mmSight: Towards Robust Millimeter-Wave Imaging on Handheld Devices

Jacqueline M Schellberg; Hem Regmi; Sanjib Sur
Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA
schellbj@email.sc.edu; hregmi@email.sc.edu; sur@cse.sc.edu

***Abstract*—We propose *mmSight*, a system that enables Synthetic Aperture Radar (SAR) imaging on handheld millimeter-wave (mmWave) devices. SAR imaging requires precise device self-localization or bulky motion controllers to reconstruct an image, but standard handheld devices suffer from various pose errors that cannot be addressed with traditional motion compensation methods. *mmSight* uses the time delay of the reflected mmWave signals across separated antennas to limit the pose error and perform improved mobile mmWave imaging. Since the mmWave signals are fundamentally limited by specularity and weak reflectivity, even a perfect pose correction may not yield a perceptible image. To this end, *mmSight* employs a generative learning model to learn the relationship between the imperceptible 3D image and a discernable 2D image and automatically classifies objects into several categories. We show that *mmSight* improves the structural quality of the mmWave images from 0.01 to 0.92, and it can be leveraged to identify several common hidden objects.**

***Index Terms*—SAR imaging; Millimeter-wave; Pose Correction; Generative Adversarial Networks.**

I. Introduction

Millimeter-wave (mmWave) imaging systems, such as those commonly used for airport security, can image hidden objects beyond obstructions, like clothing and packaging. Due to the wavelength and propagation characteristics of the transmitted signal, mmWave systems can provide information about object material, orientation, size, and shape [1], without intruding on privacy like optical cameras. 5G smart devices equipped with mmWave transceiver arrays [2] could enable efficient package and inventory accounting, through-wall imaging for construction site surveying, disaster relief, and security checks.

Traditionally, mmWave imaging systems use the Synthetic Aperture Radar (SAR) technique to produce through-obstruction images. The SAR technique coherently combines signals transmitted and received at different spatial locations to produce a synthetic aperture for image reconstruction [3]. However, mmWave imaging for handheld 5G smart devices has been difficult for two reasons. *First*, traditional SAR imaging methods rely on carefully controlled movements or prohibitively costly external motion capture systems to yield a sufficiently focused image. A system with limited motion tracking precision cannot take advantage of the fine-grained properties of the mmWave signal, such as an object's specific shape, material composition, or orientation [4]. Most smartphone self-tracking algorithms can only localize at the centimeter-scale [5], which is below the millimeter-scale accuracy required for coherent mmWave imaging. Further, existing algorithms cannot correct the large inaccuracies generated by the device. Existing works have only been able to correct unambiguous pose errors within $\frac{\lambda}{2}$, which would not translate to real handheld devices with pose errors several times the system wavelength [6]. Approximating the handheld trajectory through interpolation [7] also fails due to the characteristic drift of the poses. *Second*, mmWave reflections might experience signal specularity [8] or loss in variable conditions, which can eliminate high-frequency features, like edges, in the resultant reconstructed image. So, even if the device-reported poses were precise, the aperture sampling may not contain sufficient views to illuminate the target.

This paper proposes *mmSight*, a system that addresses image defocusing on handheld devices by leveraging the mmWave signals and the antenna separation to correct the pose inaccuracies. *First*, *mmSight* extends on previous methods to use the known antenna spacing found on the mmWave device and the motion of the scene's background to deduce the motion of the device. Since these existing methods alone are not sufficient for 3D handheld motion, we address the drift error using a novel method for accurate focusing. *Then*, *mmSight* divides the pose trajectory into overlapping windows and replaces the original, inaccurate vision-based self-tracking poses with a new mmWave-based pose estimate. *mmSight* corrects the global drift error by comparing mmWave 3D images generated within the overlap region of two estimated pose windows. Since the two poses are locally accurate, but globally separated due to drift error, we apply a transformation so that each 3D image closely coincides. *Finally*, *mmSight* reconstructs the complete image with the newly corrected poses and uses a machine learning framework to recover features that could not be obtained through the pose-corrected reconstructed images. We design a conditional Generative Adversarial Network (cGAN) to learn the association between imperceptible 3D voxels and the ground-truth object shape, and a Classifier network to classify the object for common applications automatically. These networks can be trained for several applications beyond the objects shown in this paper.

We prototype *mmSight* using Commercial-Off-The-Shelf (COTS) mmWave devices. Our results show that *mmSight* reduces the median drift error from 5 mm to 1 mm, increases the median Structural Similarity Index Measure, which measures the similarity of an image with 0 being least similar and 1 being most similar, from 0.01 to 0.92, and achieves $\sim$98% object classification accuracy, even under occlusion.

Our main contributions are: (1) We design a millimeter-level accurate device self-pose estimation method for the handheld case by exploiting features of the target scene embedded within mmWave reflected signals; and (2) We design a deep generative network to learn the missing high-frequency features in the output mmWave images that are required for shape perception and object classification. Our results show that *mmSight* could also enable precise motion tracking for several applications, such as VR/AR, device navigation, *etc.*

II. Background and Challenges

A. Time Domain Back Projection for Image Reconstruction

SAR imaging uses the motion of an RF transceiver that passes in front of a scene while transmitting and receiving signals to produce a through-obstruction image. Common SAR imaging systems transmit Frequency Modulated Continuous Wave (FMCW) signals, which use a linear frequency sweep to produce higher range resolutions according to $c/2B$, where c is the speed of light and B is the bandwidth of the system [9]. So, a system with a 77-81 GHz transmitter with a 4 GHz bandwidth can achieve a range resolution of 3.75 cm. The Time Domain Back Projection (TDBP) image reconstruction algorithm is ideal for non-linear imaging geometries, such as handheld motions, since it applies focusing to each discrete location in the 3D image scene, known as a voxel, instead of arbitrary depths [10]. TDBP requires prior knowledge of the range-compressed mmWave signal and the corresponding transceiver location. Additionally, each reconstructed voxel's location coincides with the coordinate system used for the transceiver positions, so no transformation between the collected poses and the voxel grid is necessary. TDBP image reconstruction is defined as [11]:

$$C(p) = \sum_{m=1}^{M} S(m, \Delta R(m,p)) \cdot \exp\left(-j4\pi\Delta R(m,p) \cdot \frac{f_c}{c}\right); \quad \forall p \in P \quad (1)$$

where M is the total number of transceiver frames, P is the set of image voxels, $S(m, \Delta R(m,p))$ is the complex-valued interpolated, range-compressed signal at a distance $\Delta R(m,p)$ from frame m to voxel p. TDBP generates a complex-valued datacube, C, and the absolute value represents the voxel intensity of the target scene.

B. Challenges

Motion Error: TDBP requires sub-wavelength motion tracking precision for object focusing [12]. Without accurate position measurements, each signal would destructively add during the Back Projection sum (Eq. (1)). However, existing low-cost handheld devices are incapable of generating poses within the maximum allowable error required for mmWave imaging [13]. These devices are limited by the sensors used for self-tracking – cameras are sensitive to poor lighting, scene homogeneity, occlusions, *etc.* [14]. Figure 1 shows the effects of motion error by comparing a ground truth optical image of a CD to the mmWave image produced using the poses from a vision-based self-tracking device. Clearly, there are no perceptible features in the output mmWave image, such as the hole or overall shape of the CD.

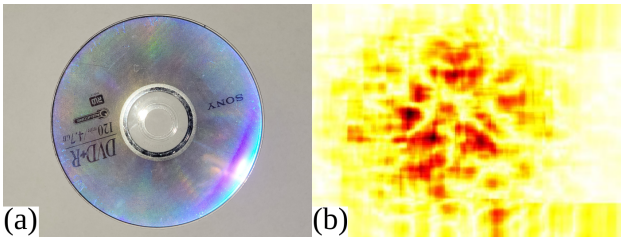

Fig. 1: (a) Ground truth optical image; (b) Defocused mmWave image using inaccurate device poses.

Although autofocus methods for SAR imaging can overcome phase errors caused by motion error, they are best suited for far-field cases with known imaging geometry and can only overcome small residual motion errors [15]. Other systems, such as [7], have succeeded in using interpolation to eliminate most motion errors caused by position inaccuracies, but the significant drift errors cause the errors accumulate and cannot be filtered out in vision-based systems.

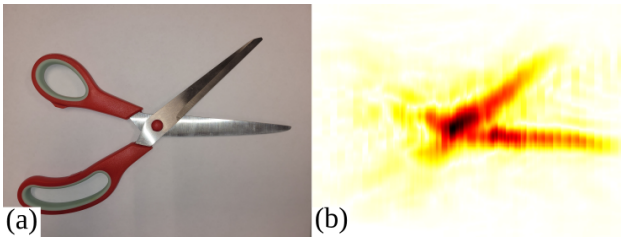

Fig. 2: (a) Ground truth optical image of a scissor with plastic handles; (b) Reconstructed mmWave image with accurate device poses still cannot recover many key features of the object.

Signal Specularity and Diverse Imaging Geometries: Even if the device poses were close to ideal, the reconstructed mmWave image may lack high-frequency human-perceptible features. During handheld motion, the hand may be biased to certain regions within the aperture, causing highly sampled regions to overpower lightly sampled regions in the reconstructed image. Signal specularity may occur when objects are not incident to the aperture plane, preventing the reflected signal from returning to the receiver, so the target would be invisible to the SAR system. Additionally, objects with a weak reflectivity may not appear in the received signal. For example, Figure 2 shows the reconstructed mmWave image of a scissor, where the plastic handles are not visible even though the synthetic aperture was sufficiently sampled.

III. *mmSight* System Design

Overview: *mmSight* improves motion tracking and SAR imaging without using external tracking setups or costly motion controllers. *mmSight* includes a mmWave transceiver and a vision-based self-tracking device that move together in front of a region of interest while collecting pose information and mmWave frames. We propose a pose estimation method to limit pose errors introduced by the vision-based self-tracking

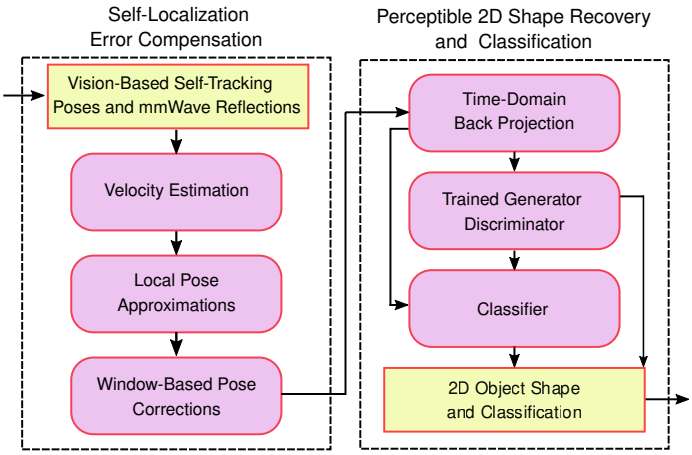

Fig. 3: (a) System overview of *mmSight*.

system. We use the antenna spacings on the mmWave device to recover the velocity and position of the device. After the new poses have been recovered, we apply an overlapping window-based pose correction method to suppress cumulative drift error. Even if the window-based pose correction method succeeded to recover the ground-truth mmWave image, the reconstructed image could still be imperceptible due to weak reflectivity or surface specularity. We customize a deep learning model, Conditional Generative Adversarial Network (cGAN), as in [6], to recover a perceptible 2D image and classify it into a series of objects with a Classifier network. The cGAN learns the relationship between the imperceptible input mmWave data and the appropriate ground-truth image. cGAN discovers portions of the image missing due to specularity or weak reflectivity after training. Figure 3 shows the overview of *mmSight*. We now describe the design components in detail.

A. Velocity Estimation

Since vision-based sensor data from devices such as gyroscopes and accelerometers are susceptible to drift and noise, *mmSight* uses the antenna spacing of the mmWave device to produce a robust local motion estimate. We extend the methods outlined in [16], [17] to produce a velocity estimate using the time delay between separated antennas as the antennas pass through each other's trajectory, and integrate the velocity to retrieve the position of the device. Different from existing velocity estimation methods, we identify prominent points in the reflection data to produce accurate velocity measurements in the presence of highly reflective targets.

For a mmWave device with multiple antennas, the spacing between 2 Tx and the 2 Rx antennas along the direction of motion is $H = (x^{t_1} - x^{t_2}) + (x^{r_1} - x^{r_2})$ [16], where x^{t_1} and x^{r_1} corresponds to the location of the Tx and Rx antennas for a single channel along the direction of motion and x^{t_2}, x^{r_2} correspond to the Tx and Rx antennas for another channel. Consider a device moving in the cross range (X) direction with a single Tx located at x^{t_1} and two Rx antennas located at x^{r_1} and x^{r_2}. For two Rx antennas sharing a single Tx, $H = (x^{r_1} - x^{r_2})$. The two Rx antennas will experience the most signal similarity once the device has moved a distance of $H/2$ along the direction of the antenna spacing [16]. To recover the velocity using the antenna spacing and reflected signals, each received frame i from x^{r_1} is cross-correlated with M previous and M following ($\forall j \in [i - M, i + M]$) frames received from x^{r_2}.

The speed and direction at the i^{th} frame can be determined by identifying the j^{th} frame which produced the maximum value from correlation sequence. Then, the delay at the ith frame is $delay(i) = j - i$. The velocity $V(i)$ corresponds to the speed of the mmWave device at the ith frame [16]:

$$V(i) = \frac{H \cdot fps}{2 \cdot delay(i)} \tag{2}$$

where fps is the frames per second at which the mmWave device captures mmWave frames.

This method can estimate velocity along any direction with an antenna separation. A faster framerate and a larger antenna spacing will produce a more reliable estimate [17], so we average the velocity estimates produced by the other available antenna spacings on the device along a direction of motion to produce a more robust velocity estimate.

Velocity Measurements of Similar Reflectors: Since the velocity estimation relies on the cross-correlation of successive signal measurements, it will not produce correct velocity estimates in the presence of highly similar signals. In cases where the signals are closely similar to each other, the delay sequence may incorrectly fluctuate between extreme values, or report the incorrect velocity altogether. For example, in Figure 4, the mmWave device passes in front of a flat metal surface. The velocity estimation method fails since the flat surface contributes most to the cross-correlation, and each signal is closely similar regardless of the tranceiver position. Previous works such as [16] have used an optimization scheme to recover correct poses based on surrounding samples, but this does not work in the small-scale handheld scenario since the device may pass by a bright object continuously and never experience a change in channel response that can be used to assume the velocity.

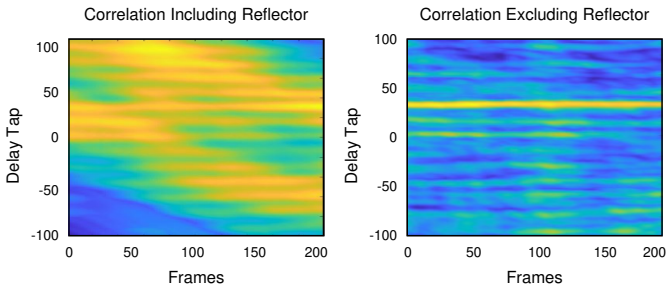

Fig. 4: (a) Correlation delay plots showing multiple frames for a device moving at a constant velocity; (b) Correlation delay plots with bright reflectors removed.

Although the reflectors most prominent in the SAR data contribute the most to the correlation, the surrounding environment is still complex enough to carry information about

the change in signal reflection. We ignore higher intensity contributions in the SAR data for velocity extraction. Fig 4 shows the new correlation plots after removing reflectors. Removing the bright reflectors reveals a clear peak at the correct delay value that can be used for robust velocity estimation.

B. Window-Based Pose Correction

The poses collected from the vision-based self-tracking device are insufficient for SAR focusing since they contain significant local noise and global drift error. *mmSight* corrects the poses to enable SAR focusing using the poses from the vision-based self-tracking device as an initial estimate. We apply an overlapping window-based pose correction method to improve the poses and facilitate improved imaging in two main steps: (1) *mmSight* addresses the local pose error caused by rapidly fluctuating noise that causes the received signals to combine during image reconstruction destructively. (2) *mmSight* addresses cumulative drift errors that cannot be solved by simply approximating the received pose trajectory.

Local Pose Correction: *First,* we produce velocity estimates using the mmWave-based velocity estimation method described in Section III-A so that each mmWave frame and pose has an associated X and Y velocity, named V^x and V^y, respectively. We additionally filter the poses so that each sample is spaced by at least $\frac{\lambda}{4}$, which is enough to the satisfy spatial sampling criterion for image reconstruction [18]. This step improves the scene sampling for the final image reconstruction so that regions with higher spatial sampling bias (*i.e.*, the hand tended to scan the same region several times) do not overpower regions with fewer samples.

Next, we divide the collected data series of poses and mmWave frames into overlapping windows with length D and overlap size Q. For a given window, we use the vision-systems pose X_1 and Y_1 at the start of the window as a reference. Then, we directly integrate the velocity to get the position X_{i+1} and Y_{i+1} of the device from the correlation based method as:

$$X_{i+1} = X_i + \frac{V_i^x}{fps}; \quad Y_{i+1} = Y_i + \frac{V_i^y}{fps} \tag{3}$$

Since there is no range-axis antenna spacing to measure motion along the Z dimension, we assume that the Z-axis can be approximated by a constant line for a sufficiently small window size. The proposed method works since the user moves the device in a relatively straight line for a small window interval, and the filtering approximations suppress the immediately destructive noises.

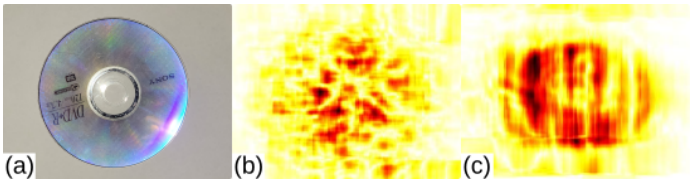

Fig. 5: (a) Ground truth optical image. (b) Reconstructed mmWave image before pose correction; (c) After pose correction.

Global Pose Error: Now, the poses are locally correct, *i.e.*, successive mmWave reflections would not immediately add destructively and could be used to generate locally coherent sub-images. However, the approximated poses still accumulate drift error that would cause the complete image to appear defocused.

We reconstruct voxels using the new overlapping poses and the TDBP Algorithm in chunks defined by the window size D, producing two coherent 3D voxels. The first voxel is produced using poses from W_n within the overlapping region Q, and the second voxel is produced using poses from W_{n+1} in the same region Q. They are both produced using the same range-compressed SAR data and should produce the same voxel if the poses were correct. However, due to the cumulative drift error, the voxels may be shifted from one another. While the sub-images generated from each window are locally coherent, they would still destructively sum during image reconstruction due to this offset.

To correct the shifted windows, we register the generated 3D voxels within W_n and W_{n+1}, with W_{n+1} as the moving window to recover the transformation between the two image voxels to limit the drift error across windows. We use the Normal Distribution Transform (NDT) [19] to align each window with good results. Different from Iterative Closest Point (ICP), NDT can be performed much quicker and considers the global shape. This method is inspired by laser scan matching [20], which finds the pose from two images by finding the transformation between two separate images. However, for our implementation, we are projecting the same mmWave data with poses that vary slightly due to drift errors, so the voxels should match. If the transformation to register W_{n+1}, is the rotation R and translation T, then the new poses can be written as [21]: $(x^*, y^*, z^*) = (x_n^i, y_n^i, z_n^i) \cdot R + T$, where R is the rotation and T is the translation associated with the transformation from W_{n+1} to W_n for all i poses within the window n.

We directly modify the poses for each window based on the linear transformation needed to register the two images. The windows are processed sequentially so that W_{n+1} becomes the fixed window for the moving window W_{n+2} and so on. Finally, we reconstruct the entire mmWave image using the TDBP algorithm with the adjusted poses. Figure 5 shows example mmWave images before and after recovery. The method can reveal the object's coarser features, such as boundaries and shape.

C. Deep Learning-Based Image Quality Improvement

Even with perfectly accurate motion data, the mmWave image reconstruction will still need to recover finer details lost due to specular reflection or weak reflectivity. Additionally, the registration-based pose correction method cannot recover perfectly accurate poses since each window is approximated and smoothed.

We train a conditional Generative Adversarial Network (cGAN) [22] with thousands of examples of pose-corrected mmWave voxels to reconstruct human-perceptible 2D shapes

and classify them into different categories. Different from previous cGAN implementations that generate images of automobiles [23] or that depend on reflection data motion-corrected to a defined 2D grid [6], *mmSight* cGAN operates on mmWave data from a mobile device with significant pose errors. The goal of the cGAN is to learn the association between the imperfect mmWave voxels and the ground-truth images that are useful in several classification tasks. The cGAN teaches a Generator G to generate a ground-truth human-perceptible image from an input pose-corrected 3D mmWave voxel. The Discriminator D improves the G-generated image by determining whether the input to D was fake (generated by G) or real.

GAN fundamentals: GAN is a supervised learning method that generates a new dataset by learning the patterns present in the real input data. GAN includes two models: (1) Generator G, which produces fake samples, and (2) Disciminator D, which tries to determine if the samples that are produced by the G are either real or fake. The GAN concept is described as a zero-sum adversarial game, in which G tries to produce a better sample to fool D, while D tries to identify whether the input samples were fake, *i.e.* produced by G, or part of the real data distribution [24]. But traditional GANs cannot restrict the modes of the generated samples since they do not limit the generated data domain. Thus, we propose to conditionally train *mmSight* on ground-truth images to control the distribution of data generated. The objective function for GAN is [25]:

$$\min_G \min_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log D(1-G(z))] \quad (4)$$

Where, p_{data} is the probability of the samples being drawn from the real data, and p_z is the probability of the samples drawn from the generated dataset. The terms of the objective function in Equation 4 represent the expected value of D to identify a real sample correctly, and the expected value of D to determine generated samples, respectively. These probability values will be between 0 and 1. The entire network should converge whenever D outputs a probability of ∼0.5, indicating that the samples produced by G are indistinguishable from the ground truth samples provided.

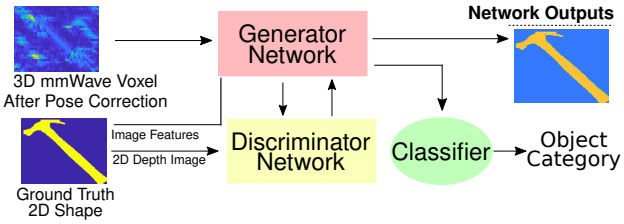

Fig. 6: Overview of *mmSight*'s deep learning framework.

***mmSight* Learning System**: *mmSight* includes a Generator G and a Discriminator D that work together in the cGAN architecture. The Classifier C uses the output 2D image to predict supervised class labels automatically. Figure 6 shows the machine learning model for *mmSight*.

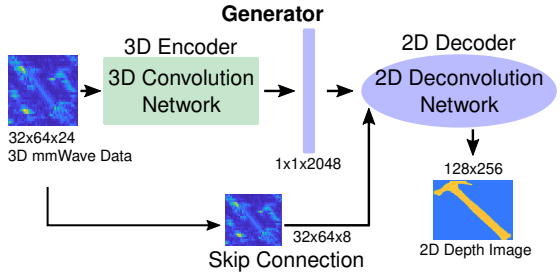

Fig. 7: *mmSight*'s Generator network architecture.

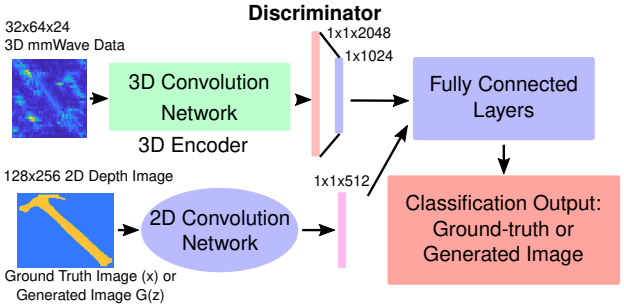

Fig. 8: *mmSight*'s Discriminator network architecture.

Generator: The Generator G is responsible for converting 3D mmWave shapes into 2D shapes with human-perceivable features. To achieve this, we use an encoder-decoder architecture [6], [26] which converts the 3D mmWave voxel shape into a 1D feature vector through several 3D convolution layers and an additional end flatten layer. The 1D representation compresses the 3D mmWave shape so that abstract features can be learned by the system. Table I shows the generator network parameters. Since the object of interest only lies in a few 2D slices of the reconstructed mmWave voxel, the encoder-decoder architecture tends to compress the entire reconstructed volume. G uses a skip connection following [6], [23] that extracts the highest energy 2D slices from the 3D mmWave voxel to preserve the high-frequency features. This connection is concatenated to the 2D deconvolution layer. Figure 7 shows *mmSight*'s generator architecture.

Discriminator: The Discriminator D teaches G to generate realistic 2D shapes through adversarial training. The main goal of D is to increase the probability of D identifying the fake samples from generated samples. Thus, D takes the 3D mmWave shape, and the 2D shape generated by G or provided as ground truth to produce an output probability. D uses similar encoder layers from G to convert the 3D mmWave shape into a 1D feature vector. In place of the decoder layers in G, D leverges several 2D convolution layers to convert the 2D shapes to 1D feature vectors with lengths equal to feature vectors for 3D mmWave shape. Then, these two feature vectors are cascaded together and fed into 2 fully-connected layers. These layers output to a single neuron layer that outputs the probability that the 2D shape is real or fake. Figure 8 shows an overview of the *mmSight* discriminator architecture. Table II shows the discriminator network parameters.

Classifier: *mmSight* recovers the 2D human-perceptible

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

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

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

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

TABLE III: Classifier Network Parameters. 2DC: 2D Convolution (with batch normalization); 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

shape from the pose-corrected mmWave voxel. *mmSight* can be used to automatically classify real-life examples, which would aid in several applications, such as discriminating common objects for security purposes. We define 8 categorical outputs including objects that may be found in an airport bag (CD, scissor, box cutter, metal pen, screw driver, hammer, metal mug, *etc.*) that may be useful to automatically classify. We propose a Classifier C that can be used for such applications, which uses the predicted 2D shape from the Generator G to classify the object automatically. C includes 7 2D convolution layers and 2 fully-connected dense layers. In this implementation, we include 9 output classes, with 1 output designated for miscellaneous items that the network had not seen. We can extend the classifier network to classify more objects for several different use cases. Table III shows the classifier network parameters.

cGAN Loss Functions: *mmSight*'s loss function must measure how well the 2D shape produced by G matches the ground truth, including the shape quality. We use both the traditional GAN loss $L(G)$ [25] and the L1-norm loss $L_1(G)$ [27] to train the cGAN networks. The traditional GAN loss maintains the adversarial game with D and G. The L1-norm loss is specific to this application and takes into account the pixel-to-pixel differences between the generated images and the ground truth images.

$$\begin{aligned} L_{cGAN} &= L(G) + \lambda_I \cdot L_1(G) \\ where, L_1(G) &= \mathbb{E}\|x_1 - G(z_1)\|_1 \end{aligned} \tag{5}$$

The Classifier network C loss L_{class} leverages the L_{cGAN}, categorical loss L_C, and binary loss L_B [6].

$$L_{class}(G) = L_{cGAN} + \lambda_C \cdot L_C(G) + \lambda_B \cdot L_B(G) \tag{6}$$

The hyperparameters, λ_I, λ_C, and λ_B, affect the system's ability to perform accurate shape reconstruction with emphasis on image perceptibility and classification.

IV. Implementation

Hardware: Since existing 5G smart devices do not have an exposed programming interface to control and synchronize the mmWave signals with other on-board hardware modules, we prototype our design with hardware that emulates the sensors available on camera-based self-positioning systems [28]. We use the Intel Realsense T265 Tracking Camera [29], which features two fish-eye lenses and a specialized vision processing unit, for self-localization. The T265 is most susceptible to drift along the Z-axis on the order of several millimeters [30], similar to common smart devices, making it useful as a testing platform. For mmWave signal collection, we use the 77-81 GHz TI IWR1443BOOST [31]. The IWR1443BOOST has 4 Rx and 3 Tx antennas, which can generate up to 12 virtual channels using time multiplexing. Our FMCW parameters include: Frequency slope: 70.295 MHz/µs, ADC Sample Count: 256, Sampling rate: 5 Msps, Ramp duration: 60 µs. These parameters allow the objects within the mmWave device's view to appear quasi-stationary within each frame, even under the speed of handheld motion.

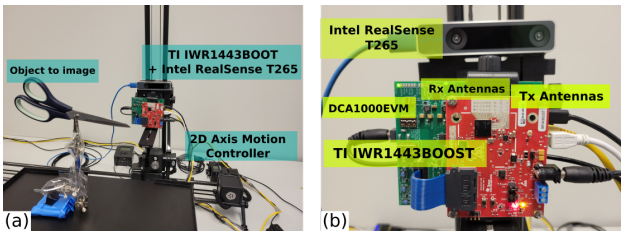

Fig. 9: (a) Experimental setup includes 2D mechanical controller, mmWave device, and vision-based self-tracking unit; (b) TI IWR1443BOOST [31] and Intel RealSense T265 [29].

We post-process the data collected through the window-based pose correction framework and the cGAN on a host PC using Matlab and Python. The post-processing takes the erroneous pose estimates with the associated mmWave reflections and generates the perceptible 2D shape and feature

category of the imaged object. Since the TI mmWave device and the T265 vision-based tracking unit cannot be triggered synchronously, we initially post-process the data so that the mmWave reflections and poses are synchronized. To this end, we keep the device stationary for 10 seconds during the start of the scanning period and align common start points between the mmWave reflections and the poses. Since the T265 samples faster than the TI IWR1443BOOST, we assign the nearest pose sample to each mmWave reflection.

Real Data Collection: It is difficult to obtain sub-millimeter scale ground truth of handheld motion data to compare the pose accuracy before and after pose correction. So, we use fixed motions that still contain significant drift errors from the vision-based self-tracking device for testing. To test the effectiveness of *mmSight* under conditions with position drift error caused by the vision-based self-tracking system's pose estimates, we mount the mmWave device and T265 together onto a 2D mechanical controller moving in a 20×20 cm^2 zig-zag motion. The controller moves at a speed of 2 cm/s while the mmWave device collects frames at a rate of 100 frames per second, which is well within the spatial sampling criterion to prevent aliasing in the ideal case. We collect data for several objects, categorized into 8 different categories, including miscellaneous items, totaling 90 real samples. We also include objects hidden behind a cloth shirt within 10 cm of the objects surface to test the occluded scenario, shown in Fig 10. We post-process the data using the pose-correction framework, which generates mmWave voxels with size $200 \times 200 \times 24$ after image reconstruction with TDBP. We generate several samples by varying the window length and overlap size from 250 to 1000 frames. To process the data into a perceptible 2D shape using the cGAN, we resize the output voxel to $32 \times 64 \times 24$. We additionally extract the 8 highest-energy voxels from the resized mmWave data to provide as skip-data to the cGAN. We manually label each dataset with the associated ground truth image and object category.

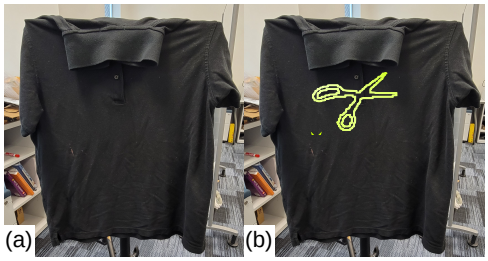

Fig. 10: (a) Occlusion setup with cloth shirt; (b) Object hidden behind occlusion.

Synthetic Data Generation: Since mmWave datasets are not generally available and are time-consuming to collect, we generate thousands of synthetic datasets to train the GAN. We first collect 2D ground truth images and generate their synthetic mmWave reflections and aperture positions using a ray tracing simulator as in [6]. We keep the same mmWave parameters previously described for the simulator. The simulator can generate several mmWave voxels at varied depths, orientations, *etc.* Once the ground-truth aperture positions and reflections have been generated, we apply a mean and standard deviation motion error, varying from 0 to 1 cm and 0 to 1 mm, respectively, to all the points to simulate imperfect pose correction. We use TDBP to reconstruct the voxels. Then, we resize the output voxel to a $32 \times 64 \times 24$ mmWave voxel for input to the cGAN.

Training: We train *mmSight* with thousands of samples to learn the association between the pose-corrected mmWave shape and the ground truth 2D image. Since real mmWave datasets for specific applications are limited, we primarily train *mmSight* using synthetic data. We first train the cGAN with 8500 synthetic samples for 1000 epochs. Then, we train the cGAN with real collected datasets. Both the cGAN and classifier network architectures are implemented in Python using TensorFlow 2.1 [32]. We train them together on a machine with 32 CPU cores @ 2.8GHz, 256 GB RAM, and an NVIDIA RTX A6000 core [33]. These networks take approximately 50 hours to train.

V. Performance Evaluation

We use the following metrics to evaluate *mmSight*:

- Mean Squared Error (MSE) - Measures the absolute distance between the ground truth poses and the poses estimated by *mmSight*.
- Structural Similarity Index Measure (SSIM) - Measures the structural similarity between two images [34]. A perfectly reconstructed image has an SSIM of 1.
- Classification Confusion Matrix - Columns represent the actual values, and rows represent the predicted values. Each value corresponds to the % of samples predicted.

Summary: (1) *mmSight* improves the velocity estimate by an average of 17 mm/s compared to the baseline T265 measurements. (2) *mmSight* reduces the magnitude of large drift error in the self-positioning device, especially along the Z (range) dimension, which suffers from the most error, from a median of 5 mm to 1 mm, to enable SAR focusing. *mmSight* improves the median SSIM of the output image voxel before and after pose correction from 0.01 to 0.08. (3) *mmSight*'s shape improvement and classification framework further increases the median SSIM to 0.92 and achieves 98% object classification accuracy.

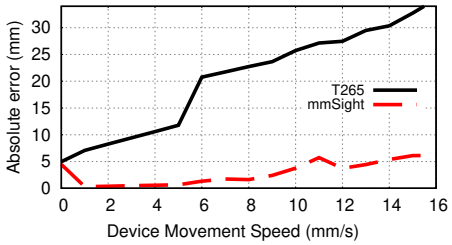

Fig. 11: Effect of device movement speed on pose error.

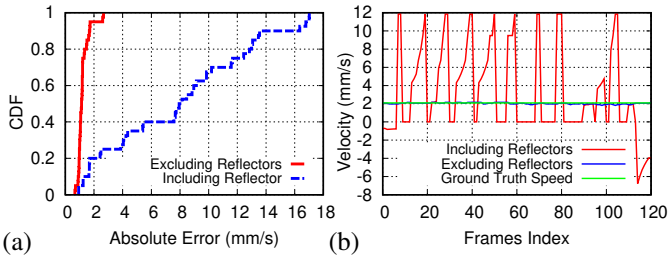

Fig. 12: (a) Velocity measurement error before and after excluding bright reflectors for velocity estimation; (b) Velocity output while the device moves in front of a flat reflector before and after excluding bright reflectors.

A. Velocity Estimation

We test the accuracy of *mmSight*'s velocity estimation method under various conditions, including: (1) Flat, incident reflectors that contribute to high signal similarity and (2) Varied speed. We use the same setup for imaging but restrict the motion to a single linear motion of 20 cm along the X-axis. We perform the tests in an office environment with many whiteboards, desks, and walls within the field of view of the device. To test the velocity measurement accuracy at varying speeds, we ensure that there are no objects at least a meter within the device's field of view. Then, we move the controller at several speeds. Figure 11 shows the error of the mmWave-based velocity estimation method compared with the velocity output reported by the T265. It can be seen that the T265 device incurs more significant errors at faster motions, while the error remains limited under the correlation-based method.

To test the accuracy of the velocity estimation method after introducing flat reflectors, we place a 20×24 cm^2 metal plate directly parallel to the aperture plane. We vary the distance of the metal plate from 10 cm to 100 cm. For all trials, we keep the velocity of the motion controller at 2 cm/s, which would provide a reliable velocity estimate. Figure 12 shows the velocity estimation error before and after introducing flat reflectors for a single linear motion. It can be seen that automatically removing bright reflectors allows the velocity estimation to succeed as if there were no homogeneous reflectors.

B. Window-Based Pose Correction

Since the velocity estimation method cannot be applied to all poses exhaustively due to Z-translation motion, we now evaluate the window-based pose correction technique (Section III-B) with the velocity estimation to identify any imaging improvements.

Figure 16 shows the ground truth optical 2D shape and the mmWave image generated with and without window-based pose correction. Although the window-based pose correction method cannot completely recover the shape, we can use key features to determine the object. Since the ground truth image is a masked version of the original optical image, the pose correction method only improves the SSIM compared to the optical image from .01 to .08. For example, the millimeter-wave image of the hammer shown in Figure 16 does not visually match the optical image since the rubber handles are not visible to traditional SAR.

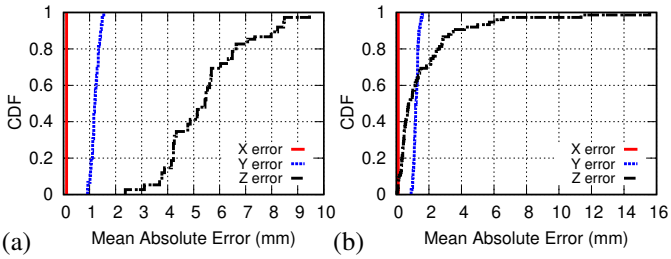

Fig. 13: (a) Device localization error before pose correction; (b) Residual error after pose correction.

Figure 13 shows the pose error before and after applying window-based pose corrections. The Z-axis suffers from the most errors, but the pose correction method suppresses most of the Z errors.

C. Shape Improvement and Classification

Shape Recovery from GAN: The output pose-corrected mmWave voxel may not be human-perceptible due to signal specularity, signal loss, and imperfect pose correction. For example, the pose correction method recovered the arch of the head of the hammer but failed to recover the handle because mmWave reflections do not return to the mmWave device (see Figure 16). Figure 15 shows a few objects after processing the output voxel through *mmSight*'s cGAN networks. Additionally, Figure 14 shows the SSIM of the pose-corrected image and the generated images across 1000 samples. For most samples, the SSIM improves to within 0.90. The pose-correction step is necessary to improve the outcome of the GAN.

Classifier: The Classifier C predicts an object category given a 2D mmWave shape. We select 520 test samples (65 samples from 8 object categories) and use cGAN to produce their 2D shape. We input these 2D shapes to C and record the predicted class labels. Table IV shows the confusion matrix related to the labels that C automatically classified the objects with, which shows around 98% accuracy in predicting the class of objects. Table V shows the confusion matrix for occluded objects.

VI. Related Works

Position Tracking: Methods for building a map and pose rely on 2D range maps generated by a sensing device, such as LiDAR [35]. Scanning radar can perform coarse-grained motion tracking, but such radar sensors are expensive and large [36]. [37] use IMU-integrated multiple scans to produce a more feature-rich scan to find the device pose from traditional scan-matching on single-chip mmWave radar. RF-based Intertial Measurement (RIM) [17] achieves motion tracking by monitoring the Channel State Information from transmitted packets and leverages MIMO hardware to deduce the device's motion. RIM can measure the distance, heading direction, and rotation angle in a 2D plane, but cannot successfully measure 3D motion due to limitations in physical antenna configurations. In contrast, our design uses the mmWave reflections and the poses reported by a coarse-grained tracking

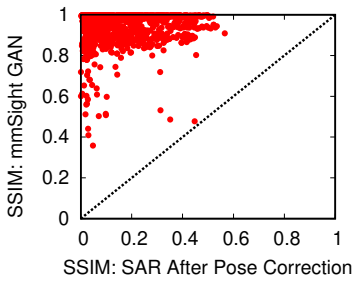

Fig. 14: *mmSight*'s cGAN improves SSIM for pose-corrected images.

TABLE IV: Confusion matrix of categorical classifier in *mmSight*.

Predicted/Actual	Knife	Toy Gun	Scissor	CD	Pen	Hammer	Clip	Screwdriver	Other
Knife	**98.5**	0	1.5	0	0	0	0	0	0
Toy Gun	0	**100**	0	0	0	1.5	0	0	0
Scissor	0	0	**98.5**	0	0	0	1.5	0	1.5
CD	0	0	0	**98.5**	0	0	0	0	1.5
Pen	0	0	0	0	**100**	0	0	0	0
Hammer	0	0	0	0	0	**98.5**	0	0	0
Clip	1.5	0	0	1.5	0	0	**98.5**	0	1.5
Screwdriver	0	0	0	0	0	0	0	**100**	0
Other	0	0	0	0	0	0	0	0	**95.5**

TABLE V: Confusion matrix of categorical classifier in *mmSight* for hidden objects.

Predicted/Actual	Knife	Toy Gun	Scissor	CD	Pen	Hammer	Clip	Screwdriver	Other
Knife	**98.5**	0	0	0	0	0	1.5	0	0
Toy Gun	0	**100**	0	0	0	0	0	0	0
Scissor	0	0	**100**	0	0	0	0	0	0
CD	0	0	0	**98.5**	0	0	1.5	0	0
Pen	0	0	0	0	**100**	0	0	0	0
Hammer	0	1.5	0	0	0	**98.5**	0	0	0
Clip	0	0	0	0	0	0	**100**	0	0
Screwdriver	0	0	0	0	0	0	0	**100**	0
Other	0	0	1.5	0	0	0	0	0	**98.5**

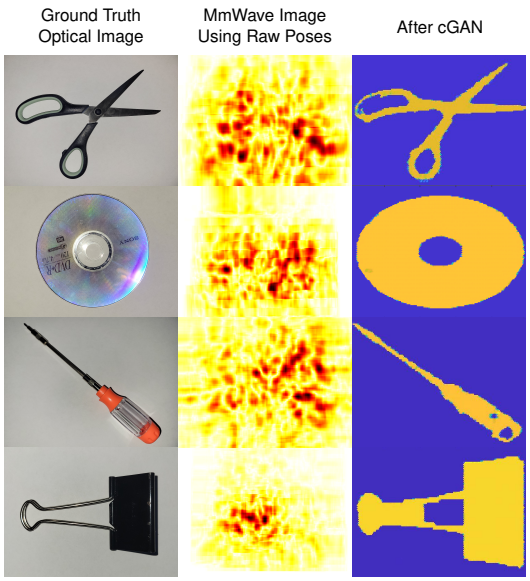

Fig. 15: Multiple *mmSight* reconstructions after cGAN.

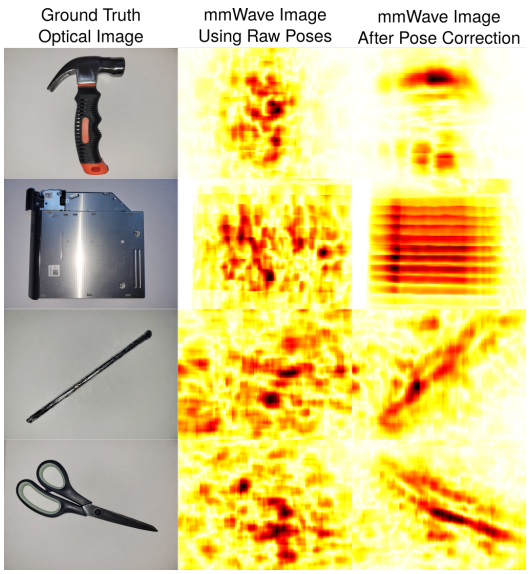

Fig. 16: MmWave image reconstructions after pose correction.

device to limit the pose errors without modifying the hardware to include antenna spacings along the Z dimension.

Mobile MmWave Imaging: [30] directly uses a similar stereoscopic self-tracking system to produce images with freehand trajectories and a delay-and-sum reconstruction algorithm, but is still susceptible to drift caused by the vision-based self-tracking device and suffers from defocusing. Radar sensors are already widely used in parking assistance and can estimate coarse ego-motion through doppler [38]; however, they can only estimate motion up to several centimeters or meters, which would not support high-quality SAR imaging. *Millipoint* [16] uses a similar motion-tracking method as in RIM to achieve the sub-millimeter accuracy required for focusing for vehicles moving in a strictly linear motion.

MmWave Image Resolution Improvement: *HawkEye* [23] uses a conditional Generative Adversarial Network architecture [22] to learn the mapping from a 3D mmWave heatmap to a 2D perceptual depth map image to address low-resolution mmWave imaging. [6] produces perceptible mmWave images for the handheld case by using motion compensation to adjust the poses and their associated radar sample so that they all lie on a uniformly sampled grid, sufficient for FFT-based image reconstruction. They use a cGAN network to learn the association between 3D mmWave shapes and their perceptible 2D images containing high-frequency information.

VII. Conclusion

This paper proposes *mmSight*, a system for mmWave imaging on mobile smart devices. *mmSight* emulates the principle of SAR imaging by correcting the erroneous poses from the vision-based self-tracking device through an initial local pose correction and then a global drift error correction. *mmSight* includes a velocity estimation method to correct local pose errors and addresses the depth drift error with linear approximations. Then, the system adjusts the locally accurate poses using a 3D feature registration method to recover globally accurate poses sufficient for improved SAR focusing. *mmSight* applies cGAN and Classifier networks to recover the perceptible 2D shape and label of the objects, demonstrating that *mmSight* can adapt to a wide variety of applications requiring beyond line-of-sight vision.

VIII. Acknowledgments

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

References

[1] Wang, Zhongmin and Chang, Tianying and Cui, Hong-Liang, "Review of active millimeter wave imaging techniques for personnel security screening," *IEEE Access*, vol. 7, 2019.

[2] Hong, Wonbin and Baek, Kwang-Hyun and Ko, Seungtae, "Millimeter-wave 5G antennas for smartphones: Overview and experimental demonstration," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, 2017.

[3] David M. Sheen and Douglas L. McMakin and Thomas E. Hall, "Near Field Imaging at Microwave and Millimeter Wave Frequencies," in *IEEE/MTT-S International Microwave Symposium*, 2007.

[4] Yanik, Muhammet Emin and Torlak, Murat, "Millimeter-wave near-field imaging with two-dimensional SAR data," in *Proc. SRC Techcon*, 2018.

[5] M. I. Duersch and D. G. Long, "Backprojection Autofocus for Synthetic Aperture Radar," 2013. [Online]. Available: https://digitalcommons.usu.edu/spacegrant/2013/Session2/1/

[6] Regmi, Hem and Saadat, Moh Sabbir and Sur, Sanjib and Nelakuditi, Srihari, "SquiggleMilli: Approximating SAR Imaging on Mobile Millimeter-Wave Devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, 2021.

[7] Saadat, Moh Sabbir and Sur, Sanjib and Nelakuditi, Srihari and Ramanathan, Parmesh, "Millicam: Hand-held millimeter-wave imaging," in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2020, pp. 1–9.

[8] Regmi, Hem and Saadat, Moh Sabbir and Sur, Sanjib and Nelakuditi, Srihari, "ZigZagCam: Pushing the Limits of Hand-Held Millimeter-Wave Imaging," in *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications*, 2021.

[9] Iovescu, Cesar and Rao, Sandeep, "The fundamentals of millimeter wave sensors," *Texas Instruments*, 2017.

[10] Duersch, Michael I and Long, David G, "Analysis of time-domain back-projection for stripmap SAR," *International Journal of Remote Sensing*, vol. 36, no. 8, 2015.

[11] A. Ribalta, "Time-domain reconstruction algorithms for fmcw-sar," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 396–400, 2010.

[12] Hu, Kebin and Zhang, Xiaoling and He, Shufeng and Zhao, Hanxing and Shi, Jun, "A less-memory and high-efficiency autofocus back projection algorithm for sar imaging," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 4, 2014.

[13] A. Poulose and D. S. Han, "Hybrid indoor localization using imu sensors and smartphone camera," *Sensors*, vol. 19, no. 23, 2019.

[15] Morrison, Robert L and Do, Minh N and Munson, David C, "SAR image autofocus by sharpness optimization: A theoretical study," *IEEE Transactions on Image Processing*, vol. 16, no. 9, 2007.

[14] A. Alapetite, Z. Wang, J. P. Hansen, M. Zajkaczkowski, and M. Patalan, "Comparison of three off-the-shelf visual odometry systems," *Robotics*, vol. 9, no. 3, 2020.

[16] Qian, Kun and He, Zhaoyuan and Zhang, Xinyu, "3D Point Cloud Generation with Millimeter-Wave Radar," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 4, 2020.

[17] Wu, Chenshu and Zhang, Feng and Fan, Yusen and Liu, KJ Ray, "RF-based inertial measurement," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019.

[18] Sheen, David M and McMakin, Douglas L and Hall, Thomas E, "Three-dimensional millimeter-wave imaging for concealed weapon detection," *IEEE Transactions on microwave theory and techniques*, vol. 49, no. 9, 2001.

[19] Biber, Peter and Straßer, Wolfgang, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, 2003.

[20] Diosi, Albert and Kleeman, Lindsay, "Laser scan matching in polar coordinates with application to SLAM," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[21] Ulaş, Cihan and Temeltaş, Hakan, "3d multi-layered normal distribution transform for fast and long range scan matching," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 1, 2013.

[22] Mirza, Mehdi and Osindero, Simon, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[23] J. Guan, S. Madani, S. Jog, S. Gupta, and H. Hassanieh, "Through fog high-resolution imaging using millimeter wave radar," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 461–11 470.

[24] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[26] Badrinarayanan, Vijay and Kendall, Alex and Cipolla, Roberto, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, 2017.

[27] 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, 2016.

[28] "Zenfone." [Online]. Available: https://www.asus.com/us/Mobile/Phones/All-series/

[29] "Intel RealSense Tracking Camera T265." [Online]. Available: https://www.intelrealsense.com/tracking-camera-t265/

[30] G. Álvarez-Narciandi, J. Laviada, and F. Las-Heras, "Towards Turning Smartphones Into mmWave Scanners," *IEEE Access*, vol. 9, 2021.

[31] Texas Instruments, "IWR1443 Single-Chip 76-to-81GHz mmWave Sensor Evaluation Module," 2018. [Online]. Available: http://www.ti.com/tool/IWR1443BOOST

[32] "Tensorflow 2.1." [Online]. Available: https://www.tensorflow.org/

[33] "Nvidia rtx a6000." [Online]. Available: https://www.nvidia.com/en-us/design-visualization/rtx-a6000/

[34] 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.

[35] Hess, Wolfgang and Kohler, Damon and Rapp, Holger and Andor, Daniel, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*, 2016, pp. 1271–1278.

[36] Callmer, Jonas and Törnqvist, David and Gustafsson, Fredrik and Svensson, Henrik and Carlbom, Pelle, "Radar slam using visual features," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, pp. 1–11, 2011.

[37] Li, Yang and Liu, Yutong and Wang, Yanping and Lin, Yun and Shen, Wenjie, "The millimeter-wave radar slam assisted by the rcs feature of the target and imu," *Sensors*, vol. 20, no. 18, 2020.

[38] Kellner, Dominik and Barjenbruch, Michael and Klappstein, Jens and Dickmann, Jurgen and Dietmayer, Klaus, "Instantaneous ego-motion estimation using doppler radar," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013.