IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 November 2023, accepted 27 December 2023, date of publication 11 January 2024, date of current version 7 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3353048

RESEARCH ARTICLE

Deep Sparse Depth Completion Using Multi-Scale Residuals and Channel Shuffle

ZHI LIU AND CHEOLKON JUNG, (Member, IEEE)
School of Electronic Engineering, Xidian University, Xi'an 710071, China

Corresponding author: Cheolkon Jung (zhengzk@xidian.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 62111540272.

ABSTRACT Depth completion aims to recover dense depth maps from sparse depth maps. Recent approaches have used additional modalities as guidance to improve depth completion performance. Image-guided depth completion uses scene information from color images, but it still produces inaccurate object boundaries. In this paper, we propose deep sparse depth completion using multi-scale residuals and channel shuffle, named ReCSNet. ReCSNet is a dual-branch network based on a U-shaped architecture. ReCSNet consists of one VIS-Semantic-Guided Branch (VSGB) and one Sparse Depth Guided Branch (SDGB) to get global color, edge information, and local accurate depth information. VSGB utilizes two encoders to extract features from the VIS-Semantic image pairs and the sparse depth maps, and employs a feature channel shuffle mechanism to blend the two sets of encoded features. The semi-dense depth map generated by VSGB is concatenated with the original sparse depth map and input into SDGB to predict the second semi-dense depth map. The confidence maps generated by the two branches are adaptively fused to generate the final depth map. Moreover, we incorporate multi-scale residuals obtained from the VIS image and concatenate them with the decoded features to further enhance the constraint on object boundaries. At the rear of the dual-branch network, we add a Repetitive Deformable Convolution Module (RDCM) to further refine the depth values in object edges. Experimental results show that ReCSNet achieves outstanding performance on the KITTI depth completion validation dataset with an improvement of 16mm in the root mean square error (RMSE) metric.

INDEX TERMS Depth completion, channel shuffle, deformable convolution, multi-scale residuals, semantic segmentation.

I. INTRODUCTION

Depth sensors can accurately measure the distance of objects in a scene and are widely used in the fields such as autonomous driving [1], 3D modeling [2], augmented reality [3] and SLAM [4]. The LiDAR camera uses sensors that introduce LiDAR technology into imaging. It can provide high-precision depth data and color images outdoors and obtain more comprehensive and accurate environment information. The KITTI dataset [5], [6], [7] was collected by the Velodyne HDL-64E rotating 3D laser scanner. As shown in Fig. 1 (c), even for expensive LiDAR, the valid points in the collected depth map are very sparse (less than 5%), making it impossible to directly apply the collected sparse data to the above fields. Therefore, sparse depth completion is required to recover the depth of unknown positions from the sparse depth maps.

With the continuous exploration of deep learning and sensor fusion, depth completion has advanced from relying on a single sparse depth map to using multiple modalities as input for better performance. The sparse depth map used as a single-modal input in the sparse depth-based method [7], [8] cannot provide structural information of the scene, thus the obtained depth map loses part of the object boundary. In image-guided multi-modal methods, the input data includes VIS images [9], surface normals [10], and semantic maps [11], [12], corresponding to the sparse depth maps. These modal information provide additional information to estimate more accurate depth values. The most widely used one is to take high-resolution color images

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

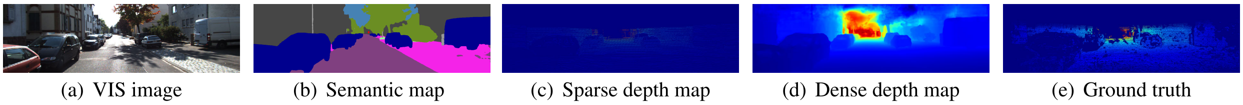

(a) VIS image (b) Semantic map (c) Sparse depth map (d) Dense depth map (e) Ground truth

FIGURE 1. Examples of three different modalities: (a) VIS image, (b) semantic map, (c) sparse depth map, (d) predicted dense depth map, and (e) the corresponding ground truth.

and sparse depth maps together as inputs and use the CNN model based on the encoder-decoder structure to densify the sparse depth map. PENet [9] utilized a dual-branch network structure to fuse the features of multi-modal inputs, i.e. sparse depth maps and VIS images. RigNet [13] repeatedly employed multiple hourglass structures to extract reliable VIS image features to provide clear guidance for depth recovery. EMDC [14] also applied a dual-branch framework to extract global and local features and merge cross-modal features between the two branches. These methods only leveraged VIS images as the guidance of global scene structure information. However, VIS image lacks strong supervision at object boundaries and sparse depth maps are accompanied by a lot of noise at object edges.

To address the issues, we propose a deep sparse depth completion network using multi-scale residuals and channel shuffle, named ReCSNet. ReCSNet is based on a dual-branch and encoder-decoder structure as shown in Fig. 2. The upper branch, i.e. the VIS-Semantic Guided Branch (VSGB), focuses on global information. This branch utilizes two encoders that respectively extract color and structural information from the VIS image and reliable object boundaries from the semantic image, to guide the completion of sparse depth maps, generating the first semi-dense depth map. To introduce complementary information between the two encoders while extracting features from different modalities, we employ channel shuffle operation to blend the two sets of encoded features. The lower branch, i.e. the Sparse Depth Guided Branch (SDGB), concentrates on extracting precise local information. By utilizing the sparse depth map as input again, the true depth values in this map are used to constrain the semi-dense depth map generated by the upper branch. The second semi-dense depth maps is obtained through the lower branch encoder-decoder. While the two branches output two semi-dense maps, they produce corresponding confidence maps. The two semi-dense depth maps are fused through a pixel-level weighted sum to obtain the final dense depth map. Moreover, we add multi-level residuals obtained from VIS image into the decoders of the two branches to recover the depth maps with sharp edges during subsequent up-sampling.

Compared with existing methods, the main contributions of this paper are as follows:

- We propose a deep sparse depth completion network using multi-scale residuals and channel shuffle, named ReCSNet. ReCSNet is based on dual-branch with a U-shaped encoder-decoder architecture, in which the upper branch generates a semi-dense depth map focusing on color and edge information in VIS and semantic maps, and the lower branch further refines the semi-dense depth map using the original sparse depth pixels.
- We use channel shuffle to mix the encoded features from two different modalities in the upper branch by introducing complementary guidance information into feature extraction.
- We utilize multi-scale residuals from VIS image to optimize ReCSNet and generate a depth map with clear object boundaries.
- We present a novel Repetitive Deformable Convolutional Module (RDCM) to further refine dense depth maps predicted by the dual-branch network.

The rest of this paper is organized as follows: Section II reviews the related work, including non-image guided and image-guided sparse depth completion methods. Section III describes the network architecture and loss function of the proposed ReCSNet in detail, while Section IV provides the experimental results on the KITTI dataset. Section V draws conclusion of this paper with future work.

II. RELATED WORK

Sparse depth completion methods refer to techniques that aim to recover dense depth maps from sparse depth maps. These methods can be categorized into two main categories based on whether they use image guidance or not. The first category, without image guidance, focuses on using only the sparse depth map and possibly additional geometric constraints to complete the dense depth map. The second category, with image guidance, utilizes additional modalities such as color or grayscale images to provide additional cues for completing the depth map.

A. NON-IMAGE GUIDED SPARSE DEPTH COMPLETION

Non-image guided sparse depth completion methods directly infer missing depth information from sparse depth maps, which can be further divided into three subcategories: sparse perception convolutional neural networks, normalized convolutional neural networks, and those training with auxiliary images. The data in sparse depth maps is extremely sparse, while standard convolutional operations are friendly to dense inputs. Therefore, researchers have used binary masks to mark the positions of real and missing points in the input map to improve the performance of standard convolution layers on sparse inputs. Uhrig et al. [7] proposed the first non-image guided method based on deep learning, which used a novel sparse convolution operation to address the mosaic effect that occurs when regular convolution processes sparse inputs. Huang et al. [15] introduced three sparse-invariant (SI) operations and built the hierarchical

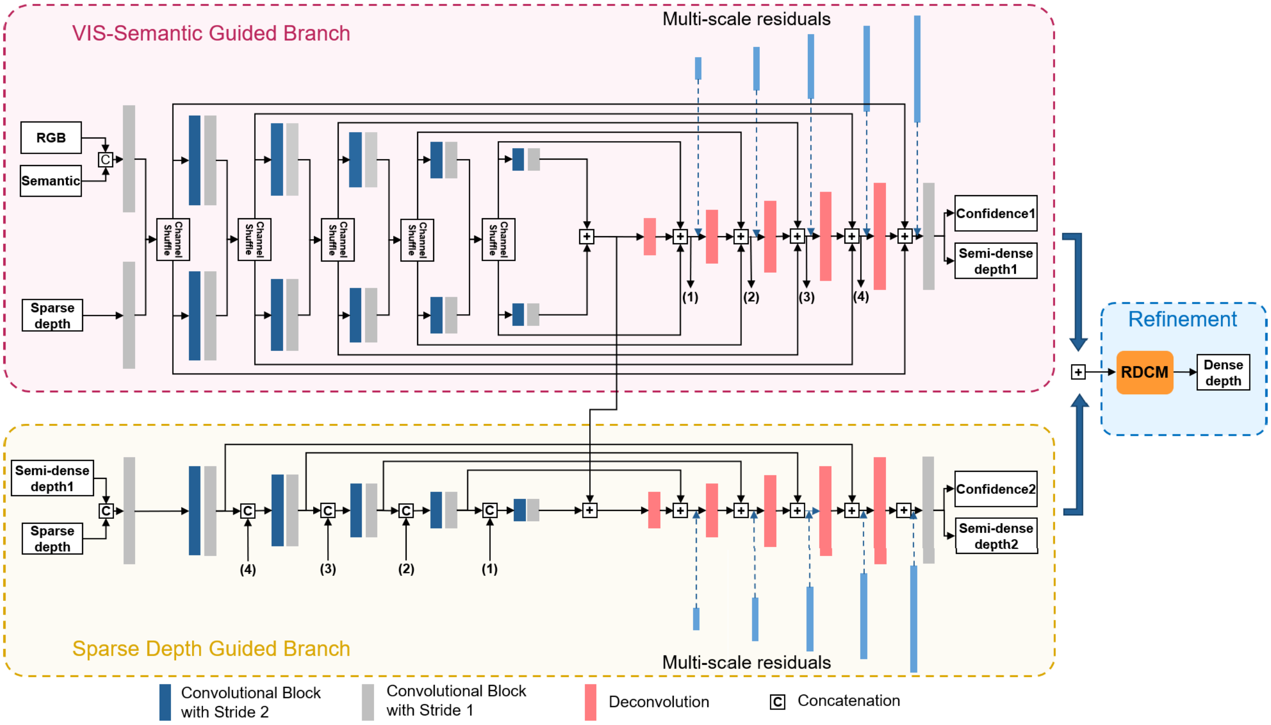


FIGURE 2. **Network architecture of the proposed ReCSNet for sparse depth completion based on dual-branch and encoder-decoder structure. The VIS-Semantic Guided Branch (VSGB) focuses on global color and edge information. The Sparse Depth Guided Branch (SDGB) concentrates on local information by using original sparse depth map.**

multi-scale sparsity-invariant network (HSMNet) based on an encoder-decoder to address the deficiency that sparse convolution is not suitable for encoder-decoder networks. The saturation in early layers can cause a reduction in model performance of the validity masks, as previously pointed out by some studies [16]. Eldesokey et al. [16] proposed a new method based on normalized convolution, called the Normalized Convolutional Neural Network (NCNN), which generated continuous uncertainty maps to weight the features. Furthermore, to achieve faster convergence, the convolution filters were non-negative constrained by the SoftPlus function [17]. To address the lack of semantic cues in the single input of sparse depth map, Lu et al. [18] proposed a framework that introduced an auxiliary learning branch to reconstruct depth. The input of this framework was only a sparse depth map, while the RGB image was treated as the learning target during training. By predicting the reconstructed image and dense depth map, this method cleverly leveraged RGB image features to improve the accuracy of depth completion. Lu et al. [19] used an autoencoder to generate RGB information in the latent space and predict the final depth at the output end. However, due to unsupervised learning and the absence of more dense depth maps as ground truth, the completion effect is unsatisfactory.

B. IMAGE-GUIDED SPARSE DEPTH COMPLETION

Although unguided sparse depth completion can quickly infer the depth information of a scene without relying on additional image information, the accuracy of predicted values is not high enough to meet the high requirements for environmental depth accuracy in applications such as autonomous driving and robot navigation. To improve the accuracy and robustness of reasoning, researchers have proposed image-guided sparse depth completion methods, which infer dense depth maps from auxiliary information such as color images, semantic maps, or normal maps corresponding to the depth maps and sparse depth information. As shown in Fig. 3, image-guided depth completion methods produce more accurate and robust results. There are five subclasses of image-guided methods: early fusion strategies, late fusion strategies, explicit 3D representation methods, residual depth methods, and spatial propagation network (SPN) based methods. The first two categories are distinguished based on the fusion strategy's position in the model, while the last three categories are classified based on different models and also use early or late fusion strategies.

The early fusion strategies typically employ an encoder-decoder structure and a two-stage structure from coarse to fine. The encoder-decoder model has a simple structure and can obtain multi-scale features of the image. Ma and Karaman, [20] concatenated the sparse depth map and RGB image, then input them into an encoder-decoder network based on ResNet-50 [21]. The coarse-to-fine prediction model first obtains a coarse depth map in the coarse prediction stage, and then uses subsequent refinement operations to predict a refined depth map from the coarse depth map and RGB image. Hambarde and Murala [22] proposed an S2DNet composed of two pyramid networks, S2DCNet and S2DFNet,

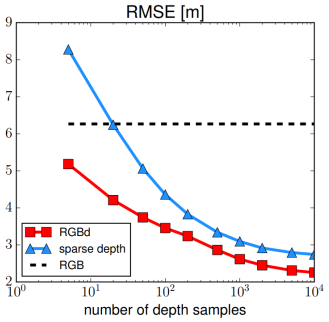


FIGURE 3. Differences in RMSE between unguided and image-guided depth completion methods on the KITTI dataset, redrawn from [20].

which performed coarse filtering prediction and refinement operations, respectively.

The methods that utilize late fusion strategies can be divided into three types: dual-encoder networks, dual-encoder-decoder networks, and global-local prediction networks. Each of them contains two encoders that extract features from RGB and sparse depth maps. Ito et al. [23] used two independent encoders to extract multiscale features from the two modalities, and directly concatenated the extracted features as the input of the decoder. Dual-encoder-decoder networks consist of an image branch and a sparse depth map branch, each of which includes an encoder and a decoder. Fusion operation is performed between the decoder of the first branch and the encoder of the second branch. RigNet [13] used the repeated hierarchical hourglass network (RHN) to extract image features and also applied the repeated guidance module (RG) based on dynamic convolution [24] for feature fusion. In the methods that employ global-local prediction networks, RGB and the original sparse depth map are regarded as global information, and a single sparse depth map is treated as local information. The global network predicts the depth map from the global information, while the local network predicts the depth values from the local information. The final depth map is obtained by merging the two depth maps. In predicting the dense depth map using the global and local network, Cheng et al. [25] also generated corresponding confidence maps, which were combined with a weighted sum to generate the final dense depth map.

Methods of using explicit 3D representations to extract 3D geometry cues from sparse depth maps. 3D-aware convolutions are used to remove disturbances caused by missing values in local neighborhoods of sparse depth maps, which is different from standard 2D convolutions. ACMNet [26] used graph propagation for non-grid convolutions. Qiu et al. [10] proposed the DeepLiDAR network, which consisted of a surface normal branch and a color branch.

The residual depth methods predict both the residual map and the depth map simultaneously. The residual map contains rich edge and texture information, which can enhance the object edge in the predicted blurry depth map. Additionally, the residual map is less computationally burdensome during training. Liao et al.'s approach first completed the sparse depth map into a blurry depth map and then predicted the residual map through the network [27]. Finally, the dense depth map was obtained by element-wise summation of the blurry depth map and the residual map. Gu et al. [28] proposed DenseLiDAR first predicted a pseudo-depth map with morphological operations and then fed the pseudo-depth map, RGB image, and sparse depth map into a dual-encoder-single-decoder module to predict the residual map, which was ultimately obtained by linear addition to generate the final depth map.

The affinity matrix represents the similarity between a reference point and its neighboring points, and is commonly used for fine-grained predictions in computer vision tasks. Cheng et al. [25], [29] first proposed the convolutional spatial propagation network (CSPN), which applied the spatial propagation network (SPN) to the task of depth completion. CSPN refined the rough depth map and affinity matrix predicted by a network based on an encoder-decoder structure. During spatial propagation, the depth value of the reference point was calculated using the diffusion process of the affinity matrix and the depth values of its local neighbors. Park et al. [2] proposed a novel non-local SPN, which learned non-local neighbors and confidence maps with affinity matrices through an encoder network and skip connections. Unlike fixed local neighbors, it performed spatial propagation using deformable convolution on K non-local neighbors of the reference point. Lin et al. [30] proposed a dynamic spatial propagation network (DySPN) based on attention mechanism, which used decoupling based on adjacent data points' distance to learn an adaptive affinity matrix for improving depth prediction performance during spatial propagation.

III. PROPOSED METHOD

The proposed sparse depth completion network consists of two branches: a VIS-Semantic Guided Branch (VSGB) and a Sparse Depth Guided Branch (SDGB). They focus on the extraction of global color, edge features and local precise features, respectively, and each predicts a semi-dense depth map and a corresponding confidence map for adaptive fusion to obtain the final dense depth map. The entire model is trained end-to-end on the KITTI depth completion dataset.

A. VIS-SEMANTIC GUIDED BRANCH

VSGB located in the red dotted box in Fig. 2 obtains global color and structure information provided by VIS, object boundary information in an aligned semantic segmentation map, and depth information from sparse depth maps. VSGB is comprised of two encoders and one decoder. The VIS image and the semantic segmentation map are concatenated as input to one of the encoders to extract rich scene information. The original sparse depth map serves as the input to the other encoder, directly extracting depth features from the collected depth data by LiDAR sensor. The encoders sequentially downsample through convolutional blocks with a stride of 2 to obtain features at various scales, followed by further extraction of richer and deeper-level features through convolutional blocks with a stride of 1. To enhance the ability of the two encoders to extract features from their respective inputs, we use a channel shuffle mechanism to mix features of the same size from the two encoders. The decoder gradually restores the feature size to the same as the original sparse depth map through transposed convolution. Finally, a convolution layer with a kernel size of 3 and a stride of 1 is applied to the final features to generate a semi-dense depth map and a confidence map. Moreover, we use skip connections to perform element-wise addition between the features of the two encoders and the decoder to reduce information loss due to down-sampling.

The whole process is defined as follows:

$$D_{VSG}, Cf_{VSG} = De_{VSG}(En_{cs}(I, Sem) + En_d(SD)) \quad (1)$$

where I, Sem, and SD correspond to the VIS image, semantic segmentation map and sparse depth map; De_{VSG} denotes the decoder; En_{cs} and En_d are the color-semantic encoder and sparse depth encoder respectively; D and Cf represent the generated semi-dense depth map and confidence map.

1) SEMANTIC SEGMENTATION MAP AS INPUT

As shown in Fig. 1(c), due to the sparsity of the input depth map and the presence of a substantial amount of noise at the edges of objects, additional information is required to supervise the training of the depth completion network. Previous approaches primarily introduced the corresponding VIS image of the depth map into the network to enhance the completion results. The semantic segmentation map, obtained from VIS image segmentation based on semantic information, is beneficial for subsequent image analysis and visual understanding. Within a segmented region, the distribution of pixel values is uniform, and there are significant differences at the boundaries of different objects. As a result, the semantic segmentation map can provide boundary information of objects and suppress boundary noise. SemAttNet [12] introduces a third semantic-guided branch, built on the foundation of color and dense depth guidance. This branch concatenates the semantic segmentation map, the semi-dense depth map obtained from the color guidance branch, and the sparse depth map as input to the encoder-decoder to reduce the variance of depth values around the object boundary. However, this makes the model complex, increases memory requirements, and extends the training time. To reduce the complexity of the model, we directly concatenate the semantic segmentation map and the VIS image and input them into a single encoder to simultaneously obtain both color and edge features.

2) FEATURE CHANNEL SHUFFLE MECHANISM

We employ two separate encoders to extract features of two different modalities of images. One encoder extracts features of semantic maps and VIS images, while the other extracts features of sparse depth maps. When the two encoders extract features independently, information of only a single input modality can be obtained. Channel shuffle, first proposed in ShuffleNet [31], enables information to circulate among different groups after group convolution. FCFR-Net [32] introduced channel shuffle into the backbone network to enhance the ability of extracting features and predicting depth residual maps. Unlike them, we use the channel shuffle mechanism to exchange features extracted by the two encoders of different modalities so that the features of one encoder can be guided by the other encoder's features. The two sets of encoder features are added to the decoder features through skip connections to directly estimate the semi-dense depth map. Features after channel shuffle can be defined as:

(a) R_1 (b) R_2 (c) R$_3$ (d) R_4 (e) R_5

FIGURE 4. Examples of muti-scale residuals from R_1 to R_5.

$$\hat{F}_i^{cs}, \hat{F}_i^{d} = Chunk(CS(Cat(F_i^{cs}, F_i^{d}))) \quad (2)$$

where F_i^{cs}, $\hat{F}_i^{cs}$, F_i^{d} and $\hat{F}_i^{d}$ are the input and output features of the i-th convolutional block in the color-semantic encoder and the sparse depth feature encoder, respectively; Cat represents the concatenation operation; CS is the feature channel shuffle mechanism; and $Chunk$ means the chunk function in Torch to split a feature into two parts.

B. SPARSE DEPTH GUIDED BRANCH

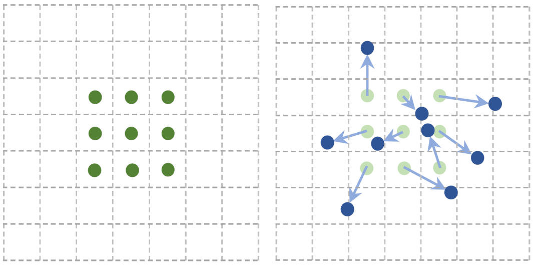
SDGB within the yellow dotted rectangle in Fig. 2 constitutes a common U-shaped architecture comprised of one encoder and one decoder. This branch is utilized for refining the semi-dense depth map produced by VSGB through the guidance of real depth values from the origin sparse depth map. Similar to the upper branch, a convolutional block with a stride of 2 is employed for down-sampling the features. Additionally, to further integrate the global features from the upper branch, the multi-scale features from the decoder output of the upper branch are concatenated with the encoder features of the same size in the lower branch. Finally, at the end of the decoder, the second semi-dense depth map and its corresponding confidence map are generated. The semi-dense depth map D_{SDG} and confidence map Cf_{SDG} generated by the SDG branch are expressed as follows:

$$D_{SDG}, Cf_{SDG} = De_{SDG}(En_{SDG}(SD, D_{VSG})) \quad (3)$$

where De_{SDG} and En_{SDG} are decoder and encoder in sparse depth guided branch, respectively.

C. MUTI-SCALE RESIDUALS

The multi-scale residuals in [33] are utilized to add with the estimated depth maps of each layer in the decoder, thus obtaining sharp edges. In our approach, we incorporate the multi-scale residuals (as seen in Fig. 4) from VIS image into the decoders of two branches, and concatenate them with the decoded features with the same size, to further enhance constraints on the object boundaries in the depth features.



(a) Traditional convolution (b) Deformable convolution

FIGURE 5. Difference of sampling points between traditional convolution and deformable convolution, redrawn from [35]. The sampling point positions of deformable convolution are adaptively generated.

The multi-scale residuals (R_k) of the input VIS image are as follows:

$$I_{i+1} = Down(I_i), i = 1, \ldots, 5 \tag{4}$$

$$I'_5 = Up(I_6), I'_j = Up(I'_{j+1}), j = 4, \ldots, 1 \tag{5}$$

$$R_k = I_k - I'_k, k = 1, \ldots, 5 \tag{6}$$

where $Down$ is the bilinear interpolation with a scale factor of 0.5; I_1 is the VIS image of the original size; and Up is a 2× interpolation up-sampling operation.

D. DEPTH FUSION

We employ the same strategy as used in Chen et al.'s [34] method and PENet [9], i.e., pixel-level adaptive fusion of two semi-dense depth maps using the confidence maps learned from two branches. The fused dense depth map D_f is represented by the following equation:

$$D_f = \frac{e^{Cf_{VSG}} \cdot D_{VSG} + e^{Cf_{SDG}} \cdot D_{SDG}}{e^{Cf_{VSG}} + e^{Cf_{SDG}}} \tag{7}$$

E. REFINEMENT MODULE

Deformable convolution [35] is a convolution operation that has the unique ability to perform deformation sampling. As shown in Fig. 5, unlike traditional convolution that employs fixed kernel sampling, deformable convolution allows for deformable kernel sampling. This crucial distinction enables deformable convolution to more effectively accommodate the deformation characteristics of diverse objects within an image. Specifically, in traditional convolution, the fixed sampling local neighborhood ignores the depth distribution of objects in the local area, which can result in foreground objects mixing with depth values of background object during propagation, and the same applies to objects located in the background. In contrast, in deformable convolution, the position of each sampling point in the convolution kernel is estimated based on the color and depth information of objects in a broad region, allowing for better adaptation to object deformation and improved depth map accuracy.

Inspired by the use of deformable convolutions in NLSPN [2] for depth completion, we propose a Repetitive Deformable Convolution Module (RDCM) as shown in Fig. 6. The RDCM module further refines the depth values of objects in the depth map obtained by fusing the two branches, particularly those located at the edges of foreground and background objects, using deformable convolutions. The RDCM module takes the concatenated features from the last convolution layer of the VSG and SDG branches as input, and generates guidance information and confidence maps through two convolution layers. The guidance information is used to generate the sampling positions of K neighboring points. The coordinates of the K non-local neighbors of the reference point (m, n) are defined as follows:

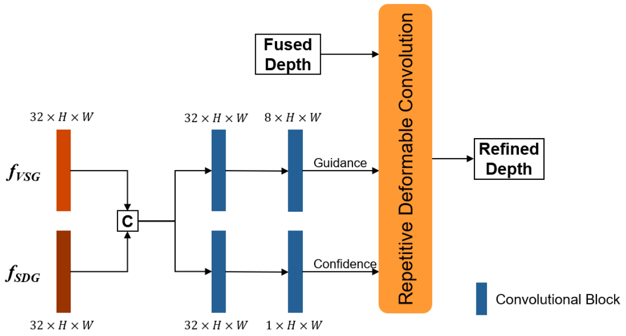


FIGURE 6. Network structure of the Repetitive Deformable Convolution Module (RDCM). RDCM further refines the fused depth map with the guidance and confidence map. f_{VSG} and f_{SDG} are the output features of VSG and SDG branches.

$$\mathcal{N}_{m,n} = \left\{x_{m+i,n+j} \mid (i,j) \in F\left(f_{VSG}, f_{SDG}, m, n\right), i,j \in \mathbb{R}\right\} \tag{8}$$

where F is the module that predicts the coordinates of K neighbors for each pixel; f_{VSG} and f_{SDG} are the features of the last layer of the VSG and SDG branches, respectively.

The affinity matrices are also obtained by guidance. In spatial propagation networks, the affinity matrix encodes the similarity between adjacent pixels and allows for propagation operations in the image or space, while transferring information from one pixel or position to its neighboring pixels or positions. The confidence map represents the reliability of the depth values, where the information of unreliable pixels (e.g., noisy pixels) should not be propagated to neighboring pixels, regardless of their affinity with neighboring pixels. By combining the confidence with normalized affinity, the interference generated by unreliable pixels during propagation can be eliminated, which generates a more accurate depth estimation.

F. LOSS FUNCTION

We utilize the ℓ_2-norm to compute the loss between the predicted dense depth map and the ground truth as shown in Eq. (9). Due to the low density of ground truth being less than 20%, it still contains a large number of invalid pixels. Hence, during the loss calculation, we only consider the pixels with valid depth values as follows:

$$\mathcal{L}(D) = \|(D - D_{GT}) \odot M_d\|_2 \tag{9}$$

where D represents the predicted depth map, D_{GT} is the ground truth, and M_d deotes the valid depth mask, which can be represented as:

$$M_d(m,n) = \begin{cases} 1, & D_{GT}(m,n) \geq 0 \\ 0, & D_{GT}(m,n) < 0 \end{cases} \tag{10}$$

To optimize two branches and generate depth maps with high quality, the total loss is the weighted sum of the losses between the semi-dense depth maps predicted by VSGB and SDGB, and the fused depth map and the ground truth during the first stage of training. In the later stage, only the loss of the fused depth map is calculated as follows:

$$\mathcal{L}_{tol} = \lambda_1 \mathcal{L}(D_{VSG}) + \lambda_2 \mathcal{L}(D_{SDG}) + (1 - \lambda_1 - \lambda_2)\mathcal{L}(D_f) \tag{11}$$

where λ_1 and λ_2 are the weights of losses for VSGB and SDGB.

IV. EXPERIMENTAL RESULTS

A. DATASET

The KITTI depth completion dataset [5], [6], [7] is a benchmark dataset designed to evaluate the performance of depth completion algorithms in autonomous driving and robotics applications. It consists of high-resolution VIS images and corresponding sparse depth maps collected from Velodyne LiDAR sensors. The dataset includes 85898 image pairs for training, with 1000 validation image pairs and 1000 test image pairs also provided by the official website. We evaluate the training performance of ReCSNet using the official validation set after each training epoch. Due to the absence of depth data in the top regions of the depth maps in the KITTI dataset, we crop the bottom of the validation images to 1216 × 352, and use image pairs with a resolution of 1216 × 320 during training. Since the corresponding semantic maps are not provided by KITTI, we use semantic segmentation maps generated by WideResNet38 [36] as presented in SemAttNet [12].

B. EVALUATION METRICS

The commonly used evaluation metrics for depth completion include root mean squared error (RMSE [mm]), mean absolute error (MAE [mm]), root mean squared error of the inverse depth (iRMSE [1/km]), and mean absolute error of the inverse depth (iMAE [1/km]).

C. IMPLEMENTATION DETAILS

We implement ReCSNet using the Pytorch framework and perform training and validation on a NVIDIA GeForce 3090 GPU. For optimization, we employ the Adam optimizer with the parameters β_1 set to 0.9 and β_2 set to 0.99. The weight decay is set to 1×10^{-6}. During the data loading stage, data augmentation techniques such as random crop, flip, and color jitter [20] are utilized. Following the training process of PENet [9], we adopt a three-stage training strategy. In the first training stage, we only train the front dual-branch network without refinement modules. The training images are cropped to 1216 × 320 with a batch size of 4, and trained for 35 epochs. The initial learning rate was set to 1.28×10^{-3} and decayed to $\frac{1}{2}$, $\frac{1}{10}$, and $\frac{1}{100}$ of the initial value at epochs 10, 15, and 25, respectively. In the early stages of training, we set $\lambda_1 = \lambda_2 = 0.2$ in Eq. 11, then set it to 0.05 in the fourth epoch and 0 in the sixth epoch. In the second training stage, we fixed the parameters of the dual-branch network and trained only the RDCM module for 4 epochs. The batch size was set to 8, and the initial learning rate was 1.28×10^{-3}. In the third training stage, we train both parts of the network together, setting the initial learning rates of the dual-branch network and RDCM module to 1.28×10^{-3} and 1.28×10^{-4}, respectively. The training lasted for 60 epochs, and the learning rates decayed to $\frac{1}{2}$, $\frac{1}{10}$, $\frac{1}{50}$, $\frac{1}{250}$ and $\frac{1}{1250}$ of their initial values at epochs 10, 20, 30, 40, and 50, respectively. To shorten the training time, we randomly crop the training images to 576 × 160 and set the batch size to 12.

TABLE 1. **Performance comparison in terms of RMSE on the KITTI depth completion validation dataset. The results of other methods are cited from their papers. Best results are shown in bold, while underline represents the second-best performance.**

Methods	RMSE mm	MAE mm	iRMSE 1/km	iMAE 1/km
PWP [37]	811.07	236.67	2.45	1.11
FCFR-Net [32]	802.62	224.53	2.39	1.00
DenseLiDAR [28]	795.97	-	-	-
UberATG-FuseNet [38]	785.00	217.00	2.36	1.08
ACMNet [26]	781.66	212.61	-	-
GuideNet [39]	777.78	221.59	2.39	1.00
ENet [9]	772.78	215.48	2.18	0.94
ReCSNet	759.24	211.13	2.17	0.92
PENet [9]	757.20	209.00	2.22	0.92
ReCSNet+RDCM	**756.38**	**208.90**	**2.15**	**0.91**

D. EVALUATION ON KITTI DATASET

Dense depth maps predicted by ReCSNet on the KITTI depth completion validation dataset are shown in Fig. 7. Quantitative measurements on the KITTI validation set are shown in Table 1. ReCSNet outperforms the baseline ENet [9] in terms of RMSE, MAE, iRMSE, and iMAE. The modifications proposed in our backbone result in significant improvements in RMSE and MAE, which decrease by 13.5mm and 4.3mm, respectively. Compared to PENet [9] optimized by the CSPN++ module [40], ReCSNet with the refinement module RDCM achieves a reduction of 0.8mm in RMSE and 0.07 1/km in iRMSE.

E. ABLATION STUDY

We conduct an ablation study on the KITTI validation set to investigate the impact of each modification on the performance. Table 2 shows the evaluation results of the model under different settings. From the results between the baseline and method (a), it can be seen that adding semantic segmentation map and concatenating VIS imag as inputs to the encoder results in a 3.5mm reduction of RMSE. In method (b), based on method (a), we utilize two encoders to extract features of color, edge, and depth, respectively, and perform feature channel shuffle operation to mix the

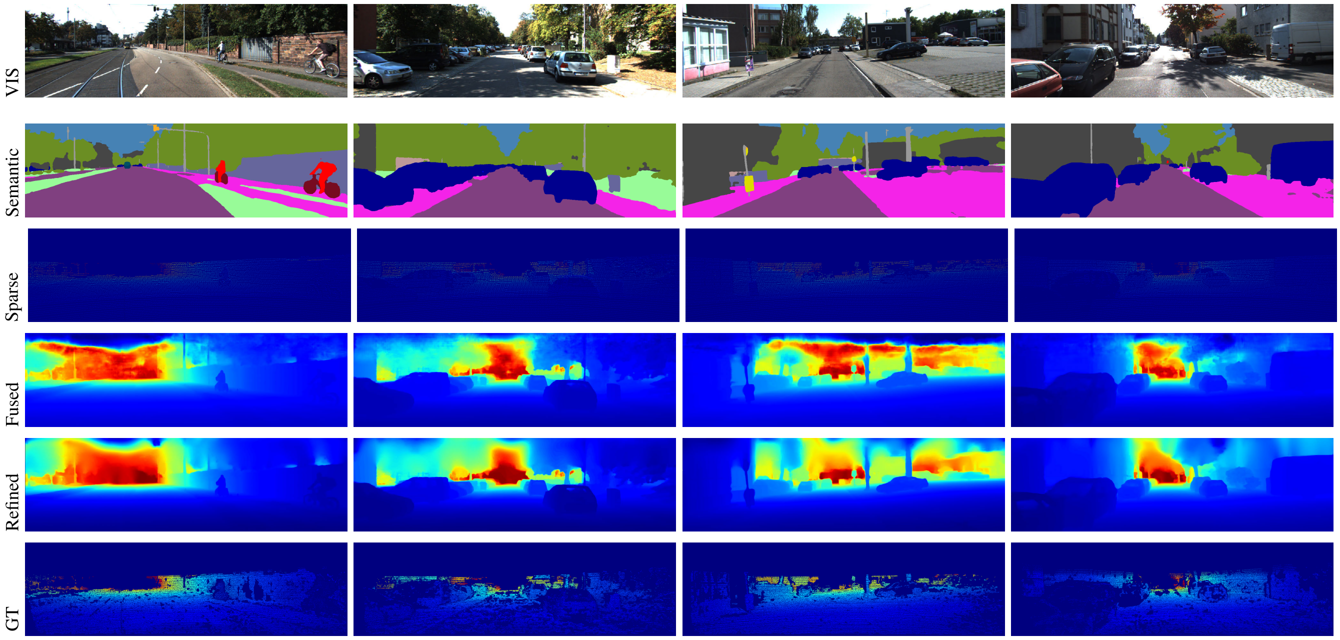


FIGURE 7. Dense depth maps predicted by the proposed ReCSNet on the KITTI depth completion validation dataset. Top to bottom: VIS image, semantic segmentation map, sparse depth map, dense depth map from the 1st stage, refined depth map, and ground truth.

TABLE 2. Ablation study on KITTI depth completion official validation dataset. ***Sem*** represents the incorporation of semantic segmentation maps as inputs, while ***CS*** denotes the utilization of feature channel shuffle mechanism to mix the features from two modalities. ***MR*** stands for multi-scale residuals and ***RDCM*** is the repetitive deformable convolution module. bold represents the best performance.

Methods	Sem	CS	MR	$RDCM$	RMSE mm	MAE mm	iRMSE 1/km	iMAE 1/km
baseline (ENet)					772.78	215.48	2.18	0.94
(a)	✓				769.20	217.02	2.23	0.96
(b)	✓	✓			764.72	214.65	2.17	0.94
(c)	✓	✓	✓		759.24	211.13	2.17	0.92
(d)	✓	✓	✓	✓	**756.38**	**208.90**	**2.15**	**0.91**

two sets of features. It can be observed from the results that this modification further reduces the RMSE by 4.5mm. As evidenced by (b) and (c), incorporating multiple-scale residuals in the decoder leads to a further decrease of 5.5mm in RMSE and a 3.5mm drop in MAE. When comparing methods (c) and (d), the values of the four indicators, namely RMSE, MAE, iRMSE and iMAE, are all reduced under the effect of RDCM to further refine the fusion depth map.

V. CONCLUSION

In this paper, we have proposed a dual-branch network for sparse depth completion based on three encoders and two decoders, named ReCSNet. We have concatenated VIS image and semantic segmentation map in VSGB to use edge information of objects in the semantic segmentation map. One encoder is used to extract color and edge features, while another encoder is used to extract depth features from the sparse depth maps. Channel shuffle is utilized to fuse the two modal features from both encoders. We have leveraged edge information in the multi-scale residuals to optimize object boundaries in the predicted depth map. Moreover, we have utilized SDGB to extract local features from the sparse depth map and the semi-dense depth map obtained by VSGB. VSGB and SDGB generate confidence maps, which are used to adaptively fuse two semi-dense depth maps. In addition, we have proposed RDCM to further refine the dense depth maps predicted by the dual-branch network. RDCM utilizes deformable convolution to perform adaptive sampling on features and improve the accuracy of the predicted depth maps. Experimental results demonstrate that ReCSNet generates high-quality dense depth maps. The ablation study shows that the proposed method reduces the RMSE value of 13.5mm by the first stage and further reduces the RMSE value of 2.5mm by the refinement stage.

Our future work includes applying sparse depth completion to 3D object detection and autonomous driving.

REFERENCES

[1] K. Wang, Z. Zhang, Z. Yan, X. Li, B. Xu, J. Li, and J. Yang, "Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16035–16044.

[2] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. So Kweon, "Non-local spatial propagation network for depth completion," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 120–136.

[3] M. Kalia, N. Navab, and T. Salcudean, "A real-time interactive augmented reality depth estimation technique for surgical robotics," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8291–8297.

[4] H.-C. Huang, C.-T. Hsieh, and C.-H. Yeh, "An indoor obstacle detection system using depth information and region growth," *Sensors*, vol. 15, no. 10, pp. 27116–27141, Oct. 2015.

[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[6] 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.

[7] 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.

[8] N. Chodosh, C. Wang, and S. Lucey, "Deep convolutional compressed sensing for LiDAR depth completion," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2019, pp. 499–513.

[9] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "PENet: Towards precise and efficient image guided depth completion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13656–13662.

[10] 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. 3308–3317.

[11] C. Zhang, Y. Tang, C. Zhao, Q. Sun, Z. Ye, and J. Kurths, "Multitask GANs for semantic segmentation and depth completion with cycle consistency," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5404–5415, Dec. 2021.

[12] D. Nazir, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "SemAttNet: Toward attention-based semantic aware guided depth completion," *IEEE Access*, vol. 10, pp. 120781–120791, 2022.

[13] Z. Yan, K. Wang, X. Li, Z. Zhang, J. Li, and J. Yang, "RigNet: Repetitive image guided network for depth completion," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 214–230.

[14] D. Hou, Y. Du, K. Zhao, and Y. Zhao, "Learning an efficient multimodal depth completion model," 2022, *arXiv:2208.10771*.

[15] Z. Huang, J. Fan, S. Cheng, S. Yi, X. Wang, and H. Li, "HMS-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion," *IEEE Trans. Image Process.*, vol. 29, pp. 3429–3441, 2020.

[16] A. Eldesokey, M. Felsberg, and F. S. Khan, "Propagating confidences through CNNs for sparse data regression," 2018, *arXiv:1805.11913*.

[17] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[18] K. Lu, N. Barnes, S. Anwar, and L. Zheng, "From depth what can you see? Depth completion via auxiliary image reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11303–11312.

[19] K. Lu, N. Barnes, S. Anwar, and L. Zheng, "Depth completion auto-encoder," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2022, pp. 63–73.

[20] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4796–4803.

[21] 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.

[22] P. Hambarde and S. Murala, "S2DNet: Depth estimation from single image and sparse samples," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 806–817, 2020.

[23] S. Ito, N. Kaneko, and K. Sumi, "Seeing farther than supervision: Self-supervised depth completion in challenging environments," in *Proc. 17th Int. Conf. Mach. Vis. Appl. (MVA)*, Jul. 2021, pp. 1–5.

[24] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11027–11036.

[25] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2361–2379, Oct. 2020.

[26] S. Zhao, M. Gong, H. Fu, and D. Tao, "Adaptive context-aware multi-modal network for depth completion," *IEEE Trans. Image Process.*, vol. 30, pp. 5264–5276, 2021.

[27] Y. Liao, L. Huang, Y. Wang, S. Kodagoda, Y. Yu, and Y. Liu, "Parse geometry from a line: Monocular depth estimation with partial laser observation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5059–5066.

[28] J. Gu, Z. Xiang, Y. Ye, and L. Wang, "DenseLiDAR: A real-time pseudo dense depth guided depth completion network," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1808–1815, Apr. 2021.

[29] 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.

[30] Y. Lin, T. Cheng, Q. Zhong, W. Zhou, and H. Yang, "Dynamic spatial propagation network for depth completion," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2022, vol. 36, no. 2, pp. 1638–1646.

[31] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[32] L. Liu, X. Song, X. Lyu, J. Diao, M. Wang, Y. Liu, and L. Zhang, "FCFR-Net: Feature fusion based coarse-to-fine residual learning for depth completion," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 3, pp. 2136–2144.

[33] M. Song, S. Lim, and W. Kim, "Monocular depth estimation using Laplacian pyramid-based depth residuals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 11, pp. 4381–4393, Nov. 2021.

[34] P.-Y. Chen, A. H. Liu, Y.-C. Liu, and Y. F. Wang, "Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2619–2627.

[35] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[36] Z. Wu, C. Shen, and A. van den Hengel, "Wider or deeper: Revisiting the ResNet model for visual recognition," *Pattern Recognit.*, vol. 90, pp. 119–133, Jun. 2019.

[37] 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.

[38] 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. 10022–10031.

[39] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," *IEEE Trans. Image Process.*, vol. 30, pp. 1116–1129, 2021.

[40] X. Cheng, P. Wang, C. Guan, and R. Yang, "CSPN++: Learning context and resource aware convolutional spatial propagation networks for depth completion," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2020, vol. 34, no. 7, pp. 10615–10622.

ZHI LIU received the B.S. degree in electronic engineering from Chang'an University, China, in 2020. He is currently pursuing the M.S. degree in electronic engineering with Xidian University, China. His research interests include image fusion and deep learning.

CHEOLKON JUNG (Member, IEEE) is a Born Again Christian. He received the B.S., M.S., and Ph.D. degrees in electronic engineering from Sungkyunkwan University, Republic of Korea, in 1995, 1997, and 2002, respectively. He was a Research Staff Member of the Samsung Advanced Institute of Technology, Samsung Electronics, Republic of Korea, from 2002 to 2007. He was a Research Professor with the School of Information and Communication Engineering, Sungkyunkwan University, from 2007 to 2009. Since 2009, he has been with the School of Electronic Engineering, Xidian University, China, where he is currently a Full Professor and the Director of the Xidian Media Laboratory. His research interests include image and video processing, computer vision, pattern recognition, machine learning, computational photography, video coding, virtual reality, information fusion, multimedia content analysis and management, and 3DTV.