Depth Estimation via Sparse Radar Prior and Driving Scene Semantics

Ke Zheng$^{1,*[0000-0002-4558-6039]}$, Shuguang Li$^{1,*,\dagger[0000-0001-6818-5345]}$,
Kongjian Qin^{2,†}, Zhenxu Li^{1}, Yang Zhao^{1}, Zhinan Peng^{1}, and Hong Cheng^{1}

[1] University of Electronic Science and Technology of China, Sichuan, China
[2] China Automotive Technology and Research Center Co. Ltd., Tianjin, China

Abstract. Depth estimation is an essential module for the perception system of autonomous driving. The state-of-the-art methods introduce LiDAR to improve the performance of monocular depth estimation, but it faces the challenges of weather durability and high hardware cost. Unlike existing LiDAR and image-based methods, a two-stage network is proposed to integrate highly sparse radar data in this paper, in which sparse pre-mapping module and feature fusion module are proposed for radar feature extraction and feature fusion respectively. Considering the highly structured driving scenario, we introduce semantic information of the scenario to further improve the loss function, thus making the network more focused on the target region. Finally, we propose a novel depth dataset construction strategy by integrating binary mask-based filtering and interpolation methods based on the nuScenes dataset. And the effectiveness of our proposed method has been demonstrated through extensive experiments, which outperform existing methods in all metrics.

Keywords: Depth estimation · Multi-sensor · Autonomous driving · Driving scene characteristic.

1 Introduction

The autonomous driving perception systems using a monocular camera can detect the target accurately, but it is difficult to confirm the location and scale of the target, due to the missing depth information. Recently, the use of convolution neural network is demonstrated to have significantly improved the accuracy of depth estimation [13, 18, 16, 35, 17]. All of the above work describes depth estimation as a regression problem. Accurate regression of depth values remains a challenge in this field. Thus, [11, 3] were significant to transform the regression problem into the classification problem. Researchers [16, 9] have attempted to improve the accuracy of depth estimation by solving multiple tasks in a joint manner.

* K. Zheng and S. Li are co-first authors.
† Corresponding authors. Email: lsg042@163.com.

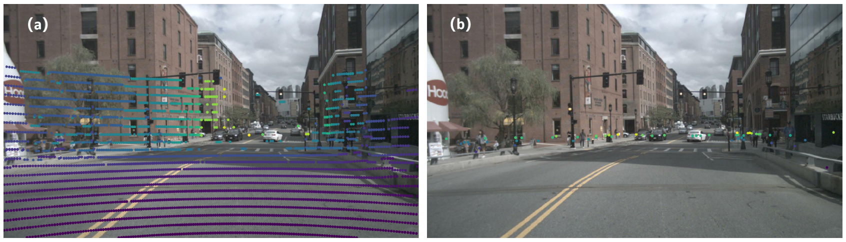


Fig. 1. Projection of the LiDAR and radar data points onto the image plane according to the calibration relationship shows their high degree of sparsity. (a) is the LiDAR projection point, (b) is the radar projection point.

However, monocular depth estimation is known as an ill-posed problem [10]. If pixels with known depth values are present in the image, the difficulty of the monocular depth estimation task is significantly reduced. A common method is the introduction of LiDAR data, i.e. the depth completion task [29, 6, 15]. Although LiDAR provides denser depth observations, the resolution is highly dependent on weather and is more expensive to acquire, the introduction of LiDAR also poses a considerable challenge to the overall system in real-time. The radar, on the other hand, is an all-weather sensor that performs well in bad weather and has a wide effective detection range. And radar has already been widely used in automotive industry applications such as Adaptive Cruise Control and Automatic Emergency Braking, which makes radar even more attractive to be applied for the depth estimation module. Therefore, we introduce radar measurements as prior knowledge to achieve high accuracy of depth estimation.

Conceivably, fusing radar and vision for depth estimation are great challenges for researchers in the field, due to the following technical barriers. 1) The only large dataset containing radar data is the nuScenes[2] dataset, but this dataset only provides the raw point cloud per frame. If only one frame of data is used as ground truth, only about 0.2% of pixels per image exist for supervision, as shown in Fig. 1(a), which would not be conducive to the model learning object contour information and detail information. 2) Compared to LiDAR, the radar point cloud is much more sparse, providing only about 0.003% of the prior depth value for each image, as shown in Fig. 1(b). Therefore, the feature of radar needs to be extracted in a way that is more suitable for dealing with sparse point clouds.

To address the above issues, we propose a new LiDAR data processing scheme to construct a denser and less noisy ground truth depth based on multiple frames of LiDAR data. The radar depth network (RDNet) is proposed as a two-stage network. Specially, considering the different input and purpose of the two stages, the sparse-coarse stage adopts a dual encoder-single decoder structure. Due to the high sparsity of the input radar data, the sparse pre-mapping module(SPM) is used to initially extract sparse feature. The dense depth map obtained in this stage is used as input to the coarse-fine stage, that single encoder-decoder structure and uses the feature fusion module(FFM), which introduces channel attention mechanism to fuse the features of the two stages. In addition, existing depth estimation networks do not take into account the characteristics of the driving scene. And they treat each pixel equally, while accurate depth values are often used in obstacle avoidance and 3D object detection, where the accuracy of the depth value of the target is required to be extremely high. Therefore, we further improve the loss function to make the network more focused on the corresponding region of the target. Our main contributions are as follows:

– We propose a denser and less noisy ground truth depth generation method, which solves the problem that the network can not learn details and edge information from sparse data.

– A new network structure better suited for collecting valid information from sparse radar data is proposed, including the sparse pre-mapping module and the feature fusion module.

– Improved loss function based on prior knowledge of the driving scenario, resulting in more accurate depth estimation of the target area.

– Various experiments have been carried out on the proposed new dataset based on nuScenes dataset to verify the effectiveness of the method compared to the state-of-the-art methods.

2 Related Work

2.1 Monocular depth estimation

Monocular depth estimation is currently addressed by training on large datasets with an attempt to acquire the ability to perform depth estimation. Early methods focused on depth estimation using hand-crafted features [27, 25]. Recently, the main methods can be summarized as: 1) Introduction of attention mechanism [20, 5, 34]. For example, Li et al. [20] used channel attention mechanism in their model to extract the distinguishing features. 2) Dispersing the continuous depth into a certain number of bins, thus the regression problem is transformed into the classification problem [11, 1]. Fu et al. [11] used spacing-increasing discretization strategy to discretize given depth in log domain. Bhat [1] used an adaptive discretization strategy to compute bin lengths for each image. 3) Jointly solving for multiple tasks [9, 30]. [9] used generic architecture to jointly solve the three tasks of depth estimation, surface normal estimation and semantic segmentation. Wang et al. [30] proposed to decompose the image into several semantic segments, predicting a normalized depth map for each segment. 4) Optimizing coarse depth maps using CRF [19, 31]. For example, Li et al. [19] introduced Hierarchical Conditional Random Fields to refine depth estimation from the super-pixel level to the pixel level.

We introduce highly sparse radar point cloud on top of the monocular depth estimation task, aiming to provide prior points and thus reduce the overall task difficulty. In addition, unlike existing methods that jointly solve multiple tasks, we introduce semantic information to make the overall depth estimation more relevant to the driving scenario requirements without increasing the computational effort of the network.

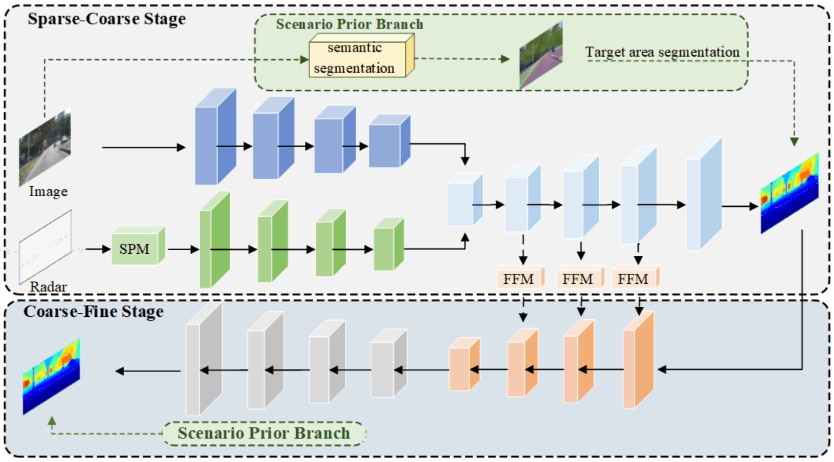


Fig. 2. Overview of proposed network architecture. It consists of two stages, the sparse-coarse stage and the coarse-fine stage.

2.2 Depth completion

Depth completion task has additional characteristics compared to monocular depth estimation, such as the depth values of sparse points should be maintained as much as possible, and the transition between sparse points and their neighborhoods should be smooth.

These methods can be roughly divided into two catalogs: 1) After the network has predicted the coarse depth map, it is optimized by local neighborhoods. Cheng et al. [7] proposed convolutional spatial propagation network, which was propagated by recurrent convolution. Xu et al. [33] proposed to learn adaptive offsets in the network. 2) Using images to guide the recovery of depth maps. Tang et al. [26] calculated weights after extracting features from an image, and multiplied the weights when encoding sparse depth inputs. There were other methods that make use of feature representations in 3D space [4], surface normal [24, 32] and so on.

The difference between radar and camera-based depth estimation and lidar and camera-based depth completion lies in the adequacy of the known depth information. In depth completion task, the image can be used as a guide to reconstructing the dense depth from the sparse input. However, because the input from radar is so sparse, it is more appropriate to provide prior depth information for the image using radar.

3 Proposed Method

In this section, we first introduce the overall architecture of RDNet, as well as the sparse pre-mapping module (SPM) for processing sparse radar data and the feature fusion module (FFM) for two stage feature fusion. The loss function used to train the network is then described, with further improvement of the loss function based on the driving scene semantics. Finally, the specific construction of the ground truth depth is presented.

3.1 Network architecture

As shown in Fig. 2, we design an end-to-end depth estimation network framework based on radar and image. As estimating an accurate depth map directly from highly sparse radar and image via a single-stage network is a relatively difficult task, we design a two-stage network with the sparse-coarse stage and the coarse-fine stage respectively. The sparse-coarse stage takes the image and sparse radar data as input to predict a dense but coarse depth map. In this stage, in order to make full use of the feature of radar data for the effective fusion of image and radar, a dual encoder-single decoder architecture is used. That is, the image and radar are fused after extracting features separately, and then the final depth map is predicted by a decoder. Specially, the image encoder is constructed using ResNet-34 [14], which is pretrained on ImageNet [8]. In depth encoder, given the highly sparse nature of the radar data, we propose sparse pre-mapping module to extract the initial feature, and then use residual blocks to extract further feature. The decoder consists of four up-projection blocks [23], followed by a 3×3 convolution that maps the output to a depth map, and finally the depth map is restored to its initial resolution using bilinear upsampling. The coarse-fine stage uses the depth map of the previous prediction as input, which uses single encoder-decoder structure and feature fusion module to fuse the features of the two stages for obtaining a more accurate prediction. And as the driving scenario is relatively structured, we introduce the scenario prior branch to further enhance the network effect.

3.2 Sparse pre-mapping module

Considering the highly sparse nature of the radar input, standard convolution in sparse data processing would result in poor performance. And there are inconsistencies between LiDAR and radar data, even with effective noise filtering, there is still variability between them. To address the problem, the sparse pre-mapping module is set up before the residual block in the depth branch to enable mapping between data and the extraction of sparse feature. First, we briefly recall the sparsity-invariant convolution in [28], which takes a sparse feature map x and a binary mask m as input. It can be formulated as:

$$f(u,v) = \frac{\sum_{i,j=-k}^{k} m_{u+i,v+j} w_{i,j} x_{u+i,v+j}}{\sum_{i,j=-k}^{k} m_{u+i,v+j} + \delta} + b \tag{1}$$

where δ denotes a very small number to avoid the problem of dividing by zeros when there is no observed depth in the convolution region.

As shown in Fig. 3(a), our sparse pre-mapping module obtains a denser feature map by stacking sparsity-invariant convolutions. And in order to complete

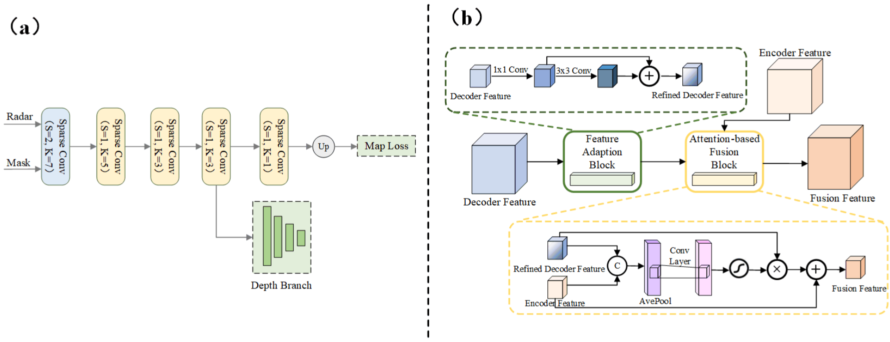


Fig. 3. (a) is the sparse pre-mapping module, which inputs are sparse radar depth map and binary mask, the whole module consists of several sparsity-invariant convolutions. (b) is the feature fusion module, the module takes features of the same resolution from the sparse-coarse stage and the coarse-fine stage as input and introduces channel attention mechanism that integrates the two stages to enhance feature representation.

the mapping between the data, the module output is bilinearly upsampled to the initial resolution and then supervision is applied. To further extract the feature of radar, the output of the fourth convolution is fed into the depth branch and further semantic features are extracted using the residual block.

3.3 Feature fusion module

To make the coarse-fine stage feature contain richer information and thus predict a more accurate final depth map, we use the strategy of decoder-encoder fusion in [26] to fuse feature from the sparse-coarse stage into the coarse-fine stage. And the feature representation is further enhanced by the introduction of channel attention mechanism. Specially, to adapt the decoder feature in the sparse-coarse stage to the encoder feature in the coarse-fine stage, we further refine the decoder feature and reduce its channel number using a convolution block with the residual connection. The structure is shown in Fig. 3(b), it can be formulated as:

$$Y_i = ReLU(Conv(F_i) + f_1(Conv(F_i))) \tag{2}$$

where F_i and Y_i denote the feature at layer i of the decoder and the corresponding output features and $f_1(\cdot)$ denotes the learnable feature refinement mapping.

After obtaining the refined feature Y_i and concatenating it with the encoder feature of the coarse-fine stage, We apply the channel attention mechanism to it, which adaptively adjusts the feature values of each channel, selectively enhancing feature containing useful information and suppressing useless feature through global information. Specially, global average pooling is used to obtain global context information, and then the attention vector is computed to enhance feature representation. It can be formulated as:

$$y = \sigma(f_2(Global[Y_i, F_i^s])) \odot Y_i + F_i^s \tag{3}$$

where *Global* represents the global pooling operation, F_i^s denotes the feature of current stage, $f_2(\cdot)$ denotes the learnable weight mapping, $\odot$ is element-wise product operator. $[\cdot, \cdot]$ denotes concatenate operation and σ denotes sigmoid function.

3.4 Loss Function

Differences in loss functions can have an impact on final depth estimation performance. We use $L1$ loss to calculate the loss between the ground truth depth and the predicted depth. Since the ground truth does not exist in every pixel, only the loss of valid pixels in the ground truth is computed, denoted as:

$$L_{depth} = \frac{1}{m} \sum_{(p,q)\in S} |d_{p,q} - \hat{d}_{p,q}| \tag{4}$$

where d and $\hat{d}$ denote the ground truth depth map and the predicted depth map respectively. S denotes the set of valid depths of d and m is the number of valid depths.

Further, we add edge-aware smoothness constraint [12] to encourage depth locally smooth. Since depth discontinuities usually occur at junctions, the image gradients are used for weighting, L_{smooth} is defined as:

$$L_{smooth} = e^{-|\partial_x(I)|}|\partial_x(\hat{d})| + e^{-|\partial_y(I)|}|\partial_y(\hat{d})| \tag{5}$$

where ∂_x, ∂_y denote the gradients along the x and y directions respectively, I denotes the input image.

In the training process, since we adopt the supervision to the depth prediction of the two stages and to the mapping result in the first stage, the overall loss function is represented as a weighted function:

$$L_{total} = \lambda_1(L_{coarse} + \lambda_2 L_{map} + \lambda_3 L_{smooth}) + (1 - \lambda_1)F_{final} \tag{6}$$

where λ_1, λ_2, λ_3 are hyperparameters. In addition, L_{coarse}, L_{map} and L_{final} denote the loss of the sparse-coarse stage, the loss of the sparse pre-mapping module and the loss of the coarse-fine stage respectively.

4 Scenario semantics_based depth estimation

Unlike other scenarios where the driving scene is relatively structured and has much prior knowledge that can be exploited. Much of the existing work has focused on the design of the network to improve its overall performance. However, the characteristics and requirements of the driving scenario have not been fully considered. Especially, in driving scenes, we usually want to get as accurate a depth value as possible for target areas, which include vehicles, pedestrians, etc.,

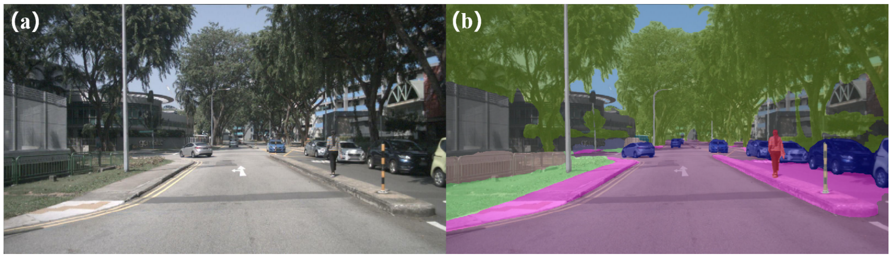


Fig. 4. Semantic segmentation result. (a) is the input image, (b) is the semantic segmentation result (In this scene, the blue and red areas are the target area we have defined).

while we do not need to get very accurate depth values for not target areas. In summary, we want the depth estimation network to focus more on the target areas of the scene, rather than treating all areas equally.

Using this idea as a starting point, we analyze the results of the depth estimation. First, we introduce a semantic segmentation network, here using SegFormer, and the segmentation results are shown in Fig. 4. After obtaining the semantic labels for each scene, the pixels within the scene are classified into target/not target areas, and the RMSE/MAE of the two areas are calculated and the results are shown in Table 1. From the results, it can be seen that the errors in the target areas are even much larger than in the not target areas. We further analyze the number of pixels in the target and not target area and find that in most scenes the number of pixels in the target area is about 1/10 of the number of pixels in the not target area. We believe this can be seen as a sample imbalance. As the samples corresponding to not target area dominate the loss function, the model also tends to favor not target area during training, resulting in poorer performance in depth estimation for the target area.

Table 1. Errors in target area versus not target area.

Area	RMSE	MAE
Target Area	8.379	4.929
Not target Area	5.591	2.463

We make an improvement to Eq. 4 by introducing semantic segmentation results into the training process of the network, weighting the depth loss of different areas separately, thus making the network focus more on the target area. The improved depth loss is calculated as:

$$L_{depth} = \frac{1}{m}(\omega \sum_{(p,q)\in S_{TAR}} |d_{p,q} - \hat{d}_{p,q}| + (2-\omega) \sum_{(m,n)\in S_{NTAR}} |d_{m,n} - \hat{d}_{m,n}|) \quad (7)$$

where S_{TAR} and S_{NTAR} denotes the set of valid depths of target area and not target area, and m is the number of valid depths. ω is a weighting factor to balance the loss in the two areas.

Since we only improve the loss function without altering the network structure, we do not introduce any additional computational effort in the inference process of the network. This is in contrast to current networks that improve the depth estimation network into a multi-task network [30, 31]. Moreover, our purpose is also different from existing networks. We are aiming to improve the accuracy of target area estimation, whereas they mostly aim to improve the representation of the backbone.

5 Dataset generation

In contrast to the ground truth depth in the KITTI dataset [28], which is supervised for approximately 30% of the pixels, the nuScenes dataset provides only the raw LiDAR point cloud, which after projection is supervised for only approximately 0.2% of the pixels per frame. As the ground truth depth is too sparse and the evaluation metrics are only calculated at the pixels where supervision is present, the evaluation metrics do not give a full picture of the depth estimation and the model does not learn the details of the scene. Therefore, we propose a new LiDAR data processing scheme by integrating the binary mask-based filtering and interpolation method to construct a dense and less noisy ground truth on the basis of multi-frame LiDAR data.

Let L and m_L denote the aggregated multi-frame LiDAR data of size $H \times W$ and their corresponding sparse mask, respectively. The filtered LiDAR data can be represented as:

$$L' = g(L, m_L) \tag{8}$$

where g is the filtering operation.

Given that there are many unobserved points in the LiDAR data, the conventional filtering algorithms fail to consider the sparsity pattern of the data, which will alter the observation points intended to be the true values for training. We propose an observation point invariant filtering technique to alleviate the issue. Our proposed operation first masks non-observed points in the LiDAR data using a sparse mask m_L, and then finds the mean value of the depth of the observed points in region S of size $n \times m$. The outliers are obtained by testing whether the difference between the depth of the observation and the mean value is greater than threshold value, function g is defined as:

$$g = \begin{cases} L(p, q) & |L(p, q) - ave(p, q)| < \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and

$$ave(p, q) = \frac{1}{M} \sum L(x, y) m_L(x, y) \tag{10}$$

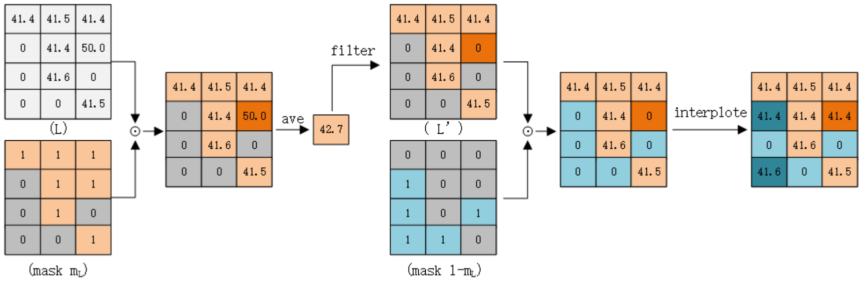


Fig. 5. Illustration of the entire process of LiDAR data processing.

where, M is the number of observed pixels, $ave(p, q)$ is the mean value at the current position (p, q), ε is the threshold value and S is a filter window of size $n \times m$.

Once the outliers are removed, the filtered LiDAR data L' and the sparse mask $1 - m_L$ are used for interpolation to obtain denser ground truth depth L_i, which can be formulated as:

$$L_i = f(L', 1 - m_L) \tag{11}$$

where f is the proposed interpolation operation.

Specifically, we use the sparse mask $1 - m_L$ to mask the observed points in the LiDAR data L' and interpolate the non-observed positions. Let T_x and T_y be the step in the x and y directions, the masked data points are traversed with the pre-determined step size, and find the nearest neighbors within a window of size (a, b) centered on the currently traversed data points (p, q). Since it is observed that the depth values vary much more in the y direction than in the x direction in the driving scenario, we set $a > b$. Function f is defined as:

$$f = \begin{cases} Nearest(p, q) & m_L(p, q) = 0 \\ L'(p, q) & m_L(p, q) \neq 0 \end{cases} \tag{12}$$

where $Nearest(p, q)$ denotes the search for nearest neighbor observations in a window centered on (p, q).

As we use a fixed step in the interpolation step, we sample the interpolated points in order to break the regularity of the interpolated data. The entire process of LiDAR data processing is illustrated in Fig. 5 and is represented as:

$$L_f = Sample(f(g(L, m_L), 1 - m_L)) \tag{13}$$

6 Experiments

6.1 Implementation detail

We use the nuScenes dataset [2] to validate the above model. The dataset consists of 1000 scenes, each of which lasts 20 seconds long, with 40 keyframes. Each frame has a resolution of 1600×900, which we crop to the size of 1600×704 for training and testing. And nuScenes dataset contains driving scenarios in various conditions, which makes it more difficult to perform depth estimation on this dataset. We use 850 scenes and divide them into 810 scenes for training and 40 scenes for evaluation.

In order to train the proposed model, we perform the data augmentation with the color and horizontal flip transformation. We use Pytorch to deploy the network and train on an NVIDIA GeForce GTX TITAN X with 12G memory. In all experiments, batch size is set to 4. We use Adam optimizer with learning rate 0.0005, and the learning rate is reduced to half every 5 epochs. The weights in loss function are set to $\lambda_1 = 0.5, \lambda_2 = 0.3, \lambda_3 = 0.001$.

The standard metrics utilized for monocular depth estimation and depth completion work [1, 31] are adopted for comparison. Let d and $\hat{d}$ be the predicted depth map and ground truth depth map, respectively, the evaluation metrics are:

Root Mean Square Error (RMSE):$\sqrt{\frac{1}{n}\sum_p^n (d_p - \hat{d_p})^2}$

Mean Absolute Error (MAE):$\frac{1}{n}\sum_p^n |d_p - \hat{d_p}|$

Mean Absolute Relative Error (REL): $\frac{1}{n}\sum_p^n \frac{|d_p - \hat{d_p}|}{d_p}$

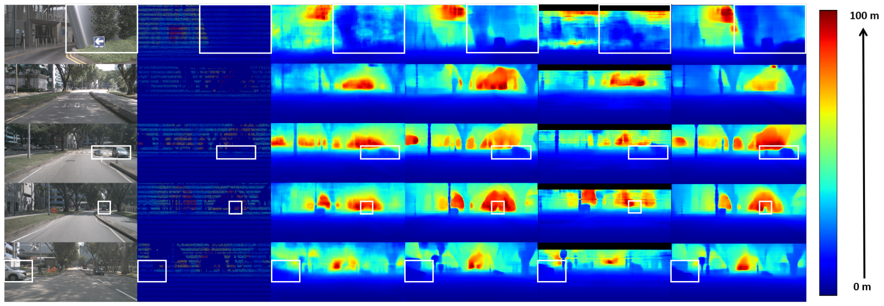
Threshold accuracy (δ_i):$max\left\{\frac{\hat{d_p}}{d_p}, \frac{d_p}{\hat{d_p}}\right\} < t, t = 1.25, 1.25^2, 1.25^3$.

6.2 Comparing Performance

To demonstrate the effectiveness of the proposed method, we train the models under the same conditions and compare our method with the image and LiDAR-based depth estimation methods proposed by Ma et al. [22] and Hu et al. [15] and the image and radar-based methods proposed by Lin et al. [21]. Those results are outlined in Table 2. Our method exhibits better performance with improvement in all evaluation metrics. In particular, compared to RadarNet [21], our RDNet reduces the RMSE by 0.518m and the MAE by 0.338m. It is also evident from the experiments that most LiDAR and image-based methods are not well adapted to radar, whereas our model has a unique capability to extract useful features from sparse data. The qualitative results shown in Fig. 6 imply that our method restores more accurate detail information than other methods, with fewer regions of incorrect depth estimation at image edges and object boundaries.

Table 2. Comparisons to advanced methods on nuScenes. Best results are in **bold** font.

Method	Error ↓			Acc ↑		
	RMSE	MAE	REL	δ_1	δ_2	δ_3
Hu et al. [15]	6.882	3.630	0.187	0.779	0.916	0.963
Lin et al. [21]	5.889	2.640	0.118	0.874	0.950	0.976
Ma et al. [22]	7.195	3.430	0.164	0.809	0.916	0.959
Ours	**5.371**	**2.302**	**0.103**	**0.897**	**0.960**	**0.980**



(a) Input image (b) Ground truth (c) Hu et al. [12] (d) Lin et al. [35] (e) Ma et al. [34] (f) Ours

Fig. 6. Qualitative results on nuScene. The proposed method obtains clearer details.

6.3 Ablation studies

We conduct a series of ablation studies to verify the effectiveness of the various components proposed in our approach, including the late fusion structure, sparse pre-mapping module and feature fusion module. At the same time, we verify the effect of different loss functions, the role of adding radar data to the model, and analyze the setting of hyperparameters.

Image Branch Encoder: We use different baselines in the image encoder to extract the image feature, here only ResNet-34 and ResNet-18 are tested, as seen in Table 3, ResNet-34 achieves better performance, which we believe is due to the expanded network allowing better extraction of image feature. So we use ResNet-34 as the backbone in the subsequent comparison experiments.

Fusion Method: In this experiment, we compare the performance of the late fusion structure with the early fusion structure. The late fusion structure extracts advanced features from image and radar respectively through a series of blocks (final resolution is $1/32$ of the input), then concatenates and fuses them. In the early fusion structure, the image and radar feature is extracted separately to $1/4$ of the resolution of the input, then concatenate and extract advanced feature. As given in Table 3, the late fusion structure outperforms the early fusion structure. We believe that this is due to the fact that radar data is highly sparse and if the early fusion structure is used, most of the locations in the feature map are invalid at an early stage, resulting in the model failing to extract valid information.

Modules: We gradually add the sparse pre-mapping module (SPM) and feature fusion module (FFM) to the baseline to investigate the effectiveness of our proposed network. As can be seen in Table 3, the addition of the two modules improves the RMSE of the baseline by 0.535m and the MAE by 0.374m. To further verify the effectiveness of the sparse pre-mapping module in different fusion modes, we also deploy the module in the early fusion structure. This also means that for sparse radar data, sparsity-invariant convolution is more suitable for the initial extraction of its features.

Introduction of Radar: In order to verify the validation of radar optimizing effect on monocular depth estimation, we compare the model with only the image

Table 3. Comparisons of performance under different model structures. In this table, EF for early fusion structure, LF for late fusion structure, SPM for sparse pre-mapping module, FFM for feature fusion module, R-18(34) for the baseline of ResNet-18(34), Only Image means the input is an only image.

Variant	RMSE	MAE	REL	δ_1
LF(R-18)	6.027	2.761	0.126	0.862
LF(R-34)	5.906	2.676	0.121	0.870
EF	6.725	3.018	0.125	0.855
EF+SPM	6.510	2.900	0.125	0.861
LF+SPM	5.838	2.639	0.123	0.873
LF+FFM	5.815	2.627	0.118	0.874
LF+SPM+FFM	**5.371**	**2.302**	**0.103**	**0.897**
Only Image	5.877	2.637	0.117	0.877

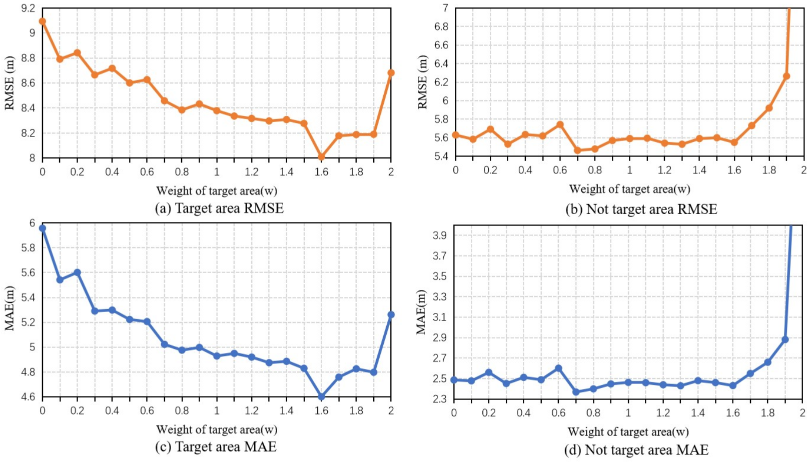
as input to our full model. The results are in agreement with our assumption that the introduction of radar does reduce the error in depth estimation compared to using only the image as input, implying the role of known depths at very few image locations in improving the performance of monocular depth estimation.

Loss Function: In Table 4, we compare the effect of four conventional loss functions on the model, i.e., $L1$, $L2$, $LogL1$ and $BerHu$ loss, from which it can be seen that the best prediction results are obtained with $L1$ loss.

Table 4. Comparisons of performance under different loss functions.

Variant	RMSE	MAE	REL	δ_1
$LogL1$	5.854	2.560	0.110	0.886
$L2$	5.857	2.922	0.137	0.842
$BerHu$	6.263	3.345	0.172	0.807
$\boldsymbol{L1}$	**5.371**	**2.302**	**0.103**	**0.894**

Hyperparameters: We conduct a series of experiments for the parameter ω in Eq. 7 and the results obtained are shown in Fig. 7. Increasing the parameter ω can be interpreted as increasing the weight of the target area to make the network more inclined towards the target area. From the experimental results, it can be seen that the RMSE and MAE of the target area gradually decrease as ω increases. In contrast, when reducing the parameter ω within a certain range, for example, $\omega = 1$ to $\omega = 0$, RMSE and MAE for the not target area vary within 0.2 m. This is because the model already favors the not target area and continuing to increase the weight of the not target area on a small scale does not have a significant impact on the depth estimation results. When the parameter ω is set to 1.6, the error in the target area reaches its minimum value and we believe that a good balance between the target area and not target area is achieved. At a more extreme, when setting $\omega = 2.0$, i.e. training the network based on the loss of the target area only, the error increases instead, which we believe is due to the fact that there is some connection between the target area and not target area, which complements each other. The accuracy of its own depth estimation also suffers when the loss of the other side is completely ignored.



(a) Target area RMSE

(b) Not target area RMSE

(c) Target area MAE

(d) Not target area MAE

Fig. 7. Parameter experiments with ω. Roughly as the weight of the target area increases, the error in the target area decreases. When not target area is completely ignored, the target area error also increases, indicating that the depth estimates of the two regions are correlated.

7 Conclusion

In this paper, we propose a two-stage depth estimation model more suitable for autonomous driving by introducing radar and driving scenario semantics. We also propose a method for generating dense depth data to ensure that the model can learn better. The validity of our model has been demonstrated through extensive experiments, and its depth estimation results outperform existing methods. We have made some attempts at fusing radar and image for depth estimation and have demonstrated that highly sparse radar point cloud can indeed provide prior information that can improve overall depth estimation performance. In future work, our group will continue to investigate more efficient methods of encoding sparse radar data and methods of fusing radar and image. Also, lightweight radar and image-based networks for deployment in autonomous driving systems are the next steps in our work.

Acknowledgement This research was funded by the Key R&D Projects of Science & Technology Department of Sichuan Province of China under Grant 2021YFG0070.

References

1. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4009–4018 (2021)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
3. Cao, Y., Wu, Z., Shen, C.: Estimating depth from monocular images as classification using deep fully convolutional residual networks. IEEE Transactions on Circuits and Systems for Video Technology **28**(11), 3174–3182 (2017)
4. Chen, Y., Yang, B., Liang, M., Urtasun, R.: Learning joint 2d-3d representations for depth completion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10023–10032 (2019)
5. Chen, Y., Zhao, H., Hu, Z., Peng, J.: Attention-based context aggregation network for monocular depth estimation. International Journal of Machine Learning and Cybernetics **12**(6), 1583–1596 (2021)
6. Cheng, X., Wang, P., Guan, C., Yang, R.: Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10615–10622 (2020)
7. Cheng, X., Wang, P., Yang, R.: Depth estimation via affinity learned with convolutional spatial propagation network. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 103–119 (2018)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE international conference on computer vision. pp. 2650–2658 (2015)
10. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. Advances in neural information processing systems **27** (2014)
11. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2002–2011 (2018)
12. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 270–279 (2017)
13. Gurram, A., Urfalioglu, O., Halfaoui, I., Bouzaraa, F., López, A.M.: Monocular depth estimation by learning from heterogeneous datasets. In: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 2176–2181. IEEE (2018)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Penet: Towards precise and efficient image guided depth completion. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 13656–13662. IEEE (2021)

16. Jiao, J., Cao, Y., Song, Y., Lau, R.: Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In: Proceedings of the European conference on computer vision (ECCV). pp. 53–69 (2018)
17. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth international conference on 3D vision (3DV). pp. 239–248. IEEE (2016)
18. Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H.: From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326 (2019)
19. Li, B., Shen, C., Dai, Y., Van Den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1119–1127 (2015)
20. Li, R., Xian, K., Shen, C., Cao, Z., Lu, H., Hang, L.: Deep attention-based classification network for robust depth prediction. In: Asian Conference on Computer Vision. pp. 663–678. Springer (2018)
21. Lin, J.T., Dai, D., Van Gool, L.: Depth estimation from monocular images and sparse radar data. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10233–10240. IEEE (2020)
22. Ma, F., Cavalheiro, G.V., Karaman, S.: Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 3288–3295. IEEE (2019)
23. Ma, F., Karaman, S.: Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In: 2018 IEEE international conference on robotics and automation (ICRA). pp. 4796–4803. IEEE (2018)
24. Qiu, J., Cui, Z., Zhang, Y., Zhang, X., Liu, S., Zeng, B., Pollefeys, M.: Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3313–3322 (2019)
25. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE transactions on pattern analysis and machine intelligence **31**(5), 824–840 (2008)
26. Tang, J., Tian, F.P., Feng, W., Li, J., Tan, P.: Learning guided convolutional network for depth completion. IEEE Transactions on Image Processing **30**, 1116–1129 (2020)
27. Torralba, A., Oliva, A.: Depth estimation from image structure. IEEE Transactions on pattern analysis and machine intelligence **24**(9), 1226–1238 (2002)
28. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: 2017 international conference on 3D Vision (3DV). pp. 11–20. IEEE (2017)
29. Van Gansbeke, W., Neven, D., De Brabandere, B., Van Gool, L.: Sparse and noisy lidar completion with rgb guidance and uncertainty. In: 2019 16th international conference on machine vision applications (MVA). pp. 1–6. IEEE (2019)
30. Wang, L., Zhang, J., Wang, O., Lin, Z., Lu, H.: Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 541–550 (2020)
31. Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., Yuille, A.L.: Towards unified depth and semantic prediction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2800–2809 (2015)

32. Xu, Y., Zhu, X., Shi, J., Zhang, G., Bao, H., Li, H.: Depth completion from sparse lidar data with depth-normal constraints. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2811–2820 (2019)
33. Xu, Z., Yin, H., Yao, J.: Deformable spatial propagation networks for depth completion. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 913–917. IEEE (2020)
34. Ye, X., Chen, S., Xu, R.: Dpnet: Detail-preserving network for high quality monocular depth estimation. Pattern Recognition **109**, 107578 (2021)
35. Yin, W., Liu, Y., Shen, C., Yan, Y.: Enforcing geometric constraints of virtual normal for depth prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5684–5693 (2019)