Depth Estimation from Camera Image and mmWave Radar Point Cloud Supplementary Materials

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

Supplementary Contents

In this document, we provide supplementary materials for our main paper. This supplement is organized as follows:

- Section A provides additional evaluation for our method and the baselines at various depths.
- Section B shows intermediate and final outputs for our method.
- Section C provides in-depth explanation of the ROIAlign method used in RadarNet (Stage-1).
- Section D shows a sensitivty study on different thresholds (for radar point error bound and prediction confidence) used by RadarNet (Stage-1)
- Section E shows the comparison of our gated fusion method with other fusion methods for FusionNet (Stage-2).
- Section F dicusses some failure modes of the radar-to-camera correspondence stage (RadarNet, Stage-1).
- Section G discusses some failure modes of our method's second stage (FusionNet, Stage-2).
- Section H shows the total parameters in our method and run-time considerations.

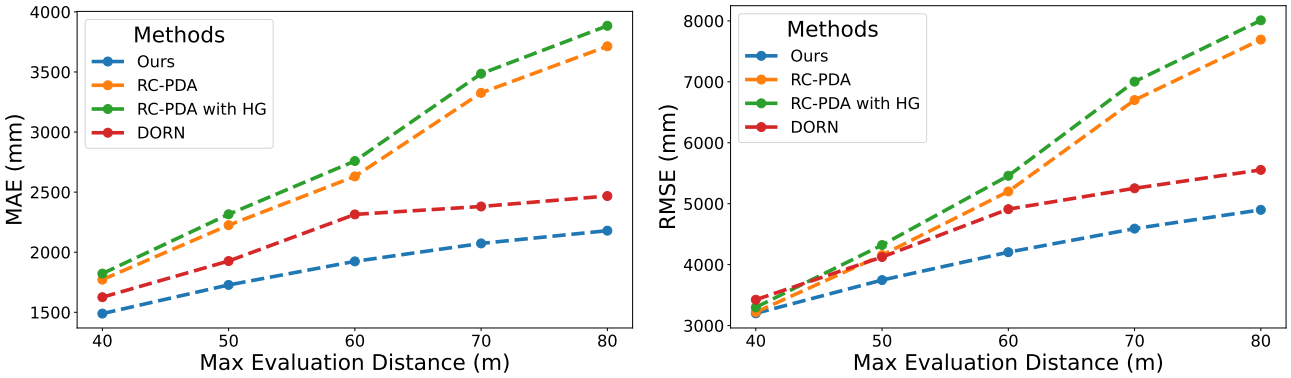

Figure A. Comparison of baselines with our method at various evaluation distances. Our method only requires a single image and radar scan as input, it consistently improves over existing baselines that use multiple radar scans and images. We attribute this to RadarNet's ability to map each point to a probable surface, unlike existing works that extends or duplicates the radar across the image vertically.

A. Evaluation at Multiple Depth Values

In this section with evaluate our model and the baselines and various depth levels. Fig. A compares the performance of all the models at 40m, 50m, 60m, 70m, and 80m using the MAE and RMSE error metrics (Table 1, main paper) respectively. Since the usable depth from the lidar used in nuScenes dataset is only up to 80m, we limit our evaluations to that value. In order to study the model performance at different increments of distances, we evaluate each model up to a particular max distance i.e. 40m, 50m, 60m, 70m, 80m. The change in performance indicates the error induced by the distance increment. The Fig. A shows that our model consistently performs better than all the baselines at both near and far distances. A notable result is that RC-PDA based methods experience a spike in error beyond 60m; whereas our model degrades linearly. We attribute to how we model errors in the radar point clouds, through which our model (RadarNet) correctly associates radar points to the image regions, and as a result, improves the downstream predictions of FusionNet.

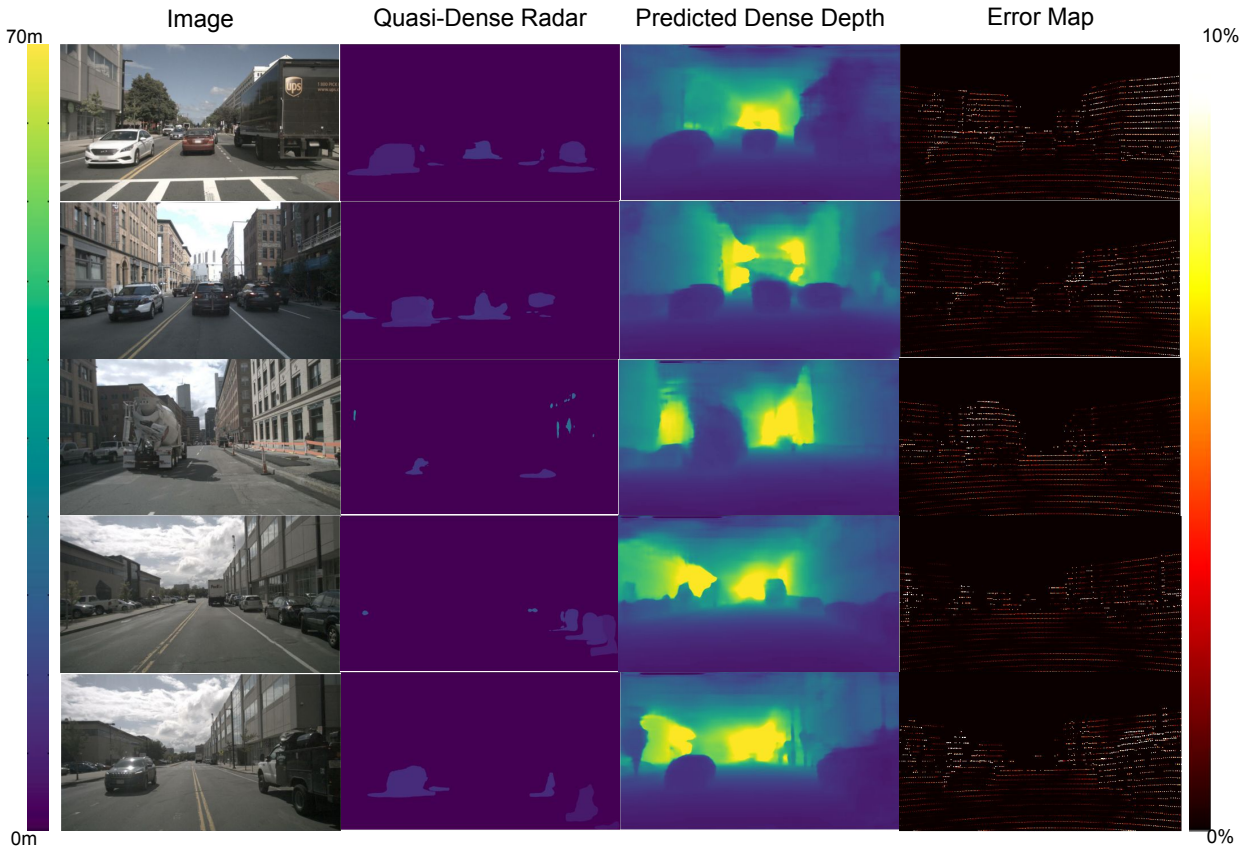

Figure B. Qualitative results for our method (best viewed in color at 5x). The figure shows (L-R) the camera image, semi- or quasi-dense output of the RadarNet, predicted dense depth (output of the FusionNet) and the error map. The range of depth is between 0-70m (as shown in the colorbar on the left) while the range of error is between 0% to 10% (as shown in the colorbar on the right).

B. Sample Output of Our Two-stage Pipeline

In this section, we show the intermediate and the final results of our method. We choose a diverse set of scenes showing driving alongside a large vehicle, cars driving towards the sensor, driving through street full of parked cars, and objects in front of the car which are in the shadows being cast by the buildings around. Metallic surfaces are a better reflector of RF than other surfaces (such as wood, cloth, trees, etc.) and hence, objects on a road such as cars, poles, traffic lights, and fire hydrants are more visible to RF sensors such as radars. In the output of our RadarNet, the semi-dense depth tends to favor such objects. Even though the raw-radar points clouds consist of 50-70 points, the RadarNet is able to densify these points around objects such as cars. The densified semi-depth map is then fused with the camera image to generate dense depth.

We selected the scenes in Fig. B to show how establishing radar to camera correspondence helps in detecting objects of interest and how it helps improve the performance of the FusionNet. As shown in Fig. 4 from the main paper, the baselines struggle to detect cars which are far from the sensors – something that our method is better at.

In the first row of Fig. B, there is a white car on the left hand side, a red card in the middle and a truck on the right. In the semi-dense depth map, one can see the corresponding radar depth for the cars and truck. Then the fusion-net combines this with the image to generate the dense depth as shown in the third column.

Similar, in the second row of Fig. B, there are three moving vehicles in the center of the image where RadarNet maps radar points to each one of them. Their shapes are propagated to the downstream FusionNet model and we observe very low errors in those regions according to the error map in the fourth column.

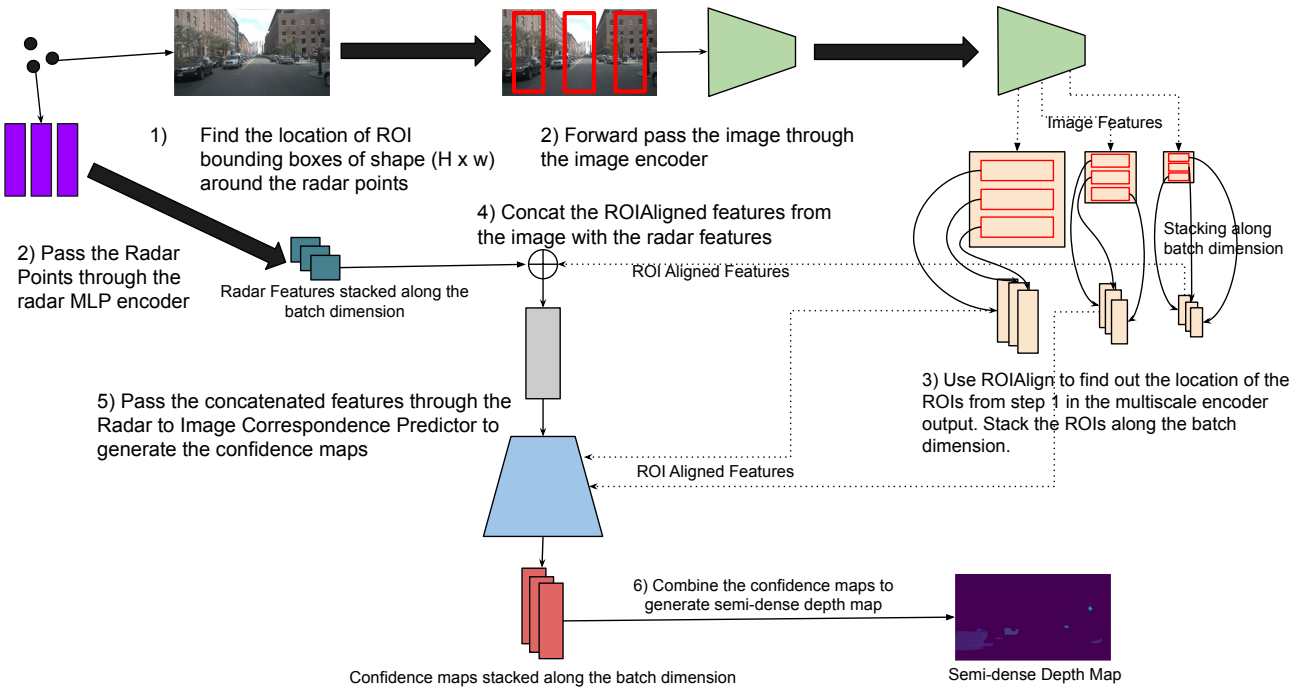

Figure C. RadarNet Overview (Stage 1).

In the fourth row of Fig. B, there are some vehicles parked (on the right side of the image). RadarNet predicts associates different radar points to each vehicle as seen by the differing colors in the visualized semi-dense depth map. Additionally, it is also is able to capture some points that are located in the far distances. In the corresponding dense-depth map, we are able to pick-up most of the vehicles in the scene (even when they are far from the sensors, thanks to RadarNet), something the baselines struggle to do as shown in Fig. 4 in the main paper. We observe a similar behavior for the last row of Fig. B. For the pickup truck parked on the right side, we observe two different clusters of points in the semi-dense depth map associated for the same truck – one group associated with the front and the other to back of the vehicle.

C. ROI Alignment in RadarNet

In this section, we describe the ROI Alignment process used to speed up the RadarNet (Stage 1) in detail. Using ROI Alignment speeds up the network by 17.2%. However, not only does using ROI Alignment improve run time, but also performance across all metrics and evaluated distances as shown in Tab. A. We attribute this largely to ROI extraction being an architectural inductive bias. Because we are dealing with a correspondence problem, the hypothesis space is dictated by the image space. Yet, as discussed in Sec. 3 and Fig. 2 from the main text, the radar point would correspond to a surface within a small window within the image. Hence, by extracting the ROI in the image for each point, we naturally reduce the search space for the correspondence problem,i.e. bias the predictions to certain regions in the image, and thus yielding performance improvements over a model that feeds in the entire image. A detailed pictorial summary of the proposed ROI Alignment process in RadarNet is shown in Fig. C with the following steps:

- **Step 1:** ROI Localization $\rightarrow$ We use the K radar points in the radar point-cloud to figure out the location of the regions of interest in the image. More specifically, for every radar point, the ROI is the region of shape $H \times w$ around the radar point.
- **Step 2:** Forward Passing $\rightarrow$ We forward pass the entire image through the image encoder and the radar points through the radar MLP [7] encoder.
- **Step 3:** ROI Alignment $\rightarrow$ Given the feature maps from the image encoder, we utilize the ROI Alignment operation to extract the local features for each of the radar point at each encoder scale and the final latent code. It should be noted that these local feature maps are stacked along the batch dimension after the ROI Alignment [8] operation.

- **Step 4:** Feature Concatenation → The latent feature learned from each radar point is concatenated with the corresponding local image features extracted in Step 3 for the decoder.
- **Step 5:** Confidence Map Generation → The decoder takes the concatenated features and the corresponding ROI aligned image features at each encoder scale to generate the confidence for each crop.
- **Step 6:** Combination to Generate Semi-Dense Depth Map → 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. We then use these confidence scores to generate the semi-dense depth map. Please refer to the Eq. 1 of the main paper.

Max Eval Distance	Method	MAE ↓	RMSE ↓
50m	Ours w/o ROI Alignment	1983.0	3980.5
	Ours	**1727.7**	**3746.8**
70m	Ours w/o ROI Alignment	2350.2	4834.3
	Ours	**2073.2**	**4590.7**
80m	Ours w/o ROI Alignment	2461.4	5141.2
	Ours	**2179.3**	**4898.7**

Table A. Ablation study on RadarNet with and without ROI Alignment. ROI Alignment not only improves run time by 17.2%, but also consistent performance across 50m, 70m, and 80m distances (by ≈300mm in MAE for 80m range). We conjecture that this is due to the inductive bias inherent in the ROI extraction procedure. By reducing the search space by more than 80%, we enable RadarNet to learn the correspondences more efficiently, yielding performance improvements.

D. Sensitivity Studies - RadarNet Thresholds for Error Bound and Confidence

In order to align radar point cloud with the image, RadarNet relies on some assumptions about the error inherent in the input radar points – namely due to noise, in the azimuth (x-) and ambiguity in the elevation (y-direction), stemming from insufficient antenna elements. Here, δ refers to the threshold to consider a pixel a probable correspondence for a radar point (where the absolute difference between its depth as measured by lidar and the depth component of a radar point is less than δ) and is set to handle the error induced by noisy azimuth and ambiguous elevation. Pixels within δ distance from a radar point are consider possible location for its projection onto the image. Choosing various values of δ controls over- and under-association of radar points to surfaces during the training phase of RadarNet. During inference, RadarNet outputs confidence scores across the image for each radar point, where each pixel with a confidence score greater than τ is considered a correspondence for the radar point and those with scores lower than τ are discarded. Choosing various values of τ controls over- and under-prediction during test time to generate semi-dense depth maps for the downstream FusionNet model. Empirically we choose 0.4m for δ and 0.5 for τ.

To demonstrate the sensitivity of the method to these two hyper-parameters, we vary them in both directions. As we vary the depth (δ) to 0.3m (more conservative threshold to consider a pixel as a correspondence to a radar point) and 0.5m (outside of our proposed noise/error bound), errors increase. This is because while $\delta = 0.3$m is more conservative in terms of supervision (smaller margin of error), it results in sparser or weaker supervision signal. On the other hand, $\delta = 0.5$m is larger than the allowable noise derived from radar point cloud geometry (Fig. 2a, main paper), yielding incorrect mapping of radar point to pixels when constructing supervision for training. We observe a similar phenomenon with τ. As τ is decreased to 0.3, RadarNet will over-predict the correspondences i.e. pixels close-by but belong to different surfaces are mapped to the same radar point; in contrast, increasing τ to 0.7 will cause RadarNet to under-predict, resulting fewer points in the semi-dense depth map. The negative effects of sparsity in depth modality are discussed in [19, 22, 26] and extend to radar depth completion as observed in the increase in error (Tab. B).

δ	MAE ↓	RMSE ↓	τ	MAE ↓	RMSE ↓
0.3m	2296.9	4725.9	0.3	2369.6	4865.9
0.4m	2073.2	4590.7	0.5	2073.2	4590.7
0.5m	2352.5	4820.5	0.7	2250.2	4654.3

Table B. Sensitivity study on RadarNet thresholds: δ (depth noise range during training) and τ (minimum confidence score of accepted correspondence during inference) at 70m.

E. Ablation Study - FusionNet with Different Types of Fusion

To ascertain the efficacy of our gated fusion method, we compare FusionNet trained with this method with other popular fusion methods such as addition and concatenation. The results are shown in Fig. D. Gated fusion method outperforms the other two fusion methods by an average of 18.8% MAE and 10.6% RMSE. This is largely due to the largely noisy radar point cloud input. Because addition and concatenation directly introduces the noisy radar features into the RGB feature volume, it can be result in incorrect predictions in the final depth map. This is observed in Fig. D where both addition and concatenation yields similar results that are much worse than the proposed gated fusion – we conjecture that concatenation performs better than addition due to the convolutional operation i.e. linear combination with fixed weighting of noisy radar and image features. Unlike concatenation, our gated fusion is conditioned on the input and adaptively predicts the contribution of noisy radar points – "gating" them to reduce negative performance impact.

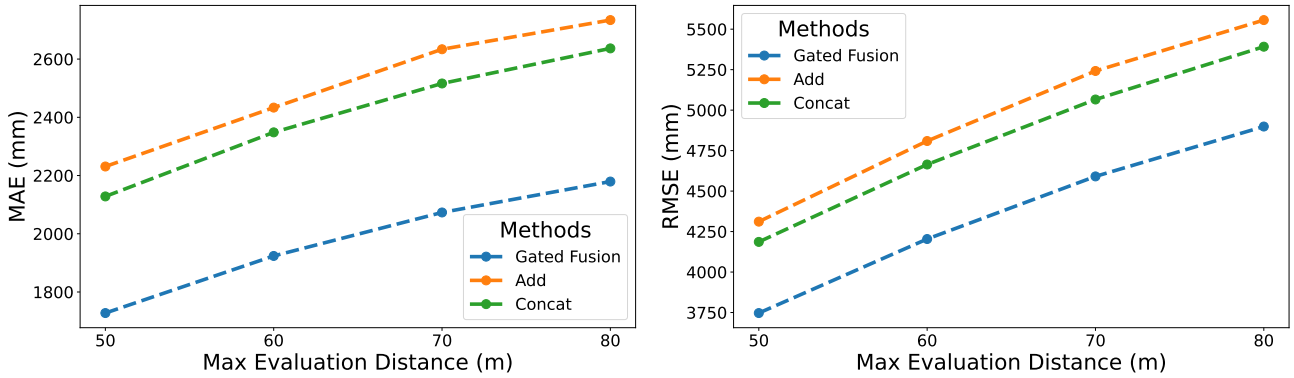

Figure D. Comparison of different fusion methods for FusionNet at various evaluation distances. The gated fusion method makes our model robust against noisy RadarNet output and hence produces the best results.

F. Failure Modes - RadarNet

In this section, we show and describe the failure modes of the RadarNet. As described in Sec. B, most of the points reflected back into the radar are from metallic surfaces such as cars – something that is also captured by the RadarNet model. In Fig. E, we show the failure modes of the RadarNet model. In the first row of Fig. E, we show a rare case where there are no points in the semi-dense depth. This is largely due to the scene being somewhat empty and often accompanied by low overall confidence scores predicted by RadarNet. With only two humans in the scene and the truck and the building being close together, the model is unsure as to where it should assign the depth values detected by the radar. As described by [19, 22, 26], the lack of points in the depth modality will degrade the performance of the network – in this case, the error of RadarNet will propagate and affect the output of FusionNet. In the second row of Fig. E, we show another empty scene. In this scene, the only objects of interest are some trees and poles on the side. The RadarNet output some points on the tree to the right as shown in red. However, it is not able to pick up any other objects. Such is also the case in the third row where the output of the RadarNet only contains a small number of points from the black SUV in the back. In scenes where half of the scene is sky (as shown in the third row), the RadarNet struggles to assign depth values to various objects in the scene because of the lack of reflected points. Additionally, we note that RadarNet is unable to associate points to sky regions, despite it being obviously far, because there are no points reflected.

Another failure mode is when the RadarNet output combines various objects and surfaces into one. In row four of Fig. E, the model tried to combine the guard rail on the side of the bridge as well as the concrete divider on which it is mounted into a single surface, as shown in red.

The fifth row of Fig. E demonstrates the manifestation of the another failure mode. When the objects are dark in color and under shadow, RadarNet is not able to assign them appropriate depth values.

G. Failure Modes - FusionNet

In this section, we show and discuss some of the failure modes of FusionNet. One common issue with using lidar as supervision is the fact that we do not have any points for the top region of the image (since due to the difference in field

Figure E. Failure modes for our RadarNet model (best viewed in color at 4x). The figure shows (L-R) the camera image, and the quasi-dense depth (output of the RadarNet). The range of depth is between 0-70m (as shown in the colorbar on the right). We mark the errors with red boxes and the corresponding objects in the image with green boxes.

of view of lidar, no lidar points project to the top of the camera image) and hence for most scenes, the top portion has no supervision. This means that the model is free to hallucinate depth in that region. Most of our failure cases are a direct result of our method being unsure as to what depth value to assign to the sky (which usually occupies the top portion of images in the nuScenes dataset). In Fig. F, we show the failure modes of our method. We plot the images on the left and the corresponding predicted dense depth on the right. The failure-modes are marked in red. We note that this issue plagues all existing methods, but they opt to crop out the sky regions for aesthetic reasons.

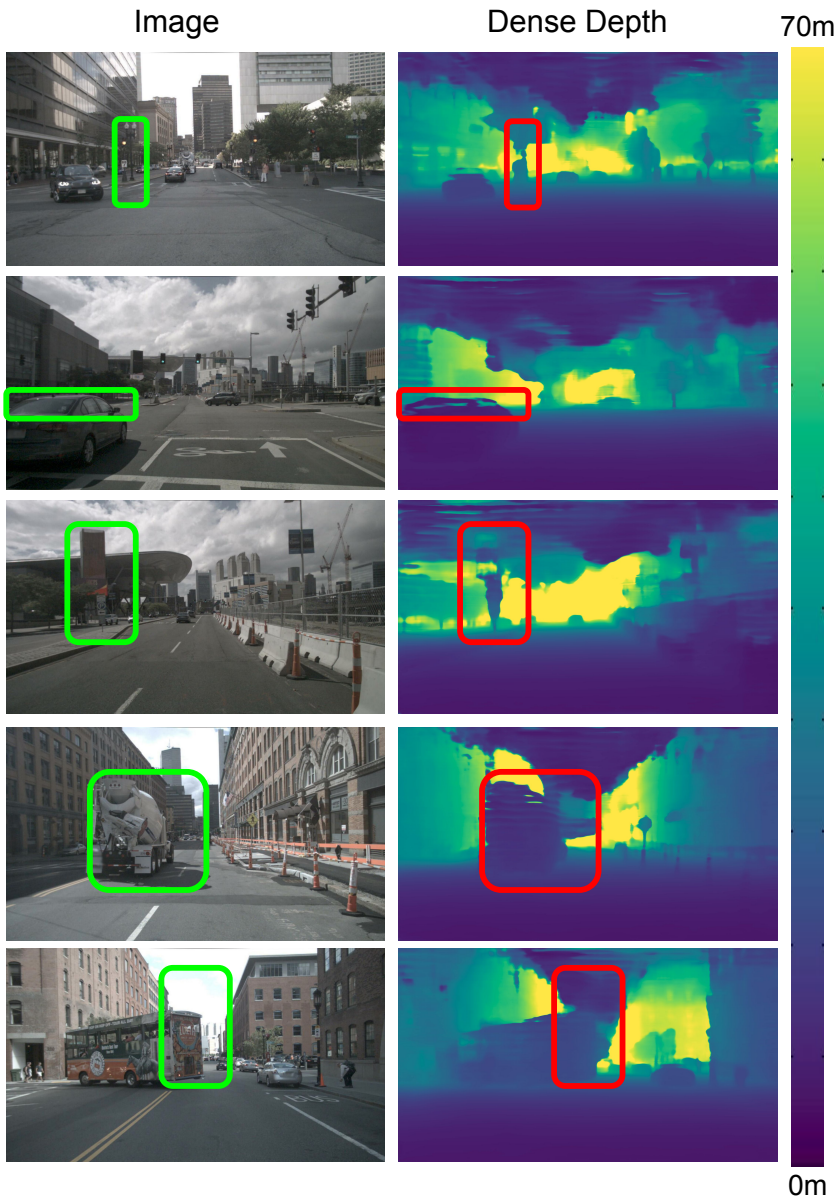

Figure F. Failure modes for FusionNet model (best viewed in color at 4x). The figure shows (L-R) the camera image, and the predicted dense depth (output of the FusionNet). The range of depth is between 0-70m (as shown in the colorbar on the right). We mark the errors with red boxes.

In the first row of Fig. F, there are a fire hydrant and a lamp post right next to each other as shown in green. Both of them have the same color. Our method confuses the two as the same and combines them as well as the region between them into the same object as shown in red. Additionally, it blends these objects into the shadow region of the building above them. The method also blends parts of the sky and parts of the top of the buildings in the scene.

The second row of Fig. F shows the same sky problem as described above. Additionally, since we are using reprojected ground truth to train the models, the sparsity of points on moving objects such as cars sometimes makes their depth output to be uneven and have gaps as shown in red.

In the third row of Fig. F, there is a signpost on the divider on the road and there is a larger sign board behind it across the road as shown in green. The method combines the two sign boards into one and then merges this joint object with the sky. This causes erroneous depth being predicted from the signboard across the road.

The fourth row of Fig. F is any extreme case of the sky-problem. While the method is able to distinguish the mixer truck driving in front of the sensors, it thinks that the sky above is an extension of this truck and assigns the sky similar depth values as the truck.

In the fifth row of Fig. F, we demonstrate another extreme case of the sky problem. The scene shows a bus turning left. The sky in the background is white – due to the lack of any clouds. In the depth output, the back of the bus merges with the sky creating one continuous object. Hence, the final prediction of our method yields incorrect depth values. Again, for objects such as the buildings and cars, the method is able to distinguish them.

Admittedly, the gap between radar depth completion and lidar depth completion is still large. For reference, top performing methods [9, 13, 14, 27] on the KITTI depth completion benchmark [15] achieve MAE scores of ≈0.2m; unsupervised methods [19–22, 26] that do not have access to ground truth accumulated lidar points clouds for supervision achieve ≈0.2-0.3m, which is an order of magnitude lower than the top radar depth completion methods. Radar depth completion performance lies between lidar and single image depth prediction [1, 4, 6, 10–12, 16, 17, 24, 25, 29] and affords metric depth which is typically absent in single image depth prediction methods outside of strong priors. While outside of the scope of this study, adversarial perturbations demonstrated on depth estimation methods [3, 18, 23] still present a challenge. Additionally, adverse weather conditions [2, 28] were not considered.

H. Model Parameters and Run-time Considerations

Our method uses two models – one for converting sparse radar point clouds into semi-dense depth (RadarNet) and one for fusing this semi-dense depth with images to output dense depth (FusionNet). The total trainable parameters for RadarNet model are $8,391,728$ while the total trainable parameters for the FusionNet model are $14,413,568$. The total trainable parameters for our method are $22,805,296$.

Method	Latency	Flow computation	Radar enhancement	Height extension	Forward pass	Total
RC PDA	166	91	51	N/A	20	328
RC PDA HG	166	91	51	N/A	7	315
DORN	154	N/A	N/A	160	56	370
Ours	**0**	**N/A**	**N/A**	**N/A**	**64 (R) + 13 (F)**	**77**
Ours (1 radar point)	**0**	**N/A**	**N/A**	**N/A**	**5.5 (R) + 13 (F)**	**18.5**

Table C. Run-time (ms) comparisons with SOTA. **R** denotes RadarNet and **F** for FusionNet. We assume 12 FPS camera and 13 FPS radar and radar point clouds of ∼30 points as in nuScenes [5].

We have provided run-times in Tab. C. Note that our method (RadarNet) does scale in run-time with number of points in the radar point cloud. Yet, on the other hand, RadarNet does take advantage of batching (see "Ours 1 radar point") the entire radar point cloud into a single forward pass; FusionNet run-time does not depend on the size of the radar point cloud. Considering that there is a dependency on the size of the radar point cloud, we do note that the point cloud provided by nuScenes [5] is already denser than most. Nonetheless, our full method (both RadarNet and FusionNet) is still faster than the fastest (RC-PDA HG) baseline method by 75.5%.

References

[1] 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. 9

[2] 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. 9

[3] 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. 9

[4] Ayush Bhandari, Achuta Kadambi, and Ramesh Raskar. *Computational Imaging*. MIT Press, 2022. 9

[5] 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. 9

[6] Xiaohan Fei, Alex Wong, and Stefano Soatto. Geo-supervised visual depth prediction. *IEEE Robotics and Automation Letters*, 4(2):1661–1668, 2019. 9

[7] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998. 4

[8] 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

[9] 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. 9

[10] 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. 9

[11] 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. 9

[12] 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. 9

[13] 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. 9

[14] 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. 9

[15] 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. 9

[16] Alexander Vilesov, Pradyumna Chari, Adnan Armouti, Anirudh Bindiganavale Harish, Kimaya Kulkarni, Ananya Deoghare, Laleh Jalilian, and Achuta Kadambi. Blending camera and 77 ghz radar sensing for equitable, robust plethysmography. *ACM Trans. Graph.(SIGGRAPH)*, 41(4):1–14, 2022. 9

[17] Zhen Wang, Shijie Zhou, Jeong Joon Park, Despoina Paschalidou, Suya You, Gordon Wetzstein, Leonidas Guibas, and Achuta Kadambi. Alto: Alternating latent topologies for implicit 3d reconstruction. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 9

[18] Alex Wong, Safa Cicek, and Stefano Soatto. Targeted adversarial perturbations for monocular depth prediction. *Advances in Neural Information Processing Systems*, 33, 2020. 9

[19] Alex Wong, Safa Cicek, and Stefano Soatto. Learning topology from synthetic data for unsupervised depth completion. *IEEE Robotics and Automation Letters*, 6(2), 2021. 5, 6, 9

[20] 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. 9

[21] Alex Wong, Xiaohan Fei, and Stefano Soatto. Voiced: Depth completion from inertial odometry and vision. *ArXiv, abs/1905.08616*, 2019. 9

[22] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5(2), 2020. 5, 6, 9

[23] 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. 9

[24] Alex Wong and Stefano Soatto. Bilateral cyclic constraint and adaptive regularization for learning a monocular depth prior. 2018. 9

[25] 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. 9

[26] 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. 5, 6, 9

[27] 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. 9

[28] 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. 9

[29] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 9