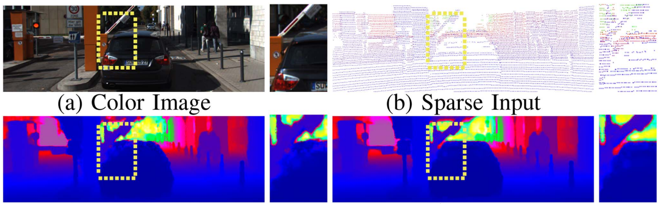
MFF-Net: Towards Efficient Monocular Depth Completion With Multi-Modal Feature Fusion

Lina Liu, Xibin Song, Jiadai Sun, *Graduate Student Member, IEEE*, Xiaoyang Lyu,
Lin Li, *Graduate Student Member, IEEE*, Yong Liu, and Liangjun Zhang

***Abstract*—Remarkable progress has been achieved by current depth completion approaches, which produce dense depth maps from sparse depth maps and corresponding color images. However, the performances of these approaches are limited due to the insufficient feature extractions and fusions. In this work, we propose an efficient multi-modal feature fusion based depth completion framework (MFF-Net), which can efficiently extract and fuse features with different modals in both encoding and decoding processes, thus more depth details with better performance can be obtained. In specific, the encoding process contains three branches where different modals of features from both color and sparse depth input can be extracted, and a multi-feature channel shuffle is utilized to enhance these features thus features with better representation abilities can be obtained. Meanwhile, the decoding process contains two branches to sufficiently fuse the extracted multi-modal features, and a multi-level weighted combination is employed to further enhance and fuse features with different modals, thus leading to more accurate and better refined depth maps. Extensive experiments on different benchmarks demonstrate that we achieve state-of-the-art among online methods. Meanwhile, we further evaluate the predicted dense depth by RGB-D SLAM, which is a commonly used downstream robotic perception task, and higher accuracy on vehicle's trajectory can be obtained in KITTI odometry dataset, which demonstrates the high quality of our depth prediction and the potential of improving the related downstream tasks with depth completion results.**

***Index Terms*—RGB-D perception, recognition.**



(a) Color Image (b) Sparse Input

(c) DenseLiDAR (d) Ours

Fig. 1. Comparison of depth completion results. Both DenseLiDAR [4] and our approach use color image (a) and sparse LiDAR depth (b) to predict dense depth map. In the zoomed area, (d) can get more details than (c), *e.g.* sharper depth edges.

I. Introduction

RECENTLY, researchers in many areas have taken depth as key information, for it can provide complementary cues in many tasks, including robotics, augmented reality, virtual reality and SLAM [1], [2], [3]. High precision depth information in centimeter-level accuracy can be obtained by LiDAR, which is commonly used for depth acquisition, such as autonomous driving and robotics. However, due to the inherent characteristics, depth information obtained by LiDAR is always sparsely distributed, which greatly limits the performance of LiDAR-based applications. Therefore, the task of depth completion is drawing more and more attention.

Feature fusion based approaches [5], [6], [7], [8] are commonly used for depth completion, which use color and sparse depth information as input, and depth completion process can be guided by color information. Various feature extraction and fusion strategies are employed to effectively improve depth completion performance. However, several problems exist. Most of the current feature fusion based approaches directly input the sparse depth map into the network for depth completion (named one-stage task). However, [9] prove that direct convolution on the sparse map can easily lead to suboptimal results, so [10], [11] first interpolate the sparse map to get the intermediate dense map, and then feed it to the network for refinement (known as two-stage task). These methods usually include one or two branches (early fusion or late fusion) for color and depth feature extraction, resulting in insufficient depth and image feature extraction. Moreover, the performance is limited due to the inherent characteristics of sparse depth input. The feature extraction of depth branch is insufficient, leading the obtained dense depth maps to suffer edge blurring. Secondly, color and depth features are extracted without information interchanging, and color guidance is missing in the depth branch, which also leads to insufficient depth feature extraction and depth details losing. Besides, concatenation and add operations are commonly used for color and depth feature fusion, and the lack of extraction and screening of key features limits the representation abilities of the fused features. Fig. 1 demonstrates the qualitative results

Manuscript received 17 August 2022; accepted 21 December 2022. Date of publication 6 January 2023; date of current version 12 January 2023. This letter was recommended for publication by Associate Editor H. Zha and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2021QNA5008. *(Corresponding authors: Xibin Song; Yong Liu.)*

Lina Liu, Xiaoyang Lyu, and Lin Li are with the State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: linaliu@zju.edu.cn; shawlyu@zju.edu.cn; lilinhitcrt@zju.edu.cn).

Yong Liu is with the State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China, and also with the Huzhou Institute of Zhejiang University, Hangzhou 310027, China (e-mail: yongliu@iipc.zju.edu.cn).

Jiadai Sun is with the School of Electronics and Information, Northwestern Polytechnical University, Xi'An 710072, China (e-mail: sunjiadai@mail.nwpu.edu.cn).

Xibin Song and Liangjun Zhang are with Robotics and Autonomous Driving Lab, Baidu Research, Beijing 100000, China (e-mail: songxibin@baidu.com; liangjunzhang@baidu.com).

Digital Object Identifier 10.1109/LRA.2023.3234776

of state-of-the-art approaches. In Fig. 1(c), the recovered depth maps often suffer from blurred edges and details.

In this paper, we propose an efficient multi-modal feature fusion based network, i.e., MFF-Net, which contains a sparse-to-dense stage and a dense-to-fine stage. The sparse-to-dense stage aims to obtain coarse dense depth maps by interpolating the sparse depth maps with only a single convolution operation. Then, the obtained coarse dense depth map and color image are input to the dense-to-fine stage. Specifically, the dense-to-fine stage consists of encoding and decoding processes. For encoding, the multi-modal features are extracted from color and coarse dense depth maps. A multi-feature channel shuffle operation (MCS) is designed to fully mix the color and depth features at multi-scale feature levels by interleaving these features at the channel level. For decoding, the extracted multi-modal features by encoding process are fused using two different fusion branches, containing concatenate fusion and multi-level weighted combination, then the features of these two decoders are fused using weighted summation. Concatenate fusion is the commonly used concatenation operation. The multi-level weighted combination operation (MLC) is proposed to fuse the features obtained by the encoding process to enhance and combine the multi-modal features with the learned weights, which can further effectively refine and fuse the color and depth features. With the operations proposed above, more effective features can be extracted, thus, more accurate results can be expected. As demonstrated in Fig. 1(d), depth maps obtained by our approach are with sharper boundaries and more depth details can be recovered.

The main contributions of the paper can be summarized as:

- We propose an efficient multi-modal feature fusion based framework, i.e., MFF-Net, which contains sparse-to-dense and dense-to-fine stages. Sparse-to-dense stage aims to obtain coarse dense depth maps with a single convolution operation, and the dense-to-fine stage further refines the coarse dense depth maps with multi-modal feature extraction and fusion strategies.
- A multi-feature channel shuffle (MCS) extraction operation is utilized for effective color and depth feature extraction, which effectively fuses the features of color and depth information at the multi-scale feature levels, and significantly improves the final performance.
- A multi-level weighted fusion (MLC) operation is utilized to further fuse the features obtained by MCS to enhance and combine the multi-modal features with the learned weights, which further refine and fuse the features extracted from color and depth information.
- Extensive experiments on different benchmarks prove that we achieve state-of-the-art among online methods. Compared to SoTA non-online work [12], our approach achieves competitive performance but runs more than 5× faster. Further RGB-D SLAM experiment demonstrates the high quality of our predicted depth and the potential of improving the downstream tasks with depth completion.

II. Related Work

A. Depth Completion

Recovering the dense depth from the sparse depth is divided into two types of methods. One uses only the sparse depth as input to achieve depth completion [13], [14], and the other uses both the sparse depth and the monocular image as input [15], [16]. In recent years, most approaches use both sparse depth and image as network input, which involves the fusion of different information. The current mainstream depth completion fusion approaches of depth and image can be divided into signal level fusion and feature level fusion.

For signal level fusion, [17], [18] directly stack the sparse depth and image before feeding into the network, which fuses them at the signal level. Some approaches [19], [20], [21] add various post-processing operations to further improve the depth completion performance. [19], [20] present a novel convolutional spatial propagation network (CSPN) for learning the affinity matrix of depth prediction. [21] proposes a non-local spatial propagation network (NLSPN) to estimate non-local neighbors of each pixel. All of the above approaches directly integrate depth and image at the signal level, and lack the integration and mutual help of semantics and feature levels.

For feature level fusion, the [4], [15] approaches use two independent encoders to extract depth and image features, and integrates the extracted features into the decoder to complete the feature-level fusion. [16] makes the depth feature guide the image to get the dense depth by changing the convolution kernel. [22] reports a recover architecture to fuse the features in multi-level. The above methods merge depth and image at the macro-level, lacking the sufficient fusion at the micro-level such as the channel level. In this work, the extracted multi-modal features exchange information at the channel level to make the fusion more sufficient and get better results.

When sparse depth is input to networks directly, convolution on sparse map can easily lead to suboptimal results [9]. To solve this problem, some approaches [11], [23] first fill the sparse depth with non-zero values, then input the filled depth into the network. [11] fills the sparse depth with nearest-neighbor interpolation. [24] expands the depth value from sparse cues while estimating the confidence of the expanded region. To improve the filling performance and the running speed of the model while reducing the network parameters as much as possible, in this work, we use a simple and efficient convolution layer to fill the sparse depth coarsely, then optimize the subsequent depth completion network.

B. Feature Fusion

Feature fusion includes homogenous feature fusion, such as image fusion, and heterogeneous feature fusion, such as multi-modal feature fusion. For image fusion, images from different sensors are fused using different strategies, such as parallax attention based images pairs feature fusion [25], image features addition fusion and L1-norm and soft-max fusion [26]. For multi-modal feature fusion, some methods consider the inter-modality and intra-modality correlation or design a new search space tailored for the multi-modal fusion [27]. These methods aim to design different rational ways to fuse multi-modal features better. The common points of image fusion and multi-modal color depth feature fusion are to merge two different maps into a new map using addition, concatenation, etc. In this work, the multi-modal features are fused at the channel level to obtain fully mixed information in the encoder. An attention-based multi-modal color and depth feature fusion is designed in the decoder, and the extracted features are fused using these new strategies to get more efficient depth completion results.

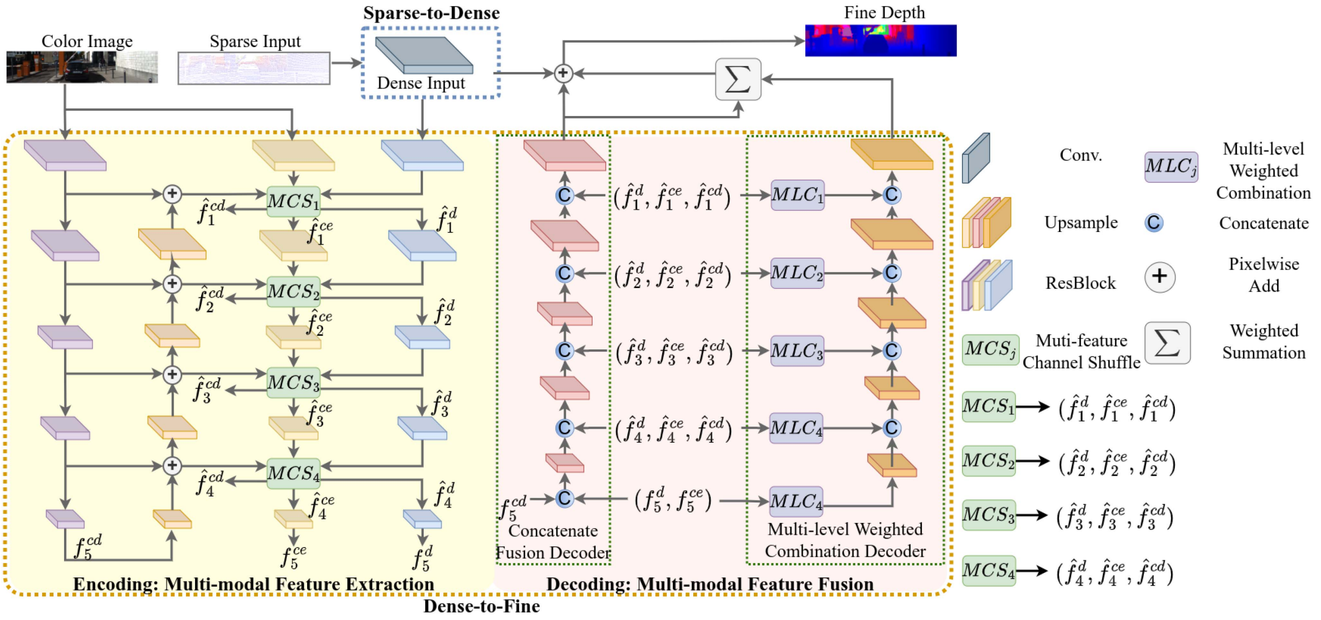

Sparse-to-Dense
Color Image
Sparse Input
Dense Input
Fine Depth
MCS_1
MCS_2
MCS_3
MCS_4
$\hat{f}_1^{cd}$
$\hat{f}_1^{ce}$
$\hat{f}_1^{d}$
$\hat{f}_2^{cd}$
$\hat{f}_2^{ce}$
$\hat{f}_2^{d}$
$\hat{f}_3^{cd}$
$\hat{f}_3^{ce}$
$\hat{f}_3^{d}$
$\hat{f}_4^{cd}$
$\hat{f}_4^{ce}$
$\hat{f}_4^{d}$
f_5^{cd}
f_5^{ce}
f_5^{d}
$(\hat{f}_1^d, \hat{f}_1^{ce}, \hat{f}_1^{cd})$
$(\hat{f}_2^d, \hat{f}_2^{ce}, \hat{f}_2^{cd})$
$(\hat{f}_3^d, \hat{f}_3^{ce}, \hat{f}_3^{cd})$
$(\hat{f}_4^d, \hat{f}_4^{ce}, \hat{f}_4^{cd})$
(f_5^d, f_5^{ce})
MLC_1
MLC_2
MLC_3
MLC_4
MLC_4
Concatenate Fusion Decoder
Multi-level Weighted Combination Decoder
Encoding: Multi-modal Feature Extraction
Decoding: Multi-modal Feature Fusion
Dense-to-Fine
Conv.
Upsample
ResBlock
MCS_j Muti-feature Channel Shuffle
MLC_j Multi-level Weighted Combination
C Concatenate
+ Pixelwise Add
∑ Weighted Summation
$MCS_1 \rightarrow (\hat{f}_1^d, \hat{f}_1^{ce}, \hat{f}_1^{cd})$
$MCS_2 \rightarrow (\hat{f}_2^d, \hat{f}_2^{ce}, \hat{f}_2^{cd})$
$MCS_3 \rightarrow (\hat{f}_3^d, \hat{f}_3^{ce}, \hat{f}_3^{cd})$
$MCS_4 \rightarrow (\hat{f}_4^d, \hat{f}_4^{ce}, \hat{f}_4^{cd})$

Fig. 2. Overview of our framework, which contains two stages: sparse-to-dense stage (blue dotted area) and dense-to-fine stage (orange dotted area). A simple convolution layer is used in sparse-to-dense stage, and dense-to-fine stage contains encoding and decoding processes. Multi-modal features extraction and fusion are utilized in sparse-to-dense and dense-to-fine stages.

III. Method

Different from previous approaches, we define depth completion as a two-stage task, including a sparse-to-dense stage and a dense-to-fine stage. Fig. 2 demonstrates the pipeline of our approach. First, the dense depth map is obtained by the sparse-to-dense stage with a single convolution operation. Second, using the dense depth map as input, the refined dense depth map is recovered by the dense-to-fine stage. In specific, a multi-feature channel shuffle (MCS) extraction operation, a multi-level weighted combination (MLC) operation and a residual learning strategy are implemented in the dense-to-fine stage. Moreover, using MCS extraction operation, different modal of features with better representative abilities from color encoding, color decoding and depth multi-modal features can be effectively extracted. Meanwhile, with MLC operation, different multi-modal features obtained from MCS can be sufficiently fused, to obtain better depth completion results. Finally, the residual learning strategy further improves the quality of depth completion by learning residual information.

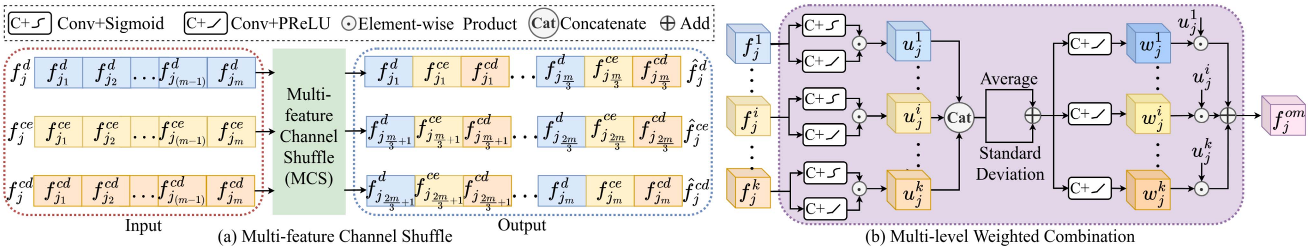
A. Sparse-to-Dense

Uhrig et al.,[9] proves that direct convolution on the sparse map can easily lead to suboptimal results. Therefore, in the sparse-to-dense stage, many handcrafted ways first interpolate the sparse depth map to the dense, such as nearest-neighbor and bilinear interpolations[10], [11]. [23] also proves that deep convolutional neural network (DCNN) based approaches can get better dense depth than handcrafted ways. In this paper, in order to improve the filling performance and improve the running speed of the model while reducing the network parameters as much as possible, we use a single convolution layer for dense depth input acquisition. The input is a sparse depth map, and the obtained dense depth map d^{sd} is input to the subsequent dense-to-fine stage.

B. Dense-to-Fine

The inputs of the dense-to-fine stage are the color image and the corresponding dense depth map obtained by the sparse-to-dense stage, which provide consecutive information in the training process. The dense-to-fine stage contains encoding and decoding processes. As shown in Fig. 2, the encoding process means the multi-modal feature extraction, which has three branches that different modal of features are extracted. Note that to sufficiently extract features from a color image, two branches are used for color feature extraction. Meanwhile, in the encoding process, inspired by [28], multi-feature channel shuffle (MCS) is exploited to enhance the extracted features, which effectively fuse the multi-modal features at the multi-scale feature levels. The MCS operation guarantees that the multi-modal features extracted from depth and color images are exchanged and mixed at the channel level sufficiently, thus, better features can be obtained. The decoding process means the multi-modal feature fusion, which contains two decoding branches: concatenate fusion and multi-level weighted combination (MLC), to sufficiently fuse the extracted multi-modal features. Note that MLC is employed to further fuse features with different modalities effectively and sufficiently.

1) Encoding: Multi-Modal Feature Extraction: [18], [20] propose to extract color and depth features with commonly used backbones, such as ResNet [29], etc. In these strategies, a single feature extractor is usually used to extract features by stacking the color image and sparse depth directly, and the information from different modalities is performed with the same feature extraction. Limited information exchanges are contained in these processes, and it is impossible to extract modal-specific features to a certain extent. Meanwhile, [15] uses two feature extractors to extract features from the color image and sparse depth separately, then combine these features using concatenate or add operations in the decoding process. However, these feature



(a) Multi-feature Channel Shuffle

(b) Multi-level Weighted Combination

Fig. 3. (a) The proposed Multi-feature Channel shuffle (MCS) operation. The blue f_j^d, yellow f_j^{ce} and orange f_j^{cd} denote depth, color encoding and color decoding features, respectively. The MCS obtains new features $\hat{f}_j^d$, $\hat{f}_j^{ce}$ and $\hat{f}_j^{cd}$ by feature mixing in channel level, and returns to their respective convolutions for the next step. (b) The proposed Multi-level Weighted Combination (MLC) operation. The blue f_j^1, yellow f_j^i and orange f_j^k features are enhanced and fused by the proposed operation, and get the fused feature f_j^{om} for the next multi-modal feature fusion process.

extraction processes cannot take advantage of the consistency of color and depth information, i.e., feature information at the same scale level. Therefore, features of different modalities cannot interact at the same scale level, and more representative features cannot be obtained. Since more representative features are not available, the performances of these methods are limited.

Inspired by [28], to make full use of the multi-modal features from the color image and depth map, a multi-feature channel shuffle feature extraction strategy (MCS) is utilized in the dense-to-fine stage. [30] extracts the two branch color and depth features with channel shuffle operation. To further enhance the multi-modal feature extraction, in this paper, we use three branches in the encoding process to extract the multi-modal features from color encoding, color decoding and depth encoding features. Then MCS is used to these three branches with different modalities at multi-scale channel levels.

Multi-feature Channel Shuffle (MCS): Fig. 3(a) shows the process of multi-feature channel shuffle operation. Given the depth encoding feature of the j-th convolution block, i.e., $f_j^d = \{f_{j_1}^d, \ldots, f_{j_m}^d\}$, and the color encoding and decoding features are $f_j^{ce} = \{f_{j_1}^{ce}, \ldots, f_{j_m}^{ce}\}$ and $f_j^{cd} = \{f_{j_1}^{cd}, \ldots, f_{j_m}^{cd}\}$ respectively, where m is the number of channels. The corresponding outputs after MCS can be obtained as follows:

$$\begin{aligned}
\hat{f}_j^d &= \left\{ f_{j_1}^d, f_{j_1}^{ce}, f_{j_1}^{cd}, \ldots, f_{j_{\frac{m}{3}}}^d, f_{j_{\frac{m}{3}}}^{ce}, f_{j_{\frac{m}{3}}}^{cd} \right\}, \\
\hat{f}_j^{ce} &= \left\{ f_{j_{\frac{m}{3}+1}}^d, f_{j_{\frac{m}{3}+1}}^{ce}, f_{j_{\frac{m}{3}+1}}^{cd}, \ldots, f_{j_{\frac{2m}{3}}}^d, f_{j_{\frac{2m}{3}}}^{ce}, f_{j_{\frac{2m}{3}}}^{cd} \right\}, \\
\hat{f}_j^{cd} &= \left\{ f_{j_{\frac{2m}{3}+1}}^d, f_{j_{\frac{2m}{3}+1}}^{ce}, f_{j_{\frac{2m}{3}+1}}^{cd}, \ldots, f_{j_m}^d, f_{j_m}^{ce}, f_{j_m}^{cd} \right\}, \qquad (1)
\end{aligned}$$

where $\hat{f}_j^d$, $\hat{f}_j^{ce}$ and $\hat{f}_j^{cd}$ denote depth encoding, color encoding and color decoding features after MCS operation.

After the features are mixed by MCS, the enhanced multi-modal feature maps $\hat{f}_j^d$, $\hat{f}_j^{ce}$ and $\hat{f}_j^{cd}$ are generated, which are the input of the $(j+1)$-th convolution block.

2) Decoding. Multi-Modal Feature Fusion: This strategy is used in the decoding process with the features obtained by the encoding process as input. The decoding process has two decoders, containing concatenate fusion decoder and multi-level weighted combination decoder. For concatenate fusion decoder, the commonly used concatenation in decoder [16] is used here to fuse the features obtained by multi-feature channel shuffle. For the multi-level weighted combination decoder, inspired by [31], in order to fuse the multi-modal features sufficiently, we propose an effective multi-level weighted combination operation (MLC) to further fuse the features obtained by multi-feature channel shuffle.

Specifically, in the decoding process, two up-sampling feature fusion U^{cat} and U^{mlc} are exploited to fuse the multi-modal features in different ways, where U^{cat} means the concatenate fusion decoder, U^{mlc} means the multi-level weighted combination decoder. d^{cat} and d^{mlc} are the fined residual depth maps generated from U^{cat} and U^{mlc}, respectively. f^{cat} and f^{mlc} are the multi-scale features in U^{cat} and U^{mlc}, where $f^{cat} = \{f_1^{cat}, \ldots, f_n^{cat}\}$, $f^{mlc} = \{f_1^{mlc}, \ldots, f_n^{mlc}\}$, n is the number of up-sample blocks in U^{cat} and U^{mlc}. The decoding process can be formulated as follows:

$$\begin{aligned}
f_n^{cat} &= U_n^{cat}(Cat(f_n^d, f_n^{ce}, f_n^{cd})), \\
f_j^{cat} &= U_j^{cat}(Cat(f_{(j+1)}^{cat}), \hat{f}_j^d, \hat{f}_j^{ce}, \hat{f}_j^{cd})), \\
d^{cat} &= CB(Cat(f_1^{cat}, f_0^d)), \\
f_n^{mlc} &= U_n^{mlc}(MLC_n(f_n^d, f_n^{ce})), \\
f_j^{mlc} &= U_j^{mlc}(Cat(f_{(j+1)}^{mlc}), MLC_j(\hat{f}_j^d, \hat{f}_j^{ce}, \hat{f}_j^{cd}))), \\
d^{mlc} &= CB(f_1^{mlc}), \qquad (2)
\end{aligned}$$

where $j \in [1, n-1]$, Cat means concatenation operation, CB means ConvBlock layer, $U^{cat} = \{U_1^{cat}, \ldots, U_n^{cat}\}$, $U^{mlc} = \{U_1^{mlc}, \ldots, U_n^{mlc}\}$, $MLC = \{MLC_1, \ldots, MLC_n\}$.

Multi-level Weighted Combination (MLC): MLC aims to solve the problem of insufficient feature fusion. Fig. 3(b) demonstrates the process of the multi-level weighted combination operation. The MLC is exploited to enhance and combine the multi-modal features with learned weights. In specific, the inputs of the MLC are the multi-modal features obtained from the encoding process, containing depth encoding, color encoding and color decoding features. The output of the MLC is defined as f^{om}, where $f^{om} = \{f_1^{om}, \ldots, f_n^{om}\}$. The MLC operation is defined as:

$$\begin{aligned}
u_j^i &= \Theta(Conv3^i(f_j^i)) \odot \Psi(Conv3^i(f_j^i)), i \in [1, k] \\
u_j &= Cat(u_j^1, \ldots, u_j^k), \\
w_j &= Avg(u_j) + Var(u_j), \\
\omega_j^i &= \Psi(Conv1^i(w_j)), \\
f_j^{om} &= \omega_j^1 \odot u_j^1 + \cdots + \omega_j^k \odot u_j^k, \qquad (3)
\end{aligned}$$

where $j \in [1, n]$, n is the number of convolution blocks in U^{mlc} which set as 5. Θ and Ψ mean Sigmoid and PReLU activate functions. $Conv3^i$ and $Conv1^i$ are the 3×3 and 1×1 convolution layers of the $i\text{-}th$ feature. $\odot$ denotes element-wise product. Cat, Avg and Var denote concatenate, average and variance operations. In our setting,

$$f_n^i = \{f_j^d, f_j^{ce}, f_j^{cd}\}, j \in [1, n-1],$$
$$f_n^i = \{f_n^d, f_n^{ce}\}, \tag{4}$$

where $i \in [1, k]$, $k = 2, 3$.

3) Residual Learning: In the dense-to-fine stage, given the depth maps obtained by the decoding process are d^{co} and d^{mo}, where d^{co} is the summation of the residual depth map d^{cat} and the result of sparse-to-dense stage d^{sd}, d^{mo} is the summation of the d^{mlc} and d^{sd}. The final depth d^o of the dense-to-fine stage can be obtained by the weighted summation of d^{co} and d^{mo}. The process can be formulated as:

$$d^{co} = d^{cat} + d^{sd}, \quad d^{mo} = d^{mlc} + d^{sd},$$
$$d^o = \alpha \cdot d^{co} + (1 - \alpha) \cdot d^{mo}, \tag{5}$$

where $\alpha \in [0, 1]$ is a learnable parameter.

C. Loss Function

The proposed two-stage network is trained in an end-to-end manner, and smooth L1 loss [32] and L2 loss are used here. The smooth L1 loss is more robust to outliers, which can reduce the appearance of outliers in the overall depth range. L2 loss is differentiable everywhere, and as the error decreases, the gradient also decreases, which is conducive to faster convergence. Therefore, we combine the smooth L1 loss and L2 loss, and the total loss function is defined as:

$$SL1 = \beta_1 \cdot s_{L1}(d^o, d^{gt}) + \beta_2 \cdot s_{L1}(d^{co}, d^{gt}) + \beta_3 \cdot s_{L1}(d^{mo}, d^{gt}),$$
$$L2 = \beta_1 \cdot ||d^o - d^{gt}||_2 + \beta_2 \cdot ||d^{co} - d^{gt}||_2 + \beta_3 \cdot ||d^{mo} - d^{gt}||_2,$$
$$\text{Loss} = SL1 + L2, \tag{6}$$

where d^{gt} is the depth map ground truth, $\beta_1 = \beta_2 = \beta_3 = 1$. s_{L1} is smooth L1 loss [32], $|| \cdot ||_2$ denotes mean-square error loss.

IV. Experiments

In this section, we evaluate the performance of our approach on diverse publicly available datasets, including the KITTI and NYUDv2 datasets. And further experiments on RGB-D SLAM also verify the high quality of our depth prediction.

A. Datasets and Implementation Details

Outdoor: The KITTI dataset [33] is a large outdoor autonomous driving dataset. We use the KITTI depth completion dataset for evaluation, where the training set contains 85 k frames, the selected validation set contains 1 k, and the test set contains 1 k. Following [16], for training and testing, the color and depth images are bottom-cropped to 256×1216.

Indoor: The NYUDv2 [34] dataset consists of video sequences of various indoor scenes recorded by the RGB-D cameras of Microsoft Kinect. A subset of 50 K images from the official training split is utilized as training data. The evaluation set contains 654 official labeled images. Following [16], the original frames 480×640 are half down-sampled to 256×320 for training. The prediction of the network is center-cropped to 228×304 during evaluation for a fair comparison.

TABLE I
Performance Comparison on KITTI Benchmark Ranked by the RMSE (in Mm)

Method	Online	RMSE↓	MAE↓	iRMSE	iMAE	FPS^1	FPS^2
B-ADT [5]	No	1480.36	298.72	4.16	1.23	8.3	-
CSPN [20]	No	1019.64	279.46	2.93	1.15	1.0	-
CSPN++ [19]	No	743.69	209.28	2.07	0.90	5.0	-
NLSPN [21]	No	741.68	199.59	1.99	0.84	4.5	6.3
GuideNet [16]	No	736.24	218.83	2.25	0.99	7.1	8.7
RigNet [22]	No	712.66	203.25	2.08	0.90	5.0	-
SemAttNet [12]	No	709.41	205.49	2.03	0.90	5.0	2.8
DySPN [36]	No	**709.12**	**192.71**	**1.88**	**0.82**	6.3	-
PSM [37]	Yes	1239.84	298.30	3.76	1.21	16.7	-
STD(gd) [18]	Yes	814.73	249.95	2.80	1.21	12.5	-
GAENet [38]	Yes	773.90	231.29	2.29	1.08	20.0	-
ABCD [7]	Yes	764.61	220.86	2.29	0.97	40.3	-
DeepLiDAR [15]	Yes	758.38	226.50	2.56	1.15	14.3	14.9
DenseLiDAR [4]	Yes	755.41	214.13	2.25	0.96	50.0	-
MDANet [39]	Yes	738.23	214.99	**2.12**	0.99	33.3	25.1
FCFR-Net [30]	Yes	735.81	217.15	2.20	0.98	10.0	10.0
PENet [35]	Yes	730.98	210.55	2.17	0.94	31.3	25.6
Ours	Yes	**719.85**	**208.11**	2.21	**0.94**	19.6	14.9

Implementaion Details: For both indoor and outdoor scenes, the sparse-to-dense and dense-to-fine stages are trained in an end-to-end manner. The proposed model is trained on NVIDIA V100 GPU. We use Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The weight decay is 0.05. The batch size is set to 8. The initial learning rate is $1e^{-3}$ and decreases by 0.5 every 5 epochs. The total model is trained for 20 epochs, which takes about 70 h to train on the KITTI and 12 h on the NYUDv2.

B. Evaluation on Outdoor Dataset

Table I shows the quantitative comparison and runtime of our method and baselines on the KITTI depth completion benchmark. Follow [4], [35], FPS^1 is quoted from the leaderboard. For a fair comparison, FPS^2 is tested on our single 1080Ti GPU with source codes released by the authors. Since the sampling frequency of most LiDARs is 10 Hz, we refer to methods faster than 10 Hz as online methods. Our method achieves the state-of-the-art (SoTA) results among all online methods, where the RMSE error drops from 730.98 to 719.85. Compared to SoTA non-online work DySPN [36] and SemAttNet [12], our method achieves competitive results but runs more than 3× faster for FPS^1 and more than 5× faster for FPS^2, getting SoTA performance under the online conditions and creating more possibilities for downstream tasks. In Fig. 4, the qualitative comparison with typically online methods is also consistent with the quantitative analysis. Our results have more complete boundaries and object details, which proves that the multi-modal feature fusion is more helpful for efficient depth completion.

C. Evaluation on Indoor Dataset

Table II is the quantitative comparison result on the indoor NYUDv2 dataset. Following CSPN++ [19], 500 random points are sampled as sparse depth input. The results show that when the random sampling points are 500, our approach is better than other SoTA methods in almost all metrics. Compared with CSPN [20] and DeepLiDAR [15], the fusion of color image and depth at the multi-modal feature level is significantly better

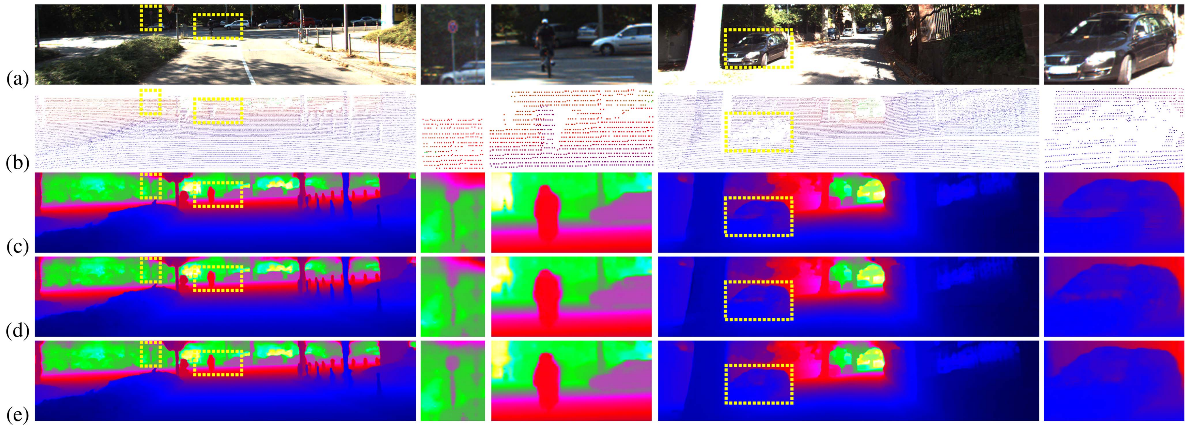


Fig. 4. Qualitative comparison on KITTI test set. (a) Image, (b) Sparse input, (c) DeepLiDAR [15], (d) STD(gd) [18], (e) Ours. The results are from the KITTI depth completion leaderboard, where depth images are colorized along with depth range.

TABLE II
PERFORMANCE COMPARISON ON NYUDv2 DATASET. ALL METHODS USE 500 SAMPLED DEPTH POINTS AS THE SPARSE INPUT

Method	RMSE (m)	REL (m)	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
STD [40]	0.230	0.044	97.1	99.4	99.8
STD(gd) [18]	0.123	0.026	99.1	**99.9**	**100.0**
CSPN [20]	0.117	0.016	99.2	**99.9**	**100.0**
CSPN++ [19]	0.116	-	-	-	-
DeepLiDAR [15]	0.115	0.022	99.3	**99.9**	**100.0**
FCFR-Net [30]	0.106	0.015	**99.5**	**99.9**	**100.0**
PRNet [17]	0.104	**0.014**	99.4	**99.9**	**100.0**
Ours	**0.100**	0.015	**99.5**	**99.9**	**100.0**

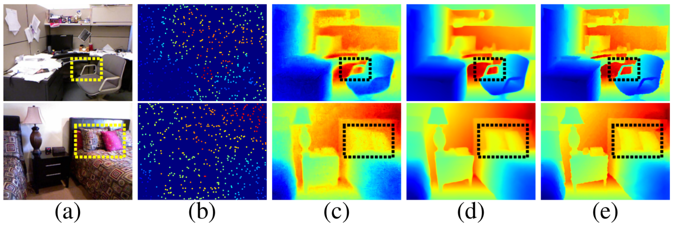


Fig. 5. Qualitative comparison on NYUDv2. (a) Color image, (b) Dilated sparse input for visualization, (c) STD(gd) [18], (d) Ours result, (e) Ground Truth. The dotted box areas show the recovery of object details, which (d) is able to capture more complete chair armrest details and full pillow edges than (c).

TABLE III
ABLATION STUDY ON KITTI DEPTH COMPLETION SELECTED VALIDATION DATASET

Name	S/D	R	E	D	MCS	MLC	WS	Loss	RMSE
SI	S		1	1				L2+SL1	928.26
DI	D		1	1				L2+SL1	909.68
DR	D	✓	1	1				L2+SL1	860.66
$D2E$	D	✓	2	1				L2+SL1	841.51
$D3E$	D	✓	3	1				L2+SL1	840.71
$D2ECS$	D	✓	2	1	✓			L2+SL1	823.34
$D3ECS$	D	✓	3	1	✓			L2+SL1	819.32
$D2D$	D	✓	3	2	✓			L2+SL1	815.98
$DMLC$	D	✓	3	2	✓	✓		L2+SL1	805.92
$DL2$	D	✓	3	2	✓	✓	✓	L2	809.49
$DL1L2$	D	✓	3	2	✓	✓	✓	L2+L1	807.53
$D2CM$	D	✓	2	2	CS	✓	✓	L2+SL1	811.85
ALL	D	✓	3	2	✓	✓	✓	L2+SL1	803.25

S/D: sparse/dense input. *R*: residual learning, *E*: extractor, *D*: decoder. *MCS*: multi-feature channel shuffle. *MLC*: multi-level weighted combination. *WS*: weighted summation. L2: l2 loss. SL1: smooth l1 loss.

than that at the signal level, proving the effectiveness of the multi-modal feature fusion in the indoor scenario. The qualitative comparison result is shown in Fig. 5. The visualization results show that our approach can capture more details of the edge structure and depth of the object. Specifically, our approach (d) is able to capture more complete chair armrest details and full pillow edges than STD(gd) (c), which proves the superiority of our approach in indoor scenes.

D. Ablation Studies

For fast training, the depth maps are sorted in time series and uniformly sampled $1/4$ of the data as mini-training data for ablation studies. Table III shows the results of ablation studies, containing sparse or dense input (S/D), residual learning (R), the number of extractors (E), the number of decoders (D), multi-feature channel shuffle (MCS), multi-level weighted combination (MLC) and different loss functions (L1, L2 and smooth L1 loss). As shown in Table III, the performance is improved when adding each module to the network, proving the effectiveness of the proposed module and method. Specifically, the final RMSE performance is 13.5% better than that of the baseline method in the first line. Lines SI and DI show the results (928.26 mm and 909.68 mm) of sparse and dense depth input under the same framework. This indicates that the consecutive information in dense depth can reduce the loss of the valid values during convolution. Lines $D3E$ and $D3ECS$ show the results (840.71 mm and 819.32 mm) without and with MCS. The results demonstrate that sufficient channel exchange between depth and image features with different modalities can improve the fusion and representation capabilities of the multiple types of information and play a role in mutual guidance. Lines $D2D$ and $DMLC$ compare the difference (815.98 mm and 805.92 mm) before and after adding the MLC. This shows that the proposed MLC can enhance and combine the multi-modal features at multiple levels to obtain new fusion features, which ultimately help to obtain better depth completion results. Lines ALL and $D2CM$ show a comparison (803.25 mm and 811.85 mm) of our approach and replacing MCS with "channel shuffle" in [30], proving that multi-modal feature shuffle can make feature extraction more

TABLE IV
QUANTATIVE RESULTS ON KITTI ODOMETRY SEQUENCE

t_{rel}/r_{rel}	00	01	02	03	04	05	06	07	08	09	10
Monocular (M)	0.80/1.32	-	-	0.49/0.37	0.70/0.25	0.88/0.90	1.10/0.42	0.89/0.23	3.06/1.41	-	1.01/9.16
Stereo (S)	0.71/0.25	1.48/0.21	0.80/0.26	0.80/0.20	0.47/0.15	0.39/0.16	0.47/0.15	0.49/0.28	1.03/0.30	0.89/0.26	0.66/0.30
M+DeepLiDARD [15]	0.83/0.39	**7.41/2.43**	1.00/0.37	1.25/0.39	3.83/3.18	1.30/0.36	3.52/1.08	1.08/0.45	2.23/0.60	2.44/0.47	2.83/1.01
M+DenseLiDARD [4]	0.81/0.38	45.70/9.24	1.03/0.39	1.24/0.37	**1.17/1.29**	0.51/0.32	0.57/0.39	0.46/0.38	1.33/0.48	0.94/0.35	0.89/0.53
M+OursD	**0.69/0.30**	10.79/3.08	**0.76/0.28**	**0.95/0.33**	1.51/1.35	**0.42/0.24**	**0.45/0.25**	**0.41/0.32**	**1.25/0.41**	**0.80/0.30**	**0.66/0.35**

'-' denotes tracking failure. t_{rel} is in %, r_{rel} is in deg/100 m.

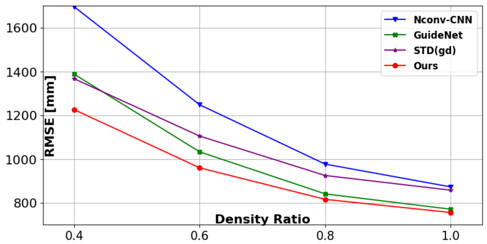


Fig. 6. Different input LiDAR point density ratio performances of different methods in RMSE (mm). The performance of our method, STD(gd) [18] and the GuideNet [16] degrades more gently compared to that of NConv-CNN [41].

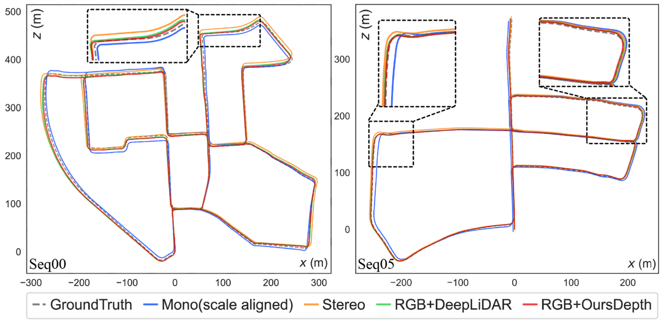


Fig. 7. Estimated trajectories of KITTI 00 and 05 sequences.

adequate, thus improving the final result. To further prove the effectiveness of the single convolution layer in the first stage, we make a comparison that our method with convolution operation for sparse depth map interpolation outperforms our methods with the nearest-neighbor for sparse depth map interpolation (803.25 mm v.s. 826.17 mm), which proves the effectiveness.

E. Robustness in Input Point Density

The sparse input of the KITTI dataset is the collected 64-line Velodyne LiDAR. In more practical applications, the inputs have 32-line, 16-line or even more sparse LiDAR. To demonstrate the performance robustness of our approach under different input point densities, we compare the performance of two state-of-the-art methods and ours under the same setting. We evaluate the performance differences of these different approaches on the KITTI dataset with different input densities, and change the density of the LiDAR input on the KITTI validation set (1 k images) to analyze the effect of sparsity on the final results. Specifically, we divide the LiDAR input from KITTI into 5 density levels, where different density levels indicate different numbers of LiDAR points input into the network. Following [16], we randomly sample the original LiDAR points according to the given density ratio, and input the sampled points into the network, where the density ratios are 0.4, 0.6, 0.8 and 1.0. All methods are trained from scratch on the KITTI dataset. In the evaluation, we only change the density ratio of the input to the model trained with 64-line LiDAR. Fig. 6 shows the RMSE performance of Nconv-CNN [41], STD(gd) [18], GuideNet [16] and ours under different density ratios. As the density ratio decreases, the performance of Nconv-CNN drops sharply, and the performance gap between GuideNet and ours gradually increases. What's more, our performance consistently outperforms the other methods on all density ratios, proving the superiority and robustness of our method with different input density ratios.

F. Application in RGB-D SLAM

To further evaluate the quality of the depth maps produced by our method, we apply the depth maps to a common robotics task: RGB-D SLAM. Following DenseLiDAR [4], we choose ORB-SLAM2 [42] as the evaluation baseline, which is a popular real-time SLAM library for Monocular, Stereo and RGB-D cameras. We evaluate our method on the KITTI odometry dataset, where for RGB-D SLAM, we input the depth completed by ours and other state-of-the-art methods (DeepLiDAR [15] and DenseLiDAR [4]) into the ORB-SLAM2 to compare the estimation results. We keep the same settings as DenseLiDAR, run the KITTI odometry 00-10 sequences 25 times, and take the average $t_{rel}(\%)$ and $R_{rel}(deg/100\mathrm{m})$ errors as the final results shown in Table IV.

Because the depth value of each ORB feature point is necessary for trajectory tracking in RGB-D mode, study [4] has shown that if 'D' is a sparse depth map, it will fail. As shown in Table IV, when 'D' is a dense depth map completed with a sparse depth map, excellent positioning results can be obtained. The accuracy of the dense depth and the positioning results obtained by ORB-SLAM2 are positively correlated as a whole. Specifically, except for the 01 sequence, our method can obtain robust results on different sequences, outperform the monocular method and other RGB-D methods using depths from existing depth completion methods, and is comparable to the stereo method. The 01 sequence is a challenging highway scene, containing a lot of weakly textured areas such as the sky, and having fewer structural objects. Therefore, due to the limited accuracy, the dense depths obtained by the depth completion methods are not as accurate as the stereo method after inputting them into the RGB-D SLAM. The qualitative trajectory comparison on selected 00 and 05 sequences are shown in Fig. 7, our method is much closer to ground truth in the overall trajectory. Specifically, in the enlarged dashed area, our trajectory is almost identical to the ground truth trajectory compared with other methods, proving the effectiveness of our method on the downstream tasks like SLAM.

V. CONCLUSION

In this paper, we propose a multi-modal feature fusion based framework (MFF-Net) for depth completion, which consists of two stages, i.e., a sparse-to-dense stage and a dense-to-fine stage. The sparse-to-dense stage fills the sparse depth into a dense depth, which provides a consecutive dense input depth map to the dense-to-fine stage, thereby improving the performance of depth completion. Meanwhile, in the dense-to-fine stage, in order to fuse the depth and color information and obtain more useful multi-modal features for depth completion in both encoding and decoding processes, we propose multi-feature channel shuffle and multi-level weighted combination operations, thus better depth completion results with more depth details and sharper boundaries can be expected. Extensive experiments across indoor and outdoor benchmarks demonstrate the superiority of our approach over state-of-the-art online approaches. And further experiment on RGB-D SLAM not only demonstrates the high quality of our depth prediction, but also proves the potential of improving the related downstream tasks with depth completion results.

References

[1] K. T. Giang, S. Song, D. Kim, and S. Choi, “Sequential depth completion with confidence estimation for 3D model reconstruction,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 327–334, Apr. 2021.

[2] A. Wong, X. Fei, S. Tsuei, and S. Soatto, “Unsupervised depth completion from visual inertial odometry,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1899–1906, Apr. 2020.

[3] J. Jeon, H. Lim, D.-U. Seo, and H. Myung, “Struct-MDC: Mesh-refined unsupervised depth completion leveraging structural regularities from visual SLAM,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6391–6398, Jul. 2022.

[4] 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. 1808–18915, Apr. 2021.

[5] Y. Yao, M. Roxas, R. Ishikawa, S. Ando, J. Shimamura, and T. Oishi, “Discontinuous and smooth depth completion with binary anisotropic diffusion tensor,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5128–5135, Oct. 2020.

[6] K. Ryu, K.-i. Lee, J. Cho, and K.-J. Yoon, “Scanline resolution-invariant depth completion using a single image and sparse LiDAR point cloud,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6961–6968, Oct. 2021.

[7] Y. Jeon, H. Kim, and S.-W. Seo, “ABCD: Attentive bilateral convolutional network for robust depth completion,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 81–87, Jan. 2022.

[8] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli, “Aerial single-view depth completion with image-guided uncertainty estimation,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1055–1062, Apr. 2020.

[9] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, “Sparsity invariant CNNs,” in *Proc. Intl. Conf. 3D Vis.*, 2017, pp. 11–20.

[10] 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. Automat.*, 2017, pp. 5059–5066.

[11] Z. Chen, V. Badrinarayanan, G. Drozdov, and A. Rabinovich, “Estimating depth from RGB and sparse sensing,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 176–192.

[12] D. Nazir, M. Liwicki, D. Stricker, and M. Z. Afzal, “SemAttNet: Towards attention-based semantic aware guided depth completion,” 2022, *arXiv:2204.13635*.

[13] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, “Uncertainty-aware CNNs for depth completion: Uncertainty from beginning to end,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12011–12020.

[14] Y. Wang, Y. Dai, Q. Liu, P. Yang, J. Sun, and B. Li, “CU-Net: LiDAR depth-only completion with coupled U-Net,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11476–11483, Oct. 2022.

[15] J. Qiu et al., “DeepLiDAR: Deep surface normal guided depth prediction for outdoor scene from sparse LiDAR data and single color image,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3308–3317.

[16] 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.

[17] B.-U. Lee, K. Lee, and I. S. Kweon, “Depth completion using plane-residual representation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13916–13925.

[18] F. Ma, G. V. Cavalheiro, and S. Karaman, “Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 3288–3295.

[19] X. Cheng, P. Wang, C. Guan, and R. Yang, “CSPN: Learning context and resource aware convolutional spatial propagation networks for depth completion,” in *Proc. Conf. Artif. Intell.*, 2020, pp. 10615–10622.

[20] X. Cheng, P. Wang, and R. Yang, “Depth estimation via affinity learned with convolutional spatial propagation network,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 108–125.

[21] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, “Non-local spatial propagation network for depth completion,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 120–136.

[22] Z. Yan et al., “RigNet: Repetitive image guided network for depth completion,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 214–230.

[23] L. Liu, Y. Liao, Y. Wang, A. Geiger, and Y. Liu, “Learning steering kernels for guided depth completion,” *IEEE Trans. Image Process.*, vol. 30, pp. 2850–2861, 2021.

[24] Y.-K. Huang et al., “S3: Learnable sparse signal superdensity for guided depth estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16701–16711.

[25] X. Liang and C. Jung, “Deep cross spectral stereo matching using multi-spectral image fusion,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5373–5380, Apr. 2022.

[26] H. Li and X.-J. Wu, “DenseFuse: A fusion approach to infrared and visible images,” *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2614–2623, May 2019.

[27] P. Sun, W. Zhang, H. Wang, S. Li, and X. Li, “Deep RGB-D saliency detection with depth-sensitive attention and automatic multi-modal fusion,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1407–1417.

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

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[30] L. Liu et al., “FCFR-Net: Feature fusion based coarse-to-fine residual learning for depth completion,” in *Proc. Conf. Artif. Intell.*, 2021, pp. 2136–2144.

[31] X. Li, W. Wang, X. Hu, and J. Yang, “Selective kernel networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 510–519.

[32] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[33] 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, 2013.

[34] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.

[35] 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. Automat.*, 2021, pp. 13656–13662.

[36] Y. Lin, T. Cheng, Q. Zhong, W. Zhou, and H. Yang, “Dynamic spatial propagation network for depth completion,” in *Proc. Conf. Artif. Intell.*, 2022, pp. 01–06.

[37] Y. Zhao, L. Bai, Z. Zhang, and X. Huang, “A surface geometry model for LiDAR depth completion,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4457–4464, Jul. 2021.

[38] W. Du, H. Chen, H. Yang, and Y. Zhang, “Depth completion using geometry-aware embedding,” *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 8680–8686.

[39] Y. Ke et al., “MDANet: Multi-modal deep aggregation network for depth completion,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 4288–4294.

[40] F. Mal and S. Karaman, “Sparse-to-dense: Depth prediction from sparse depth samples and a single image,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 4796–4803.

[41] A. Eldesokey, M. Felsberg, and F. S. Khan, “Confidence propagation through CNNs for guided sparse depth regression,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2423–2436, Oct. 2020.

[42] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.