and (d) *MilliPCD*. *MilliPCD* outperforms the traditional mmWave imaging and the existing deep learning method, preserving better geometrical structures of the environment.

Fortunately, millimeter-wave (mmWave) wireless sensors provide an alternative to the traditional vision sensors: MmWave signals can penetrate small obstacles and work under poor or zero visibility. So, mmWave imaging can enable "*beyond traditional vision*" applications [12–14] and could allow the devices to "see" the 3D environments under challenging conditions. Besides, mmWave transceivers are lightweight and cheap and will soon become ubiquitous in all 5G-and-beyond smart devices, such as smartphones or wearables. However, generating high-quality PCD from mmWave sensors is still challenging for two key challenges. (1) *Poor Resolution and Specularity*: Due to the limited number of antennas and bandwidth of mmWave devices, the resolution of mmWave imaging is extremely low compared to the RGB-D sensors or LiDARs. Thus, the generated PCD will be extremely sparse with a few points (see Fig. 1b). Besides, only a small amount of transmitted mmWave signals that fall on the normal to the indoor surfaces are specularly reflected towards the receiver. Thus, a significant amount of information on the 3D space cannot be captured by the receiver at a fixed location (*e.g.*, a flat surface like a wall will appear as a single point in the mmWave image). (2) *Noise*: The mmWave transceiver is sensitive to a strong reflector (like a metal cabinet) behind a weak reflector (like drywall); so it will generate noisy points that are non-existent inside an environment, increasing the difficulty of reconstructing the PCD of that environment. Thus, the generated PCD may contain ambiguous shapes of objects with many noisy points (Fig. 1b).

Previous researchers aimed to solve some of these challenges, and the approaches can be categorized into two types: (1) traditional signal processing [12, 15, 16] and (2) deep learning [11, 17]. To improve mmWave imaging resolution, traditional approaches use a very large physical antenna array [18, 19] or use SAR imaging technique to virtually form a large antenna array [12, 20, 21]. However, incorporating a large antenna array on handheld devices is challenging. Besides, to form the virtual antenna in SAR imaging, the transceiver has to move along straight lines with a fixed speed and view direction [12]. Such constraints are difficult to enforce in practice with handheld devices, since users may walk randomly at varying speeds and directions. Besides, to generate the PCD of a large indoor environment, it is necessary to combine the mmWave signals from multiple viewpoints. But traditional approaches reconstruct the reflected points independently and fail to explore their inherent relationships. What's more, as the mmWave transceiver is sensitive to strong reflectors behind a weak reflector, a traditional approach may generate noisy points that are not present within the environment.

Deep learning based approaches take advantage of neural networks to learn a complex mapping from mmWave signals to geometrical shapes. For example, [17] proposed a deep-learning based model for 3D PCD generation of a single object from mmWave signals. The approach takes input as a 3D Radar intensity map scanned from 4

specific viewpoints to generate 4 2D depth images, which are then fed into a generator to output a 3D PCD. But this work is limited to reconstructing PCD of single objects, such as a car, and requires fixed viewpoints to collect reflected signals from objects, which is unsuitable for reconstructing large indoor environments with free-hand scanning. DeepPoint [11] introduces a better PCD reconstruction system that builds atop Generative Adversarial Network (GAN) [22], which takes a coarse PCD as input and outputs a refined PCD. However, DeepPoint uses MLP based encoder and decoder and can not capture the local geometry of neighbor points. As a result, points in generated PCD may not distribute uniformly in 3D space (see Fig. 1c). Besides, the reflected mmWave signals from indoor environments are more complex due to the widely different building structures and different scanning paths, and thus, previous deep learning methods cannot be applied to indoor scenarios.

*We propose MilliPCD, a system to generate high-quality PCD using mmWave devices via deep neural networks.* Our method does not rely on the user scanning along a fixed straight line trajectory or use fixed viewpoints, and is able to overcome the fundamental challenges of poor resolution, specularity, and noisy points in traditional imaging. *MilliPCD*'s key idea is intuitive: Since a mmWave transceiver can measure the distance of reflecting points from one viewpoint, we can identify the general structure of an environment by measuring and combining the reflections from multiple viewpoints; however, such structures will be sparse and noisy. *MilliPCD* then designs a customized deep graph neural network to learn the relationship between the true geometrical shape and the general structure from hundreds of data samples, and at run-time, generates accurate and denser PCD. To this end, *MilliPCD* designs two modules: *Reflection Points Generation* and *Noise-Aware Shape Reconstruction. First*, *MilliPCD* combines the received mmWave signals from different random scan points using the traditional Backprojection algorithm [23] and generates a candidate incomplete PCD with coarse and noisy points, containing potential ambiguous shapes. *Then*, *MilliPCD* uses a customized deep neural network to learn the association between the candidate noisy PCD and ground truth PCD. The customized network is built atop a Dynamic Graph Convolution Neural Network (DGCNN) [24] that predicts a confidence score for each point in the noisy PCD, a PointNet++ [25] block that takes both confidence scores and candidate points to extract a global *shape code* that encodes the global structure of the environment, and a generation network with Seed Generator [26] and Point Upsampling blocks that regenerate high-quality PCD from global shape codes. Mobile users or robots can then freely roam in the indoor environment, collecting mmWave reflection signals from arbitrary viewpoints, and *MilliPCD* can reconstruct the accurate shape of the environment, even under poor lighting or dark conditions.

We design and prototype *MilliPCD* on a Commercial-Off-The-Shelf (COTS) mmWave device and conduct microbenchmark experiments for indoor PCD. Since current COTS mmWave networking devices do not provide user access to the raw signal reflections yet, we built a customized setup using a 60 GHz mmWave transceiver [27] and an ASUS ZenFone AR [28]. During the data collection process, we walk inside the indoor building by holding the device to collect the reflected mmWave signals, ground truth PCD, and local poses of the device simultaneously. We collect data across 13 large indoor spaces that generate 37 unique PCD, and our dataset consists of nearly 17.3 GB of samples. We further augment the number of data samples with different rotations and translations and split them into small patches. In total, we have 1100 training samples and 165 testing samples. We compare the effectiveness of *MilliPCD w.r.t.* traditional imaging and existing deep learning methods, and *MilliPCD* can successfully reconstruct the structure of indoor environments with good quality. The testing result shows that *MilliPCD* can achieve a median L1 Chamfer Distance (ChD) [29] and Earth Mover's Distance (EMD) [30] of 0.256 m and 0.496 m, respectively, outperforming both traditional and existing deep learning methods (compare Fig. 1(b-d) for a visual result). Furthermore, *MilliPCD* can consistently generate PCD with different densities of points but still preserve good geometric shapes, and is robust under limited scan times, and has low memory and computational footprints for PCD generation at run-time.

In summary, we have two contributions: (1) We design a handheld system that allows users to collect the mmWave reflections and generate PCD of an indoor environment easily. (2) We design novel deep-learning frameworks to

facilitate the generation of high-quality indoor PCD from the mmWave reflected signals. *MilliPCD* is generalizable to many diverse indoor environments and works under challenging environmental conditions. For the purpose of reproducing our approach and catalyzing the "beyond traditional vision" research with mmWave, we have released our codebase and the datasets through our project repository [31].

## 2 RELATED WORK

### 2.1 Millimeter-Wave Imaging for Indoor Mapping

Traditional mmWave imaging systems generate high resolution results using either a large physical antenna array or a virtual antenna array emulated through device movements in constrained paths [20, 32–38]. A key advantage of mmWave devices is that they can work under adverse environmental conditions, such as smoke, fog, and dust. Previous researchers have used this advantage in generating the indoor floor map under challenging conditions [10, 21, 39]. Different from indoor 3D PCD generation, they aim to reconstruct the 2D floor map of an indoor environment. Authors in [39] propose an orthogonal frequency-division multiplexing (OFDM) based radar processing algorithm to generate a 2D grid map of the indoor environment by using a 5G mmWave device. In this system, the mmWave device continuously steers its beam towards different directions and collects the reflections to estimate the distance and draw a coarse map of the surrounding environment. However, the quality of generated floor maps is poor, containing many noise grids, due to the low resolution and specularity of mmWave signals. To increase the resolution, authors in [21] take advantage of multi-view from multiple radars along with the SAR technique to form larger virtual antennas. While, authors in [10] present a deep learning based method, called milliMap, that can reconstruct a dense grid map with accuracy comparable to LiDAR. Especially, they first build a data collection system based on a single-chip mmWave radar, to generate the coarse and noisy indoor grid maps. Then, they overcome the sparsity and specularity noise of mmWave signals by a post-processing system that is built on a generative deep learning algorithm under the supervision of a co-located LiDAR ground truth grid map, to refine the raw and noisy grid maps.

However, these methods are designed for generating 2D indoor floor maps on the X-Y panel, and cannot be applied to generating 3D indoor PCD. First, they only generate reflection points with 2D coordinates and overlook the Z coordinate, which cannot express the 3D geometric information of indoor PCD, such as the height of walls and the position of ceilings. Besides, they require a fixed elevation angle of mmWave devices during data collection. While in our cases, the user can hold the device and collect mmWave signals from any view direction. Thus, we cannot apply their systems to generate 3D indoor PCD.

### 2.2 PCD Generation

Unlike Voxels or Meshes, a PCD can represent an object or environment in 3D with high resolution but has a low memory footprint. Hence, it has been used in many existing research and commercial applications, *e.g.*, simultaneous localization and mapping [40], vehicle detection in autonomous driving [41], 3D object detection and classification [9], *etc.* Existing PCD reconstruction techniques can be classified into two categories: Depth based and Image based. Depth based methods, such as [42–44], first detect the depth information of the target scene from depth sensors, like IR and LiDARs, and then reconstruct the 3D PCD using geometric transformation *w.r.t.* the device poses. Image based methods, such as [45–47], directly use a sequence of images along with the knowledge of the camera's extrinsic and intrinsic parameters to reconstruct the 3D PCD. Extrinsic parameters specify the location and orientation of the camera and intrinsic parameters map pixel coordinates to coordinates in the world frame. Apart from using multiple images, many single image 3D reconstruction methods have been proposed by using deep neural networks [29, 48, 49].

The mmWave based PCD generation algorithms fall under the category of depth based methods, where the device first measures the depth information of the reflecting points and then maps it to the 3D space. But the

resolution of mmWave is very limited, which affects the quality of generated PCDs. Past works aimed to improve traditional signal processing methods to improve the resolution of mmWave images, using techniques like SAR imaging [12, 20] or using a large antenna array [19]. However, SAR imaging requires the devices to move along a straight path with a fixed "aiming direction" to form a large virtual antenna array [20], which is quite challenging for practical handheld systems. Even though these requirements are met, the generated PCD is still very sparse and mostly contains ambiguous shapes of objects [12]. Thus, to optimize the ambiguous PCD, several deep learning based mmWave PCD algorithms have been proposed recently. These algorithms have outperformed traditional algorithms by a great gap, due to their ability to automatically extract features from data samples. Authors in [17] proposed a two-stage PCD generation system, called 3DRIMR, to generate PCD for single objects like cars. In the first stage, they generate 2D depth images from 4 different viewpoints using FMCW radars. Then in the second stage, they reconstruct PCD from these depth images using a deep neural network. However, they only focus on a single object and require fixed viewpoints, which can not be applied to our test cases. Apart from this, DeepPoint [11] introduces a better PCD reconstruction system that builds atop generative adversarial networks [22], which takes a coarse PCD as input and outputs a refined PCD. However, DeepPoint uses MLPs based encoder and decoder to extract point-wise features and overlooks the local geometry of nearby points. Besides, the number of output points is exactly the same as input due to the structure of MLPs, and it cannot upsample and generate a denser PCD. To solve these problems, we design a better PCD generation system with graph neural networks to extract local geometric information and upsampling blocks to generate denser PCD.

## 3 BACKGROUND AND CHALLENGES

### 3.1 Millimeter-Wave Imaging

Traditional mmWave imaging techniques rely on Frequency Modulated Continuous Waves (FMCW) that are transmitted and received by the mmWave transceiver to generate an image [50]. The transceiver periodically transmits FMCW signals, where each pulse linearly sweeps through a certain frequency band (for example, 77 to 81 GHz, where 4 GHz is the signal bandwidth), and receives the signals reflected back from various objects in the surrounding. Since objects at different distances will reflect back signals at slightly different times, the receiver can identify the amplitudes and phases of the signals reflected from the objects in time. By measuring the difference in frequencies between the received and transmitted signal, the transceiver can obtain the signal time of flight (ToF), which then can be translated into object distance [50] (See Fig. 2(a-b)). Furthermore, by leveraging multiple antennas along the vertical and horizon dimensions, the transceiver can locate different objects at relative azimuth and elevation angles by estimating the phase differences of each signal *w.r.t.* a reference antenna. Then, based on the angular and distance information, the strong reflection points can be mapped into the 3D plane to generate a PCD. The output resolution in depth, azimuth, and elevation depends on the device's bandwidth, number of horizontal antennas, and number of vertical antennas, respectively. But common mmWave transceivers, such as those found in handheld or wearable devices, or mobile robots [51–53], have a limited number of antennas, such as 2×4 or 2×8 [1]; so, the resolution of generated images is very low. To improve the resolution, researchers aimed to bring the classical SAR technique [20] to mobile devices [12, 54]. However, it requires the transceiver to move along a pre-defined, rigid path, such as a straight line, to form a large virtual antenna array. Such constrains are not only difficult to enforce in practical, free-hand scanning but also do not improve the resolution beyond the theoretical limits or solve the fundamental challenges.

[1] X×Y represents the number of antennas across vertical and horizontal directions.

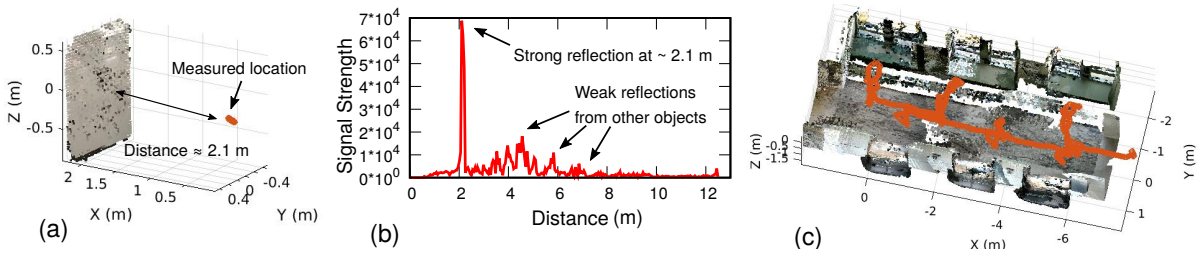


Figure 2. (a) An example of reflection measurement where the mmWave device is at ~2.1 from a solid reflector. (b) The reflection signal profile shows high signal strength at that distance. (c) An example of a real free-hand scanning trajectory in one indoor environment: The trajectory is non-linear, and the environment contains various objects with complex shapes.

### 3.2 Fundamental Challenges with mmWave Point Cloud Generation

Generating a high-quality PCD via a handheld or mobile mmWave device is challenging for three reasons: (1) *Limited, non-linear device movements*: Existing PCD generation systems, such as [12], depend on device movement in a rigid trajectory in uniformly spaced poses to form a virtual antenna array for SAR imaging. But under handheld or mobile settings, this requirement could be challenging to impose. For example, Fig. 2c shows the non-linear trajectory when we collect data with free hand. Furthermore, even if one can enforce such movement requirements for SAR technique, obtaining measurements along the vertical axis is challenging and time consuming, and hence the final output will have an extremely low resolution in elevation angles. (2) *Variable indoor structures and signal specularity*: Typical indoor structures contain various objects, and different objects may reflect mmWave signals differently. For example, a metal cabinet would likely reflect strong signals, and the weak reflections from other objects could be buried under these strong reflections. Besides, due to the small wavelength of the mmWave signal, most objects reflect the signals specularly and deflect the signal away from the device. So, the shapes could be fully reconstructed only when the objects are oriented in parallel to the aperture plane. Such fundamental limitations would lead to a partial, ambiguous shape reconstruction only. Fig. 1b shows an example of such sparse reconstruction of a large indoor environment by the traditional method, where the output PCD contains a handful of points with ambiguous and distorted shapes. (3) *Noisy Points from Strong Reflectors*: Since mmWave signals can penetrate through certain objects, such as drywalls, the receivers can measure strong reflection outside of an environment. For example, a metal cabinet placed outside a room will reflect signals strongly, so the generated PCD will contain these spurious reflecting points, even if the object is non-existent inside the room, increasing the difficulty of reconstructing the PCD of that room.

## 4 SYSTEM DESIGN

### 4.1 Overview

*MilliPCD* aims to reconstruct high-quality indoor PCD from the signals collected by a mobile mmWave device by solving the above challenges. This could enable many new applications in navigation [55], localization [40], and detection [9] in practical environments under challenging conditions, such as low light or dark conditions. *MilliPCD* designs two modules: A *Reflection Points Generator*, which converts the raw mmWave signals and device poses into a set of candidate reflection points, and a *Noise-Aware Shape Reconstructor*, which reconstructs a high-quality PCD from these candidate points. Specifically, the reflection points generator focuses the mmWave signals from multiple poses, identifies potential strong reflection points, and generates a coarse PCD via the Backprojection algorithm [23]. However, due to the low resolution, specularity, weak and multipath reflectivity, the generated PCD is sparse, ambiguous, and noisy; so, it cannot show the accurate shape of the current environment.

*MilliPCD* then designs the noise-aware shape reconstructor that inputs this coarse PCD and reconstructs a high-quality PCD. Considering that the coarse PCD still contains some useful shape information that correlates

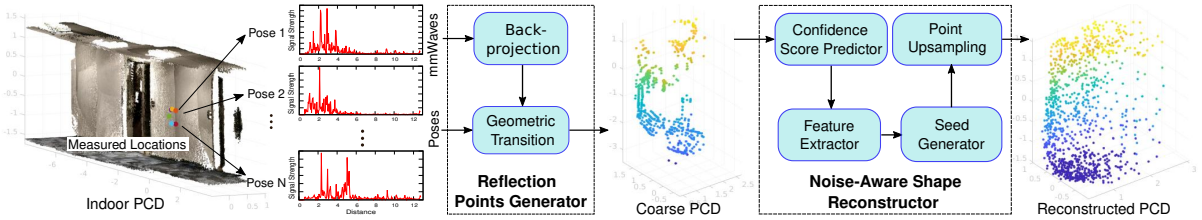


Figure 3. The system pipeline of *MilliPCD*. Given the poses and corresponding mmWave reflected signals, *MilliPCD* first decodes them to generate a coarse-grained PCD using the Reflection Points Generator module and then reconstructs a fine-grained PCD using the Noise-Aware Shape Reconstructor module.

with the true shape of the environment, the shape reconstructor trains a deep learning model with hundreds of data samples and learns the relationship between the coarse PCD and true visual PCD. Then, at run-time, it reconstructs a high-quality PCD without requiring the visual PCD. To train *MilliPCD*'s shape reconstructor, we first collect a diverse dataset containing the mmWave reflections and device poses from different indoor environments and their visual PCD. The collected data are then sanitized, and the device poses are synchronized and resampled in time to align with the mmWave signals. These datasets are then used to train the shape reconstructor, which includes a Confidence Score Predictor to assign a confidence score to each candidate point, a Feature Extractor to extract a shape code of the environment, and a Seed Generator and Upsampling Network to reconstruct the final PCD. Fig. 3 shows the overall system pipeline of *MilliPCD*.

Compared to the traditional mmWave imaging systems, which can only generate coarse and noisy indoor PCD due to the sparsity and specularity of mmWave signals, *MilliPCD* incorporates a deep learning pipeline that improves the quality of the generated PCD. Although DeepPoint [11] also takes a deep learning approach, the generated points in their PCD are poorly distributed due to the use of MLPs for feature extraction. We overcome this drawback by designing a noise-aware shape reconstructor that can identify the noisy points, extract better features, and generate well-distributed points in the PCD.

## 4.2 Reflection Points Generator

*The core objective of this module is to generate coarse points of the environment from the raw mmWave signals collected from multiple device poses.* Given a sequence of mmWave signals with different poses, the reflection points generator decodes the distance information from mmWave signals and infers the global 3D coordinates of reflected points. To this end, the generator first uses a Backprojection, a popular mmWave imaging algorithm for data collection in non-linear trajectories [23], to estimate the local coordinates of reflection points *w.r.t.* the device. Then, it uses a geometric translation to transform the local coordinates into global coordinates.

*4.2.1 Backprojection for Nonlinear Device Trajectory.* Since mmWave device in a free-hand scanning moves in a non-linear trajectory, applying the SAR imaging technique, that require rigid linear movement of the transceiver, is not practical. Instead, we use the Time Domain Backprojection to form the 3D intensity map, since it works reasonably well with highly non-linear trajectories [23]. Then, we decode the map into a coarse point set to generate a local PCD. Algorithm 1 shows the steps. *First*, we create a virtual 3D coordinate grid representing each potential reflection point. *Next*, we simulate the ideal reflection signal that received from each grid, and correlate it with real, measured signals; if they are highly correlated, then it will generate a high-intensity value at this grid. *Then*, we accumulate the correlated signal from all antenna arrays for each grid, form a 3D intensity map for all coordinate grids, and threshold it with a predefined value to select high intensity reflections. *Finally*, we translate these high intensity 3D voxels into coordinates to form a local candidate point set.

**Algorithm 1** Backprojection and Local PCD Generator

**Input**: mmWave reflected signals $S$; carrier wavelength $\lambda$; threshold $\tau$.
**Output**: Local 3D PCD with coarse points, $L_{PCD}$.
Initialize a 3D intensity map $V$.
**for** each voxel $v$ **in** $V$ **do**
  Calculate Euclidean distance $d_n$ from $v$ to each antenna $a_n$.
  Sample real collected signal, $S(a_n, d_n)$ at distance $d_n$ of antenna $a_n$.
  Apply phase and amplitude correction, $S'(a_n, d_n) = S(a_n, d_n)\frac{1}{4\pi d_n^2} exp(\frac{-j4\pi d_n}{\lambda})$.
  Accumulation, $V_v = \sum_{n=1}^{N} S'(a_n, d_n)$, where $N$ is the total number of antennas.
**end for**
For each $V_v > \tau$, convert it to coordinates, and save it to $L_{PCD}$.
**return** $L_{PCD}$

*4.2.2 Geometric Transition.* Since the candidate points are generated in the local coordinate *w.r.t.* the device pose, we translate them into global coordinates to form a global PCD. Given a pose, $[x, y, z, \theta_z, \theta_y, \theta_x]$ for each viewpoint, we transform the local reflection points to their global positions by the following transformation function: $P_{global} = Rot(P_{local}) + [x, y, z]$, where $Rot$ is the 3×3 rotation matrix calculated from the current view angle $[\theta_z, \theta_y, \theta_x]$ [56]. After geometric transformation, we merge all the points together to form the global PCD. However, due to the low resolution, specularity, weak and multipath reflectivity, the generated PCD is sparse, ambiguous, and noisy; so, it may not show the accurate shape of the current environment.

## 4.3 Noise-Aware Shape Reconstructor

*The core objective of the shape reconstruction module is to generate a higher-quality PCD with a better geometric shape from the coarse PCD.* But designing an algorithm to process and extract geometric information from PCD is challenging for three reasons: (1) The algorithm should be robust to an unordered sequence of input points from the PCD; (2) It should be able to extract accurate geometric information from a cluster of points efficiently; and (3) It should be robust to the noise points in input PCD. To solve the first two challenges, we take advantage of graph neural networks that can deal with unordered points efficiently. To solve the third challenge, our intuition is that the performance of neural networks can be improved if we can teach them to identify potential noisy points and avoid learning features from them. The shape reconstruction module is designed based on a customized version of the existing Dynamic Graph Convolution Neural Network (DGCNN) [24] and PointNet++ [25] to learn the noise-robust features from coarse inputs and map them to ground truth PCD. Fig. 4 shows the detailed design of this module, with three major blocks: Confidence Score Predictor, Noise-Aware Shape Feature Extractor, and Seed Generator and Point Upsampling modules, which we describe next.

*4.3.1 Confidence Score Predictor.* Before feeding the reflection points into the feature extractor, the Confidence Score Predictor aims to assign a confidence score to each input point, which encodes the confidence of whether this point is close to the ground truth (*a.k.a.*, not a noise point). Specifically, *a low confidence score means this point is likely a noise point.* By assigning this score, the following reconstruction networks can learn to avoid these noisy points and achieve better performance. However, it is non-trivial to design such a predictor, as the number of reflected points across environments can vary, and these points may be distributed in 3D space unorderly: So, a traditional Convolution Neural Network (CNN) cannot be applied. Traditionally, the PointNet-based network structures are used to process the 3D PCD [25, 57, 58]. They use a Farthest Point Sampling (FPS) strategy to select the cluster of points in the coordinate space to create local graphs, which tend to select the farthest noisy points. Without additional information, such noisy points will have equal importance as clean points, and the network

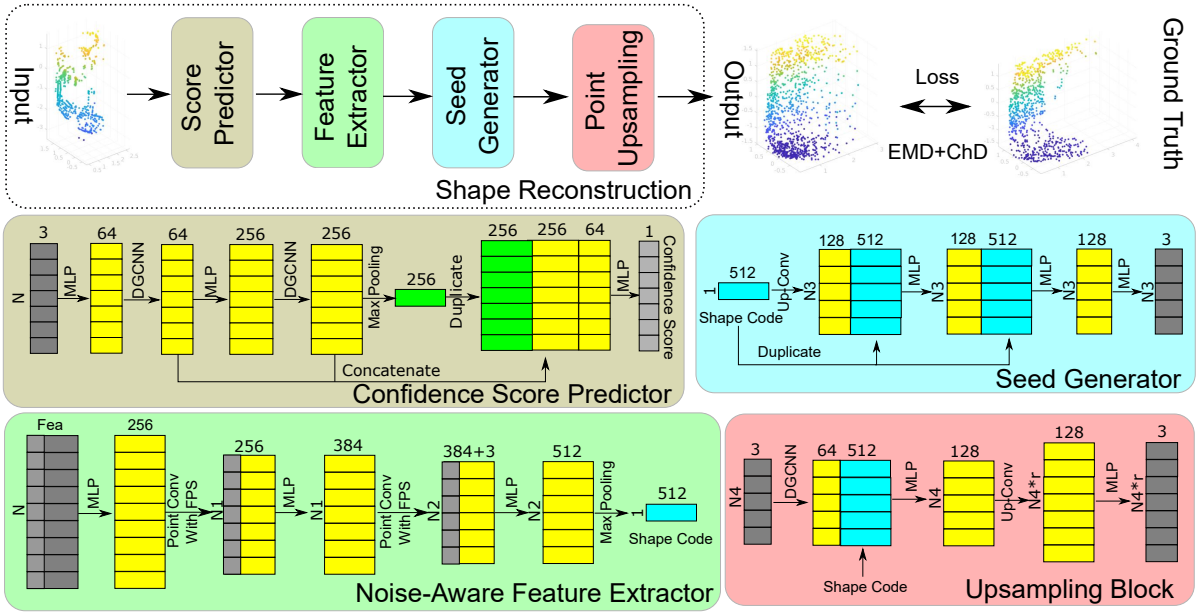


Figure 4. *MilliPCD*'s Noise-Aware Shape Reconstructor Network.

will extract inaccurate shape features. Thus, the PointNet-based structure alone is not good enough to eliminate the noisy points. To this end, we design a score prediction network based on the recently proposed DGCNN. The DGCNN is able to extract multi-scale geometric features as well as local and non-local features for each point to produce richer contextual information. Specifically, the DGCNN layer takes the point feature maps as input, dynamically constructs a graph using their $k$ nearest neighbors in the feature space with the aggregation operator *Agg*, *e.g.*, "add," "mean," "max," to aggregate them, and outputs new feature maps. Since DGCNN creates graphs in the feature space, it is more robust to noisy PCD. We use the default setting for the DGCNN layer where $k = 16$ and *Agg*="max."

Table 1. Confidence Score Predictor network parameters. MLP: Multi-Layer Perceptron; DGCNN: Dynamic Graph Convolution Neural Network layer; ReLU: Rectifier Linear Unit.

| Layer | MLP1 | MLP2 | DGCNN1 | MLP3 | MLP4 | DGCNN2 | MaxPool | MLP5 | MLP6 |
|---|---|---|---|---|---|---|---|---|---|
| Number of Points | N | N | N | N | N | N | 1 | N | N |
| Input Channels | 3 | 32 | 64 | 64 | 128 | 256 | 256 | 512+64 | 256 |
| Output Channels | 32 | 64 | 64 | 128 | 256 | 256 | 256 | 256 | 1 |
| Activation Function | ReLU | ReLU | | ReLU | ReLU | | | ReLU | Sigmoid |

At a high level, to predict a confidence score for each point, the network should be parameterized by the current point coordinate as well as its nearby points. If it is far away from nearby points, then it should output a low confidence score. Thus, based on this idea, we use DGCNN layers to aggregate the nearby features and explore the geometric relation to produce a better confidence score. Fig. 4 shows the detailed design of this predictor. Especially, it first uses a stack of densely connected DGCNN layers to learn the multi-scale local features for each point and a global max pooling layer to learn a global feature. Then, it concatenates these multi-scale features and global features together and uses an MLP with a sigmoid function to predict the final confidence score. These

confidence scores will be used for the feature extractor to generate a noise-robust global shape feature. Table 1 summarizes the score predictor network.

*4.3.2 Noise-Aware Feature Extractor.* Given the confidence scores and the coordinates of a point set, the noise-aware feature extractor aims to extract a noise-robust global shape code from them, which encodes the 3D geometric structure of the current environment. As mentioned before, due to the existence of noisy points in the PCD and the naturally unordered distribution of points, the feature extractor should work robustly and be able to extract accurate geometric features from the noisy point set. To achieve this, the previous method uses multi-layer perceptrons (MLPs) for encoding and decoding the point cloud data [11]. However, as MLPs process each point independently, it fails to explore the context of nearby points, resulting in an inaccurate shape code prediction. Another possible solution is to use the DGCNN for the feature extractor. Although DGCNN is good at handling noise, its layer requires huge computation resources to build a graph in the feature space. To solve this challenge, we use the feature extractor backbone introduced in PointNet++ [25], which shows a good ability to extract local geometric features, runs fast, and has been widely used in many PCD-related tasks [57, 58]. It first samples a set of center points via Farthest Point Sampling from the input point set and then aggregates their nearby points using point convolution operation to explore the local geometric structures. However, the original PointNet++ backbone only takes point coordinates as input, while we have extra confidence scores to help generate a noise-robust global shape code. Thus, we customize the PointNet++ backbone and design a noise-aware feature extractor that takes the confidence scores with point coordinate features as input. Fig. 4 shows the detailed network structure of the shape feature extractor. Especially, we first concatenate confidence score with point coordinates and use blocks of point convolution with FPS layers to decrease the number of points and aggregate their local context. Then, we use a global max pooling layer to generate the global shape code. These shape codes will then be used for generating denser PCD with better shapes and structures. Table 2 summarizes the noise-aware feature extraction network parameters.

Table 2. Noise-Aware Feature Extractor network parameters. PConv: Point Convolution with Farthest Point Sampling; F + 3: F is the feature dimension, and 3 is the (x,y,z) coordinate value of the point.

| Layer | MLP1 | MLP2 | PConv1 | MLP3 | MLP4 | PConv2 | MLP5 | MLP6 | MaxPool |
|---|---|---|---|---|---|---|---|---|---|
| Number of Points | N | N | N/2 | N/2 | N/2 | N/4 | N/4 | N/8 | 1 |
| Input Channels | 1+3 | 64 | 256 | 256+3 | 256 | 384 | 384+3 | 384 | 512 |
| Output Channels | 64 | 256 | 256 | 256 | 384 | 384 | 384 | 512 | 512 |
| Activation Function | ReLU | ReLU | | ReLU | ReLU | | ReLU | ReLU | |

*4.3.3 Seed Generator and Point Upsampling.* To generate complete and denser PCD from the global shape codes, the previous method uses a Folding technique [59], where it first duplicates the shape code multiple times and concatenates them with predefined 2D grids, and then uses MLPs to regress for the point coordinates. However, we find that it is cumbersome to directly expand the one-dimensional shape code thousand times. With so many duplications, many of them may fall into the same coordinate value, making it difficult for the network to find optimal weights. Besides, previous works, such as DeepPoint [11], used MLPs based networks to generate the PCD, which cannot capture the local geometric information from nearby points; so, the generated points are not well distributed, and the PCD mostly contains ambiguous global shapes. Thus, to generate well distributed PCD, we follow the idea in [26], where it uses upconvolution layers to generate a sparse but complete seed point cloud from the global shape code, and then uses a specially designed upsampling layer to increase the density of PCD. Ideally, the coarse seed points should contain sufficient structure information to facilitate the following upsampling operations. To achieve this, we use a Seed Generator module, similar to [26], to generate a coarse but complete 3D PCD from the global shape code. Especially, it first uses a 1D up-convolution block to expand

the number of feature maps given the global shape code. Then, these feature maps will be concatenated with the shape code to predict the coordinate of seed points via MLPs. Fig. 4 and Table 3 show the detailed network structure of the PCD seed generator module.

Table 3. Seed Generator network parameters. UConv: 1D Up Convolution.

| Layer | UConv1 | MLP1 | MLP2 | MLP3 | MLP4 | MLP5 | MLP6 |
|---|---|---|---|---|---|---|---|
| Number of Points | 1 | N | N | N | N | N | N |
| Input Channels | 512 | 128+512 | 256 | 128+512 | 128 | 128 | 64 |
| Output Channels | 128 | 256 | 128 | 128 | 128 | 64 | 3 |
| Activation Function | | ReLU | ReLU | ReLU | ReLU | ReLU | |

Given a set of seed points, the point upsampling module aims to upsample them multiple times to generate the final PCD. But upsampling the PCD is not a simple task. A strawman approach to upsampling would be to use a 3D interpolation [60–62], but these traditional methods cannot handle the sparse PCD and could produce wrong shapes. An ideal point upsampler should not only increase the number of points but also preserve an accurate underlying shape of PCD. Previous researchers have proposed many deep-learning-based upsampling techniques, such as feature space upsampling via multi-branch MLPs [63, 64] or duplication and concatenation via Folding [30, 59]. However, these techniques fail to explore the relation among nearby points during upsampling. As a result, the upsampled points may not preserve a good underlying shape. Inspired by the DGCNN [24], we design the upsampling module by introducing point local relation into the upconvolution operator. Specifically, given an input of sparse but complete PCD, we first aggregate their nearby point features by a DGCNN layer for each point to identify the local shapes. Next, we concatenate its aggregated feature with the global shape code and use MLPs to embed them together. Then, we use an upconvolution layer to upsample the embedded features in the feature space. Finally, these upsampled features will be fed into another MLP to regress for their point coordinates. Unlike using multi-branch MLPs, we can easily upsample the input PCD for arbitrary times by setting different parameters in the upconvolution layer or stack many upsampling blocks together, which also makes the network more robust to a large upsampling ratio. Fig. 4 and Table 4 show the detailed network structure and parameters of the proposed upsampling module.

Table 4. Point Upsampling network parameters. r is the upsampling ratio.

| Layer | MLP1 | MLP2 | DGCNN1 | MLP3 | MLP4 | UConv1 | MLP5 | MLP6 |
|---|---|---|---|---|---|---|---|---|
| Number of Points | N | N | N | N | N | rN | rN | rN |
| Input Channels | 3 | 32 | 64 | 64+512 | 256 | 128 | 128 | 64 |
| Output Channels | 32 | 64 | 64 | 256 | 128 | 128 | 64 | 3 |
| Activation Function | ReLU | ReLU | | ReLU | | | ReLU | |

*4.3.4 Loss Function.* To ensure a near-optimal network convergence, we build the loss function to measure the geometrical similarity between the predicted and ground truth PCD. To this end, we first use the Chamfer Distance (ChD), a well-known metric to find the quantitative difference between two point sets [29]. ChD measures the average squared L2-norm distance among two point sets and is defined as:

$$L_{ChD}(S_1, S_2) = \frac{1}{N_1} \sum_{x \in S_1} \min_{y \in S_2} ||x - y||_2^2 + \frac{1}{N_2} \sum_{y \in S_2} \min_{x \in S_1} ||x - y||_2^2 \quad (1)$$

where $S_1$ and $S_2$ are the point sets and $N_1$ and $N_2$ are the number of points in them.

However, we notice that ChD loss alone may not guarantee a good distribution of points; this is because ChD only measures the distance between closest points, and may not fully capture the underlying point distribution

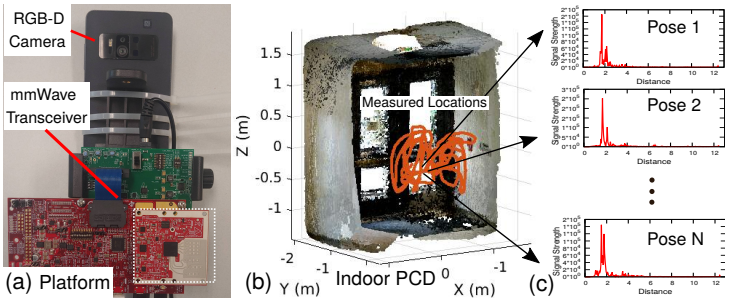


Figure 5. (a) Hardware platform of *MilliPCD*. (b) Example visual PCD with various poses. (c) Example reflection profiles.

differences, especially when the two PCDs are large. Thus, to balance the point distribution, we also introduce an Earth Movers' Distance in the loss function, which is defined as:

$$L_{EMD}(S_1, S_2) = \frac{1}{N_1} \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} ||x - \phi(x)||_2 \tag{2}$$

where $S_1$ and $S_2$ are the point sets, $N_1$ is the number of points in $S_1$, and $\phi : S_1 \to S_2$ is a Bijection function to ensure that each element of $S_1$ is paired with exactly one element of $S_2$. The EMD function first finds the minimal Bijection function from point set $S_1$ to $S_2$ and then calculates the average euclidean distance between corresponding points. Intuitively, the EMD function attempts to find a minimum effort in transforming one PCD into another PCD. Note that due to the Bijection function, two PCDs should have an equal number of points. As the exact computation of EMD is too expensive for training, we use an approximation based implementation introduced in [29].

What's more, as we introduce the confidence score predictor to assign a confidence score for each input point, we also need to design a loss function to measure the difference between ground truth confidence scores and predicted scores. But, the challenge we are facing is not only measuring the difference in scores but also defining the ground truth confidence score. Intuitively, for those points that are far away from ground truth PCD, there is a high probability that these points are noises and should have smaller confidence scores. Inspired by this intuition, we explore different designs of confidence score functions and use the following equation to calculate the ground truth confidence score: $Score(x) = exp(-\min_{y \in S_2} ||x - y||_2)$, where $S_2$ is the ground truth point set, and the value of this score goes from 0 to 1. Especially, if the point is closer to the ground truth PCD, the score is closer to 1. Based on the *Score*, we design a Mean Square Error (MSE) loss for the confidence score predictor $L_{Score} = MSE(pred - Score)$, where *pred* is the predicted score. Finally, as we use the seed generator to generate sparse but complete seed points, we also anticipate that the generated seed points should be close to the ground truth PCD; so, we measure their difference using another ChD loss $L_{seed}$. The final loss function is:

$$L = \alpha L_{Score} + L_{EMD} + L_{ChD} + \beta L_{seed} \tag{3}$$

where $\alpha$ and $\beta$ are the hyper-parameters to balance the weight of different components.

## 5 IMPLEMENTATION

### 5.1 Hardware platform and Data Collection

Due to the unavailability of open-sourced indoor mmWave reflection data with PCD, we design a data collection platform to collect the ground truth PCD, poses, and reflected mmWave signals by ourselves. We design a handheld device integrating an RGB-D camera and a mmWave transceiver for data collection. Figs. 5(a–c) show

the platform and the collected data samples. The platform uses a co-located Asus ZenPhone with an RGB-D camera [28] and a TI IWR6843ISK 60 GHz mmWave transceiver [27] to survey an environment. The smartphone is equipped with a 22.7 megapixel (MP) RGB camera and a 0.3 MP depth sensor with 77° Field of View (FoV). The camera and depth sensor have sampling rates of 30 fps and 1.8 fps, respectively, but the maximum range of the depth sensor is 4 m only. Thus, to scan a large environment, we walk around holding the phone in different poses. By using RTAB-Map [42], we can collect a ground truth 3D PCD and device poses simultaneously. TI IWR6843ISK mmWave transceiver operates at the unlicensed 60 GHz mmWave frequency band and uses a bandwidth of 4 GHz and a sampling rate of 50 fps. The FoV of this transceiver is 28 and 56 degrees in azimuth and elevation angles, respectively. By using this co-located device, we can collect poses, PCD, and mmWave data simultaneously. After collecting each raw, complete PCD, we notice three problems. *First*, the sampling rate of the mmWave transceiver is different from the poses collected by Asus ZenPhone. To this end, we use spline interpolation to upsample the collected poses and synchronize them with mmWave signals. *Second*, the ground truth PCD could still contain many redundant parts and floating points caused by the RGB and depth sensor pollution. Since the noisy data in ground truth PCD could affect the network training and applications, we need to pre-process them. To this end, we use the MeshLab [65], which allows us to remove unnecessary parts manually. *Third*, the FoV of the mmWave transceiver is different than the RGB-D camera; so, it is challenging to obtain accurate ground truth PCD since some parts of the environment could be missing in the mmWave reflected signals. To solve this problem, we select the visible parts as ground truth in the original PCD collected by the RGB-D camera based on the poses and FoV of the transceiver.

We collect datasets from 13 different indoor environments in building A, and due to the limited scanning time of the mmWave transceiver, each large indoor environment is split into multiple distinct areas. Especially, we have 37 different data samples, and each of them is from different parts of indoor environments. We use 32 of them as training data and the rest as testing data. For each data sample, we can further split them into small patches based on the poses. For example, the mmWaves, poses, and PCD of each continuous 750 poses will be treated as a small patch. In total, we have 1100 training patches and 165 testing patches. These testing patches are different from training patches. What's more, to evaluate the generalization ability of our method in unseen environments, we also collect additional data samples from 3 different indoor environments in building B with 231 testing patches in total. Table 5 shows the properties of collected indoor environments in 2 buildings. Note that each small patch of PCD may contain millions of points. To expedite the training/testing time, we downsample the ground truth PCD to contain 2,048 points using random sampling.

## 5.2 Network Training

We explore different network settings to ensure *MilliPCD* converges to the optimal values. To train the network, we set the total epoch to be 700 and use "Adam" as the optimization function with a learning rate of $1{\times}10^{-4}$ at the beginning. Then, we use a smart learning scheduler to speed up the training process by decreasing the learning rate with a scale of 0.5 every 100 epochs. It allows the network to converge faster than a fixed learning rate. We also explore different combinations of hyper-parameters by training the network multiple times and find that our network performs much better when $(\alpha, \beta) = (1, 0.5)$. This fits our intuition because *MilliPCD* cares more about the final global reconstruction results than the seed point cloud, and the confidence score loss should be as important as reconstruction loss to make sure the confidence score predictor can learn correctly. Across all training datasets, our network converges successfully within 700 epochs. Our network is implemented using PyTorch [66] with Python 3.8 and trained in a server with Nvidia's RTX A6000 [67]. The network takes ~6 hours to complete the training.

Table 5. Properties of the indoor environments in 2 buildings where the data is collected. Each large indoor environment is split into multiple distinct areas.

| Env. | Area | L (m)×W (m)×(H (m) | Train/Test | Env. | Area | L (m)×W (m)×(H (m) | Train/Test |
|---|---|---|---|---|---|---|---|
| **A.1** | 1 | $7.6 \times 7.7 \times 3.4$ | Train | **A.1** | 2 | $10.2 \times 6.1 \times 3.3$ | Test |
| **A.2** | 1 | $2.9 \times 2.4 \times 3.3$ | Train | **A.2** | 2 | $3.9 \times 3.5 \times 3.3$ | Train |
| **A.2** | 3 | $4.7 \times 1.9 \times 3.3$ | Train | **A.2** | 4 | $2.2 \times 2.5 \times 3.3$ | Train |
| **A.3** | 1 | $13.0 \times 6.3 \times 4.8$ | Train | **A.4** | 1 | $6.6 \times 5.5 \times 3.5$ | Train |
| **A.4** | 2 | $5.5 \times 5.1 \times 3.5$ | Train | **A.5** | 1 | $8.2 \times 3.5 \times 3.2$ | Train |
| **A.5** | 2 | $8.6 \times 3.0 \times 3.2$ | Train | **A.5** | 3 | $6.3 \times 3.6 \times 3.5$ | Train |
| **A.5** | 4 | $12.5 \times 2.7 \times 3.2$ | Test | **A.5** | 5 | $11.1 \times 2.5 \times 3.2$ | Train |
| **A.6** | 1 | $11.7 \times 5.6 \times 3.4$ | Train | **A.6** | 2 | $10.0 \times 4.3 \times 3.4$ | Train |
| **A.7** | 1 | $5.7 \times 6.0 \times 3.0$ | Train | **A.7** | 2 | $6.1 \times 3.3 \times 3.0$ | Train |
| **A.7** | 3 | $1.9 \times 2.0 \times 3.2$ | Test | **A.8** | 1 | $12.7 \times 3.1 \times 4.0$ | Train |
| **A.8** | 2 | $7.8 \times 4.1 \times 3.4$ | Train | **A.9** | 1 | $8.8 \times 8.3 \times 3.1$ | Train |
| **A.10** | 1 | $10.1 \times 4.5 \times 3.6$ | Train | **A.10** | 2 | $10.7 \times 8.5 \times 3.1$ | Train |
| **A.11** | 1 | $6.8 \times 6.7 \times 3.1$ | Train | **A.11** | 2 | $7.2 \times 8.1 \times 2.9$ | Train |
| **A.11** | 3 | $10.0 \times 2.4 \times 3.0$ | Train | **A.11** | 4 | $10.9 \times 3.5 \times 3.0$ | Train |
| **A.12** | 1 | $9.4 \times 6.3 \times 3.3$ | Train | **A.12** | 2 | $6.9 \times 6.3 \times 3.2$ | Test |
| **A.12** | 3 | $8.3 \times 2.9 \times 3.0$ | Train | **A.12** | 4 | $10.6 \times 2.3 \times 3.0$ | Train |
| **A.12** | 5 | $7.7 \times 2.6 \times 3.0$ | Train | **A.13** | 1 | $12.7 \times 6.1 \times 3.0$ | Train |
| **A.13** | 2 | $12.7 \times 2.6 \times 3.0$ | Test | **A.13** | 3 | $11.5 \times 6.1 \times 3.0$ | Train |
| **A.13** | 4 | $11.0 \times 6.6 \times 3.0$ | Train | **B.1** | 1 | $4.4 \times 6.1 \times 3.8$ | Test |
| **B.1** | 2 | $6.3 \times 3.7 \times 3.9$ | Test | **B.2** | 1 | $7.0 \times 2.2 \times 2.8$ | Test |
| **B.2** | 2 | $7.3 \times 2.8 \times 2.9$ | Test | **B.3** | 1 | $4.5 \times 3.0 \times 3.5$ | Test |
| **B.3** | 2 | $3.9 \times 2.7 \times 3.5$ | Test | **B.3** | 3 | $8.2 \times 2.8 \times 3.9$ | Test |
| **B.3** | 4 | $8.3 \times 4.6 \times 3.9$ | Test | | | | |

## 6 EXPERIMENTAL RESULTS

After training, we evaluate and study the network design of *MilliPCD* carefully. Especially, we first show the performance of *MilliPCD* compared to previous algorithms and then evaluate *MilliPCD*'s robustness and network design via multiple ablation studies. We compare the performance of all algorithms using the following two standard metrics that are commonly adopted to measure the quality of generated PCD *w.r.t.* ground truths.

**L1 Chamfer Distance (L1-ChD):** Different from the original Chamfer Distance, L1 Chamfer Distance changes the squared L2 Distance to the Euclidean Distance. It has physical meanings and the unit of L1-ChD is a meter, which intuitively reflects the physical distance between two point sets. The smaller the L1-ChD, the better the reconstructions. Its scale goes from 0 to $\infty$.

**Earth Mover's Distance (EMD):** The objective measure of the distribution distortion of generated PCD *w.r.t.* ground truth. It measures the minimal cost that must be paid to transform one PCD into the other [30]. Thus, we include it as our second measurement to better reflect the quality of reconstruction results. Its scale goes from 0 to $\infty$, where 0 means a perfect match between the coordinates of ground truth PCD and generated PCD.

**Evaluation Summary:** The evaluation result on 165 testing samples shows that the proposed *MilliPCD* achieves a median L1-ChD of 0.256 m and a median EMD of 0.496 m, which outperforms previous non-coherent imaging and GAN based approaches [11], where they achieve the median L1-ChD of 0.460 m and 0.329 m only, and the median EMD of 0.673 m and 0.607 m. Furthermore, the ablation study shows that *MilliPCD* can achieve robust reconstruction results under different settings.

Table 6. Quantitative difference between *MilliPCD* generated PCD and ground truth, and comparison with other methods.

| Metrics | Traditional | DeepPoint | *MilliPCD* |
|---|---|---|---|
| L1-ChD (median) | 0.460 | 0.329 | **0.256** |
| L1-ChD ($90^{th}$ %-ile) | 0.636 | 0.575 | **0.405** |
| EMD (median) | 0.673 | 0.607 | **0.496** |
| EMD ($90^{th}$ %-ile) | 0.952 | 0.904 | **0.723** |

## 6.1 Microbenchmark Results

We first test *MilliPCD* with real data, where the poses, building structures, and reflected mmWave signals are complex and challenging. We evaluate the effectiveness of the proposed system on 165 real data samples collected from building A and compare the result with previous mmWave-based PCD generation systems: Traditional non-coherent imaging approach and DeepPoint [11]. For the traditional non-coherent imaging approach, where it processes received mmWave signals along the trajectory separately and then combines them together to generate PCD, we implement it with Matlab and for the DeepPoint, we implement their algorithm with PyTorch.

*6.1.1 Quantitative Results.* Table 6 shows the quantitative results of all algorithms. We see that, *MilliPCD* outperforms all counterparts with a large gap. Especially, *MilliPCD* achieves a median and 90 percentile L1-ChDs of only 0.256 m and 0.405 m, and a median and 90 percentile EMDs of only 0.496 m and 0.723 m. In contrast, the median and 90 percentile L1-ChD achieved by the DeepPoint are 0.329 m and 0.575 m, and the median and 90 percentile EMD of DeepPoint are 0.607 m and 0.904 m, respectively, While the traditional method achieves the worst performance. Fig. 6 shows the CDF plot of different algorithms over all testing samples.

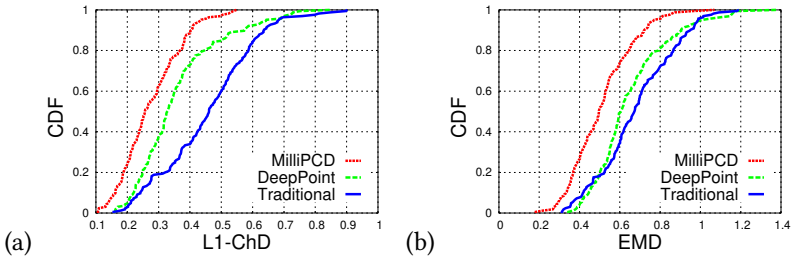


Figure 6. Loss distributions for different algorithms. Loss is measured using (a) L1-ChD and (b) EMD.

*6.1.2 Qualitative Results.* Since we achieved the best quantitative result, we then visually check the outputs of *MilliPCD* and compared them with other algorithms. Fig. 7 shows the visual examples of generated PCD by different algorithms. Note that as the ground truth PCD are generated based on the FoV and poses of the mmWave transceiver, they are naturally incomplete due to limited scanning trajectories and FoV. But even with incomplete ground truth PCD, the proposed network is still able to learn correct shapes from coarse PCD. We see that *MilliPCD* generates the PCD that preserves good geometric structures and resembles optical and ground truth PCD. Due to the limited scanning trajectories, specularity, and noisy reflection points, the traditional non-coherent imaging can only generate very sparse and noisy PCD and fails to infer the potential shape from these points. On the other hand, with the support of deep learning, DeepPoint is able to extract high-level shape features from noisy points and generate better PCD. We can identify the coarse shape of each environment from its results, but their generated points are not well distributed due to their MLP-based network design. To improve the performance, we first generate coarse seed points and then upsample them for better point distribution.

Specifically, in the last column of Fig. 7, the hallway generated by *MilliPCD* contains accurate geometric shapes and can be easily recognized, while the PCD generated by DeepPoint and traditional non-coherent imaging are difficult to recognize. Based on these experiment results, we can conclude that *MilliPCD is able to reconstruct PCD from mmWave reflected signals for indoor environments.*

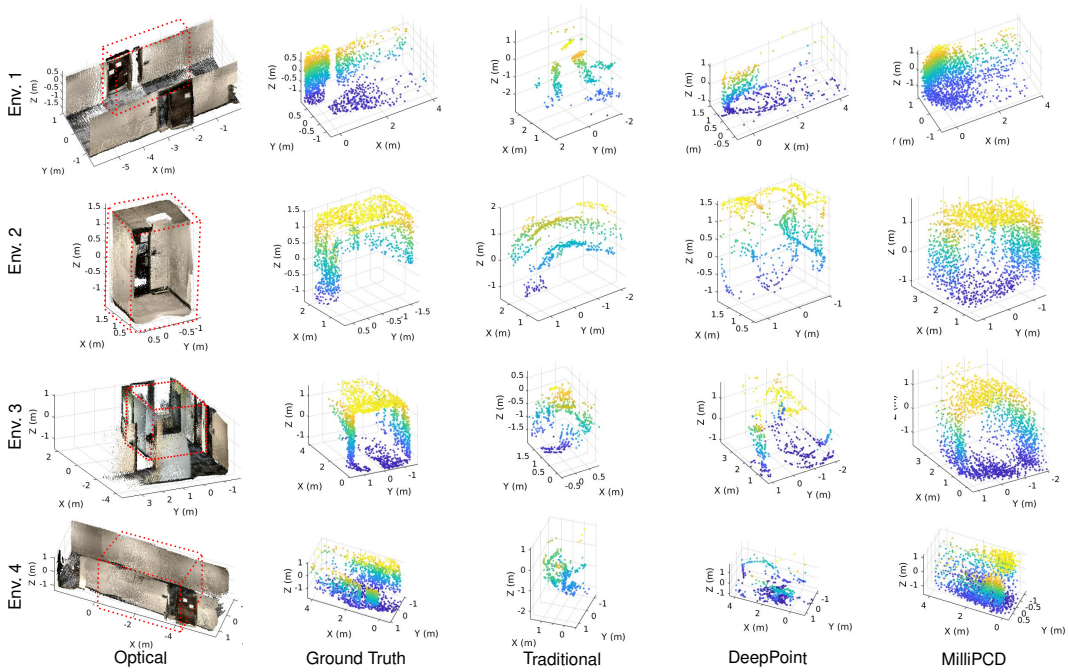


Figure 7. The visual results of generated PCD via different algorithms. Optical: The original PCD collected by the RGB-D camera. The red bounding box indicates the volume of the ground truth patch. Ground truth: The PCD derived from the depth information in optical PCD, and updated based on the FoV and poses of the mmWave transceiver. Traditional: PCD from non-coherent mmWave imaging. DeepPoint: PCD output from an existing system [11]. *MilliPCD*: Our predicted PCD.

## 6.2 Ablation Study

We now analyze the major components of *MilliPCD* to help us understand the system better. To this end, we perform explicit ablation studies on *MilliPCD*'s components. Specifically, we evaluate the robustness of *MilliPCD* under a variety of settings (*e.g.*, different number of upsampled points, required scanning time, and new unseen buildings), the effect of different network designs (*e.g.*, confidence score predictor and different design of ground truth confidence score), and the model size and run time complexity.

*6.2.1 Effect of Different Density of Points in PCD.* Recall that our PCD generation system can populate PCD with different numbers of points easily by stacking multiple point upsampling blocks with different upsampling ratios as introduced in Section 4. We then evaluate *MilliPCD*'s performance when generating a different number of points in each PCD. Specifically, we test several cases where the system generates PCDs with 1024, 2048, and 4096 points. To generate 1024 points, we stack 3 upsampling blocks with upsampling ratios of [2, 2, 4], respectively. Similarly, for 2048 points the ratio is set to [2, 4, 4] and for 4096 points the ratio is set to [4, 4, 4]. On the other hand, to facilitate the training, we also downsample the ground truth PCD with these many points. Intuitively,

with more points in a specific range of area, the L1-ChD and EMD among two point sets are expected to be smaller. However, more points also mean more computational time, and training *MilliPCD*'s network would cost more time. Thus, we also show the training time of *MilliPCD* along with its performance.

Table 7. Performance of *MilliPCD* under a different number of points.

| Number of Points | L1-ChD (median) | L1-ChD ($90^{th}$ %-ile) | EMD (median) | EMD ($90^{th}$ %-ile) | Training Time |
|---|---|---|---|---|---|
| 1024 | 0.283 | 0.442 | 0.521 | 0.757 | 3.2 hours |
| 2048 | **0.256** | **0.405** | 0.496 | 0.723 | 6 hours |
| 4096 | 0.260 | 0.412 | **0.474** | **0.717** | 12.6 hours |

Table 7 shows the quantitative result of this ablation study. We see that the median and $90^{th}$ percentile EMD loss under 1024 points could be up to 0.52 and 0.75, respectively. But these losses reduce when the number of points increases to 2048, which fits our intuition. But a larger number of points do not always improve *MilliPCD*'s performance: There is hardly any L1-ChD improvement if the number of points continues to increase. Besides, with more points, the training time also increases from approximately 3.2 hours to 12.6 hours and is almost doubled when the number of points is doubled. Fig. 8 shows example visual results under different numbers of points. We notice that even with only 1024 points, the generated PCD still preserves a good geometric shape of the indoor environment. When the number of points increases, the geometric shape does not change too much. This proves that *the proposed system accurately learns the geometric shape of the surrounding environment and thus is robust to the number of points to be generated.* Note that to balance between the training time and accuracy, we use 2048 points as default for all other experiments.

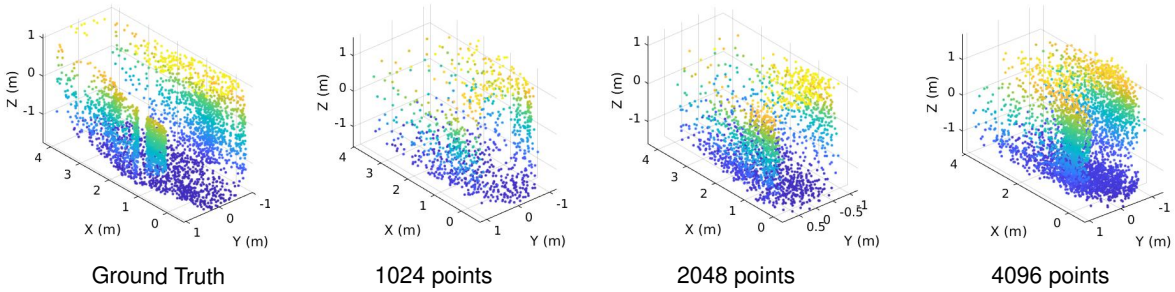


Figure 8. (a) Visible ground truth PCD collected by an RGB-D camera with 4096 points. (b) Generated PCD with 1024 points. (c) Generated PCD with 2048 points. (d) Generated PCD with 4096 points.

*6.2.2 Performance Under Different Scanning Time.* Recall that when generating the training and testing samples, we fix the number of scanning poses to 750 for simplicity. Intuitively, with more scanning poses, users can collect

Table 8. Performance of *MilliPCD* with different scanning times.

| Scanning Time | 5s (250 poses) | 10s (500 poses) | 15s (750 poses) |
|---|---|---|---|
| L1-ChD (median) | 0.267 | 0.258 | 0.256 |
| L1-ChD ($90^{th}$ %-ile) | 0.431 | 0.421 | 0.405 |
| EMD (median) | 0.498 | 0.493 | 0.496 |
| EMD ($90^{th}$ %-ile) | 0.748 | 0.763 | 0.723 |

more information about the current environment to generate a PCD with better structure details. But on the other hand, the required scanning time also increases. Ideally, we hope *MilliPCD* can work robustly even in the case that users quickly scan the environment.

So, to understand the effects of required scanning time on *MilliPCD*, we study its performance under different scanning times. But, collecting the data samples with the same trajectory but different scanning times is difficult. To solve this problem, we synthesize this data collection process by uniformly downsampling the original sequence of collected poses and mmWave signals with different ratios. Especially, for each PCD we generate data samples with different numbers of poses from 250 to 750 at a step of 250 poses, which is equal to the expected scanning time increasing from 5s to 15s at a step of 5s. Table 8 shows the quantitative result and Fig. 9 shows the visual result. We see that with a scan time of 15s (750 poses) *MilliPCD* achieves the best performance with the lowest L1-ChD and EMD. But, we also notice that even with a scan time of 5s (250 poses), *MilliPCD* is still able to produce considerable good PCD, and there hardly has any improvement if the scan time continues to increase. This proves that MilliPCD *is robust to the required scan time as long as the scanning viewpoints cover the entire environment, which means users can quickly scan over the environment without worrying about a huge performance drop.*

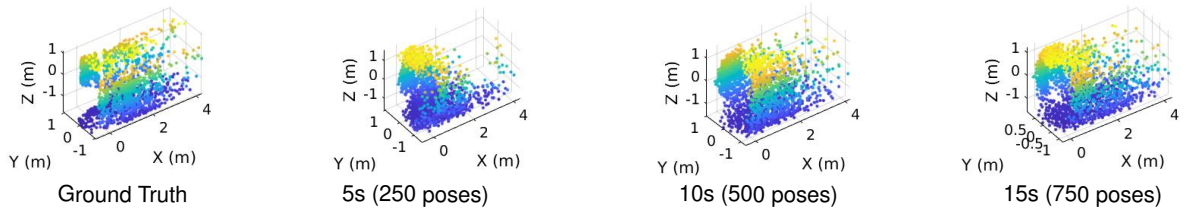


Figure 9. (a) Visible ground truth PCD collected by an RGB-D camera with 750 poses (15s). (b) Generated PCD with 250 poses (5s). (c) Generated PCD with 500 poses (10s). (d) Generated PCD with 750 poses (15s).

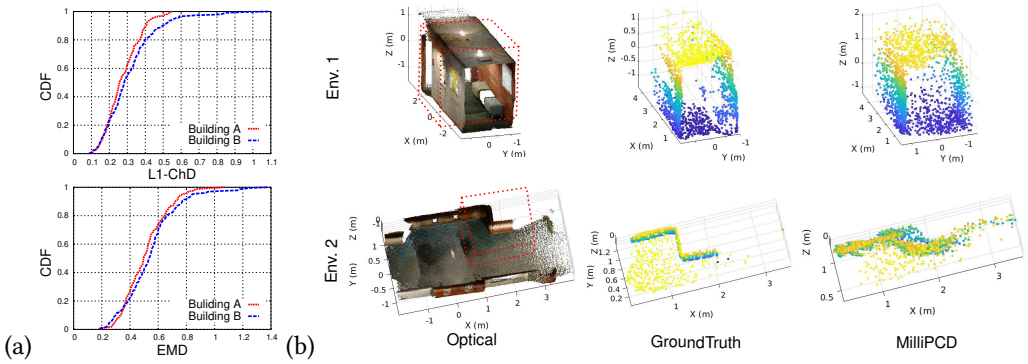


Figure 10. The visual results of generated PCD on unseen data. *MilliPCD* is trained on the data collected from building A and tested on the data collected from building B. (a) The CDF plots of L1-ChD and EMD on data from two buildings. (b) Visual examples of reconstruction results in an unseen environment B.

*6.2.3 Generalization Ability on Unseen Environments.* Furthermore, to evaluate the generalization ability of *MilliPCD*, we also test it on data samples collected from an unseen building B, with different properties of building structures, such as materials, floor heights, and indoor furniture. Specifically, *MilliPCD* is trained using only the training data collected from building A and tested on the data collected from building B. Fig. 10 shows the experiment results. Compared to the quantitative result from building A, we see that the median L1-ChD increases from 0.256 m to 0.283 m, and the median EMD increases from 0.496 m to 0.534 m. Even if the quantitative performance has a slight drop, the generated PCD are still quite similar to the ground truth. *These results illustrate that MilliPCD generalizes well on unseen indoor buildings.*

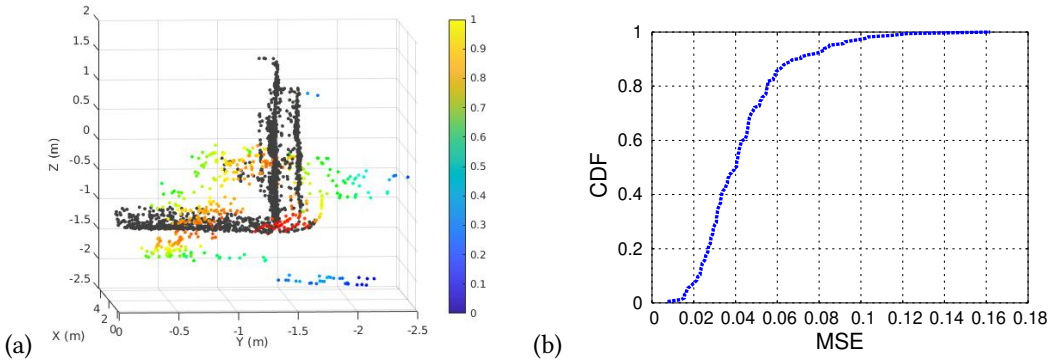


Figure 11. (a) The heat-map of predicted confidence scores: Black points are ground truth, and colored points are raw refection points with different confidence scores. (b) Distribution of Mean Square Error for predicted confidence score.

*6.2.4 Effect of Confidence Score Predictor.* Next, we focus on the design of the shape reconstruction network. and conduct explicit ablation studies on its major components. To begin with, we study the effectiveness of Confidence Score Predictor and its importance for the rest part of the network. Recall that the score predictor outputs a score indicating the importance of each point. The questions one might ask are that can it correctly output the score and can it bring improvement to the overall performance of *MilliPCD*. To validate the design of the confidence score predictor, we check the score of each output point. Fig. 11a shows the heat-map of predicted confidence scores for raw noisy inputs along with the ground truth PCD. We see that, when the point is closer to the ground truth, it has a high confidence score, and when the point is far away from the ground truth, it has a low confidence score. Fig. 11b shows the CDF plot of the mean square error of predicted confidence scores. This result fits our anticipation and proves the design.

Then, we validate the importance of confidence score predictor for the rest part of the network. Especially, we remove the score predictor, and directly feed reflection points into the feature extractor to extract the shape code.

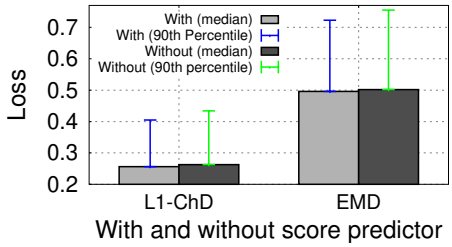


Figure 12. Performance of *MilliPCD* with and without confidence score predictor.

Note that the feature extractor is modified for features with confidence scores. To adapt the feature extractor so that it can directly process reflection points, we modified it by replacing the score feature with point coordinates. For fairness, we train this ablation network with the same parameter setting as the previous network. Fig. 12 shows the performance of *MilliPCD* by removing the score predictor. We see that with the confidence score predictor the $90^{th}$ %-ile L1-ChD improves from 0.434 m to 0.405 m and the $90^{th}$ %-ile EMD improves from 0.755 m to 0.723 m. This proves our intuition that the confidence score predictor helps the following reconstruction network. *With the confidence score, our feature extractor can focus on high confidence points and thus be more robust to noise, and* MilliPCD *can achieve better performance.*

*6.2.5 Different Design of Ground Truth Confidence Score.* After verifying the effectiveness of the confidence score predictor, we then explore different functions for defining the ground truth confidence score. As mentioned in system design (Section 4), we calculate the ground truth confidence score using the exponential function of distance. One may ask about the possibility of other score functions. Actually, during the design of this score metric, we propose three different functions: (1) Exponential, $Score = e^{-d}$, (2) Fractional, $Score = 1/(1+d)$, and (3) Normalization, $Score = 1 - d/max(d)$, where d is distance. Recall that, our target is to design a function that has a value of 1 when the point is exactly located on the ground truth PCD and reduce quickly when the point is far from the ground truth. Apparently, the exponential-based function meets our requirement best, where it can reduce quicker than others when the distance goes larger. What's more, to quantitatively show the performance

Table 9. Performance of different score metrics.

| Metrics | L1-ChD (median) | L1-ChD ($90^{th}$ %-ile) | EMD (median) | EMD ($90^{th}$ %-ile) |
|---|---|---|---|---|
| Exponential | **0.256** | **0.405** | **0.496** | **0.723** |
| Fractional | 0.261 | 0.421 | 0.493 | 0.730 |
| Normalization | 0.264 | 0.422 | 0.497 | 0.741 |

of *MilliPCD* by using these confidence score functions, we train and test *MilliPCD* with these score metrics. During training, we only modify the score function and fix the rest part of the network for a fair comparison. Table 9 shows the reconstruction result with each design. We see that the exponential-based score function achieves the best performance with a median L1-ChD of 0.256m and a median EMD of 0.496m, which fits our intuition.

*6.2.6 Run-time Complexity of MilliPCD.* We currently build and validate *MilliPCD*'s system on a GPU server, which has strong computational power. But for mobile and ubiquitous computing environments, the running time is important for user experience. To understand the time complexity of *MilliPCD*, and the feasibility of running the proposed system on mobile devices, we evaluate its average inference time and model size. Table 10 shows the average inference time on the RTX A6000 GPU server. We see that the network can run fast in GPU server, with an average inference time of 0.27 s, and the model size is quite small, only 68.8 MB. However, considering that mobile devices do not have such strong computational ability, it is infeasible to directly run our system on them. But, it is possible to run in a server-client model, where a user can upload the collected poses and mmWave signals to the GPU server and download the generated PCD quickly.

Table 10. Model size and inference time for *MilliPCD*.

| Model Size | Inference Time |
|---|---|
| 68.8 MB | 0.27 s |

## 7 LIMITATIONS AND FUTURE WORKS

**Limitation in Measuring the Ground Truth PCD**: We collect the original PCD using RTAB-Map [42] software running on the ASUS ZenPhone AR with RGB-D cameras [28]. But, directly using them as the ground truth may not be a good approach as the FoV of the RGB-D camera is larger than the mmWave transceiver; so, it may confuse the algorithm and make it difficult to learn the correct mapping from mmWave signals to PCD. To this end, we select the visible parts of the collected PCD as the ground truth based on the poses and FoV of the mmWave transceiver. However, due to limited scanning trajectories, the visible ground truth PCD could be incomplete, and the generated outputs of *MilliPCD* may also be incomplete. Considering that the proposed algorithm can work with other PCD processing methods, such as indoor point cloud completion [68, 69], it is possible to integrate them to generate better ground truth PCD. We leave this as future work.

**Dynamic Reflections Behind the Walls**: Since mmWave signals can penetrate through certain obstructions, the transceiver can receive reflected signals from objects behind thin drywalls or doors. For static objects behind the walls, such as metal cabinets, the noise-aware shape reconstructor can remove those spurious points and generates a high-quality indoor PCD. However, for dynamic reflections behind the walls, such as a person walking behind a thin door during the data collection process, could create multiple spurious reflections within the environment that may confuse the shape reconstructor. Training the model to learn such spurious reflections due to dynamic objects is very challenging since we cannot collect any ground truth with visual sensors, such as RGB-D or LiDARs, for non-line-of-sight objects and events. Currently, *MilliPCD* relies on the assumption that the target-scene remains stationary during the entire data collection procedure. We leave the analysis of the effects of dynamic reflections and the design to handle this as future work.

**Imaging Small Indoor Targets**: *MilliPCD* relies on the reflection from an object to identify its shape. Applying *MilliPCD* for reconstructing both the general structure of a large building and the specific structure of small objects, such as desks, chairs, and bins is another challenging task. Since the resolution of the mmWave transceiver is low, the reflected signals from small objects, especially those providing very weak reflectivity, could be easily buried under the large building walls and floors: Even a deep learning method may not be able to recover the shapes; so, the system cannot generate accurate shapes of small objects. Usually, to image small indoor objects, a large antenna array is necessary. Another solution might be combining LiDARs or IR cameras with mmWave sensors to identify and generate PCD of small objects. We leave this task as a future job.

**Extending *MilliPCD* to Reconstruct Outdoor PCD**: Extending *MilliPCD* to outdoor scenarios is more challenging. It requires the mmWave transceiver to capture reflected signals from complex reflectors with different sizes, orientations, and reflection profiles. Besides, the outdoor scene contains moving objects which can not be controlled and are difficult to distinguish from noisy points. To collect ground truth datasets and reconstruct medium-sized objects, such as trees, the mmWave transceiver may need to move 10s of meters to collect sufficient data. One option for data collection could be to fit such mmWave transceiver on Drones, and program it to automatically fly around and collect the mmWave signals and visual PCD, such as [3, 70]. We leave the extension of *MilliPCD* for outdoor PCD reconstruction as future work.

## 8 CONCLUSION

In this work, we propose *MilliPCD*, a system that leverages the advanced capabilities of 5G mmWave devices and deep learning models to enable "beyond traditional vision" PCD generation, for challenging environments where optical systems could fail. *MilliPCD* combines the traditional Backprojection based mmWave imaging algorithm to first construct a coarse shape of the environment, and then designs a customized Dynamic Graph Convolution Neural Network to exploit the local and global geometric shapes among reflection points to generate

a high-quality vision-like PCD. We evaluate our method with different under various environmental conditions, and the experiment results prove the effectiveness of *MilliPCD*.

## ACKNOWLEDGMENTS

We sincerely thank the reviewers and the editors for their comments and feedback. This work is partially supported by the NSF under grants CNS-1910853, CAREER-2144505, and MRI-2018966.

## REFERENCES

[1] Radu Bogdan Rusu and Zoltan Csaba Marton and Nico Blodow and Mihai Dolha and Michael Beetz, "Towards 3D Point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, 2008.
[2] Cui, Yaodong and Chen, Ren and Chu, Wenbo and Chen, Long and Tian, Daxin and Li, Ying and Cao, Dongpu, "Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–18, 2021.
[3] Chen, Fangping and Lu, Yuheng and Cai, Binbin and Xie, Xiaodong, "Multi-Drone Collaborative Trajectory Optimization for Large-Scale Aerial 3D Scanning," in *IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2021.
[4] Placitelli, Alessio Pierluigi and Gallo, Luigi, "Low-Cost Augmented Reality Systems via 3D Point Cloud Sensors," in *International Conference on Signal Image Technology Internet-Based Systems*, 2011.
[5] Gotsman, Craig and Gu, Xianfeng and Sheffer, Alla, "Fundamentals of Spherical Parameterization for 3D Meshes," in *ACM SIGGRAPH*, 2003.
[6] Wu, Zhirong and Song, Shuran and Khosla, Aditya and Yu, Fisher and Zhang, Linguang and Tang, Xiaoou and Xiao, Jianxiong, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
[7] Lars Linsen, "Point Cloud Representation," Karlsruhe Institute of Technology, Germany, Tech. Rep., 2001.
[8] Abdelaal, Mohamed and Reichelt, Daniel and Dürr, Frank and Rothermel, Kurt and Runceanu, Lavinia and Becker, Susanne and Fritsch, Dieter, "ComNSense: Grammar-Driven Crowd-Sourcing of Point Clouds for Automatic Indoor Mapping," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, 2018.
[9] Shi, Shaoshuai and Wang, Xiaogang and Li, Hongsheng, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
[10] Lu, Chris Xiaoxuan and Rosa, Stefano and Zhao, Peijun and Wang, Bing and Chen, Changhao and Stankovic, John A. and Trigoni, Niki and Markham, Andrew, "See through Smoke: Robust Indoor Mapping with Low-Cost MmWave Radar," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020.
[11] Sun, Yue and Zhang, Honggang and Huang, Zhuoming and Liu, Benyuan, "DeepPoint: A Deep Learning Model for 3D Reconstruction in Point Clouds via mmWave Radar," 2021. [Online]. Available: https://arxiv.org/abs/2109.09188
[12] K. Qian, et al., "3D Point Cloud Generation with Millimeter-Wave Radar," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 4, 2020.
[13] Zhao, Mingmin and Li, Tianhong and Alsheikh, Mohammad Abu and Tian, Yonglong and Zhao, Hang and Torralba, Antonio and Katabi, Dina, "Through-Wall Human Pose Estimation Using Radio Signals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[14] Regmi, Hem and Saadat, Moh Sabbir and Sur, Sanjib and Nelakuditi, Srihari, "SquiggleMilli: Approximating SAR Imaging on Mobile Millimeter-Wave Devices," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 3, 2021.
[15] Zhang, Guangcheng and Geng, Xiaoyi and Lin, Yueh-Jaw, "Comprehensive mPoint: A Method for 3D Point Cloud Generation of Human Bodies Utilizing FMCW MIMO mm-Wave Radar," *Sensors*, vol. 21, no. 19, 2021.
[16] TI, "Point cloud visualization for robotics using TI's mmWave sensor," 2017. [Online]. Available: https://training.ti.com/point-cloud-visualization-robotics-using-tis-mmwave-sensor
[17] Sun, Yue and Huang, Zhuoming and Zhang, Honggang and Cao, Zhi and Xu, Deqiang, "3DRIMR: 3D Reconstruction and Imaging via mmWave Radar based on Deep Learning," in *IEEE International Performance, Computing, and Communications Conference (IPCCC)*, 2021.
[18] Guan, Junfeng and Madani, Sohrab and Jog, Suraj and Gupta, Saurabh and Hassanieh, Haitham, "Through Fog High-Resolution Imaging Using Millimeter Wave Radar," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
[19] A. Ganis, E. M. Navarro, B. Schoenlinner, U. Prechtel, A. Meusling, C. Heller, T. Spreng, J. Mietzner, C. Krimmer, B. Haeberle, S. Lutz, M. Loghi, A. Belenguer, H. Esteban, and V. Ziegler, "A Portable 3-D Imaging FMCW MIMO Radar Demonstrator With a $24 \times 24$ Antenna Array for Medium-Range Applications," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 1, 2018.
[20] Mehrdad Soumekh, *Synthetic Aperture Radar Signal Processing*. John Wiley & Sons, Inc., 1999.
[21] Chen, Weiyan and Zhang, Fusang and Gu, Tao and Zhou, Kexing and Huo, Zixuan and Zhang, Daqing, "Constructing Floor Plan through Smoke Using Ultra Wideband Radar," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 4, 2022.

[22] Goodfellow, Ian and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and Ozair, Sherjil and Courville, Aaron and Bengio, Yoshua, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.
[23] Zaugg, Evan C. and Long, David G., "Generalized Frequency Scaling and Backprojection for LFM-CW SAR Processing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, 2015.
[24] Wang, Yue and Sun, Yongbin and Liu, Ziwei and Sarma, Sanjay E. and Bronstein, Michael M. and Solomon, Justin M., "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, 2019.
[25] C. R. Qi and L. Yi and H. Su and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.
[26] Xiang, Peng and Wen, Xin and Liu, Yu-Shen and Cao, Yan-Pei and Wan, Pengfei and Zheng, Wen and Han, Zhizhong, "SnowflakeNet: Point Cloud Completion by Snowflake Point Deconvolution with Skip-Transformer," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.
[27] Texas Instruments, "IWR6843 Single-Chip 60-GHz MmWave Sensor Evaluation Module," 2020. [Online]. Available: https://www.ti.com/tool/IWR6843ISK
[28] AsusTek Computer Inc., "Zenfone AR: Go Beyond Reality," 2021. [Online]. Available: https://www.asus.com/us/Phone/ZenFone-AR-ZS571KL/
[29] H. Fan, et al., "A Point Set Generation Network for 3D Object Reconstruction from a Single Image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[30] W. Yuan and T. Khot and D. Held and C. Mertz and M. Hebert, "PCN: Point Completion Network," in *International Conference on 3D Vision (3DV)*, 2018.
[31] Sanjib Sur, "MilliPCD Project," 2022. [Online]. Available: https://syrex.cse.sc.edu/research/ubiquitous-sensing/millipcd/
[32] David M. Sheen and Douglas L. McMakin and Thomas E. Hall, "Three-Dimensional Millimeter-Wave Imaging for Concealed Weapon Detection," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, no. 9, 2001.
[33] M. E. Yanik and M. Torlak, "Near-Field MIMO-SAR Millimeter-Wave Imaging With Sparsely Sampled Aperture Data," *IEEE Access*, vol. 7, 2019.
[34] Soumekh, Mehrdad, "A System Model and Inversion for Synthetic Aperture Radar Imaging," *IEEE Transactions on Image Processing*, vol. 1, no. 1, 1992.
[35] B. Mamandipoor and G. Malysa and A. Arbabian and U. Madhow and K. Noujeim, "60 GHz Synthetic Aperture Radar for Short-Range Imaging: Theory and Experiments," in *IEEE Asilomar Conference on Signals, Systems and Computers*, 2014.
[36] Yanzi Zhu and Yibo Zhu and Ben Y. Zhao and Haitao Zheng, "Reusing 60GHz Radios for Mobile Radar Imaging," in *ACM MobiCom*, 2015.
[37] C. M. Watts and P. Lancaster and A. Pedross-Engel and J. R. Smith and M. S. Reynolds, "2D and 3D Millimeter-Wave Synthetic Aperture Radar Imaging on a PR2 Platform," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
[38] J. Guan and A. Paidimarri and A. Valdes-Garcia and B. Sadhu, "3D Imaging using mmWave 5G Signals," in *IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, 2020.
[39] Barneto, Carlos Baquero and Riihonen, Taneli and Turunen, Matias and Koivisto, Mike and Talvitie, Jukka and Valkama, Mikko, "Radio-based sensing and indoor mapping with millimeter-wave 5g nr signals," in *International Conference on Localization and GNSS (ICL-GNSS)*, 2020.
[40] Li, Xuyou and Du, Shitong and Li, Guangchun and Li, Haoyu, "Integrate Point-Cloud Segmentation with 3D LiDAR Scan-Matching for Mobile Robot Localization and Mapping," *MDPI Sensors*, vol. 20, no. 1, 2020.
[41] Zeng, Yiming and Hu, Yu and Liu, Shice and Ye, Jing and Han, Yinhe and Li, Xiaowei and Sun, Ninghui, "RT3D: Real-Time 3-D Vehicle Detection in LiDAR Point Cloud for Autonomous Driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, 2018.
[42] Labbe, Mathieu and Michaud, Francois, "RTAB-Map as an Open-Source Lidar and Visual Simultaneous Localization and Mapping Library for Large-Scale and Long-Term Online Operation," *Journal of Field Robotics*, vol. 36, no. 2, 2019.
[43] Qian, Kun and Ma, Xudong and Fang, Fang and Yang, Hong, "3D environmental mapping of mobile robot using a low-cost depth camera," in *IEEE International Conference on Mechatronics and Automation*, 2013.
[44] Pal, Bishwajit and Khaiyum, Samitha and Kumaraswamy, Y. S., "3D point cloud generation from 2D depth camera images using successive triangulation," in *International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2017.
[45] Fremont, V. and Chellali, R., "Turntable-based 3D object reconstruction," in *IEEE Conference on Cybernetics and Intelligent Systems*, 2004.
[46] Burschka, D. and Ming Li and Taylor, R. and Hager, G.D., "Scale-invariant registration of monocular stereo images to 3D surface models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.
[47] Seitz, S.M. and Curless, B. and Diebel, J. and Scharstein, D. and Szeliski, R., "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
[48] Zeng, Wei and Karaoglu, Sezer and Gevers, Theo, "Inferring Point Clouds from Single Monocular Images by Depth Intermediation," 2018. [Online]. Available: https://arxiv.org/abs/1812.01402
[49] Navaneet, K L and Mathew, Ansu and Kashyap, Shashank and Hung, Wei-Chih and Jampani, Varun and Babu, R. Venkatesh, "From image collections to point clouds with self-supervised shape and pose networks," 2020. [Online]. Available: https://arxiv.org/abs/2005.01939

[50] Meta, Adriano and Hoogeboom, Peter and Ligthart, Leo P., "Signal Processing for FMCW SAR," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, 2007.
[51] Google, "Project Soli," 2022. [Online]. Available: https://www.google.com/atap/project-soli/
[52] Bang, Jihoon and Hong, Youngtaek and Choi, Jaehoon, "MM-wave phased array antenna for whole-metal-covered 5G mobile phone applications," in *International Symposium on Antennas and Propagation (ISAP)*, 2017.
[53] Parchin, Naser Ojaroudi and Shen, Ming and Pedersen, Gert Fralund, "UWB MM-Wave antenna array with quasi omnidirectional beams for 5G handheld devices," in *IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, 2016.
[54] Palm, Stephan and Sommer, Rainer and Stilla, Uwe, "Mobile Radar Mapping—Subcentimeter SAR Imaging of Roads," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, 2018.
[55] Fang, Zheng and Zhou, Sifan and Cui, Yubo and Scherer, Sebastian, "3D-SiamRPN: An End-to-End Learning Method for Real-Time 3D Single Object Tracking Using Raw Point Cloud," *IEEE Sensors Journal*, vol. 21, no. 4, 2021.
[56] Diebel, James, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, 2006.
[57] Sheshappanavar, Shivanand Venkanna and Kambhamettu, Chandra, "A Novel Local Geometry Capture in PointNet++ for 3D Classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.
[58] Lian, Yanchao and Feng, Tuo and Zhou, Jinliu, "A Dense Pointnet++ Architecture for 3D Point Cloud Semantic Segmentation," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019.
[59] Yang, Y. and Feng, C. and Shen, Y. and Tian, D., "FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[60] Ohtake, Y. and Belyaev, A. and Seidel, H.P., "A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions," in *Shape Modeling International.*, 2003.
[61] Huang, Hui and Wu, Shihao and Gong, Minglun and Cohen-Or, Daniel and Ascher, Uri and Zhang, Hao (Richard), "Edge-Aware Point Set Resampling," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 1, 2013.
[62] Prudhvi Gurram and Shuowen Hu and Alex Chan, "Uniform grid upsampling of 3D lidar point cloud data," in *Three-Dimensional Image Processing (3DIP) and Applications*, 2013.
[63] Yu, Lequan and Li, Xianzhi and Fu, Chi-Wing and Cohen-Or, Daniel and Heng, Pheng-Ann, "PU-Net: Point Cloud Upsampling Network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[64] Tchapmi, Lyne P. and Kosaraju, Vineet and Rezatofighi, Hamid and Reid, Ian and Savarese, Silvio, "TopNet: Structural Point Cloud Decoder," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
[65] Cignoni, Paolo and Callieri, Marco and Corsini, Massimiliano and Dellepiane, Matteo and Ganovelli, Fabio and Ranzuglia, Guido, "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, 2008.
[66] Open-Source, "PyTroch," 2021. [Online]. Available: https://pytorch.org/
[67] NVIDIA, "RTX A6000," 2021. [Online]. Available: https://www.nvidia.com/en-us/design-visualization/rtx-a6000/
[68] Angela Dai and Daniel Ritchie and Martin Bokeloh and Scott Reed and Jurgen Sturm and Matthias Niebner, "ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[69] Cai, Pingping and Sur, Sanjib, "DeepPCD: Enabling AutoCompletion of Indoor Point Clouds with Deep Learning," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 2, 2022.
[70] Ian C. McDowell and Rahul Bulusu and Sanjib Sur, "Poster: MilliDrone: A Drone Platform to Facilitate Scalable Survey of Outdoor Millimeter-Wave Signal Propagation," in *Proceedings of ACM International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2022.