SquiggleMilli: Approximating SAR Imaging on Mobile Millimeter-Wave Devices

HEM REGMI, University of South Carolina, USA
MOH SABBIR SAADAT, University of South Carolina, USA
SANJIB SUR, University of South Carolina, USA
SRIHARI NELAKUDITI, University of South Carolina, USA

This paper proposes *SquiggleMilli*, a system that approximates traditional Synthetic Aperture Radar (SAR) imaging on mobile millimeter-wave (mmWave) devices. The system is capable of imaging through obstructions, such as clothing, and under low visibility conditions. Unlike traditional SAR that relies on mechanical controllers or rigid bodies, *SquiggleMilli* is based on the hand-held, fluidic motion of the mmWave device. It enables mmWave imaging in hand-held settings by re-thinking existing motion compensation, compressed sensing, and voxel segmentation. Since mmWave imaging suffers from poor resolution due to specularity and weak reflectivity, the reconstructed shapes could be imperceptible by machines and humans. To this end, *SquiggleMilli* designs a machine learning model to recover the high spatial frequencies in the object to reconstruct an accurate 2D shape and predict its 3D features and category. We have customized *SquiggleMilli* for security applications, but the model is adaptable to other applications with limited training samples. We implement *SquiggleMilli* on off-the-shelf components and demonstrate its performance improvement over the traditional SAR qualitatively and quantitatively.

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

Additional Key Words and Phrases: Millimeter-Wave, See-through Imaging, Generative Adversarial Networks

ACM Reference Format:
Hem Regmi, Moh Sabbir Saadat, Sanjib Sur, and Srihari Nelakuditi. 2021. SquiggleMilli: Approximating SAR Imaging on Mobile Millimeter-Wave Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 125 (September 2021), 26 pages. https://doi.org/10.1145/3478113

1 INTRODUCTION

Millimeter-wave (mmWave) systems enable through-obstruction imaging and are widely used for screening in state-of-the-art airports and security portals [1, 2]. They can detect hidden contrabands, such as weapons, explosives, and liquids, by penetrating wireless signals through clothes, bags, and non-metallic obstructions [3]. Besides, mmWave imaging systems could enable applications to track beyond line-of-sight [4–7], see through walls [8–10], recognize humans through obstructions [10–12], and analyze materials without contaminating them [13]. MmWave systems also have advantages over other screening modalities: Privacy preservation and low-light condition usages over optical cameras; very weak ionization effect over X-Ray systems; and shape detection of non-metallic objects over metal detectors. Furthermore, the ubiquity of mmWave technology in

Authors' addresses: Hem Regmi, hregmi@email.sc.edu, University of South Carolina, USA; Moh Sabbir Saadat, msaadat@email.sc.edu, University of South Carolina, USA; Sanjib Sur, sur@cse.sc.edu, University of South Carolina, USA; Srihari Nelakuditi, srihari@sc.edu, University of South Carolina, USA.

2474-9567/2021/9-ART125 $15.00
https://doi.org/10.1145/3478113

5G-and-beyond devices enable opportunities for bringing imaging and screening functionalities to hand-held settings. Hidden shape perception by humans or classification by machines not only will enable applications, such as in-situ security check without pat-down search, baggage discrimination without opening the baggage, packaged inventory item counting without intrusions, discovery of faults in water pipes or gas lines without tearing up walls, *etc.*, but also will enable flexible, multi-purpose functionalities on 5G mobile networking devices.

Traditional mmWave imaging systems operate under the Synthetic Aperture Radar (SAR) principle [14–19]. They use bulky, mechanical motion controllers or rigid bodies that move the mmWave device in a pre-determined trajectory forming an *aperture* [1, 2, 14]. As it moves along the aperture, the device transmits a wireless signal and measures the reflections bounced off of the nearby objects. Combining all the reflected signals *coherently* across the known trajectory allows the system to discriminate the objects with higher reflectivity against the background noise. The spatial resolution of the final 2D or 3D shape depends on the span of the apertures in horizontal and vertical axes and the bandwidth of the system [16, 20]. Besides, the reflections also need to be collected from uniformly and densely spaced measurement locations to avoid aliasing in the object's shape [14]. Mechanical controllers or rigid bodies are essential for satisfying such constraints in practice.

However, emulating the SAR principle on a hand-held mmWave device is challenging for two key reasons.

First, in the absence of a mechanical controller, a user would be required to move the device with a manual, fluidic hand motion. Such movement would introduce two issues: Non-linearity and non-uniformity in the aperture. While motion non-linearity could potentially be addressed by leveraging existing motion compensation techniques [16], non-uniformity of the measurement locations depends solely on the hand-held movement. Slowly moving the hand with fast signal sampling by the device could increase the measurement density, but the method not only is time-consuming but also does not ensure that the locations are distributed uniformly. Furthermore, appropriate SAR focusing requires the knowledge of the object's depth, and focusing at an arbitrary depth yields de-focused, blurry shapes. So, without addressing these issues, the measurements from the hand-held setting would prohibit focusing the signals appropriately and retrieving the object's shape correctly.

Second, mmWave signals are highly specular due to their small wavelength, *i.e.*, many objects introduce mirror-like reflections [21, 22]. Thus, the effective strength of the reflections from various parts of the object depends highly on its orientation, *w.r.t.* the aperture plane. So, even if some parts of the object could reflect mmWave signal strongly, those reflections may not arrive at the receiver. Consequently, some parts and edges of the object do not appear in the reconstructed mmWave shape. What's more, due to the weak reflectivity of various materials, its reflected signals may be buried under the signals from strong reflectors. Thus, the weak reflecting parts of the object may have poor, blurry resolution, or often be missing from the final shape completely, allowing for a partial shape reconstruction only. The resultant shape could lack discriminating features for automatic object classification as well as could be imperceptible by humans.

We propose *SquiggleMilli*, a system that enables high-quality mmWave imaging under hand-held settings by overcoming these fundamental challenges. *SquiggleMilli* relies only on the hand-held movement of the mmWave device to measure the reflected signals from objects. It employs a three-dimensional mmWave imaging framework that can retrieve the 2D shape of hidden objects viewed from the hand-held movement plane and the objects' 3D features, such as mean depth and orientation in 3D plane. Rather than asking users to collect uniformly and densely spaced measurements, *SquiggleMilli* lets the user freely *squiggle* the device over the air in front of the target scene. Then, by processing the reflected signals, *SquiggleMilli* outputs human perceivable and interpretable 2D shapes, 3D features, and categories for all reflecting objects, hidden or in line-of-sight, in the scene in front of the *squiggle* motion plane. To achieve this, *SquiggleMilli* employs two core design techniques:

(1) It leverages the camera system in the hand-held device to self-localize the relative locations of the *squiggle* motion path and applies multi-antenna based optimization to estimate reflected signals in a majority of the uniform grid locations. Then, *SquiggleMilli* exploits a compressed sensing based technique on the reflected signals to recover the samples missing from some of the uniform locations. Since mmWave wireless signal exhibits high

signal sparsity, this method can converge quickly and estimate the missing samples accurately. Finally, on the reconstructed volume, *SquiggleMilli* applies a multi-focusing and voxel segmentation to reconstruct the 3D shape of individual objects without their prior depth information.

(2) Even if the reconstructed shapes could be missing high spatial frequency, such as the edges, due to specularity and weak reflectivity, the low-frequency, partial shape information provides opportunities for a machine learning model to improve the shape quality effectively and classify objects automatically. *SquiggleMilli* is inspired by the existing works in enhancing low resolution visual images to high resolution using conditional Generative Adversarial Networks (cGAN) [23, 24], and it aims to not only improve the resolution but also restore the missing high spatial frequency information. Finally, *SquiggleMilli* uses the fully reconstructed shapes to quantify their 3D features and classify them into categories. Although in this work, *SquiggleMilli*'s classifier is customized for hand-held security applications, the class labels could be adapted by re-training and fine-tuning the networks with limited samples for different applications, such as packaged inventory counting.

We have prototyped *SquiggleMilli* on an off-the-shelf mmWave device and conducted field experiments with multiple objects and multiple *squiggle* motions to verify its performance. Due to the lack of large-scale real training data, *SquiggleMilli* is trained in two phases: Large-scale training with synthesized data and fine-tuning with real, measured data. We build a realistic data synthesizer that uses CAD models of various objects and generates 3D mmWave shapes. Our synthesized dataset consists of 9800 samples (14.7 GB) of various objects' shapes generated by *squiggle* motion paths. Furthermore, we have built a real-world data collection platform that integrates a Google Tango based AR device [25] with a 77–81 GHz mmWave device [26] to collect real, hand-held *squiggle* motion based mmWave reflected signals from objects. Our real dataset consists of 1568 samples (2.4 GB) with 9 object categories, each with an average 10 sub-categories. These real measurements are used to both fine-tune our machine learning models and benchmark the effectiveness of the two design components. We find that *SquiggleMilli* can reconstruct the 2D shapes with a similarity score ranging from 0.85 to 0.95 (1 is a perfect match) *w.r.t.* to the ground-truth shape. For 3D features, *SquiggleMilli* can predict the mean depth with less than 1% error and rotation angle below 1.5° error for 90^{th} percentile of shapes. It can classify the objects in categorical and binary labels with an average 90% and 96% accuracy, respectively.

In sum, we make the following contributions: (1) We design a framework for mmWave shape reconstruction in hand-held settings by re-thinking the traditional motion compensation, compressed sensing, and voxel segmentation. To the best of our knowledge, *SquiggleMilli* is the first system to enable 3D mmWave imaging with a free-hand, *squiggle* device motion. (2) We design customized deep convolution networks to not only improve the shape resolution but also recover the missing spatial frequencies, retrieve 3D features, and categorize the objects automatically. Our results demonstrate that *SquiggleMilli* is generalizable under real conditions with different background noise and environmental movements. To catalyze the hand-held mmWave imaging research, we will open-source the measured dataset, data synthesizer, and cGAN implementation through our project repository.

2 BACKGROUND AND CHALLENGES

2.1 3D Millimeter-Wave Image Reconstruction

Traditional SAR imaging relies on a moving antenna that periodically transmits and receives Frequency Modulated Continuous Wave (FMCW) signals. The transmitter sends the signal during a short time period, and the receiver measures the reflections bounced off of the nearby objects. The objects' shapes could be discriminated against the background by estimating the reflection strengths at different distances. To obtain the object's 3D shape, the traditional SAR system moves the device in a 2D grid and measures the reflections at uniformly and densely spaced grid locations [14]. Consider that the object consists of a set of N reflecting points, each at a coordinate (x_n, y_n, z_n) with reflectivity σ_n. As long as the imaging system can estimate the reflectivity σ_n at the correct coordinate, it can recover the object's shape. The FMCW device, moving along a 2D grid, sends a chirp signal $p(t)$

from its location $(x, y, 0)$ and receives the reflection. The reflected signal consists of reflections from all object points and can be modeled as: $s(x, y, t) = \sum_{n \in N} \sigma_n \cdot p[t - 2d_n/c]$, where c is the wireless propagation speed ($\sim 3{\times}10^8$ m/s), and $2d_n$ is the round-trip distance between the n^{th} reflecting point and the measurement location [16]. The SAR imaging system then converts and combines the measured reflections from all 2D grid locations to construct the 3D shape of the object. *First*, it applies two-successive Fourier Transforms, a 1D FFT across the time t, and a 2D FFT across the space x and y, to obtain the spatial frequencies in the object [14]:

$$s(x, y, \omega) = \mathbf{FFT}_t[s(x, y, t)] \qquad\qquad s(k_x, k_y, \omega) = \mathbf{FFT}_{(x,y)}[s(x, y, \omega)] \tag{1}$$

where (k_x, k_y) are the spatial frequencies in the 2D grid, and ω is the temporal frequency [14]. *Then*, it focuses the frequencies at the mean depth of the object z_0 by applying a matched filter on $s(k_x, k_y, \omega)$ [14]:

$$F(k_x, k_y, k_z) = s(k_x, k_y, \omega)e^{-jk_z z_0} \tag{2}$$

where k_z is the spatial frequency across z (depth). Since the mean depth of the object is unknown before reconstructing the shape, the system uses the centroid of the reconstruction volume as the mean depth. *Finally*, it applies a 3D Inverse Fourier Transform to obtain the reflectivity of the object at different (x_n, y_n, z_n).

$$f(x, y, z) = \mathbf{IFFT}_{(k_x,k_y,k_z)}[F(k_x, k_y, k_z)] \qquad and, \sigma_n = |f(x_n, y_n, z_n)| \tag{3}$$

where $|\cdot|$ denotes the absolute value, *i.e.*, the strength of the reflecting points. Even if the measurement locations could be uniformly spaced by precise mechanical movement of the device, $F(k_x, k_y, k_z)$ is non-uniformly sampled across the third dimension. This is because k_z is a non-linear function of k_x, k_y, and ω [14]. Thus, a straightforward **IFFT** in Eq. (3) does not work. To overcome the challenge, the SAR system applies interpolation on the spatial frequency signal $F(k_x, k_y, k_z)$ before obtaining a focused 3D image [14]. Besides, the imaging framework requires the uniformly spaced grid locations to adhere to the Nyquist criterion so that the reconstruction is alias-free [14]. Figure 1(a) shows an example 2D slice output of the uniformly and densely spaced SAR image reconstruction with a 77 GHz mmWave device [26]. The object is a 12 cm long scissor placed 30 cm away from the 2D grid.

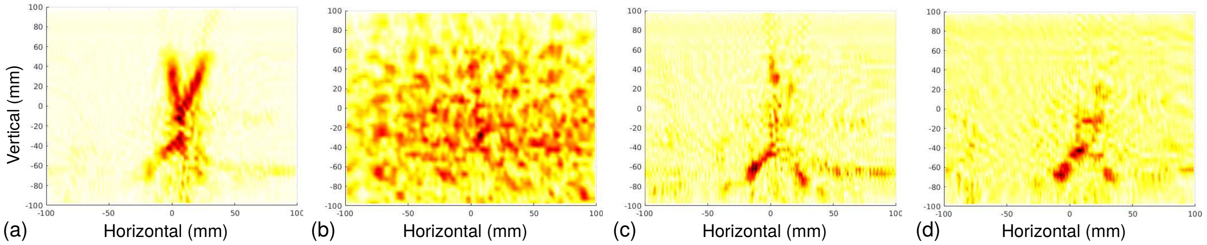

Figure 1. Shape reconstruction with traditional SAR: (a) Uniform and dense 2D grid based aperture. (b) Hand-held motion based aperture. (c) Uniform and dense 2D grid but focused at a depth off by 2 cm. (d) Object with 5° incorrect alignment *w.r.t.* the aperture plane. Traditional SAR in hand-held setting distorts and blurs the shape and misses its features.

2.2 Challenges in Hand-held Settings

In practice, emulating SAR on a hand-held mmWave system is challenging for two reasons:

(1) **Non-linear, Undersampled Measurement Locations**: The image reconstruction in Section 2.1 depends on uniformly and densely spaced measurement locations on the 2D grid. Unfortunately, both the requirements are impractical under hand-held settings. Even if the device's relative locations could be precisely estimated, (following [25, 27]), focusing the reflected signals measured from non-linear movements would produce a distorted shape. Furthermore, obtaining dense measurements, especially along the vertical axis, is challenging and time-consuming. Undersampling of measurements in space creates shape alias in the scene, resulting in ghost objects. To understand such effect, we try to emulate the movement along 2D grid using the hand-held device and apply

the traditional 3D SAR imaging on the measured reflections of the same scissor object. Figure 1(b) shows the resulting object shape: It is completely distorted, showing multiple spurious, strong reflections at incorrect coordinates, forming ghost objects where no object exists. Besides, focusing the reflected signals at an unknown depth yields a de-focused shape. Figure 1(c) further shows that the shape is de-focused even with reflections from the uniform and dense 2D grid, when signals are focused at depth off by just 2 cm.

(2) **Variable Reflectivity and Specularity**: Various parts of an object reflect mmWave signals differently. While the metallic part would likely reflect strong signals, the reflections from the non-metallic parts could be buried under the additivity of all reflections. Besides, due to the small wavelength of the mmWave signal and specular reflections from objects, 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 shape reconstruction only. Figure 1(d) illustrates this effect by reconstructing the shape when the object is aligned incorrectly by 5° only *w.r.t.* the 2D grid, and contrast the result with the reconstructed shape in Figure 1(a). We have two observations: *First*, even if in Figure 1(a), the object is perfectly aligned, various edges and parts are missing from the non-metallic parts, such as the scissor handle. *Second*, for incorrectly aligned scissor in Figure 1(d), the specular reflections prohibit reconstructing all of the metallic parts, such as the scissor blades, and misses important object features.

3 *SQUIGGLEMILLI* DESIGN

3.1 Overview

SquiggleMilli aims to bring SAR imaging to cheap, ubiquitous mobile mmWave devices by addressing the practical challenges in hand-held settings. It relies on the user freely moving the device in the air in a *squiggle* manner and measures the reflected signals from the scene in front of the *squiggle* plane. Then, by processing the signals, *SquiggleMilli* emulates the traditional SAR imaging system as if the reflections were measured from uniformly and densely spaced grid locations. Still, the fundamental limits of specularity and weak reflectivity of mmWave signals yield poor resolution and only allow for partial shape reconstruction. To this end, *SquiggleMilli* uses cGAN, an adversarial learning framework, to not only improve the resolution but also restore the missing parts in the object's shape. Figure 2 shows an overview of the *SquiggleMilli* system.

The reflected signals and *squiggle* aperture locations are used in a non-linear motion compensation that maps *squiggle* locations to the nearest uniform grid locations. Still, the number of *squiggle* locations may not be enough to map to all the uniform grid locations. To this end, *SquiggleMilli* leverages the sparse reflection properties of the mmWave domain and applies a compressed sensing based framework [13] to recover the missing samples. Then, it focuses the reflected signals at different depths from the *squiggle* plane and applies voxel segmentation to extract 3D mmWave shape of individual objects.

Since specularity and weak reflectivity may not allow reconstructing the full shape, *SquiggleMilli* designs a full shape recovery and automatic classification framework. It leverages a pre-trained model using cGAN that learns, from thousands of previous examples, the association between the 3D mmWave shape to its ground-truth shape. This model is generalizable for various objects and can determine an object's accurate 2D shape given the partial 3D mmWave

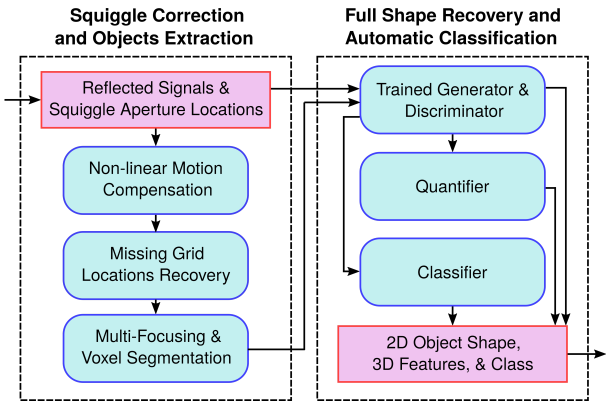
Squiggle Correction and Objects Extraction
Full Shape Recovery and Automatic Classification
Reflected Signals & Squiggle Aperture Locations
Non-linear Motion Compensation
Missing Grid Locations Recovery
Multi-Focusing & Voxel Segmentation
Trained Generator & Discriminator
Quantifier
Classifier
2D Object Shape, 3D Features, & Class

Figure 2. System overview of *SquiggleMilli*.

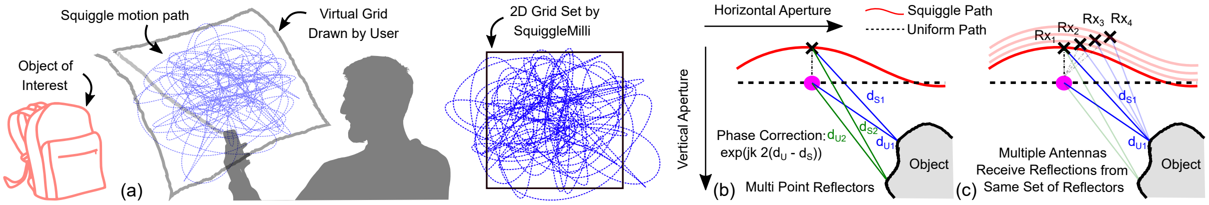

Figure 3. (a) User draws a virtual 2D grid and *squiggles* the device. (b) Non-linear motion compensation maps *squiggle* locations to nearest uniform grid locations. (c) Multiple antennas receive reflections from the same set of reflectors.

shape. Besides, the framework can quantify various 3D features of the object, *e.g.*, its mean depth and orientation in 3D plane, and classify the objects into different categories. We now describe these design components in detail.

3.2 Squiggle Correction and Objects Extraction

Traditional 3D SAR imaging ensures accurate focusing and no aliasing of the shape by requiring the 2D grid locations to be spaced within $\lambda/2$ distance, where λ is the signal wavelength [14] (for example, $\lambda \approx 3.9$ mm for 77 GHz). Even if the camera system, such as in AR smartphones [25], could self-localize the device within that precision, such uniform and dense measurement constraints are impractical in hand-held settings. Thus, instead of relying on this constraint, *SquiggleMilli* uses whatever the user could measure with the device's *squiggle* motion in the air and applies corrections. Still, the reconstructed shape resolution is fundamentally limited by the aperture span in both horizontal and vertical axes, and the number of measurement locations. So, *SquiggleMilli* would like to maximize the measurements to improve its focusing ability.

To this end, it leverages the AR camera service to guide the user through visual aid: Camera feed overlaid with the device's locations on the screen, such as in Google Tango [25, 28]. *First*, as the user points her hand-held device towards a scene, *SquiggleMilli* asks the user to draw a virtual boundary of a 2D grid within which she will *squiggle* her device (Figure 3[a]). This virtual boundary helps the AR system to continuously track and provide feedback when the user is overshooting out of the area. *Second*, as the user moves the device over the air, the overlaid display helps her see the *squiggle* locations in real-time, implicitly prompting her to collect more measurements at sparser areas, similar to existing Tango services [28]. *Finally*, when the user stops, *SquiggleMilli* sets a maximum area boundary and minimum grid resolution. The 2D area is set based on the density of the measured locations above a threshold, and the resolution is selected based on the condition to avoid shape aliasing [14]. Once *SquiggleMilli* has collected all the reflected signals, aperture locations, and determined the boundary and resolution of the virtual 2D grid, it aims for: (1) *Squiggle* correction with non-linear motion compensation; (2) Missing grid locations recovery; and (3) Object extraction with multi-focusing and voxel segmentation.

Non-linear Motion Compensation: The core purpose of the motion compensation is to map as many measurements on the *squiggle* motion path as possible to its nearest uniform grid location. First, let's consider a point reflector on the target object in Figure 3(b) with distance d_s from the measurement location on the *squiggle* path (black ×). From the measured sample $s(x, y, \omega)$ at the point on the *squiggle* path, we will need to estimate the equivalent sample $s_u(x_u, y_u, \omega)$, which would be received at the nearest point on the uniform grid (pink •) with distance d_u between the point reflector and the point on the uniform grid. This is achieved through a phase correction of the samples. Since the signal traverses twice the distance between the *squiggle* location and point reflector, the reflector will contribute to the phase change of the reflected signal by $e^{jk \cdot 2d_s}$, where k is the wavenumber ($k = 2\pi/\lambda$). Similarly, the point reflector will contribute to the phase change at the uniform location by $e^{jk \cdot 2d_u}$. We can then virtually move the *squiggle* location to the uniform location by correcting for these phase changes: Subtracting the phase change at the *squiggle* location and adding the phase change at the uniform location. Thus, we can estimate $s_u(x, y, \omega)$ from $s(x, y, \omega)$ as follows:

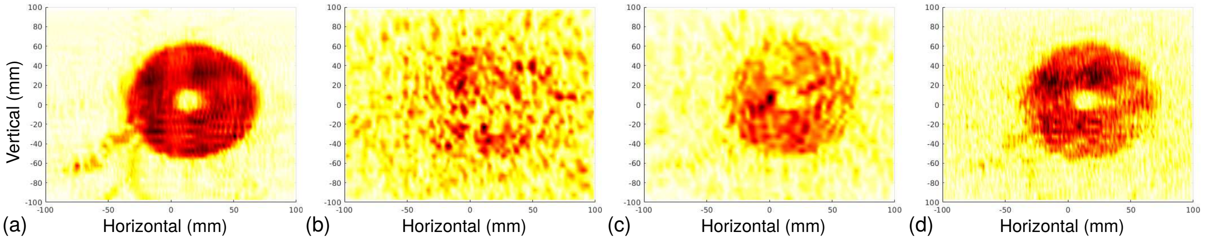

Figure 4. (a) Ground-truth shape (of a CD) from a perfect 2D grid of measurements. Shape reconstruction results: (b) With motion error; (c) With motion compensation; (d) With motion compensation and recovery of missing grid locations.

$$s_u(x_u, y_u, \omega) = s(x, y, \omega) \cdot \frac{e^{jk\cdot 2d_u}}{e^{jk\cdot 2d_s}} = s(x, y, \omega) \cdot e^{jk\cdot 2(d_u - d_s)} \tag{4}$$

Unfortunately, such a straightforward recovery does not work in practice for two reasons. (1) The object is not a single point reflector but consists of many reflecting points (Figure 3[b]); each point contributes to various phase changes, and the *squiggle* location measures only the sum contribution. (2) The phase correction only works if the total required correction is less than 2π, *i.e.*, the absolute distance difference $|d_u - d_s|$ is less than $\lambda/2$. To overcome these challenges, *SquiggleMilli* leverages two opportunities: *First*, mmWave devices usually comprise of multiple receive antennas that can measure the reflections simultaneously. Due to the antenna separation, the measured signals are different, but the reflections come from the same set of points on the object (Figure 3[c]). Thus, accurate phase correction for each antenna from the *squiggle* location towards the same uniform grid location should yield the same reflected signal. Said differently, if we can perform accurate phase correction of each antenna's signals, the resultant differences between recovered signals from any pair of antennas would be close to zero. *Second*, practical objects are not a collection of random points, but can be considered as a collection of "patches" with uniform reflectivity and phase change contributions [18]. So, the phase corrections are only needed for the contribution of the individual patches.

Thus, observations from multiple receive antennas and limited unknowns with "patch" assumption allow *SquiggleMilli* to formulate the phase correction as an optimization problem. Given any pair of receive antennas $\{i, j\}$, assume that the corresponding phase corrected reflected signals on the uniform grid location from the antennas are $s_u^i(x_u, y_u, \omega)$ and $s_u^j(x_u, y_u, \omega)$, respectively. Since under a perfect phase correction, these two reflected signals should be identical, the optimization problem could be modeled as:

$$\min \sum_{i\in N, j\in N, i\neq j} \left| s_u^i(x_u, y_u, \omega) - s_u^j(x_u, y_u, \omega) \right|_{k=\omega/c} \quad s.\ t. \quad |d_u - d_s| < \lambda/2 \tag{5}$$

where N is the total number of receive antennas on the mmWave device. This optimization is applied to each temporal frequency bin ω that comprises reflected signals from different distances. However, the number of patches in the scene and the corresponding distances d_u and d_s are unknown. Thus, *SquiggleMilli* sets a bounding box of the 3D volume of the scene it will reconstruct and applies the optimization iteratively over the small neighborhood voxel area. Figures 4(a–c) show an example 2D shape (of a CD) from the multi-antenna motion compensation (Figure 4[c]) and compare it with no motion error compensation (Figure 4[b]). Clearly, the motion compensation improves the quality, but could not produce as high quality shape as the ground-truth (Figure 4[a]) because not all *squiggle* locations could be compensated due to the constraint in Eq. (5).

Missing Grid Locations Recovery: Although the motion compensation could map the *squiggle* locations to the nearest uniform locations, it alone does not ensure estimating the reflected signals at all the uniform locations. Even if a user could *squiggle* multiple times to increase the scan density, some *squiggle* locations could still map onto the same grid locations; hence, some of the uniform grid locations may remain unavailable. Focusing the reflected signals with missing grid samples would cause shape aliasing [14]. ***So, before focusing,* SquiggleMilli *attempts to recover the missing reflected signals through the compressed sensing (CS) framework*** [29–32]. Since mmWave signals are sparse in the reflected signal domain, the CS technique could estimate the missing samples based on the spatial arrangement and adjacency of the uniform grid locations [13, 33].

The key intuition is that even if the uniform grid is missing a few samples, combining several measurements around the missing location could yield an accurate prediction since a majority of objects in mmWave spatial frequency domain has high-degree of sparsity [34–37]. Equation 6(i) illustrates the sparsity of the reflected signal, $s \in \mathbb{C}^{N \times 1}$. This is decomposed on a sparsifying matrix Ψ to a sparse representation f. Ψ is an $N \times N$ sparsifying matrix, and f ($\in \mathbb{C}^{N \times 1}$) is the sparse signal. f is called *k-sparse* if there are only k significantly large samples. Since we have missing samples from our reflected signal y, this can be represented as Equation 6(ii), where Φ is the measurement matrix. If we have only M of the total N samples in y ($M < N$; N - M missing samples), then Φ is an $M \times N$ matrix. Equation 6(ii) also represents the relation between y and f through A, where $A = \Phi\Psi$. Thus, 6(ii) becomes an under-determined system of linear equations with infinitely many solutions. However, when $k \ll N$, the vector f can be recovered with high reliability [38], and Equation 6(i) can then be used to recover the full signal s. We formulate the recovery of f as an L1-norm minimization problem, as shown in Equation 6(iii). *SquiggleMilli* leverages the 1D sparse recovery technique in [39] and extends it for the 2D. Considering the motion compensated samples on the uniform grid locations as the "compressed measurements," *SquiggleMilli* applies the *Discrete Cosine Transform (DCT)* as the sparsifying matrix. Then, the L1-norm minimization (Equation 6[iii]) is solved by the unconstrained basis pursuit de-noising method in [39].

$$(i) \quad s = \Psi f; \qquad (ii) \quad y = \Phi s = Af; \qquad (iii) \quad \min ||f||_1 \quad s.\ t. \quad y = Af \tag{6}$$

However, there are two practical challenges in applying the CS technique in *SquiggleMilli*. *First*, the L1-norm based minimization typically fails to converge, or outputs unreliable estimation, if the missing sample locations are not randomized enough. This could be an issue with an unguided scan since users could often be biased to scan over a certain grid area, leaving other areas sparser. Fortunately, the visual aid in *SquiggleMilli*, with an overlaid camera feed and device's locations, helps the user randomize the *squiggle* and distribute the measurement locations throughout the virtual 2D grid. In case our CS recovery fails to converge, *SquiggleMilli* can prompt the user for a repeat scan too. *Second*, since *SquiggleMilli* receives the reflected signals using a wide-bandwidth mmWave device, each *squiggle* location's signals are too unwieldy for efficient CS recovery, and oftentimes recovery fails to converge. However, the signals include reflections from all objects in the scene, even far away. For example, with 256 samples in each *squiggle* location and 4 GHz bandwidth, our system is capable of measuring reflections up to 9.6 m. Clearly, objects from such a far away distance may not only reflect very weak signals but also be irrelevant for short-range imaging applications. Thus, *SquiggleMilli* considers a maximum range and removes the samples that are beyond the threshold. In our evaluation (with 1350 test samples), CS reconstruction always converges by setting a maximum range of 4 m. Figure 4(d) shows the output image with the CS technique.

Multi-Focusing and Voxel Segmentation: Appropriate focusing of the signals requires the knowledge of the mean depth of the object (Eq. (2)). However, a practical scene may consist of multiple objects at different depths, and focusing the signals at a single mean depth will not only yield poor shape resolution but also blur some of the objects. A straightforward approach could be first to focus the signal at the mean depth, then apply existing voxel segmentation to separate multiple objects [40], and finally, apply post-processing to improve sharpness. But this approach does not work because focusing signals at an incorrect depth suppresses some of the high spatial frequency features [41]. Instead, **SquiggleMilli *focuses the signals multiple times, each time extracting out a strongest reflecting object and subtracting its contribution from measured reflections.***

First, *SquiggleMilli* uses the mean depth, estimated from the median time-of-flight of the reflected signals, to reconstruct a volume following Section 2.1. Since this volume may consist of multiple objects, *SquiggleMilli* applies a *k-means* clustering [42] with 2 clusters to separate out the object and the background voxels. Since objects likely have higher reflections than the background, one of the clusters would consist of only the voxels from multiple objects. *Then*, *SquiggleMilli* leverages the existing *DBSCAN* segmentation [40] within the object cluster to automatically segment multiple objects and sort the segments based on the sum energy of voxels. So, the first segment likely consists of a de-focused object with the strongest reflections. *SquiggleMilli* finds the first segment's voxel centroid, applies processes in reverse order in Section 2.1 to get back the reflected signals corresponding to the strongest object only, and then applies focusing at the correct depth. *Finally*, *SquiggleMilli* subtracts the contributions of the strongest object from the original reflected signal, and the process repeats . . ., until *k-means* could no longer separate the background and object clusters.

Still, effective *DBSCAN* segmentation relies on two search parameters [40]: Minimum number of voxels in each segment, μ; and the radius of "neighborhood" around a voxel, ϵ. While [40] proposed a heuristic for computing appropriate ϵ, μ is application/context-dependent. The challenge is that setting a too large μ would enforce *DBSCAN* to put multiple objects to the same segment, and setting a too small μ would create many segments. To balance between the choices, we set μ to the number of voxels corresponding to the smallest object that we can reconstruct with the resolution limit [14]. Still, this method could generate separate small segments for the same object. *SquiggleMilli* merges these small segments based on whether the closest points of approach for each pair of segments are above the resolution criteria. For example, with 20×20 cm^2 aperture and 4 GHz bandwidth, the resolutions across the horizontal, vertical, and depth axes are 9.49 mm, 9.49 mm, and 3.75 cm, respectively. The resulting segments now contain the 3D mmWave shape of the individual objects in the target scene.

3.3 Full Shape Recovery and Automatic Classification

The reconstructed 3D shapes from Section 3.2 may not always be human perceptible due to specularity and weak reflectivity of objects. For example, Figures 1(a) and (d) show example cases where a scissor could be perceivable in (a), but not in (d) because it is missing a majority of the edges and parts. To improve the human perceptibility of the mmWave shapes, we propose to use cGAN [23, 24, 43]. The high-level idea is intuitive: *SquiggleMilli* trains a cGAN framework by showing thousands of examples of mmWave shapes from *squiggle* correction and reconstruction and the corresponding ground-truth shapes. cGAN framework uses a *Generator* **G** to learn the association between the 3D mmWave shape to the 2D ground-truth shape, and uses a *Discriminator* **D** that teaches **G** to learn better association at each iteration [43]. During the run-time, when cGAN has been trained appropriately, **G** can estimate an accurate 2D depthmap outlining the shape without the ground-truth. In addition to the shape, we also use a *Quantifier* **Q** that predicts the mean depth and orientation in the 3D plane, and a *Classifier* **C** to automatically classify the objects into different categories. In what follows, we first describe the GAN fundamentals briefly and then discuss the network components in detail.

GAN Fundamentals: Generative modeling is the classical machine learning area where unsupervised learning could be used to automatically discover and learn the regularities and patterns in input data; so that the model can generate a new dataset, plausibly correlated with the original dataset. GAN advances the concept further by re-structuring the problem as a supervised learning and improving the quality of the new outputs, and even producing outputs in different domains than inputs [44]. GAN uses two sub-models: (1) *Generator* **G**, which it trains to generate new examples; and (2) *Discriminator* **D**, which tries to discriminate examples as either real (from ground-truth) or generated (by **G**) and outputs the probability that the example is real. The problem is formulated as a zero-sum, adversarial game [45], until **D** could no longer discriminate between real or generated examples, which indicates that **G** is trained enough to be able to generate plausible examples. Mathematically, if $V(\mathbf{D}, \mathbf{G})$ is the expected value in the GAN architecture, then the objective function could be modeled as [44]:

$$\min_{\mathbf{G}} \max_{\mathbf{D}} V(\mathbf{D}, \mathbf{G}) = \mathbb{E}_{x \sim p_{data}(x)}[\log \mathbf{D}(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - \mathbf{D}(\mathbf{G}(z)))] \tag{7}$$

Here the first term represents the expected value from **D** correctly identifying real samples, *i.e.*, ground-truth, and the second term represents the expected value from incorrectly identifying generated samples, *i.e.*, generated by **G**. The probabilities that the samples being drawn from the real or the generated dataset are denoted by $p_{data}(x)$ and $p_z(z)$, respectively. With no condition provided to the generative model, there is no way to control the modes of data being generated or restrict it to a certain domain [44]. Therefore, in *SquiggleMilli*, we propose a conditional GAN (cGAN) based architecture [43], where the ground-truth dataset is only restricted to the mmWave generated shapes and shape output is conditioned on the visual ground-truth shape.

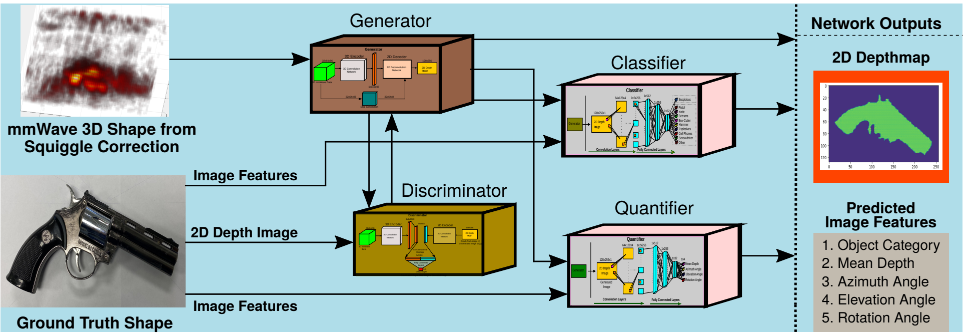

Figure 5. Overview of the *SquiggleMilli* learning model.

***SquiggleMilli* Learning System**: Figure 5 shows the machine learning model in *SquiggleMilli*. The model consists of 4 network blocks: Generator (**G**), Discriminator (**D**), Quantifier (**Q**), and Classifier (**C**). **G** and **D** networks together constitute the cGAN architecture that generates the full object shape. **Q** network leverages the cGAN outputs and ground-truth image features to learn and predict the mean depth and the orientation of the object in the 3D plane. Finally, **C** network leverages the cGAN outputs and supervised class labels to learn and classify the objects into different categories automatically.

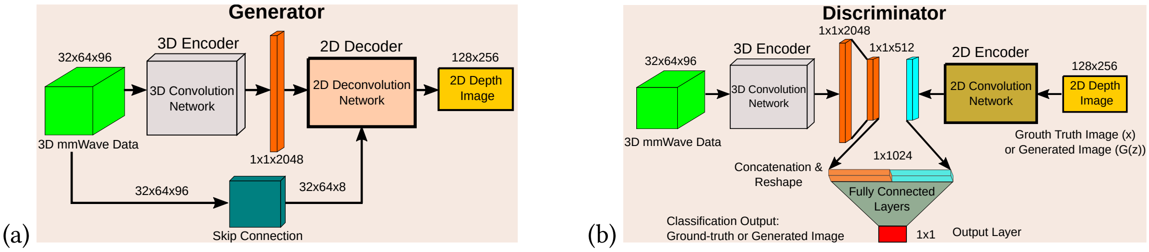

Figure 6. (a) Generator and (b) Discriminator networks of the *SquiggleMilli* system.

Generator: The core purpose of the *Generator* **G** is to convert the imperceptible 3D mmWave shape to a human perceivable 2D shape with all the edges, parts, and high spatial frequencies. To this end, we use the traditional encoder-decoder architecture [46]. The encoder layer converts the 3D mmWave shape into 1D feature vector using multiple 3D convolution layers and an end flatten layer; this 1D representation compresses the 3D shape so that the deeper layers could learn the high-level abstract features. By the end of 3D convolutions, we convert the spatial 3D data to 1×1×1, and at this point, the number of channels has been increased to hold these abstract features. The decoder layer leverages these 1D features, and applies multiple deconvolution layers to decrease the number of channels and increase the spatial dimensions. Deconvolution stops when we reach the desired output size, and at that point, we have a single channel for a 2D shape. In our design, we follow [47] to use 6 3D convolution layers and the 8 2D deconvolution layers at the encoder and the decoder, respectively (Figure 6[a]). However, our training set of samples could not exhaustively account for all possible hand-held *squiggle* motions. To generalize **G** to many *squiggle* possibilities, we concatenate 3D gaussian noise layer to the convolutional layers to boost the network immunity [48].

Yet, passing the 3D mmWave shape through the encoder-decoder layers in the network may yield a loss of detailed high-frequency information during encoding [49]. This is because the object could spread over the reconstructed volume, but only a few 2D slices contain the high spatial frequencies; however, the encoder compresses them while converting the 3D shape into abstract 1D features. ***To preserve such high-frequency details,*** G ***employs a skip connection [47, 49] between the input layer to the 6th deconvolutional layer***. The skip connection extracts the highest energy 2D slice from the 3D shape and concatenates it to the 2D deconvolution layer. However, due to different orientations of the object, various parts of it may not appear at a single highest energy slice; thus, a single 2D slice may not capture all the relevant high-frequency depth information and might cause instability in the network [47]. Therefore, **G** first finds the plane that intersects with the 3D voxel and likely has the highest energy from the object. Then, it selects a few neighboring 2D slices parallel to the highest-energy plane towards and away from the *squiggle* plane. In practice, 4 neighboring slices from both sides of the highest energy plane perform well. Finally, **G** leverages the feedback from the *Discriminator* to adjust the weights of its encoder-decoder layers to learn and predict the accurate 2D shapes. Table 1 summarizes the **G** network parameters.

Table 1. Generator Network Parameters. 3DC: 3D Convolution (with batch normalization); 2DDC: 2D DeConvolution (with batch norm.); Act. Fcn: Activation Function; LRelu: LeakyRelu; Output layer uses linear activation.

	3DC1	3DC2	3DC3	3DC4	3DC5	3DC6	2DDC1	2DDC2	2DDC3	2DDC4	2DDC5	2DDC6	2DDC7	2DDC8	Output
Filter #	16	32	64	128	256	1024	1024	512	256	128	64	16	8	1	
Filter Size	6x6x6	6x6x6	6x6x6	6x6x6	6x6x6	6x6x6	4x3	4x4	4x4	4x4	4x4	4x4	4x4	4x4	
Dilation	2x2x2	2x2x2	2x2x2	2x2x2	2x2x2	2x2x2	1x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2	
Act. Fcn	LRelu	LRelu	LRelu	LRelu	LRelu	LRelu	Relu	Relu	Relu	Relu	Relu	Relu	Relu	Relu	Linear

Discriminator: The purpose of the *Discriminator* **D** is to teach **G** a better association between the 3D mmWave shape and its 2D ground-truth shape. **D** achieves this by distinguishing real and generated samples during the training process. It takes two inputs in the form of the 3D mmWave shape and the 2D shape that either is a real shape or is generated by **G** and produces output as a probability that the input is real (Figure 6[b]). Recall that the goal of **D** is to increase the expected value from correctly discriminating between real and generated samples (Eq. (7)). To this end, **D** uses a similar architecture of the encoder layers in **G** to represent the 3D mmWave shape into a 1D feature vector. But instead of the decoder layers of **G**, **D** uses multiple 2D convolution layers that convert input 2D shapes to the same length 1D feature vector. Finally, the two 1D feature vectors from both 3D and 2D convolutions are cascaded and fed into 2 fully-connected dense layers that finally reach the single neuron output layer. The output layer is passed through a sigmoid activation function and outputs the probability that the given 2D shape is real. By **G** trying to minimize the expected value (Eq. (7)) and **D** trying to maximize it, the entire cGAN will converge when **D** consistently outputs close to 0.5 probability of recognizing inputs correctly, *i.e.*, real and generated shapes have an equal probability of being real. This ensures that **G** has learned enough to produce the correct 2D shapes. Table 2 summarizes the **D** network parameters.

Table 2. Discriminator Network Parameters. 3DC: 3D Convolution (with batch norm.); FC: Fully Connected; 2DC: 2D Convolution (with batch norm.); Act. Fcn: Activation Function; LRelu: LeakyRelu; Output layer uses sigmoid activation.

	3DC1	3DC2	3DC3	3DC4	3DC5	3DC6	FC1	2DC1	2DC2	2DC3	2DC4	2DC5	2DC6	2DC7	FC2	FC3	FC4	Output
Filter #	16	32	64	128	256	1024		4	8	16	32	64	128	256				
Filter Size	6x6x6	6x6x6	6x6x6	6x6x6	6x6x6	6x6x6		4x3	6x6	6x6	6x6	6x6	6x6	6x6				
Dilation	2x2x2	2x2x2	2x2x2	2x2x2	2x2x2	2x2x2		2x2	2x2	2x2	2x2	2x2	2x2	2x2				
Act. Fcn	LRelu	LRelu	LRelu	LRelu	LRelu	LRelu	Relu	LRelu	LRelu	LRelu	LRelu	LRelu	LRelu	LRelu	Relu	Relu	Relu	Sigmoid

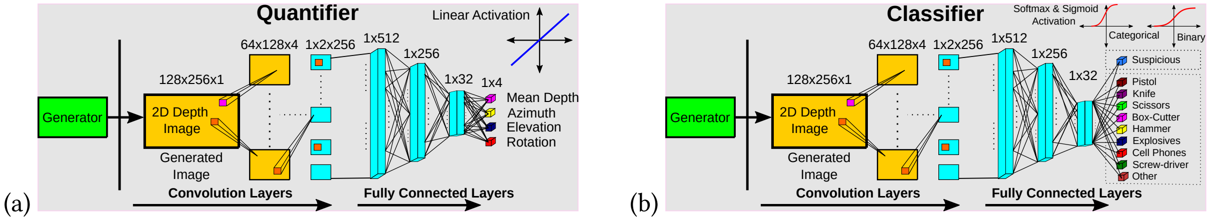

Figure 7. (a) Quantifier and (b) Classifier networks of the *SquiggleMilli* system.

Quantifier: Although our cGAN can recover most of the missing edges and parts of the objects, its output is only a 2D shape. Rather than predicting the entire 3D shape directly from the cGAN, which would be not only computationally expensive but also hard to learn due to inadequate input 3D data [50, 51], ***SquiggleMilli* leverages a *Quantifier* Q that can estimate the 3D features of the object: Mean depth and its orientations in the 3D plane**. Figure 7(a) shows our *Quantifier* network. Similar to **D**, **Q** leverages multiple 2D convolution layers to convert the 2D shape to a 1D feature vector. **Q** starts with the 2D shape as the input and applies 7 2D convolutional layers until it reaches the 1D fully-connected layer with 512 neurons. The network then passes through 2 fully-connected dense layers to reach the output layer with 4 output neurons corresponding to the 4 3D features: Mean depth (d); Azimuth (ϕ); Elevation (θ); and Rotation (α). These output neurons have linear activation functions to predict the actual value of these features. Table 3 summarizes the **Q** network parameters.

Table 3. Quantifier Network Parameters. 2DC: 2D Convolution (with batch norm.); FC: Fully Connected; Act. Fcn: Activation Function; LRelu: LeakyRelu; Output layer uses linear activation.

	2DC1	2DC2	2DC3	2DC4	2DC5	2DC6	2DC7	FC1	FC2	FC3	Output
Filters #	4	8	16	32	64	128	256				
Filter Size	4x3	6x6	6x6	6x6	6x6	6x6	6x6				
Dilation	2x2	2x2	2x2	2x2	2x2	2x2	2x2				
Act. Fcn	LRelu	LRelu	LRelu	LRelu	LRelu	LRelu	LRelu	Relu	Relu	Relu	Linear

Classifier: So far, *SquiggleMilli* recovers the full 2D shape and 3D features of an object from its 3D mmWave shape. We now elevate *SquiggleMilli*'s capability to detect and classify various real-life objects automatically. This is useful in non-intrusive applications, *e.g.*, automated packaged inventory counting, remote pat-down searching, *etc.* To this end, ***we propose a Classifier* C*, customized for a hand-held security application, that leverages the predicted 2D shape to label it to one of the object classes automatically***. Similar to **D** and **Q**, **C** leverages 7 2D convolution layers and 2 fully-connected dense layers to predict the classes. In our design, we select 8 types of items used by most security screening procedures (pistols, knives, scissors, hammers, boxcutters, cellphones, explosives, and screwdrivers [3]) as the categorical outputs. In addition, to these categories, we add one extra "Other" category to include various other items, *e.g.*, books, key-ring, wallet, key-chain, *etc.* Hence, the categorical output has 9 neurons in the output layer. Although **C** is currently not trained on a wider array of interesting items, we note that our network is scalable to more objects without requiring substantial changes in the layers or training with large samples. We leave more generalized object classifications as a future extension of *SquiggleMilli*. In addition to the fine-grained classification, we also incorporate a binary classification of objects being suspicious or not. Dangerous objects which should not be missed during classification are labeled as suspicious, *e.g.*, knives, pistols, explosives, *etc.* Such binary output could be very useful for hidden object annotations so that security personnel could perform additional checks. Finally, **C** uses the softmax and sigmoid activation functions for the categorical and binary output layers, respectively. Table 4 summarizes the **C** network parameters.

Network Loss Functions: All the network blocks rely on their loss functions to appropriately tune the convolution/deconvolution weights and train themselves. We use the L1-norm loss $\mathbf{L}_1(\mathbf{G})$ [52] as well as traditional GAN loss $\mathbf{L}(\mathbf{G})$ [44] to train the cGAN consisting of **G** and **D**. L1 loss helps the network in predicting a better 2D shape

Table 4. Classifier Network Parameters. 2DC: 2D Convolution (with batch norm.); FC: Fully Connected; Categorical class output layer uses softmax, and Binary output layer uses sigmoid activation functions.

	2DC1	2DC2	2DC3	2DC4	2DC5	2DC6	2DC7	FC1	FC2	FC3	Category Output	Binary Output
Filters #	4	8	16	32	64	128	256					
Filter Size	4x3	6x6	6x6	6x6	6x6	6x6	6x6					
Dilation	2x2	2x2	2x2	2x2	2x2	2x2	2x2					
Act. Fcn	LRelu	LRelu	LRelu	LRelu	LRelu	LRelu	LRelu	Relu	Relu	Relu	Softmax	Sigmoid

by estimating pixel-to-pixel mean absolute error, while traditional GAN loss maintains the adversarial game. Our combined cGAN loss is determined by:

$$\mathbf{L}_{cGAN} = \mathbf{L}(\mathbf{G}) + \lambda_I \cdot \mathbf{L}_1(\mathbf{G}) \quad where, \mathbf{L}_1(\mathbf{G}) = \mathbb{E}||x_I - \mathbf{G}(z_I)||_1 \tag{8}$$

where λ_I is the shape hyper-parameter. **Q** network leverages the cGAN loss $\mathbf{L}_{cGAN}$ and 3D features' loss between the ground-truth and the prediction to determine its loss function:

$$\mathbf{L}_Q = \mathbf{L}_{cGAN} + \lambda_F \cdot \mathbf{L}_F(\mathbf{G}) \quad where, \mathbf{L}_F(\mathbf{G}) = \mathbb{E}||x_F - \mathbf{G}(z_F)||_1 \tag{9}$$

where λ_F is the feature hyper-parameter. Finally, **C** network leverages $\mathbf{L}_{cGAN}$, categorical loss $\mathbf{L}_C$, and binary loss $\mathbf{L}_B$. The categorical and binary losses are computed as the cross-entropy losses between actual probabilities and predicted probabilities of different categories and binary classes [53], and are calculated as:

$$\mathbf{L}_{class}(\mathbf{G}) = \mathbf{L}_{cGAN} + \lambda_C \cdot \mathbf{L}_C(\mathbf{G}) + \lambda_B \cdot \mathbf{L}_B(\mathbf{G}) \tag{10}$$

$$where, \mathbf{L}_C(\mathbf{G}) = -\sum_{i=1}^{9} t_i \, log(c(s_i)), \quad and, \mathbf{L}_B(\mathbf{G}) = -(t_0 \, log(p_0) + (1 - t_0) log(1 - p_0)) \tag{11}$$

where $c(s_i)$ and t_i are the predicted and actual probabilities of i^{th} class (categorical output), and p_0 and t_0 are the predicted and actual probabilities of suspicious object (binary output). The hyper-parameters (λ_I, λ_F, λ_C, λ_B) represent the networks' focus on shape reconstruction, features prediction, and classification. Our goal is to find the set of values for these parameters, which would minimize the individual losses. However, determining the exact values is tricky and difficult. But intuitively, the value for λ_I should be the largest, since it is responsible for accurate reconstruction of human perceivable 2D shapes. We will discuss the hyper-parameters tuning in more detail in Section 4. These networks with their optimized loss functions enable *SquiggleMilli* to fill up the missing edges and parts in 2D shapes, predict the 3D features, and classify the objects accurately.

4 IMPLEMENTATION AND EXPERIMENTAL SETUP

Hardware Platform: We implement and evaluate *SquiggleMilli* using real data collected from a 77−81 GHz mmWave device, TI IWR1443BOOST [26], and a Google Tango device, ASUS Zenfone AR [25] (Figure 8). The mmWave device is equipped with 4 receive antennas that can collect reflected signals independently. To collect the signals in real-time, we attach a Data Capture Module, TI DCA1000EVM [54] to IWR1443BOOST. The DCA1000EVM module can temporarily store up to 2 GB of reflected signals and transfer them in real-time over an Ethernet cable connected to a laptop. The mmWave device can operate on a 4 GHz of bandwidth; however, due to frequency to space interpolation (Section 2.1), the effective bandwidth is around 3.32 GHz. We use the following FMCW parameters: Start frequency, 77.33 GHz; baseband sampling rate, 5 Msps; frequency ramp slope, 70.3 MHz/μS; number of ADC samples, 256; sweep duration, 56.9 μS; pulse repetition rate, 1 kHz; and maximum receive antenna gain, 10.5 dBi. At any aperture location, all reflected signals are collected within 51.2 μS, so even a fast hand-held *squiggle* speed, such as 3 m/s [55], would appear quasi-stationary in the mmWave signal space. We implement *SquiggleMilli* in Matlab and Python environments running on a host PC, which uses the reflected signals and *squiggle* locations as inputs and generates 2D shapes, 3D features, and categories as outputs.

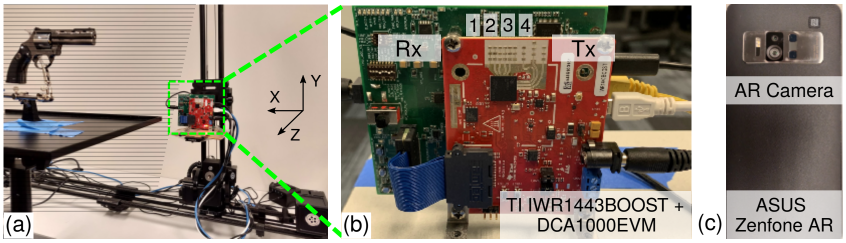

Figure 8. Ground-truth data collection setup: (a) 77–81 GHz device placed on a 2D linear controller that moves in horizontal and vertical axes; (b) mmWave device with 1 Tx and 4 Rx antennas; (c) ASUS Zenfone AR for *squiggle* pattern collection.

Real Data Collection: Since a real-time, tight synchronization between the mmWave and AR device is unavailable currently, we emulate a hand-held mmWave system in three steps: *First*, we collect the pose information of several *squiggle* motion patterns from volunteers using RTAB-Map [28]. *Then*, we place the mmWave device over a precise mechanical controller [56] that scans over a rectangular grid of area 20 × 20 cm^2 with a resolution of ~0.2 mm ($\sim \lambda/18$) (Figure 8[a]). *Finally*, for each *squiggle* pattern, we align its pose center to the grid center and find the closest grid locations for the *squiggle* locations: The real pose information from squiggle motion is used to filter out the dense measurement, so only those measurement points which trace out the approximate *squiggle* path are used as input to our system. Since human hand-held motion is continuous and varies in speed, the input data no longer is sampled uniformly and carries the effect of natural hand-held movement.

The object is placed in the aperture for Line-Of-Sight (LOS) data collection and is hidden inside the cloth for Non-Line-Of-Sight (NLOS) data collection. To emulate a practical hand-held scanning at non-uniform speed, we require very dense measurements: The fluidic arm motion by the user is continuous; thus, along the trajectory of the hand-held scanning, there will be an almost continuity of points. Thus, the resolution of the measurements from the mechanical controller setup is $\sim \lambda/18$, even though we need measurements at $\lambda/2$ resolution to avoid spatial aliasing. Besides, measurements at significantly higher resolution help in reducing discretization error in motion patterns when we emulate the hand-held movements. For a uniformly and densely spaced, perfect 2D grid based shape reconstruction, we resample the uniform grid at $\lambda/2$ resolution and apply traditional 3D SAR imaging (Section 2.1): This process generates the ground-truth mmWave shape. With the 20 × 20 cm^2 aperture area and 3.32 GHz bandwidth, the ground-truth shape theoretically achieves ~9.49 mm resolution at 50 cm depth in the horizontal and vertical directions and 4.51 cm resolution in the depth direction [14].

To collect the ground-truth shape and 3D features, we co-locate the AR device with mmWave and combine multiple snapshots from different aperture locations to find the mean depth and 3D orientation of the object. Then, we apply a background mask to trace the 2D ground-truth shape, and resize it to a 128×256 image. Our imaging framework (Section 3.2) generates a volume of size 40×1000×236 (~9.4 million voxels); so, to expedite our training/testing time, without loss in shape quality, we apply a 3D background mask, extract the object, and resize the volume to 32×64×96. We select 8 different categories of objects, each with 10 sub-categories, following TSA screening classes [3], and manually label each object with their ground-truth class. Besides, we use reflections from random objects, *e.g.*, books, key-ring, wallet, key-chain, *etc.*, and label them as the "Other" category. The measured 3D mmWave shape, and 2D ground-truth shape, 3D features, and class label are then concatenated to form a real data sample for *SquiggleMilli*'s learning model. Our real dataset consists of 2918 samples.

Synthetic Data Generation: Although real samples enable accurate learning, the data collection process is time-consuming. Besides, large-scale data samples for hand-held mmWave imaging are publicly unavailable. So, to overcome the data scarcity and to expedite our learning, we implement a data synthesizer, similar to [47], that uses the 2D shape and 3D features of an object and outputs mmWave 3D shape. We collect various 3D CAD models in different categories from the ShapeNet [57], and project their shapes from different viewing angles and distances. The projected shapes are then converted into the grayscale and resized into 128×256 to match with the real samples. We also record their mean depth and 3D orientation for ground-truth features. The 2D shapes are then converted to 3D voxels by applying the rotation matrices along the 3 axes [58]. Finally, the data synthesizer applies the standard ray tracing [59] to generate the 3D mmWave shape. Furthermore, the synthesizer considers accurate hardware parameters of our mmWave device, practical *squiggle* paths, and various blockage effects of mmWave signals [60] to faithfully generate 3D mmWave shapes, hidden or in LOS. Finally, the ray-traced 3D mmWave shape, and 2D ground-truth shape, 3D features, and class label form the synthetic data samples for training. Our synthetic dataset consists of 9800 samples.

Network Training: *SquiggleMilli* is mostly trained on synthetic samples and mainly tested on real samples. We train *SquiggleMilli* in two phases: (1) With 8000 synthesized samples for 1000 epochs; and (2) With only 218 real samples for another 1000 epochs. The rest of the real samples (2700) with LOS and NLOS objects are used for testing and benchmarking all our design components. We explore the effect of different combinations of hyper-parameters by training the networks multiple times and found that the networks performed much better when the ratio between λ_I and λ_F is close to 100×, *e.g.*, (λ_I, λ_F)=(1000, 10). So, the networks work well when they are more focused on shape reconstruction than feature prediction. Similarly, for the *Classifier* network, the best combination of hyper-parameters are $(\lambda_I, \lambda_C, \lambda_B)$ = (1000, 20, 50). All network architectures are implemented in Python with TensorFlow 2.1 [61] using PyCharm IDE [62] and Anaconda distribution [63] in a PC with Intel Xeon CPU @ 3.5 GHz, 32 GB RAM, and Nvidia's GeForce GTX 1070 [64]. Our networks take ~36 hours to complete the training, but in the future, training time could be improved with more powerful GPUs [64] or Cloud TPUs [65].

5 PERFORMANCE EVALUATION

We evaluate *SquiggleMilli* using 4 metrics commonly adopted to compare 2D shapes, 3D features, and classification results and contrast them with 2D grid based traditional SAR with no motion errors or missing grid samples.

► **Structural Similarity Index Measure (SSIM)**: An objective measure of distortion of structural information in a 2D reconstructed shape with reference to a 2D ground-truth shape [66]. The scale goes from 0 to 1, where 1 means a perfect pixel-to-pixel match.

► **Mean Depth Error**: The estimation error of an object's centroid in a volume measured from the *squiggle* motion plane in comparison to the ground-truth.

► **Orientation Error**: The estimation error of an object's 3D orientation, *i.e.*, azimuth, elevation, and rotation, in the volume reconstructed from the *squiggle* motion plane in comparison to the ground-truth.

► **Classification Confusion Matrix**: Probability of correctly classifying categorical and binary class labels, with each row of the matrix representing predicted probabilities, and each column representing actual probabilities.

Evaluation Summary: (1) *SquiggleMilli* improves the median SSIM by 0.41 from the traditional SAR with ± 5 mm standard deviation of motion error. Compensation gain under practical *squiggle* motion is limited, but the CS technique can improve the SSIM gain by a factor of 3.9× under moderate scan density. Besides, CS technique reduces the scan time by almost 30× in the median. (2) *SquiggleMilli*'s machine learning model further pushes the average SSIM from 0.44 to 0.9 and consistently outputs high-quality 2D shapes. It can accurately predict the mean depth with less than 1% error in 90^{th} percentile and 3D orientation angles with less than 7.6° error in azimuth and elevation and less than 1.22° error in rotation. Besides, it outputs categorical and binary class labels with an average 90% and 96% accuracy, respectively. (3) Finally, under field trials with samples collected in the wild, *SquiggleMilli* has similar SSIM and depth, angles, and class prediction accuracies, indicating that the system is generalizable under real conditions with different background noises and environmental movements.

5.1 Squiggle Correction and Objects Extraction

Motion Error Compensation: To evaluate the effectiveness of motion error compensation, we use the densely measured 2D grid at ∼ $\lambda/18$ resolution for the test samples with a single object. The depth of the object varies from 20 cm to 1 m from the aperture plane. To emulate the motion error, we apply a conservative ±5 mm of standard deviation on the ∼ $\lambda/2$ grid resolution and extract the reflected signals from the ∼ $\lambda/18$ resolution grid. We then emulate the motion error 20 times on each test sample, and apply motion compensation. To study the effect of motion error only, we used a very high scan density of average ∼100 points/cm^2. For the ground-truth, we use the ∼ $\lambda/2$ resolution grid and apply traditional SAR reconstruction (Section 2.1). Our experiments are conducted with the object mounted in parallel to the aperture plane to ensure the highest ground-truth quality. Then, we extract the 2D slice corresponding to the highest energy from each volume reconstructed by the ground-truth, with motion error, and with error compensation. Finally, we measure the SSIM between ground-truth and motion error induced shape and between ground-truth and *SquiggleMilli*.

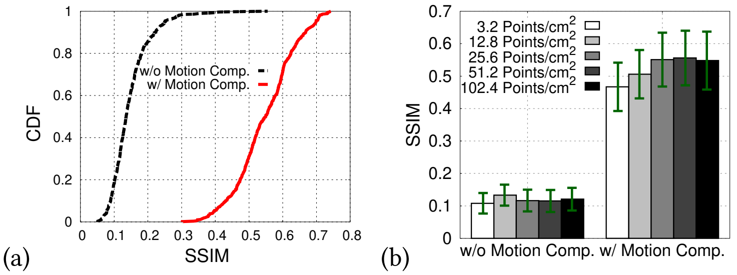

Figure 9. (a) SSIM distribution with and without motion error compensation. (b) Effect of varying scan density. Bars and errorbars represent the median and standard deviation across 64 test cases, each with 20 motion error paths.

Figure 9(a) shows the error compensation performance with CDF. The median and 90th percentile SSIM without error compensation are only 0.14 and 0.22, respectively. In contrast, *SquiggleMilli* significantly improves the structural quality, and the median and 90th percentile SSIM are 0.55 and 0.67, respectively. Figure 4(c) (Section 3.2) shows a visual result of the motion error compensation. In practice, however, the scan density varies with the devices' sampling rate, user hand movement speed, *etc.* To systematically test the effect of scan density, we resampled the motion error paths and reduced the samples by a factor of 1/2, from an average 102.4 points/cm^2 to an average 3.2 points/cm^2. Figure 9(b) shows the resultant SSIM and compares the cases with and without motion compensation. We have two observations: *First*, higher scan density does not improve the structural quality without motion compensation; the median SSIM increases from 0.11 under 3.2 points/cm^2 to 0.15 under 100.24 points/cm^2. This indicates the motion compensation is needed even if the user *squiggles* multiple times. *Second*, higher scan density does improve the quality with motion compensation: Median SSIM improves from 0.47 under 3.2 points/cm^2 to 0.55 under 100.24 points/cm^2. However, the overall improvement is 17% only. This indicates that motion compensation helps, but improvement is limited.

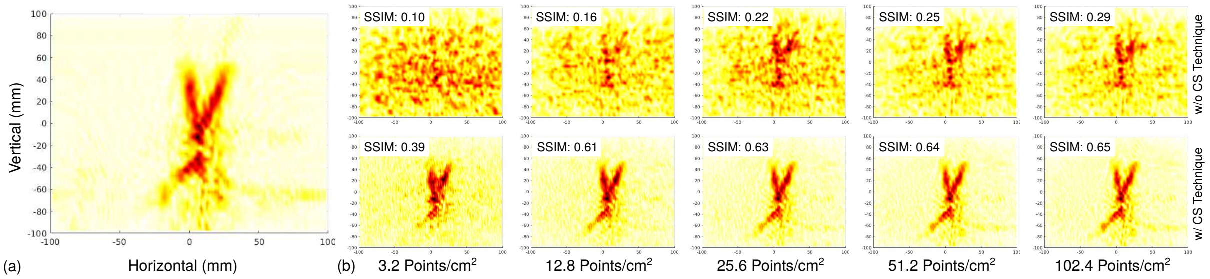

Figure 10. (a) Ground-truth shape with a perfect 2D grid. (b) Shapes with *squiggle* motion with and without CS technique.

Missing Grid Locations Recovery: Improvement from motion compensation is further limited when practical *squiggle* motion is applied. Figure 11(a) shows that under *squiggle* motion, the SSIM from motion compensation is hardly 0.29, even with 102.4 points/cm^2 scan density ("w/o CS" line). This is because the error deviation under practical *squiggle* motion is higher than the conservative ±5 mm standard deviation we have used before. We now evaluate the effectiveness of the CS technique (Section 3.2) on top of the motion compensation to improve the structural quality. We follow the process in Section 4 to measure reflected signals from 50 true hand-held *squiggle* motions, and apply motion compensation and CS technique. To evaluate the improvement from CS, we compare it against motion compensation only reconstruction. For each case, we vary the scan density as before, and reconstruct the volume with and without CS technique and find the SSIM.

Figures 10(a–b) show the ground-truth reconstruction with perfect ($\sim \lambda/2$) 2D grid resolution, and contrast the results with and without CS technique under different scan densities. Even though the CS reconstruction could not completely match the ground-truth shape, it's structural quality improves with increasing scan density. However, the visual quality of the CS shapes in Figure 10(b) do not improve significantly even if the scan density increases exponentially, there is hardly 0.04 SSIM improvements between 12.8 points/cm^2 and 102.4 points/cm^2. Still, the CS technique, together with the motion compensation, shows significant improvement over motion compensation only: SSIM improves between 0.30 to 0.45, a significant structural quality improvement.

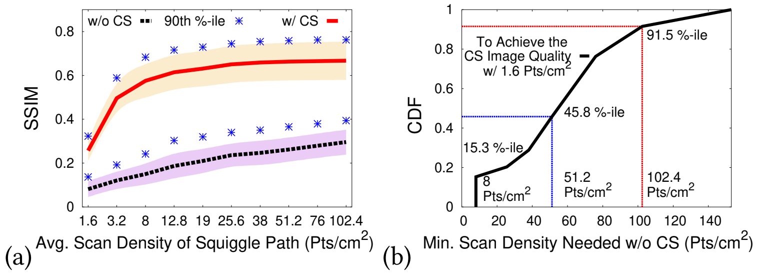

Figure 11. (a) Quality improvement with increasing the scan density on without and with the CS technique. (b) Minimum number of scan density needed in reconstruction without CS to achieve the minimum quality in the CS technique.

Figure 11(a) further shows the SSIM under varying scan density: Each line plots the median SSIM with and without the CS, and error bands and blue stars represent the standard deviations and 90th percentile, respectively. Similar to Figure 10, CS technique improves with increasing scan density, but the improvement plateaus off beyond 25.6 points/cm^2. Figure 11(b) further shows that without CS, it would require the user to *squiggle* multiple times needing a median scan density of 51.2 points/cm^2 to achieve the same shape quality as in CS technique with just 1.6 points/cm^2. This represents an almost 30× reduction in the total scan time with the CS technique.

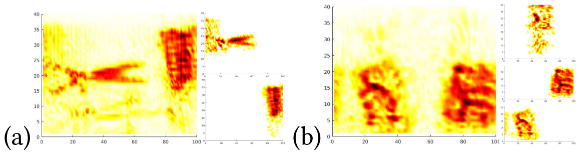

Figure 12. Segmentation example with (a) two and (b) three objects.

Table 5. Segmentation confusion matrix.

Actual/Predicted	1	2	3
1	**97.06**	2.94	0
2	0	**100**	0
3	0	0	**100**

Voxel Segmentation: *SquiggleMilli*'s learning model is trained on one object at a time; however, a practical scene may consist of multiple objects at various depths from the *squiggle* plane. We now evaluate *SquiggleMilli*'s ability to extract single objects from multi-object scenes. We vary the number of objects, from 1 to 3, in front of our data collection setup by placing them at different depths and spatial locations. *SquiggleMilli* then uses motion compensation, compressed sensing, and multi-focusing and voxel segmentation (Section 3.2). For each case, we count the number of objects predicted by *SquiggleMilli* and compare it with the ground-truth. Figure 12 shows two examples of 2D projected scenes for 2 and 3 objects, and their corresponding voxel segmented

2D projections. Table 5 shows the voxel segmentation results across the test samples in the form of a confusion matrix. Each row is the predicted probability of the number of objects in the scenes. For example, we see that for all one object cases, *SquiggleMilli* can accurately predict the correct number more than 97% of times. Furthermore, *SquiggleMilli* predicts the correct numbers accurately across all test samples with more than 1 objects.

5.2 Full Shape Recovery and Automatic Classification

Shape Improvement from cGAN: Motion compensation and CS technique improve the SSIM and output shapes with a structural quality close to the ground-truth. However, due to the specularity and weak reflectivity, the ground-truth mmWave shape itself could be missing many parts and edges. The resultant shape may not only fail in automatic classification but also be human imperceptible. Figure 13 shows some of the example ground-truth shapes generated by the perfect 2D grid based reconstruction. Even if *SquiggleMilli*'s imaging framework could match such shapes, they are clearly not perceivable by humans.

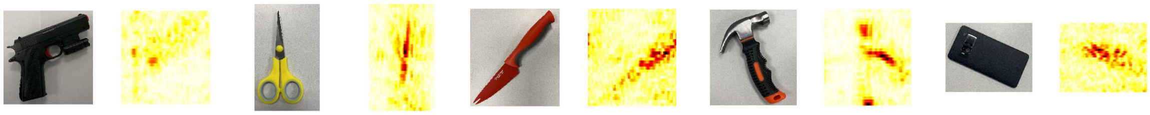

Figure 13. Examples where the ground-truth shapes are imperceptible by humans.

We now evaluate *SquiggleMilli*'s cGAN architecture in enhancing the shapes. Figures 14(a–b) show both the qualitative and quantitative results. *First*, Figure 14(a) shows three test objects' shape reconstruction in cGAN and contrast the result with traditional SAR with perfect 2D grid based measurements. Even if *SquiggleMilli* is never trained on these samples, it can accurately reconstruct the shapes with all the parts and edges and key discriminating features, such as barrel, butt, and trigger. *Second*, to evaluate the generalizability of *SquiggleMilli*, we run cGAN over 150 test samples, and calculate the SSIM by considering the 2D ground-truth shapes as the reference. Figure 14(b) shows the SSIM results with a scatter plot. Each point on the plot represents a test sample: X-value is the traditional SAR's SSIM (*e.g.*, column 3 in Figure 14[a]), and Y-value is the *SquiggleMilli*'s SSIM. While traditional SAR could only achieve an average SSIM of 0.44, *SquiggleMilli*'s has an average SSIM of more than 0.9 across the 150 test samples. We further test the shape reconstruction of NLOS objects which are of a similar category but have never been used in training (referred to as unseen samples) and find that the SSIM score is still as high as it is in the LOS cases and is able to reconstruct shapes of unseen objects with a median similarity score of 0.67 (Figure 14[b]). Figure 17 in the Appendix shows more shape reconstruction results.

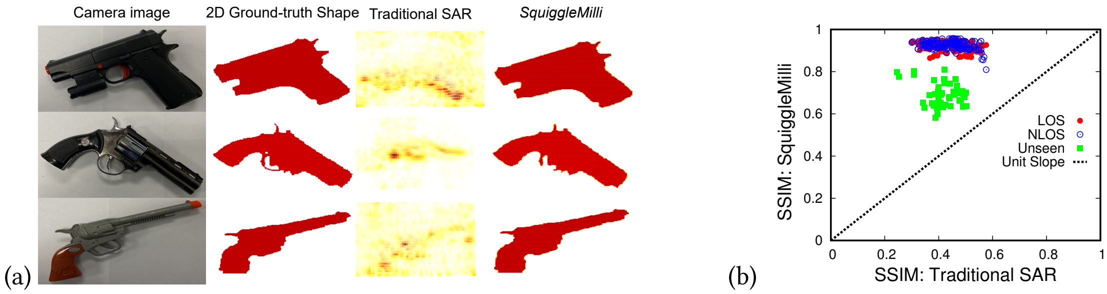

Figure 14. (a) Shapes reconstructed by *SquiggleMilli* from 3 test samples. (b) SSIM comparison between traditional SAR and *SquiggleMilli* across 150 test samples for LOS, NLOS, and Unseen samples.

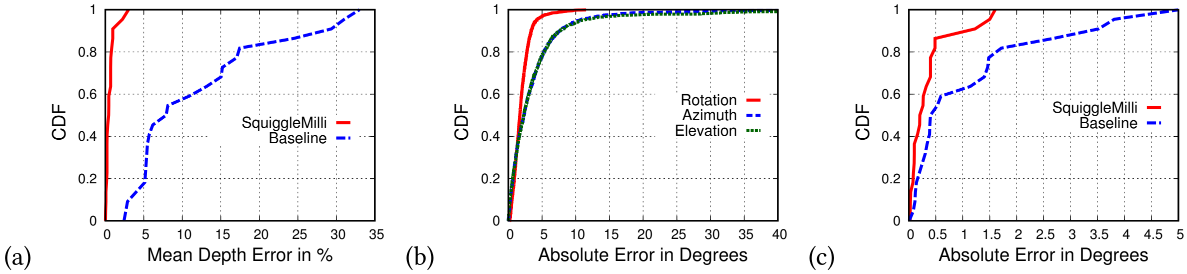

Figure 15. (a) Percentage error in mean depth prediction in real samples. (b) Absolute error in orientation prediction in synthetic samples. (c) Absolute error in rotation angle prediction in real samples.

3D Features Prediction: Recall that *Quantifier* **Q** leverages the generated 2D shape to predict the object's 3D features: Mean depth and 3D orientation. We use the previous 150 test samples, and estimate the error in predicting the features. We also compare the results with a baseline network that uses the shapes reconstructed by the traditional SAR only. To create the baseline, we use **Q**'s architecture but train the layers with traditional SAR generated shapes. This baseline network is also trained with identical sets of synthesized and real samples for the same number of epochs that were used in *SquiggleMilli* training.

Figure 15(a) shows the CDF of depth error for *SquiggleMilli* and baseline. We observe that under the baseline, the median depth error is about 8% and 90^{th} percentile could reach up to 29.35%. In contrast, under *SquiggleMilli*, the median depth error is about 0.43% and 90^{th} percentile is less than 1%. Such high depth estimation accuracy is attributed to the cGAN reconstructed accurate 2D shapes, where pixel values already embed the depth information and aid **Q** to learn it better. Figures 15(b–c) further evaluate **Q** in terms of 3D orientation prediction. Due to a constraint in mounting objects with different azimuth and elevation angles, we first evaluate the 3D orientation prediction with synthetic samples. Then, evaluate the rotation angle prediction with real samples. Figure 15(b) shows that in 90% of samples, both the predicted azimuth and elevation angles have less than 7.6° error. The rotation angle prediction shows the least error, less than 3.4° in 90^{th} percentile. We also verified the rotation angle prediction with real samples: Figure 15(c) shows that 90^{th} percentile error is less than 1.22° only. Both the shape improvement and 3D features prediction results indicate that **SquiggleMilli** ***generalizes its model well in real scenes with various object shapes and sizes, even if the model is trained mainly on synthesized data and only on limited real samples.***

Classifier: Recall that *Classifier* **C** can predict 9 object categories along with their binary classes. We randomly select 540 test samples (60 from each of the categories) and use the cGAN to produce the accurate 2D shapes. Then, we input these 2D shapes to **C** to predict their class labels. Since **C** is customized towards security application, we use 0.98 as the class probability threshold; so any object with less than 98% confidence is placed under the "Other" class. We also use the same set of samples for binary classification of labeling the objects as suspicious or not.

Table 6. Confusion matrix of categorical classifier in *SquiggleMilli* for LOS samples.

Actual/Predicted	Boxcutter	Cellphone	Explosive	Hammer	Knife	Pistol	Scissor	Screw	Other
Boxcutter	**90**	0	0	0	0	0	0	0	10
Cellphone	0	**100**	0	0	0	0	0	0	0
Explosive	0	0	**100**	0	0	0	0	0	0
Hammer	0	0	0	**100**	0	0	0	0	0
Knife	0	0	0	0	**92**	0	0	0	8
Pistol	0	0	0	0	0	**100**	0	0	0
Scissor	0	0	0	0	0	0	**100**	0	0
Screw	0	0	0	0	0	0	0	**70**	30
Other	0	13	25	0	0	5	0	0	**57**

Table 7. Confusion matrix of categorical classifier in *SquiggleMilli* for NLOS samples.

Actual/Predicted	Boxcutter	Cellphone	Explosive	Hammer	Knife	Pistol	Scissor	Screw	Other
Boxcutter	**94**	0	0	0	0	0	1	0	5
Cellphone	0	**69**	0	4	0	2	0	0	25
Explosive	0	0	**85**	8	0	0	0	0	7
Hammer	0	0	0	**93**	0	0	3	0	4
Knife	5	0	0	0	**67**	0	8	0	20
Pistol	2	0	0	0	0	**87**	2	1	8
Scissor	0	0	0	0	0	0	**100**	0	0
Screw	0	0	0	3	0	0	19	**45**	33
Other	0	0	19	6	0	21	2	0	**52**

Table 8. Binary class confusion matrix in *SquiggleMilli* for LOS samples.

Actual/Predicted	Suspicious	Non-suspicious
Suspicious	**98.25**	1.75
Non-suspicious	6	**94**

Table 9. Binary class confusion matrix in *SquiggleMilli* for NLOS samples.

Actual/Predicted	Suspicious	Non-suspicious
Suspicious	**90.75**	9.25
Non-suspicious	13.2	**86.8**

Table 6 shows the confusion matrix of categorical labeling with rows as the predicted probability for LOS samples. Cellphones, explosives, hammers, pistols, and scissors all show 100% accuracy; this is because, these objects reflect mmWave signals strongly, and cGAN can accurately reconstruct their shapes, aiding C to do a perfect classification. We also observe that 13% and 25% of "Other" categories are classified as cellphone and explosives because of their shape similarity (*e.g.*, wallet and key chains, *etc.*). To observe the performance of **C** for NLOS objects, we test the trained model with NLOS objects. Table 7 shows the confusion matrix of categorical labeling with rows as the predicted probability for NLOS samples. We find that the NLOS scissor shows 100% accuracy because of its shape peculiarity. Also, the other NLOS objects, such as boxcutters, explosives, hammers, pistols, still show a similar classification accuracy as in LOS cases. For the remaining NLOS cases, *e.g.*, cellphones, knives, screws, even if the classification accuracy is lower than the LOS cases, the objects are mostly labeled as the "Other" categories, which may require human intervention and further inspection. Overall, **C** has an average prediction accuracy of ~87.9%. Instead of 98% confidence, we could use the highest output probability to predict the labels. We still find that the average prediction accuracy is ~83.4% with LOS and NLOS objects, indicating that our model does not fit data to any one of the particular categories excessively. Tables 8 and 9 show the binary classification for LOS and NLOS objects, respectively, which is more accurate than categorical classification. This is expected since there are only two class labels. Still, on average, between LOS and NLOS objects, we get 9.6% false positives (non-suspicious items classified as suspicious); this is mostly due to the wrong classifications of "Other" categories. The average false negatives in our test samples are low, 5.5% only, which makes *SquiggleMilli* promising for security applications. To further study the performance of samples that have never been used in training, we use 32 different unseen samples of knives and guns with various orientation angles and depths. After shape reconstruction and classification, we find that majority of knives and guns are classified to the right class with low probability to different classes. The result shows that *SquiggleMilli* is able to achieve an accuracy of ~75% in the multi-class classification for unseen object categories. Similarly, for binary classification, the system is able to achieve ~85% accuracy.

5.3 Field Trial Results

We now evaluate *SquiggleMilli* with a larger number of test samples collected in the wild with various background scenes and people walking around the data collection setup. We collect reflection data from 60 objects, hidden and in LOS, and apply 20 different *squiggle* motions to it; this generates 2400 test samples (1200 in LOS and

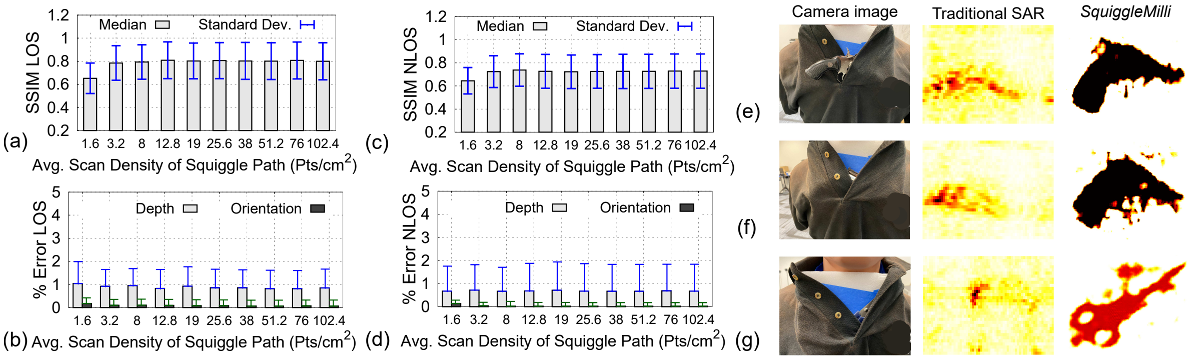

Figure 16. Field trial results: (a) SSIM score for LOS objects; (b) Percentage error in depth and orientation prediction for LOS objects. (c) SSIM score for NLOS objects; (d) Percentage error in depth and orientation prediction for NLOS objects. Shape reconstruction examples for different cases; (e) Partially occluded gun; (f) Fully occluded gun; (g) Fully occluded scissor.

1200 in NLOS). For each sample, we record the 2D ground-truth shapes and 3D features, and we resample the *squiggle* motion paths to emulate different hand-held speeds. Figure 16(a) shows the SSIM of 2D shapes of LOS objects: Even with a very low average scan density of 1.6 points/cm^2, the similarity index in the median is close to 0.65 and increases quickly to 0.8 with 8 points/cm^2. Also, for NLOS objects, *SquiggleMilli* achieves a median similarity score of 0.72 with 8 points/cm^2 (Figure 16[c]). Figure 16(b) also shows that the percentage errors of both the depth and orientation angle estimation are very low, less than 1% and 0.2% in the median, respectively. This holds true for NLOS objects, where we have a similar median error and slightly higher standard deviation (Figure 16[d]). Finally, Figures 16(e–g) show three example visual results when the items are mounted on a human dummy. While the traditional SAR fails to generate any interpretable results, either in partially or fully occluded scenes, *SquiggleMilli* can clearly show sharp images with discriminating features, even if it has never learned the scene before. ***These results demonstrate that* SquiggleMilli *is well generalizable under real conditions with different background noise and movements in the environment*.**

6 RELATED WORK

Radio Imaging: Conventional mmWave radio imaging systems achieve high resolution using mechanical motion controllers, bulky arrays, or rigid bodies [14–19, 67, 68]. They work at short-range and scan the target with a pre-determined trajectory from multiple viewpoints to reduce the effect of specular reflectivity. Similarly, MobiTagbot [69] utilizes the motion-enabled robot carrying microwave radios around and helping to locate the RFID tagged objects. The system can help to automatically reorder or relocate different objects in libraries, automation facilities, offices, *etc.* By carefully analyzing the relation between the channel and phase, it can achieve the ordering accuracy of up to 100% for the objects with 3-6 cm spacings. Similar to *SquiggleMilli*, MobiTagbot also relies on a mobile device to emulate the SAR principle. However, in contrast to *SquiggleMilli*, MobiTagbot relies on microwave band RFID tags attached to the object. Besides, it can only localize the objects but cannot determine their shape. All these systems rely on the principle of traditional SAR imaging techniques [16, 70–72]. Past works attempted to create portable radar-based imaging systems [73–75], but they still rely on bulky mechanical support that needs to be carried around for precise mechanical movement. Hence, these systems would be too cumbersome for a hand-held setting. Recent works in [76–78] also propose to use mmWave radars to detect object curvatures, boundaries, and 3D point clouds. However, the devices typically have to travel 10s of meters to reconstruct an object's or environment's shape. Hence, they are infeasible for short-range imaging applications. [27] proposes a hand-held mmWave imaging system, but it does not address the challenges with specularity and weak reflectivity artifacts in practical, hand-held settings. [68] aims to incorporate 3D imaging to mmWave networking devices without interfering with the networking functionality, but the reconstruction results lack shape and structural information, and all objects appear as blobs. Besides, their system is designed for 5G picocells and thus would not be applicable under hand-held settings. Tomographic imaging [79] technique also relies on reflected signals from objects; but instead of mechanically movable device, they require many radios around the target scene. Apart from mmWave, other imaging systems based on infrared and thermal cameras can be used in hand-held settings, but they are typically unsuitable for through-obstruction imaging [80–83]. Lidars can produce an accurate point cloud of an environment, but they do not work under obstructions, such as clothing [84, 85].

Resolution Improvement by Learning: Prior works have used neural networks to improve optical images' resolution [86–93]; in particular, deep learning framework has achieved the most significant improvements for camera images [23]. The networks learn the association between low resolution and high resolution images using RGB color and structural information. However, mmWave shapes have a very poor resolution compared to optical images and significantly lack high spatial frequency information. Besides, due to the specularity and weak reflectivity, many parts of an object do not appear in the mmWave reconstructed shapes. Thus, the traditional super-resolution techniques could not be applied to the mmWave domain. [47] recently proposed to use cGAN to generate high resolution depth images from low-resolution mmWave shapes. But the dataset and training domain is limited to vehicles; besides, the system is trained on reflected signals collected from a perfect 2D grid based SAR imaging system. In contrast, *SquiggleMilli* is designed and trained to recognize general-purpose shapes in security applications, using mmWave reflected signals from fluidic hand motion.

Recently, a few approaches incorporated deep-learning into radio signal based imaging directly [94–98]. But they focus on low-frequency, long-range, airborne SAR images reconstructed using 100s of meters length aperture, created by rigid bodies, *e.g.*, drones and airplanes. Besides, they use the measured reflected signals as both the input and ground-truth in training; so, their learning systems would be fundamentally limited by the specularity and weak reflectivity issues. [99] uses a static mmWave device and deep learning techniques to enhance the 3D representation of a scene. But their output is still limited to blob shapes; so they are imperceptible by humans. In contrast, *SquiggleMilli* is designed for hand-held settings, solves the challenges of fundamental specularity and weak reflectivity in mmWave signals, and is able to reconstruct shapes that are human perceivable.

7 CONCLUSION

In this work, we demonstrate that *SquiggleMilli* can be a promising solution to bring high resolution, through-obstruction imaging to cheap, ubiquitous mobile mmWave devices. The system approximates traditional SAR with a hand-held *squiggle* motion and improves the human perceptibility of the shape through a combination of signal processing and machine learning. We have customized *SquiggleMilli* for hand-held security applications, but the network is adaptable to different domains by training with limited samples. We believe, bringing *SquiggleMilli* to next-generation mobile devices may also inspire new perception algorithms and applications in the future.

ACKNOWLEDGMENTS

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

REFERENCES

[1] "ProVision Automatic Target Detection," 2015. [Online]. Available: http://www.sds.l-3com.com/advancedimaging/provision-at.htm

[2] Transportation Security Administration Press Office., "TSA Takes Next Steps to Further Enhance Passenger Privacy," July, 2011. [Online]. Available: https://www.tsa.gov/news/releases/2011/07/20/tsa-takes-next-steps-further-enhance-passenger-privacy

[3] Transportation Security Administration, "What Can I Bring?" 2017. [Online]. Available: https://www.tsa.gov/travel/security-screening/whatcanibring

[4] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3D Tracking via Body Radio Reflections," in *Proc. of USENIX NSDI*, 2014.

[5] S. Nannuru, Y. Li, Y. Zeng, M. Coates, and B. Yang, "Radio-Frequency Tomography for Passive Indoor Multitarget Tracking," in *IEEE Transactions on Mobile Computing*, 2013.

[6] J. Xiong and K. Jamieson, "ArrayTrack: A Fine-Grained Indoor Location System," in *USENIX NSDI*, 2013.

[7] Thales Visionix Inc., "IS-900," 2014. [Online]. Available: http://www.intersense.com/pages/20/14

[8] Redecomposition, "See Through Wall Radar Imaging Technology," 2015. [Online]. Available: http://redecomposition.wordpress.com/technology/

[9] F. Adib and D. Katabi, "See Through Walls with WiFi!" in *Proc. of ACM SIGCOMM*, 2013.

[10] F. Adib, C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand, "Capturing the Human Figure Through a Wall," in *Proc. of ACM SIGGRAPH Asia.* Los Angeles, California, USA: Association for Computing Machinery, 2015.

[11] Y. Tian, G. Lee, H. He, C. Hsu, and D. Katabi, "RF-Based Fall Monitoring Using Convolutional Neural Networks," in *ACM IMWUT*, 2018.

[12] C.-Y. Hsu, Y. Liu, Z. Kabelac, R. Hristov, D. Katabi, and C. Liu, "Extracting Gait Velocity and Stride Length from Surrounding Radio Signals," in *ACM CHI*, 2017.

[13] H. Kajbaf, R. Zheng, and R. Zoughi, "Improving Efficiency of Microwave Wideband Imaging using Compressed Sensing Techniques," *Materials Evaluation*, vol. 70, pp. 1420–1432, 2012.

[14] D. M. Sheen, D. L. McMakin, and T. E. Hall, "Three-Dimensional Millimeter-Wave Imaging for Concealed Weapon Detection," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, no. 9, 2001.

[15] M. E. Yanik and M. Torlak, "Near-Field MIMO-SAR Millimeter-Wave Imaging With Sparsely Sampled Aperture Data," *IEEE Access*, vol. 7, pp. 31 801–31 819, 2019.

[16] M. Soumekh, *Synthetic Aperture Radar Signal Processing.* John Wiley & Sons, Inc., 1999.

[17] ——, "A System Model and Inversion for Synthetic Aperture Radar Imaging," *IEEE Transactions on Image Processing*, vol. 1, no. 1, 1992.

[18] B. Mamandipoor, G. Malysa, A. Arbabian, 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.

[19] Y. Zhu, Y. Zhu, B. Y. Zhao, and H. Zheng, "Reusing 60GHz Radios for Mobile Radar Imaging," in *ACM MobiCom*, 2015.

[20] J. Nanzer, *Microwave and Millimeter-Wave Remote Sensing for Security Applications.* Artech House, 2013.

[21] H. Xu, V. Kukshya, and T. S. Rappaport, "Spatial and Temporal Characteristics of 60-GHz Indoor Channels," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 3, 2002.

[22] P. F. M. Smulders, "Statistical Characterization of 60-GHz Indoor Radio Channels," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 10, 2009.

[23] C. Ledig and L. Theis and F. Huszar and J. Caballero and A. Cunningham and A. Acosta and A. Aitken and A. Tejani and J. Totz and Z. Wang and W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *IEEE/CVF CVPR*, 2017.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *IEEE/CVF CVPR*, 2017.

[25] AsusTek Computer Inc., "Zenfone AR: Go Beyond Reality," 2021. [Online]. Available: https://www.asus.com/us/Phone/ZenFone-AR-ZS571KL/

[26] Texas Instruments, "IWR1443 Single-Chip 76-GHz to 81-GHz MmWave Sensor Evaluation Module," 2020. [Online]. Available: https://www.ti.com/tool/IWR1443BOOST

[27] M. S. Saadat, S. Sur, S. Nelakuditi, and P. Ramanathan, "MilliCam: Hand-held Millimeter-Wave Imaging," in *IEEE International Conference on Computer Communications and Network (ICCCN)*, 2020.

[28] IntRoLab, "Real-Time Appearance-Based Mapping," 2021. [Online]. Available: http://introlab.github.io/rtabmap/

[29] L. Anitori, M. Otten, and P. Hoogeboom, "Compressive Sensing for High Resolution Radar Imaging," in *2010 Asia-Pacific Microwave Conference*, 2010, pp. 1809–1812.

[30] M. A. Herman and T. Strohmer, "High-Resolution Radar via Compressed Sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2275–2284, 2009.

[31] S. Li, Y. Zhu, Y. Xie, and S. Gao, "Dynamic Magnetic Resonance Imaging Method Based on Golden-Ratio Cartesian Sampling and Compressed Sensing," *PLOS ONE*, vol. 13, p. e0191569, 01 2018.

[32] S. K. Gunasheela and H. S. Prasantha, "Compressed Sensing for Image Compression: Survey of Algorithms," in *Emerging Research in Computing, Information, Communication and Applications*, 2019.

[33] I. Stankovic, I. Orovic, and S. Stankovic, "Image Reconstruction from a Reduced Set of Pixels using a Simplified Gradient Algorithm," in *IEEE Telecommunications Forum Telfor (TELFOR)*, 2014.

[34] H. Hosseini, N. Marvasti, and F. Marvasti, "Image Inpainting Using Sparsity of the Transform Domain," *CoRR*, vol. abs/1011.5458, 11 2010.

[35] M. Marim, E. Angelini, and J.-C. Olivo-Marin, "A Compressed Sensing Approach for Biological Microscopy Image Denoising," *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 04 2009.

[36] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, "Light Field Reconstruction Using Sparsity in the Continuous Fourier Domain," *ACM Transactions on Graphics*, vol. 34, no. 1, 2015.

[37] D. Yang, G. Wu, J. Li, C. Chang, B. Luo, H. Lin, S. Sun, Y. Xu, and L. Yin, "Image Recovery of Ghost Imaging with Sparse Spatial Frequencies," *Optics Letters*, vol. 45, no. 19, 2020.
[38] E. J. Candes and M. B. Wakin, "An Introduction To Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, 2008.
[39] J. Yang and Y. Zhang, "Alternating Direction Algorithms for L1 Problems in Compressive Sensing," *SIAM Journal on Scientific Computing*, vol. 33, 2011.
[40] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.
[41] C. Zhou and Stephen Lin and S. Nayar, "Coded Aperture Pairs for Depth from Defocus," in *IEEE 12th International Conference on Computer Vision*, 2009.
[42] S. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
[43] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," 2014. [Online]. Available: https://arxiv.org/abs/1411.1784
[44] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," in *ACM International Conference on Neural Information Processing Systems*, 2014.
[45] H. Ge, Y. Xia, X. Chen, R. Berry, and Y. Wu, "Fictitious GAN: Training GANs with Historical Models," in *European Conference on Computer Vision ECCV 2018*, 2018.
[46] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, 2017.
[47] J. Guan, S. Madani, S. Jog, S. Gupta, and H. Hassanieh, "Through Fog High-Resolution Imaging Using Millimeter Wave Radar," in *IEEE/CVF CVPR*, 2020.
[48] Z. You, J. Ye, K. Li, Z. Xu, and P. Wang, "Adversarial Noise Layer: Regularize Neural Network by Adding Noise," in *IEEE ICIP*, 2019.
[49] H. Ahn and C. Yim, "Convolutional Neural Networks Using Skip Connections with Layer Groups for Super-Resolution Image Reconstruction Based on Deep Learning," *Applied Sciences*, vol. 10, p. 1959, 03 2020.
[50] C.-L. Li, M. Zaheer, Y. Zhang, B. Poczos, and R. Salakhutdinov, "Point Cloud GAN," 2018. [Online]. Available: https://arxiv.org/abs/1810.05795
[51] E. Smith and D. Meger, "Improved Adversarial Systems for 3D Object Generation and Reconstruction," 2017. [Online]. Available: https://arxiv.org/abs/1707.09557
[52] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss Functions for Image Restoration With Neural Networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.
[53] Raul Gomez Blog, "Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss," 2021. [Online]. Available: https://gombru.github.io/2018/05/23/cross_entropy_loss/
[54] Texas Instruments, "DCA1000EVM: Real-time Data-Capture Adapter for Radar Sensing Evaluation Module," 2020. [Online]. Available: https://www.ti.com/tool/DCA1000EVM
[55] M. Elgendi, F. Picon, N. Magnenat-Thalmann, and D. Abbott, "Arm Movement Speed Assessment via a Kinect Camera: A Preliminary Study in Healthy Subjects," *BioMedical Engineering OnLine*, vol. 13, no. 88, 2014.
[56] Cinetics, "Lynx 3 Axis Slider." [Online]. Available: https://cinetics.com/lynx-3-axis-slider/
[57] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford-Princeton-Toyota Technological Institute at Chicago, Tech. Rep., 2015.
[58] Y.-B. Jia, "Rotation in the Space," 2020. [Online]. Available: http://web.cs.iastate.edu/~cs577/handouts/rotation.pdf
[59] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, M. Barbiroli, M. Zoli, L. Tian, X. Yin, D. A. Dupleich, R. Muller, C. Schneider, and R. S. Thoma, "Ray-Tracing-Based mm-Wave Beamforming Assessment," *IEEE Access*, vol. 2, pp. 1314–1325, 2014.
[60] New York University and NYU WIRELESS, "NYUSIM," 2020. [Online]. Available: https://wireless.engineering.nyu.edu/nyusim/
[61] Open-Source, "TensorFlow," 2021. [Online]. Available: https://www.tensorflow.org/
[62] JetBrains Open-Source, "PyCharm," 2021. [Online]. Available: https://www.jetbrains.com/pycharm/
[63] Open-Source, "ANACONDA," 2021. [Online]. Available: https://www.anaconda.com/
[64] NVIDIA, "GEFORCE," 2021. [Online]. Available: https://www.nvidia.com/en-us/geforce/
[65] Google, "Cloud TPU," 2021. [Online]. Available: https://cloud.google.com/tpu
[66] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, 2004.
[67] C. M. Watts, P. Lancaster, A. Pedross-Engel, J. R. Smith, and M. S. Reynolds, "2D and 3D Millimeter-Wave Synthetic Aperture Radar Imaging on a PR2 Platform," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
[68] J. Guan, A. Paidimarri, A. Valdes-Garcia, and B. Sadhu, "3D Imaging using mmWave 5G Signals," in *2020 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, 2020.
[69] L. Shangguan and K. Jamieson, "The Design and Implementation of a Mobile RFID Tag Sorting Robot," in *ACM MobiSys*, 2016.
[70] M. Cheney and B. Borden, *Fundamentals of Radar Imaging.* Society for Industrial and Applied Mathematics, 2009.
[71] H. Griffiths and C. Baker, *Fundamentals of Tomography and Radar.* Springer Netherlands, 2006.

[72] J. Nanzer, *Microwave and Millimeter-Wave Remote Sensing for Security Applications.* Artech House, 2012.

[73] A. Gromek, “High Resolution SAR Imaging Trials Using a Handheld Vector Network Analyzer,” in *International Radar Symposium (IRS)*, 2014.

[74] K. Browne, R. Burkholder, and J. Volakis, “A Novel Low-Profile Portable Radar System for High Resolution Through-Wall Radar Imaging,” in *IEEE Radar Conference*, 2010.

[75] A. O. Boryssenko, D. L. Sostanovsky, and E. S. Boryssenko, “Portable Imaging UWB Radar System With Two-Element Receiving Array,” in *Ultra-Wideband Short-Pulse Electromagnetics.* Springer, 2007.

[76] Y. Zhu, Y. Zhu, B. Y. Zhao, and H. Zheng, “Reusing 60GHz Radios for Mobile Radar Imaging,” in *Proc. of ACM MobiCom*, 2015.

[77] Y. Zhu, Y. Yao, B. Y. Zhao, and H. Zheng, “Object Recognition and Navigation using a Single Networking Device,” in *Proc. of ACM MobiSys*, 2017.

[78] K. Qian, Z. He, and X. Zhang, “3D Point Cloud Generation with Millimeter-Wave Radar,” in *ACM IMWUT*, 2020.

[79] B. Wei, A. Varshney, W. Hu, N. Patwari, and et al, “dRTI: Directional Radio Tomographic Imaging,” *CoRR*, vol. abs/1402.2744, 2014.

[80] H.-M. Chen, S. Lee, R. M. Rao, M. . Slamani, and P. K. Varshney, “Imaging for Concealed Weapon Detection: A Tutorial Overview of Development in Imaging Sensors and Processing,” *IEEE Signal Processing Magazine*, vol. 22, no. 2, 2005.

[81] D. Kim, S. Lee, and J. Paik, “Active Shape Model-Based Gait Recognition Using Infrared Images,” in *Springer FGIT-SIP*, 2009.

[82] Vision-Systems, “Infrared Imaging Systems Mounted on Drone Platforms Designed to Assist Demining Efforts in Afghanistan,” 2019. [Online]. Available: https://www.vision-systems.com/non-factory/scientific-industrial-research/article/16748212/infrared-imaging-systems-mounted-on-drone-platforms-designed-to-assist-demining-efforts-in-afghanistan

[83] P. W. Kruse, *Uncooled Thermal Imaging Arrays, Systems, and Applications.* SPIE Press, 2001.

[84] C. Glennie and D. D. Lichti, “Static Calibration and Analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning,” *MDPI Remote Sensing*, vol. 2, no. 6, 2010.

[85] G. Satat, M. Tancik, and R. Raskar, “Towards Photography Through Realistic Fog,” in *2018 IEEE International Conference on Computational Photography (ICCP).* Pittsburgh, PA, USA: IEEE, 2018.

[86] C. Dong, C. C. Loy, K. He, and X. Tang, “Image Super-Resolution Using Deep Convolutional Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, 2016.

[87] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, “Deep Learning for Single Image Super-Resolution: A Brief Review,” *IEEE Transactions on Multimedia*, vol. 21, no. 12, 2019.

[88] K. de Haan, Y. Rivenson, Y. Wu, and A. Ozcan, “Deep-Learning-Based Image Reconstruction and Enhancement in Optical Microscopy,” *Proceedings of the IEEE*, vol. 108, no. 1, 2020.

[89] S. Ravishankar, J. C. Ye, and J. A. Fessler, “Image Reconstruction: From Sparsity to Data-Adaptive Methods and Machine Learning,” *Proceedings of the IEEE*, vol. 108, no. 1, 2020.

[90] F. Knoll, K. Hammernik, C. Zhang, S. Moeller, T. Pock, and D. K. Sodickson, “Deep-Learning Methods for Parallel Magnetic Resonance Imaging Reconstruction: A Survey of the Current Approaches, Trends, and Issues,” *IEEE Signal Processing Magazine*, vol. 37, no. 1, 2020.

[91] M. Qin, S. Mavromatis, L. Hu, F. Zhang, R. Liu, J. Sequeira, and Z. Du, “Remote Sensing Single-Image Resolution Improvement Using A Deep Gradient-Aware Network with Image-Specific Enhancement,” *Remote Sensing*, vol. 12, no. 5, 2020.

[92] Y. Cotte, M. F. Toy, N. Pavillon, and C. Depeursinge, “Microscopy Image Resolution Improvement by Deconvolution of Complex Fields,” *Optics Express*, vol. 18, no. 19, 2010.

[93] A. Guei and M. Akhloufi, “Deep Learning Enhancement of Infrared Face Images Using Generative Adversarial Networks.” *Applied Optics*, vol. 57, no. 18, 2018.

[94] X. Tang, L. Zhang, and X. Ding, “SAR Image Despeckling with a Multilayer Perceptron Neural Network,” *International Journal of Digital Earth*, vol. 12, no. 3, 2019.

[95] L. H. Hughes, M. Schmitt, L. Mou, Y. Wang, and X. X. Zhu, “Identifying Corresponding Patches in SAR and Optical Images With a Pseudo-Siamese CNN,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, 2018.

[96] L. Mou, M. Schmitt, Y. Wang, and X. X. Zhu, “A CNN For the Identification of Corresponding Patches in SAR and Optical Imagery of Urban Scenes,” in *Joint Urban Remote Sensing Event (JURSE)*, 2017.

[97] G. Chierchia, D. Cozzolino, G. Poggi, and L. Verdoliva, “SAR Image Despeckling through Convolutional Neural Networks,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[98] P. Wang, H. Zhang, and V. M. Patel, “SAR Image Despeckling Using a Convolutional Neural Network,” *IEEE Signal Processing Letters*, vol. 24, no. 12, 2017.

[99] S. Fang and S. Nirjon, “AI-Enhanced 3D RF Representation Using Low-Cost MmWave Radar,” in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018.

A MULTIPLE SHAPE RECONSTRUCTION RESULTS

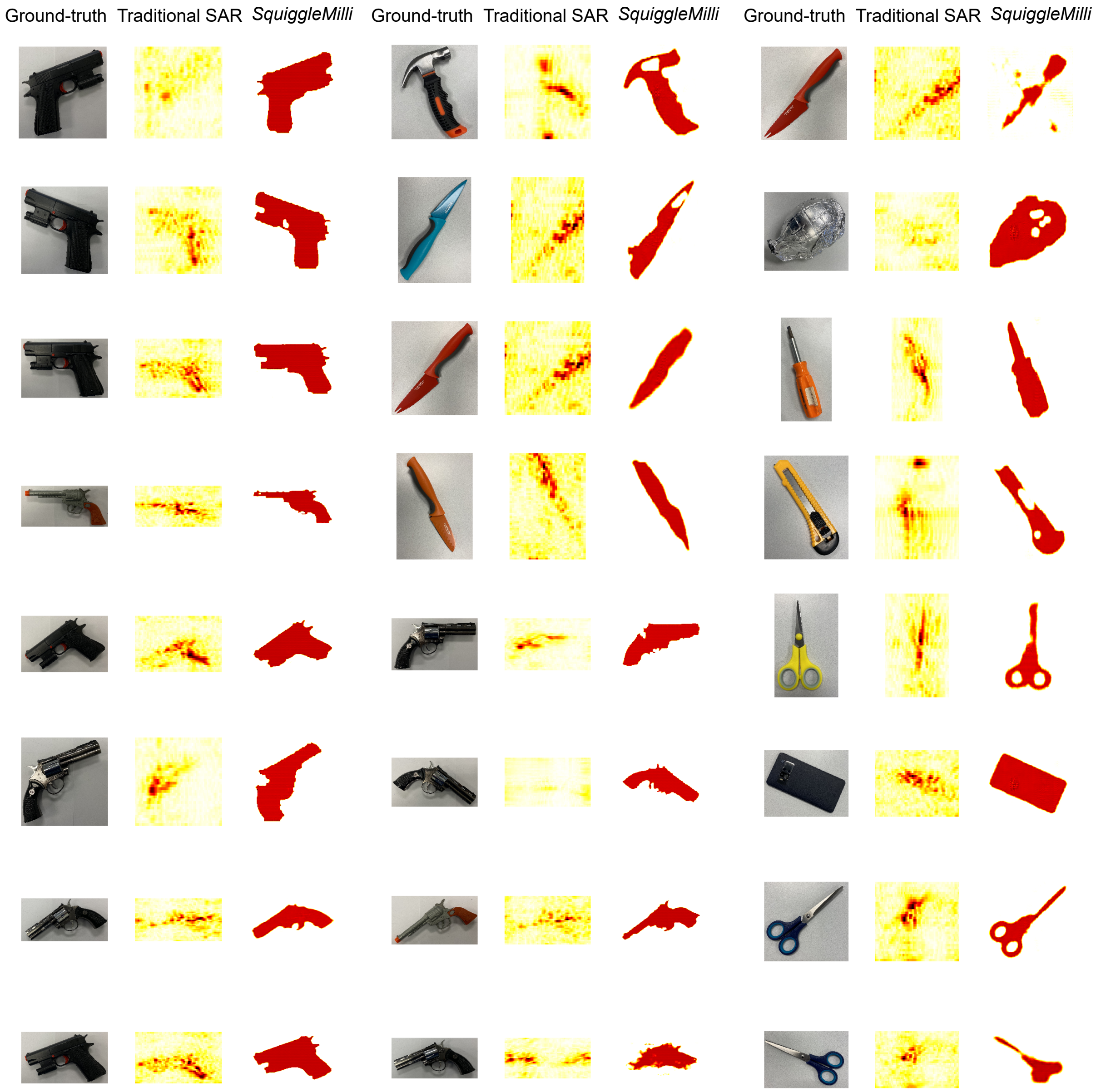

Figure 17. Multiple shape reconstructions from *SquiggleMilli*.