Depth Estimation from Camera Image and mmWave Radar Point Cloud

Akash Deep Singh, Yunhao Ba, Ankur Sarker, Howard Zhang, Achuta Kadambi,
Stefano Soatto, Mani Srivastava
University of California, Los Angeles
{akashdeepsingh, yhba, ankursarker, hwdz15508, achuta, soatto, mbs}@ucla.edu

Alex Wong
Yale University
alex.wong@yale.edu

Abstract

We present a method for inferring dense depth from a camera image and a sparse noisy radar point cloud. We first describe the mechanics behind mmWave radar point cloud formation and the challenges that it poses, i.e. ambiguous elevation and noisy depth and azimuth components that yields incorrect positions when projected onto the image, and how existing works have overlooked these nuances in camera-radar fusion. Our approach is motivated by these mechanics, leading to the design of a network that maps each radar point to the possible surfaces that it may project onto in the image plane. Unlike existing works, we do not process the raw radar point cloud as an erroneous depth map, but query each raw point independently to associate it with likely pixels in the image – yielding a semi-dense radar depth map. To fuse radar depth with an image, we propose a gated fusion scheme that accounts for the confidence scores of the correspondence so that we selectively combine radar and camera embeddings to yield a dense depth map. We test our method on the NuScenes benchmark and show a 10.3% improvement in mean absolute error and a 9.1% improvement in root-mean-square error over the best method. Code: https://github.com/nesl/radar-camera-fusion-depth.

1. Introduction

Understanding the 3-dimensional (3D) structure of the scene surrounding us can support a variety of spatial tasks such as navigation [33] and manipulation [9]. To perform these tasks, an agent is generally equipped with multiple sensors, including optical i.e., RGB camera and range i.e., lidar, radar. The images from a camera are "dense" in that they provide an intensity value at each pixel. Yet, they are also sparse in that much of the image does not allow for the establishing of unique correspondence due to occlusions or the aperture problem to recover the 3D structure lost to the image formation process. On the other hand, range sensors are typically sparse in returns, but provide the 3D coordinates for a subset of points in the scene i.e., a point cloud. The goal then is to leverage the complementary properties of both sensor observations – an RGB image and a radar point cloud that is synchronized with the frame – to recover the dense 3D scene i.e., camera-radar depth estimation.

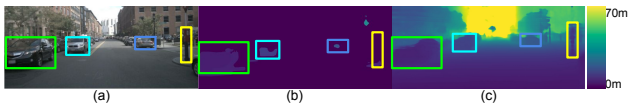


Figure 1. Depth estimation using a mmWave radar and a camera. (a) RGB image. (b) Semi-dense depth generated from associating the radar point cloud to probable image pixels. (c) Predicted depth. Boxes highlight mapping of radar points to objects in the scene.

While sensor platforms that pair lidar with camera have been of recent interest i.e., in autonomous vehicles, they are expensive in cost, heavy in payload, and have high energy and bandwidth consumption [36] – limiting their applications at the edge [37]. On the other hand, mmWave [20] radars are orders of magnitude cheaper, light weight, and power efficient. Over the last few years, developments in mmWave radars and antenna arrays [18] have significantly advanced the performance of these sensors. Radars are already ubiquitous in automotive vehicles as they enable services such as cruise control and collision warning [10]. Methods to perform 3D reconstruction with camera and radar are also synergistic with the joint communication and sensing (JCAS) paradigm in 6G cellular communication [1, 43, 57], where cellular base-stations will not only be the hub of communication, but also act as radars, to sense the environment.

The challenge, however, is that a mmWave radar is a point scatterer and only a very small subset of points (50

to 80 per frame [6]) in the scene, often noisy due to its large beam width, are reflected back into the radar's receiver. Compared to the returns of a lidar, this is 1000x more sparse. Additionally, most radars used in automotive vehicles either do not have enough antenna elements along the elevation axis or do not process the radar returns along the elevation axis. Hence, the elevation obtained from them is either too noisy or completely erroneous (see Sec. 3).

As a result, camera-radar depth estimation requires (i) mapping noisy radar points without elevation components to their 3D coordinates (and with calibrating their 2D image coordinates i.e., radar-to-camera correspondence) and (ii) fusing the associated sparse points with images to obtain the dense depth. Existing works have projected the radar points onto the image and "extended" the elevation or y-coordinate in the image space as a vertical line [28] or relied on multiple camera images to compute the optical-flow which in-turn has been used to learn the radar-to-pixel mapping [30]. These approaches overlook that radar returns have *noisy* depth, azimuth and *erroneous* elevation. They also assume access to multiple consecutive image and radar frames, so that they may use the extra points to densify radar returns in both the close (from past frames) and far (from future frames) regions. In the scenario of obtaining instantaneous depth for a given frame, the requirement of future frames makes it infeasible; if delays are permitted, then an order of hundreds of milliseconds in latency is incurred.

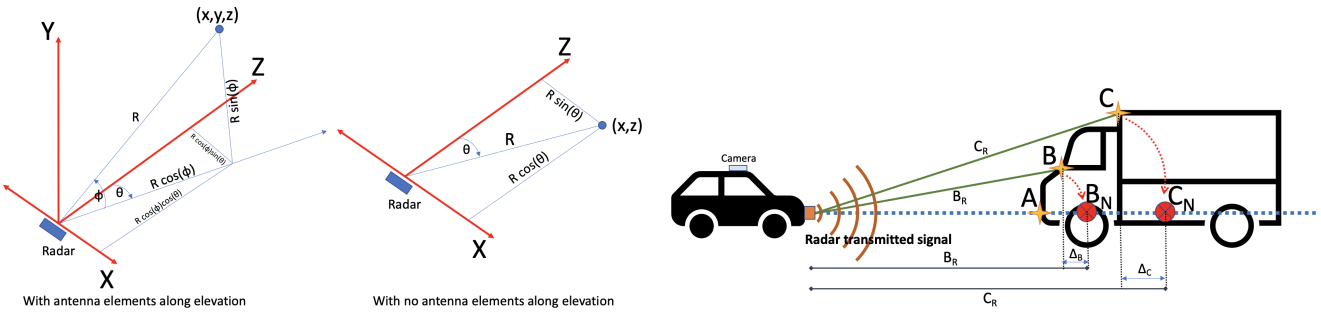
Instead, we propose to estimate depth from a single radar and image frame by first learning a one to many mapping of correspondence between each radar point and the probable surfaces in the image that it belongs to. Each radar point is corresponded to a region within the image (based on empirical error of projection due to noise) via a ROI alignment mechanism – yielding a semi-dense radar depth map. The information in the radar depth map is further modulated by a gated fusion mechanism to learn the error modes in the correspondence (due to possible noisy returns) and adaptively weight its contribution for image-radar fusion. The result of which is used to augment the image information and decoded to a dense depth map.

Our contributions are: (i) to the best of our knowledge, the first approach to learn radar to camera correspondence using a single radar scan and a single camera image for mapping arbitrary number of ambiguous and noisy radar points to the object surfaces in the image, (ii) a method to introduce confidence scores of the mapping for fusing radar and image modalities, and (iii) a learned gated fusion between radar and image to adaptively modulate the trade-off between the noisy radar depth and image information. (iv) We outperform the best method that uses multiple image and radar frames by 10.3% in mean absolute error (MAE) and 9.1% in root-mean-square error (RMSE) to achieve the state of the art on the NuScenes [6] benchmark, despite only using a single image and radar frame.

2. Related Work

Camera and lidar based depth estimation [4, 5, 12, 16, 19, 27, 32, 35, 40, 45–49, 52, 53, 55, 59] leverages an RGB image as guidance to densify a sparse lidar point cloud. Most of the works are focus on addressing the sparsity problem. For example, [7, 31, 38, 55] designed network blocks to effectively deal with the sparse inputs. [5] estimates the lidar sampling location and predicts the depth map more accurately without requiring high sampling rates. [25] used a cascade hourglass network, [16, 19, 55] used separate image and depth networks and fused their representations, and [17] proposed an upsampling layer and joint concatenation and convolution. [40] leveraged confidence maps to fuse predictions from different modalities, [35, 53, 58] used surface normals for guidance and [8, 34] use convolutional spatial propagation networks. [27, 46] proposed adaptive learning frameworks. Another line of work [45, 48, 52] focused on densifying the inputs through interpolation [48] or spatial pyramid pooling [45, 52]. However, lidars are expensive, have high energy consumption and are limited in real world applications. On the other hand, single image depth [2, 3, 11, 21, 22, 24, 44, 50, 51, 56] lacks scale in the absence of strong priors; whereas, mmWave radars are cheap to purchase and common in many sensor platform, and grounds predictions to metric scale. Adapting these methods to radar point clouds is nontrival since they assume point cloud sizes of ≈30k points that are aligned to the image; in contrast radar point clouds are on orders of 50 points with noisy azimuth and ambiguous elevation.

Camera and radar based depth estimation uses sparse mmWave radar point clouds and camera images [13, 26, 28–30]. Unlike camera-lidar depth estimation, it brings new challenges due to the sparsity and noise of the radar point clouds. [30] learn a mapping from radar data to image pixels using a radar-to-pixel association and then train a network to using a lidar point cloud is used predict dense depth. To deal with the sparsity, however, [30] reproject multiple radar sweeps into the current frame to increase the density of points and use multiple camera images (some from the future) to compute the optical and radar flow – something which is not practical in real world. [26] propose a two-stage encoder-decoder architecture to reduce the noise in radar point cloud, and like [30], also uses future frames. Similarly, [28] create a height-extended radar representation and then fuse it with camera images to generate dense depth. [13] fuses sparse point clouds as a weak supervision signal during training and uses it as an extra input to enhance the estimation robustness at inference time. However, these works either ignore the noise and error in radar points or use multiple (future) images and radar scans to obtain denser points with additional points in close and far



(a) mmWave radar point cloud geometry

(b) Elevation ambiguity in radar point clouds

Figure 2. Challenges with radar point clouds – an illustration of elevation ambiguity (y) and noise in azimuth (x) and depth (z) components. (a-left) Shows the geometry of how the mmWave radar point clouds are obtained in an ideal setting. (a-right) Shows the geometry of how the mmWave radar point clouds are obtained when the radar lacks information along the elevation axis. As a result, the radar assumes that all the points reflecting back into the receiver are reflecting from the plane perpendicular to the radar. This means, that the value of y obtained will always be 0 – which renders it useless for any task. Due to this assumption, The values of both x and z will also be noisy. $\Delta x = Rsin\theta(1 - cos\phi)$, $\Delta y = Rsin\phi$, $\Delta z = Rcos\theta(1 - cos\phi)$. (b) Shows a real world manifestation of this noise – modified from Fig. 4(a) of [30]. Due to the ambiguity in height, The points B and C have a difference in their depth while the point A is accurate since it lies in the plane perpendicular to the plane of the radar. Hence, a projection of radar point clouds on to the image plane using camera intrinsics and pose will yield erroneous points in the resulting sparse depth map – one of the most common representations of range modality.

regions. Unlike them, we only require a single image and radar scan to produce dense depth.

3. mmWave Radar Point Cloud Generation

mmWave radars, like other radars send an electromagnetic (EM) [39] wave through their transmitter. This wave hits the objects in the scene, reflected back and collected at the receiver. Unlike visible light, whose wavelength is in μm, mmWave radars suffer from the challenge of specular reflections, due to wavelengths larger than visible light. The lack of diffused reflections (which is the case of visible light) means that only the objects that reflect back into the receiver of the radar are captured. Hence, only a small portion of the scene is visible to the radar. The point clouds generated from a mmWave radar will be sparser than a camera image by several orders of magnitude.

To resolve the location of these reflections, radars use multiple receivers. However, popular mmWave radars used in autonomous driving (such as the one used in nuScenes [6] data collection), lack the ability to resolve height of objects in the scenes – a direct result of either not having antenna elements to capture elevation information or not having sufficient compute to process reflections along the elevation.

As shown in Fig. 2, since the radar has no way of knowing where a reflection is coming from along the elevation, it assumes that every reflection is coming from the plane perpendicular to the radar. This causes ambiguity in elevation (y) while also making the azimuth (x) and depth (z) noisy. In addition, the wider beam-width of mmWave radars also leads to some noise along these axes. As a result, it is not possible to directly project the radar point clouds onto the image plane using the pose and camera intrinsics and use them as a means of obtaining depth of the scene. Some of the previous works [13, 26, 28, 29] do not account for this, i.e. treating the incorrect projections as is, or perform post-processing operations such as extending each radar point along the y-axis of an image. Unlike them, we learn to map radar points to probable surfaces in the scene to recover denser radar point clouds. [30] tries to correct for the noise in radar by learning a 'depth-association' between the camera and the radar, however, they assume that the source of noise is occlusion and not the ambiguity in elevation as shown in Fig. 2.

4. Our Approach

Formulation. Our goal is to recover the 3D scene from a single RGB image $I \in \mathbb{R}^{3\times H\times W}$ and K points in a point cloud $\mathbf{z}$, where a point $z \in \mathbb{R}^3$, H and W are the height and width of the image – here K (akin to a batch dimension) may vary from point cloud to point cloud, which our method handles through our RadarNet (Sec. 4.1). We assume that the point clouds are captured by mmWave radars, which typically have incorrect elevation (y-) along with noisy azimuth (x-) and depth (z-) readings. We propose to learn a function that outputs the dense depth $\hat{d} \in \mathbb{R}_{+}^{H\times W}$ for every pixel in the image. Rather than directly learning a map from I and z to $\hat{d}$, we divide it into two sub-problems and solve them sequentially – (i) find correspondences between each point in the noisy radar point clouds and its probable projection onto the image plane to yield a semi-dense radar depth map and (ii) fuse information from the semi-dense radar map and the camera images to output $\hat{d}$.

Our approach is realized as two sequential deep neural networks: (i) RadarNet h_θ parameterized by θ takes an

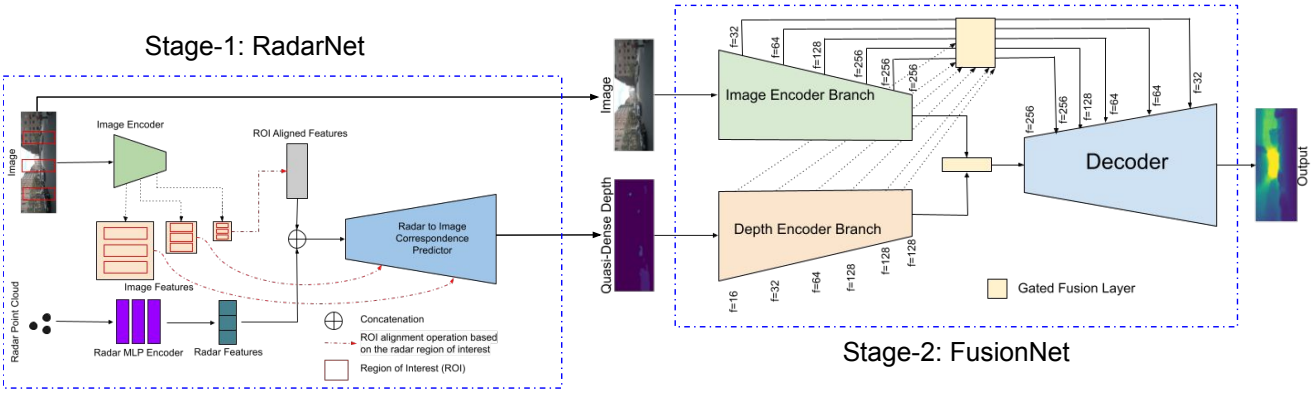


Figure 3. System Overview - Our two-stage architecture for estimating dense depth from a mmWave radar point cloud and a camera image.

RGB image I and a radar point z as input and outputs a confidence map $h_\theta(I, z) \in [0, 1]^{H \times W}$ for the probable surfaces that the point maps to in the image. Alternatively, for K points in the point cloud $\mathbf{z}$, $h_\theta(I, \mathbf{z})$ outputs K confidence maps from which we construct a semi-dense radar depth map by selecting the z-component of the radar point corresponding to the maximum response greater than a threshold $\tau = 0.5$ in the hypothesis $h_\theta(I, \mathbf{z})$ for each pixel to yield radar depth map $\hat{z} \in \mathbb{R}_+^{H \times W}$. This allows us to process point clouds with any arbitrary number of points – for a single point, we naturally default to selecting its z-component for any response greater than τ. (ii) FusionNet f_ω parameterized by ω further fuses together I, $\hat{z}$ and its confidence for each correspondence $\hat{h}_\theta(I, \mathbf{z})$ to yield the dense depth map $\hat{d} = f_\omega(I, \hat{z}, \hat{h}_\theta(I, \mathbf{z})) \in \mathbb{R}_+^{H \times W}$. Fig. 3 shows the system overview of our two stage approach.

4.1. Learning Radar to Image Correspondence

We assume that we are given a dataset with training samples comprised of an RGB image I, radar point cloud $\mathbf{z}$, ground truth lidar depth map $d_{gt} \in \mathbb{R}_+^{H \times W}$. RadarNet h_θ is comprised of two encoders, one standard ResNet18 backbone [15] with 32, 64, 128, 128, 128 filters in each of its layers, respectively, to process the image and a multi-layer perception (MLP) of 5 fully connected layers with 32, 64, 128, 128, 128 neurons, respectively, to encode the radar points. The latent of the point cloud is mean-pooled and reshaped to the size of the image latent, then together with skip connections from intermediate layers in the encoder, decoded into response maps or logits. We apply sigmoid activations to the logits to obtain the confidence scores $h_\theta(I, \mathbf{z})$.

To illustrate the challenge of radar to image correspondence, we note that there exists inherent ambiguities in determining radar to image correspondence since the point cloud lacks a viable elevation component. Also due to the noise in radar points, both depth and azimuth can vary between 10cm in the regions near the sensor and up to 40cm in the far regions [54]. Thus, unlike previous works [28] that "extend" the radar point along the elevation (which would yield incorrect correspondences) by copying its z-component along the vertical (y-) direction to create "radar lines", we propose to associate the radar points to the many probable surfaces in the image within a search range of $H \times w$ image crop centered on the position of the point.

ROIAlign for efficient inference. A naive approach to mapping a radar point to probable regions in the image is to simply score the entire image. However, this would require an $H \times W$ search space for each point where most of it will yield low confidence scores – likely regions will be localized to a $H \times w$ crop. Instead, one may observe that the K points in $\mathbf{z}$ maps to the same scene and thus we only need to perform a single forward pass on the image and K forward passes for $\mathbf{z}$. To accelerate the process of finding these correspondences between the radar points and the camera image, we propose to extract regions of interest (ROIs) in the feature maps corresponding to each $H \times w$ crop using a ROI alignment mechanism [14]. Each ROI is the region within which the true position of the radar point lies – anywhere along the vertical axis H and within some region along the horizontal axis w as shown in Sec. 3.

Hence, to process an image and its associated point cloud, we extract ROIs from the image features for each point and stack them along the batch dimension. Hence, for K points, we will also have K corresponding ROIs for each encoder scale, which will be passed to the decoder to yield K confidence maps. We predefine a $K \times H \times W$ volume of zeros and transfer the output K number of $H \times w$ confidence scores to their respective ROI locations in the full $H \times W$ image lattice to yield $h_\theta(I, \mathbf{z})$.

We formulate the radar to image correspondence problem as a binary classification of each pixel for a given radar point z, where high responses in $h_\theta(I, z)$ indicate probable surfaces for a given point. As a final step in the forward pass to yield the corresponded radar depth map $\hat{z}$, for each pixel $x \in \Omega \subset \mathbb{R}^2$ i.e., the image spatial domain, we choose the

maximum response over the K confidence maps $h_\theta(I, \mathbf{z})$:

$$\hat{z}(x) = \begin{cases} \mathbf{z}[\hat{k}], & \text{if} \quad h_\theta(I,\mathbf{z})(x)_{[\hat{k}]} > \tau \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $\hat{k} = \arg\max_k h_\theta(I, \mathbf{z})(x)_{[k]}$ and $\tau = 0.5$ a threshold.

Training RadarNet. we simplify the forward pass to just predicting the $H \times w$ confidence score maps i.e. without the need to choose the maximum response. For supervision, ROIs corresponding to the radar points are extracted from accumulated (by reprojecting from neighboring frames) depth maps d_{acc}. To construct the labels for binary classification, any pixel location in d_{acc} that is within 40cm of the depth (z-) component of the radar point is set to belong to the positive class i.e., a correspondence, otherwise the negative class. In practice, ground truth (lidar) d_{gt} can be sparse or semi-dense depending on the specification of the lidar so there is a lack of supervision signal in regions where there are no lidar returns; hence, we opt to use d_{acc}.

To address this, we assume that world surfaces are locally connected and piece-wise smooth and build a scaffolding [48] over the scene d_{acc} to approximate its dense structure. We then construct labels $y_{gt} \in \{0,1\}^{H\times w}$ from the scaffolding and minimize a binary cross entropy loss:

$$\begin{aligned} \ell_{BCE} = \frac{1}{|\Omega|} \sum_{x\in\Omega} & -(y_{gt}(x) \log y(x) \\ & + (1 - y_{gt}(x)) \log(1 - y(x))), \end{aligned} \tag{2}$$

where $\Omega \subset \mathbb{R}^2$ denotes the spatial image domain, $x \in \Omega$ a pixel coordinate, and $y = h_\theta(I, \mathbf{z})$ the hypothesis of radar to camera image correspondence. By training RadarNet to map radar points to regions in the image space, we are able to query RadarNet with arbitrary number of points to support the varied number of radar returns at each time frame to yield a semi-dense depth map that are several orders of magnitude denser than the radar point cloud.

4.2. Radar and Camera Image Fusion

Given the associated radar depth map $\hat{z}$ and its confidence map $\hat{h}(x) = \max_k h_\theta(I, \mathbf{z})(x)_{[k]}$, we propose to learn FusionNet f_ω to fuse $\hat{z} \in \mathbb{R}_+^{H\times W}$ and $\hat{h} \in [0,1]^{H\times W}$ with the RGB image I. FusionNet is comprised of two encoders with ResNet18 backbones, one with 32, 64, 128, 256, 256, 256 filters to encode the image $\phi(I) \in \mathbb{R}^M$ and the other with 16, 32, 64, 128, 128, 128 filters to encode the depth map concatenated with the confidence map $\psi([\hat{z}, \hat{h}]) \in \mathbb{R}^N$. The two branches are processed separately and later fused together via an adaptive weighting layer that learns the contribution of the depth encodings. This is because the radar points are inherently noisy and the depth map putative correspondences, and thus we use a learned gating mechanism to limit incorrect information flow from the depth branch. The re-weighted depth encodings are added to the image encodings and passed as skip connections to the decoder to yield the dense depth map $\hat{d} \in \mathbb{R}_+^{H\times W}$.

Gated Fusion. While $\hat{z}$ is denser than the measured radar returns, it is admittedly still on orders of magnitude sparser than an image; hence, there will be many "empty" regions in an encoding of $\hat{z}$ and thus typical naive concatenation of the image and depth encodings [12, 16, 19, 19, 32, 35, 40, 45, 48, 52, 53, 55] would result in convolving over many zero activations. To address this, we propose to augment the image features $\phi(I)$ with depth encodings by learning a set of weights $\alpha = \sigma(p^\top \psi([\hat{z}, \hat{h}])) \in [0,1]^M$ and projecting $\psi([\hat{z}, \hat{h}])$ to match the dimensionality of $\phi(I)$ via $\psi'(z) = q^\top \psi([\hat{z}, \hat{h}]) \in \mathbb{R}^N$, where p and q are trainable linear transformations and $\sigma(\cdot)$ the sigmoid function. The fusion step is given by $\alpha \cdot \psi'(z) + \phi(I)$ to produce the skip connection at each encoder scale and also the latent, which are fed to the decoder to yield $\hat{d} = f_\omega(I, \hat{z}, \hat{h}_\theta(I, \mathbf{z}))$ (see Fig. 3). Our gated fusion mechanism modulates the amount of depth information being passed to the decoder based on the training data and in effect learns the error modes of the radar depth map $\hat{z}$ and its confidence scores $\hat{h}$.

Training FusionNet. We assume access to the ground truth lidar depth d_{gt} and accumlated (by reprojection) depth d_{acc}. We minimize the difference between the predictions d, and d_{gt} and d_{acc} with an L_1 penalty:

$$\ell_{L_1} = \frac{1}{|\Omega_{gt}|} \sum_{x\in\Omega_{gt}} |d_{gt}(x) - \hat{d}(x)| + \tag{3}$$

$$\frac{\lambda}{|\Omega_{acc}|} \sum_{x\in\Omega_{acc}} |d_{acc}(x) - \hat{d}(x)|, \tag{4}$$

where $\Omega_{gt}, \Omega_{acc} \subset \Omega$ denotes the domains where ground truth (lidar) depth d_{gt} and accumulated depth d_{acc} have valid values, respectively; λ is chosen to be 1.

5. Implementation Details

Dataset. We use the nuScenes [6] outdoor driving dataset for our evaluation. The dataset contains 1000 scenes of 20s duration each. A car is fitted with sensors such as a lidar, mmWave radar, camera and IMU and is driven around Boston and Singapore to collect these scenes. Since each sensing modality captures the scene at a different frequency, [6] provides frames where the time-stamps of data from all sensors is very close to each other, called keyframes, which are annotated with object bounding boxes. The dataset contains around 40,000 keyframes (≈40 samples per scene). We use the nuScenes train-test split – 700 scenes for training, 150 for validation and 150 for testing.

Data Preprocessing. Following [13, 26, 28–30], we accumulate future and past lidar frames by projecting the lidar

point cloud at each time step to the frame of reference of the given frame – we use 161 frames in total (80 frames each from the future and past, and the given frame) to yield d_{acc}. Note: dynamic objects given by the bounding boxes are removed from the point clouds from each time step before projecting the points to the given frame. We perform scaffolding [48] on d_{acc} to obtain an interpolated depth map, and use it to create labels y_{gt}. We used d_{acc} and d_{gt} and interpolated depth map to supervise FusionNet. For RadarNet, we use y_{gt} for supervision. Note: We only use the accumulated lidar points for training; for evaluation, we use the lidar depth maps d_{gt} provided by [6].

RadarNet (Stage-1). We use ROIs of size $H = 900$ and $w = 288$ for the input image size of 900×1600. For constructing y_{gt}, any point in d_{acc} within 0.4m of the z-component of a given radar point is marked as a positive example. We set the weight of positive class to 2 and train using a batch size of 6. We used Adam [23] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize RadarNet with a learning rate of $2e^{-4}$ for 75 epochs. We use horizontal flip, saturation, brightness and contrast for data augmentations where each has a 50% probability of occurring. The values of brightness, contrast and saturation adjustment are random uniformly sampled from 0.8 to 1.2. Training takes ≈ 36 hours for 75 epochs on a NVIDIA RTX A5000 GPU.

FusionNet (Stage-2). We used Adam [23] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize our network with a learning schedule $1e^{-3}$ for 400 epochs, then reduced to $5e^{-4}$ for another 50 epochs, and finally reduced $1e^{-4}$ for 50 epochs (for further fine-tuning). The augmentations used during the training include horizontal flip, and brightness, saturation, and contrast adjustments, each with 50% probability of occurring. Like RadarNet, the values of brightness, contrast and saturation are random uniformly sampled from 0.8 to 1.2. We use a batch size of 16 with random crops of 448×448. Training takes ≈ 36 hours for 200 epochs on a NVIDIA RTX A5000.

6. Experiments and Results

Baselines. We compared our method against different methods [13, 26, 28, 30, 32, 41] in Table 2 using error metrics in Table 1. We downloaded the pre-trained models from the official repositories for [25, 28, 30] and tested them on the official nuScenes [6] test set. Results from [13, 26, 32, 41] were taken from their paper because code was unavailable or did not reproduce their numbers. We note that several baselines utilize either multiple images or multiple radar point clouds or both to estimate depth. For instance, RC-PDA [30] uses three camera images and five radar scans to compute "Flow" where the additional frames and scans include those from future timestamps. In real-world, one cannot expect to have access to information from the future, so this is not feasible. Additionally, they project future (for increasing density in far regions) and past (for increasing density in close regions) radar scans onto the current frame to densify the radar returns. RC-PDA with HG is a variant of [30] that uses an hourglass (HG) [25] network. DORN [28] combines 5 radar scans from 3 different radars.

Metric	units	Definition
MAE	*mm*	$\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\hat{d}(x) - d_{gt}(x)\rvert$
RMSE	*mm*	$(\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\hat{d}(x) - d_{gt}(x)\rvert^2)^{1/2}$

Table 1. Error metrics for evaluating the depth estimation benchmarks, where d_{gt} is the ground truth lidar depth map.

We also provide an ablation study in Table 2 to gauge the gain from RadarNet. Instead of using the semi-dense dense map concatenated with the confidence map, we simply train our FusionNet to directly estimate dense depth using raw radar points and a camera image. The results of this model as shown in Table 2 as Ours (No RadarNet). This also demonstrates the drawback of projecting the raw radar points onto the image plane and treating the sparse depth map (about 50 to 70 points per frame) as input – as customary in existing works [13, 26, 28, 32, 41].

Depth Considerations. According to the nuScenes [6] documentation, the range of the lidar sensor used is between 80 - 100 meters. However, the usable range is only up to 70 to 80 meters [42]. Hence, we test all models with working code between 0-50, 0-70 and 0-80 meters.

Quantitative Results. We compare our methods with the existing methods at 50, 70, and 80 meters depth range in Table 2. Compared to baseline RC-PDA [30], our method improves MAE by 22.3%, 37.6% and 41.3% and RMSE by 9.8%, 31.4% and 36.3% when the depth is being evaluated up to 50, 70 and 80 meters respectively. Compared to RC-PDA with HG [30], our method improves MAE by 25.3%, 40.5%, 43.8% and RMSE by 13.3%, 34.4%, 38.8%. Our method outperformed DORN [28] by 10.3%, 13%, 11.7% when compared based on MAE and evaluated up to 50, 70 and 80 meters respectively. Similarly, our method improves RMSE by 9.1%, 12.6%, and 11.8% for those ranges. Overall, our method outperformed the best baseline evaluated by 10.3% MAE and 9.1% RMSE. We attribute our success largely to RadarNet being able to correctly correspond radar points to the objects in the scene, which has limited existing methods [13, 26, 28, 30, 32, 41] that either directly used the erroneous points or perform adhoc post-processing i.e. vertical extension [28] on them.

Efficacy of RadarNet Ours (No RadarNet) method performs better than several baselines in Table 2. The input to this method is a sparse depth map generated by projecting the raw radar points onto the image plane. Although this depth map contains a somewhat accurate distribution of depths in the scene, the locations are completely erroneous (Sec. 3). The difference in performance of our method with

Max Eval Distance	Method	# Radar frames	# Images	MAE ↓	RMSE ↓
50m	RC-PDA [30]	5	3	2225.0	4156.5
	RC-PDA with HG [30]	5	3	2315.7	4321.6
	DORN [28]	5(x3)	1	1926.6	4124.8
	Ours (no RadarNet)	**1**	**1**	1942.5	3986.1
	Ours	**1**	**1**	**1727.7**	**3746.8**
70m	RC-PDA [30]	5	3	3326.1	6700.6
	RC-PDA with HG [30]	5	3	3485.6	7002.9
	DORN [28]	5(x3)	1	2380.6	5252.7
	Ours (no RadarNet)	**1**	**1**	2318.2	4825.0
	Ours	**1**	**1**	**2073.2**	**4590.7**
80m	RC-PDA [30]	5	3	3713.6	7692.8
	RC-PDA with HG [30]	5	3	3884.3	8008.6
	DORN [28]	5(x3)	1	2467.7	5554.3
	Lin [26]	3	1	2371.0	5623.0
	R4Dyn [13]	4	1	N/A	6434.0
	Sparse-to-dense [32]	3	1	2374.0	5628.0
	PnP [41]	3	1	2496.0	5578.0
	Ours (no RadarNet)	**1**	**1**	2441.0	5141.4
	Ours	**1**	**1**	**2179.3**	**4898.7**

Table 2. We compare our method to the pre-trained baselines that use multiple camera images and radar scans for radar-camera depth estimation. The authors in [13] do not provide MAE numbers. In DORN [28], the authors use 5 radar scans from 3 different radars.

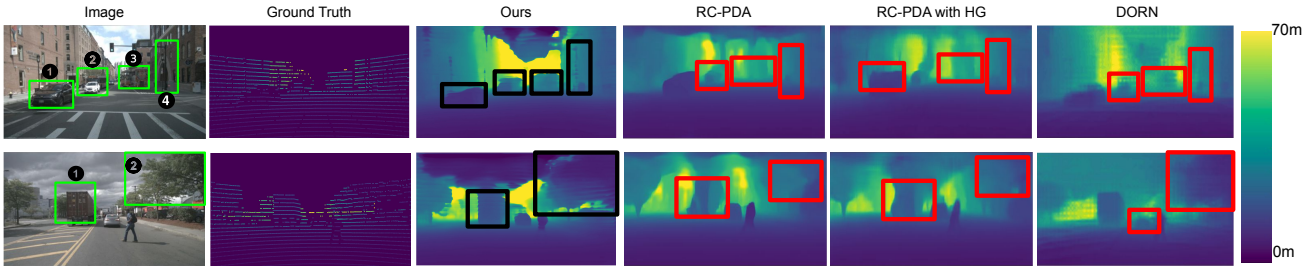


Figure 4. Qualitative results on nuScenes test set (best viewed in color at 5x). Column 1 shows the input image; column 2 shows the ground truth; columns 3 to 6 show the outputs of our method followed by those of the baselines. Bounding boxes highlight errors for comparison.

and without RadarNet demonstrates the advantage of our RadarNet model which not only helps in correcting the errors in the radar point cloud but also densifies the radar output into a semi-dense depth map. This study additionally confirms the detriment of input noisy radar points as a sparse depth map [28]. A qualitative comparison of our method with and without RadarNet is shown in Fig. 5.

Qualitative Results. In Fig. 4, we plot the dense-depth output of our method and the baselines on the nuScenes test dataset. We choose two representative scenes, one of a busy intersection (first row) and one of a pedestrian crossing a road while the traffic is moving during overcast weather (second row). We plot the ground truth next to the image on the right. The columns following the ground truth contain the output for our method as well as the baselines. We note that there is no supervision available in the top part of the scene (the top part generally contains the sky), so all the models hallucinate depth values for those pixels. For the scene shown in the first row, all models think the sky to be a continuation of the buildings along the road. For the scene shown in the second row, since the weather is overcast, all models think the sky to be a part of one of the nearby surfaces (due to the dark color). We use a black box to mark the qualitative advantages of our model when compared to the baselines (whose mistakes are marked with red boxes).

In the first row, our method is the only one which is able to pick up the bus in front (green box, number 3) that is trying to switch lanes. RC-PDA with HG outputs the wrong shape for the black car on the left side (number 1) of the scene. For the building on the left side of the scene, our method shows a smooth increase in depth whereas the other methods have abrupt changes. Box number 2 contains a white car in front of a bus. Only our method picks up this change in depth for the two vehicles while some baselines fail to capture it. Such is also the case for the traffic light post (4) that is captured by our method but not by others.

In the second row, the scene depicts a pedestrian who is trying to cross the street in the middle while a car is in the right lane and a truck (number 1) is in the turning lane. The weather is overcast and the clouds above are dark. There is

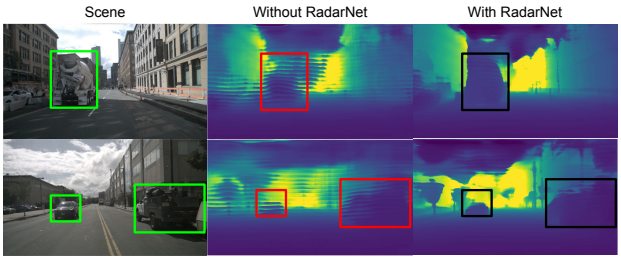


Figure 5. Qualitative comparison between our method with and without RadarNet. RadarNet's quasi-dense output enables FusionNet to learn shapes of objects effectively as shown in the bounding boxes above while also improving it's quantitative performance.

a tree branch (number 2) followed by a set of trees on the right top corner which is missed by both the RC-PDA baselines. The DORN baseline is able to pick the tree branch but missed all the trees behind it. It also misses a big portion of the car stopped in front. Both the RC-PDA methods map two or three different depth values to the back of the truck. Since the background immediately to the left of the truck is dark, both method exhibit over-smoothing on the truck due to similarity in intensity to the background.

In Fig. 5 we show the qualitative comparison between our method with and without RadarNet. While the model without RadarNet outperforms the baselines on quantitative metrics, it is unable to effectively learn shapes of the objects in the scene such as cars. This can be attributed to the fact that since metallic surfaces are better reflectors of radars, RadarNet is able to learn the shapes of these surfaces. These learned shapes act as a priors and enables FusionNet to learn scene geometry effectively.

7. Discussion

As radar sensors become more ubiquitous in our surroundings, it is essential to develop methods to integrate them with the existing inference pipelines such as the depth estimation frameworks. However, unlike other sensors, the long range, large field of view and high ambiguity of radar point clouds make it challenging for them to be used in the same manner as other sources of point clouds such as lidars.

In this paper, we focus on a single camera image and a single radar point cloud because with the advent of JCAS with 6G, radars are going to be a part of our infrastructure. Our cellular base-stations will sense their surroundings while also acting as the hub of communication. In such cases, since the sensor itself is not moving, there will be little to no benefit of combining consecutive frames (such is the case with RC-PDA that use multiple camera images) as to the sensor, the environment is stationary (ignoring small movements such as human beings, cars, etc.). Hence, it is imperative that we work towards creating methods that can estimate dense depth from a single vantage point.

The gated fusion mechanism of our FusionNet presents us with some advantages. Firstly, in case of very noisy radar points, the model can learn to rely more on the image branch by assigning a smaller weight to the depth branch. Secondly, in case the radar points are completely erroneous, the model can learn to solely rely on the image input. Thirdly, this mechanism ensures that in situations where the correspondence model is not very good, the performance of the depth completion stage does not significantly suffer since it can correct for error-prone values.

However, such a mechanism does have drawbacks. If the camera-radar setup is mis-calibrated or mis-aligned, the network may assume the radar values to be 'bad' and only rely on camera branch to predict depth. Additionally, it is well-known that softmax activations of a deep neural network are neither calibrated nor a substitute for uncertainty. Hence, there can be erroneous over-confident correspondences in the our RadarNet predictions. The goal of our gated fusion layer is to counteract such a case.

8. Conclusion

Unlike depth completion with images and lidar points, radar-camera depth completion introduces a series of challenges largely due to the assumptions made while obtaining radar point clouds. These assumptions introduce ambiguity when projecting the point clouds onto the image plane. In this paper, we addressed this challenge by proposing a two-step approach for obtaining dense depth via the fusion of radar point cloud and an image. The method is motivated by our understanding of radar point cloud generation mechanics and designed to correspond noisy and ambiguous radar points to image regions in a data-driven fashion. While we do not bar the case where correspondences are off i.e. over-prediction, our experiments show that the proposed method achieves better results compared with other methods, i.e. 10.3% in mean absolute error (MAE) and by 9.1% in root-mean-square error (RMSE) improvement, of obtaining dense depth via radar-camera fusion.

Acknowledgements: The research reported in this paper was sponsored in part by the Army Research Laboratory (ARL) under Cooperative Agreement (CA) W911NF2020158; the IoBT REIGN Collaborative Research Alliance funded by the ARL under CA W911NF1720196; Office of Naval Research (ONR) under N00014-22-1-2252; and, the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARL, DARPA, SRC, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] Mohammed Alloulah, Akash Deep Singh, and Maximilian Arnold. Self-supervised radio-visual representation learning for 6g sensing. In *ICC 2022-IEEE International Conference on Communications*, pages 1955–1961. IEEE, 2022. 1

[2] Yunhao Ba, Alex Gilbert, Franklin Wang, Jinfa Yang, Rui Chen, Yiqin Wang, Lei Yan, Boxin Shi, and Achuta Kadambi. Deep shape from polarization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 554–571. Springer, 2020. 2

[3] Yunhao Ba, Howard Zhang, Ethan Yang, Akira Suzuki, Arnold Pfahnl, Chethan Chinder Chandrappa, Celso M de Melo, Suya You, Stefano Soatto, Alex Wong, et al. Not just streaks: Towards ground truth for single image deraining. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 723–740. Springer, 2022. 2

[4] Zachary Berger, Parth Agrawal, Tian Yu Liu, Stefano Soatto, and Alex Wong. Stereoscopic universal perturbations across different architectures and datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2022. 2

[5] Alexander W Bergman, David B Lindell, and Gordon Wetzstein. Deep adaptive lidar: End-to-end optimization of sampling and depth completion at low sampling rates. In *2020 IEEE International Conference on Computational Photography (ICCP)*, 2020. 2

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 3, 5, 6

[7] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10023–10032, 2019. 2

[8] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10615–10622, 2020. 2

[9] Changhyun Choi and Henrik I Christensen. Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation. In *2010 IEEE International Conference on Robotics and Automation*, pages 4048–4055. IEEE, 2010. 1

[10] Angela H Eichelberger and Anne T McCartt. Toyota drivers' experiences with dynamic radar cruise control, pre-collision system, and lane-keeping assist. *Journal of safety research*, 56:67–73, 2016. 1

[11] Xiaohan Fei, Alex Wong, and Stefano Soatto. Geo-supervised visual depth prediction. *IEEE Robotics and Automation Letters*, 4(2):1661–1668, 2019. 2

[12] Chen Fu, Christoph Mertz, and John M Dolan. Lidar and monocular camera fusion: On-road depth completion for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019. 2, 5

[13] Stefano Gasperini, Patrick Koch, Vinzenz Dallabetta, Nassir Navab, Benjamin Busam, and Federico Tombari. R4dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes. In *2021 International Conference on 3D Vision (3DV)*, pages 751–760. IEEE, 2021. 2, 3, 5, 6, 7

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[16] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. *arXiv preprint arXiv:2103.00783*, 2021. 2, 5

[17] Zixuan Huang, Junming Fan, Shenggan Cheng, Shuai Yi, Xiaogang Wang, and Hongsheng Li. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *IEEE Transactions on Image Processing*, 29:3429–3441, 2019. 2

[18] Hideo IIZUKA, Toshiaki Watanabe, Kazuo Sato, and Kunitoshi NISHIKAWA. Millimeter-wave microstrip array antenna for automotive radars. *IEICE transactions on communications*, 86(9):2728–2738, 2003. 1

[19] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018. 2, 5

[20] Stephen L Johnston. Millimeter wave radar. *Dedham*, 1980. 1

[21] Achuta Kadambi, Vage Taamazyan, Boxin Shi, and Ramesh Raskar. Polarized 3d: High-quality depth sensing with polarization cues. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3370–3378, 2015. 2

[22] Achuta Kadambi, Vage Taamazyan, Boxin Shi, and Ramesh Raskar. Depth sensing using geometrically constrained polarization normals. *International Journal of Computer Vision*, 125:34–51, 2017. 2

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations (ICLR)*, 2015. 6

[24] Dong Lao, Alex Wong, and Stefano Soatto. Does monocular depth estimation provide better pre-training than classification for semantic segmentation? *arXiv preprint arXiv:2203.13987*, 2022. 2

[25] Ang Li, Zejian Yuan, Yonggen Ling, Wanchao Chi, Chong Zhang, et al. A multi-scale guided cascade hourglass network for depth completion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 32–40, 2020. 2, 6

[26] Juan-Ting Lin, Dengxin Dai, and Luc Van Gool. Depth estimation from monocular images and sparse radar data. In *2020 IEEE/RSJ International Conference on Intelligent*

Robots and Systems (IROS), pages 10233–10240. IEEE, 2020. 2, 3, 5, 6, 7

[27] Tian Yu Liu, Parth Agrawal, Allison Chen, Byung-Woo Hong, and Alex Wong. Monitored distillation for positive congruent depth completion. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 35–53. Springer, 2022. 2

[28] Chen-Chou Lo and Patrick Vandewalle. Depth estimation from monocular images and sparse radar using deep ordinal regression network. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3343–3347. IEEE, 2021. 2, 3, 4, 5, 6, 7

[29] Yunfei Long, Daniel Morris, Xiaoming Liu, Marcos Castro, Punarjay Chakravarty, and Praveen Narayanan. Full-velocity radar returns by radar-camera fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16198–16207, 2021. 2, 3, 5

[30] Yunfei Long, Daniel Morris, Xiaoming Liu, Marcos Castro, Punarjay Chakravarty, and Praveen Narayanan. Radar-camera pixel depth association for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12507–12516, 2021. 2, 3, 5, 6, 7

[31] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, 2019. 2

[32] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018. 2, 5, 6, 7

[33] Daniel Maier, Armin Hornung, and Maren Bennewitz. Real-time navigation in 3d environments based on depth camera data. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 692–697. IEEE, 2012. 1

[34] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In-So Kweon. Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision, ECCV 2020*. European Conference on Computer Vision, 2020. 2

[35] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3313–3322, 2019. 2, 5

[36] Thinal Raj, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim, and Aini Hussain. A survey on lidar scanning mechanisms. *Electronics*, 9(5):741, 2020. 1

[37] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016. 1

[38] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 2

[39] Lev A Vainshtein. Electromagnetic waves. *Moscow Izdatel Radio Sviaz*, 1988. 3

[40] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th international conference on machine vision applications (MVA)*, 2019. 2, 5

[41] Tsun-Hsuan Wang, Fu-En Wang, Juan-Ting Lin, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. Plug-and-play: Improve depth estimation via sparse data propagation. *arXiv preprint arXiv:1812.08350*, 2018. 6, 7

[42] Xinshuo Weng, Yunze Man, Jinhyung Park, Ye Yuan, Matthew O'Toole, and Kris M Kitani. All-in-one drive: A comprehensive perception dataset with high-density long-range point clouds. 2021. 6

[43] Thorsten Wild, Volker Braun, and Harish Viswanathan. Joint design of communication and sensing for beyond 5g and 6g systems. *IEEE Access*, 9:30845–30857, 2021. 1

[44] Alex Wong, Safa Cicek, and Stefano Soatto. Targeted adversarial perturbations for monocular depth prediction. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[45] Alex Wong, Safa Cicek, and Stefano Soatto. Learning topology from synthetic data for unsupervised depth completion. *IEEE Robotics and Automation Letters*, 6(2), 2021. 2, 5

[46] Alex Wong, Xiaohan Fei, Byung-Woo Hong, and Stefano Soatto. An adaptive framework for learning unsupervised depth completion. *IEEE Robotics and Automation Letters*, 6(2):3120–3127, 2021. 2

[47] Alex Wong, Xiaohan Fei, and Stefano Soatto. Voiced: Depth completion from inertial odometry and vision. *ArXiv, abs/1905.08616*, 2019. 2

[48] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5(2), 2020. 2, 5, 6

[49] Alex Wong, Mukund Mundhra, and Stefano Soatto. Stereopagnosia: Fooling stereo networks with adversarial perturbations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2879–2888, 2021. 2

[50] Alex Wong and Stefano Soatto. Bilateral cyclic constraint and adaptive regularization for learning a monocular depth prior. 2018. 2

[51] Alex Wong and Stefano Soatto. Bilateral cyclic constraint and adaptive regularization for unsupervised monocular depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5644–5653, 2019. 2

[52] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12747–12756, 2021. 2, 5

[53] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of*

the IEEE/CVF International Conference on Computer Vision, 2019. 2, 5

[54] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. Radarnet: Exploiting radar for robust perception of dynamic objects. In *European Conference on Computer Vision*, pages 496–512. Springer, 2020. 4

[55] Yanchao Yang, Alex Wong, and Stefano Soatto. Dense depth posterior (ddp) from single image and sparse range. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5

[56] Howard Zhang, Yunhao Ba, Ethan Yang, Varan Mehra, Blake Gella, Akira Suzuki, Arnold Pfahnl, Chethan Chinder Chandrappa, Alex Wong, and Achuta Kadambi. Weatherstream: Light transport automation of single image deweathering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[57] J Andrew Zhang, Fan Liu, Christos Masouros, Robert W Heath, Zhiyong Feng, Le Zheng, and Athina Petropulu. An overview of signal processing techniques for joint communication and radar sensing. *IEEE Journal of Selected Topics in Signal Processing*, 2021. 1

[58] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2018. 2

[59] Yiming Zhao, Lin Bai, Ziming Zhang, and Xinming Huang. A surface geometry model for lidar depth completion. *IEEE Robotics and Automation Letters*, 6(3), 2021. 2