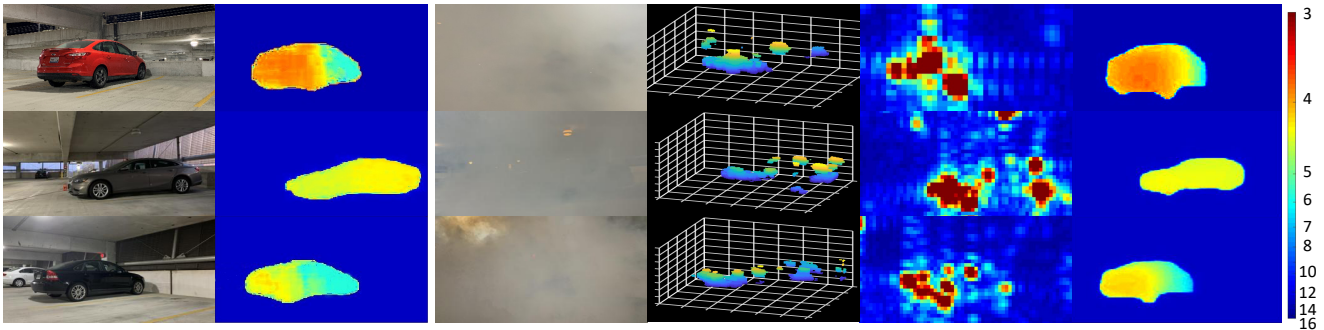
Through Fog High Resolution Imaging Using Millimeter Wave Radar

Junfeng Guan Sohrab Madani Suraj Jog Saurabh Gupta Haitham Hassanieh
University of Illinois at Urbana-Champaign



(a) Original Scene (b) Ground Truth (c) Camera in Fog (d) Radar Point Cloud in Fog (e) Radar Heatmap in Fog (f) Recovered Image

Figure 1: Millimeter wave radar can image through fog and bad weather. However, specularity, artifacts and low resolution result in poor perceptual quality as shown in Figure (d) and (e) above. To overcome this limitation, this paper leverages a cGAN architecture to recover high resolution images from the low resolution mmWave radar heatmaps. The figure above shows (a) the original scene, (b) ground truth depth map captured with stereo camera and cropped using MaskRCNN (before fog is generated), (c) the scene in fog, (d) the millimeter wave radar point cloud generated in the presence of fog, (e) the corresponding radar heatmap, and (f) the recovered output of our system.

Abstract

This paper demonstrates high-resolution imaging using millimeter wave (mmWave) radars that can function even in dense fog. We leverage the fact that mmWave signals have favorable propagation characteristics in low visibility conditions, unlike optical sensors like cameras and LiDARs which cannot penetrate through dense fog. Millimeter wave radars, however, suffer from very low resolution, specularity, and noise artifacts. We introduce HawkEye, a system that leverages a cGAN architecture to recover high-frequency shapes from raw low-resolution mmWave heatmaps. We propose a novel design that addresses challenges specific to the structure and nature of the radar signals involved. We also develop a data synthesizer to aid with large-scale dataset generation for training. We implement our system on a custom-built mmWave radar platform and demonstrate performance improvement over both standard mmWave radars and other competitive baselines.

1. Introduction

Achieving fully autonomous vehicles, referred to as *Level 5* in the standards for driving automation, has gain significant interest from major companies like Tesla, Ford, Honda, Waymo, Toyota, Uber, and NVIDIA [6, 7, 22, 37, 48, 60] *Level 5* autonomy requires the ability to operate in severe weather conditions such as dense fog, smog, snowstorms, and sandstorms [22, 23]. Autonomous vehicles, however, mainly use cameras or LiDARs, to obtain an accurate and reliable view of the environment, which suffer in low visibility conditions and bad weather [39, 51, 52, 8, 59, 21]. Cameras also suffer at night in low light conditions. This is problematic as many manufacturers including Tesla avoid using LiDAR altogether, making cameras their primary sensory module [58].[1]

Millimeter wave (mmWave) radars offer more favorable characteristics due to their ability to work at night and penetrate through fog, snow and dust [25, 11]. However, car manufacturers today, still use mmWave radar for the sole purpose of unidirectional ranging, i.e., to determine the distance to other vehicles [24, 9]. Imaging using mmWave radar is challenging for several reasons. First, the resolution of mmWave is extremely low compared to LiDARs or cameras. Fig. 1 (d,e) show an example where the mmWave image appears as blobs of radar reflections and carries little to no contextual and perceptual information as compared to the corresponding camera shown in Fig. 1 (a). Second, unlike optical signals, wireless mmWave signals are highly specular, i.e., the signals exhibit mirror-like reflections from the car [43]. As a result, not all reflections from the car propagate back to the mmWave receiver and major parts of the car do not appear in the image, making it impossible to detect its shape as can be seen in Fig. 1 (d,e). Finally, wireless reflections from the car can also bounce off the road and other cars and travel along multiple paths to the mmWave receiver creating shadow reflections and artifacts in various

[1] Other modalities such as thermal imaging also fail in dense fog [5].

locations in the scene as shown in Fig. 1 (d,e).

Today's commercial mmWave imaging systems, like airport scanners, use human-sized mechanically steerable arrays to improve the resolution. They also isolate the object being imaged in the near field to eliminate multipath reflections and rotate the arrays around the object to address specularity [45, 47]. However, such a design would be extremely bulky and not practical for self-driving cars as we have no control over the cars being imaged.

In this paper, we present HawkEye, a system that consolidates advances to enable practical use of mmWave imaging in realistic applications. Using a mmWave radar, we can sense and predict shape for cars in the presence of dense fog. Figure 1 column (e) shows our system's predicted depth (as predicted from the sensed radar signal visualized in column (d)). Column (b) shows the ground truth depth map (recovered using a stereo camera in the absence of fog). While obviously, we are far from a practical system that can use mmWave imaging on board self-driving cars, we have made huge advances towards this. This paper describes the different aspects that have enabled this advance.

Our central contribution is to cast the problem of predicting high-frequency shape from raw mmWave heatmaps as a learning problem. Use of learning provides robustness to hard-to-model radar reflections and sources of noise like specularity and multipath reflections. At the same time, learning can effectively leverage priors on shapes of cars to make reasonable predictions from coarse mmWave heatmaps. However, use of learning for this task was non-trivial. We had to innovate on the design of the neural network, loss functions for training, and development of large-scale realistic datasets for training. We employed Generative Adversarial Network [31] based architectures that consume mmWave heatmaps and predict high-resolution depth maps. The specific nature of the signal required the design of custom neural network architectures (that map 3D input heatmaps to 2D depth maps, and skip connections that project 3D information to 2D), and custom loss functions (combination of perceptual, L_1, and adversarial loss). We built a realistic radar data synthesizer that captures unique characteristics of radar. We used this module to create a synthesized dataset which was used to train our expressive neural networks. Finally, we built a real-world data collection platform to collect real data for fine-tuning and benchmarking. These all collectively enable the end-to-end system of HawkEye, which to the best of our knowledge, is the first system that can deliver the results shown in Fig. 1(f).

HawkEye has four modules as shown in Fig. 2: (i) A custom-built mmWave imaging module to capture 3D mmWave heatmaps, (ii) A wide baseline stereo camera system to capture high-resolution 2D depth maps for ground truth, (iii) Synthesizer to augment training dataset with synthesized data from 3D CAD models of cars and mmWave

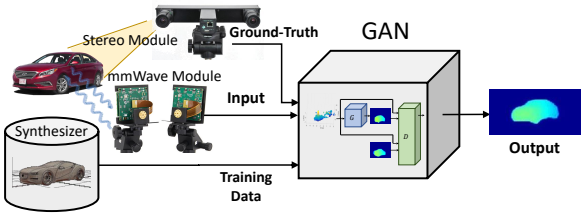


Figure 2: HawkEye's System Overview.

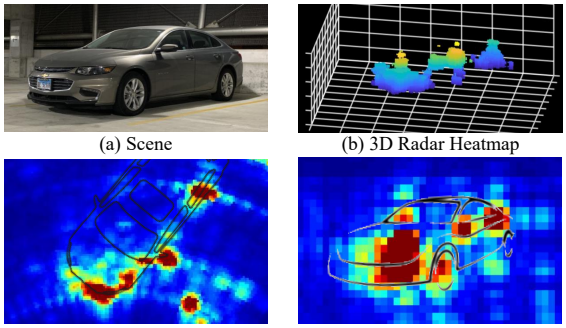
ray tracing algorithms, and (iv) a GAN architecture customized for mmWave imaging in the context of self-driving cars. Our results show that HawkEye is able to generate high-resolution depth maps from raw 3D mmWave heatmaps and accurately reconstruct the car in the real scene even in low visibility conditions like fog. Our data synthesizer code and datasets are available at our project webpage.

2. Related Work

Super-Resolution: Neural networks have been used to increase the resolution of camera images and near-Infrared images [28, 32, 38, 40]. Such techniques rely on the correspondence of image patches between low and high resolution images and can achieve an upscaling factor of $4\times$. The closest to our work are techniques for upsamplng sparse 3D LiDAR data to create dense 2D depth maps [18, 12, 34, 61, 17, 49]. However, these works either require an RGB camera in addition to LiDAR [18, 12, 34, 61] and, hence, do not work in low visibility conditions, or rely on high frequency visual features like edges to cluster and upsample objects [17, 49]. Millimeter wave images, however, have significantly lower spatial resolution where high frequency visual features like boundaries and edges are not apparent. Millimeter wave also suffers from artifacts and specularities that cannot be addressed with traditional super-resolution and upsampling techniques.

LiDAR in Fog: Recent work aims to improve the performance of LiDAR in fog [39, 51, 52]. However, even state-of-the-art research systems either require knowing a depth map of the scene a priori [39] or work only when the object is static by estimating the statistical distribution of the photon reflected off the object [51, 52]. These systems also work only up to 54 cm and have limited resolution (32×32 pixels) and field of view. Millimeter wave radar, on the other hand, can penetrate through dense fog and does not require the object to be static [29, 14].

Radar Imaging Systems: There exist millimeter wave radar imaging systems that can achieve high resolution [47, 55, 53, 27, 45]. However, these systems can only work at very near distances ($< 50cm$) and use very bulky human-sized arrays similar to airport security scanners [47]. Other radar systems that can achieve high resolution at longer distances, are integrated with optical components like a large focusing lens and a mechanically scanning raster [46, 27, 2].



(a) Scene (b) 3D Radar Heatmap

(c) 2D Top View of Radar Heatmap (d) 2D Front View of Radar Heatmap

Figure 3: Output of the Millimeter Wave Imaging Radar.

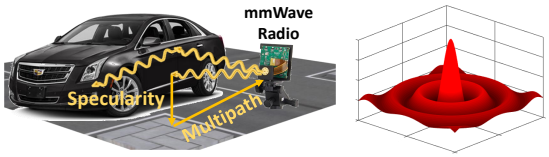
Hence, they are bulky and perform poorly on mobile platforms like self driving cars [55].

Past work also leverages deep learning in the context of millimeter wave radar data. [15] extends the PointNet architecture from [10] to perform 2D object detection from radar data but cannot perform high resolution depth imaging. [19] and [3] apply neural networks to radar acquisitions to enhance their resolution. Both [19, 3], however, work only at short distances and use radar data both as input and ground-truth to their system, making them inherently incapable of dealing with challenges like specularity and multipath. HawkEye, on the other hand, achieves much better results by training using high resolution depth maps to recover the visual representation of the cars and learn to cope with specularity and multipath.

Recent work showed significant progress in using low frequency wireless radar (below 6 GHz) to estimate the 3D pose of humans and track them through walls and occlusions [1, 64, 62, 63]. The work leverages human motion to combat specularity by combining reflections from different body parts over time and stitching them to form the full human body. The work also uses deep convolutional neural networks to label limbs and joints and map them to 3D models of the human skeleton. However, unlike humans in indoor settings, cars move as one single rigid body and only a single viewpoint of the car is typically observed in practice. Therefore, even during motion, most portions of the car will remain invisible due to specularity. Our system adopts a conditional GAN [44] architecture that is able to address specularity without relying on the object's mobility and, hence, can also image static objects like parked cars and cars stopped at traffic lights.

3. Millimeter Wave Imaging Background

Millimeter wave radar works by transmitting a wireless signal and receiving back the reflections from various objects in the scene. It operates in the high frequency bands such as 24 GHz, 60 GHz, and 77 GHz and uses techniques like FMCW (Frequency Modulated Continuous Wave) and



(a) Specularity and Multipath (b) 2D sinc function

Figure 4: Challenges in Millimeter Wave Imaging.

antenna arrays to separate the received reflections.[2] The antenna arrays are electronically steered to capture and separate reflections from the scene based on their spatial direction (ϕ, θ) whereas FMCW is used to separate reflections based on the range (ρ) from the reflecting object. This allows us to compute a 3D heatmap $x(\phi, \theta, \rho)$ where each point represents the energy reflected from a voxel in space.

Figure 3 shows an example of the output of our radar. The 3D heatmap corresponding to the car in (a) is shown as a point cloud in (b). The point cloud is generated by thresholding out the voxels where the reflected signal energy is very weak. We also show projections of the 3D heatmap in the 2D top view in (c), and the 2D front view in (d). The figure also overlays the car's silhouette on the 2D heatmaps to better demonstrate where the reflections are coming from.

As can be seen from the figure, mmWave radar imaging has fundamentally different challenges compared to camera and LiDAR data. First, the imaging resolution is significantly lower than vision. For example, in our system, the range resolution is 10 cm which is $3.3\times$ worse than that of the commercial LiDAR [42]. The azimuth and elevation resolution is 5° which is $50\times$ worse than LiDAR [42]. Range resolution depends on the FMCW signal bandwidth and can potentially be improved using more expensive hardware. However, angular resolution depends on the aperture of the antenna array. To achieve sub-degree angular resolution, similar to LiDAR, we would need a 9 m long antenna array which is impractical both in terms of cost and form factor.[3] For practical aperture sizes (few centimeters), the output mmWave radar image gets convolved with a very wide 2D *sinc* function along the azimuth and elevation dimensions similar to the one shown in Fig. 4 (b). The 2D *sinc* function eliminates almost all high frequency perceptual content such as object boundaries. That is why the mmWave image in Fig. 3 looks like blobs. The *sinc* sidelobes also create artifacts and noise in the image as can be seen in the 2D projections in Fig. 3

Resolution, however, is not the only challenge. Unlike light, mmWave signals do not scatter as much and mainly reflect off surfaces. Hence, the car is highly specular and acts as a mirror reflector of radar signals. As a result, most

[2] At such high frequencies, there is abundant bandwidth available for FMCW signals. The signal wavelength is also small (millimeters) which enables the design of large compact antenna arrays [54, 65].

[3] Note that for systems like the airport security scanners, the target being imaged is in short range and hence, human sized arrays are sufficient.

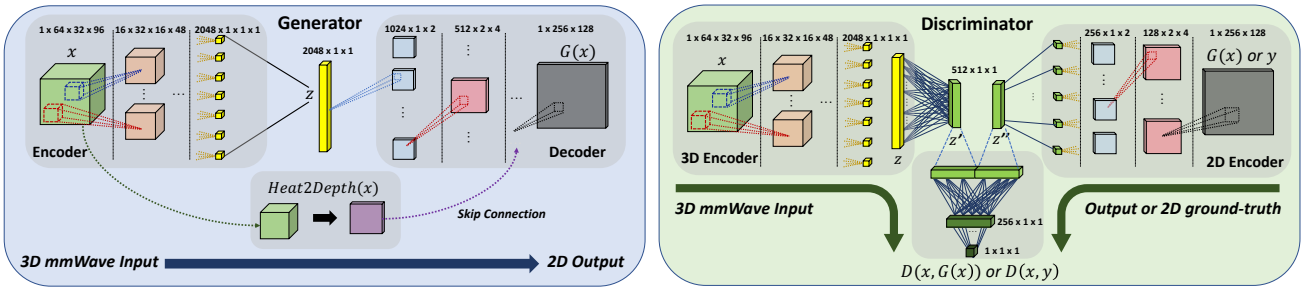


Figure 5: Network Architecture of the Generator G and Discriminator D.

reflections never trace back to the mmWave receiver. This leads to specularity as shown in Fig. 4 (a), making certain portions of the car impossible to image as can be seen in Fig. 3, where a large portion of the car's surface is missing. Moreover, due to multipath propagation, some reflections bounce off the street and other obstacles and trace back to the receiver as shown in Fig. 4 (a) creating many artifacts in the image as can be seen in Fig. 3 (c,d). Finally, radar data has a different representation and perspective as compared to cameras. We must accommodate the above challenges in designing a neural network framework that is able to recover the shape, size, location, ... of the car being imaged.

4. HawkEye's Architecture

Overview: We propose a conditional GAN [44] based architecture. Given an input mmWave RF heatmap x, we learn a conditional generator G. This conditional generator employs an encoder decoder architecture. Although mmWave heatmaps have low spatial resolution, they can achieve high resolution in the depth dimension due to their large sensing bandwidth. In order to retain these high-frequency details in depth, we use skip-connections [50] in our design. Our discriminator D takes in (x, y) or $(x, G(x))$ pairs, and learns to discriminate between them. The generator and discriminator are trained jointly, with the discriminator trying to distinguish generated output from ground truth, and the generator trying to fool the discriminator. We additionally use L_1 and perceptual losses to make the output of the generator consistent with the instance being input to it. Figure 5 shows our architecture. The peculiarities of the raw mmWave signal requires us to carefully consider the design choices involved. We next provide necessary technical details and emphasize the important design choices. Further details of HawkEye's neural network architecture can be found in the supplementary material.

Input and Output Representation: We use the per-voxel energy in the sensed mmWave heatmap as our input representation in the 3D spherical coordinate system (ϕ, θ, ρ). The output from the GAN is the high-frequency shape of the object. We represent the predicted shape in the form of a 2D depth map in the stereo camera frame, where the GAN predicts the depth for each pixel in the image. The generator learns a mapping from $\mathbb{R}^{64\times32\times96}$ to $\mathbb{R}^{256\times128}$.

Most past works that employ conditional GANs, either study 2D to 2D or 3D to 3D transformations. However, our problem requires design of a hybrid 3D to 2D transformation. The sensed mmWave signal is very low resolution, and we do not want to introduce further aliasing by projecting the sensed 3D heatmap to 2D. At the same time, we desire the high-frequency shape as output. It is challenging to predict high-resolution 3D heatmap for computational and optimization reasons [57, 41]. Thus, we chose to represent our outputs as 2D depth maps. Further, note that mmWave signals only provide reflections from metal surfaces in the line-of-sight, since mmWave is shielded by metal surfaces. Thus, the 2D depth map representation of the car serves as a meaningful intermediate representation that can be post-processed to construct full 3D predictions.

Generator Architecture: We follow standard encoder-decoder architecture [4] for representing the generator. The generator is implemented using a deep neural network that maps the input 3D heatmap to a low-dimensional representations z using the encoder. This low-dimensional representation is used by a decoder to produce the 2D depth prediction. We use a 2048 dimensional z-vector. The encoder consists of 6 $3D$ convolution layers along with Leaky-ReLU activation functions and BatchNorm layers. The decoder starts with the 2048 dimensional z vector and uses 8 deconvolution layers to produce the 2D depth map. We use BatchNorm and ReLU activations for the deconvolution layers. Finally, we apply the hyperbolic tangent function alongside a linear transformation that map the output of the generator to the absolute depth in the scene.

Skip Connections: We also use skip connections [50] in the generator. Skip connections provide higher layers in the decoder with high-frequency information from the input / early layers of the encoder. While this is simple in networks that map 2D to 2D or 3D to 3D, our network learns a 3D to 2D projection. Thus, our skip connection design projects the input 3D heatmap into a 2D image, which is concatenated with the higher layers of the decoder. This projection is done as follows. We compute the following 2D image by recording the location that corresponds to the highest value

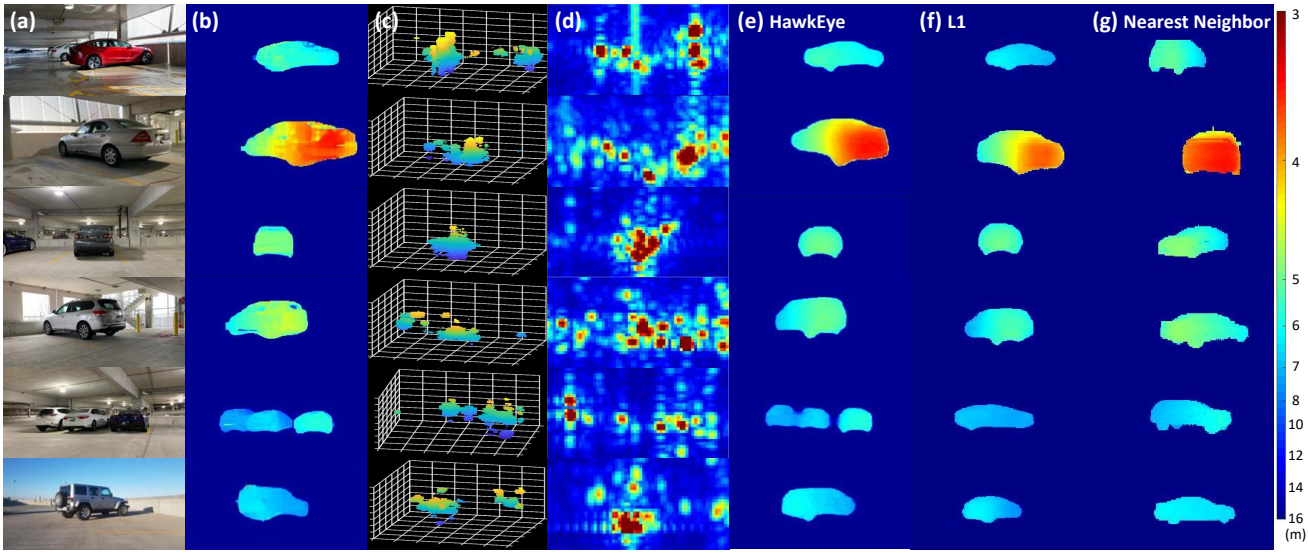


Figure 6: HawkEye's qualitative performance on real test data in clear weather conditions. Column (a) and (b) show the original clear scene with the car and the corresponding stereo depth map. Column (c) and (d) show the radar heatmap as 3D point cloud and 2D front-view projection. Column (e) shows the output from HawkEye, while Columns (f) and (g) show the output from L1 and Nearest Neighbor baselines. The scalebar shows the absolute depth metric in the depth map.

along a ray that projects at that location:

$$x_{2D}(\phi, \theta) = \arg\max_{r} \; x_{3D}(\phi, \theta, r). \tag{1}$$

Simply choosing the depth corresponding to the largest value is unstable. Thus, we choose the m largest values and create $m = 8$ channels of $2D$ feature maps, ordered from highest to lowest power. These 2D feature maps are concatenated with the features maps at the 6^{th} layer in the decoder, so that the high-resolution depth information from the radar heatmap is directly extracted and passed to the output in order to retain the high-frequency details in depth. Note that, this projection is done in a spherical coordinate frame, while our output is in the camera coordinate frame. As the field of view of the camera is not very large, the two images are still reasonably well aligned. This is a non-differentiable operation and is only done with the input.

Discriminator Architecture: The discriminator takes two inputs: the 3D heatmap x and a 2D depth map that either was the ground-truth y or was generated $G(x)$. It outputs the probability of the input being real. Typically, the input and output to the generator are of the same type (both 2D or both 3D). However, in our case input is 3D and output is 2D. Thus, we adopt a two-stream architecture that uses two separate networks to map x and y to 1D feature vectors, and then fuses them to classify real vs generated samples, as shown in Fig. 5. Heatmap x is processed through a 3D CNN with the same architecture as used in the generator but with different weights, while the depth map y or $G(x)$ is passed through an 8 layer CNN. Both networks produce 512 dimensional representation that are concatenated and passed to 2 fully connected layers to output the final classification.

Loss Function: The output of the discriminator D and generator G are used to calculate the vanilla GAN loss function $\mathcal{L}(G)$ [31]. As with past work [36, 38], we also include $\mathcal{L}_1$ loss (between the ground truth and the prediction), and a perceptual loss term $\mathcal{L}_p$ [38, 26] (on activations of a pre-trained neural network, VGG [56] in our case, on y and $G(x)$). During training, D and G are optimized to minimize the $\mathcal{L}_H(G)$ loss as below:

$$\mathcal{L}_1(G) = \mathbf{E}\|y - G(x)\|_1 \tag{2}$$

$$\mathcal{L}_p(G) = \mathbf{E}\|VGG(y) - VGG(G(x))\|_1 \tag{3}$$

$$\mathcal{L}_H(G) = \mathcal{L}(G) + \lambda_1\mathcal{L}_1 + \lambda_p\mathcal{L}_p \tag{4}$$

While $\mathcal{L}_1$ losses aren't effective for pixel prediction, our outputs are depth values and thus $\mathcal{L}_1$ makes sense. We use the feature space of a VGG network [56] to compute the perceptual loss. We replicate the depth image into a 3 channel image and feed it into a pre-trained VGG network. HawkEye employs a combination of three losses (Eq. 4), λ_1 and λ_p are hand-tuned relative weights of the loss functions. Using this loss function enables HawkEye to accurately capture both the low and high frequency components in the image. This results in perceptually interpretable high-resolution images that faithfully represent the scene.

5. HawkEye's Data Synthesizer

Collecting real-world mmWave data using our custom-built mmWave module is very time-consuming. Hence, trainig with real data would take a prohibitively long time. To address this, we build a synthesizer to generate paired 3D mmWave heatmaps and 2D depth maps of cars from

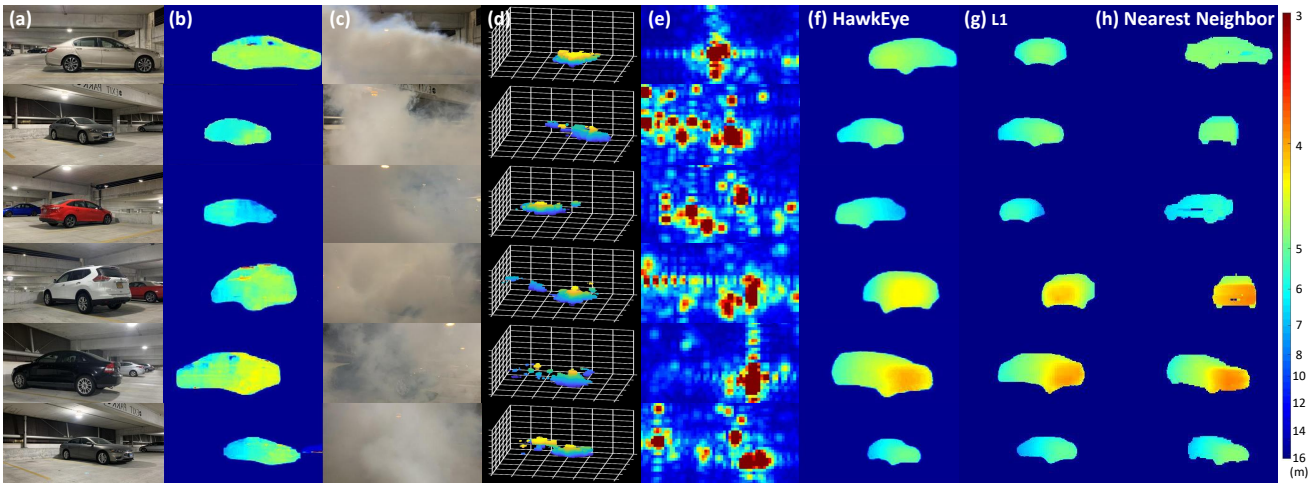


Figure 7: HawkEye's performance with fog in scene. Column (a) and (b) show the original scene and corresponding stereo depth map. Column (c) shows the scene filled with fog. Column (d) and (e) show the radar heatmap in the fog scene as 3D point cloud and 2D front-view projection. Column (f) shows the output from HawkEye, while Column (g) and (h) show the output from L1 and Nearest Neighbor baselines. The scalebar shows the absolute depth metric in the depth map.

3D CAD models. Our synthesizer is designed to create 3D point reflector models of cars and then simulate mmWave radar signals using ray tracing. It takes into account multipath reflections as well as specularity based on reflection angles to generate realistic mmWave 3D heatmaps. Simulation has 3 stages:

(1) *Scene Generation*: We first generate scenes of cars based on two types of datasets: 3D CAD models for autonomous driving [20] and Cityscapes [13], a street view video recordings dataset. The 3D CAD models provide us with precise 3D meshes of a wide variety of vehicles, while the street view photos offer references for car placement in the camera frame. We apply Mask R-CNN [33] on the street views to detect objects of interest.

(2) *Ray Tracing*: Here we model the mmWave reflectors in the scene. First, we remove occluded bodies through spherical projection. Then, we model the remaining parts as clusters of point reflectors, where the number of points represents the size of the radar cross section. We classify the specularity of each cluster as scattering corners or mostly specular surfaces by referring to the known car outline. Finally, we perform standard ray tracing [16] on the point reflectors with their specularity taken into account.

(3) *mmWave Heatmap and Ground-truth Generation*: We simulate the received signal based on the point reflector model with background noise introduced. We add thermal noise and phase noise to the mmWave signals. Additionally, to avoid the nontrivial extrinsic calibration for the field point and point of view between the mmWave and stereo camera modules in our experimental setup, we import the same displacement into our synthesizer to make predictions at the stereo camera view point, and to accurately train and test HawkEye's GAN architecture. Similarly, our model can be re-trained to make predictions from any other viewpoint as well (with appropriately modified skip connections). By applying mmWave processing as described in Section 3, we get the 3D mmWave heatmap. The ground-truth 2D depth map is generated to match the stereo camera frame.

6. Experiments

6.1. Dataset and Implementation

Dataset: Since there is no publicly available mmWave radar dataset, we collect our own dataset using a custom-built mmWave imaging platform. We emulate a 2D antenna array with a 60 GHz radio and transmit a standard radar waveform to capture 3D mmWave heatmaps. To capture the corresponding high resolution 2D depth maps for ground truth, we build a custom wide baseline stereo camera system. We mount an iPhone camera on a linear slider with sub-mm accuracy to capture multiple images of the scene, and apply a standard stereo image processing algorithm [35] to extract 2D depth maps. The linear slider setup is stable, which allows us to calibrate once and apply the same rectification to all experiments. In the ground truth, we filter out pixels that do not belong to the vehicles of interest using labeled object masks generated from Mask R-CNN [33]. Supplementary material includes a more detailed description of our data collection platform.

We imaged 327 scenes of cars in 3 types of backgrounds: indoor parking garage, outdoor lot, and outdoor house drive-through. The dataset includes 9 categories of cars spanning 60 different models: 2 Sub-compact, 12 Compact, 16 Mid-sized, 7 Full-sized, 5 Sports, 11 SUVs, 1 Jeep, 2 Vans, and 4 Trucks. We tested all 360° orientations of the car with respect to the radar. The distance from the radar

to the car is between 3.3 to 11.9 meters, with a mean of 6.2 m and a standard deviation of 1.66 m. We then created a dataset of paired 3D mmWave heatmaps, RGB camera images, and stereo camera depth maps. In addition to real data, we also have 4000 synthesized scenes generated from HawkEye's data synthesizor for 120 car models.

Controlled Experiments in Fog/Rain: Out of the 327 real scenes we imaged, there are 101 experiments in fog and rain to test HawkEye's performance in poor visibility conditions where today's optical sensors fail. Due to practical limitations such as the risk of water damage to our setup, we conduct controlled experiments where we emulate real fog and rain. We use a fog machine along with a high-density water-based fog fluid to emulate severe and realistic fog conditions, similar to previous studies [29, 30]. We emulate rain using a water hose in a confined region around the object of interest (the car).

Training: HawkEye's GAN is trained in two stages. In the first stage, we train for 170 epochs using a synthesized dataset of 3000 images with batch size 4. In the second stage, we fine-tune the model for 60 additional epochs with 100 real mmWave images captured in clear weather. It is important to note that HawkEye's GAN model is never trained on examples collected in fog or rain. The training takes 12 hours on an Nvidia Titan RTX GPU. We test HawkEye's performance on 1000 synthesized images, and the remaining 227 real images including the fog and rain experiments. For testing, we follow standard k-fold cross-validation with $k = 5$ to test all 327 scenes while ensuring examples in the test dataset are not used during training.

Baselines: We compare HawkEye to:

(i) *mmWave Radar:* We compare against raw mmWave radar heatmaps to evaluate HawkEye's improvement over the low resolution and artifact-ridden radar images.

(ii) L_1 *Based Loss:* To determine the utility of the GAN and discriminator in HawkEye, we compare against an identical neural network trained only with the L_1 based loss function, $\mathcal{L} = \mathcal{L}_1 + \lambda_p \mathcal{L}_p$, as defined in Eq. (1) and (2).

(iii) *Nearest Neighbor:* One could argue that our method overfits and simply memorizes sample points from the training dataset. To understand this, we compare against a Nearest Neighbor scheme, which retrieves samples in the input feature space of 3D radar heatmaps with the minimum Euclidean distance.

6.2. Qualitative Results

We show HawkEye's performance in clear weather and fog in Fig. 6 and Fig. 7 respectively.[4] In both visibility conditions, HawkEye accurately reconstructs the shape and size of the car in the scene, and captures key defining features such as its wheels and orientation comparably better than the other baselines. HawkEye can also accurately determine the distance to the car in 3D space, as can be seen from the intensity in the depth maps. HawkEye's ability to accurately image in fog and rain[5], despite not being trained with such examples, demonstrates that our model can generalize well in different weather conditions due to the favorable propagation characteristics of mmWave signals. Further, note that although HawkEye is trained primarily on synthesized data, it could generalize well to real scenes with different backgrounds and visibility conditions with only a small amount of fine-tuning. Hence, the simulator faithfully emulates the real mmWave heatmaps.

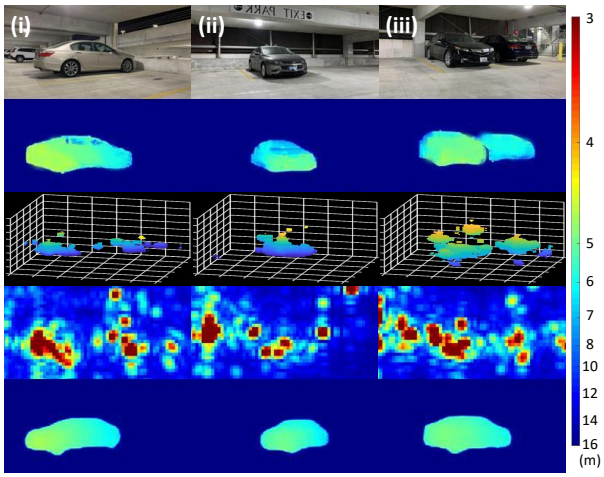


Figure 8: Examples where HawkEye fails. The first and second rows show the original scene and corresponding stereo depth map. Third and fourth row show the radar heatmap as point cloud and front view projection. The fifth row shows HawkEye's output.

Failure Examples: Figure 8 shows some typical failure cases for HawkEye. (i) and (ii) are from the fog experiments. In (i), although HawkEye estimates the correct bounding box, it misjudges the front and back of the car. In (ii), although HawkEye successfully detects the corner of the car, due to both strong fictitious reflections and specularity in the heatmap, it incorrectly estimates the orientation of the car. Lastly, a current limitation of our system is that its performance deteriorates when the scene has multiple cars (Fig. 8(iii)). To address this, a potential future direction is to adopt a *Region Proposal Network*, similar to [64], where HawkEye can first isolate the reflections from the cars in the scene, and then reconstruct each car individually.

6.3. Quantitative Results

Metrics: We evaluate on range, size (length, width, height), and orientation of the car, as they represent the contextual information of the car in the scene. We define the distance to

[4] We show additional qualitative results for the synthetic test dataset and for experiments in rain in the supplementary material.

[5] See supplementary material for results in rain.

Experiment	System	Error in Ranging	Error in Length	Error in Width	Error in Height	Error in Orientation	% Fictitious Reflections	% Car Surface Missed
Clean Air	HawkEye	**30 cm**	**47 cm**	**29 cm**	**9 cm**	**27°**	**1.5%**	**12.9%**
	mmWave	53 cm	179 cm	89 cm	45 cm	64°	15.6%	30.5%
	L_1 Based Loss	40 cm	97 cm	76 cm	13 cm	37°	2.5%	13.1%
	Nearest Neighbor	90 cm	114 cm	70 cm	17 cm	68°	3.5%	16.0%
Fog	HawkEye	**50 cm**	**83 cm**	**44 cm**	**11 cm**	**29°**	**2.5%**	15.4%
	mmWave	67 cm	222 cm	99 cm	53 cm	72°	20.9%	31.9%
	L_1 Based Loss	60 cm	108 cm	80 cm	12 cm	38°	3.5%	**13.8%**
	Nearest Neighbor	121 cm	117 cm	76 cm	18 cm	45°	3.6%	22.3%
Synthesized Data	HawkEye	23 cm	**64 cm**	**37 cm**	**8 cm**	**30°**	**1.3%**	10.2%
	mmWave	29 cm	182 cm	77 cm	31 cm	62°	10.8%	19.2%
	L_1 Based Loss	**20 cm**	113 cm	73 cm	14 cm	47°	3.4%	**9.3%**
	Nearest Neighbor	81 cm	81 cm	57 cm	13 cm	64°	5.2%	17.5%

Table 1: Quantitative Results. See text for more details.

the closest corner of the car as the range, and orientation as the angle between the longer edge of the car and the 0° azimuth of the mmWave heatmap. We also evaluate accuracy in shape prediction by comparing *(a) % of Car's Surface Missed* (false negatives) and *(b) % of Fictitious Reflections* (false positives) in HawkEye's output along the front view of the scene. Note that (a) is indicative of the specularity effects whereas (b) is indicative of artifacts such as multipath and ambient reflections in the image. Supplementary material includes a more detailed description of how we extract our quantitative metrics.

Results: Table 1 shows median errors comparing HawkEye to the baseline schemes. These results are extracted from 168 scenes in clean air, 59 scenes in fog, and 510 synthesized scenes. We summarize HawkEye's performance compared to each baseline below.

(i) *mmWave radar:* HawkEye achieves an improvement in ranging accuracy of $1.35\times$ in fog and $2\times$ in clear weather. Although mmWave radars can achieve high ranging resolution, the artifacts in the radar heatmaps lead to high ranging error. The skip connections in HawkEye's design allow for the direct transfer of the high ranging resolution from the mmWave radar input to HawkEye's output, while additionally HawkEye's GAN model corrects for the sinc artifact to achieve lower median ranging error. However, note that HawkEye's gains over mmWave radar become more apparent for the other metrics, spanning from $2\times$ to $12\times$ gain for percentage of fictitious reflectors. This is because the other metrics are a lot more sensitive to the specularity and multipath artifacts, and HawkEye can significantly improve these metrics by correcting for these noise sources.

(ii) L_1 *based Loss:* The L_1 loss baseline achieves good performance in terms of ranging error compared to HawkEye. This is expected since optimizing for L_1 loss over 2D depth maps would directly optimize for ranging error. However, L_1 loss cannot capture the high frequency components of the output shape, resulting in blurring of boundaries. As a result, the errors in estimated size, orientation and fictitious reflectors are high for L_1 loss, with HawkEye achieving approximately $2\times$ performance gains across these metrics. These results demonstrate the importance of the GAN architecture in HawkEye.

(iii) *Nearest Neighbor:* HawkEye outperforms the Nearest Neighbor baseline, achieving an improvement of $1.3\times$ to $3\times$ in clear weather, and $1.4\times$ to $2.4\times$ in fog across various metrics. This demonstrates that our model is not overfitting and can generalize well to new data points in the test set.

For the synthesized dataset, the performance trends are similar. The above results show that HawkEye can faithfully reconstruct an accurate and high resolution image of the car in the scene in both clear weather and in low visibility conditions. One should note that HawkEye's performance in fog degrades slightly compared to clear weather. This can be attributed to the poor propagation characteristics of 60 GHz RF signals in the presence of water particles in fog. It is worth noting that due to FCC regulations, we are constrained to build our experimental setup at the 60 GHz unlicensed spectrum, which suffers from higher attenuation from water particles compared to other frequencies in the mmWave band. We believe that implementing HawkEye with commercial grade mmWave radars built at the 77 GHz frequency band, which is allocated specifically for automotive radar applications, would resolve the performance degradation observed here.

7. Conclusion

In this paper, we show that HawkEye is a promising approach for achieving high resolution imaging with mmWave wireless systems, through the novel design of neural network architectures for processing mmWave data. We evaluate HawkEye in low visibility conditions such as heavy fog and show that it can significantly improve performance over mmWave radars today. While significant future work is required before HawkEye becomes a practical system that can be used on board self-driving cars, we have made huge advances toward this goal.

References

[1] F. Adib, C. Hsu, H. Mao, D. Katabi, and F. Durand. Capturing the human figure through a wall. *ACM Trans. Graph.*, 34(6), Oct. 2015.

[2] R. Appleby and R. N. Anderton. Millimeter-wave and submillimeter-wave imaging for security and surveillance. *Proceedings of the IEEE*, 95(8):1683–1690, Aug. 2007.

[3] K. Armanious, S. Abdulatif, F. Aziz, U. Schneider, and B. Yang. An adversarial super-resolution remedy for radar design trade-offs. *2019 27th European Signal Processing Conference (EUSIPCO)*, Sept. 2019.

[4] 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*, 39(12):2481–2495, Dec. 2017.

[5] K. Beier and H. Gemperlein. Simulation of infrared detection range at fog conditions for enhanced vision systems in civil aviation. *Aerospace Science and Technology*, 8(1):63 – 71, 2004.

[6] Bloomberg. Autonomous-car tech investment skyrockets on softbank deals, *Press Release, 2018*.

[7] Bloomberg. Uber has spent more than $1 billion on driverless cars, *Press Release, 2019*.

[8] Bloomberg Businessweek. Self-driving cars can handle neither rain nor sleet nor snow, *Press Release*, 2018.

[9] Bosch. Adaptive cruise control. https://www.bosch-mobility-solutions.com/en/, 2020.

[10] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, July 2017.

[11] C. C. Chen. Attenuation of electromagnetic radiation by haze, fog, clouds, and rain. Technical report, RAND Corp., 1975.

[12] L. Chen, Y. He, J. Chen, Q. Li, and Q. Zou. Transforming a 3-d lidar point cloud into a 2-d dense depth map through a parameter self-adaptive framework. *IEEE Transactions on Intelligent Transportation Systems*, 18(1):165–176, Jan. 2017.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, June 2016.

[14] L. Csurgai-Horvth and J. Bit. Fog attenuation on v band terrestrial radio and a low-cost measurement setup. In *2010 Future Network Mobile Summit*, pages 1–9, June 2010.

[15] A. Danzer, T. Griebel, M. Bach, and K. Dietmayer. 2d car detection in radar data with pointnets. pages 61–66, Oct. 2019.

[16] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, M. Barbiroli, M. Zoli, L. Tian, X. Yin, D. A. Dupleich, R. Mller, C. Schneider, and R. S. Thom. Ray-tracing-based mm-wave beamforming assessment. *IEEE Access*, 2:1314–1325, 2014.

[17] M. Dimitrievski, P. Veelaert, and W. Philips. Semantically aware multilateral filter for depth upsampling in automotive lidar point clouds. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1058–1063, June 2017.

[18] L. Ding and G. Sharma. Fusing structure from motion and lidar for dense accurate depth map estimation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1283–1287, Mar. 2017.

[19] 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*, page 414415, 2018.

[20] S. Fidler, S. Dickinson, and R. Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS12, page 611619, 2012.

[21] Forbes. How autonomous vehicles will navigate bad weather remains foggy, *Press Release*, 2016.

[22] Forbes. The future with level 5 autonomous cars, *Press Release*, 2019.

[23] Forbes. Weather creates challenges for next generation of vehicles, *Press Release, 2019*.

[24] Ford. Adaptive cruise control. https://www.ford.com/technology/driver-assist-technology/adaptive-cruise-control/, 2020.

[25] K. Garcia, M. Yan, and A. Purkovic. Robust traffic and intersection monitoring using millimeter wave sensors. Technical report, Texas Instruments, 2018.

[26] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, June 2016.

[27] M. T. Ghasr, M. J. Horst, M. R. Dvorsky, and R. Zoughi. Wideband microwave camera for real-time 3-d imaging. *IEEE Transactions on Antennas and Propagation*, 65(1):258–268, Jan. 2017.

[28] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *2009 IEEE International Conference on Computer Vision (ICCV)*, pages 349–356, Sept. 2009.

[29] Y. Golovachev, A. Etinger, G. Pinhasi, and Y. Pinhasi. Millimeter wave high resolution radar accuracy in fog conditions–theory and experimental verification. *Sensors*, 18(7):2148, 2018.

[30] Y. Golovachev, A. Etinger, G. A. Pinhasi, and Y. Pinhasi. Propagation properties of sub-millimeter waves in foggy conditions. *Journal of Applied Physics*, 125(15):151612, 2019.

[31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014.

[32] A. Guei and M. Akhloufi. Deep learning enhancement of infrared face images using generative adversarial networks. *Applied Optics*, 57(18):D98–D107, 2018.

[33] K. He, G. Gkioxari, P. Dollr, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, Oct. 2017.

[34] Y. He, L. Chen, and M. Li. Sparse depth map upsampling with rgb image and anisotropic diffusion tensor. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 205–210, June 2015.

[35] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, Feb. 2008.

[36] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, July 2017.

[37] The Wall Street Journal. Toyota investing $500 million in uber in driverless-car pact, *Press Release, 2018*.

[38] J. Justin, A. Alexandre, and F. Li. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[39] M. Laurenzis, F. Christnacher, E. Bacher, N. Metzger, S. Schertzer, and T. Scholz. New approaches of three-dimensional range-gated imaging in scattering environments. In *Electro-Optical Remote Sensing, Photonic Technologies, and Applications V*, volume 8186, pages 27 – 36, 2011.

[40] C. Ledig, L. Theis, F. Huszr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, July 2017.

[41] C. Li, M. Zaheer, Y. Zhang, B. Poczos, and R. Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018.

[42] Velodyne Lidar. https://velodynelidar.com/, 2020.

[43] J. S. Lu, P. Cabrol, D. Steinbach, and R. V. Pragada. Measurement and characterization of various outdoor 60 ghz diffracted and scattered paths. In *2013 IEEE Military Communications Conference*, pages 1238–1243, Nov. 2013.

[44] Mehdi M. and Simon O. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[45] B. Mamandipoor, G. Malysa, A. Arbabian, U. Madhow, and K. Noujeim. 60 ghz synthetic aperture radar for short-range imaging: Theory and experiments. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 553–558, Nov. 2014.

[46] Y. Meng, A. Qing, C. Lin, J. Zang, Y. Zhao, and C. Zhang. Passive millimeter wave imaging system based on helical scanning. *Scientific Reports*, 8(1):7852, May 2018.

[47] National Academies of Sciences, Engineering, and Medicine. *Airport Passenger Screening Using Millimeter Wave Machines: Compliance with Guidelines*. The National Academies Press, 2017.

[48] NVIDIA. The journey to zero accidents - nvidia drive. https://resources.nvidia.com/en-us-drive-overview, May 2019.

[49] C. Premebida, L. Garrote, A. Asvadi, A. P. Ribeiro, and U. Nunes. High-resolution lidar-based depth mapping using bilateral filter. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2469–2474, Nov. 2016.

[50] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*, page 234241, 2015.

[51] G. Satat, B. Heshmat, D. Raviv, and R. Raskar. All photons imaging through volumetric scattering. *Scientific Reports*, 6(1):33946, Sept. 2016.

[52] G. Satat, M. Tancik, and R. Raskar. Towards photography through realistic fog. In *2018 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10, May 2018.

[53] T. Savelyev, X. Zhuge, B. Yang, A. Yarovoy, L. Ligthart, M. Drozdov, and B. Levitas. Development of uwb microwave array radar for concealed weapon detection. In *11-th INTERNATIONAL RADAR SYMPOSIUM*, pages 1–4, June 2010.

[54] S. Shahramian, M. J. Holyoak, A. Singh, and Y. Baeyens. A fully integrated 384-element, 16-tile, w -band phased array with self-alignment and self-test. *IEEE Journal of Solid-State Circuits*, 54(9):2419–2434, Sept. 2019.

[55] D. M. Sheen, D. L. McMakin, and T. E. Hall. Near field imaging at microwave and millimeter wave frequencies. In *2007 IEEE/MTT-S International Microwave Symposium*, pages 1693–1696, June 2007.

[56] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[57] E. Smith and D. Meger. Improved adversarial systems for 3d object generation and reconstruction. *arXiv preprint arXiv:1707.09557*, 2017.

[58] Tesla, Inc. Autopilot. https://www.tesla.com/autopilot, 2020.

[59] The New York Times. 5 things that give self-driving cars headaches, *Press Release*, 2016.

[60] The New York Times. Honda to invest 2.75 billion usd in gm's cruise autonomous vehicle unit, *Press Release, 2018*.

[61] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. pages 1–6, May 2019.

[62] M. Zhao, T. Li, M. A. Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi. Through-wall human pose estimation using radio signals. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, June 2018.

[63] M. Zhao, Y. Liu, A. Raghu, H. Zhao, T. Li, A. Torralba, and D. Katabi. Through-wall human mesh recovery using radio signals. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10112–10121, Oct. 2019.

[64] M. Zhao, Y. Tian, H. Zhao, M. A. Alsheikh, T. Li, R. Hristov, Z. Kabelac, D. Katabi, and A. Torralba. Rf-based 3d skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM 18, page 267281, 2018.

[65] S. Zihir, O. D. Gurbuz, A. Kar-Roy, S. Raman, and G. M. Rebeiz. 60-ghz 64- and 256-elements wafer-scale phased-array transmitters using full-reticle and subreticle stitching techniques. *IEEE Transactions on Microwave Theory and Techniques*, 64(12):4701–4719, Dec. 2016.