Learning Complementary Correlations for Depth Super-Resolution With Incomplete Data in Real World

Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Guangyu Li, Jun Li, *Member, IEEE*, and Jian Yang, *Member, IEEE*

***Abstract*— Depth information is a significant ingredient to visually perceive the physical world. However, mainstream depth sensors, e.g., time-of-flight (ToF) cameras, often measure incomplete and low-resolution depth data, resulting in low-quality visual perception. In this article, we try to address a potentially valuable task, i.e., depth super-resolution (DSR) with incomplete data, which recovers dense and high-resolution depth map from incomplete and low-resolution one. To tackle this task, we introduce a novel incomplete DSR (IDSR) framework, including a primary branch for DSR to recover high-frequency details, and an auxiliary branch for depth completion (DC) to fill missing pixels. More importantly, we propose two modules, joint correlation learning (JCL) and iterative-cross (IC), to enhance the learning of complementary information flows between the two branches. The former module aims to learn the correlative relationships of the two branches, whilst the latter module adequately fuses higher level representations for more precise predictions. Extensive experiments show that our framework is effective and achieves the state-of-the-art performance on the real-world RGB-D-D and the synthetic NYUv2 datasets.**

***Index Terms*— Complementary information, depth super-resolution (DSR), iterative-cross (IC), joint correlation learning (JCL).**

I. Introduction

DEPTH information has received increasing attention in industry as it is an important ingredient to visually perceive the world in many applications, such as self-driving [1], [2], [3], scene analysis and understanding [4], [5], [6], 3-D vision [7], [8], and augmented/virtual reality [9], [10]. With the rapid development of mobile terminals, e.g., mobile phones, more and more depth sensors [11], [12], [13] have been embedded into these devices to help meet the needs of 3-D applications. In fact, the success of these applications heavily depends on both dense and high-resolution depth information [14], [15], [16], as current RGB cameras can easily capture super high-definition pictures. However, existing mainstream sensors, e.g., time-of-flight (ToF) camera [12], Intel RealSense [14], and Microsoft Kinect [15], usually measure incomplete and low-resolution depth maps, which are caused by the quality of sensors, long distance, material properties, and other various challenging environments. For example in Fig. 1, we can easily observe that: 1) the depth data from ToF lens is incomplete and 2) the low-resolution depth map mismatches with the high-resolution color image from RGB camera. These two observations lead to a challenging task, i.e., depth super-resolution (DSR) with incomplete data: given incomplete and low-resolution depth input, how to recover the dense and high-resolution one?

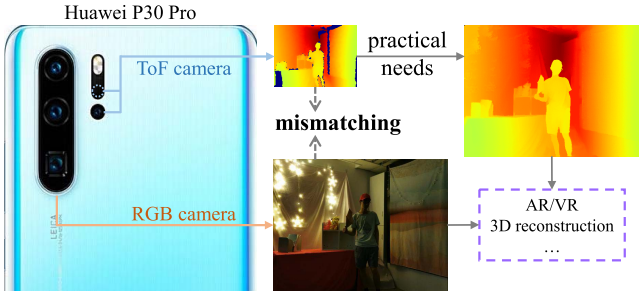


Fig. 1. Problem of DSR with incomplete data in mobile systems: 1) the depth map captured from ToF lens is incomplete and 2) the low-resolution depth does not match with the high-resolution color image.

Recently, a large number of methods concerned with DSR [10], [17], [18] and depth completion (DC) [19], [20], [21], [22] have been presented to recover high-resolution depth maps and complete the missing depth data, respectively. However, for one thing, depth SR often focuses on restoring high-resolution depth from low-resolution and dense input, which is not adept at dealing with incomplete data. For another thing, DC usually concentrates on recovering dense depth from sparse and same-resolution input, which has a limited command of handling high-resolution data. Therefore, it is difficult or insufficient for independent depth SR and completion to tackle the challenging task when simultaneously facing incomplete and low-resolution data from real world.

Manuscript received 29 January 2022; revised 25 June 2022; accepted 18 September 2022. This work was supported in part by the National Science Fund of China under Grant U1713208, Grant 62072242, and Grant 62006119; and in part by the Natural Science Foundation of Jiangsu under Grant BK20190444. The work of Xiang Li was supported by the Postdoctoral Innovative Talent Support Program of China under Grant BX20200168 and Grant 2020M681608. *(Corresponding authors: Guangyu Li; Jun Li.)*

The authors are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: guangyu.li2017@njust.edu.cn; junli@njust.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2022.3208330.

Digital Object Identifier 10.1109/TNNLS.2022.3208330

In this article, we develop a novel incomplete DSR (IDSR) framework for the challenging but very valuable task. We first introduce a primary SR branch for recovering high-resolution depth, and an auxiliary completion branch for filling the missing depth values. Furthermore, we propose the novel joint correlation learning (JCL) and iterative-cross (IC) modules to fuse complementary information flows between the two branches, since they can boost each other with more effective information, i.e., high-frequency details in primary branch and filled pixels in auxiliary branch. Specifically, the JCL module, which is embedded between the same-size middle-level layers of the two branches, consists of lightweight pixel-wise and channel-wise correlations to capture long-range contextual dependencies for further enhancing feature representations. The IC module conducts iterative connections between the different-size high-level output layers of the two branches to reuse their complementary information flows. These two modules can effectively handle the challenging task based on the proposed depth SR and completion architecture.

In summary, our main contributions are as follows.

1) Inspired by practical applications, we introduce a potentially valuable task, i.e., DSR with incomplete data, and decompose this task into depth SR and DC based on a dual-branch architecture.
2) We present the JCL and IC modules, to propagate the high-frequency depth details and filled depth values between our primary depth SR and auxiliary DC branches.
3) Extensive experiments show the effectiveness of our IDSR framework, which achieves the state-of-the-art performance on both the RGB-D-D and the NYUv2 datasets.

II. Related Work

A. Depth Super-Resolution

DSR approaches can be roughly classified into two categories: conventional learning based and deep learning based, most of which usually employ bicubic degradation to generate the low-resolution depth maps.

For conventional learning-based depth SR, without color image guided, the literatures [23], [24] apply Markov random field to recover high-resolution depth. However, such depth maps have blurry edges. To alleviate this problem, Ferstl *et al.* [25] present the total generalized variation strategy. With color image guided, such depth SR methods are proved effective [12], [26] for the recovery of depth edges. For example, Matsuo and Aoki [27] utilizes the auxiliary color image information to compute local tangent planes that are beneficial for high-quality depth predictions. Li *et al.* [28] presents a learning-based approach that constructs joint filters to selectively transfer salient structures which are consistent with both guidance and target images.

For deep learning-based depth SR, without color image guided, Riegler *et al.* [29] combines convolutional neural networks with total variation to produce high-resolution depth maps. Ruget *et al.* [30] develop a deep network to take advantage of multiple features extracted from a camera's histogram data. With color image guided, Jiang *et al.* [31] propose deep edge guided method that utilizes edge prediction subnetwork to facilitate the depth SR subnetwork. Peri and Xiong [32] present to jointly utilize spatial, intensity, and depth information of neighborhood pixels to reconstruct high-resolution depth map. Recently, many works [13], [17], [33] tend to input upsampled low-resolution depth maps, which have the same resolution as the high-resolution color images. Given such RGB-D pairs, joint image filtering methods [28], [34], [35], [36] are proposed to guide the detailed depth generations. Furthermore, Ye *et al.* [37] design a progressive multibranch aggregation network to gradually recover desired depth guided by color images. Additionally, Yao *et al.* [38] employ texture-depth transformer to guide the primary depth SR branch.

In recent months, the first large-scale real-world depth dataset, which is named RGB-D-D, has been constructed by [13]. This dataset provides both low-resolution and high-resolution depth maps from real world. Based on this dataset, we simultaneously predict high-dense depth maps from low-incomplete ones, while existing depth SR works mainly pay attention to low-resolution but dense input.

B. Depth Completion

Based on a mass of popular DC papers, we split them into two groups: unguided methods (without color image) and guide approaches (with color image).

For unguided DC, researchers only use sparse depth maps to recover dense predictions without considering color images. For example, Uhrig *et al.* [19] introduce sparsity invariant convolution to deal with the sparse depth data. Jaritz *et al.* [39] propose the sparse-to-dense framework to complete the sparse depth input. Van Gansbeke *et al.* [40] present to predict uncertainty map for robust DC. In particular, Lu *et al.* [41] also take sparse depth as the only input, but they use color image as auxiliary supervision when training.

For guided DC, multisensor information fusion becomes a clear trend recently. For example, Ma *et al.* [42] directly feed the concatenation of a sparse depth and the color image into networks, contributing to better performance. Qiu *et al.* [43] propose to estimate surface normals as the intermediate representation to benefit depth recovery. Xu *et al.* [20] utilize surface normals and confidence maps to further enhance their model. Tang *et al.* [21] and Yan *et al.* [44] present dynamic convolution to filter sparse depth map according to the corresponding color image context. Zhu *et al.* [45] propose uncertainty-driven loss functions to strengthen the robustness of DC and tackle its uncertainty. In recent months, ACMNet [46], FCFRNet [47], and PENet [48] utilize an independent color image branch to gradually guide the depth recovery. Besides, spatial propagation networks (SPNs) is an integral part of DC. They [49], [50] employ color images to study the affinity matrix to refine feature extraction. For example, Park *et al.* [22] and Xu *et al.* [51] greatly promote the depth prediction nearby object boundaries by recurrent nonlocal and dynamic SPN design. Finally, unsupervised DC

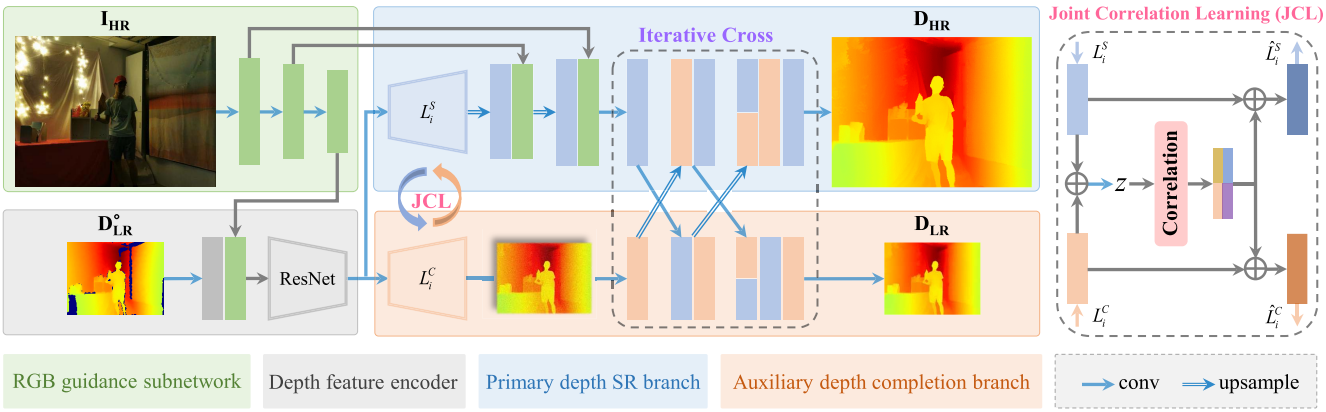


Fig. 2. Overview of our incomplete depth SR framework. "JCL" refers to our JCL module and "correlation" is illustrated in Fig. 3.

works [52], [53], [54], [55], [56], [57], [58] also contribute to the comprehensive development of this domain.

Different from them which focus on recovering the same resolution depth maps with the input, we directly generate high-dense depth predictions from the low-incomplete ones using real-world data captured from mobile terminals.

C. Depth-Correlative Joint-Task Learning

Numerous joint-task learning methods concerned with depth have greatly benefited depth SR, such as image guided depth SR [59], [60], depth SR and denoising [61], and depth SR and deblurring [62]. Recently, an auxiliary depth estimation task [63], only deployed during the training stage, is constructed to boost depth SR. Furthermore, BridgeNet [33] proposes to jointly model depth SR and monocular depth estimation to relieve the negative effect when fusing multi-modal data. Different from them, when considering that the real-world depth data are incomplete and low-resolution, we make an attempt to jointly learn depth SR and DC to boost the depth SR task, i.e., learning their complementary correlative information between parallel dual branches.

III. Incomplete Depth Super-Resolution

In this section, we introduce our IDSR framework in detail. We first demonstrate an overview of the network architecture, then elaborate the JCL and IC modules, and finally present the loss function.

A. Overview of Network Architecture

The architecture of the proposed method is illustrated in Fig. 2, which is an example of $\times 4$ SR. The input consists of a low-resolution depth map $D^{\circ}_{\mathrm{LR}} \in \mathbb{R}^{1\times H/4\times W/4}$ with incomplete pixels, and a high-resolution color image $I_{\mathrm{HR}} \in \mathbb{R}^{3\times H\times W}$. H and W refer to height and width, respectively.

1) *RGB guidance subnetwork:* It employs three continuous convolutional layers to encode the color image input I_{HR}, each of which contains two convolutions with kernels 3×3 and 1×1, generating three different-size image features with resolutions $H \times W$, $H/2 \times W/2$, and $H/4 \times W/4$, severally.

2) *Auxiliary DC branch:* It consists of an encoder–decoder network. The encoder is the ResNet34 and the decoder is composed of upsampling and convolutional layers. The depth input D°_{LR} is first encoded by a 3×3 convolution with stride 1 to obtain its initial mapping. Then we combine the mapping with the $H/4 \times W/4$ image feature and feed them into the auxiliary DC branch. Finally, we perform a 1×1 convolution with stride 1 to produce the feature representation of DC.

3) *Primary depth SR branch:* The primary and auxiliary branches share the same encoder. The decoder of the primary branch is similar to that of the auxiliary one. Differently, behind the decoder of the primary branch in $4\times$ case, there exists two additional pixel shuffle layers to produce high-resolution depth features with the guidance of the color image features. Finally, a 1×1 convolution with stride 1 is employed to generate the feature representation of depth SR.

4) *JCL and IC modules:* The primary depth SR and auxiliary DC branches predict high-dense D_{HR} and low-dense D_{LR}, respectively. Between the primary and auxiliary branches, we employ our JCL module to learn their complementary information flows of the middle-level features. On the output sides of the two branches, our IC strategy is conducted to further enhance representation of the high-level features. We now give the details of the JCL and IC modules below.

B. Joint Correlation Learning (JCL)

JCL is proposed to learn complementary features, including high-frequency depth details and filled depth values in depth SR and DC branches, respectively. The module is shown in the right of Fig. 2, where the correlation design (see Fig. 3) consists of pixel-wise and channel-wise correlations. As contextual dependencies can provide depth holes with similar references, the pixel-wise correlation is designed to capture the nonlocal contexts [64], [65], [66] in each channel map. Further, since every channel map can be regarded as an object-specific response and all channel maps are associated or referred with each other [67], the channel-wise correlation is designed to integrate the associated features among all channel maps for learning similar contextual references.

At the ith ($1 \le i \le 4$) upsampling stage in the depth SR and completion branches, our JCL_i module combines the inputs

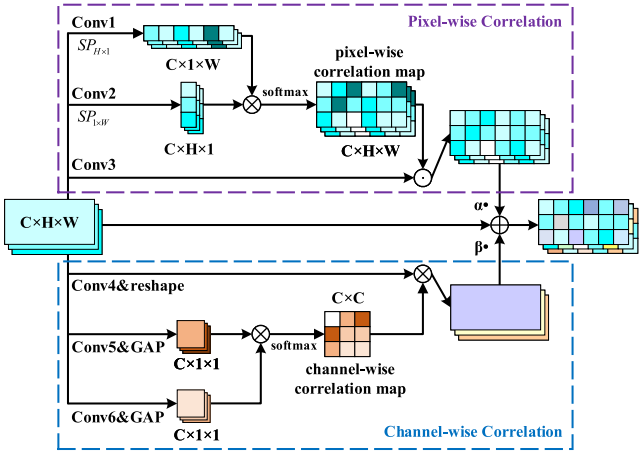


Fig. 3. Overview of our correlation design. $SP_{H\times1}$: $H \times 1$ strip average pooling. $SP_{1\times W}$: $1 \times W$ strip average pooling. $\otimes$: matrix multiplication. $\odot$: element-wise multiplication. GAP: global average pooling.

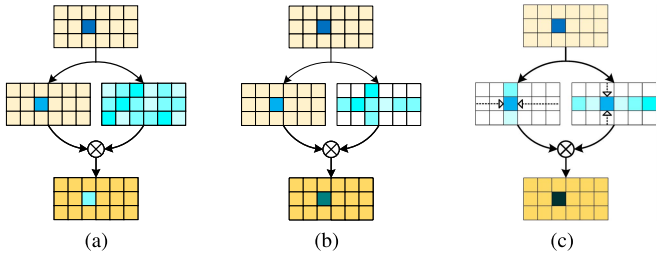


Fig. 4. Comparisons of three pixel-wise correlation methods for one position (e.g., dark blue). All feature sizes are $H \times W$. (a) Nonlocal block's [64] complexity is $O((H \times W)^2)$. (b) Criss-cross block's [65] complexity is $O((H \times W)(H + W - 1))$. (c) Our JCL_p block, whose complexity is only $O(H \times W)$, creates its correlation map by employing strip average pooling.

L_i^S and L_i^C into a single convolutional layer to generate their feature mapping z, then feeds it into correlation operation, and finally employs residual connections between the inputs and the outputs of correlation. The updating rules of $\widehat{L}_i^S$ and $\widehat{L}_i^C$ are described as

$$\begin{aligned} \widehat{L}_i^S &= L_i^S + \text{Cor}(z) \\ \widehat{L}_i^C &= L_i^C + \text{Cor}(z) \end{aligned} \tag{1}$$

where $\text{Cor}(z) = z + a + b$, $z = h(L_i^C + L_i^S)$, $h(\cdot)$ is a 1×1 convolution, and $\text{Cor}(\cdot)$ represents our correlation operation in Fig. 3. Next, we will describe their details, and compute a and b, which are defined in (2) and (3), respectively.

1) Pixel-Wise Correlation: Here, we introduce the pixel-wise correlation (JCL_p) that is illustrated in the dashed purple box of Fig. 3. Its key is to quickly learn the nonlocal contextual dependencies. Unfortunately, the existing nonlocal approaches usually suffer from the high computational cost. For example, the complexity of the nonlocal block [64] in Fig. 4(a) is $O((H \times W)^2)$. The complexity of the criss-cross block [65] in Fig. 4(b) is still high, $O((H \times W)(H+W-1))$, although it generates the correlation map by applying criss-cross method. Our goal is to introduce an efficient and fast pixel-wise correlation in Fig. 4(c) for modeling long-range contextual relationships over local features, whose complexity is only $O(H \times W)$.

Given the input z, we employ three 1×1 convolutional operations to map it to features $\{h_1(z), h_2(z), h_3(z)\} \in \mathbb{R}^{C\times H\times W}$. Then, $1 \times W$ and $H \times 1$ strip average pooling operations are applied into obtain the long-range representations, $f_j(h_1(z)) \in \mathbb{R}^{C\times H\times 1}$ and $f_k(h_2(z)) \in \mathbb{R}^{C\times 1\times W}$, where $1 \le j \le W$, $1 \le k \le H$, $f_i(\cdot)$ and $f_j(\cdot)$ are the horizontal and vertical average pooling operations, respectively. Next, a pixel-wise correlation map is produced using matrix multiplication and softmax function, and we multiply the pixel-wise correlation map by $h_3(z)$ to obtain the final spatial feature correlation $a \in \mathbb{R}^{C\times H\times W}$. The computing process is defined as

$$a_{jk} = \alpha \frac{\exp\big(f_j(h_1(z)) \otimes f_k(h_2(z))\big) h_3\big(z_{jk}\big)}{\sum_{j,k}^{N} \exp\big(f_j(h_1(z)) \otimes f_k(h_2(z))\big)} \tag{2}$$

where α is a learnable parameter (torch.nn.Parameter()) that can be automatically optimized during training. $N = H \times W$, $\otimes$ is the matrix multiplication, and a_{jk} denotes the pixel correlation response at the position of jth row and kth column.

2) Channel-Wise Correlation: Here, we present the channel-wise correlation (JCL_c) shown in the dashed blue box of Fig. 3. Our goal is to explore the internal dependencies among different channels for enhancing the similar contextual references. As shown in Fig. 5, DANet [67] obtains channel-wise correlation map by $(C \times HW) \otimes (HW \times C) \Rightarrow C \times C$, while we perform $(C \times 1) \otimes (1 \times C) \Rightarrow C \times C$ with global average pooling. Hence, our JCL_c can produce channel-wise correlation maps with less cost than DANet.

Given the input z, we employ three 1×1 convolutions to map it to three features $\{h_4(z), h_5(z), h_6(z)\} \in \mathbb{R}^{C\times H\times W}$, and apply two global average pooling functions $g_1(\cdot)$ and $g_2(\cdot)$ to obtain nonlocal representations, $g_1(h_5(z)) \in \mathbb{R}^{C\times 1\times 1}$ and $g_2(h_6(z)) \in \mathbb{R}^{C\times 1\times 1}$. Then, we reshape them to $\mathbb{R}^{C\times 1}$ and transpose one of them to $\mathbb{R}^{1\times C}$. Next, the channel-wise correlation map is produced by employing matrix multiplication and softmax function. Finally, we conduct matrix multiplication between the channel-wise correlation map and $h_4(z)$ to obtain the final channel feature correlation. The computing process is defined as

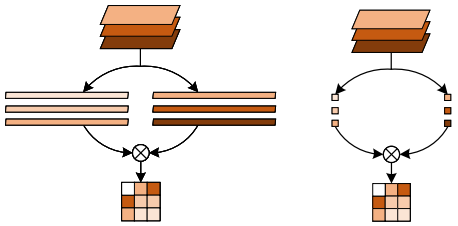
$$b_{jk} = \beta \frac{\exp(g_1(h_5(z)) \otimes g_2(h_6(z))) h_4\big(z_{jk}\big)}{\sum_{j,k}^{N} \exp(g_1(h_5(z)) \otimes g_2(h_6(z)))} \tag{3}$$

where β is a learnable parameter (torch.nn.Parameter()) that can be automatically optimized during training.

C. Iterative-Cross (IC) Module

The depth SR and DC branches contain high-frequency depth details and reliable depth reference respectively, which should be further utilized to refine the recovery of high-resolution depth. Hence, we present the IC module to fuse high-level representations of the two branches and learn their complementary information flows.

We assume that the channels and iterations of the outputs in the two branches are c and n severally. Concretely, after obtaining the two output features $F_1^S \in \mathbb{R}^{c\times h\times w}$ and $F_1^C \in \mathbb{R}^{c\times (h/4)\times (w/4)}$ from the two branches, we predict the final



(a) (b)

Fig. 5. Comparisons of channel-wise correlation. (a) DANet subblock [67]. (b) JCL_c block generates correlation maps by employing global average pooling.

Algorithm 1 Iterative-Cross Module

Require: Initial feature $F_1^{\mathcal{S}}$, $F_1^{\mathcal{C}}$, hyper-parameter N, K.

1: **while** not done **do**
2: $c \in \{2^k | k = 3, 4, \ldots, K\}$
3: **for** all c **do**
4: **for** $n \leftarrow 1, 2, \ldots, N$ **do**
5: $F_{n+1}^{\mathcal{S}} = F_n^{\mathcal{S}} \| Up(F_n^{\mathcal{C}})$
6: $F_{n+1}^{\mathcal{C}} = F_n^{\mathcal{C}} \| \text{Down}(F_n^{\mathcal{S}})$
7: **end for**
8: $D_{\text{HR}} = h(F_{n+1}^{\mathcal{S}})$
9: $D_{\text{LR}} = h(F_{n+1}^{\mathcal{C}})$
10: **end for**
11: **end while**

high-resolution depth $D_{\text{HR}} \in \mathbb{R}^{c \times h \times w}$ and filled depth $D_{\text{LR}} \in \mathbb{R}^{c \times (h/4) \times (w/4)}$. The whole process is stated in Algorithm 1. N and K represent the maximum values of iterations and channels. $\|$, $Up(\cdot)$, and $\text{Down}(\cdot)$ denote concatenation, upsampling, and Downsampling operations, respectively. In one single loop, we obtain $F_{n+1}^{\mathcal{S}}$ and $F_{n+1}^{\mathcal{C}}$ by transforming $F_n^{\mathcal{S}}$ and $F_n^{\mathcal{C}}$ in lines 5 and 6. Then in lines 8 and 9 we employ convolutions to map $F_{n+1}^{\mathcal{S}}$ and $F_{n+1}^{\mathcal{C}}$, generating the final low-resolution and dense D_{LR} and high-resolution and dense D_{HR} depth predictions.

D. Loss Function

In this article, the ℓ_1 norm is employed to train our model. Specifically, given the predicted depths D_{HR} and D_{LR} in Algorithm 1 and ground-truth depth GT, the loss is written as

$$\mathcal{L}_{\text{IDSR}} = \sum_{p \in P_v} \left| D_{\text{HR}\,p} - GT_{\text{HR}\,p} \right| + \sum_{\bar{p} \in \bar{P}_v} \left| D_{\text{LR}\,\bar{p}} - GT_{\text{LR}\,\bar{p}} \right| \quad (4)$$

where P_v ($\bar{P}_v$) represents the set of all valid pixels and p ($\bar{p}$) refers to one pixel in the ground-truth depth GT_{HR} (GT_{LR}).

IV. Experiment

In this section, we first introduce the two popular datasets and implementation details of our method. Then, we compare with state-of-the-art (SoTA) approaches, and present the quantitative and qualitative results on both the real-world RGB-D-D [13] and the synthetic NYUv2 [15] datasets. Finally, extensive ablation studies are performed to verify the effectiveness of each module. Note that all the related DSR methods are retrained FROM SCRATCH on the IDSR task. We thank all authors for their released codes.

A. Datasets and Metrics

RGB-D-D is the first real-world IDSR dataset, which consists of 4811 pairs. 2215 pairs are used for training and 405 pairs for testing. Each pair contains a high-resolution (512 × 384) color image from Huawei P30 Pro, a real-world low-resolution (192 × 144) depth map from the low-power ToF camera equipped in the mobile phone, and a high-resolution (512 × 384) depth map from industrial ToF camera. Considering depth holes caused by occlusion effect of projection processing and some low-reflection objects, e.g., glass surface and infrared absorbing surface, over-segmentation algorithm [68], and colorization method [69] are used to produce dense high-resolution ground-truth depth. Please note again that the raw low-resolution depth maps captured from Huawei P30 Pro ToF are incomplete due to the equipment itself and the complex environment.

NYUv2 dataset [15] is composed of 464 indoor scenes, containing paired color images and depth maps captured by Microsoft Kinect. Following the common split setting [34], we use the first 1000 pairs for training and test on the last 449 pairs, whose resolutions are both 640 × 480. We further validate the generalization ability of our method on this dataset.

Metric: We choose the root mean square error (RMSE) as the primary metric to be consistent with related depth SR works [10], [18], [70], which is defined as

$$\text{RMSE} = \left(\sum_{p \in P_v} \left\| D_{\text{HR}\,p} - GT_{\text{HR}\,p} \right\|^2 \right)^{1/2}. \quad (5)$$

B. Implementation Details

Our IDSR is implemented on the Pytorch [71] framework, and is particularly trained with four TITAN RTX GPUs. For RGB-D-D dataset, the total training epoch is 60 and the initial learning rate is 0.0005, which reduces to half every 40 k iterations. For NYUv2 dataset, the initial learning rate is 0.0005 and reduces to half every 80 k iterations. The training process is stopped after 200 epochs.

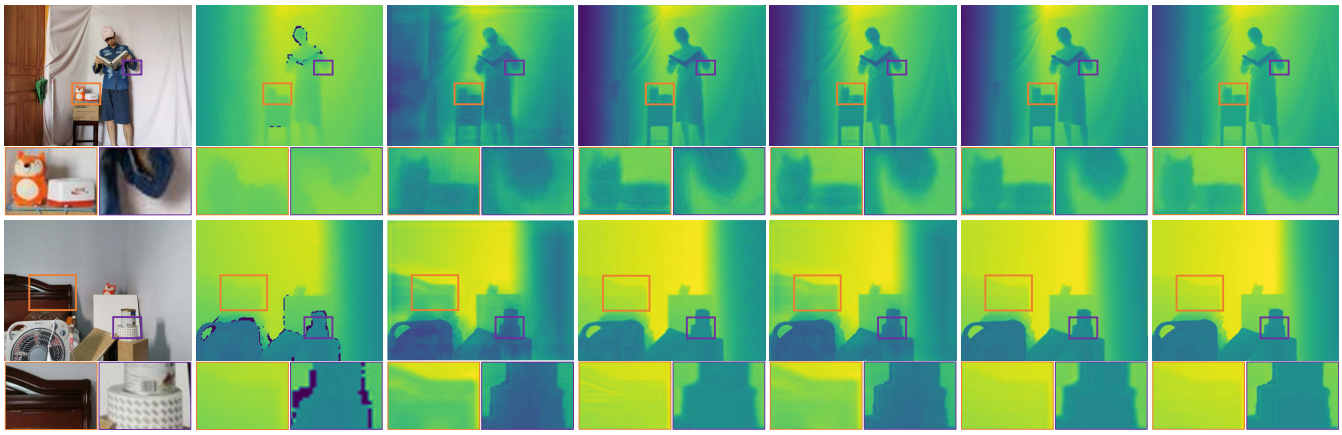
C. Comparison With SoTA Methods

In this subsection, we compare our IDSR with SoTA works with publicly available codes, including the single-task (depth SR) methods, i.e., DJF [34], DJFR [28], PAC [35], DKN [36], FDKN [36], JIIF [72], and FDSR [13], and the joint-task approaches, i.e., SVLRM [62] (depth SR and depth deblurring), and BridgeNet [33] (depth SR and depth estimation). DJF and DJFR propose a learning-based approach for constructing joint filters. PAC introduces a pixel-adaptive convolution to replace the standard convolution, in which the filter weights are spatially varying. DKN and FDKN present to instead learn explicitly sparse and spatially variant kernels. JIIF formulates the guided depth SR as a neural implicit

TABLE I

QUANTITATIVE COMPARISONS ON THE RGB-D-D AND THE NYUV2 DATASETS. RAW: RAW INCOMPLETE AND LOW-RESOLUTION DEPTH INPUT. JBF/COLOR: RAW DEPTH INPUT PREPROCESSED WITH JOINT BILATERAL FILTERING/COLORIZATION METHOD TO FILL DEPTH HOLES. DEPTH VALUES ARE MEASURED IN CENTIMETER. THE BEST AND THE SECOND-BEST RESULTS ARE MARKED IN BOLD AND UNDERLINE, RESPECTIVELY

	Method	RGB-D-D						NYUv2						reference
		real-world			downsample (80%)			downsample (65%)			downsample (5%)			
		raw	JBF	color	×4	×8	×16	×4	×8	×16	×4	×8	×16	
single depth SR	DJF	35.9	9.5	7.9	16.6	17.8	25.6	10.6	13.7	21.8	46.0	86.9	99.8	ECCV16
	DJFR	24.7	9.3	8.0	15.7	16.9	19.1	10.1	13.4	22.8	54.7	83.6	100.0	PAMI19
	PAC	27.3	10.2	8.1	25.6	28.1	30.7	5.2	8.4	12.1	13.1	21.0	34.0	CVPR19
	DKN	46.2	8.6	7.4	24.9	26.0	33.2	6.5	10.9	27.0	79.3	95.7	106.1	IJCV20
	FDKN	40.9	9.3	7.5	28.9	29.6	31.9	9.4	16.1	33.9	83.4	93.2	113.5	IJCV20
	JIIF	23.4	8.1	6.7	22.6	27.3	33.9	10.2	16.3	19.7	24.2	33.5	55.7	ACM MM21
	FDSR	21.6	7.4	5.5	26.7	28.7	30.2	6.3	9.8	14.9	17.1	28.5	52.8	CVPR21
joint	SVLRM	54.6	10.2	8.1	15.6	25.1	33.7	5.3	8.3	14.1	20.3	38.9	65.4	CVPR19
	Bridge	17.1	7.8	5.7	10.3	11.9	13.2	5.3	7.4	11.1	12.6	19.8	**24.1**	ACM MM21
	IDSR	**11.5**	**5.7**	**4.9**	**4.5**	**6.1**	**9.5**	**3.2**	**4.3**	**6.8**	**8.5**	**13.7**	24.4	-



(a) (b) (c) (d) (e) (f) (g)

Fig. 6. Visual results of incomplete depth SR on RGB-D-D dataset. (b)–(e) are retrained using real-world depth input (192 × 144 → 512 × 384). "IDSR-down" means that we train IDSR in ×4 downsampling manner. (a) RGB. (b) Raw Depth. (c) FDSR. (d) Bridge. (e) IDSR. (f) IDSR-down. (g) GT.

image interpolation problem with a novel joint implicit image function. FDSR provides a fast depth SR baseline, in which the high-frequency component is adaptively decomposed from RGB image to guide the depth SR. SVLRM designs a new joint filter based on a spatially variant linear representation model to jointly resolve depth SR, deblurring, denoising, etc. BridgeNet presents a joint learning network of depth SR and depth estimation which reinforce each other. We retrain all of them on the realistic RGB-D-D dataset in real-world and downsampling manners. To further verify the generalization capability of IDSR, we also conduct experiments on the commonly used NYUv2 dataset in downsampling manner [33]. Numerical results are reported in Table I, and visual comparisons are shown in Figs. 6 and 7.

For one thing, the real-world manner indicates the depth inputs of networks are captured from real world. Furthermore, besides the raw real-world data, we also evaluate the proposed method on the preprocessed real-world data. Since the missing values in depth maps are represented as zero, which would bring heavily noisy depth inputs to existing depth SR methods when conducting the commonly used bicubic strategy. For sufficient comparison, following [60] and [69], we respectively use the joint bilateral filtering (JBF) and colorization (color) to fill the depth holes, and then feed such dense depth into the depth SR methods to see their differences.

For another thing, the downsampling manner takes as input synthetic low-resolution depth maps degraded (e.g., bicubic) from ground-truth annotations [13], [34]. For the RGB-D-D dataset, the density of raw depth maps captured from ToF camera is more than 80% [13]. Therefore, referring to the common pattern of generating valid pixels in DC field [22], [50], we randomly sample 80% valid regions, producing the final incomplete and low-resolution depth input. Likewise, for the NYUv2 dataset, we randomly sample 65% [13] valid regions to generate the desired depth input.

1) Real-World Manner on RGB-D-D: As shown in the column of "raw" in Table I, our IDSR method outperforms the second-best BridgeNet by 32.7%, and is superior to other approaches with large margins. Fig. 6 demonstrates that our model can recover sharper boundaries and more accurate depth

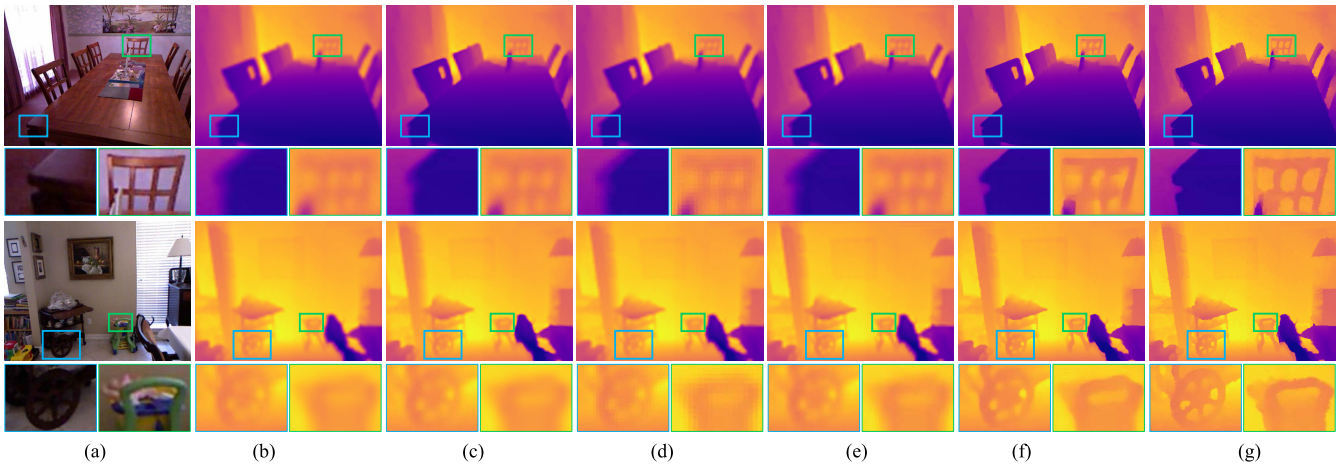


Fig. 7. Visual comparisons of $\times 8$ incomplete depth SR on NYUv2 dataset. Density of the depth input is 65%. From left to right: (a) RGB image, (b) DJFR, (c) DKN, (d) JIIF, (e) FDSR, (f) our IDSR, and (g) GT.

TABLE II
COMPARISONS BETWEEN OUR PARALLEL DUAL-BRANCH IDSR AND THE TANDEM DC+DSR ON NYUV2 WITH 5% DENSITY

DC+DSR	$\times 4$	$\times 8$	$\times 16$
GuideNet+FDSR	15.1	23.3	28.6
GuideNet+Bridge	14.3	22.7	27.8
NLSPN+FDSR	14.6	23.4	27.1
NLSPN+Bridge	13.2	21.9	26.0
RigNet+FDSR	13.8	21.6	26.5
RigNet+Bridge	13.0	20.1	25.4
IDSR (our)	**8.5**	**13.7**	**24.4**

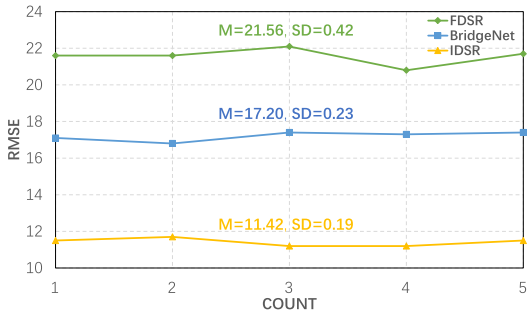


Fig. 8. Comparisons of the mean value (M) and standard deviation (SD).

values than others. In addition, as illustrated in the columns of "JBF and color" in Table I, all approaches can predict more accurate depth results when taking preprocessed data as input. In all three cases, our method still has outstanding performance. It is worth noting that SVLRM, which linearly represents the target image with the guidance image, performs not well on the real-world dataset. The main reason is probably that this linear representation cannot describe very accurately the real data distribution.

2) Downsampling Manner on RGB-D-D: As reported in the column of "downsampling (80%)" in Table I, our IDSR is much more outstanding among those SoTA approaches, surpassing the second best BridgeNet by average 44.1% in terms of the three scaling factors.

On the whole, when comparing the results of the two manners in Table I on RGB-D-D dataset, we find it is more difficult for abovementioned methods to handle the raw depth data in real world. We can observe from (e) and (f) of Fig. 6 that our IDSR trained in downsampling manner produces smoother and more precise depth predictions than that trained in real-world manner. The qualitative and quantitative evidences indicate the traditional downsampling synthetic strategy cannot simulate very well the practical correspondence between the low-resolution and high-resolution depth pairs, which is a sore point existed in the domain of depth SR for a long time.

3) Downsampling Manner on NYUv2: The column of "downsampling (65%)" in Table I shows that, our IDSR achieves superior or competitive performance among all methods, surpassing the second best BridgeNet by average 20.7%. As illustrated in Fig. 7, benefiting from the robust joint-task learning and the complementary-correlation enhancement, our IDSR framework can recover more clear object boundaries and more accurate structure-detailed depth values.

In addition, to further test our model's stability, we randomly sample 5% valid pixels, producing the extremely incomplete depth input. As shown in the column of "downsampling (5%)" in Table I, the DJF, DJFR, DKN, and FDKN all fail to handle the very incomplete depth input, recovering high-resolution depth with large errors. On the contrary, although the low-resolution depth is sparse, our model can still generate better depth results with much lower errors.

It is worth noting that the BridgeNet also has a better performance than previous depth SR methods due to the joint learning of depth estimation and depth SR. When encountering the incomplete depth data, its depth estimation subnetwork still can predict proper depth values from color images, which can weaken the negative impact of bad prediction in its depth SR subnetwork. The auxiliary depth estimation and DC strategies probably provide potential guidance for researchers to handle the challenging but valuable IDSR task.

TABLE III

ABLATION STUDIES OF ×8 INCOMPLETE DEPTH SR CASE. DSR AND DC REPRESENT THE PRIMARY DEPTH SR AND THE AUXILIARY DC SUBNETWORKS, RESPECTIVELY. $\text{JCL}_{p/c}$: JOINT LEARNING WITH PIXEL/CHANNEL-WISE CORRELATION. IC: ITERATIVE-CROSS MODULE

	DSR	DC	JCL_p	JCL_c	IC	RGB-D-D	NYUv2
①	✓					14.95	5.16
②	✓	✓				13.42	4.80
③	✓	✓	✓			12.72	4.63
④	✓	✓		✓		12.24	4.51
⑤	✓	✓	✓	✓		11.86	4.43
⑥	✓	✓	✓	✓	✓	**11.50**	**4.29**

4) Parallel IDSR Versus Tandem DC + DSR: As shown in Table II, given the sparse and low-resolution depth input, we use the latest SoTA DC methods RigNet [44], NLSPN [22], and GuideNet [21], to fill missing depth first, and then conduct the latest SoTA DSR approaches FDSR [13] and BridgeNet [33] to produce high-resolution predictions. It can find that our parallel dual-branch IDSR still significantly outperforms such tandem dc + DSR strategy.

5) Mean Value and Standard Deviation: At last, we conduct five separate experiments to validate the robustness. As shown in Fig. 8, we compare the mean values (*M*) and standard deviations (SD) of FDSR, BridgeNet, and IDSR. We observe that our IDSR achieves the lowest *M* and SD. The SD of IDSR is 0.04 and 0.23 superior to BridgeNet and FDSR, respectively. This result further verifies the superior robustness of our IDSR over the competing methods.

Overall, the qualitative and quantitative results demonstrate that our IDSR method has strong generalization capability and robust stability. More detailed experiments and visualizations can be found in the supplementary material.

D. Ablation Study

Here, we further verify the key components of IDSR, including the joint-task architecture, JCL and IC modules. We report the ×8 incomplete depth SR results on raw RGB-D-D and NYUv2 in real-world and downsampling (65%) manners, respectively.

1) All Key Components: As shown in Table III, the 1st and 2nd rows are the results of single depth SR branch (DSR) and joint-task architecture (DSR + dc). We can find that the auxiliary dc branch, which provides the DSR branch with dense and valid depth reference, brings great improvement to the task of depth SR with incomplete data. Next, our pixel-wise and channel-wise JCL together promote the model's performance by aggregating the complementary correlations of the depth SR and DC tasks. Then, the IC module further benefits our model via adequately fusing higher level feature representations. Finally, we combine all components, obtaining the lowest errors. The visual process of gradual refinement can be found in Fig. 9. Obviously, as more and more components are deployed, the depth predictions become more clear and complete, especially nearby the region of object boundaries.

TABLE IV

COMPLEXITY COMPARISON OF OUR JOINT CORRELATION LEARNING MODULE WITH EXISTING RELEVANT APPROACHES ON THE RAW RGB-D-D DATASET. "CORR" DENOTES CORRELATION, "TIME" REFERS TO THE TIME CONSUMPTION OF FORWARD INFERENCE (*ms*), "PI" REPRESENTS THE PIXEL-WISE CORRELATION, AND "CH" INDICATES THE CHANNEL-WISE CORRELATION

Method	Corr	GFLOPs ↓	Time ↓	RMSE ↓
IDSR w/o JCL	-	0	0	13.42
+Non-local	pi	6.91	51	12.68
+Criss-cross	pi	3.26	35	12.70
$+\text{DA}_p$	pi	7.35	53	**12.66**
$+\text{JCL}_p$ (our)	pi	**2.07**	**14**	12.72
$+\text{DA}_c$	ch	4.01	31	**12.22**
$+\text{JCL}_c$ (our)	ch	**2.08**	**20**	12.44
$+\text{DA}_p+\text{DA}_c$	pi+ch	27.38	76	12.05
$+\text{JCL}_p+\text{JCL}_c$	pi+ch	**9.50**	**28**	**11.86**

TABLE V

ABLATION STUDY OF DIFFERENT CHANNELS (c) AND ITERATIONS (n) IN OUR ITERATIVE-CROSS MODULE. MODEL IS TRAINED ON THE RAW RGB-D-D DATASET IN REAL-WORLD MANNER. METRIC IS RMSE

IC		n		
		1	2	3
c	8	13.37	13.28	13.30
	16	13.26	13.18	13.28
	32	13.25	13.06	13.24
	64	13.20	**12.97**	13.19

TABLE VI

COMPARISONS OF DIFFERENT FUSION METHODS IN 8× CASE, INCLUDING "ADD" [73], "CONCAT" [74], "LINEAR" [75], AND OUR "IC." "NONE" DENOTES THAT WE DIRECTLY OUTPUT DEPTH PREDICTIONS WITHOUT FUSION. THE METRIC IS RMSE (CM)

Dataset	none	add	concat	linear	IC (our)
RGB-D-D	11.86	11.83	11.75	11.64	**11.50**
NYUv2	4.43	4.41	4.35	4.36	**4.29**

2) Joint Correlation Learning (JCL) Module: As reported in Table IV, just with little sacrifice of accuracy, our JCL_p and JCL_c have lower GFLOPs and faster forward inference speeds than nonlocal [64], criss-cross [65], and DANet [67] blocks. Specifically, JCL_p increases GFLOPs by 2.07 and slows down the test time by 14 ms but reduces RMSE by 0.7. JCL_c reduces RMSE from 13.42 to 12.44 with 2.08 and 20 ms damage in GFLOPs and test time, severally. When combining "JCL_p" and "JCL_c," our model achieves the best result among these methods, which reduces RMSE by 1.56. All of these facts show the effectiveness of our JCL design.

a) Iterative-Cross module (IC): Table V shows the influence of different channels and iterations in the IC module. We can roughly observe that the error gradually drops with the increase of c. When $c = 64$ and $n = 2$, our model obtains the lowest error. All of these evidences indicate that our IC module is effective.

Table VI compares our IC fusion strategy with other similar methods, i.e., addition [73], concatenation [74], and linear cross-stitch network [75], which are embedded in the same

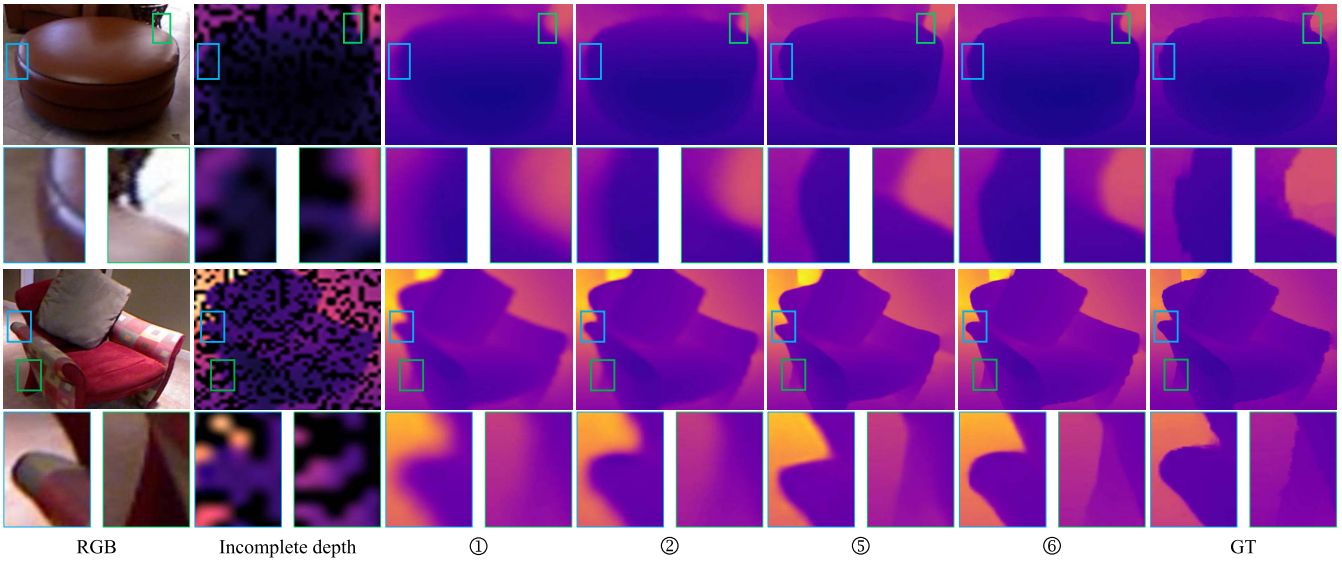


Fig. 9. Corresponding visual comparisons of our IDSR's key components reported in Table III. Results are obtained from ×8 incomplete depth SR case. Black points in incomplete depth inputs refer to the region where depth values are missing.

layers as IC. Among these approaches, our IC achieves the lowest errors both on RGB-D-D and NYUv2 datasets, verifying that the IC module is effective for the challenging task to learn complementary information flows between the depth SR and DC branches.

V. Conclusion

In this article, we seek to address a challenging but potentially valuable task inspired by realistic applications, i.e., DSR with incomplete data, which recovers dense and high-resolution depth maps from incomplete and low-resolution ones. We propose a cross-task IDSR framework to deal with the difficult data, exploring for the first time to utilize a parallel auxiliary DC branch to provide the primary depth SR branch with dense and valid depth reference. Then, we design the JCL module and IC strategy to adequately learn complementary information flows between the two branches at different stages. Benefiting from these improvements, our method achieves outstanding performance consistently on the real-world RGB-D-D and the synthetic NYUv2 datasets. Additionally, with the development of camera hardware in the future, dense and high-resolution depths may be directly generated. By then, the actual applications such as virtual fit, immersive gaming, and panoramic home design will be significantly enhanced. These evidences further show the potential value of our framework.

A. Limitation and Future Work

There are two major limitations in current version.

1) The IDSR framework is not lightweight. The main parameters come from the depth feature encoder and the corresponding two decoders.

2) The strip pooling in our pixel-wise correlation module is fixed (i.e., $H \times 1$, $1 \times W$), leading to limited mapping capability although it significantly reduces the complexity.

Therefore, there are two elements in our future work: 1) designing a lightweight IDSR with high efficiency and accuracy and 2) developing a flexible strip pooling (i.e., $H \times N$, $N \times W$) to enhance its mapping capability.

References

[1] C. Häne *et al.*, "3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection," *Image Vis. Comput.*, vol. 68, pp. 14–27, Dec. 2017.

[2] Z. Zhou, X. Fan, P. Shi, and Y. Xin, "R-MSFM: Recurrent multi-scale feature modulation for monocular depth estimating," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12777–12786.

[3] K. Wang *et al.*, "Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16055–16064.

[4] W. Zhuo, M. Salzmann, X. He, and M. Liu, "Indoor scene structure analysis for single image depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 614–622.

[5] H. Jiang, G. Larsson, M. M. Greg Shakhnarovich, and E. Learned-Miller, "Self-supervised relative depth learning for urban scene understanding," in *Proc. ECCV*, 2018, pp. 19–35.

[6] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4106–4115.

[7] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2485–2494.

[8] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

[9] A. Dey, G. Jarvis, C. Sandor, and G. Reitmayr, "Tablet versus phone: Depth perception in handheld augmented reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Nov. 2012, pp. 187–196.

[10] X. Song *et al.*, "Channel attention based iterative residual learning for depth map super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5631–5640.

[11] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1623–1630.

[12] D. Ferstl, C. Reinbacher, R. Ranftl, M. Ruether, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 993–1000.

[13] L. He *et al.*, "Towards fast and accurate real-world depth super-resolution: Benchmark dataset and baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9229–9238.

[14] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 567–576.

[15] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. ECCV*. Springer, 2012, pp. 746–760.

[16] D. Scharstein *et al.*, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. GCPR*, 2014, pp. 31–42.

[17] X. Song, Y. Dai, and X. Qin, "Deep depth super-resolution: Learning depth super-resolution using deep convolutional neural network," in *Proc. ACCV*, 2016, pp. 360–376.

[18] O. Voynov *et al.*, "Perceptual deep depth super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5653–5663.

[19] 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.

[20] 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.

[21] 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.

[22] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, "Non-local spatial propagation network for depth completion," in *Proc. ECCV*, 2020, pp. 120–136.

[23] O. Mac Aodha, N. D. Campbell, A. Nair, and G. J. Brostow, "Patch based synthesis for single depth image super-resolution," in *Proc. ECCV*, 2012, pp. 71–84.

[24] J. Xie, R. S. Feris, and M.-T. Sun, "Edge-guided single depth image super resolution," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 428–438, Jan. 2016.

[25] D. Ferstl, M. Ruther, and H. Bischof, "Variational depth superresolution using example-based edge representations," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 513–521.

[26] J. Lu and D. Forsyth, "Sparse depth super resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2245–2253.

[27] K. Matsuo and Y. Aoki, "Depth image enhancement using local tangent plane approximations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3574–3583.

[28] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Joint image filtering with deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1909–1923, Aug. 2019.

[29] G. Riegler, M. Rüther, and H. Bischof, "ATGV-Net: Accurate depth super-resolution," in *Proc. ECCV*, 2016, pp. 268–284.

[30] A. Ruget, S. McLaughlin, R. K. Henderson, I. Gyongy, A. Halimi, and J. Leach, "Robust super-resolution depth imaging via a multi-feature fusion deep network," *Opt. Exp.*, vol. 29, no. 8, pp. 11917–11937, 2021.

[31] Z. Jiang, H. Yue, Y.-K. Lai, J. Yang, Y. Hou, and C. Hou, "Deep edge map guided depth super resolution," *Signal Process., Image Commun.*, vol. 90, Jan. 2021, Art. no. 116040.

[32] C. Peri and Y. Xiong, "Image guided depth super-resolution for space-warp in XR applications," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2021, pp. 1–6.

[33] Q. Tang *et al.*, "BridgeNet: A joint learning network of depth map super-resolution and monocular depth estimation," in *Proc. ACM MM*, 2021, pp. 2148–2157.

[34] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep joint image filtering," in *Proc. ECCV*. Cham, Switzerland: Springer, 2016, pp. 154–169.

[35] H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, and J. Kautz, "Pixel-adaptive convolutional neural networks," in *Proc. CVPR*, Jun. 2019, pp. 11166–11175.

[36] B. Kim, J. Ponce, and B. Ham, "Deformable kernel networks for joint image filtering," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 579–600, Feb. 2021.

[37] X. Ye *et al.*, "PMBANet: Progressive multi-branch aggregation network for scene depth super-resolution," *IEEE Trans. Image Processing*, vol. 29, pp. 7427–7442, 2020.

[38] C. Yao, S. Zhang, M. Yang, M. Liu, and J. Qi, "Depth super-resolution by texture-depth transformer," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.

[39] M. Jaritz, R. D. Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with CNNs: Depth completion and semantic segmentation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 52–60.

[40] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy LiDAR completion with RGB guidance and uncertainty," in *Proc. MVA*, May 2019, pp. 1–6.

[41] 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. 11306–11315.

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

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

[44] Z. Yan, K. Wang, X. Li, Z. Zhang, J. Li, and J. Yang, "RigNet: Repetitive image guided network for depth completion," 2021, *arXiv:2107.13802*.

[45] Y. Zhu, W. Dong, L. Li, J. Wu, X. Li, and G. Shi, "Robust depth completion with uncertainty-driven loss functions," 2021, *arXiv:2112.07895*.

[46] 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.

[47] L. Liu *et al.*, "FCFR-Net: Feature fusion based coarse-to-fine residual learning for depth completion," in *Proc. AAAI*, 2021, vol. 35, no. 3, pp. 2136–2144.

[48] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "PENet: Towards precise and efficient image guided depth completion," in *Proc. ICRA*, May/Jun. 2021, pp. 13656–13662.

[49] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," in *Proc. ECCV*, 2018, pp. 103–119.

[50] 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*, 2020, pp. 10615–10622.

[51] Z. Xu, H. Yin, and J. Yao, "Deformable spatial propagation networks for depth completion," in *Proc. ICIP*, Oct. 2020, pp. 913–917.

[52] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (DDP) from single image and sparse range," in *Proc. CVPR*, Jun. 2020, pp. 3353–3362.

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

[54] B. Krauss, G. Schroeder, M. Gustke, and A. Hussein, "Deterministic guided LiDAR depth map completion," 2021, *arXiv:2106.07256*.

[55] A. Wong, X. Fei, B.-W. Hong, and S. Soatto, "An adaptive framework for learning unsupervised depth completion," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3120–3127, Apr. 2021.

[56] A. Wong, S. Cicek, and S. Soatto, "Learning topology from synthetic data for unsupervised depth completion," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1495–1502, Apr. 2021.

[57] A. Wong and S. Soatto, "Unsupervised depth completion with calibrated backprojection layers," in *Proc. ICCV*, Oct. 2021, pp. 12747–12756.

[58] D. Teutscher, P. Mangat, and O. Wasenmüller, "PDC: Piecewise depth completion utilizing superpixels," in *Proc. ITSC*, Sep. 2021, pp. 2752–2758.

[59] L. Zhao, H. Bai, J. Liang, B. Zeng, A. Wang, and Y. Zhao, "Simultaneous color-depth super-resolution with conditional generative adversarial networks," *Pattern Recognit.*, vol. 88, pp. 356–369, Apr. 2019.

[60] S. Gu, W. Zuo, S. Guo, Y. Chen, C. Chen, and L. Zhang, "Learning dynamic guidance for depth image enhancement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3769–3778.

[61] J. Xie, R. Feris, S.-S. Yu, and M.-T. Sun, "Joint super resolution and denoising from a single depth image," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1525–1537, Sep. 2015.

[62] J. Pan, J. Dong, J. S. Ren, L. Lin, J. Tang, and M.-H. Yang, "Spatially variant linear representation models for joint filtering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1702–1711.

[63] B. Sun, X. Ye, B. Li, H. Li, Z. Wang, and R. Xu, "Learning scene structure guidance via cross-task knowledge transfer for single depth super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7792–7801.

[64] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. CVPR*, Jun. 2018, pp. 7794–7803.

[65] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

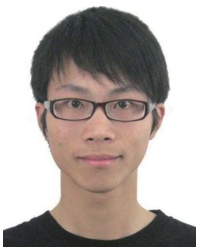
[66] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. ECCV*. Cham, Switzerland: Springer, 2020, pp. 402–419.
[67] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.
[68] H. T. Nguyen, M. Worring, and R. V. D. Boomgaard, "Watersnakes: Energy-driven watershed segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 3, pp. 330–342, Mar. 2003.
[69] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proc. ACM SIGGRAPH Papers (SIGGRAPH)*, 2004, pp. 689–694.
[70] Y. Xiao, X. Cao, X. Zhu, R. Yang, and Y. Zheng, "Joint convolutional neural pyramid for depth map super-resolution," 2018, *arXiv:1801.00968*.
[71] A. Paszke et al., "Automatic differentiation in PyTorch," in *Proc. Adv. Neural Inf. Process. Syst.-Autodiff Workshop*, 2017, pp. 1–4.
[72] J. Tang, X. Chen, and G. Zeng, "Joint implicit image function for guided depth super-resolution," in *Proc. ACM MM*, 2021, pp. 4390–4399.
[73] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture," in *Proc. ACCV*, 2016, pp. 213–228.
[74] X. Wang, D. F. Fouhey, and A. Gupta, "Designing deep networks for surface normal estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 539–547.
[75] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3994–4003.

Zhiqiang Yan received the B.E. degree from the Nanjing University of Science and Technology, Nanjing, Jiangsu, China, in 2018, where he is currently pursuing the Ph.D. degree.

His research interests include computer vision, especially on depth estimation and completion self-driving, robotic vision, and related 3-D visual perception.

Kun Wang received the B.E. degree in smart grid information engineering from the Qingdao University of Science and Technology, Qingdao, China, in 2017. He is currently pursuing the Ph.D. degree in control science and engineering with the Nanjing University of Science and Technology, Nanjing, China.

His research interests include computer vision, especially on depth estimation and completion, 3-D reconstruction, and related topics.



Xiang Li received the B.S. degree in computer science the Ph.D. degree in pattern recognition and intelligent system from the Nanjing University of Science and Technology, Nanjing, Jiangsu, China, in 2013 and 2020, respectively.

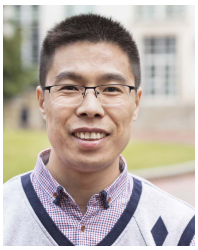
His research interests include computer vision, pattern recognition, data mining, and deep learning.

Zhenyu Zhang received Ph.D. degree from the Department of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2020, supervised by J. Yang.

In 2019, he was a Visiting Student with the MHUG Group, University of Trento, Trento, Italy, supervised by Nicu Sebe. His research interests include computer vision, photography, and reconstruction.

Guangyu Li received the B.S. degree from the China University of Mining and Technology, Xuzhou, China, in 2008, the M.S. degree from Tongji University, Shanghai, China, in 2011, and the Ph.D. degree from the University of Paris-Sud, Paris, France, in 2015.

He is currently working as an Assistant Professor with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing, China. His current research interests include wireless networks and computer vision.



Jun Li (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, China, in 2015.

From October 2012 to July 2013, he was a Visiting Student with the Department of Statistics, Rutgers University, Piscataway, NJ, USA. From December 2015 to October 2018, he was a Post-Doctoral Associate with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. From November 2018 to October 2019, he was a Post-Doctoral Associate with the Institute of Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests are computer vision, machine learning, and data mining.

Dr. Li has served as a SPC/PC Member of ICML/NeurIPS/ICLR/AAAI and a reviewer for more than ten international journals, such as IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Jian Yang (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002.

In 2003, he was a Post-Doctoral Researcher with the University of Zaragoza, Zaragoza, Spain. From 2004 to 2006, he was a Post-Doctoral Fellow with Biometrics Centre, The Hong Kong Polytechnic University, Hong Kong. From 2006 to 2007, he was a Post-Doctoral Fellow with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Chang-Jiang Professor with the School of Computer Science and Technology, NUST. He is the author of more than 200 scientific articles in pattern recognition and computer vision. His papers have been cited more than 6000 times in the Web of Science and 15 000 times in the Scholar Google. His research interests include pattern recognition, computer vision and machine learning.

Dr. Yang is a fellow of IAPR. He is/was an Associate Editor of *Pattern Recognition*, *Pattern Recognition Letters*, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Neurocomputing*.