Multiscale Adaptation Fusion Networks for Depth Completion

Yongchi Zhang, Ping Wei*, Huan Li, Nanning Zheng
Xi'an Jiaotong University, Xi'an, China
pingwei@xjtu.edu.cn

***Abstract*—Depth completion is becoming a particularly important yet challenging problem with the growingly rapid progress of depth sensing technologies. Depth completion aims to complete sparse and noisy depth images to generate dense depth images. In this paper, we propose a multiscale adaptation fusion network (MAFN) for depth completion. The depth features are fused with RGB features at multiple scales with adaptation modules, where a neighbour attention mechanism is designed to adapt the local structures of the RGB image and the depth image. The fusion and completion process are unified under the encoder-decoder framework which is learned in an end-to-end way. By exploiting the detailed structural relationships of RGB images and depth images, our MAFN model can accurately complete and restore the invalid depth values on the sparse depth images. We test the proposed method on the challenging KITTI depth completion benchmark. The experimental results prove the effectiveness and strength of the proposed method.**

***Index Terms*—depth completion, adaptation fusion, neighbour attention, neural network**

I. Introduction

With the rapid development of depth sensing technologies such as ToF, LiDAR, and RADAR, depth data is playing increasingly important roles in myriads of applications such as automatic vehicles and intelligent robots. Depth data could provide precise and reliable range and geometry information for surrounding environments. However, limited by the technology bottleneck of depth sensors (such as scan discontinuity in LiDAR) and the influence of natural scene conditions (such as strong surface reflection of objects), the captured depth images are often very noisy and depth values of many pixels are missing. These factors make the depth data inaccurate and unreliable. In the public KITTI depth dataset [1], the sampling points with valid depth values usually only account for a very small portion of all the pixels in the LiDAR depth image. This would lead to huge errors in applications due to the lack of complete depth information. Thus, it is of great importance to repair the sparse depth images.

Given a sparse depth image with invalid depth values such as noise and holes, depth completion aims to generate a dense depth image by completing and recovering the invalid depth values, as shown in Fig. 1. Some previous studies take a sparse depth image as input and adopt traditional filtering methods [2] or neural network models [1], [3] to estimate the missing depth values. However, sparse depth images themselves are usually insufficient to recover the missing depth values due to wide-range discontinuity or large black holes. With the development of new depth sensors which can capture depth images and the corresponding RGB images simultaneously, most recent approaches [4]–[6] integrate depth images and RGB images for depth completion, which has been demonstrated to perform better than the approaches with depth images only. Actually, RGB images provide rich information such as semantics, geometric structures, neighbour relationships, boundaries, and region discontinuity, which serve as hidden priors to guide the recovery of depth images. Thus, in this paper, we use both depth images and RGB images for depth completion.

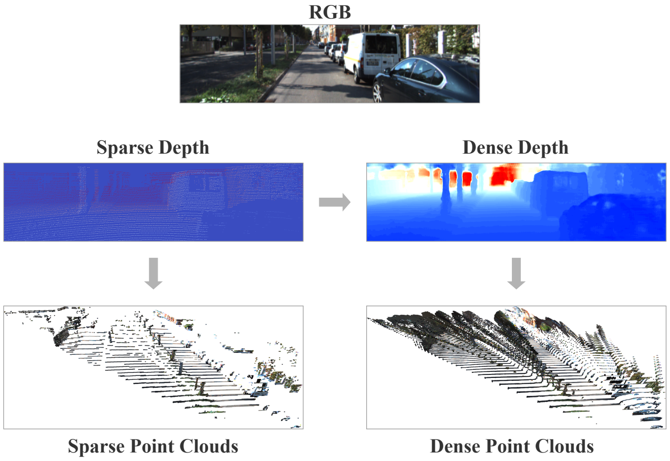

Fig. 1. The illustration of depth completion. Our network takes a sparse depth image and a corresponding RGB image as inputs and outputs a dense depth image. The RGB image provides abundant information in local regions for guiding the depth completion. After depth completion, the point clouds become dense.

While the previous studies have made considerable progress in integrating RGB and depth information for depth completion [5], [7], [8], most of them combine RGB and depth features by simple addition or concatenation. Such fusion obscures the corresponding relations between RGB features and depth features and cannot effectively utilize the hidden local structure information, which therefore may lead to unsatisfactory results. Though RGB features are beneficial to guide the depth value recovery, how to integrate RGB features and depth features should be learned from data rather than simple concatenation or addition.

In this paper, we propose a multiscale adaptation fusion

Ping Wei is the corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

network (MAFN) to integrate depth and RGB features for depth completion. Different from previous approaches simply concatenating or adding the features, our MAFN model integrates depth and RGB features by multiscale adaptation modules, where the fusion of the two type of features is learned from data. Inspired by the observation that the essence for depth completion is to infer the missing values from the surrounding neighbours, in each adaptation module a neighbour attention mechanism is adopted to enhance the adaptation of RGB features and depth features in neighbourhoods. The fusion and completion process are unified under the encoder-decoder framework which is learned in an end-to-end way. Our model is trained and validated on the KITTI depth dataset [1], which is one of the representative public datasets of outdoor scenes for depth completion. The results of comprehensive experiments demonstrate effectiveness and strength of the proposed method.

II. Related Work

We review the related work of depth completion from the following streams of research.

A. Depth Estimation

Depth estimation aims to directly infer a dense depth map only from an RGB image by learning a mapping relationship from RGB images to depth images. Actually, limited by the acquirement of depth sensors, obtaining a dense depth image from an RGB image is a simple and direct approach. Early studies [9], [10] are mainly based on hand-crafted feature extraction. When neural network models show great ability in image processing, most current studies have shifted their attention to the combination with neural networks. Liu *et al.* [11] proposed a deep convolution neural field model based on CNN and CRF. Chen *et al.* [12] designed a structure-aware residual pyramid network to learn multi-scale structure features. Xu *et al.* [13] introduced a continuous CRF to combine multi-scale features extracted by convolutional neural networks with structured attention modules. Cheng *et al.* [7] proposed a convolutional spatial propagation network to learn affinity matrices.

Depth estimation is a different task from depth completion but can provide much inspiration for depth completion. Depth estimation methods are often limited by manually defined scene constraints and therefore the estimated depth values are not always accurate.

B. Depth Completion

Compared with depth estimation, depth completion seeks to generate a dense depth map mainly from a sparse depth map, i.e. from 'sparse' to 'dense'. Some depth completion approaches only take depth images as inputs while more studies try to add additional RGB images as guidances to predict the dense depth images. Some early studies [2], [14], [15] exploited hand-crafted features to produce a dense depth map from a sparse depth map. With the advancement of neural network models, Uhrig *et al.* [1] proposed an efficient sparsity invariant convolution model which takes the locations of the missing values into consideration. In current studies, RGB images are proved to be of great help for depth completion, as RGB features can provide additional detailed semantic information. Zhang *et al.* [16] proposed to predict surface normals and occlusion boundaries, and then used a global optimization for depth completion in indoor scenes. In outdoor scenes, Eldesokey *et al.* [6] proposed a normalized convolution operation to deal with irregular and sparse depth data and propagate confidence through CNNs. Ma *et al.* [17] designed a deep regression framework by self-supervised learning without ground-truth depth image. Huang *et al.* [4] proposed three sparsity-invariant operations to utilize multi-scale features for handling the sparse data. Jaritz *et al.* [5] promoted the depth completion by combining it with semantic segmentation task. Yang *et al.* [18] presented a system to infer the posterior distribution of a dense depth map for depth completion. Qiu *et al.* [19] proposed an encoder-decoder framework which uses surface normal for depth completion.

These studies have achieved much progress in depth completion. However, how to effectively learn the relations between RGB and depth features remains an open problem, which inspires us to pursue new frameworks for depth completion.

C. Data Fusion

The ways of depth and RGB feature fusion for depth completion mainly consist of early fusion [7], late fusion [5], [8], and multi-level fusion [20]. Early fusion means the RGB image and depth map are combined at the initial stage, and then jointly sent to neural networks for extracting features. The late fusion refers to combining the RGB features and depth features at the end stage. Multi-level fusion means feature extraction and combination alternate at multiple levels. Cheng *et al.* [7] proposed to learn the affinity among neighboring pixels using CNN with early feature fusion of RGB and depth. On the other hand, some other studies prove that late fusion can achieve better performance than early fusion. The work [5], [8] adopted a late feature fusion scheme, which uses two streams to extract RGB and depth features separately in the encoder stage, and then combines them to perform upsampling operation in the decoder stage. Wang *et al.* [20] integrated multi-scale RGBD features for depth completion, while Hu *et al.* [21] aggregated RGB and depth features at multi level for semantic segmentation.

One major drawback of these methods is that they integrate RGB and depth features by simple addition or concatenation, which cannot effectively utilize the hidden local structure information. Inspired by these approaches, we propose a multiscale adaptation fusion network, which learns from data to preferably combine the features of RGB and depth at multiple scales.

III. Model

In this section, we introduce the proposed multiscale adaptation fusion network model.

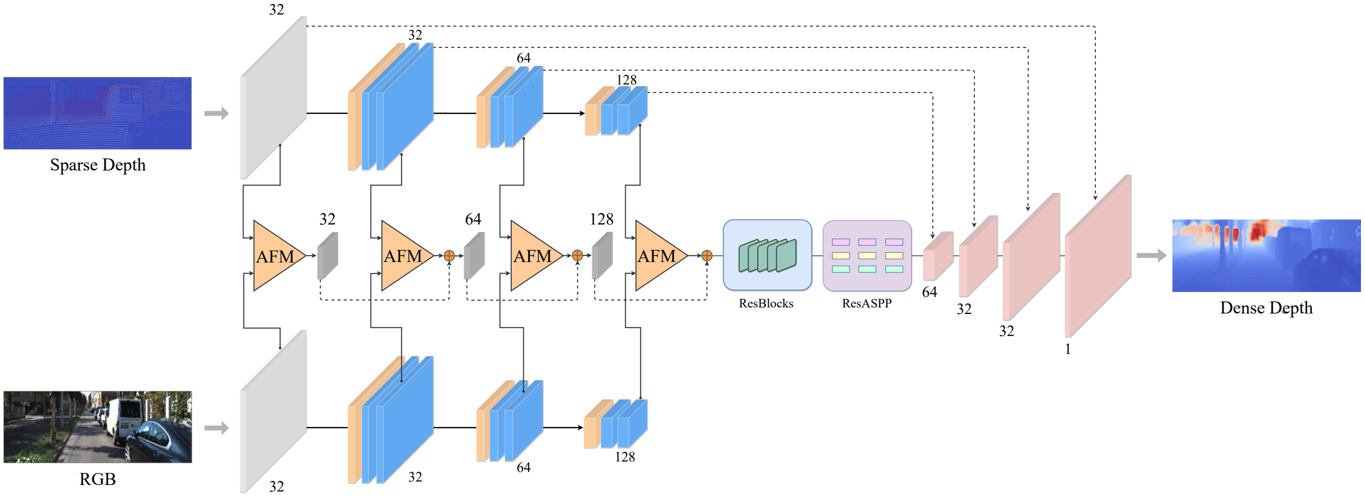

Fig. 2. The proposed multiscale adaptation fusion network (MAFN). The network takes a sparse depth image and a corresponding RGB image as inputs, and predicts a dense depth image. The features are integrated at multiple scales via the proposed adaptation fusion modules (AFM).

A. *Network Architecture*

Fig. 2 shows the general architecture of our multiscale adaptation fusion network (MAFN). The inputs of the model is the sparse depth image to be recovered and the corresponding RGB image. The output is the recovered dense depth image. As shown in Fig. 2, from the left to right, the general structure of our MAFN is under an encoder-decoder framework. The encoder aims to extract and integrate RGB and depth features, and the decoder seeks to recover the output depth image from the fusion features.

The encoder structure mainly consists of two paralleled streams which address the sparse depth image and the RGB image respectively. The two streams share identical structures but have different parameters. Each stream is composed of four convolution blocks where the first block has one convolutional layer with 32 convolutional kernels (kernel size 3 × 3 and stride 1). The other three convolution blocks contain three convlutional layers where the stride is 2 in the first layer and 1 in the other two layers. The sizes of all the kernels are all 3 × 3 and the kernel number in the three blocks are 32, 64, and 128 respectively.

Between the RGB and the depth streams, four adaptation fusion modules (AFMs) connect the two streams and integrate the features at different convolution scales, as the triangle blocks shown in Fig. 2. After each convolution block, the features separately extracted in the RGB stream and the depth stream are input into the adaptation fusion module. The fused features output from AFM pass through a transition layer to produce new features which are then added to the outputs of the next AFM. The transition layer performs the convolution operations with the kernel size 3×3 and the stride 2. The kernel numbers of the three transition layers at different scales are 32, 64, 128, respectively. It should be noted that the feature fusion with AFM is not the simple concatenation or addition of different features. In Section III-B, we will introduce the inner structures of the adaptation fusion modules in detail.

The final fusion features output from the four AFMs are fed to ResBlocks module which consists of five cascaded residual blocks [22] for further deepening the features without losing the resolution. The features output from ResBlocks module are input into the residual atrous spatial pyramid pooling (ResASPP) [23] module, which is used for learning combination of the features with different receptive fields. The ResASPP consists of three ASPP groups, each of which contains three dilated convolutions with dilation rate of 1, 4, 8, and then these three groups are added in a cascading manner.

In the decoder, the feature maps are upsampled three times with four deconvolution layers to output the ultimate dense depth map. The kernel number of the four deconvolution layers are 64, 32, 32, and 1 respectively. For better feature fusion and upsampling, we also add the skip connections by concatenating the features of the depth stream in the encoder with the corresponding one of the decoder.

B. *Adaptation Fusion Module*

In order to integrate the RGB and depth features, we design adaptation fusion modules (AFMs) which connect RGB and depth streams at different scales, as the triangle blocks AFM shown in Fig. 2. The inside structure of one AFM is shown in Fig. 3. An AFM receives the features from the RGB stream and the depth stream respectively. Inside the AFM, the features from the two streams separately pass through the neighbour attention module (NAM) to extract local neighbouring relational information among pixels. After the NAM, the RGB and depth features are concatenated to feed two cascaded convolution blocks and output the fusion features, as Conv4 and Conv5 shown in Fig. 3. The Conv4 is an one- layer convolution with kernel size 1 × 1. The Conv5 is a residual convolution layer [22].

The essence of depth completion is to infer the missing depth values from its neighboring pixels. The RGB image provide rich semantic information of local region relationships.

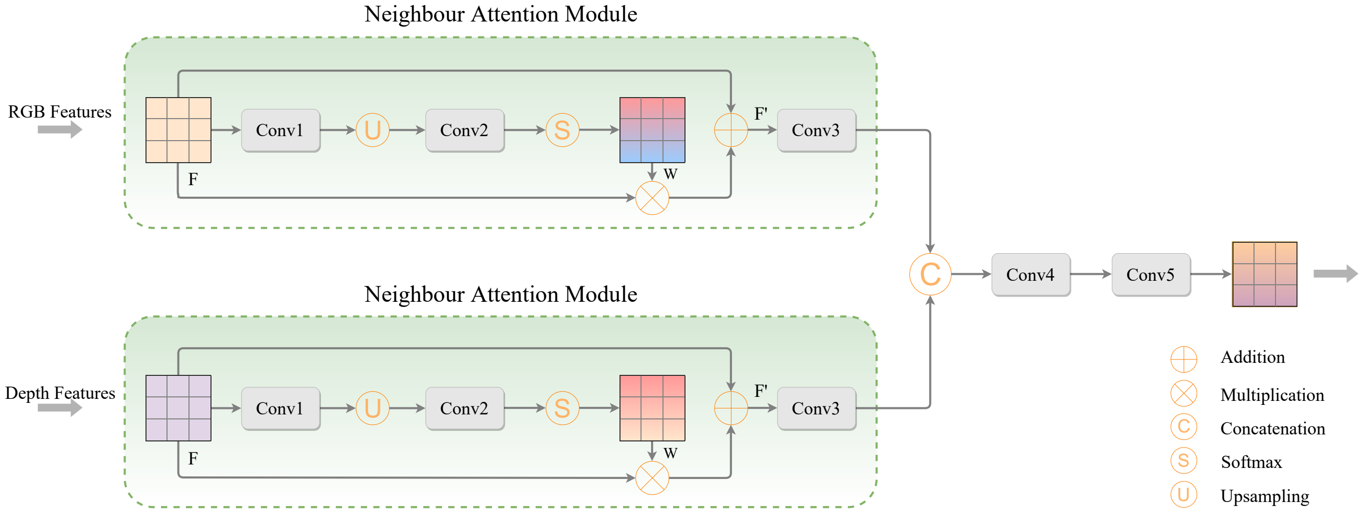

Fig. 3. The proposed adaptation fusion module (AFM). The features from RGB stream and depth stream are input into neighbor attention modules respectively. Then the features are concatenated and fed into two convolution blocks which output the fusion features.

Therefore we design two paralleled neighbor attention modules to combine the information of RGB and depth in local regions.

As shown in Fig. 3, suppose F denotes the feature map of RGB stream or depth stream that is input to the adaptation fusion module. The feature map F is downsampled to half the size of the original resolution by a general convolution block with a kernel size 3×3 and a stride size 2, as Conv1 shown in Fig. 3. Then in order to exploit the relationship of pixels in local regions, we use the nearest neighbor upsampling, enlarging the feature map to the same size as the original feature input, as 'U' shown in Fig. 3. Next, we add a 1×1 convolution to perform information integration across channels, as Conv2 shown in Fig. 3. A softmax activation function is applied to the output of the Conv2 module and produces the ultimate weighted map W with values between 0 and 1. Finally, a pixel-wise multiplication is conducted for the input feature map and the weight map, whose result is added to the input feature map to produce the weighted feature vector F'. This process is expressed as:

$$F' = F + W \otimes F, \tag{1}$$

where $\otimes$ denotes the element-wise multiplication.

The weighted feature map F' is fed to a convolution module Conv3 to produce the features output from the neighbour attention module. The convolution module Conv3 has a 3×3 kernel with stride 1.

C. Loss Function

We use the Mean Squared Error (MSE) to calculate the loss between the predicted dense depth image and the ground truth depth image. The MSE loss function is defined as

$$L = \frac{1}{|m|} \sum_{i,j \subset m} ||D^{p}_{i,j} - D^{gt}_{i,j}||^2, \tag{2}$$

where $D^{p}_{i,j}$ and $D^{gt}_{i,j}$ represent the predicted depth value and the corresponding ground truth value at location (i, j) respectively. m is the set of valid pixels in ground truth and $|m|$ denotes the number of the set elements. In practice, the ground truth depth map for depth completion is semi-dense. Thus following the standard evaluation metrics in other work [1], we compute the valid pixels in ground truth.

IV. Experiments

A. Dataset and Setting

Dataset. We test the proposed method on the challenging KITTI depth completion dataset [1]. It is one of the most challenging public evaluation datasets for depth completion, containing 85898 frames for training, 6852 frames for validation, 1000 frames for evaluation, and 1000 for test on online server, with depth images and corresponding RGB images. The ground truth depth values are created by projecting LiDAR points into the image plane and accumulating 11 LiDAR scans from frames around the current frame. Following other studies, we crop the size of all data (includes RGB images, sparse depth maps, ground truth) to the resolution of 256×1216 during training. Furthermore, we horizontally flip all inputs at random for data augmentation.

Evaluation metrics. We adopt four metrics to evaluate and compare different approaches: root mean square error (RMSE), mean absolute error (MAE), root mean squared error of the inverse depth (iRMSE), and mean absolute error of the inverse depth (iMAE). The metric units of RMSE, MAE, iRMSE, and iMAE are millimeter (mm), millimeter (mm), 1/kilometre (km), and 1/kilometre (km), respectively. Since RMSE is the decisive evaluation metric for ranking all methods submitted on KITTI depth completion benchmark [1], we mainly use it to compare our approach with others. Other evaluation metrics are used as references.

Implementation details. Specifically, two GTX 2080Ti GPUs are used for training with batch size of 8. We adopt

TABLE I
COMPARISON WITH THE OTHER METHODS ON THE TEST SET OF KITTI DEPTH COMPLETION DATASET [1]. NOTE: THE ROOT MEAN SQUARE ERROR(RMSE) IS THE DECISIVE METRIC RANKING ALL METHODS ON THE KITTI DEPTH COMPLETION BENCHMARK.

Method	**RMSE** (mm)	MAE (mm)	iRMSE (1/km)	iMAE (1/km)
SparseConv [1]	1601.33	481.27	4.94	1.78
IP-Basic [2]	1288.46	302.60	3.78	1.29
NConv-CNN [6]	1268.22	360.28	4.67	1.52
Spade-sD [5]	1035.29	248.32	2.60	0.98
ADNN [3]	1325.37	439.48	59.39	3.19
DFuseNet [8]	1206.66	429.93	3.62	1.79
CSPN [7]	1019.64	279.46	2.93	1.15
Spade-RGBsD [5]	917.64	234.81	2.17	0.95
NConv-CNN-L2 [6]	829.98	233.26	2.60	1.03
DDP [18]	832.94	203.96	2.10	0.85
Sparse-to-Dense [17]	814.73	249.95	2.80	1.21
HMS-Net [4]	841.78	253.47	2.73	1.13
Ours	**803.50**	279.37	3.02	1.48

Adam [24] as the optimizer with an initial learning rate of 10^{-3} which is decayed to 10^{-4} at 20, weight decay of 10^{-6} and betas (a coefficient used to calculate the average value of the gradient run and its square) of (0.9, 0.999). We also add an Instance Normalization [25] layer after each convolutional layer.

B. Comparison with State-of-Art

We perform overall comparison with other methods on KITTI depth completion benchmark [1]. In order to avoid overfitting, the ground truth of the test set are not provided and all the results should be submitted to an online server for evaluation[1]. The quantitative comparative results of our proposed method with other previous related approaches are listed in Table I. In this table, the first five rows of methods only use depth images for depth completion and other methods use both depth images and RGB images.

On this dataset, our approach achieves an RMSE of 803.50, which outperforms other methods by a large margin. Different from other methods which simply add or concatenate the RGB and depth features, our method integrates these two types of features at multiple scales in an adaptation fusion way, which makes our method achieve a better performance.

Some visual qualitative comparison with some methods are shown in Fig. 4. As the regions of the dashed line boxes shown in this figure, it is clear that our approach maintains better details in some local regions of the depth maps. Furthermore, this figure also shows that our method can reconstruct the missing depth values better. The proposed method combine the multi-scale features with different receptive fields, which could capture multi-level semantic and detailed information.

TABLE II
QUANTITATIVE COMPARISON WITH DIFFERENT FUSION SCHEMES AND MODULES ON THE SELECTED VALIDATION SET OF KITTI DEPTH COMPLETION DATASET [1]. HERE 'W/O' MEANS 'WITHOUT' AND 'W' INDICATES 'WITH'.

Scheme	**RMSE** (mm)	MAE (mm)	iRMSE (1/km)	iMAE (1/km)
RGB Only	3430.61	1632.46	11.10	6.95
Depth Only	1041.02	344.08	4.09	1.78
Early Fusion	918.91	308.71	3.47	1.61
Late Fusion	851.98	283.30	3.02	1.43
Multiscale Fusion	863.41	290.45	3.18	1.49
AFM w/o NAM	832.94	278.62	3.00	1.39
AFM w NAM	826.72	269.86	2.99	1.36

C. Ablation Study

Since the ground truth of KITTI [1] test set are not provided, to validate the effectiveness of our designed modules, we conduct ablation studies with the selected validation set on the KITTI dataset. We compare different fusion schemes and modules as follows. The ablation study results are shown in Table II.

1) **RGB Only**. We consider the way to directly predict a dense map only from RGB images. As shown in Table II, the results are with huge errors. Actually, it is ambiguous to generate a depth value because the RGB image cannot provide precise 3D information in the world coordinates.
2) **Depth Only.** This method only takes sparse depth maps as inputs without the corresponding RGB images. The RMSE on the selected validation set is more than 1000.0, which indicates that it is very hard to restore depth values only relying on a sparse depth map without the guidance of the RGB information.
3) **Early Fusion.** We concatenate the sparse depth map (1 channel) and the RGB image (3 channels) as inputs (4 channels), which are then input into a single stream network. The performance is improved slightly compared to the RGB Only method and the Depth Only method.
4) **Late Fusion.** Two paralleled streams extract RGB and depth features respectively, which are then added before the upsampling operation. It performs better than the above three ablation methods but still much lower than the proposed method. Indeed, it just combines the high-level information, ignoring the utilization of low-level information. Consequently, it cannot achieve satisfactory results compared to the proposed method.
5) **Simple Multiscale Fusion.** We replace our adaptation module with simple addition operation at multiple scales. As shown in Table II, it has a lower performance than our proposed method with the adaptation fusion module, which proves the effectiveness of the adaptation fusion design.
6) **Adaptation Fusion without neighbor attention module (NAM).** We also validate the effectiveness of our neighbor attention design by excluding it from each adaptation fusion module. Compared with the proposed method with

[1] http://www.cvlibs.net/datasets/kitti/eval_depth.php

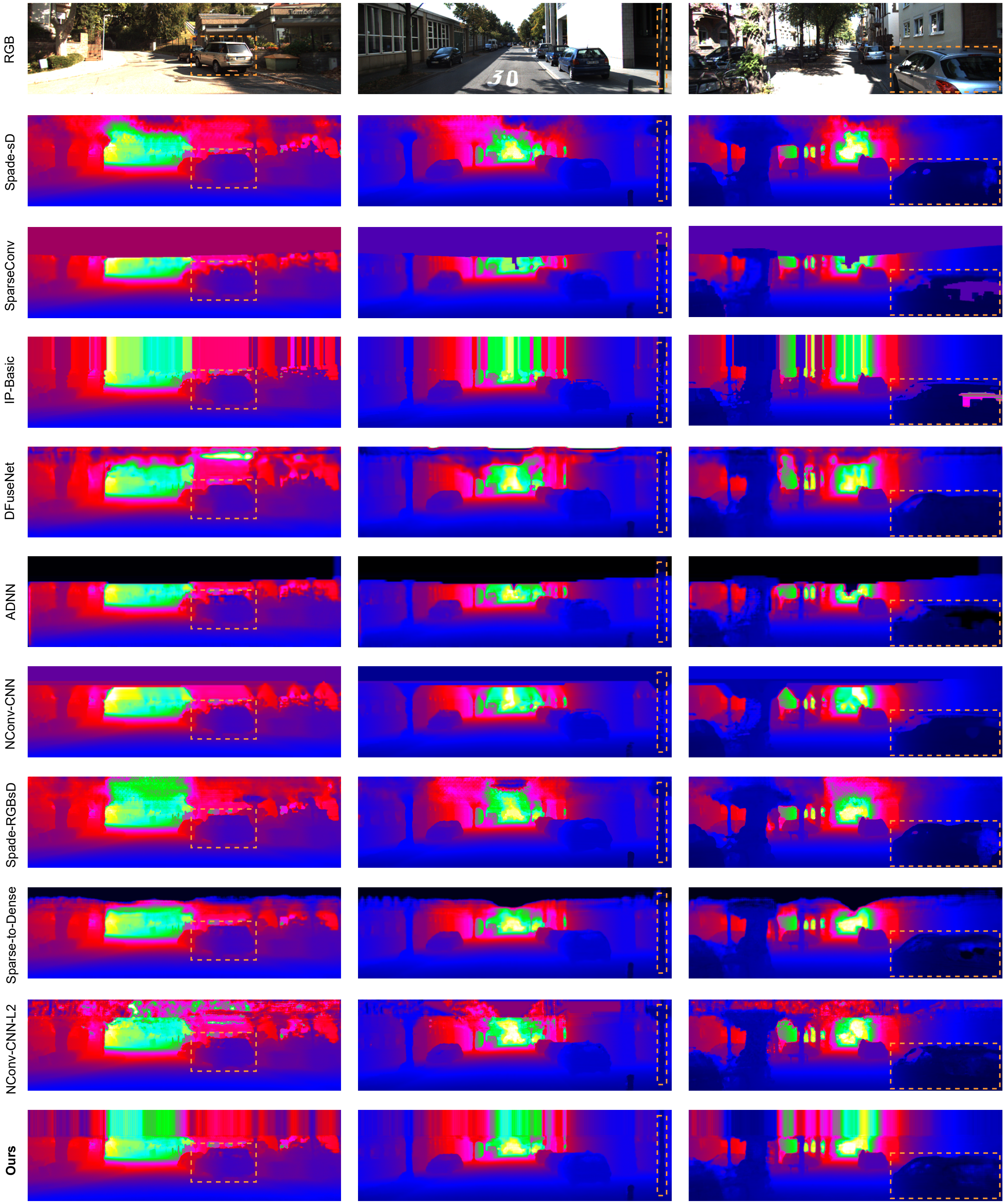

Fig. 4. Comparison of visual results with other methods on KITTI Depth dataset [1].Intuitively, the generated depth maps by our method have more detailed structures, such as some regions such as cars and poles marked with the dashed line boxes.

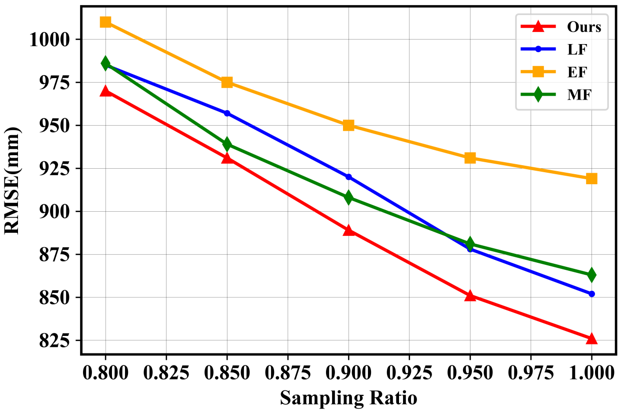

Fig. 5. The performance comparison of different fusion schemes at different levels of sparsity. EF, LF and MF refer to early fusion, late fusion, and simple multiscale fusion, respectively.

NAM, this method has a lower performance, which demonstrates the effectiveness of the neighbor attention design.

All these ablation study results show that the proposed MAFN framework is reasonable, effective, and advantageous.

Additionally, we compare the performance of different schemes at varying levels of sparsity, as shown in Fig. 5. We conduct a sub-sampling on the raw LiDAR depth map with the sparsity ratio of 0.80, 0.85, 0.90 and 0.95. The lower ratio means the depth image is more sparse. Under all these sparse conditions, our method performs better than other comparison approaches. It proves that our model has better generalization ability in the sparse situations.

V. Conclusion

In this paper, we propose a multiscale adaptation fusion network (MAFN) to combine the information of RGB and depth for depth completion. In order to infer the relational information among local pixels, we propose a neighbor attention to reason about depth values from neighbors. The fusion and completion process are unified under the encoder-decoder framework which is learned in an end-to-end way. By exploiting the detailed structural relationships of RGB images and depth images, our MAFN model can accurately complete and restore the invalid depth values on the sparse depth images. Extensive comparison and ablation experiments demonstrate that our proposed method is reasonable, effective, and advantageous.

Acknowledgment

This research was supported by the grants National Natural Science Foundation of China No. 61876149 and China Postdoctoral Science Foundation 2018M643657.

References

[1] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *International Conference on 3D Vision*, 2017, pp. 11–20.

[2] J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the CPU," in *Conference on Computer and Robot Vision*, 2018, pp. 16–22.

[3] N. Chodosh, C. Wang, and S. Lucey, "Deep convolutional compressed sensing for lidar depth completion," in *ACCV*, 2018, pp. 499–513.

[4] Z. Huang, J. Fan, S. Yi, X. Wang, and H. Li, "Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion," *CoRR*, vol. abs/1808.08685, 2018.

[5] M. Jaritz, R. de Charette, É. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with cnns: Depth completion and semantic segmentation," in *nternational Conference on 3D Vision*, 2018, pp. 52–60.

[6] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through cnns for guided sparse depth regression," *CoRR*, vol. abs/1811.01791, 2018.

[7] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *ECCV*, 2018, pp. 108–125.

[8] S. S. Shivakumar, T. Nguyen, I. D. Miller, S. W. Chen, V. Kumar, and C. J. Taylor, "Dfusenet: Deep fusion of RGB and sparse depth information for image guided dense depth completion," in *IEEE Intelligent Transportation Systems Conference*, 2019, pp. 13–20.

[9] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *CVPR*, 2014, pp. 89–96.

[10] A. Saxena, S. H. Chung, and A. Y. Ng, "3-d depth reconstruction from a single still image," *International Journal of Computer Vision*, vol. 76, no. 1, pp. 53–69, 2008.

[11] F. Liu, C. Shen, G. Lin, and I. D. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, 2016.

[12] X. Chen, X. Chen, and Z. Zha, "Structure-aware residual pyramid network for monocular depth estimation," in *International Joint Conference on Artificial Intelligence*, 2019, pp. 694–700.

[13] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *CVPR*, 2018, pp. 3917–3925.

[14] N.-E. Yang, Y.-G. Kim, and R.-H. Park, "Depth hole filling using the depth distribution of neighboring regions of depth holes in the kinect sensor," 08 2012, pp. 658–661.

[15] L. Liu, S. H. Chan, and T. Q. Nguyen, "Depth reconstruction from sparse samples: Representation, algorithm, and sampling," *IEEE Trans. Image Processing*, vol. 24, no. 6, pp. 1983–1996, 2015.

[16] Y. Zhang and T. A. Funkhouser, "Deep depth completion of a single RGB-D image," in *CVPR*, 2018, pp. 175–185.

[17] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *International Conference on Robotics and Automation*, 2019, pp. 3288–3295.

[18] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (DDP) from single image and sparse range," in *CVPR*, 2019, pp. 3353–3362.

[19] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, "Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[20] B. Wang, Y. Feng, and H. Liu, "Multi-scale features fusion from sparse lidar data and single image for depth completion," *Electronics Letters*, vol. 54, no. 24, pp. 1375–1377, 2018.

[21] X. Hu, K. Yang, L. Fei, and K. Wang, "ACNET: attention based network to exploit complementary features for RGBD semantic segmentation," in *IEEE International Conference on Image Processing*, 2019, pp. 1440–1444.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[23] L. Wang, Y. Wang, Z. Liang, Z. Lin, J. Yang, W. An, and Y. Guo, "Learning parallax attention for stereo image super-resolution," in *CVPR*, 2019, pp. 12 250–12 259.

[24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[25] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016.