IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 25, 2020, accepted April 11, 2020, date of publication April 24, 2020, date of current version May 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990212

Deep Architecture With Cross Guidance Between Single Image and Sparse LiDAR Data for Depth Completion

SIHAENG LEE[1], (Member, IEEE), JANGHYEON LEE[2], (Member, IEEE), DOYEON KIM[2], (Member, IEEE), AND JUNMO KIM[1,2], (Member, IEEE)
[1]Division of Future Vehicle , Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
[2]School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea

Corresponding author: Junmo Kim (junmo.kim@kaist.ac.kr)

This work was supported in part by the Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) under Grant 2016-0-00563, and in part by the Research on Adaptive Machine Learning Technology Development for Intelligent Autonomous Digital Companion and Hyundai Motor Company.

ABSTRACT It is challenging to apply depth maps generated from sparse laser scan data to computer vision tasks, such as robot vision and autonomous driving, because of the sparsity and noise in the data. To overcome this problem, depth completion tasks have been proposed to produce a dense depth map from sparse LiDAR data and a single RGB image. In this study, we developed a deep convolutional architecture with cross guidance for multi-modal feature fusion to compensate for the lack of representation power of their modality. Two encoders, which are part of the proposed architecture, receive different modalities as inputs. They interact with each other by exchanging information in each stage through the attention mechanism during encoding. We also propose a residual atrous spatial pyramid block, comprising multiple dilated convolutions with different dilation rates, which are used to derive highly significant features. The experimental results of the KITTI depth completion benchmark dataset demonstrate that the proposed architecture shows higher performance than that of the other models trained in a two-dimensional space without pre-training or fine-tuning other datasets.

INDEX TERMS Depth estimation, depth completion, LiDAR data, cross guidance, multi-scale dilated convolutional block.

I. INTRODUCTION

An accurate depth map with an RGB image allows users to utilize the information to solve complicated computer vision tasks. However, as shown in Figure 1, depth maps acquired from a LiDAR sensor have sparse structures. Therefore, they cannot be applied to autonomous driving or robotics applications. To use LiDAR depth maps, the missing pixels must be provided. To this end, *depth completion tasks* have been introduced by [1], [2]. An RGB image acquired from a camera and a sparse depth map acquired from a LiDAR sensor are used as inputs, and the output is the corresponding dense depth map, as shown in Figure 1. A precise depth map, which is useful as the prior information for processing an RGB image (*e.g.*, object classification, detection, and segmentation), is valuable in both academic and industrial research. However, for obtaining high accuracy, dense depth data is very expensive. Therefore, to improve the performance of dense depth maps produced from sparse data in the field of computer vision, the depth completion task remains to be solved.

There are several challenges in generating dense depth maps from sparse LiDAR data. The depth values of LiDAR data are spaced irregularly and sparsely. Therefore, constructing more accurate pixel-wise annotations of the ground truth is complicated and expensive. Furthermore, the LiDAR depth and RGB images that produce dense depth maps have different modalities, and the multi-modal feature fusion is still in its early stages of development. To address these problems, many studies have attempted to train artificial neural networks to be applied in depth completion tasks.

Recently, artificial neural network models with deep learning have been used in state-of-the-art technologies of pattern recognition and machine learning. In particular, convolutional neural networks (CNNs) exhibit excellent performance in many computer vision tasks. While conventional CNNs [3]–[5] comprise blocks of stacking convolution

The associate editor coordinating the review of this manuscript and approving it for publication was Bohui Wang.

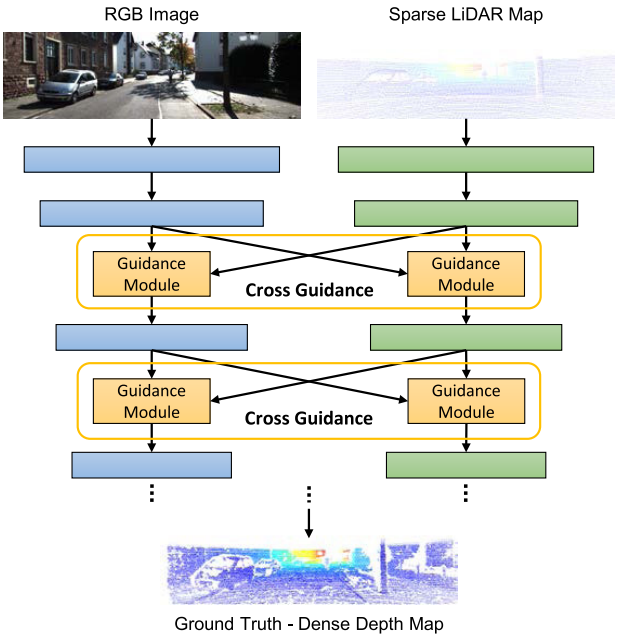

FIGURE 1. Illustration of proposed cross guidance. Two encoders that take different inputs, a single RGB image and a sparse LiDAR map, interact with their information in each stage to complement representation power.

layers, recent studies suggest specific techniques suitable for target tasks. For example, He *et al.* [6] proposed residual blocks with skip connection to construct a deeper architecture. GoogleNet [7] stacks the inception module, which comprises convolutional layers of different kernel sizes, to improve the representation power. For the segmentation task, many studies used atrous convolution, or dilated convolution [8]–[10]. Dilated convolution allows a convolution to increase its receptive field size without additional parameters or computation, while improving the performance. Another specifically designed technique is the attention mechanism. Its effectiveness has been extensively investigated in previous studies [11]–[17]. The attention mechanism derives specific areas that networks should focus on to improve the representation of feature maps.

Recent works [18]–[21] on depth completion tasks based on CNNs have utilized both RGB images and sparse LiDAR maps. These methods use simple operations, such as concatenation and element-wise sums to address multi-modal feature fusion. Most of these operations are performed at the beginning and end of the encoding or while decoding. In other words, there is no interaction between multi-modal features while encoding. Rather than using simple operations to fuse multiple modalities, our study focuses on managing the fusion of multiple modality features and suggests a more sophisticated fusion module that enables complex mapping for the depth completion task.

The present work is inspired by the abovementioned techniques to solve the depth completion task in outdoor scenarios. The contribution of this paper is threefold:

- We propose a cross-guidance method for combining features that have different modalities to compensate for the lack of representation power. At each stage in the encoder, the guidance module that employs the attention mechanism receives information from the feature maps of other modalities and merges it with their own information. We then evaluate the effectiveness of cross-guidance through ablation studies.
- A residual atrous spatial pyramid (RASP) block is proposed to analyze the large input dataset. This comprises multiple dilated convolutional layers with different dilation rates that work in parallel to create a wider receptive field and derive more significant features. In ImageNet-1K experiments, the RASP block shows improved performance against the widely used models in terms of the depth and the number of parameters.
- We verify the performance of the proposed architecture by achieving state-of-the-art results on the KITTI depth completion benchmark dataset in two-dimensional (2D) space without pre-training or fine-tuning.

II. RELATED WORK

A. DEPTH COMPLETION

The depth completion task is related to the sparse patterns of inputs. Depth maps acquired from LiDAR sensors have non-uniformly structured sparsity owing to their discrete polar scanning behavior. There are no depth data for most pixels when the LiDAR map is transformed to be aligned with an RGB camera [1]. Previous studies considered sparse inputs as an inpainting problem using classical image processing methods such as hand-crafted kernels or interpolation [22]–[25]. Recently, many approaches that employ deep learning and CNNs have shown successful results. To manage sparse inputs, Uhrig *et al.* [2] proposed a sparse convolution layer that explicitly considers the location of missing data by evaluating only the observed pixels and then normalizing the output appropriately. Cheng *et al.* [18] proposed a convolutional spatial propagation network that performs a recurrent convolution operation with an affinity matrix for interpolation with neighboring pixels. In [19], with a normalized convolution layer similar to the sparse convolution layer, the confidence of the convolution operation was also considered. Ma *et al.* [20] developed a deep regression model with an early fusion method for direct mapping from the sparse depth to the dense depth and a self-supervised training framework. The surface normal was introduced by [26], [27] to obtain more accurate three-dimensional (3D) geometric information for the depth completion task. The confidence mask for refining the results from the network was the same as in [19]. Van Gansbeke *et al.* [21] introduced a framework with global and local networks. A guidance map with the information extracted from the global network was used as the input to the local network, which improved the performance. A conditional prior network proposed by [28] calculated the depth posterior probability of the images of a large-scale synthetic dataset. Chen *et al.* [29] used projected 2D LiDAR maps and 3D point clouds to learn the 2D and 3D representations

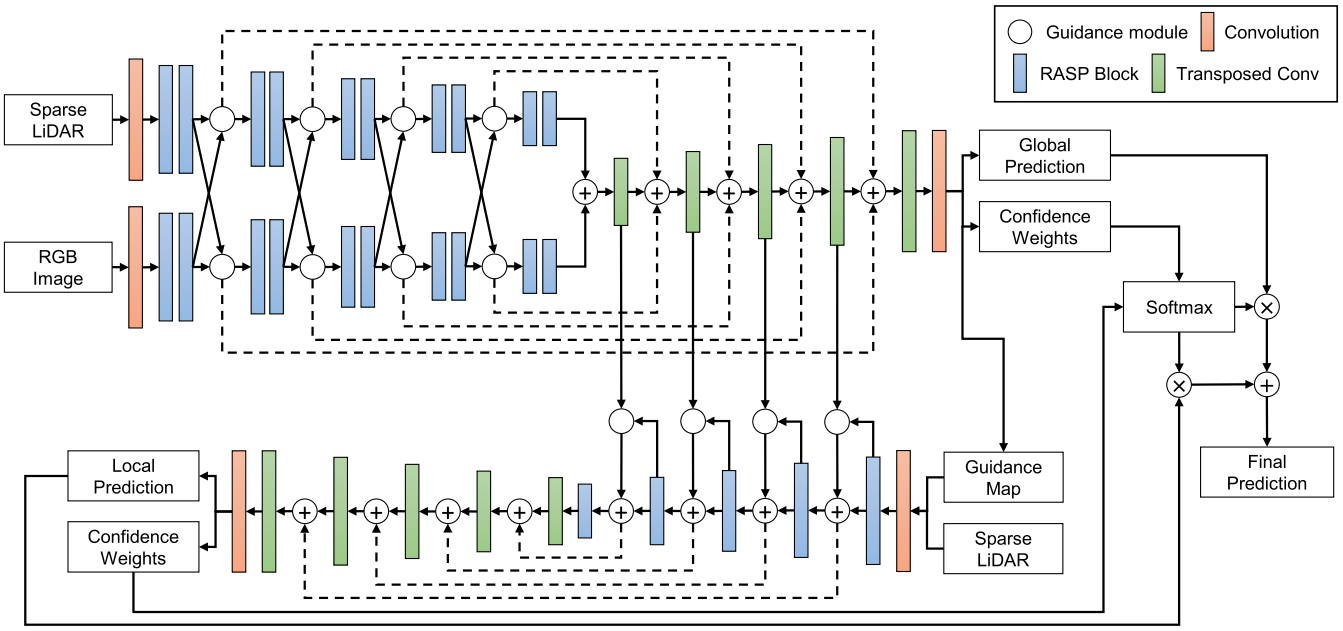

FIGURE 2. Illustration of the proposed architecture. The proposed architecture is an end-to-end framework that has two sub-networks, namely, the global (top) and local (bottom) networks. The final prediction of the dense depth map is the weighted sum of the two dense depth maps from the networks with score weights input to the softmax layer.

simultaneously through 2D-3D fusion blocks. These blocks use the information in 2D and 3D spaces at the same stage.

B. ATTENTION MECHANISM

Several attention methods have been proposed to improve the performance of CNNs in computer vision tasks by amplifying the important features and attenuating unnecessary ones. Hu *et al.* [12] introduced a squeeze-and-excitation block that adaptively recalibrates channel-wise feature responses with channel-wise attention from the global average-pooled features. Besides channel-wise attention, a convolutional block attention module (CBAM) [16] sequentially infers attention maps along both channel and spatial dimensions. Then, the attention maps are multiplied by the input feature map for adaptive feature refinement. CBAM also uses average-pooled and max-pooled features to obtain attention coefficients. Fu *et al.* [11] proposed a dual attention network for scene segmentation. It consists of two types of attention modules, which model the semantic interdependencies in spatial and channel dimensions using attention matrices. Wang *et al.* [15] presented a non-local block for capturing long-range dependencies, which computes the response at a position as a weighted sum of features at all positions. To achieve state-of-the-art results in image classification tasks using very deep networks, a residual attention network [30] comprising multiple attention modules and skip connections was proposed.

C. DILATED CONVOLUTION

An atrous convolution, or a dilated convolution, has recently been introduced for semantic segmentation [10]. This allows a convolution to increase its receptive field size without additional parameters or computation steps. As the sizes of the input and output images are considerably larger than those processed in a classification task, a network with atrous convolutions can process them more effectively with a reasonable amount of memory and within a certain amount of time. Chen *et al.* [8] proposed a specific architecture called the atrous spatial pyramid pooling (ASPP) block, which consists of multiple convolutions with different dilation rates arranged in parallel, thereby summing all the outputs, similar to an inception architecture [7]. The ASPP block can provide features with a range of scales from a single-feature map without any changes in the spatial size or an increase in the filter size. The ASPP is applicable to various semantic segmentation datasets; *e.g.*, an ASPP block can be added to the top of VGG-16 [7] and ResNet-101 [6] in [8]. Mehta *et al.* [31] proposed ESPNet, which is a fast and efficient lightweight CNN for semantic segmentation.

III. METHODS

A. GUIDANCE MODULE

We propose a cross-guidance module, which is an attention method that combines the features of different modalities to compensate for the lack of representation power. While most attention methods are self-attention methods, our cross-guidance module refines a feature with attention weights obtained from its own feature and from other features.

As shown in Figure 3, the overall structure of our guidance module is similar to CBAM [16]. However, to refine an input feature F_I, *e.g.*, a feature from an RGB image, we also use a corresponding guidance feature F_G, *e.g.*, a feature from a LiDAR map, to compensate for the information not available

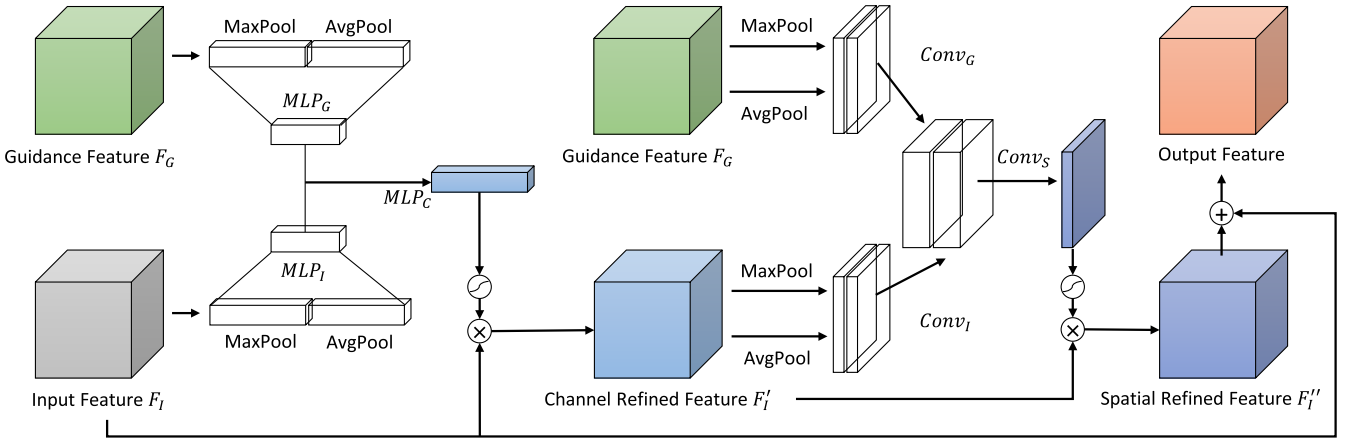

FIGURE 3. Guidance module. It refines an input feature with attention weights obtained from its own features and from other guidance features.

in F_I. We first perform channel-wise attention and then another step of attention-over spatial dimensions sequentially.

For the channel-wise attention, the spatial information of each feature is aggregated by average-pooling and max-pooling over spatial dimensions. Then, the average-pooled and max-pooled features are concatenated and passed through a multi-layer perceptron (MLP) with one hidden layer and ReLU activation to produce a channel feature vector.

$$F_{I,C} = MLP_I(\text{MaxAndAvgPool}_{\text{Spatial}}(F_I)) \quad (1)$$
$$F_{G,C} = MLP_G(\text{MaxAndAvgPool}_{\text{Spatial}}(F_G)) \quad (2)$$

A fully connected layer MLP_C takes the channel feature vectors $F_{I,C}$ and $F_{G,C}$ as inputs to produce a channel attention vector A_C, with the activation of the sigmoid function σ.

$$A_C = \sigma(MLP_C(F_{I,C}, F_{G,C})) \quad (3)$$

A_C is expanded over the spatial dimensions and multiplied by the original input feature F_I to obtain a channel refined feature F_I'.

In the second stage, *i.e.*, for the attention over spatial dimensions, a procedure similar to the one for channel-wise attention is performed. However, the roles of the spatial and channel axes are interchanged. To model a spatial feature map, each feature is average-pooled and max-pooled over channel dimensions and then passed through a BN-ReLU-Conv block with a kernel size of 7×7.

$$F_{I,S} = Conv_I(\text{MaxAndAvgPool}_{\text{Channel}}(F_I')) \quad (4)$$
$$F_{G,S} = Conv_G(\text{MaxAndAvgPool}_{\text{Channel}}(F_G)) \quad (5)$$

A convolutional layer $Conv_S$ aggregates the spatial feature maps $F_{I,S}$ and $F_{G,S}$ with sigmoid activation to produce a spatial attention map A_S.

$$A_S = \sigma(Conv_S(F_{I,S}, F_{G,S})) \quad (6)$$

A_S is expanded over the channel dimension and multiplied by the channel refined feature F_I' to obtain a spatial refined feature F_I''. Finally, F_I'' is added to the original input feature F_I by a skip connection, and the final output feature is obtained.

A cross-guidance module for RGB and LiDAR features was constructed by using the proposed guidance module twice with different roles (Figure 1). The first one takes the LiDAR feature as the input and the RGB feature as the guidance feature, while the second one takes the RGB feature as the input and the LiDAR feature as the guidance feature. Through this cross-attention method, each feature can receive information from the features of the other modalities and merge it with its own information.

B. RESIDUAL ATROUS SPATIAL PYRAMID BLOCK

Previous studies used general convolution blocks or residual blocks [6] to extract features from the input data. In this study, we used the RASP blocks to derive more significant features and enhance the performance of our model. The RASP block has several branches composed of dilated convolutional layers, similar to GoogleNet [7] and ResNext [32] (Figure 4). If a large receptive field is used, the area visible in the image becomes larger. Conventional CNNs use pooling operations to prevent overfitting after a large receptive field, which is used to capture the overall characteristics of an image. However, this can lead to loss of information. We used a dilated convolution that employs an extended receptive field containing several zero values to overcome the disadvantages. The processed input features with C channels generate d_n consecutive dilated convolution blocks, such as an ASPP block [8]. After all the dilated convolutional layers, the generated features are concatenated as one feature map. In this block, a residual connection [6] is added between the input feature and the feature that has passed the final convolution to enable efficient learning and a higher level of performance. If a downsampling operation is added, similar to a general residual block [6], the first dilated convolutional layers in the RASP block have a stride of 2. To demonstrate the effectiveness of the RASP block, we discuss the results of ImageNet-1K experiments in Section IV-A.

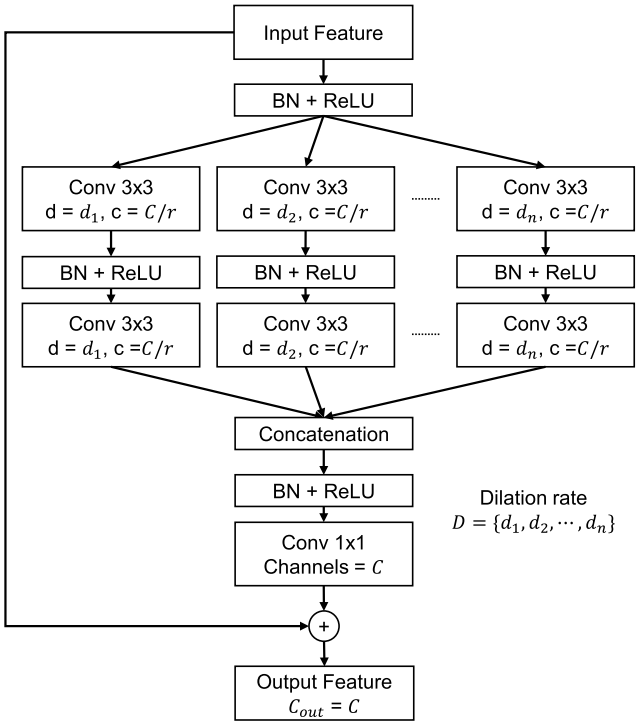

FIGURE 4. Proposed RASP block. *d* and *c* denote the dilation rate and the number of output channels of the convolution layer, respectively. *r* is the reduction ratio of channels to reduce the number of parameters.

TABLE 1. Architectures for the depth completion task. The numbers in the brackets are the input and output channels of the RASP block. D_n denotes the dilation rate for RASP blocks of stage *n*.

	Stage	Output size	Global Network	
			RGB	Lidar
	Initial Conv	1216×256	$7 \times 7,\ 3,\ 32$	$7 \times 7,\ 1,\ 32$
Encoder	Stage 1	608×128	$\begin{bmatrix} RASP,\ 32,\ 64 \\ RASP,\ 64,\ 64 \end{bmatrix}$	
			$D_1 = \{1, 2, 4, 8, 16, 32\}$	
	Stage 2	304×64	$\begin{bmatrix} RASP,\ 64,\ 128 \\ RASP,\ 128,\ 128 \end{bmatrix}$	
			$D_2 = \{1, 2, 4, 8, 16, 32\}$	
	Stage 3	152×32	$\begin{bmatrix} RASP,\ 128,\ 128 \\ RASP,\ 128,\ 128 \end{bmatrix}$	
			$D_3 = \{1, 2, 4, 8\}$	
	Stage 4	76×16	$\begin{bmatrix} RASP,\ 128,\ 128 \\ RASP,\ 128,\ 128 \end{bmatrix}$	
			$D_4 = \{1, 2, 4, 8\}$	
	Stage 5	38×8	$\begin{bmatrix} RASP,\ 128,\ 128 \\ RASP,\ 128,\ 128 \end{bmatrix}$	
			$D_5 = \{1, 2\}$	
Decoder	Up-Sample 5	76×16	$3 \times 3,\ 128,\ 128$	
	Up-Sample 4	152×32	$3 \times 3,\ 128,\ 128$	
	Up-Sample 3	304×64	$3 \times 3,\ 128,\ 128$	
	Up-Sample 2	608×128	$3 \times 3,\ 128,\ 64$	
	Up-Sample 1	1216×256	$3 \times 3,\ 64,\ 32$	
	Last Conv	1216×256	$3 \times 3,\ 32,\ 3$	

C. NETWORK

Figure 2 shows the architecture. We used a framework that consists of global and local networks, as proposed by [21]. The top and bottom of Figure 2 show the global and local networks, respectively. We changed the global and local networks to our proposed networks using the RASP block. The detailed structure of the global network is summarized in Table 1.

The global network has two encoders that take different modalities as inputs. Two inputs, an RGB image and a sparse LiDAR map, passed through each stage of the encoder and exchanged their information with each other through proposed cross-guidance modules. At the end of the two encoders, we performed an element-wise sum of the two feature maps. Then, it was passed through a decoder consisting of transposed convolutional layers using the skip connection between the encoder and the decoder [33]. After passing through the encoders and the decoder, the global network output three components, namely, a prediction map of the global network, a confidence weight, and a guidance map.

The local network comprises an encoder and a decoder. Unlike the global network, each stage of the encoder in the local network has one RASP block. The concatenation of the sparse LiDAR map and the guidance map from the global network was used as the input to the local network, in order to utilize the information that includes an attribute of the RGB image. The skip connections were also performed between the encoder and the decoder, as in the global network. Four guidance modules connect the global and local networks. The features of the guidance modules correspond to the decoder in the global network. Two outputs were generated from the local network: the prediction of the local networks and the confidence weight. Then, the two confidence weights from the global and local networks were input into the softmax layer to generate score maps.

Finally, the predictions from the two networks were integrated by a weighted sum with score maps to predict the desired dense depth map. The final predicted dense depth is calculated as follows:

$$D = w_g \cdot D_g + w_l \cdot D_l, \tag{7}$$

where D_g and D_l are the estimated depth values from the global and local networks, and w_g and w_l are the score maps of the global and local networks, respectively.

D. LOSS FUNCTION

The network updates the model by calculating the error between the predicted value and the ground truth value. We use the mean squared error (MSE) to calculate the loss. The loss function is expressed as follows:

$$L_d(p) = \| \mathbb{1}_{\{y_i > 0\}} \cdot (p_i - y_i) \|_2^2, \tag{8}$$

where p_i and y_i are the estimated value and ground truth value for the i-th point in an input image, respectively.

The overall loss function of our framework can be expressed as follows:

$$L = \lambda_1 \cdot L_d(p_{out}) + \lambda_2 \cdot L_d(p_{global}) + \lambda_3 \cdot L_d(p_{local}), \quad (9)$$

where $L_d(p_{out})$, $L_d(p_{global})$, and $L_d(p_{local})$ are the loss of the final predicted depth map, the predicted depth map of the global network, and the predicted depth map of the local network, respectively. λ_1, λ_2, and λ_3 are the weights of the loss functions. We set $\lambda_1 = 1$, $\lambda_2 = 0.1$, and $\lambda_3 = 0.1$.

IV. EXPERIMENTS

We conducted comprehensive experiments to demonstrate the RASP block on ImageNet-1K [34] and test the proposed architecture using RASP blocks and guidance modules on the KITTI depth completion benchmark dataset [2]. All experiments were implemented in the Pytorch framework [35].

A. ImageNet-1K

ImageNet-1K [34] is the most widely used dataset for evaluating network performance. It includes 1.2M training images and 50k validation images with 1,000 classes. We followed common data augmentation rules: 224 × 224 random crop, horizontal flip, and normalization [6], [16]. The network weights were initialized using He's method [36]. We used an SGD with a mini-batch size of 128 for 90 epochs. The learning rate started at 0.05 and was divided by 10 every 30 epochs. During the test, we applied a 224 × 224 single-crop evaluation.

1) NETWORK ARCHITECTURE FOR ImageNet-1K

The architecture for the ImageNet experiments is presented in Table 2. The 7 × 7 convolutional layer was used in the initial stage. Stages 1–5 consist of two RASP blocks. To reduce the size of the feature maps, the first RASP block of each stage had a downsampling operation, and the first convolution layers of blocks had a stride of 2. We set the reduction ratio of all RASP blocks to 2. The network ends with a 7 × 7 global average pooling operation and a 1,000-d fully connected layer with softmax for classification. Except for stages 1–5, the rest of the network is the same as ResNet architectures [6] for the ImageNet-1k experiments, and the total number of layers and parameters are 32 and 19.27M, respectively.

2) IMAGE CLASSIFICATION ON ImageNet-1K

The experimental results for the ImageNet dataset are summarized in Table 3. We compared the proposed network to the widely used ResNet [6] for the backbone network in many computer vision tasks. The results show that our RASP network has a top-1 error rate of 24.29%, outperforming both ResNet-34 and ResNet-50. The error rates of our results are 2.4% and 0.27% lower than those of ResNet-34 and ResNet-50, respectively. These two ResNet networks have more parameters than the proposed network. As a result, the proposed RASP block has better representation power than the general residual block.

TABLE 2. Architecture for ImageNet-1K experiments. The numbers in the brackets represent the input and output channels of the RASP block. D_n denotes the dilation rate for RASP blocks of stage n.

Stage	output size	RASP-32
Initial Conv	224 × 224	7 × 7, 3, 32
Stage 1	112 × 112	$\begin{bmatrix} RASP,\ 32,64 \\ RASP,\ 64,64 \end{bmatrix}$
		$D_1 = \{1, 2, 4, 8, 16\}$
Stage 2	56 × 56	$\begin{bmatrix} RASP,\ 64,128 \\ RASP,\ 128,128 \end{bmatrix}$
		$D_2 = \{1, 2, 4, 8\}$
Stage 3	28 × 28	$\begin{bmatrix} RASP,\ 128,256 \\ RASP,\ 256,256 \end{bmatrix}$
		$D_3 = \{1, 2, 4, 8\}$
Stage 4	14 × 14	$\begin{bmatrix} RASP,\ 256,512 \\ RASP,\ 512,512 \end{bmatrix}$
		$D_4 = \{1, 2\}$
Stage 5	7 × 7	$\begin{bmatrix} RASP,\ 512,512 \\ RASP,\ 512,512 \end{bmatrix}$
		$D_5 = \{1, 2\}$
Pooling	1 × 1	7 × 7, global average pooling
Final	1000-d Fully-Connected, softmax	
# Params	19.27M	

TABLE 3. Evaluation results on ImageNet-1K dataset. Except for our architecture, all experimental results are reported in [16].

Architecture	# of Params	Top-1 Err (%)	Top-5 Err (%)
ResNet-18 [6]	11.69 M	29.60	10.55
ResNet-34 [6]	21.80 M	26.69	8.60
ResNet-50 [6]	25.56 M	24.56	7.50
RASP-32 **(ours)**	19.27 M	24.29	7.11

B. KITTI DEPTH COMPLETION

We evaluated the proposed architecture against the KITTI depth completion dataset [2], which includes RGB images and depth maps from projected LiDAR point clouds. The depth maps are extremely sparse with approximately 5% of the pixel values as shown in Figure 1. The semi-dense depth maps created by aggregating the LiDAR scans are provided as ground truth, as shown in Figure 1. The KITTI depth dataset comprises 85,898 items of training data, 1,000 items of validation data, and 1,000 items of test data without ground truth.

1) IMPLEMENTATION DETAILS

We used a mini-batch size of 8 and adopted the ADAM optimizer with an initial learning rate of 0.005 for 14 epochs. The learning rate was halved at 8 and 12 epochs. Our network was trained from scratch (without any pretrained weights) with the training set of the KITTI depth dataset only. A horizontal flip was used for data augmentation. Because of the lack of valid pixels in the top of the sparse LiDAR map, we

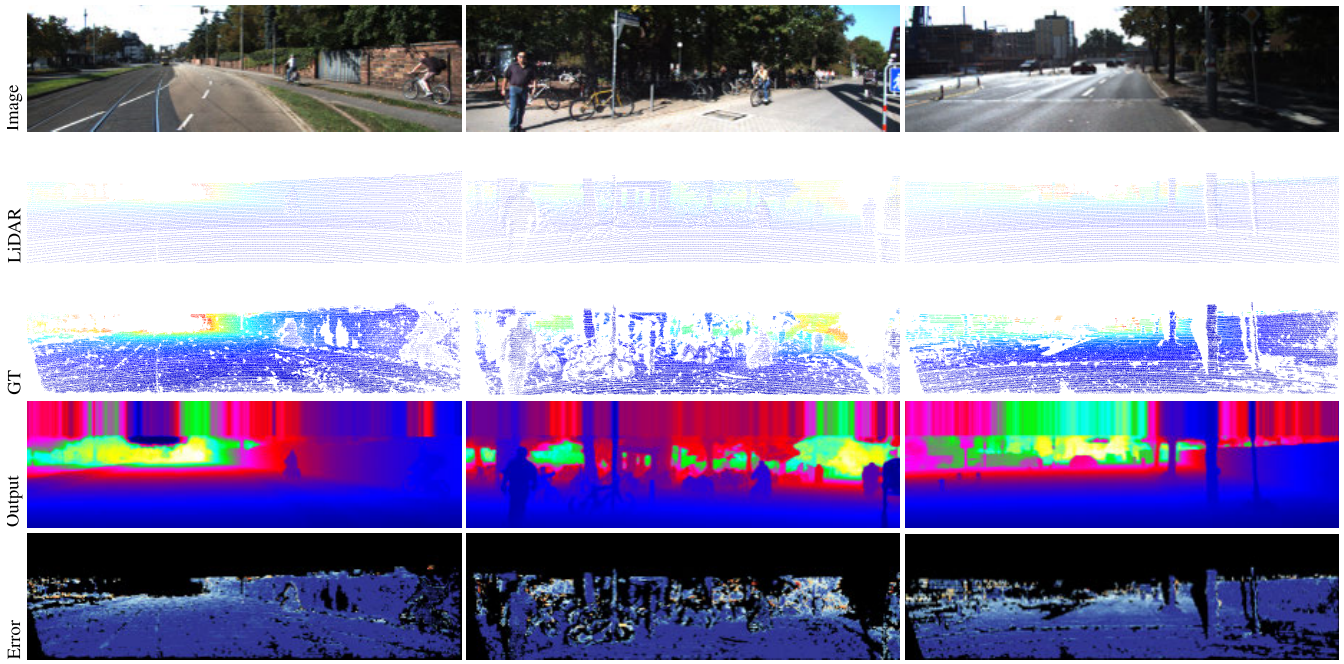

FIGURE 5. Results of the proposed network on the selected validation set of KITTI depth completion.

inferred that irrelevant values in the feature maps affect batch normalization and pooling operation. Therefore, we applied a bottom crop of size 1216×256, for both training and testing procedures. Furthermore, we did not use data normalization in our best model training.

2) EVALUATION METRICS

For the evaluation of the proposed architecture against the KITTI depth completion benchmark, we used the official error metric defined in [2], as follows:

- Root mean squared error (RMSE):

$$\sqrt{\frac{1}{n}\sum_i \left(d_i - d_i^*\right)^2}$$

- Mean absolute error (MAE):

$$\frac{1}{n}\sum_i \left|d_i - d_i^*\right|$$

- Root mean squared error of the inverse depth (iRMSE):

$$\sqrt{\frac{1}{n}\sum_i \left(\frac{1}{d_i} - \frac{1}{d_i^*}\right)^2}$$

- Mean absolute error of the inverse depth (iMAE):

$$\frac{1}{n}\sum_i \left|\frac{1}{d_i} - \frac{1}{d_i^*}\right|,$$

where d_i and d_i^* are the estimated depth values and ground truth, respectively. n is the collection of the valid pixels of the ground truth. We mainly focused on the RMSE for comparison because the RMSE is more sensitive to large errors (*e.g.*, outliers) and the base metric on the KITTI depth completion benchmark. Note that the final result depends on the loss function used, and each metric is not completely dependent on other metrics because of the inverse operation.

TABLE 4. Comparison of the proposed architecture with state-of-the-art methods using the test set of the KITTI depth completion benchmark.

Method	RMSE [mm]	MAE [mm]	iRMSE [1/km]	iMAE [1/km]
2D space and trained with the official dataset only				
SparseConvs [2]	1601.33	481.27	4.94	1.78
Morph-Net [22]	1045.45	310.49	3.84	1.57
CSPN [18]	1019.64	279.46	2.93	1.15
Spade-RGBsD [37]	917.64	234.81	2.17	0.95
NConv-CNN [19]	829.98	233.26	2.60	1.03
Sparse-to-Dense [20]	814.73	249.95	2.80	1.21
CrossGuidance **(ours)**	807.42	253.98	2.73	1.33
2D space and trained with additional datasets				
DDP [28]	832.94	203.96	2.10	0.85
RGB&certainty [21]	772.87	215.02	2.19	0.93
2D and 3D space				
PwP [27]	777.05	235.17	2.42	1.13
DeepLiDAR [26]	758.38	226.50	2.56	1.15
UberATG-FuseNet [29]	752.88	221.19	2.34	1.14

3) COMPARISON TO STATE-OF-THE-ART METHODS

We evaluated our architecture on the KITTI depth completion dataset [2]. We set the reduction ratio of all RASP blocks to 2. Table 4 shows the comparisons with other state-of-the-art models on the KITTI depth completion leaderboard. For a fair comparison, we divided the state-of-the-art models into three

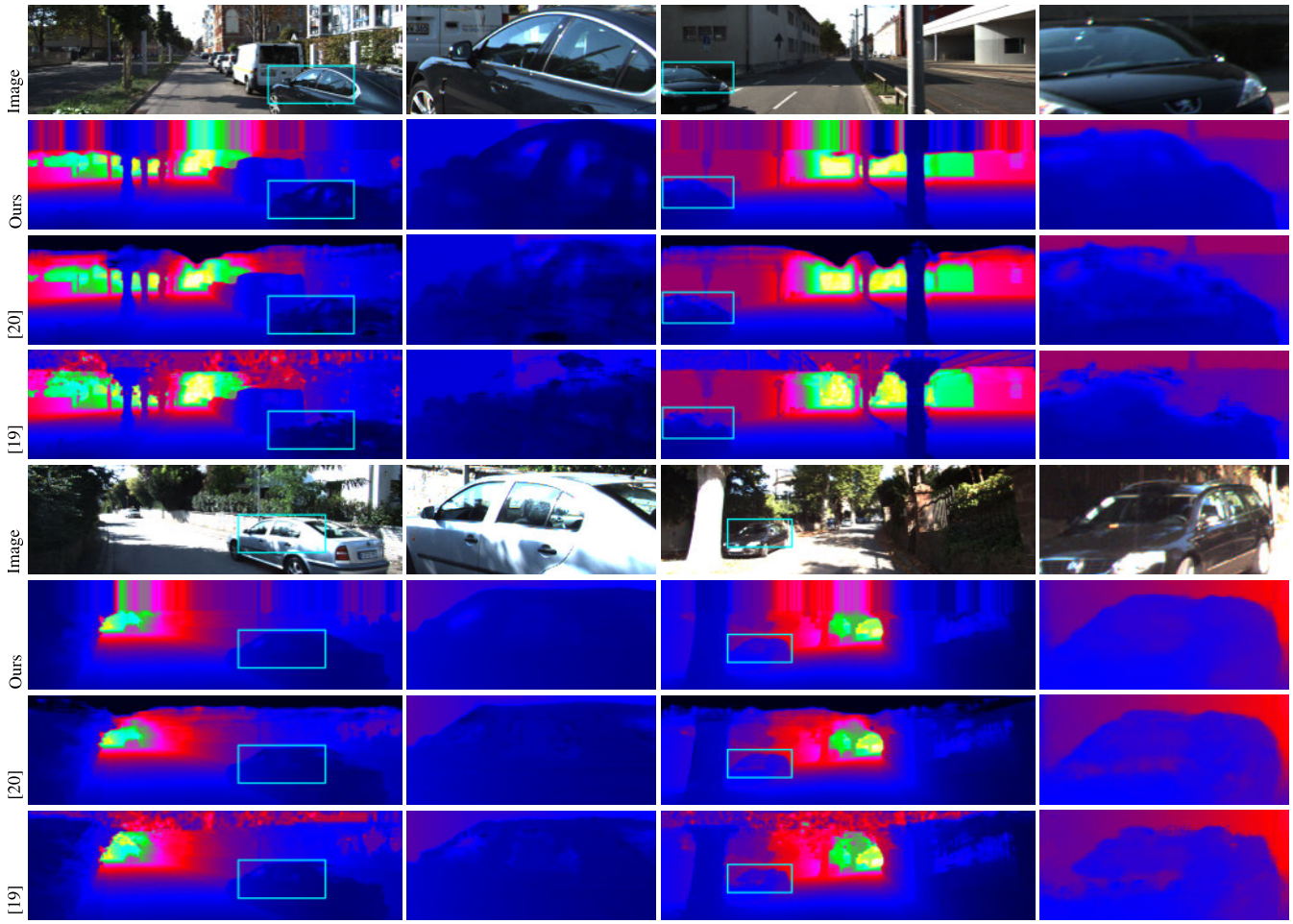

FIGURE 6. Qualitative comparison with other methods. Our results were compared with sparse-to-dense [20] and NConv-CNN [19]. Our results reconstructed the 3D objects well and provided cleaner boundaries.

categories. First, UberATG-Fusenet [29] was divided because it exploits the 3D point cloud of LiDAR data. DeepLidar [26] and PwP [27] use the surface normal information to determine the representation power for the depth completion task. Then, we divided the model trained with only the official dataset and the model trained with the official dataset and additional data. RGB&certainty [21] and DDP [28] use a pre-trained model on the Cityscapes dataset [38] and the virtual KITTI dataset [39], which is similar to the real world KITTI dataset [1]. The proposed architecture outperformed all the other methods trained with only the official dataset in terms of RMSE, which is the main metric of the benchmark. Our model showed lower RMSE than the sparse-to-dense [20] method, which includes a residual block [6] with twice the number of parameters.

4) ABLATION STUDIES

We conducted extensive ablation studies to verify the effectiveness of the proposed guidance module and learning parameters in training the architecture. Table 5 shows that the proposed guidance module is effective in improving the performance. The first row means that the global network in our architecture does not include the guidance module; *i.e.*, the two encoders do not share a connection. The global network with the cross-guidance modules is represented in the second row. The global model with cross-guidance modules shows an RMSE that is lower by 24.04 mm than the RMSE of the model without the cross-guidance modules. Furthermore, this model performs better than the model that adds the local network to the global network and connects the global network and the local network by element-wise sum without the guidance module. Finally, the architecture running all the guidance modules showed the highest performance, demonstrating the

TABLE 5. Ablation study for each component on selected validation set of KITTI depth completion. The model with the entire guidance module achieves the best result.

Models	RMSE [mm]	MAE [mm]	iRMSE [1/km]	iMAE [1/km]
Global	869.99	267.10	2.88	1.26
+ Cross Guidance	845.94	254.98	2.98	1.26
Global + Local	851.30	247.75	2.54	1.10
+ All Guidance modules	830.64	250.76	2.70	1.25

TABLE 6. Ablation study for the network size and an initial learning rate on selected validation set of KITTI depth completion. Our full architecture with guidance modules was used in the experiment. The first column represents the output channels of stages 2–5 in all encoders. r is the reduction ratio of RASP blocks in the architecture.

# of out channels	r	Init LR	# of Params.	RMSE [mm]	MAE [mm]
128	4	0.001	5.4M	848.81	248.81
128	4	0.005	5.4M	865.49	253.62
128	2	0.001	11.02M	832.22	244.67
128	2	0.005	11.02M	830.64	250.76
256	2	0.001	29.73M	840.86	252.88
256	2	0.005	29.73M	854.38	263.47

efficiency of the guidance module in the depth completion task (the last row of Table 5).

Table 6 presents the performance of the overall architecture, depending on learning parameters and network sizes. The results show that the KITTI depth dataset does not require a large network to train, as mentioned in [21].

V. CONCLUSION

In this study, we developed a deep architecture comprising multiple cross-guidance modules and residual atrous spatial pyramid (RASP) blocks to complete a dense depth map with an RGB image and a sparse LiDAR map as inputs. We proposed the cross-guidance method, which is an attention method combining the features of different modalities to compensate for the lack of their representation power. The RASP block comprises multiple dilated convolutions using different dilation rates in parallel with skip connections. This allows us to derive more significant features and have a large receptive field. Extensive experiments on ImageNet-1K and KITTI depth completion showed that the proposed cross-guidance method and the RASP block effectively improved performance. The proposed architecture attained a state-of-the-art performance score against the KITTI depth completion benchmark models in 2D space without pre-training or fine-tuning the other datasets. In future, we plan to pre-train our architecture using a large-scale dataset and extend the model from 2D space to 3D space.

REFERENCES

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[2] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 11–20.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, PP. 1–14.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[9] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, PP. 1–13.

[10] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 636–644.

[11] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[12] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.

[13] M. Jaderberg, K. Simonyan, A. Zisserman, and K. kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 2017–2025.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[15] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[16] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[17] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 2048–2057.

[18] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 103–119.

[19] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through CNNs for guided sparse depth regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 17, 2019, doi: 10.1109/TPAMI.2019.2929170.

[20] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised Sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3288–3295.

[21] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy LiDAR completion with RGB guidance and uncertainty," in *Proc. 16th Int. Conf. Mach. Vis. Appl. (MVA)*, May 2019, pp. 1–6.

[22] M. Dimitrievski, P. Veelaert, and W. Philips, "Learning morphological operators for depth completion," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.* Cham, Switzerland: Springer, 2018, pp. 450–461. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-01449-0_38

[23] D. Doria and R. J. Radke, "Filling large holes in LiDAR data by inpainting depth gradients," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 65–72.

[24] J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the CPU," in *Proc. 15th Conf. Comput. Robot Vis. (CRV)*, May 2018, pp. 16–22.

[25] Y. Shen, J. Li, and C. Lu, "Depth map enhancement method based on joint bilateral filter," in *Proc. 7th Int. Congr. Image Signal Process.*, Oct. 2014, pp. 153–158.

[26] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3313–3322.

[27] Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li, "Depth completion from sparse LiDAR data with depth-normal constraints," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2811–2820.

[28] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (DDP) from single image and sparse range," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3353–3362.

[29] Y. Chen, B. Yang, M. Liang, and R. Urtasun, "Learning joint 2D-3D representations for depth completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10023–10032.
[30] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3156–3164.
[31] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 552–568.
[32] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
[33] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28
[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2019, pp. 8024–8035.
[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[37] M. Jaritz, R. D. Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with CNNs: Depth completion and semantic segmentation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 52–60.
[38] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[39] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4340–4349.

SIHAENG LEE (Member, IEEE) received the B.S. degree in mechanical system design engineering from the Seoul National University of Science and Technology, Seoul, South Korea, in 2014, and the M.S. degree from the Division of Future Vehicle, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and deep learning especially with regard to their application to autonomous vehicles.

JANGHYEON LEE (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree. His research interests include computer vision, deep learning, and machine learning.

DOYEON KIM (Member, IEEE) received the B.S. degree in computer science from Korea University, Seoul, South Korea, in 2016, and the M.S. degree in robotics program from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2018, where she is currently pursuing the Ph.D. degree in electrical engineering. Her research interests include computer vision, deep learning, and machine learning.

JUNMO KIM (Member, IEEE) received the B.S. degree from Seoul National University, Seoul, South Korea, in 1998, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, in 2000 and 2005, respectively. From 2005 to 2009, he was a Research Staff Member with the Samsung Advanced Institute of Technology (SAIT), South Korea. He joined the Faculty of KAIST, in 2009, where he is currently an Associate Professor of electrical engineering. His research interests are in image processing, computer vision, statistical signal processing, machine learning, and information theory.