GET-UP: GEomeTric-aware Depth Estimation with Radar Points UPsampling

Supplementary Material

Supplementary Contents

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

- Section 1 provides motivation and evaluation of the proposed Adaptive Sparse Convolution Block (ASCB).
- Section 2 we analyse the effectiveness of proposed model.
- Section 3 further analyses the difficulty of the radar point cloud upsampling submodule.
- Section 4 introduces the evaluation metrics used for depth estimation task.
- Section 5 shows more qualitative results under different weather conditions.

1. Adaptive Sparse Convolution Block Analysis

As outlined in [4], the sparse convolutional layer was initially applied to lidar-only depth completion tasks. It operates on a sparse lidar depth map by generating a mask to identify pixels filled with data. A sequence of sparse convolutional layers with varying kernel sizes then processes the sparse depth map alongside the generated mask, facilitating information flow from known to unknown pixel positions.

In the context of driving, objects farther from the ego-vehicle appear smaller on the image plane, and radar detections are predominantly triggered by moving objects. These observations guide our selection of an adaptive combination of sparse convolutional layers aimed at achieving optimal receptive fields for different distances.

This section begins by analyzing the dimensions of 2D bounding boxes across specified distance ranges: $[0, 40)$, $[40, 70)$, and $[70, +\infty)$. Subsequently, we delve into the specifics of our ASCB.

Fig. 1 showcases a boxplot illustrating the variation in width and height of 2D bounding boxes within these ranges, highlighting the significant size disparity between the $[0, 40)$ range and the $[70, +\infty)$ range. This variation motivates our approach to process sparse radar depth maps D_R by generating distinct masks for different distances: M_0^{40}, M_{40}^{70}, and $M_{70}^{+\infty}$, based on the depth value of each radar detection.

For instance, the mask for the $[0, 40)$ range is defined as follows:

$$M_0^{40}(i,j) = \begin{cases} 1 & \text{if } 0 \leq D_R(i,j) < 40 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

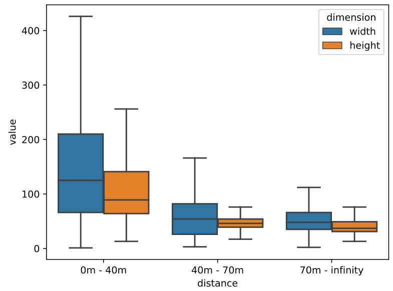

Figure 1. Analysis of the width and height of the 2D bounding boxes appear within ranges $[0, 40)$, $[40, 70)$, and $[70, +\infty)$.

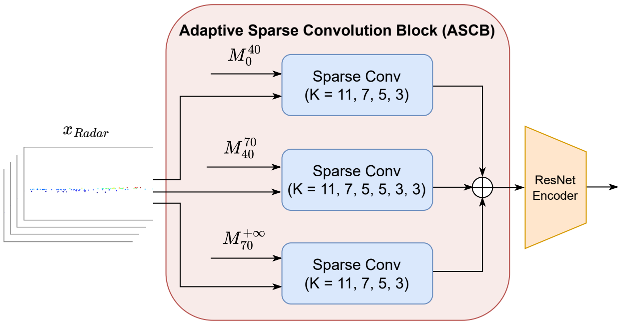

Figure 2. Visualization of the proposed Adaptive Sparse Convolution Block, which takes radar projected map x_{Radar} and the extracted masks M_0^{40}, M_{40}^{70}, and $M_{70}^{+\infty}$ as inputs.

Kernel size combinations for each distance range are then chosen based on the observed bounding box sizes. Considering the median bounding box size is approximately 80 pixels in the $[0, 40)$ range, achieving a receptive field of 80 requires a substantial increase in parameters. Ultimately, we select the kernel size sets of $[11 \times 11, 7 \times 7, 7 \times 7, 5 \times 5, 5 \times 5, 3 \times 3]$, $[11 \times 11, 7 \times 7, 5 \times 5, 5 \times 5, 3 \times 3, 3 \times 3]$, and $[11 \times 11, 7 \times 7, 5 \times 5, 3 \times 3]$ for the distance ranges of $[0, 40)$, $[40, 70)$, and $[70, +\infty)$ meters, respectively. These configurations, with receptive fields of 33, 29, and 23, are tailored to each range, ensuring appropriate propagation lengths for effective radar data processing. The proposed ASCB architecture is visualized in Fig. 2.

2. Model Efficiency Analysis

Our model comprises 50 million training parameters, including 22 million for image feature extraction, 16 million assigned to the radar encoder, and 12 million for decoding. Within the radar encoder, 75% of the parameters are allocated to the ResNet encoder. We conducted additional experiments using only the ResNet18 encoder for radar feature extraction while keeping the rest of the architecture unchanged. This configuration achieved an MAE of 1.925. By incorporating an additional 4M parameters to extract 3D features, aggregate 2D and 3D features, and perform point cloud upsampling, our final proposed model achieves a 6.01% improvement in MAE, demonstrating the effectiveness of our method.

3. Radar Point Cloud Upsampling Submodule Analysis

This section first describes the difficulties of the proposed upsampling task. Then, we visualize the upsampling results.

3.1. Upsampling Ground Truth Generation

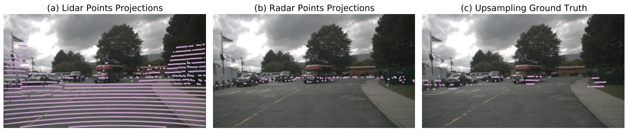
(a) Lidar Points Projections
(b) Radar Points Projections
(c) Upsampling Ground Truth

Figure 3. Comparison of the radar and lidar point cloud density. (c) visualize the selected lidar point projections employed for point cloud upsampling.

Since there is no ground truth to upsample the radar point cloud to make it denser and clearer, we detail a method for generating ground truth for point cloud upsampling in the main paper, utilizing lidar data. Specifically, for a given frame, we compute the Chamfer distance [1] between the lidar and radar point clouds, selecting the N_L lidar points closest in the distance. Fig. 3 displays the projection of radar and lidar points in parts (a) and (b), respectively, and showcases the projection of the chosen ground truth points in part (c).

Difficulties. Due to the inherent noisiness and ambiguity of radar point clouds, many points are inaccurately positioned. This is depicted in Fig. 4, where points within the blue box are accurately returned from a moving car, while those within the yellow box predominantly represent noise. Consequently, as illustrated in part (b) of Fig. 4, the majority of the upsampling ground truth points are concentrated in the blue box, closer to the precise radar detections. For clearer visualization, Fig. 4 (c) presents a combined plot of ground truth points and radar points, depicted in blue and yellow respectively.

This proximity poses a challenge for the network in learning the offsets for radar points that are significantly distant from their corresponding ground truth points.

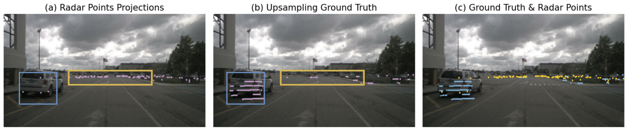
(a) Radar Points Projections
(b) Upsampling Ground Truth
(c) Ground Truth & Radar Points

Figure 4. Visulization of the difficult case.

3.2. Upsampling Results

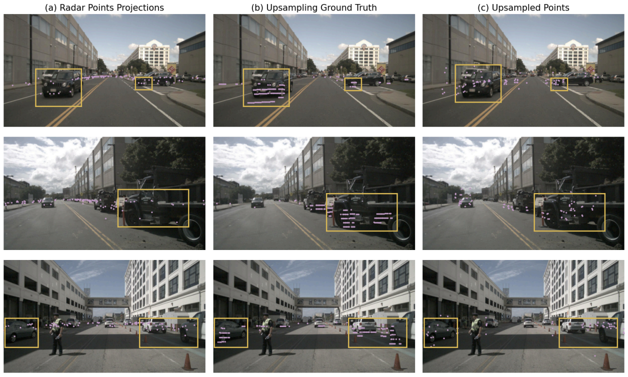

Figure 5. Visulization of upsampled points along with the original radar points and the sampled upsampling ground truth.

In our approach, point cloud upsampling serves as an auxiliary task to depth estimation, designed to derive meaningful features from precise lidar data. Consequently, our upsampling model is introduced as a compact plug-in module, consisting of merely around 600 thousand parameters. Given this limited parameter set, achieving highly accurate upsampled positions relative to the ground truth, particularly for the challenging scenarios outlined in Sec. 3.1, proves to be difficult.

However, the results show that the upsampled points are slightly moving toward the ground truth. Fig. 5 illustrates this through a series of visualizations: the first column (a) displays the projected radar points; the second column (b) shows the lidar points selected as upsampling ground truth; and the third column (c) presents the predicted upsampled points. Notably, in these examples, most of the ground truth points cluster around the detected object, highlighted by the yellow box in (b). In the estimated upsampled points, the density of the points around the detected objects increases in the regions highlighted in yellow boxes. At the same time, the number of noisy points decreases where no ground truth points are located. These results indicate the efficacy of this upsampling submodule.

4. Evaluation Metrics

Table 1 outlines the evaluation metrics employed in this study for comparing performance. Here, Ω denotes the set of 2D pixels for which ground truth LiDAR depth values are available.

5. Qualitative Results

In this section, we compare depth maps predicted by our proposed GET-UP and the best-performing fusion model [2] under sunny, rainy, and night conditions.

Table 1. Metrics definition for depth estimation task.

	Definition
MAE	$\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\hat{d}(x)-d_{gt}(x)\rvert$
RMSE	$(\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\hat{d}(x)-d_{gt}(x)\rvert^2)^{1/2}$
AbsRel	$\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\hat{d}(x)-d_{gt}(x)\rvert/d_{gt}(x)$
log10	$\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\log_{10}\hat{d}(x)-\log_{10}d_{gt}(x)\rvert$
RMSElog	$\sqrt{\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\lvert\log_{10}\hat{d}(x)-\log_{10}d_{gt}(x)\rvert\rvert^2}$
δ_n Thre	$\delta_n=\lvert\{\hat{d}(x):max(\frac{\hat{d}(x)}{d_{gt}(x)},\frac{d_{gt}(x)}{\hat{d}(x)})<1.25^n\}\rvert/\lvert\Omega\rvert$

5.1. Under Night Scenario

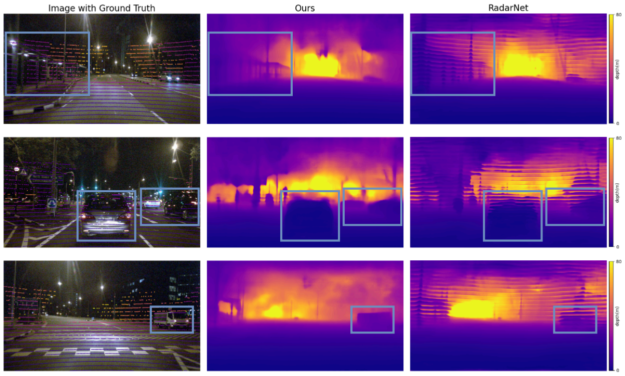

Figure 6. Qualitative Comparison between our GET-UP and RadarNet [3] under night condition.

As illustrated in Fig. 6, the depth maps generated by [3] exhibit noticeable discontinuities, in contrast, our approach succeeds in producing consistently smooth and accurate dense depth maps, even in low-light conditions. Objects are highlighted in blue boxes to demonstrate our method's capability to precisely detect objects and define their boundaries under challenging lighting scenarios.

5.2. Under Rainy Scenario

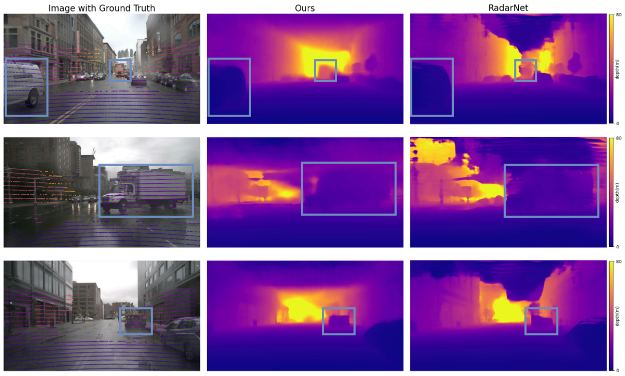

Figure 7. Qualitative Comparison between our GET-UP and RadarNet [3] under rainy condition.

As depicted in Fig. 7, RGB images become blurry in rainy conditions, posing additional challenges. Despite this, our GET-UP method successfully identifies objects at long distances, even in rainy scenarios. For instance, in the first row, our approach accurately detects a truck at a far distance, whereas RadarNet [3] fails to estimate the correct range for this truck. Objects are highlighted in blue boxes to facilitate a clearer comparison of the outcomes.

5.3. Under Sunny Scenario

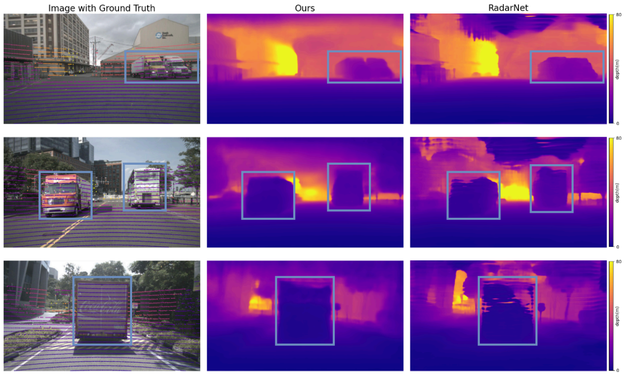

Figure 8. Qualitative Comparison between our GET-UP and RadarNet [3] under sunny condition.

Fig. 8 showcases the dense depth maps predicted in sunny conditions. Our GET-UP method successfully identifies and distinguishes between two trucks in the first row, a task where RadarNet falls to differentiate between the two vehicles. In subsequent rows, our technique demonstrates its capability to define clear boundaries for each object, in contrast to RadarNet, which struggles to accurately capture the shape of the objects.

In summary, our approach exhibits superior performance across various weather conditions, underscoring the effectiveness of the proposed method.

References

[1] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2

[2] Felix Fent, Philipp Bauerschmidt, and Markus Lienkamp. Radargnn: Transformation invariant graph neural network for radar-based perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 182–191, 2023. 2

[3] Akash Deep Singh, Yunhao Ba, Ankur Sarker, Howard Zhang, Achuta Kadambi, Stefano Soatto, Mani Srivastava, and Alex Wong. Depth estimation from camera image and mmwave radar point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9275–9285, 2023. 3

[4] 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. 1